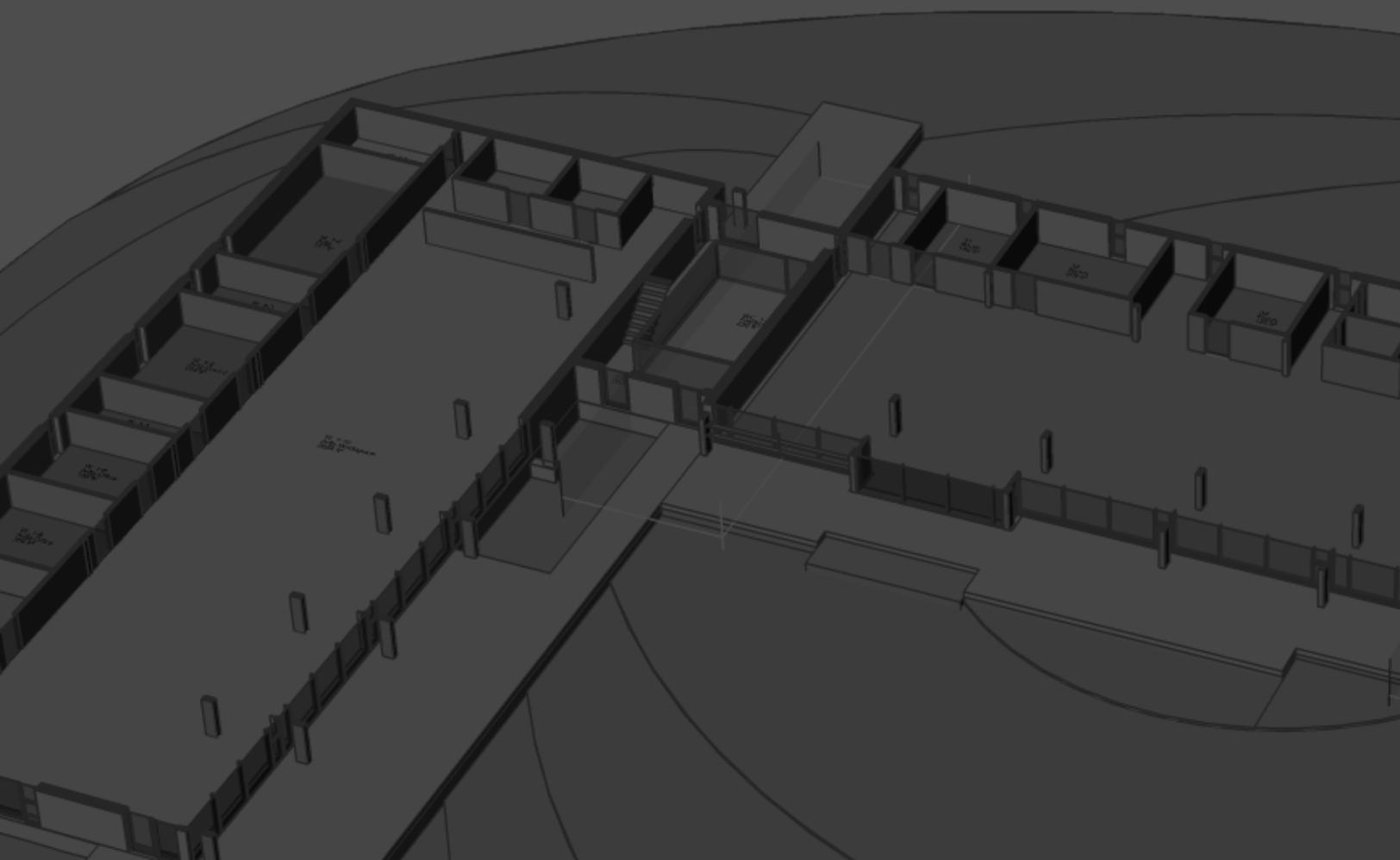


UPDATED FOR V17



INSIDE BRICSCAD®

- A Quick Tour Through BricsCAD
- Navigating the BricsCAD Interface
- Setting Up A New Drawing
- Creating Your First Drawing
- Adding Details to Drawings
- Making Changes to Drawings
- Adding Notes and Dimensions
- Bills of Material
- Modeling 2D Regions and Booleans
- Direct 3D Modeling & Editing
- Dimensional & Geometric Constraints



Payment Information

This book is covered by copyright. As the owner of the copyright, upFront.eZine Publishing, Ltd. gives you permission to make one print copy. You may not make any electronic copies, and you may not claim authorship or ownership of the text or figures herein.

By Email

Acrobat PDF format: **\$20.00**

Allow for a 17MB download

PayPal

To pay by PayPal, send payment to the account of grabowski@telus.net at www.paypal.com.

PayPal accepts funds in US, Euro, Yen, Canadian, and 100+ other currencies.

Check or Money Order

We can accept checks from the following regions of the world:

- US funds drawn on a bank with address in the USA
- Canadian funds drawn on a bank with a Canadian address (includes GST).

Make cheque payable to 'upFront.eZine Publishing'

Please mail your payment to:

"Inside BricsCAD V17"

upFront.eZine Publishing, Ltd.

34486 Donlyn Avenue

Abbotsford BC

V2S 4W7 Canada

Visit the *Inside BricsCAD* Web site at www.upfrontezine.com/lib8. At this Web page, editions of this book are available for BricsCAD V8 through V17.

Purchasing an ebook published by upFront.eZine Publishing, Ltd. entitles you to receive the *upFront.eZine* newsletter weekly. To subscribe to our "The Business of CAD" newsletter separately, send an email to grabowski@telus.net.

Copyright Information

Copyright © 2016 by upFront.eZine Publishing, Ltd.

All rights reserved worldwide.

This eighth edition is based on BricsCAD V17

23 November 2016

Updated 26 January 2017

Technical Writer **Ralph Grabowski**

All brand names and product names mentioned in this book are trademarks or service marks of their respective companies. Any omission or misuse (of any kind) of service marks or trademarks should not be regarded as intent to infringe on the property of others. The publisher recognizes and respects all marks used by companies, manufacturers, and developers as a means to distinguish their products.

This book is sold as is, without warranty of any kind, either express or implied, respecting the contents of this book and any disks or programs that may accompany it, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither the publisher, authors, staff, or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by this book.

Table of Contents

1 — A Quick Tour Through BricsCAD	1
How to Start BricsCAD V17	3
Starting BricsCAD on Windows	3
Windows Vista and 7	3
Windows 8.....	4
Windows 10.....	4
Starting BricsCAD on Linux	5
Starting BricsCAD on Mac MacOS.....	5
BricsCAD V17 User Interface	6
Getting Started	8
The BricsCAD Window.....	8
A Basic Tour of the User Interface.....	9
Crosshair and Arrow Cursors	9
Command Bar	10
All About Command Prompts.....	11
Undoing What You've Done: U.....	13
Seeing What You Did Before: Command History	14
Typing Less: Aliases and AutoComplete	15
UCS Icon.....	17
Online Help	18
Exiting BricsCAD	18
What's New in BricsCAD V17.....	19
What's New in Generated Views	35
What's New in Rendering.....	36
SUPPORTED GRAPHICS BOARDS	44

2 — Navigating the BricsCAD Interface 45

Above the Drawing Area	46
Title Bar	47
Menu Bar.....	48
Toolbars	49
Toolbar Buttons and Macros	50
The Standard Toolbar	51
Flyouts	51
Droplists.....	52
Changing Properties.....	53
Other Panels	56
USER INTERFACE ELEMENTS OF PANELS	57
Ribbon Tabs and Panels	59
Drawing Tabs.....	59
In the Drawing Area.....	60
Shortcut Menus.....	61
quad cursor	61
Quick Properties	62
Scroll Bars	63
Prompt Menu.....	63
Look-From Control.....	63
The UCS Icon in 2D Drafting, 3D Modeling, and DUCS.....	65
Tips Widget.....	67
Below the Drawing Area	69
Layout Tabs.....	69
Status Bar	70
Summary	72

3 — Setting Up A New Drawing..... 73

Before You Begin	75
Starting A New Drawing.....	75
Default Settings.....	77
Selecting the Linear Units.....	77
WAYS TO START NEW DRAWINGS	78

Selecting the Angle Style	80
Choosing the Plot Style	81
Setting Entity Properties	82
Finishing the Wizard.....	82
Additional Important Settings.....	83
Setting the Drawing Limits	83
Accessing and Changing Variables	85
Changing the Snap and Grid Spacing.....	87
Creating Layers	89
QUICK SUMMARY OF LAYERS	90
Naming Layers	91
Saving Drawings.....	93
QUICK SUMMARY OF THE SAVEAS COMMAND.....	94
Making Backups Automatic	94
QUICK SUMMARY OF ALTERNATIVE SAVE FORMATS.....	96
Summary	96

4 — Creating Your First Drawing 97

Reopening Drawings	99
Drawing the Lot Boundary	100
Planning the Next Steps.....	102
Changing Layers	103
Drawing the House Outline	104
QUICK SUMMARY OF THE PLINE COMMAND.....	105
Direct Distance Entry	106
Dynamic Input	106
QUICK SUMMARY OF ENTITY SNAP MODES.....	107
Moving the House into Position	110
Starting on the Driveway	111
Finishing the Driveway	112
Mirroring Entities.....	113
Putting Drawings to Paper	115
QUICK SUMMARY OF PRINT PREVIEW	117
Solving Printing Problems.....	118

Generating PDFs of Drawings.....	119
Specifying PDF Output Options.....	120
Summary	121

5 — Adding Details to Drawings 123

Dividing the Lot	125
Smoothing Polylines.....	128
QUICK SUMMARY OF THE PEDIT COMMAND.....	129
Grips Editing	130
QUICK SUMMARY OF EDITING WITH GRIPS	132
Hatching the Lawn	133
Creating Symbols	136
Drawing Circles.....	136
Zooming in Real Time	136
QUICK SUMMARY OF DRAWING CIRCLES	137
Creating Arrays.....	138
Making Blocks.....	140
Adding Many More Trees	141
Drawing the Pond	142
QUICK SUMMARY OF DRAWING ELLIPSES.....	144
Array Paths.....	145
Drawing Splines.....	145
QUICK SUMMARY OF SPLINE.....	146
Zooming to Objects.....	147
Paving Stones from Polygons	147
QUICK SUMMARY OF POLYGON	148
Arraying Along a Path	148
QUICK SUMMARY OF ARRAYPATH.....	149
Summary	150

6 — Making Changes to Drawings	151
Changing the Look of Lines.....	153
Loading Linetypes	153
Changing Properties.....	155
QUICK SUMMARY OF PROPERTIES PANEL.....	156
Changing the Linetype Scale.....	157
Selecting Entities by Other Methods.....	158
Selecting Entities by Their Properties	158
Selecting an Entity by Tabbing.....	159
QUICK SUMMARY OF THE LENGTHEN COMMAND	160
Changing Line Lengths	160
Stretching the Pond	161
QUICK SUMMARY OF SELECTION SET OPTIONS.....	162
Moving Entities	164
Adding the Fence.....	164
Listing Data.....	166
Summary	167
7 — Adding Notes and Dimensions.....	169
Adding Notes to Drawings	172
Determining the Size of Text.....	173
QUICK SUMMARY ON CALCULATING TEXT HEIGHTS.....	174
Creating Text Styles	175
Simple Text	176
QUICK SUMMARY OF TEXT JUSTIFICATION MODES	177
Changing Text	178
Placing Rotated Text.....	178
Placing Multiple Lines of Text.....	180
Searching and Replacing Text.....	180
Placing Dimensions in Drawings.....	181
Preparing the Drawing for Dimensioning	181
Dimensioning the Yard	182
Vertical and Baseline Dimensions	184

Baseline Dimensioning.....	184
Aligned and Radial Dimensions	185
Radial Dimensions.....	186
Annotatively-Scaled Text and Dimensions	186
Using Annotative Scaling.....	188
A FEW EXTRA THINGS ABOUT ANNOTATIVE SCALING	192
Summary	193

8 — Bills of Material..... 195

About Attribute Data.....	197
Creating Blocks with Attributes	198
Drawing Blocks.....	200
Defining Attributes.....	200
Adding More Attributes.....	202
Combining Entities and Attributes into Blocks.....	203
Inserting Blocks with Attributes.....	205
Alternatives to the Insert Command.....	206
Exporting Data from Drawings.....	207
Data Extraction	207
Importing Data Files into Spreadsheets.....	213
Placing Data in Drawings as Tables	215
Summary	216

9 — Modeling 2D Regions and Booleans 217

About Regions	219
How to create Regions.....	219
Tutorial: Creating Boundaries.....	220
QUICK SUMMARY OF BOUNDARY OPTIONS	221
Boolean Operations	225
Tutorial: Creating a Waffle Shape.....	225
QUICK SUMMARY OF BOOLEAN OPERATIONS	226

Measuring Regions.....	232
About Mass Properties.....	232
Summary	233

10 — Direct 3D Modeling & Editing 235

About 3D Solid Models 237

Planning Ahead.....	237
---------------------	-----

ELEMENTS OF 3D MODELS.....238

Planning Ahead by Deconstructing 3D.....	239
--	-----

Traditional Solid Modeling Commands..... 239

Traditional Extrusion Method.....	240
-----------------------------------	-----

Preparing the Drawings	240
------------------------------	-----

Modeling a Box.....	242
---------------------	-----

QUICK SUMMARY OF EXTRUDE COMMAND243

3D View Rotation	245
------------------------	-----

Extrusion by Direct Modeling.....	246
-----------------------------------	-----

Applying Dynamic UCS	246
----------------------------	-----

Extruding Directly with the Quad Cursor	248
---	-----

Subtracting Solids To Make Holes	250
--	-----

PushPull Modeling.....	251
------------------------	-----

QUICK SUMMARY OF LOOKFROM COMMAND252

Aligning the UCS.....	254
-----------------------	-----

Pulling, Instead of Pushing.....	255
----------------------------------	-----

Filleting 3D Solids.....	259
--------------------------	-----

Joining Parts with Union.....	261
-------------------------------	-----

Generating 2D Drawings 262

QUICK SUMMARY OF VIEWBASE OPTIONS263

Slicing Sections And Adding Detailed Views.....	265
---	-----

Resizing Views	265
----------------------	-----

Making Section Views	265
----------------------------	-----

QUICK SUMMARY OF VIEWSECTION OPTIONS.....267

Creating Detail Views	268
-----------------------------	-----

QUICK SUMMARY OF VIEWDETAIL OPTIONS.....270

Summary	271
----------------------	------------

11 — Dimensional & Geometric Constraints 273

Working with Constraints 275

About Dimensional Constraints275

QUICK SUMMARY OF DIMENSIONAL CONSTRAINTS276

Using Dimensional Constraints277

QUICK SUMMARY OF 2D GEOMETRIC CONSTRAINTS280

About Geometric Constraints281

Identifying Geometric Constraints281

Using Geometric Constraints282

QUICK SUMMARY OF 3D CONSTRAINTS285

Summary 287

APPENDICES

A — Concise Summary of Command Aliases 289

BricsCAD Aliases Sorted by Alias Name	290
BricsCAD Aliases Sorted by Command Name.....	294

B — Concise Summary of System Variables and Settings 297

C — Concise Summary of Command Names 317

Chapoo Commands.....	322
Dimension Commands	323
Dimensional Constraint Commands	324
Direct Modeling Commands.....	324
Geometric Constraint Commands.....	326
Layer Commands	328
Sheet Metal Commands.....	334
Vba Commands.....	336
ViewBase Commands.....	336

CHAPTER 1

A Quick Tour Through BricsCAD

Welcome to BricsCAD V17!

You want to know how to create drawings with BricsCAD, and this book shows you how to — in as little as a day! But before doing any kind of drafting, you really should take a tour of the user interface to learn your way around BricsCAD. Even if you know other CAD programs, it may be useful for you to skim this chapter to take note of the areas in which BricsCAD might operate differently from what you expect.

Here you learn how to start this popular 2D/3D CAD program, take a tour through its user interface, and then get your feet wet by drawing a few lines.

IN THIS CHAPTER

- Starting BricsCAD V17
- Becoming familiar with parts of the user interface
- Understanding the crosshair cursor, command bar, auto-complete, and UCS icon
- Drawing lines
- Reversing errors
- Accessing online help

KEY TERMS IN THIS CHAPTER

Button — executes associated command when clicked

Cursor — provides feedback from the operating system and from BricsCAD

Flipscreen — switches between the drawing window and text window

Flyout — shows a secondary toolbar when clicked

Icon — represents commands pictorially

Layout — defines how drawings are plotted

Pickbox — specifies the points being picked (selected)

Right-click — involves pressing the right mouse button to display context-sensitive (shortcut) menus

Toolbar — collects buttons into a single, useful strip

USEFUL ABBREVIATIONS

Alt	Alternate key on PCs
Cmd	Command key on Macs
Ctrl	Control key on PCs
F	Function key
U	Undoes the last command or option
UCS	User-defined coordinate system

NEW COMMANDS

Command	Shortcut*	Menu Selection**	Ribbon
Help	? or F1	Help Help	Home Help Help
Line	L	Draw Line	Draw Draw Line
Quit	Alt+F4	File Exit	...
TextScr	F2	View Prompt History Window	...
Undo	Ctrl+Z	Edit Undo	...
UcsIcon

* F1 means function key F1

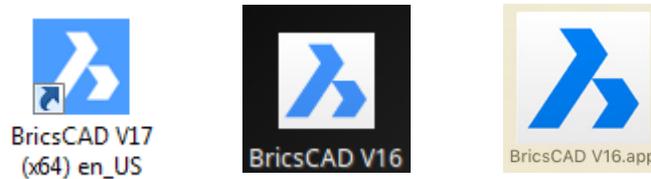
Alt+F4 means hold down the Alt key, and then press function key F4.

** The vertical bar separates menu selections. **Draw | Line** means: from the Draw menu, select the Line item.

How to Start BricsCAD V17

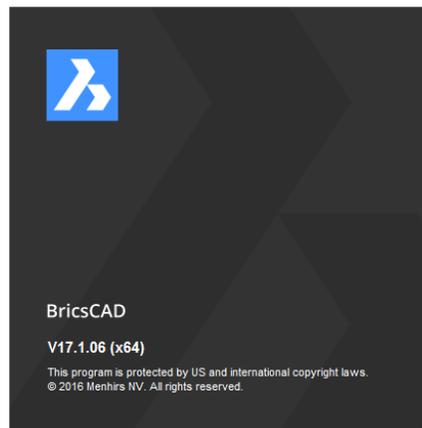
If BricsCAD is not yet set up on your computer, do so. To run BricsCAD, your computer must be operating recent releases of Windows, just about any recent dialect of Linux, or MacOS on Macs.

To start BricsCAD, double-click the **BricsCAD** icon found on the computer desktop.



*Left to right: BricsCAD icon on Windows, Linux, and Mac
(BricsCAD was available as V16 for Linux at time of writing)*

Depending on the speed of your computer, it can take from 10 to 30 seconds to load BricsCAD. During this time, a “splash screen” appears and disappears as BricsCAD starts up.



STARTING BRICSCAD ON WINDOWS

BricsCAD V17 works with Windows Vista and newer.

Windows Vista and 7

In Windows Vista and 7, you can start the program from the task bar using the following steps:

1. Click the task bar's **Start**  icon.
2. Choose **All Programs**.
3. Select **Bricsys**, followed by the **BricsCAD V17** folder, and then click on **BricsCAD V17**.

The exact name you see depends on the language version you downloaded. For example, “BricsCAD V17 (x64) en_US” is the name of the 64-bit program for English speakers in the US dialect.

Windows 8

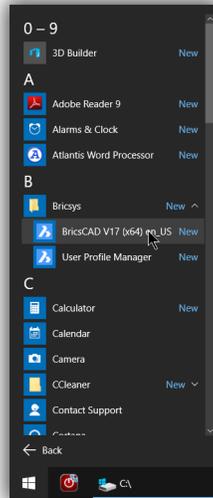
In Windows 8.x, you make these moves:

1. If necessary, switch to the **Start** screen.
2. In the Start screen, tap on the **BricsCAD V17** icon.

Windows 10

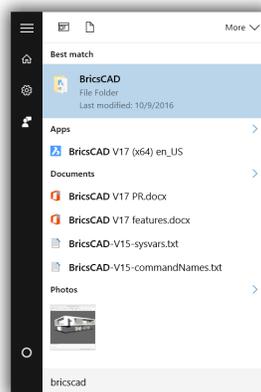
Under Windows 10 starting BricsCAD reverts more to like it started with Windows 7:

1. Tap the **Start** button.
2. Choose **All Apps**.
3. In the **B** menu, tap **Bricsys** folder, and then tap the **BricsCAD V17** item.



If the item is not visible in the menu, then follow these steps:

1. Tap the **Start** button.
2. Start typing “briscad”
3. When you see **Bricsys V17**, tap it.



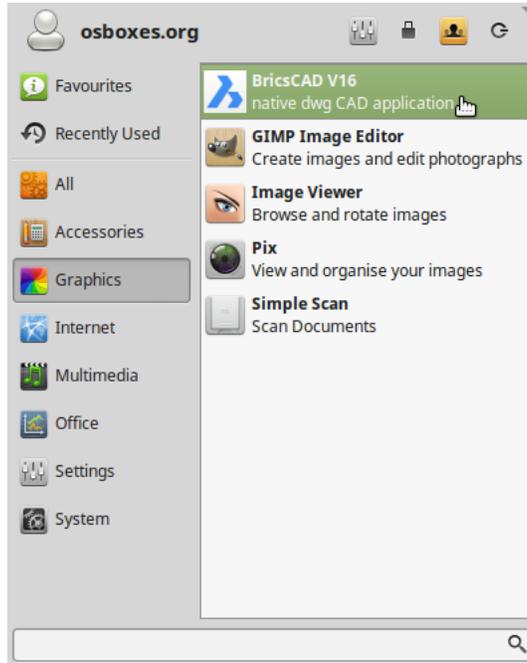
TIPS Instructions in this book specific to Linux and Mac are shown in gray text.

No matter the operating system, BricsCAD looks pretty much the same in each one, as illustrated on the following pages.

STARTING BRICSCAD ON LINUX

BricsCAD works with just about any recent release of Linux, but is specifically supported on Fedora, OpenSuse, and Ubuntu. To start the program, follow these steps:

1. Click the task bar's **Main Menu** button.
2. Choose **Graphics**.
3. Click on **BricsCAD**.



STARTING BRICSCAD ON MAC MACOS

BricsCAD works with recent releases of MacOS on Mac computers, 10.8 or higher. (MacOS is the new name for OS X.) On the dock, click the **BricsCAD V17** icon:



If you do not see the icon there, then follow these steps:

1. In the dock, open the **Application** folder.



2. Find the **BricsCAD V17** icon, and then click it.

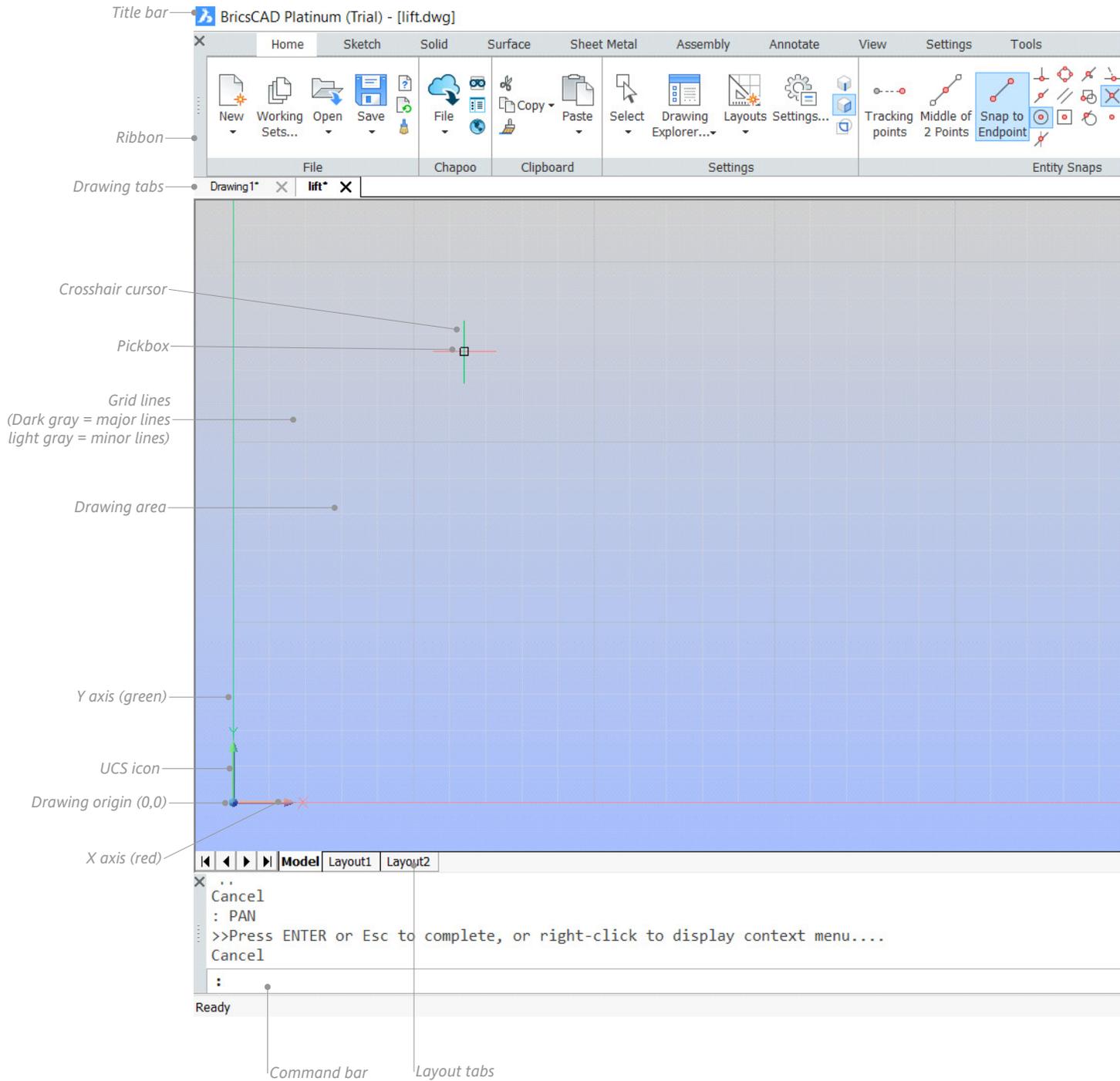


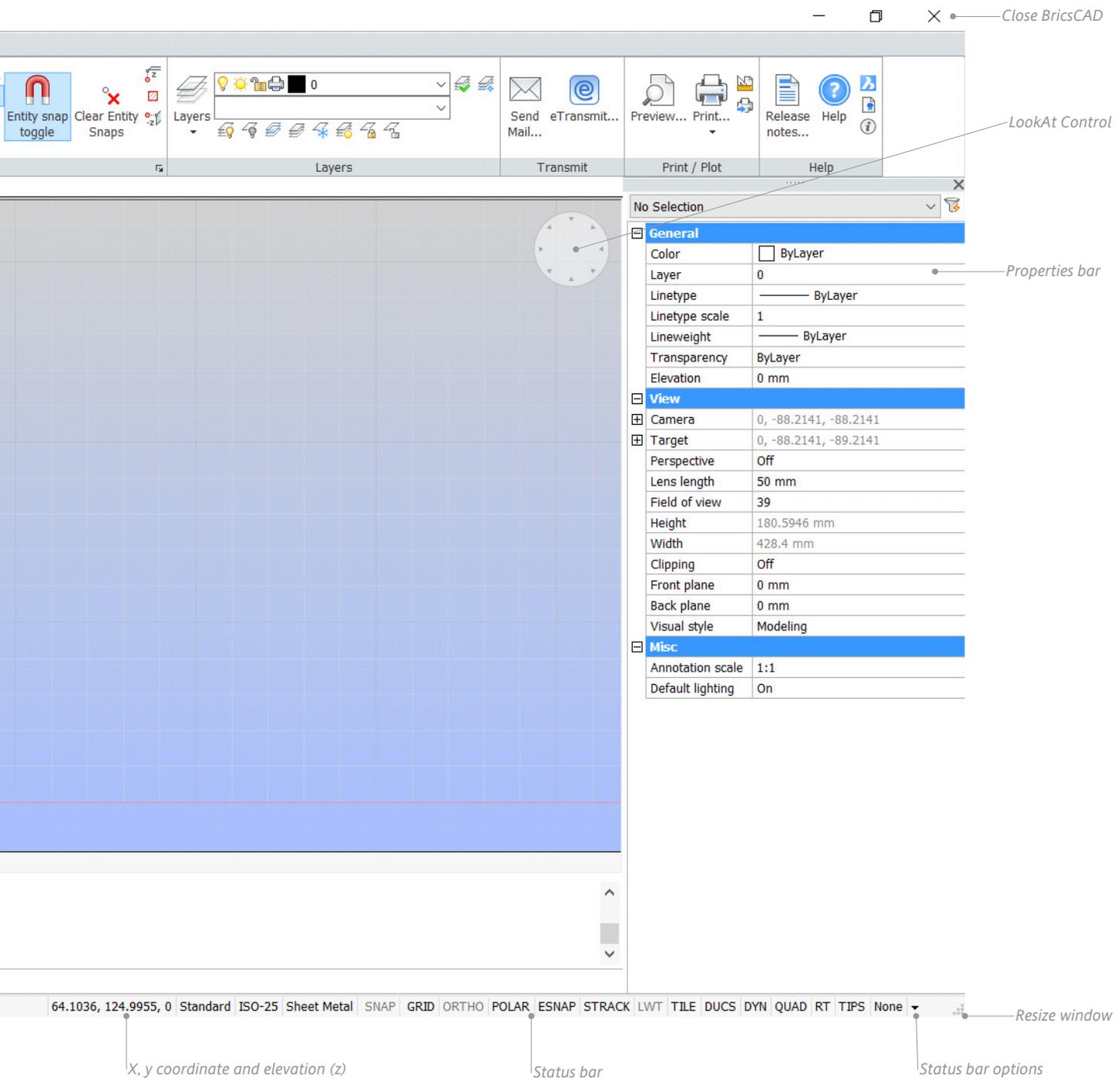
To keep the icon in the dock, follow these steps

1. Right-click the BricsCAD icon.
2. From the shortcut menu, choose **Options**, and then choose **Keep in Doc**.

BRICSCAD V17 USER INTERFACE

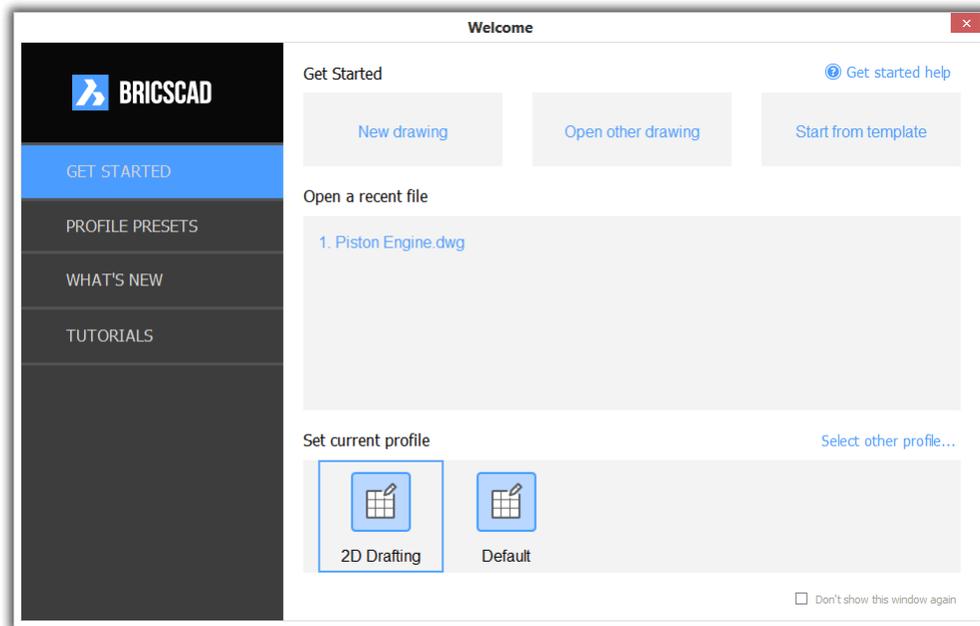
Illustrated is BricsCAD running on Windows 10.





Getting Started

(NEW IN V17) The first thing BricsCAD displays is the Welcome dialog box. (It replaces the Getting Started dialog box from earlier releases of BricsCAD.)



There are many options in this dialog box:

- Get Started** — starts with a new, recent, or other drawings, or else selects a template drawing
- Profile Presets** — shows the available workspaces and sets the units to metric or Imperial
- What's New** — reproduces “Release Notes” from <https://www.bricsys.com/common/releasenotes.jsp>
- Tutorials** — accesses the video tutorials hosted by Bricsys TV at <https://www.bricsys.com/tv>



Left to right: Profile Presets, What's New, and Tutorials

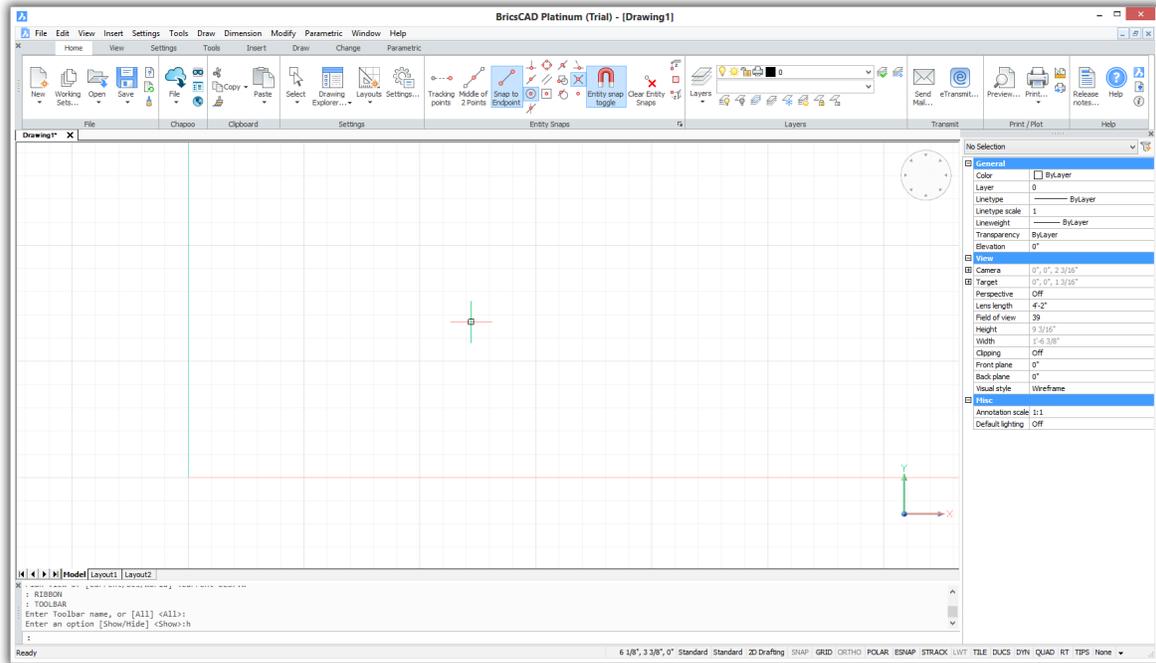
For now click **New Drawing** to enter BricsCAD.

THE BRICSCAD WINDOW

Take a look at the details of the BricsCAD window by checking out the figure spread across the earlier two pages.

BricsCAD’s central area consists of a large graphical drawing region. Here you can see the red-green cursor with its pickbox, and the red-green UCS icon with its x,y axes. The drawing region is surrounded by several panels of information — toolbars or ribbon, status bar, and so on.

Along the very top of the BricsCAD window, you see the title bar. Below it is the menu bar, below them the fat ribbon, and then the thin row of ribbon tabs. (Depending on how BricsCAD is configured, you might see toolbars.)



BricsCAD showing one possible configuration of its user interface

Along the bottom of BricsCAD are the layout tabs, the command prompt area, and then at the very bottom is the status bar.

A BASIC TOUR OF THE USER INTERFACE

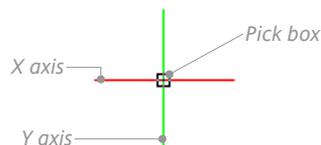
The user interfaces of CAD programs have many elements, and so it can be daunting to learn all of it at once. Here you look at just a few UI elements:

- Crosshair and arrow cursors
- Command bar
- UCS icon

Later, Chapter 2 provides a detailed tour of BricsCAD and its UI (short for “user interface”).

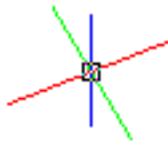
Crosshair and Arrow Cursors

The cursor gives you feedback from BricsCAD, Windows, and other software. When the cursor is in the BricsCAD drawing area, it looks like a *crosshair* that shows you where “you” are in the drawing, precisely.



Try moving the cursor now around the BricsCAD window: move your mouse.

Notice that the crosshair cursor has colors. These help you orient yourself, particularly in 3D (three dimensional) drafting. Here's what the cursor looks like in 3D, and what the colors mean:



- **Red line** represents the x axis
- **Green line** represents the y axis
- **Blue line** represents the z axis; it is hidden when you draw in 2D mode
- **Black square** is the pick box, for selecting entities

The black square  at the center of the crosshairs is called the “pickbox.” It shows you exactly where you are picking entities in the drawing. Entities outside the pickbox will not be picked. You use the pickbox during Chapter 5, “Adding Details to Drawings.”

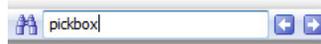
You can change the colors and the size of the cross hairs, as well as the size of the pick box, with the Settings command; see the tip coming up soon.

When you move the crosshair out of the drawing area, the cursor changes its shape to an arrow — one that you probably are familiar with from other software. You use the arrow cursor to make menu selections, pick buttons on the toolbar or ribbon, and so on. The cursor can change to other shapes. For example, when the cursor becomes a double-headed cursor, you can resize windows and palettes.



Left: Arrow cursor outside the drawing area, used to select UI elements; right: Double-headed cursor used to change size of UI elements

TIPS Many user interface options are changed in BricsCAD with the **Settings** command. It provides you with over 600 settings that let you control how BricsCAD works and looks. The best way to find a specific setting quickly is to enter its name in its search field, as shown below.



Changing the Pickbox Size. To change the size of the pickbox, enter “pickbox.” The default size that you see on the BricsCAD screen is 3 pixels wide, but you can change the size from 0 (gone) to 50 pixels (huge); I suggest you change it to 5 to make it just a bit bigger.

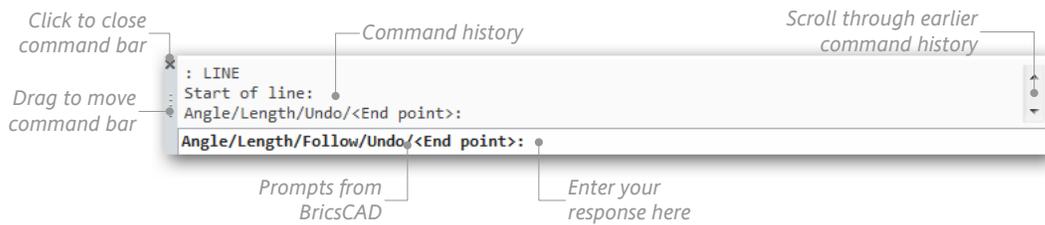
Changing the Crosshair Cursor Size. Should you find the crosshair cursor too small or too large, you change its size also through the Settings command: search for “crosshair.” The default value is 5, which means the length of the cross hairs is 5% of the screen’s size. When set to 100, the cross hairs stretch across the entire drawing area.

COMMAND BAR

Commands	CommandLine, CommandLineHide
Ribbon	...
Shortcuts	Ctrl+9, Shift+F2
Alias	...

The command bar is near the bottom of the BricsCAD window. This is one place where you can enter the names of commands and their options. If you are a touch typist like me, then you’ll find that you probably prefer specifying commands by typing them — instead of hunting through a menu or the ribbon. The command bar is also the place where BricsCAD prompts you for any additional information it needs to complete a command.

While you can turn off the Command bar with the **CommandLineHide** command, I don't recommend doing this; there is no good reason to do so! These are the important parts of the command bar:



Let's take a look at how the command bar works.

All About Command Prompts

When you see the ' :' (colon) symbol by itself in the Command bar, like this...

:

...it means that BricsCAD is ready for you to enter a command. The colon is called the "prompt." Should you wish to enter a command but there is text after the ' : ', press the **Esc** key once or twice to clear the command line.

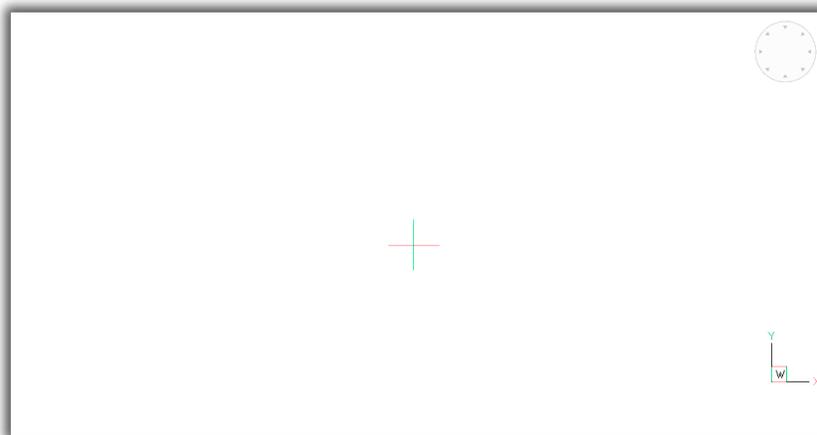
Try drawing a few lines with the **Line** command now:

1. Enter the Line command, as follows:
: line (Press **Enter**)

This means that you should type the word **line**, and then press the **Enter** key. Pressing Enter tells BricsCAD that you are finished typing the name of the command, and that it can now execute the command.

2. Notice that BricsCAD changes the prompt from ' : ' to ' Start of line: ' as follows:
Start of line:

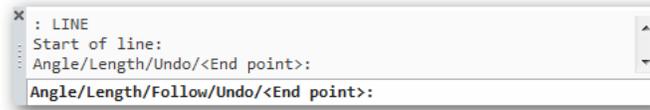
BricsCAD is asking you where you want it to start the line. You move the cursor (to the spot at which you wish the line to begin) by moving the mouse. As you move the mouse, notice that the crosshair cursor moves in concert.



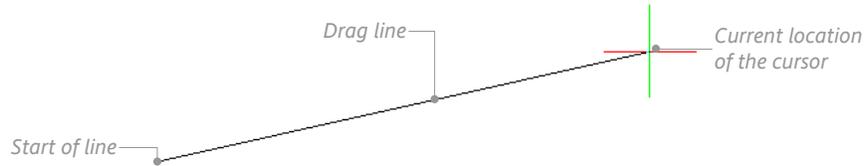
3. At any spot in the drawing area, pick a point on the screen by pressing the **first** button on your mouse. (The first button is the *left-most* one.) The left button is known universally in the CAD world as the "pick button."

Notice that the Command bar changes the prompt wording by adding more options. I'll tell you their meanings later.

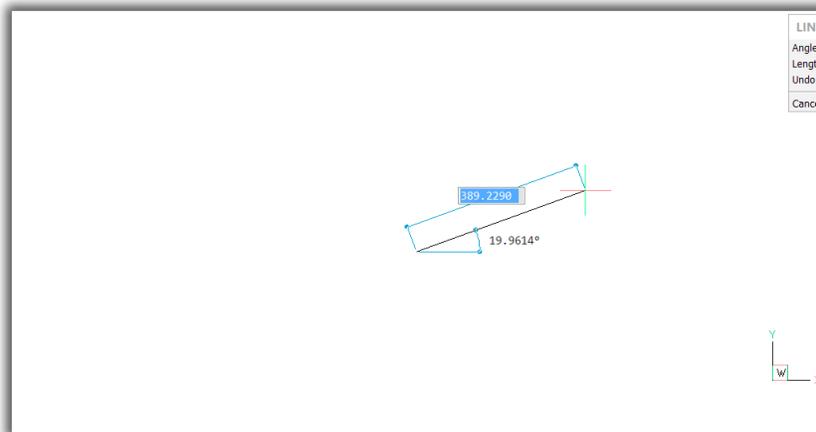
Angle/Length/Undo/<End point>: (Pick another point)



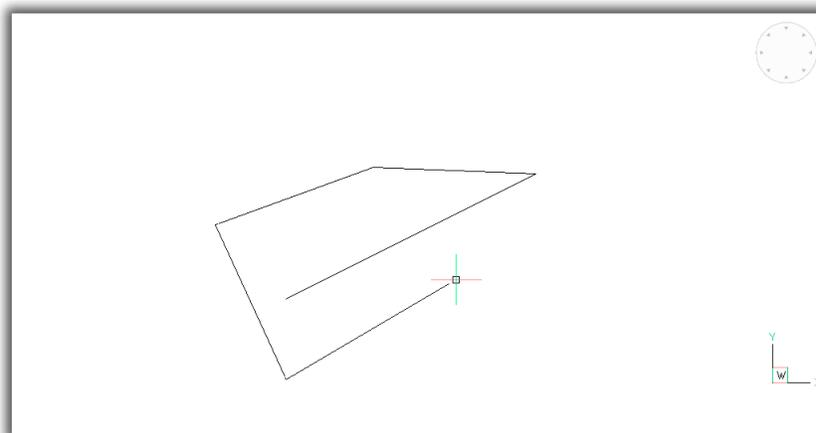
4. As you keep moving the mouse, notice the line that stretches like a “rubber band” from the point you picked. This rubber band is also known as the “drag line.” It shows where the line would be located if you were to click now.



Notice that BricsCAD may be providing you with information about the state of the line, specifically its length and its angle. If this appears on-screen, then it is called “direct distance entry.” You learn more about it in a later chapter.



5. Move the mouse some more, and then press the pick button again. There: you’ve drawn your first line with BricsCAD!
6. Continue drawing some more lines by repeating the same steps: (a) move the mouse and then (b) press the pick button. Draw as many lines as you like.



- To end the Line command, press the **Esc** key. Pressing Esc stops just about any command, although in some commands you may need to press the key two or three times.

Angle/Length/Follow/Close/Undo/<End point>: (Press **ESC**)

I want to point out that the prompt text you see — ‘Angle/Length/Follow/Close/Undo/<End point>’ — has subtle aspects that are not immediately obvious. For instance, you can specify an option by typing just the letters displayed in uppercase, such typing ‘a’ to start the **Angle** option.

The other aspect to notice is that the *default* option (or value) is always shown angle brackets, such as <End Point>. “Default” means that this is what BricsCAD will do when you just press **Enter**, without picking an option. This becomes a pretty fast way of working.

TIP Pressing **Enter** when you use the Line command has different effects, depending on the prompt that is currently active. (See the table below.) This is why it is important to always keep an eye on the prompts displayed by BricsCAD on the Command bar.

Prompt	Effect of Pressing Enter
Start of line:	Pressing Enter makes BricsCAD continue drawing from the last point, that was placed as a line or an arc. This is a great way to ensure that new lines are perfectly tangent to the ends of previous lines or arcs.
<End point>:	Pressing Enter terminates the Line command, just as Esc does.
:	Pressing Enter repeats the last command, which in this case is the Line command.

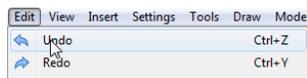
Undoing What You’ve Done: U

To erase the lines you drew, type **U** at the ‘:’ prompt to *undo* the lines, as follows:

: u

Alternatively, you could also access the U command in these ways:

- > Click the **Undo**  icon on the Standard toolbar
- > Select **Undo** from the **Edit** menu



- > Press **Ctrl+Z** — the shortcut keystroke for undo, and one that your fingers should memorize!

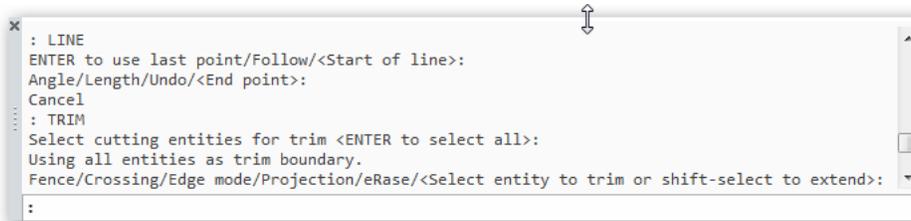
As you can see from the Undo example, BricsCAD provides several ways to perform actions. In the days to come, you will probably find yourself using a combination of keyboard typing, toolbar or ribbon icons, menu picks, keyboard shortcuts — whichever one you find the most convenient.

TIP At any time, right-click in the drawing area to display shortcut menus. (Press the mouse’s right button.) These menus show commands that are relevant to the current action or the state of the drawing. Because these menus are *context-sensitive*, their content changes depending on what’s going on at the time you right-clicked.

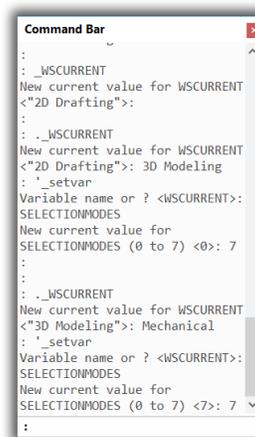
Seeing What You Did Before: Command History

The command bar typically displays three or four lines of *history*, which is the text of previously displayed prompts. When you need to see more lines of history, then you have a couple of choices:

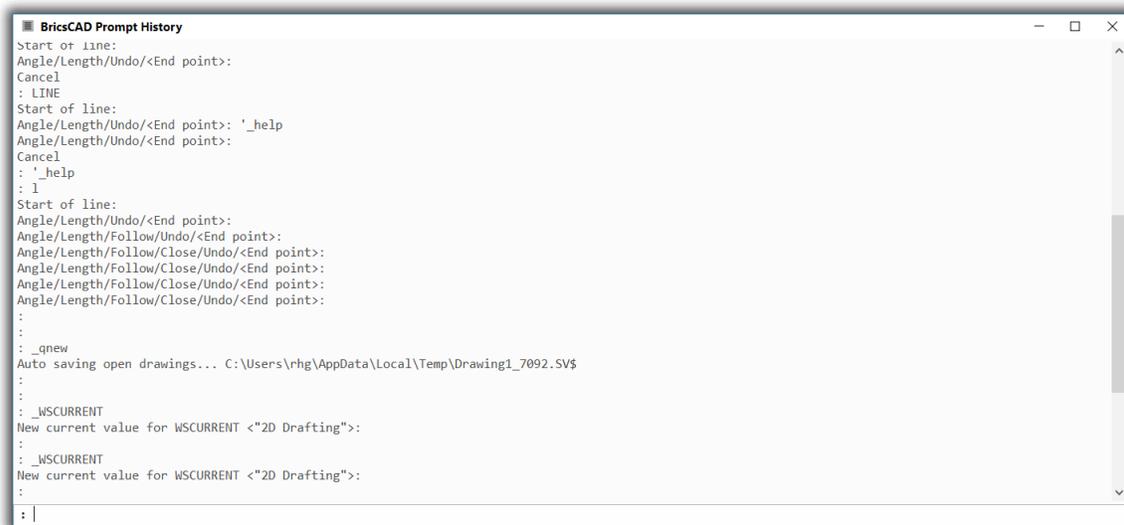
- ▶ Drag the command bar's top border to stretch it taller or shorter. For the exact point at which to do this, see the location of the double-ended arrow cursor in the figure below.



- ▶ Drag the bar away from its docked position, and then resize it, as shown below.



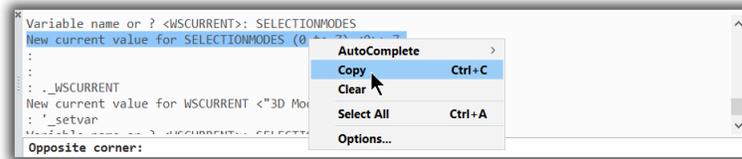
- ▶ Press **F2** to see the prompt window. BricsCAD displays a second window on the computer screen. You can reposition it anywhere, such as on a second screen.



The Prompt History window displays the most recent 400 lines of command text. You can scroll back to earlier prompts and other text by clicking on the vertical scroll bar along the right edge of the window. You can leave the window up, or else dismiss it by again pressing **F2**.

To keep a copy of the history, enter the **LogFileOn** command. Everything types in the command bar is recorded to a .log file with the same name as the drawing in this folder: *C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V17x64\en_US*. Use the **LogFilePath** variable to specify a more convenient folder, and the **LogFileOff** command to turn off command logging.

Alternatively, you can copy the text to the clipboard: select the text, then press **Ctrl+C**. Once copied, you can paste the text in any word processor or text editor. (This is process we use to get command prompts and path names into this book!) Right-click the text window for more options in the shortcut menu.



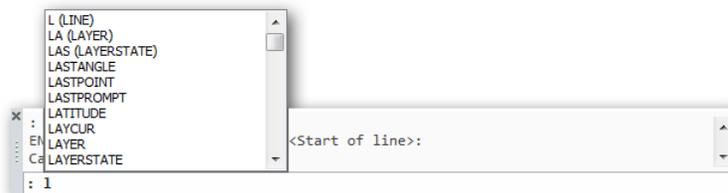
Typing Less: Aliases and AutoComplete

There are two ways to spend less time entering command names: by entering *aliases* and by taking advantage of the *auto-complete* function. I use both.

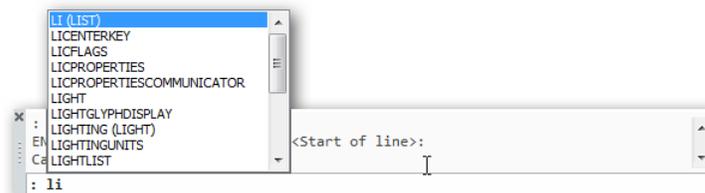
- > "Aliases" are command abbreviations, such as 'c' for the Circle command
- > "Auto-complete" lists the names of all commands that begin with the same letters as you are typing

Aliases are described later in this book and are fully listed in Appendix A. Here is how auto-complete works with the Line command:

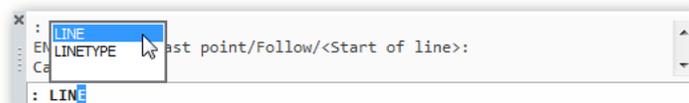
1. At the command prompt, type **L**. Notice that a box pops with listing the names of all commands and system variables that start with 'l'. To see them all, scroll down the list, all the way to where it ends with "lwunits."



2. Now type **l**, the second letter of the Line command. Notice that the suggestion list shortens to just the names that begin with 'li.'



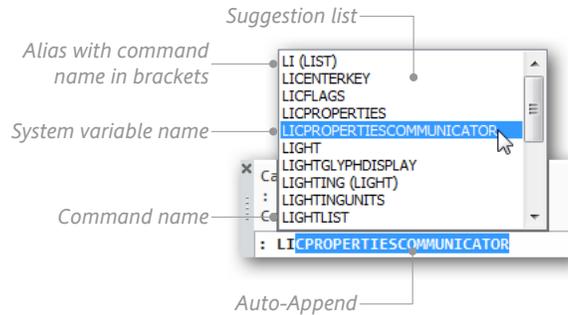
- a. You can keep typing letter of the command name to further reduce the suggestion list.



Notice that the last letter ('E') is highlighted in blue. This means that if you now press **Enter**, the entire command shown ('LINE') will be executed.

- b. Or you can use the cursor to select a name from the list. It's your choice.

Here are the user interface elements of the auto-complete function:



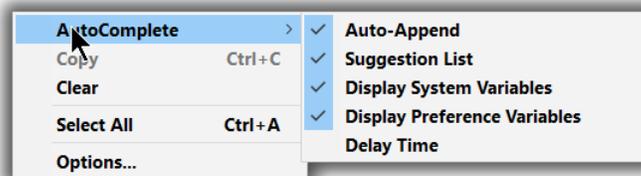
Suggestion list does not distinguish between the names of commands and system variables, but you can turn off the display of system variables, as described below.

Aliases are identified easily by the full command name being in brackets, such as LI (LIST).

Systems variables hold settings that determine how BricsCAD looks and acts; often, they are the same as AutoCAD's. *Preference variables* are similar, but are unique to BricsCAD. See Appendix B for the list of all of them.

Auto-Append highlights in blue the letters of the command name that BricsCAD is automatically completing for you.

To change the auto-complete settings, right-click the command bar and then choose **AutoComplete** from the shortcut menu.



Here is what the options in the shortcut menu mean:

Auto-Append — turns on or off (toggles) the command-completion characters, which are highlighted in blue

Suggestion List — toggles the display of the auto-complete list

Display System Variables — toggles the display of system variable names in the list

Display Preference Variables — toggles the display of preference variable names in the list

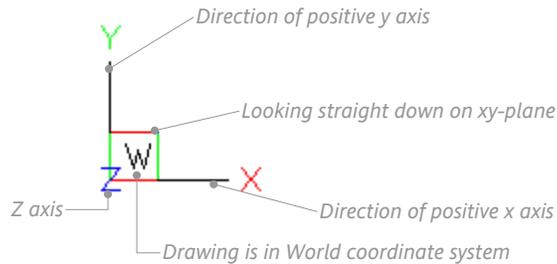
Delay Time — determines how long BricsCAD waits before displaying the auto-complete list

To change other things like the font or color used by the command line, right-click and then choose **Options** from the shortcut menu.

UCS ICON

Command	UcsIcon
Menu	...
Alias	...

The *UCS* icon is usually found somewhere in the drawing area, and often in the lower-right corner. *UCS* is short for “user-defined coordinate system.” Its job is to help you understand the drawing’s orientation in 3D. In 2D mode, it looks like this, packing a bunch of information:



X,Y Axes. The red X and green Y lines point in the direction of the positive x and y axes. At the intersection of the two lines is where the Z line points up, invisible now in 2D mode.

Drawing Origin. This UCS icon is usually (but not always) placed at the origin of the drawing. The origin is the intersection of the x and y axes, precisely where $x = 0$ and $y = 0$ are located.

World Coordinate System. When there is a ‘W’ at the origin, it means you are “looking straight down” the z axis, and straight onto the x,y-plane. The *W* means “world coordinate system,” which is the normal Cartesian system you may have learned about in school.

User-defined Coordinate System. When the *W* is missing, it means the drawing is in a “user-defined coordinate system” — *UCS*, for short, and one that you or someone else created. User-defined coordinate systems are especially handy when drawing on the slopes of roofs or the undersides of boxes.

I find the UCS icon gets in the way of 2D drafting, and so I recommend that you turn it off, as follows:

1. Type the **UcsIcon** command at the ‘:’ prompt in the command bar, as follows:
: **ucsetIcon** (Press **Enter**)

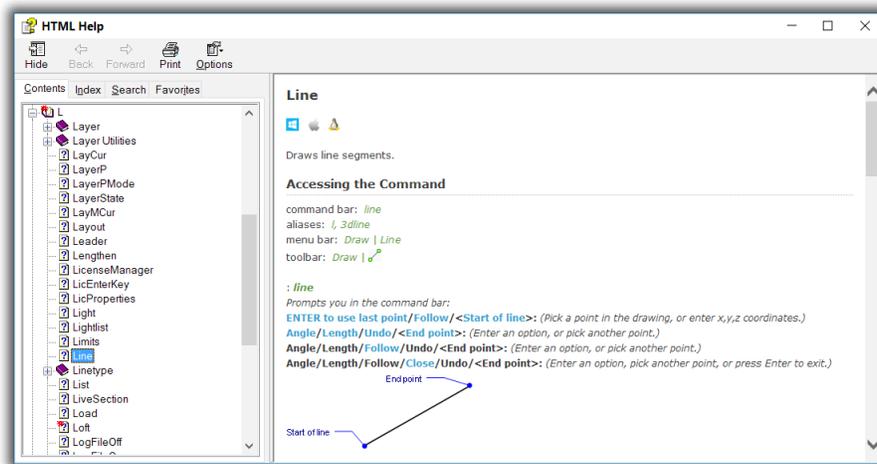
Press **Enter** to execute the command.
2. Notice the next prompt displayed by BricsCAD:
ON/OFF/All/Origin/<ON>: **off** (Press **Enter**)
3. Type ‘Off,’ and then press **Enter**. Notice that the UCS icon disappears from the drawing area.

You’ll get to try out the UCS icon later during the 3D modeling portion of this book.

ONLINE HELP

Command	Help
Ribbon	Home Help Help
Menu Bar	Help Help
Shortcut	F1
Alias	?

To peruse help during a command, press **F1** and BricsCAD displays a separate Help window, such as the one illustrated below for the Line command.



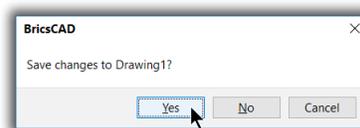
Icons indicate whether the command operates in each of the supported operating systems, Windows, MacOS, or Linux. Most commands do. Other icons indicate which edition the command works with, such as Pro, Platinum, or Sheet Metal. For the complete list of commands, see Appendix C.

In addition, there is an online version of help information that you access through a Web browser. Enter the following URL: https://www.bricsys.com/bricscad/help/en_US/V17/UsrGui/index.html.

Exiting BricsCAD

Command	Quit
Menu Bar	File Exit
Shortcuts	Ctrl+Q, Alt+F4

To exit BricsCAD, use the **Quit** command. When BricsCAD asks if you want to save the drawing, click **No**.



Alternatively, press **Ctrl+Q**, or else select **Exit** from the **File** menu.

What's New in BricsCAD V17

This list of new and changed BricsCAD functions was compiled from version 17.1.07. Bricsys continually updates this software, and so for information on functions added since this book was published, please see <http://www.bricsys.com/common/releasenotes.jsp>.

Changes are highlighted throughout this book, but be aware that information on these pages is not comprehensive. Command and variable names new since the last edition of this book are shown in boldface **blue**, updated ones are in boldface **black**. Commands and variables are listed in alphabetical order, sorted into the following sections:

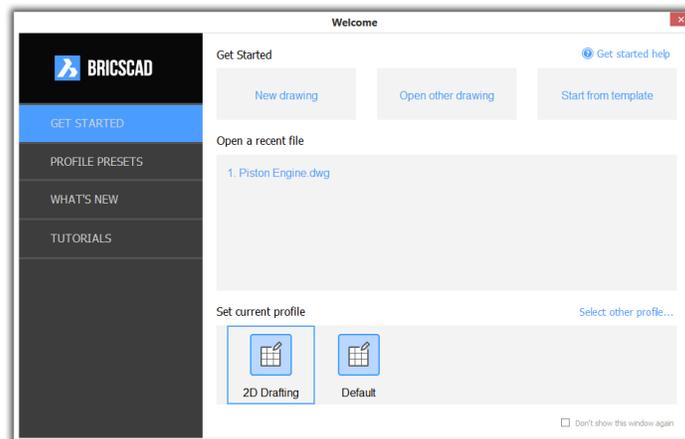
- > User Interface
- > 2D Drawing and Editing
- > Text and Dimensions
- > 3D Modeling
- > Generated Views
- > Rendering
- > BIM Module
- > Sheet Metal Module
- > Communicator Module
- > Mapping
- > Files
- > APIs

BricsCAD V17 installs and runs independently from previous BricsCAD versions.

What's New in the User Interface

BricsCAD V17 supports ultra-high resolution monitors with an extra-large set of icons and re-worked dialog boxes.

When BricsCAD V17 starts, it displays a redesigned Getting Started dialog box that makes available the following functions:

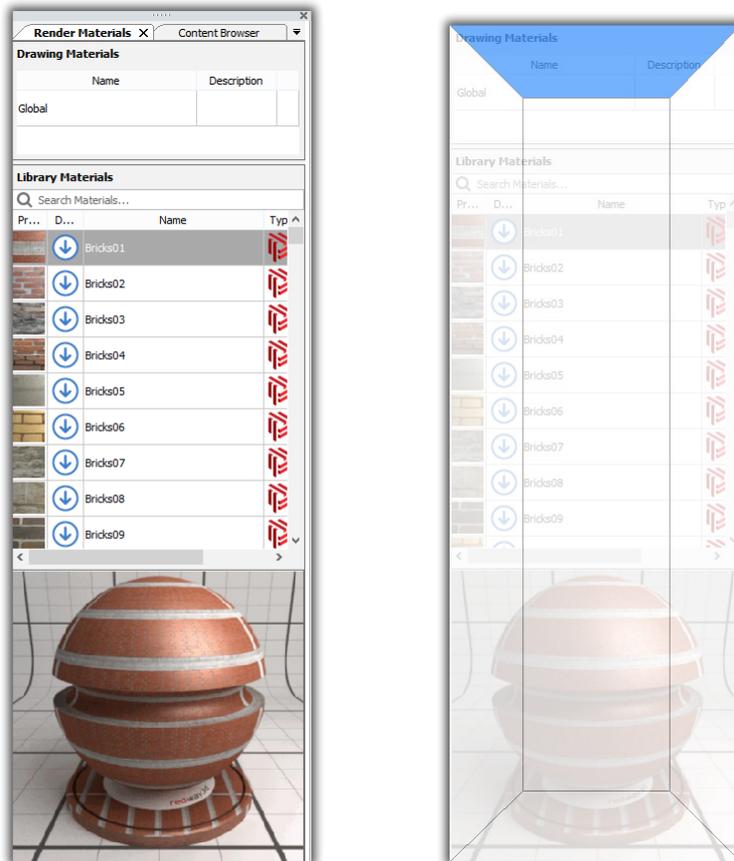


The Welcome window replaces the Getting Started screen

- > New and existing drawings
- > Workspaces
- > User profiles
- > Release notes
- > Online tutorials

TIP The **OnSwitch** list (commands executed when switching between workspaces) is emptied in V17 so that changes in workspaces are unaffected.

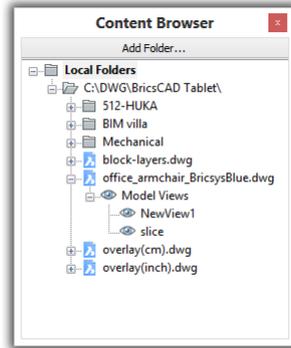
BricsCAD adds more *panels* (new name for ‘panes’), and multiple panels can be docked in an overlapping manner. Each docked panel has its own tab: Command Bar, Compositions (for BIM), Content Browser, Layers, Mechanical Browser, Properties Bar, Render Materials, Ribbon, Sheet Sets, Structure, and Tool Palettes.



Left: Tabbed panels stacked one over the other; right: moving a panel onto others

- > To dock a panel, drag it by its title bar over another panel, choosing one of the five offered locations (see figure above, at right)
- > To dock the panel beside, above or under an existing panel, drop it on the corresponding drop location

ContentBrowserOpen command displays the new Content Browser panel with a tree view of drawings in user-selected folders, along with model space views, which can be dragged into the current drawing.



Content Browser panel displaying the content of user-chosen folders

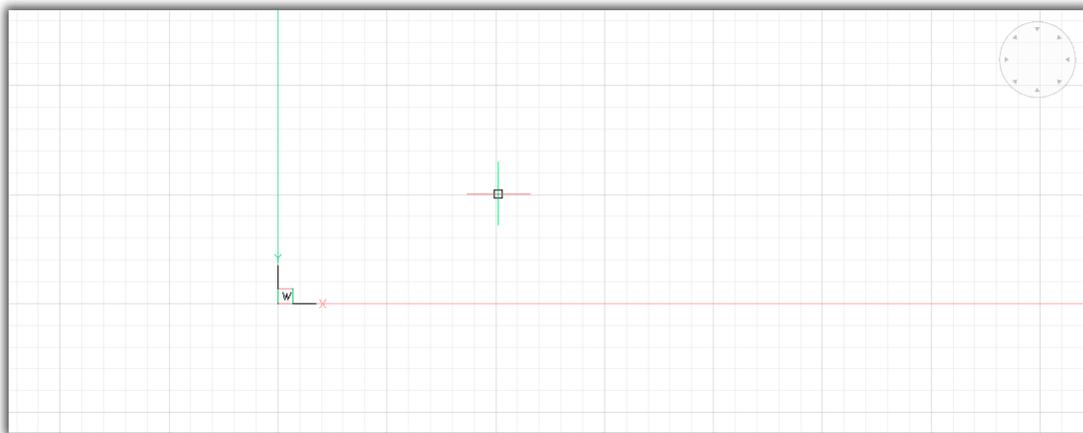
TIP Dragging model views from the Content Browser activates the new **Placeview** command automatically. Bricsys plans to add more drawing content, such as blocks and dimension styles.

ContentBrowserClose command closes the Content Browser pane.

Dynamic UCS behavior on curved surfaces is improved in V17.

Explorer command adds an option to hide and show xref symbols, which applies to all symbol tables. It gains Copy/Paste/Cut for MLeader Styles, MLine Styles, and Layer Filters.

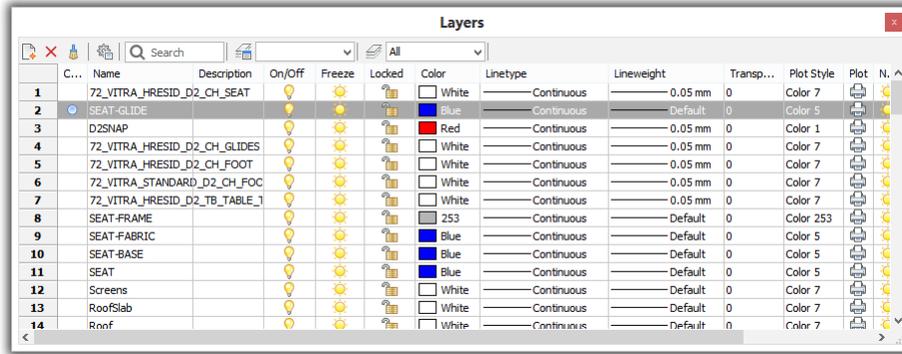
Grid command now draws grid lines partially transparent, and adaptive grid density is reduced to make the grid display less intrusive.



Pale grid lines

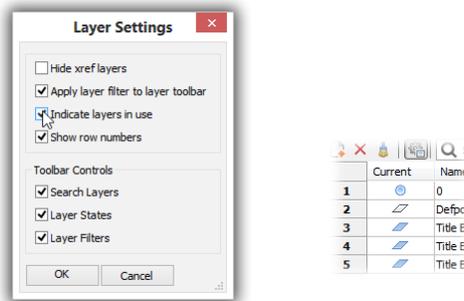
Layer command now displays which layers are in use in the Current column. (You turn it on in the LayerPanelOpen panel.) This command now displays a warning dialog box when opening drawings with more than 250 layer filters, and then offers to remove the filters, as they can cause performance penalties. In any case, the performance of opening drawings with many layer filters is improved.

LayersPanelOpen command displays layer names and properties in a new dockable panel.



New Layers panel

The **Settings** button toggles the display of the panel's UI elements, including the new **Indicate Layers in Use** option.



Left: The Layer Settings dialog box; right: new layer-in-use icons

The white layer icon indicates layer is “unused,” and so has no entities on it; the blue layer icon indicates layer is being used. (The blue dot indicates the current layer, as in earlier releases.)

LayersPanelClose command closes the dockable Layers panel.

Localization is improved localization for Hebrew menu files.

-Pan command returns the old manner in which pan operations were carried out; **-P** is changed to be its alias.

: -PAN

Set pan base point or [Left/Right/Up/Down/PaGe Left/PaGe Right/PaGe Up/PaGe Down]:
Pan displacement point:

PromptOptionFormat variable determines how command options are displayed on the command line and in the prompt menu; option 4 is meant for international versions of the software:

PromptOptionFormat	Meaning
0 (default)	Show description only Set end of arc or [draw Lines/Angle/CEnter/CLose/...
1	Show keywords only Set end of arc or [Line/Angle/CEnter/CLose/...
2	Show description, with keywords in brackets Set end of arc or [Draw lines(Line)/Angle/Center(CEnter)/Close(CLOSE)]...
3	Show description, with shortcuts in brackets Set end of arc or [Draw lines(L)/Angle/Center(CE)/Close(CL)]...
4	Show local keyword, with global keyword in brackets

TIP Use this variable to keep the command line as compact as possible, or as clear as possible at the expense of extra length.

PromptOptionTranslateKeywords variable toggles the use of international commands. When off, the underscore (_) prefix is not needed during command input; default = on.

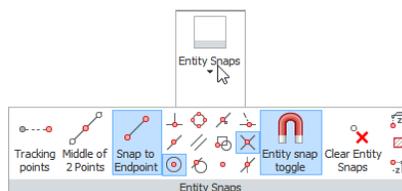
Properties command adds geometry properties for surfaces, such as Watertight, Loops, Holes, Lumps, and Faces. Read-only properties can now be copied to the Clipboard. Iterating through vertices of 2D and 3D polylines is improved.

QuadDisplay variable's value is now preserved when using **F12** to toggle its state.

QuadExpandTabDelay variable specifies the number of milliseconds between hovering over quad tabs and the tabs expanding; default = 50msec.

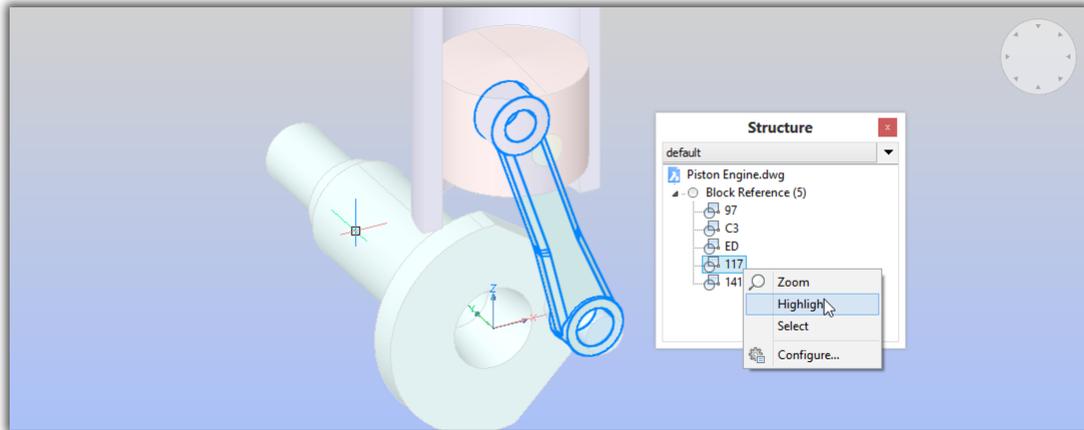
-Rename command now renames layers, blocks, and so on using wild cards.

Ribbon command shows extra items after a panel break in a slide-out panel.



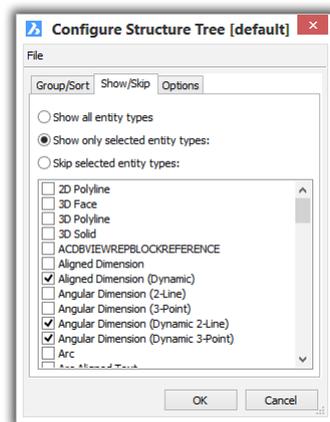
Condensed Entity Snaps panel showing all items in a slide-outpanel

Structure panel displays a structured tree view of the drawing's content. When entities are selected in the structure tree, they are highlighted in the drawing — and vice versa. The panel operates in model space only.



New Structure panel showing the structure of a 3D assembly and highlighting a selected part

The format of the panel can be customized through the Configure dialog box, and then saved and loaded through .cst configuration files.



Customizing the display of the Structure panel

StructureTreeConfig command loads .cst customize structure files from folders, such as `C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\Support`.

```
: STRUCTURETREECONFIG
New value for StructureTreeConfig, or . for none/⟨"bim.cst"⟩
```

ToolPalettes command now displays Group names in the context menu.

-ToolPanel command opens panels by name at the command bar:

```
: -TOOLPANEL

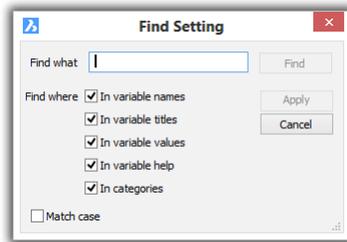
Enter Tool Panel name <* for all>:

Enter an option [Show/Hide/Toggle] <Show>:
```

TpNavigate command displays the tool palette or palette group specified by the user; meant for use at the command line.

```
: TPNavigate  
Specify tool palette to display or [palette Group]:
```

Settings command extends search options to string values in control labels.



Updated Find Setting dialog box

VisualStyle command switches between rendered visual styles quicker.

What's New in 2D Drawing and Editing

Note: BricsCAD does not have a block editor; the presence of these variables indicates it may be added to a future release.

BlockEditLock variable prevents the Block Editor from opening when a block is double-clicked; default = 0.

BlockEditor variable reports whether the Block Editor is open (read-only); default = 0.

BvMode variable toggles the display of hidden entities in the Block Editor; default = 0.

ResetBlock command resets dynamic blocks to their default values.

Center command and **CENter** entity snap now snaps to circular and elliptical viewports in paperspace.

Fillet command now applies a radius of 0 when the **Shift** is held down while selecting the second entity.

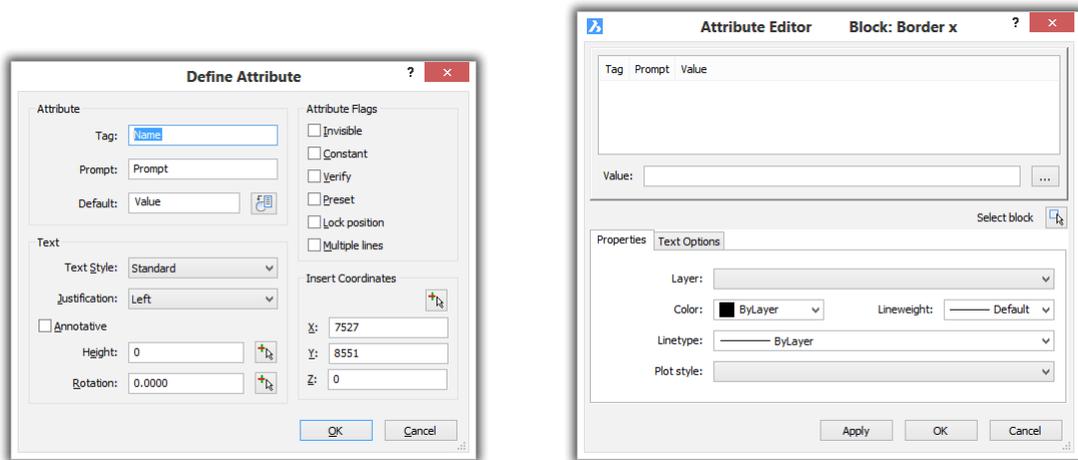
GCE (Geometric Center Esnap) now snaps to the center of rectangular and polygonal paperspace viewports, both clipped and unclipped.

SplinEdit command edits splines:

```
: SPLINEDIT  
Select spline:  
Edit spline [Close/Join/Fit data/Edit vertex/convert to Polyline/Reverse/Undo/eXit] <eXit>:
```

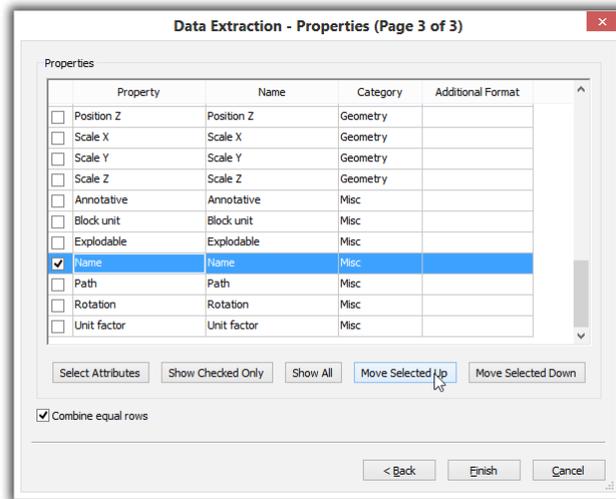
What's New in Text

AttDef, **EAttEdit**, and **BAttMan** commands boast improved dialog boxes.



Redesigned attribute dialog boxes

DataExtraction command now moves property rows using the new **Move Selected Up/Down** buttons or with the right mouse button. It now supports properties specific to BIM and sheet metal objects.



Data Extraction dialog box's new Move Selected buttons

Properties command adds the **Misc > Multiple Lines** option to convert single-line attribute definitions to multi-line ones.

Spell command offers improvements and bug fixes.

-Style and **Explorer** commands now show local font name if available, such as Chinese.

Table command now selects a delimiter for .csv files from a combo box: semicolon, tab, comma, and space.

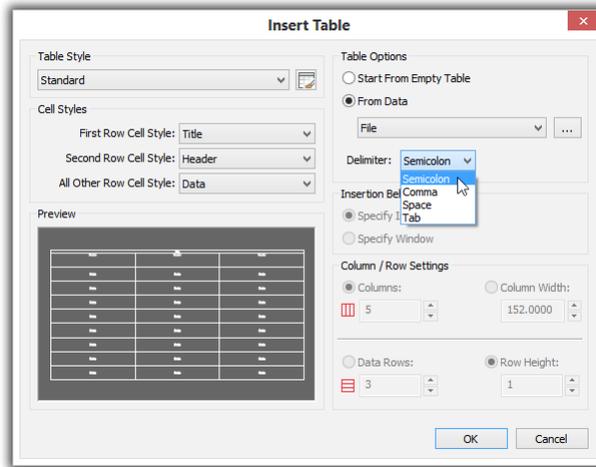


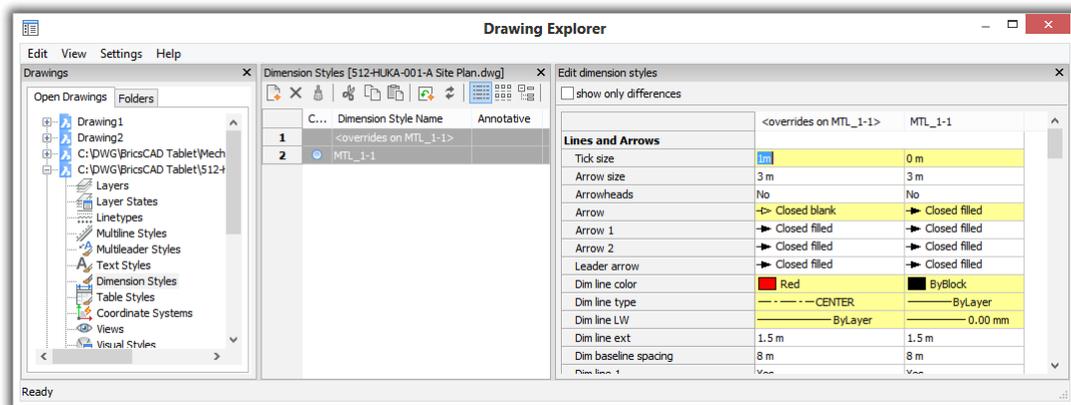
Table dialog box's new Delimiter option

What's New in Dimensions

aiDimPrec command sets the precision (number of decimal places or accuracy of fractions) of dimension text, and then stores the result in the **dimDec** variable.

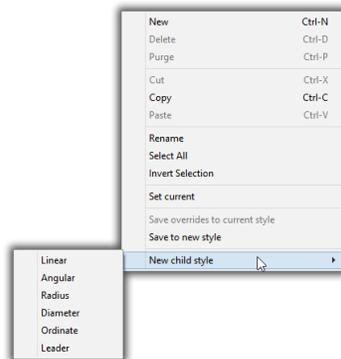
aiDimFlipArrow command mirrors selected arrows of dimension lines.

dimStyle command now offers an **Edit Dimension Styles** panel in the Dimension Styles explorer to see the differences between selected dimension styles (highlighted in yellow); values can be edited directly in the comparison table.



Differences in dimension styles being highlighted in yellow

Dimension style families start with a parent style (ie, a regular dimension style), and the define one or more child styles that are variations of linear, angular, diameter, radius, ordinate, or leader styles.



Creating new child styles

TIP To create a child style, right-click the parent style in the Dimension Styles explorer and then select the **New child style** option in the popup-menu.

Sub-units factor sets the number of sub units to a unit, and is used when the distance is less than one unit. For example, enter 100 if the suffix is m and the sub-unit suffix is to display in cm. This turns 0.96m into 96cm. Dimension styles display of dimension distances less than one unit in sub-units when the dimZIN variable is turned on (normally suppresses leading zeroes).

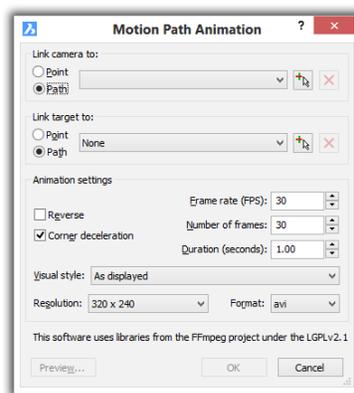
dimTxtDirection variable is added to the Properties panel and the Drawing Explorer.

dimTEdit command now immediately accepts preselected entities, when there is only one entity in the preselection. If more than one, or none, in the preselection, the command asks to select an entity.

What's New in 3D Modeling

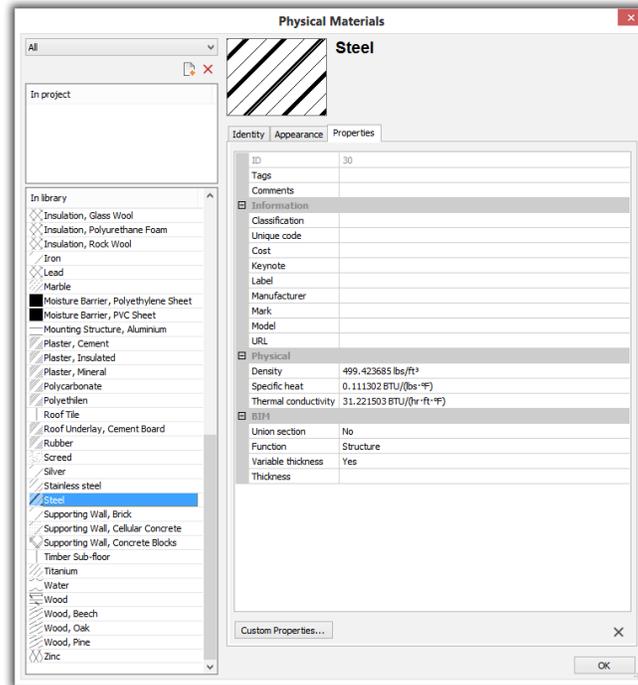
(*bm = bricsCAD modeling; dm = direct modeling*)

AniPath command records a series of images from a camera moving along a path in 3D models, and then saves it to .avi, .mpg, or .wmv files.



Dialog box for controlling the creation and output of movie files

Component materials define materials with physical properties, so that the mass and other geometric properties are accurately calculated. Materials are accessed from built-in or user-defined libraries of the Physical Materials dialog box. The default material is <Inheret>.



Dialog box for defining real-world properties of materials

TIP There is no command to access this dialog box. Instead, open the BIM Composition panel (right-click a toolbar or the ribbon, and then choose **BIM Composition**). In its toolbar, click the  **Physical Materials** button.

bmAutoUpdate variable controls whether locally stored copies of external components are reloaded automatically on opening the assembly document.

bmBom command's bill of materials table can now include the **Material** column.

bmBrowser command's Mechanical Browser pane now shows constraint arguments in the tree with their properties. Arguments can be removed from rigid sets or added to them from the current selection set.

bmExternalize command now preserves features and constraints attached to components in a larger number of cases, and now resolves file name conflicts when using the 'automatic mode' option.

bmInsert command now supports the insertion of local components, sheet metal form features, and arranges inserted items in linear arrays. The new Edit option modifies component parameters during insertion.

TIP When a component definition file contains 3D solids on BC_UNITE or BC_SUBTRACT layers, then the solids are added or subtracted from target 3D solid upon component insertion. The Insert as property of the component definition specifies whether the component is inserted as local or external.

bmLocalize command now preserves features and constraints attached to components in a larger number of cases.

bmMassProp command now takes into account the density of materials assigned to components. The group “Density” is removed from component properties and is not taken into account by this command. To define density, materials are to be used instead.

bmReplace command’s new **Similar inserts** option control if all similar inserts are replaced or only a selected one. It also preserves features and constraints attached to components in a larger number of cases.

bmUpdateMode variable determines when external assembly components are reloaded:

bmUpdateMode	Meaning
0 (default)	Update only modified components
1	Update all components

dmAngle3d command applies a constraint that controls the top angle of a cone; specify the angle between the axis and the cone face (= cone half angle).

```
: DMANGLE3D
Select first entity or specify [cone Angle constraint]:
Select second entity:
Specify angle value or set [set Axis] <90.00>:
```

dmAudit command replaces the **dmRepair** command to validate 3D solid and 3D surface geometry. The former **dmRepair** command structure and the clarity of reported issues has been improved.

```
: DMAUDIT
Select entities to audit [Entire model] <Entire model>:
Entire model will be processed, number of entities: 2
Choose action [Check/Fix/Options] <Fix>:
Selected count: 2

----- Solid -----
Handle: 393
Name in Mechanical Browser: Body_1
Errors: None

----- Skipped: -----
1 Line

No errors were found.
```

Delete key deletes 3D solid sub-entities. **dmDelete** command is replaced by the Delete key, but still works in V17.

dmDistance3d command measures between the nearest points on boundaries, central points, or the axes of geometry on cylinders, circles, and spheres.

```
: DMDISTANCE3D
Select a first entity or specify [Measurement mode]:
Select a second entity or specify [Measurement mode]:
Specify distance value <12.51>:
```

dmExtrudeMode variable controls Boolean operations for **dmExtrude** command's **Auto** option:

dmExtrudeMode	Meaning
0	Unite with new 3D solid
1	Create new 3D solids that extrude from a face
2	Subtract from solids that intersect
3 (default)	Both

dmSelect command is enhanced with the following options:

```
: DMSELECT
Select [Selection/Attribute/Relation/Primitive/feaTure/Finish] <Finish>:
Select [Face/Edge/Loop/edge Network] <Face>:
```

- > New **Primitive > EdgeNetwork** option to find adjacent edges with similar convexity
- > New **Primitive > Loop** option is enhanced to select borders of selected sets of faces
- > Selects G1- or G2-connected faces

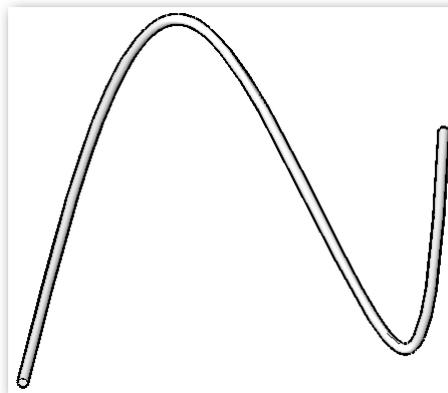
dmMove command now rotates adjacent planar faces, instead of translating edges.

dmRecognize command is no longer affected by design intent options during parametric components recalculations.

dmStitch command no longer preserves tolerance between command runs.

dmThicken command now creates tube-like 3D solids from wireframe entities, such as lines, splines, and polylines.

```
: DMTHICKEN
Select entities/subentities to thicken:
Entities in set: 1
Select entities/subentities to thicken:
Specify thickness value:
```

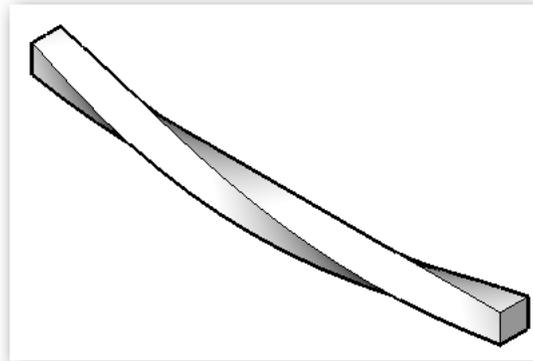


Tube made from a spline with the dmThicken command

dmTwist command modifies 3D solids, 3D surfaces, or 2D regions by twisting them about an axis by a specified angle.

```
: DMTWIST
Select object to twist:
Entities in set: 1
Select object to twist:

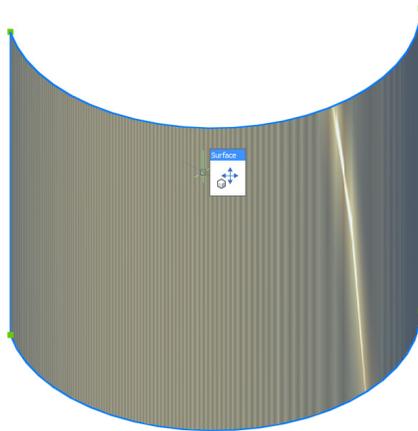
Enter start point of twisting axis:
Enter end point of twisting axis:
Pick start point of twisting:
Specify twist angle or set [Continuity]:
```



Square bar twisted by the dmTwist command

Erase command now accepts edges and faces of 3D solids and 3D surfaces, such as to erase a hole in a surface (after all edges are selected).

Extrude command now creates surfaces from open curves, instead of just solids from closed ones.



3D surface extruded from an arc

Interfere command now supports block references and nested selection of 3D solids inside blocks.

Intersect command now combines 3D solids with 3D surfaces or 2D regions.

Loft command now creates surfaces from open and closed curves.

MassPropAccuracy variable defines the number of decimal places, but is now stored as an integer with range of 2 - 12, instead of as a double-precision value (range 0.01-0.000000000001).

MassUnits variable specifies the units the Properties pane uses to report the mass of 3D solids; default = “z lbs stone mg g kg tonne”.

PolySolid command can now snap to itself during creation, and now closes itself when the last point equal to the starting point.

```
: POLYSOLID
Current settings: Height=80, Width=5, Justification=Center, Separate solids=On, Dynamic=On
Start point or [Height/Width/Justification/Entity/Separate solids/Dynamic] <Entity>:
Set next point or [draw Arcs/Distance/Follow]:
Set next point or [draw Arcs/Distance/Follow/Undo]:
Set next point or [draw Arcs/Close/Distance/Follow/Undo]:c1
Height of polysolid <80>:
```

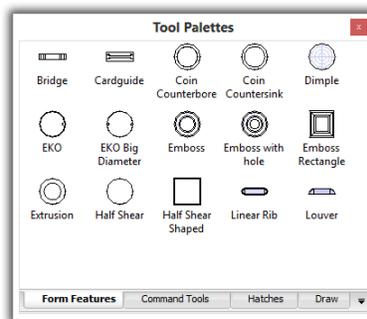
Perspective command now interprets perspective view parameters in a DWG-compatible manner, which may cause perspective views created with older BricsCAD versions to look different when opened in V17.

Properties command now controls the visibility of a particular component insert parameter by the new **Exposed** property.

Subtract command now subtracts 3D solids with 3D surfaces or 2D regions.

Sweep and **Revolve** commands now create surfaces from open curves, instead of just closed ones.

ToolPalettes command now supports components insertion.



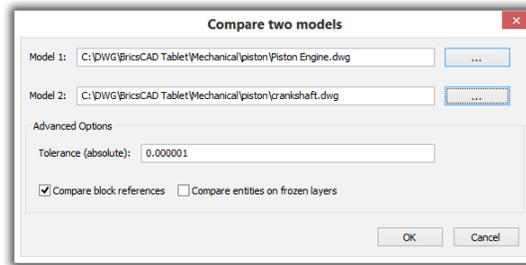
Tool Palettes panel showing available form features

TIP Use the right-click menu to add components from files listed in the Folders tab of the Drawing Explorer to the current tool palette.

XEdges command creates line, circle, or arc entities from the edges of 3D solids, 3D surfaces, and 2D regions.

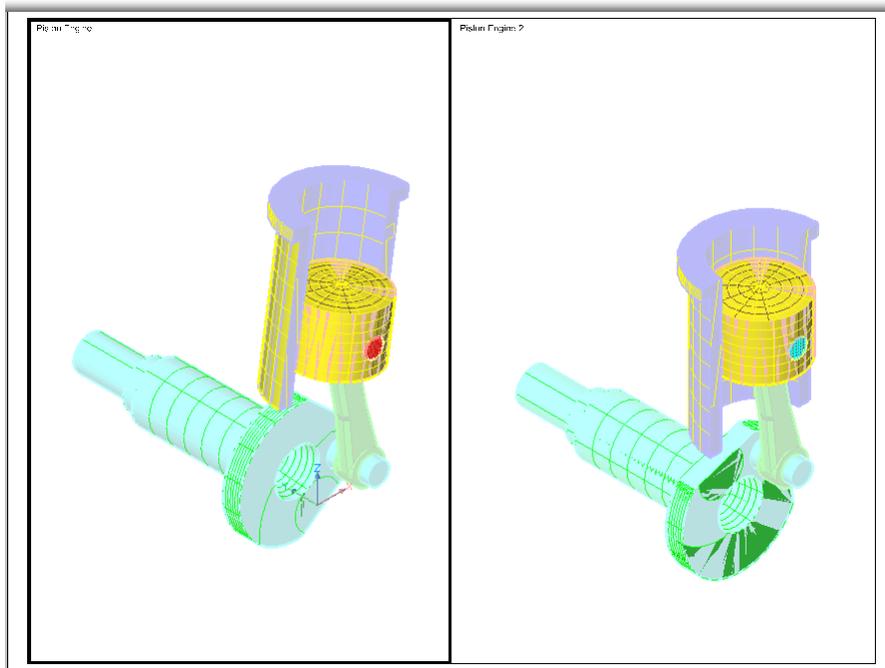
ZINtersection command (or **zint**) is a new 3D snap that snaps to the intersections of edges and tracking lines with faces.

3dCompare command loads two drawing files and then finds differences among 3D solids and surfaces using color coding. New panel in the ribbon.



Selecting two drawings for comparing changes in 3D objects

3dCompareMode variable determines if the results of the comparison results are shown in one or in two viewports; default = 3.



Two viewports showing differences in 3D models

3dCompareMode	Meaning
0	Show models without differences
1	Show differences in layout 'Comparison' left viewport
2	Show differences in layout 'Comparison' right viewport
4	Show differences in model space

3D constraints now take lines, circles, arcs, xlines, and rays as arguments of 3D constraints. If a constraint of the same type already exists, the new constraint is created with the "Disabled" flag.

WHAT'S NEW IN GENERATED VIEWS

AutoVpFitting variable controls the auto-fitting and resizing of viewport borders surrounding generated drawings. It moves derived views automatically when the parent view moves.

AutoVpFitting	Meaning
0	Keeps viewport border sizes fixed
1 (default)	Resizes viewport borders automatically

FlatShot and **SectionPlaneToBlock** commands lose the combo box listing preset orientations; use Dynamic UCS instead to control the orientation during insertion.

GenerateAssocViews variable determines whether the **ViewBase**, **ViewSection**, **ViewDetail**, and **bimSection** commands update the views and associative dimensions attached to 2D drawings automatically when the source 3D model changes. Upon changes to the 3D model, these views will be updated automatically or in course of **bimSectionUpdate** and **ViewUpdate** commands.

GenerateAssocViews	Meaning
0 (default)	bimSectionUpdate and Viewupdate manually update views
1	Automatically updates views and associative dimensions

PlaceView command places a model view from a source drawing into the paper space layout of the current drawing:

- ▶ Source drawing is inserted as an xref in the model space of the current drawing, using the same layer(s) as the source drawing
- ▶ Paper space viewport is added that matches the source view
- ▶ Only layers of the xref are visible in the viewport; view is not disturbed by other drawing content
- ▶ When the current drawing belongs to a sheetsheet, a matching sheetsheet view is created and a view label block is added

Properties command now shows additional properties when a generated view is selected.

ViewBase command's new **Select objects** option includes or excludes entities from the selection set of the base view.

```
: VIEWBASE  
Preset: "None", View scale: "Adapt to paper size"  
Select objects or [Entire model/presets] <Entire model>:
```

ViewDetail command has new options:

```
Select option [Scale/Hidden lines/Tangent lines/anChor/Annotation/Boundary/model Edge] <Cancel>:
```

- ▶ **Boundary** option chooses between rectangular or circular boundaries for detail views
Detail boundary [Rectangular/Circular]:
- ▶ **Model Edge** option determines how the leader line is drawn between the detail view and the detail boundary.
Model edge type [smooth with Border/smooth with Connection line]:
- ▶ View properties can be edited before the command is completed

ViewEdit command has new options:

- > **Anchor** option fixes view center in paper space
- > **Depth** option specifies the depth of sectioned views
- > **Select** option includes and exclude objects from base view
- > Hidden lines settings and scales are propagated from the parent view to section and detail views.

```
: VIEWEDIT  
Select option [Scale/Hidden lines/Tangent lines/anChor/Annotation/Boundary/model Edge] <Cancel>:
```

ViewSection command's new **Aligned** option chooses alternative projection types.

```
: VIEWSECTION  
Select drawing view:  
Specify start point of section line or [Type] <Type>: t  
Select type [Full/Half/Offset/Aligned] <Full>:
```

View properties can be edited before the command is completed.

```
Select option [Scale/Hidden lines/Tangent lines/anChor/Annotation/Depth/Projection] <Cancel>:
```

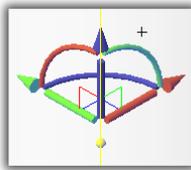
It now uses material-defined hatch patterns for mechanical components and assemblies.

WHAT'S NEW IN RENDERING

MaterialMap command adjusts how rendering textures are mapped on to basic shapes like planes, boxes, cylinders, and spheres.

```
: MATERIALMAP  
Select an option [Box/Planar/Spherical/Cylindrical/copY mapping to/Reset mapping] <Box>: c  
Select faces or entities:  
Entities in set: 1  
Select faces or entities:  
Edit the mapping or [reseT/switch mapping mode] <Accept current mapping>:
```

The manipulator gizmo controls the origin, rotation angle, and scale factor of the texture.



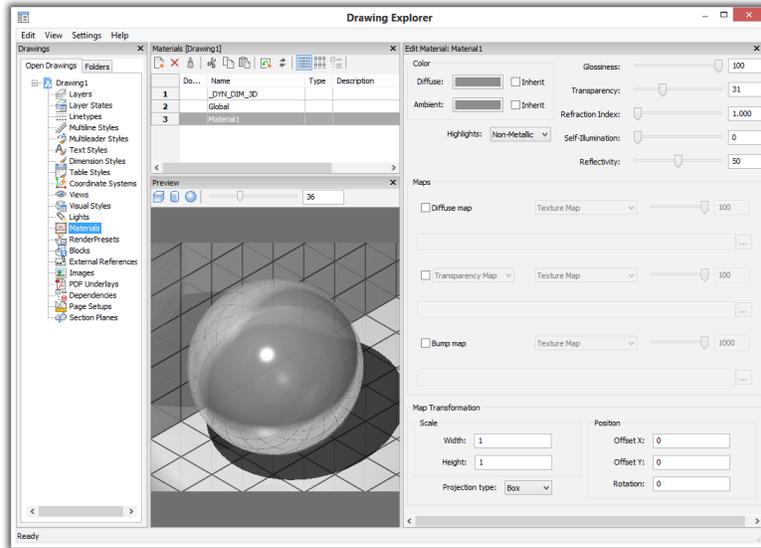
3D gizmo for controlling position of materials on surfaces

MatchPerspective command changes the perspective viewpoint of the current view in model space view to match a background image; this is done by selecting at least three point pairs. The command works only when the Perspective variable = on.

```
: MATCHPERSPECTIVE  
Enter Model Point:  
Enter Image Point or [Undo]:  
Enter Model Point or [Undo]:  
Enter Image Point or [Undo]:  
Enter Model Point or [Undo]:  
Enter Image Point or [Undo]:  
Enter Model Point or [Undo] <Match>:
```

Materials commands adds the following functions:

- ▶ New columns indicate the render material definition type — regular or RedWay — and the download status
- ▶ New option convert RedWay material definitions to a regular definitions
- ▶ New preview object size control
- ▶ Each material now has a projection type: planar, box, cylinder, or sphere



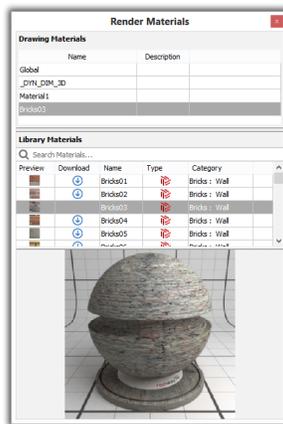
Renovated Materials section in the Drawing Explorer

- ▶ **Width** and **Height** scale values are interpreted differently, depending on the projection type:

Projection Type	Width	Height
Cylindrical	Number of cylinder rounds for full texture width	Number of drawing units correspond to full image height
Spherical	Number of sphere rounds for full texture image width	Number of sphere rounds covered by one full image height

TIP Double-click the material's download icon to start downloading.

MatBrowserOpen command (opens the Render Materials pane) now supports double-clicking a material to open it in the Materials Explorer for editing. Materials can be drag-and-dropped from the Library Materials list to the Drawing Materials list.



Updated Render Materials panel

What's New in the BIM Module

BIM is an optional extra-cost add-on as of V17. (bim = building information modeling)

BricsCAD BIM V17 is certified for IFC export at IFC2x3 Coordination View V2.0 level. BIM elements carry all of the properties defined by the IFC2x3 Coordination View CV2.0, and are accessible in the Properties panel.

bimAutoUpdateRoom variable automatically updates rooms when bounding walls are modified; new walls are not detected.

bimClassify now has the **Other** option to classify the following new BIM elements: Covering, CurtainWall, FlowTerminal, Footing, FurnishingElement, Member, Pile, Railing, Ramp, RampFlight, Roof, Site, Stair, and StairFlight. This command applies Window and Door classifications on window and door definition files.

TIP To reclassify a drawing, enter the **bimClassify** command, select the **Window** or **Door** option, and then press **Enter** to select nothing. The Properties panel edits the drawing's Window and Door properties when nothing in the drawing is selected.

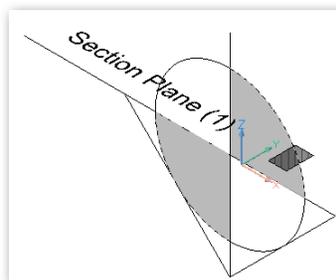
bimRoom command defines a room by clicking inside a room area or by selecting a 3D solid.

```
: BIMROOM
Pick a point or [select 3d Solid]: S
Select 3d solid or [pick a Point]:
Successfully created a room.
```

TIPS Rooms keep the relationship with their bounding walls. When clicking a point inside the room area, BricsCAD uses dynamic UCS to define the bottom plane of the room. BricsCAD finds the area enclosed by walls, and then places a room marker consisting of a block made of a hatch and attributes for the room's name, number, and area.

When a room is defined by the click method, then it will report the finishing materials of the wall sides inside the room.

bimSection command adds the **Detail** option, which creates a section with Volume state by default.



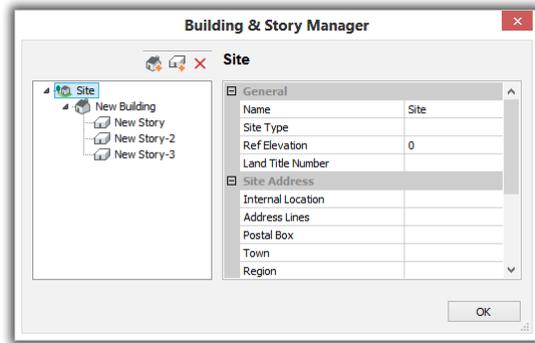
Room section label

TIPS To create the volume section, the command prompts for three points using dynamic UCS to define the base plane of the box and becomes section plane.

Starting **bimSection** from the Quad creates detail section boxes based in the same plane as the section over which the cursor is hovering.

bimSection now displays hatch patterns from compositions live on the 3D model when the Clip Display property is active.

bimSpatialLocations command opens the Buildings & Stories Manager dialog box to create and edit the site (one per drawing), buildings (one or more per site; default = 3), and stories (one or more per building).



Buildings & Stories Manager dialog box

bimUpdateRoom command executes the room-finding algorithm to redefine the room, such as if new walls were added.

DataExtraction command now exports the properties of all BIM elements, including windows, doors, and rooms.

PlaceView command drags 3D views from the Content Browser onto the paper space of a section result drawing.

Properties command is updated for BIM to list **Display Composition**. When on, the selected 3D solid shows its composition of ply faces, which can then be separately selected.

StructureTreeConfig variable loads a .cst “configuration structure” file that formats the new Structure panel. In BIM, it examines all aspects of the BIM model, and can be customized by the user. The default structure organizes the building spatially: first by Building, then by Story, BIM type, and composition. The new Structure panel groups sections by type: Section, Plan, Elevation, or Detail. See comments on the Structure panel earlier in this chapter.

TIP The Structure panel can be configured to group and sort by any property, including all IFC properties.

What's New in the Sheet Metal Module

The Sheet Metal module is an optional extra-cost add-on. (*sm* = sheet metal)

smBendLineExtentValue variable specifies the bend line properties; default = 0.25.

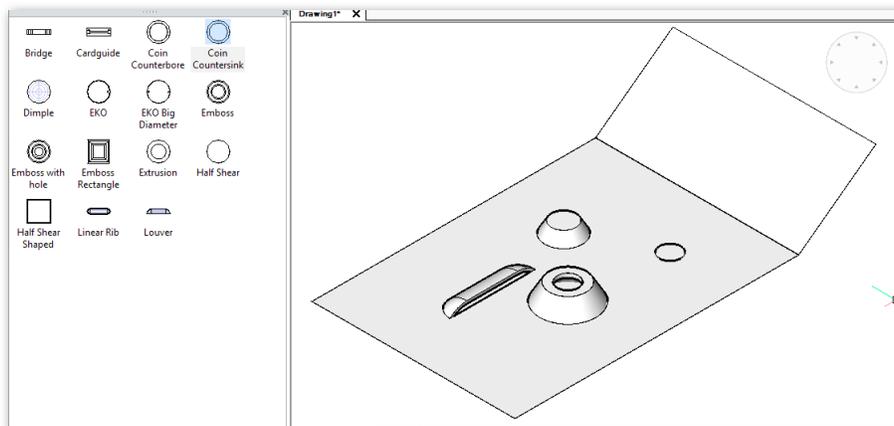
smConvert now recognizes cylindrical lofted bodies as lofted bends, form features in imported geometry, and more types of wrong bend features.

smExportOsm command now creates *.osm* files when bends are adjacent to lofted bends, and adds information about component materials to *.osm* files.

smFlangeBend command bends existing flanges along lines, obeying the k-factor for the given bend radius.

smFlangeEdge command now improves relief creation.

smForm command converts a selected set of faces to form features. A drawing file with a user-defined form feature can be saved and then used with the *bmInsert* command to insert the form feature.



Adding form features to a sheet metal part

TIP Form features are a new kind of sheet metal feature that mimics applying a forming tool to the sheet metal, such as bridges, louver, and embosses. They inserted from built-in or user-defined libraries; BricsCAD recognizes form features in imported geometry. Form features are listed in the Mechanical Browser pane with their parameters; they can be edited directly or parametrically through Properties panel. `C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\Support\DesignLibrary\SheetMetal\FormFeatures`

smFormFeatureUnfoldMode variable controls the appearance of form features in 2D and 3D unfolded model representations; this variable must be modified through the Settings dialog box.

smFormFeatureUnfoldMode	Meaning
0	Keep
1	Remove
2	Project
3	Contour
4 (default)	Symbol

smJunctionCreate adds the option to select “Entire model” and 3D solids.

smKFactor variable specifies the default K-factor; default =0.27324.

smLoft command’s new **Auto** option for fillet radius creates bodies with the smallest possible fillet radius (given the thicken type).

smReliefSwitch command adds options for bend reliefs: “Switch to Smooth”, “Switch to Round” and “Switch to Rip”. The new **auto** value for corner relief extensions now means “Keep the extension, which is set in the feature. Switching corner reliefs near flange splits (a mitter) to V-type are automatically converted to two smooth bend reliefs.

smReliefCreate command optionally forces the creation of bend reliefs. It creates correct relief geometry when the bend radius is not equal to the default bend radius set in the drawing.

smRepair command replaces the **smRethicken** command, which is removed. It now repairs wrong bends by converting them into regular bends, changes the “Enable lofted bend repair” prompt to “Merge lofted bends”, with improved support for adjacent lofted bends. It gains automatic repair of coincident faces cases for **WrongBend** features.

smReplace command replaces form features (including recognized ones) in sheet metal parts with form features from built-in or user library — even if their dimensions are different.

smSelect command selects hard edges, same and similar form features, and is added to the Select section of the Quad menu.

smRethicken command is removed; its function is replaced by the **smRepair** command.

_Sm_Thickness component parameter can be edited during and after insertion with the **bmInsert** command.

smUnfold command adds information about component materials to *.dxf* files, and displays a warning message about torn lofted bends.

What’s New in the Communicator Module

Communicator is an optional extra-cost add-on. BricsCAD V17 is not compatible with Communicator V16, and so Communicator must be upgraded.

ExportProductStructure variable determines whether the product (assembly) structures are exported.

ImportHiddenParts variable controls if hidden parts are imported

Communicator now exports the following data:

- › Product (assembly) structures to IGES/STEP

Communicator now imports the following data:

- › Materials with physical properties, if they are assigned to the parts of imported products
- › Root assembly component names are set to the imported assembly file name automatically.
- › Alternate search paths search for imported assembly parts from Creo, Inventor, NS, SolidEdge, and Solidworks file.
- › XCGM file format

What's New in Mapping

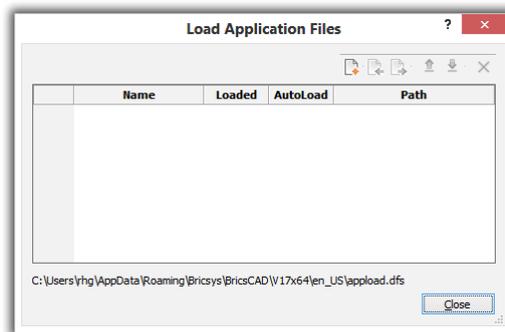
BricsCAD V17 now supports the following coordinate reference systems:

- › Czechia/Slovakia S-JTSK
- › Croatia EPSG 3765
- › Netherlands EPSG 28992

GeographicLocation command gains a fast filter-as-you-type control when searching coordinate reference systems.

What's New in Files

AppLoad command's dialog is redesigned to make auto-loading applications easier, adding support for .LSP, .NET, and .VBA (in addition to .BRX and .TX).



Redesigned AppLoad dialog box

CheckDwlPresence variable reports whether .dwl and/or .dwl2 locks files are associated with the drawing being opened.

-eTransmit command is the new command line version of the **eTransmit** command. Both commands now handle the additional files required for BIM and mechanical assembly modeling.

ExportPdf command takes over PDF exporting from the **Export** command. Text in exported PDF files is now searchable for all visual styles. (Text in clipped inserts is not yet exported as searchable text.)

ImageAttach command now allows multiple selections of images from a single folder to attach multiple images at once. This is especially useful for images with geo-information attached. The images are laid on top of each other.

-ImageAttach now support relative and absolute paths in the command line version.

PdfMergeControl variable determines how overlapping lines are printed.

PdfMergeControl	Meaning
0 (default)	Lines overwrite
1	Lines merge

XrefOverride variable controls the display of properties like color, linetype, lineweight, transparency, and plot style in referenced layers.

What's New in APIs

ACIS is upgraded to v2017 1.0 (R27).

BricsCAD V17 is compiled with Visual Studio 2013 (platform toolset = v120), and so to be compatible C++ extension .dlls need to be compiled with the same platform toolset.

VBA is upgraded v7.1. It add support for the following items:

- > Allows 64-bit operations and is compatible with earlier versions
- > Provides compilation constants VBA7 and Win64
- > Adds keywords LongLong, LongPtr, and PtrSafe

TIP VBA is no longer installed by default, but requires a separate installation from the VBA subfolder of the application installation folder, such as in `C:\Program Files\Bricsys\BricsCAD V17 en_US\VBA`.

The following BricsCAD-specific APIs are added to BRX:

- > An API for the Quad in BRX and .Net
- > An unmanaged C++ Ribbon API
- > An API for 3D constraints and parameters
- > Subentities are supported by the C++ OPM API part of BRX
- > Ribbon API was added for BRX and .NET

BRX 17 interface is updated to be source code-compatible with ARX 2015/2016 SDK, such as overrules. As there are some exceptions, BRX supports newer and legacy interfaces when possible.

TIP Sample applications installed with BricsCAD are updated to demonstrate these new APIs, such as under the `C:\Program Files\Bricsys\BricsCAD V17 en_US\API` folder.

License Requirements

Pro or Platinum license is required for rendering, 3D modeling, and Drawing Views functions.

Platinum license is required for creation of 3D Constraints, Mechanical Assembly design, and Deformable Modeling functions.

BIM design and Sheet Metal design functions require a separate license for each on top of a BricsCAD Platinum license.

This has been your introduction to seeing and using BricsCAD. Next, you get into greater depth using the user interface. But first... a summary of what's new in the V17 release of BricsCAD.

SUPPORTED GRAPHICS BOARDS

BricsCAD works with whatever graphics board is built into your computer. When it comes to non-wireframe renderings, however, BricsCAD employs RedSDK GPU-acceleration technology provided by Redway3D. (GPUs are the processing chips on graphics boards.) Download the latest drivers from <http://www.redway3d.com/supported-gpu-tables-for-redsdk/list-of-the-available-drivers/>.

WINDOWS

On Windows, RedSDK hardware acceleration supports graphics boards made by AMD, Intel, and nVidia. See http://www.redway3d.com/downloads/public/documentation/bk_ba_gpu_chipset_reference.html; for the comprehensive list.

MAC

Apple does not allow developers to have full access to the graphics hardware inside Mac computers, and so there are no third-party drivers available to speed up the display of CAD programs.

LINUX

On Linux, RedSDK support graphics chip sets for 3D graphics hardware acceleration from AMD and nVidia. Intel is not supported in Linux, nor are laptops with discrete graphics systems. You are advised to download the latest recommended drivers from NVIDIA and AMD.

Navigating the BricsCAD Interface

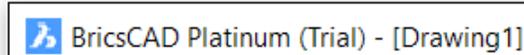
We continue here our tour of BricsCAD's user interface (UI), looking at parts of it in greater detail. You will work your way from the top of the screen to the bottom — from the title bar down to the status bar.

In subsequent lessons, you'll learn about some of the nuances to the UI, such as entering aliases and working with relative coordinates.

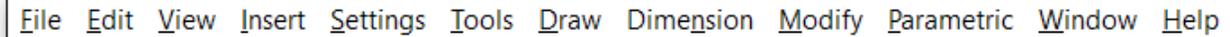
IN THIS CHAPTER

- Working with the upper half of BricsCAD: title bar, menu bar, toolbars, and ribbon
- Trying out some user interface elements of the drawing area: Quad, LookFrom widget, and UCS icon
- Finding out about the lower half of BricsCAD: layout tabs, scroll bar, command bar, and status bar
- Understanding panels (palettes)

Title bar: program name and name of the current drawing, with program window controls at the far right



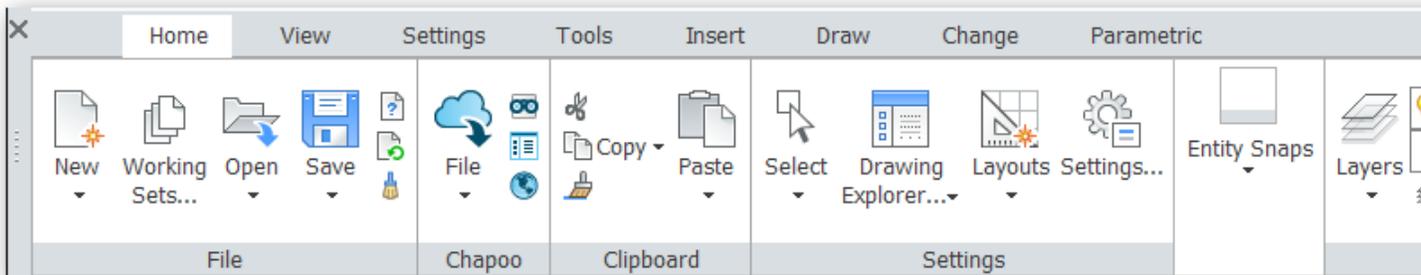
Menu bar: commands arranged in logical order



Toolbar: buttons in toolbars, with flyouts and droplists



Ribbon: tabs with panels with buttons, flyouts, and droplists to access commands



Drawing tabs: quick access to all open drawings



Above the Drawing Area

The figure above explodes the top part of BricsCAD's interface into horizontal bars. Let's take a look at them, from top to bottom:

Title bar — reports the name of the program and current drawing file, as in “BricsCAD - [Drawing1.dwg]”

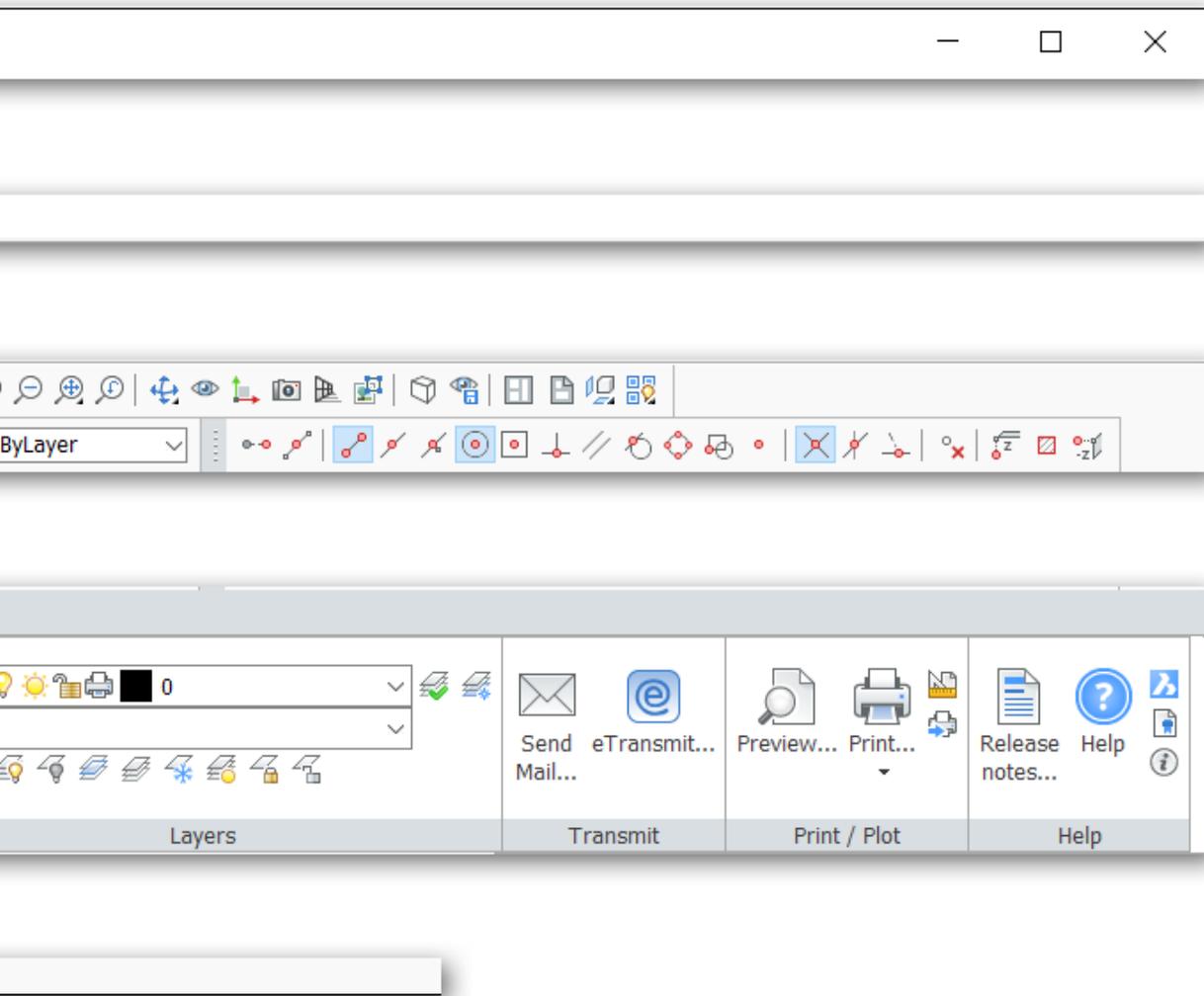
Menu bar — contains pull-down menus for accessing commands logically

Toolbars — collects buttons labeled with *icons* (miniature pictures) into bars of the same type of commands

Ribbon — collects tabs and panels with icons that execute commands

Drawing tabs — switches between open drawings quickly

Let's take a look at each one to better understand their function in BricsCAD.



TITLE BAR

The title bar displays the name of the BricsCAD program, its edition (such as “Platinum”), and the drawing you are working with, such as...

[Office Fixtures.dwg]

If the drawing is read-only (cannot be saved by the same name to its original file location), then a note is added to the file name that looks like this:

[Office Fixtures.dwg (Read-Only)]

The title bar has a couple of hidden tricks. To quickly maximize the BricsCAD window, *double-click* the title bar; to restore the window, double-click the title bar a second time. To quickly open a drawing, drag it from the file manager to BricsCAD’s title bar.

At the far left end of the title bar is a rarely used menu that performs the same function as the trio

of buttons at the other end: they — minimize and maximize or  restore the BricsCAD window, and  exit BricsCAD.



Left: Control menu at left end of title bar; right: equivalent control buttons at right end of title bar.

MENU BAR

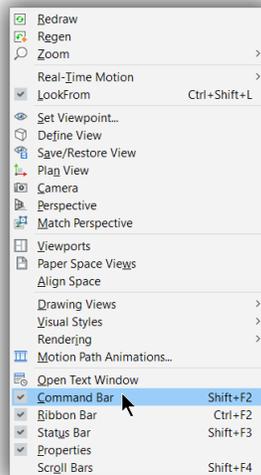
The operations of BricsCAD’s menu bar and its menus are identical to that of other programs that run on Windows, MacOS, and Linux. If you do not see the menu, enter the **MenuBar** variable:

: **MENUBAR**

New current value for MENUBAR (Off or On) <Off>: **on**



Click the words on the menu bar, such as **View**, to reveal the dropdown menus, such as this one:



Some notes on what you see in the menu:

- ▶ The > marker indicates submenu, which group together command options. Submenus can also have sub-submenus!
- ▶ To the right of command names are sometimes *keystroke shortcuts*. For example, to the right of the Command Bar is **Shift+F2**. Instead of selecting items from menus, you can also do it by pressing function keys — Shift+F2, in this case. Here’s how shortcut keystrokes work: (a) Hold down the **Shift** key, and then press **F2**.
- ▶ The check mark in front of a command means it is a *toggle*. “Toggle” means to turn on and off. The presence of check mark means the item is turned on; the lack of a check mark means it is turned off.
- ▶ The ... after a command name means clicking it will open a dialog box.

The check mark and the keystroke shortcuts are examples of the symbols used by menus to indicate special meanings. They indicate the status of commands, as shown by the table below:

Menu	Symbol	Example	Meaning
1t...	... (ellipsis)	Named Views...	Indicates command displays a dialog box
▶	▶ (arrowhead)	Zoom ▶	Indicates the presence of a submenu
☑	✓ (check mark)	✓ Clean Screen	Indicates that the command is turned on
t+F	+ (plus)	Shift+F2	Uses the Shift key for command shortcuts
C _o	_ (underline)	C _o mmand Bar	Uses the Alt key to access menu items

TIPS If you pick a menu item accidentally, you can “unselect” it by picking it a second time.

Some users find pressing keys on the keyboard faster than selecting items from the menu or ribbon.

You can access the menu bar without a mouse! Here’s how to do it:

1. Hold down the **Alt** key. On the menu bar, notice that each word has a letter underlined, such as **V**iew. See figures on the facing page.
2. Press the **v** key to drop the View menu. Notice that every menu item has a letter underlined.
3. Press a key, such as **c** key to execute the **C**ommand Bar command.

Advanced users may be interested in customizing menus and other parts of BricsCAD with the **Customize** command; see the *Customizing BricsCAD* ebook, which is available for purchase from <http://www.worldcadaccess.com/cb8>.

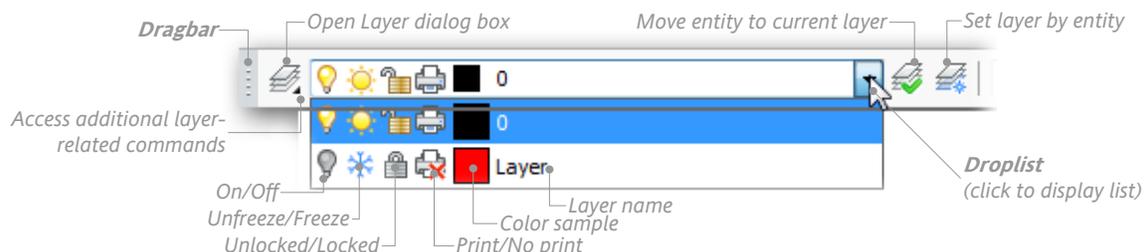
TOOLBARS

Below the menu bar might be several toolbars. *Toolbars* are collections of similar functions — a bar of tools. For instance, the “Draw 2D” toolbar has commands for drawing 2D entities, such as lines, arcs, and rectangles; the “3D Constraints” toolbar contains commands for connecting and sizing entities automatically with 3D geometric and dimensional constraints.



Left: Toolbar for drawing 2D entities; **right:** toolbar for connecting entities with 3D constraints

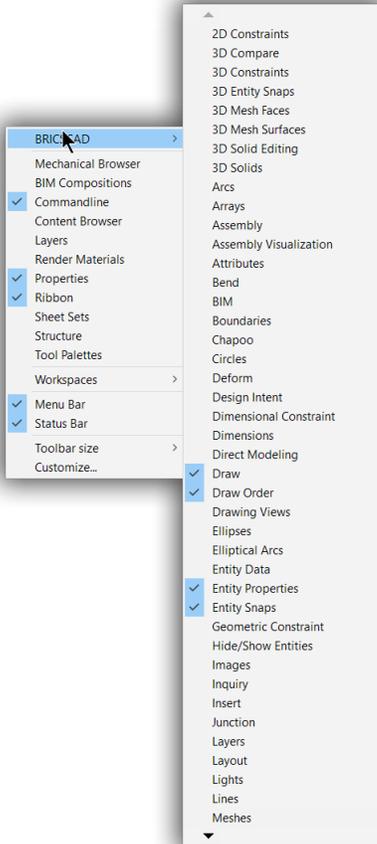
Each toolbar consists of a row of buttons and/or list boxes — these are known as “controls.” Click a button or select an item from a list box to execute the related command. The figure shows examples of of controls you can expect to find on toolbars.



Toolbars have several visual elements for controlling them:



- ▶ Drag bar at the left end of a docked toolbar lets you move the toolbar around the screen. To drag a floating toolbar, grab it by its title bar.
- ▶ Flyout (small black triangle) displays an embedded toolbar; hold down on the button to see the flyout
- ▶ Droplist (a.k.a. list box) lists items; click the arrow at the right end of the box to access the list.



BricsCAD has more than thirty toolbars, but you see only a few of them now. The visible ones are placed along the top and side edges of the drawing area. To see the complete list of toolbar names, right-click any toolbar or the ribbon, and then choose **BRICSCAD**.

The shortcut menu that appears lists the names of all the toolbars; see the figure on the side. Those names prefixed with a check mark are currently displayed. You can *toggle* (switch on or off) the display of a toolbar by selecting its name from the list.

BricsCAD lets you change the look of the icons displayed by all toolbars, the function of the icons, and even the shape of the toolbar. These operations are described in the *Customizing BricsCAD* ebook.

Toolbar Buttons and Macros

Buttons have small pictures called “icons.” Icons are pictorial representations of commands. For example, one button on a toolbar shows the  icon, which represents... Well, what does it represent? The three sheets of paper are the symbol for layers, but the blue star (or is it a snowflake?) is not as clear.

Because icons are pictures, their meaning is not always clear. For this reason, BricsCAD also displays word descriptions. Pass the cursor over an icon, and then wait for a second. A small tag, called a “tooltip,” appears; below, the tooltip tells you that the button executes “Set Layer by Entity” (LayMCur command).



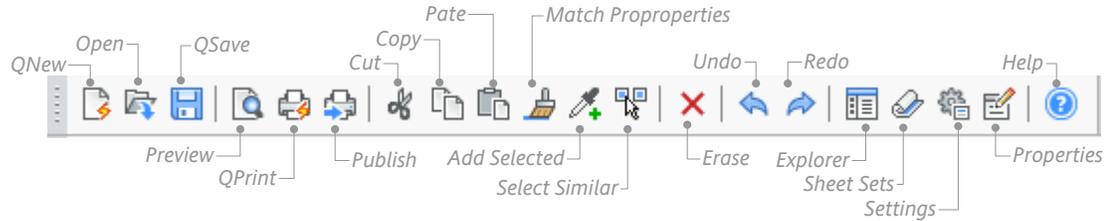
Keep the cursor over the icon, and look down at the status line at the very bottom of the BricsCAD window. It displays a one-sentence description of the button’s meaning.

Sets the current layer to that of the selected entity

In other words, when you click the button, BricsCAD prompts you to choose an entity in the drawing, and then changes the working layer to that belonging to the entity.

The Standard Toolbar

Of all the toolbars, the topmost one contains those buttons that you are probably familiar with from other Windows or Linux applications — as well as a few unique to BricsCAD. This one is called the “Standard” toolbar, because it is standard to most Windows and Linux applications. It’s important enough that I point out the meaning of all its buttons.



From left to right, the icons have the following meaning:

QNew creates new drawing files; Q is short for “quick.”

Open opens existing drawing files.

QSave saves the current drawing.

Preview shows what the drawing will look before it is printed or plotted.

QPrint immediately prints the drawing to the default printer; no dialog box is displayed.

Publish prints collections of drawings.

Cut (CutClip command) copies entities to the clipboard, erases them from the drawing; “clip” is short for clipboard.

Copy (CopyClip command) copies entities to the clipboard.

Paste (PasteClip command) pastes graphical or text data from the clipboard into the drawing.

Match Properties (MatchProp command) copies properties from an entity and applies them to other entities.

Add Selected (AddSelected command) adds additional entities to the selection set.

Select Similar (SelectSimilar command) selects additional entities similar to the first one selected.

Erase deletes the selected entities.

Undo (U command) undoes the last command(s).

Redo redoes the last undo.

Explorer opens the Drawing Explorer for controlling named entities, such as linetypes and blocks.

Sheet Sets (SheetSet command) opens the Drawing Explorer for creating and modifying sets of sheets.

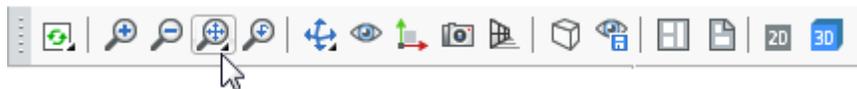
Settings opens the Settings dialog box for adjusting the values of all system variables.

Properties opens the Properties pane; reports and edits the properties of selected entities.

Help displays the help window.

Flyouts

Some toolbar buttons contain *flyouts*, which are sub-toolbars containing two or more additional buttons “hidden” underneath. When you take a close look at the **View** toolbar...



...notice that in the lower-right corner of the **Zoom Extents** button is a tiny triangle:



The  triangle indicates that the button contains the flyout, a sub-toolbar with additional buttons. To see how a flyout works, move the cursor over the Zoom Extents button:

1. Move the cursor down to the button you want, keeping the left mouse button depressed.
2. Let go of the mouse button.

Notice that the command is executed; the button you selected now appears on the toolbar. It can be a bit tricky accessing a flyout the first few times, so practice this procedure until it works.



Left: Holding down on the button displays the flyout; **right:** Selecting a button from the flyout

TIPS Toolbars can be dragged around BricsCAD. Toolbars can stick to any side of the drawing area or float anywhere on the desktop. If your computer has two monitors, you may want to drag the toolbars to the second one to create a larger drawing area. BricsCAD remembers toolbar placement.

You can make the icons larger and smaller. Right-click any toolbar, select **Toolbar Size**, and then choose **Small icons**, **Large icons**, and **Extra-large icons**. Each icon size is twice as large:

Small = 16x16 pixels

Large = 32x32 pixels

Extra-large = 64x64 pixels

(NEW IN V17) The extra-large icons are meant for very high-resolution computer screens, such as 4K.



Droplists

Droplists in toolbars provide instant access to useful lists, such as the names of layers and of colors.

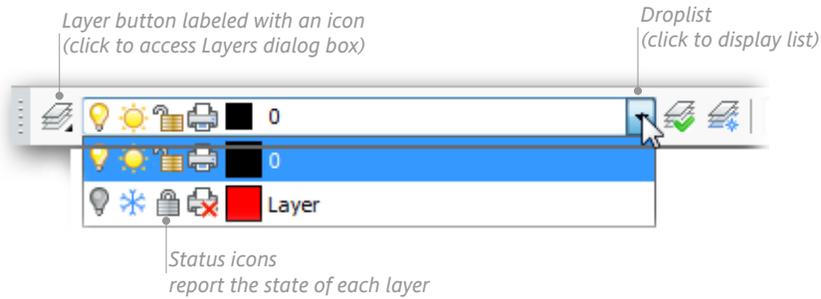


From left to right: Droplists access layer names and their properties; colors; linetypes (patterns); and line weights (entity widths)

Droplists operate the same, whether they are found in a toolbar, ribbon panel, or Properties panel. (There are no droplists in menus.) See the description of the Properties panel later in this chapter. The exception is, however, the layers droplist. It is found in the Entity Properties toolbar and the Home > Layers panel. (You learn more about layers in Chapter 3.)

TIP If the toolbars are turned off, such as in a ribbon-only environment, then use the Properties panel to view and change the properties of entities.

Here is what the Layers droplist looks like in a toolbar. (The ribbon version is similar.)



To change the status of a layer, open the droplist and then click one of the symbols next to a layer name. The symbols have the following meanings:



Light bulb — turns the layer on and off

Sun or Snowflake — thaws and freezes the layer in all viewports

Sun on page — thaws and freezes the layer in the current viewport only; this symbol appears only when the drawing is in paper space

Padlock — unlocks and locks the layer

Printer — prints or doesn't print the layer

Square — specifies the color assigned to the layer (black in the figure above)

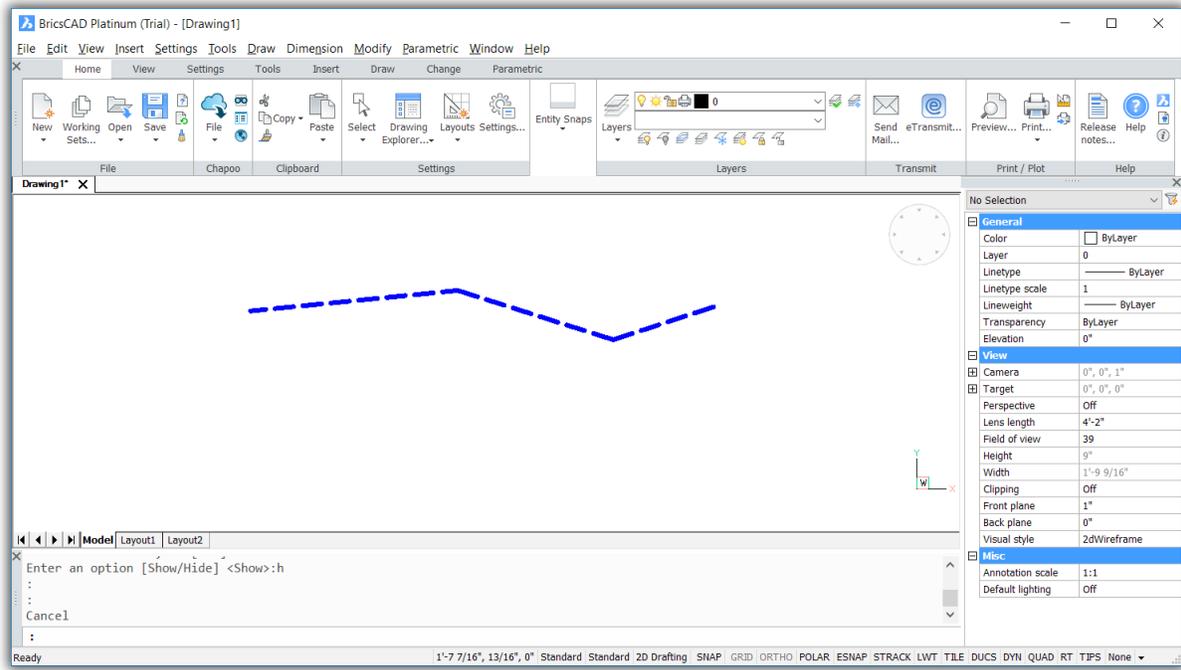
Name — specifies the name of the layer ("o" in the figure above)

TIP Click the color square to access the Colors dialog box, which lets you change the layer's color.

Changing Properties

The two big jobs in CAD are making and editing drawings, but following closely in third place is viewing and changing properties. Properties are things like the color and location of entities. The Properties panel provides immediate feedback of the properties of the entities with which you are working. It parks on the edge of the drawing area, so it is always handy while not taking up too much space.

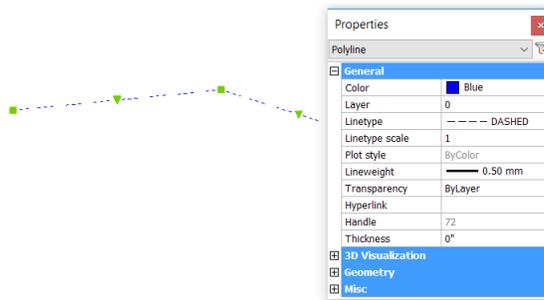
If the Properties panel is not visible, you turn it on with the Properties command. Look for it at the right edge of the drawing area.



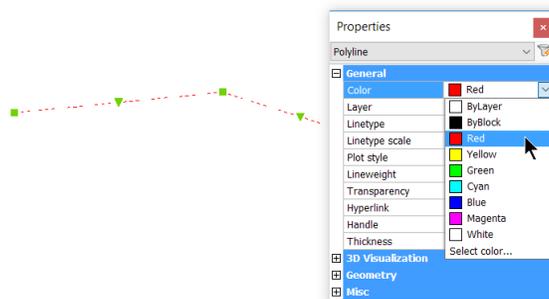
Here the ways in which you can use the Properties panel to view and change properties of entities:

When no entities are selected, then the Properties panel reports the current (active) property names, such as the name of the current layer, color, lineweight, and linetype. For instance, the default color is usually “ByLayer,” meaning that the current layer defines the default color. See figure above.

When an entity is selected, the Properties panel reports the properties associated with the entity. For example, select a blue, dashed polyline. The panel reports the color (Blue) and linetype (Dashed).



When an entity is selected and you choose a different property from the Properties panel, then the entity takes on the new property. For instance, select “Red” from the Color droplist, and the polyline turns red.

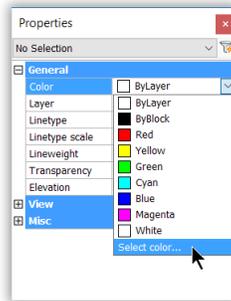


In the same way, you change other properties of the selected entity — elevation (distance in the z direction), transparency (level of see-through-ness), and even the coordinates of its location in the drawing.

Some droplists in the Properties panel contain extra functions. Here is how to access them:

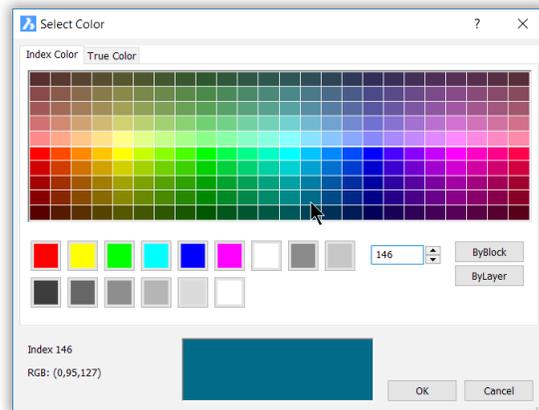
Adding Colors. Initially the Color droplist shows nine colors. If you want a different color from these nine, follow these steps:

1. Go to the end of the Color droplist, and click then **Select Color...**



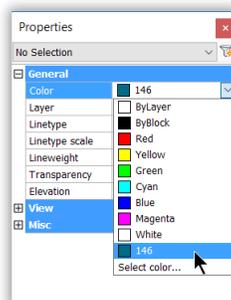
Choosing the Select Color option from the Color droplist

2. Notice the Select Color dialog box, Choose a color from one of 255 “index” colors (standard colors among BricsCAD and AutoCAD uses) or 17.6 million “true” colors (standard in the broader world of graphics).



Select Color dialog box showing the Index Color tab and its 255 colors

3. Click **OK**. Each color you choose from this dialog box is added to the droplist, so that you can reuse them in the future.



New color added to the Color droplist

Adding Linetypes. You use the Linetypes droplist to change the linetype of selected entities, such as dashed or dotted. But in new drawings it shows only three linetypes — Continuous, ByLayer, and ByBlock. The steps to add linetypes are similar to that of colors:

1. Go to the end of the Linetypes droplist, and then click **Other...**
2. Notice the Load Linetypes dialog box. Choose one or more linetypes from the dialog box. To choose more than one at a time, hold down the **Ctrl** key.
3. Click **OK**. Their names of the linetypes you picked are added to the end of the droplist.

TIP To remove unwanted linetypes, use the **Purge** command. Unwanted colors cannot be removed from drawings.

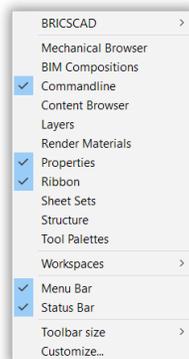
But! It is good CAD drafting practice to not override properties with these droplists. The better way to assign colors and linetypes is through layers. Indeed, colors and linetypes named ByLayer and ByBlock have a special meaning in BricsCAD:

- ▶ **ByLayer** — entities take the color and linetype defined by the layer they reside on.
- ▶ **ByBlock** — entities take the color and linetype defined by the block to which they belong

This is why you often see “ByLayer” in property droplists.

Other Panels

BricsCAD has more panels, in addition to the Properties panel. You can see the full list when you right-click a toolbar or the ribbon:

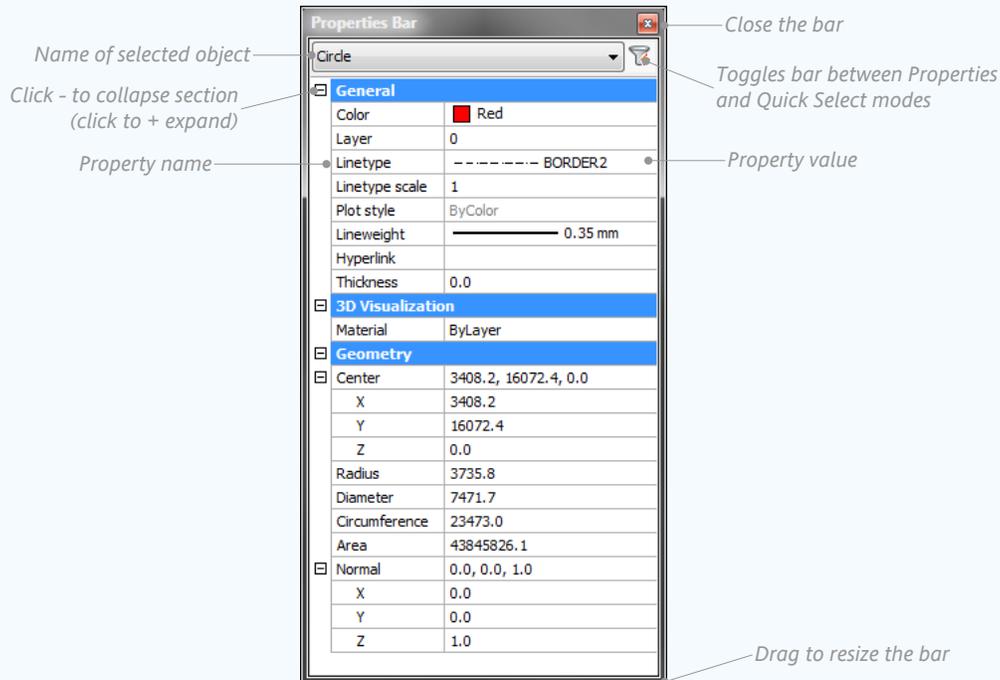


A blue check mark means the panel is open. The panels perform the following functions:

Panel	Command	Displays
Mechanical Browser	bmBrowser	Assembly and parts browser, libraries
BIM Compositions	bimComposition Dialog	BIM compositions manager
(NEW IN V17) Content Browser	ContentBrowserOpen, ContentBrowserClose	Content of user-selected folders
(NEW IN V17) Layers	LayerPanelsOpen, LayerPanelsClose	Layers panel
Render Materials	Materials	Rendering materials manager
Sheet Sets	Sheetset, SheetsetHide	Sheet set manager
(NEW IN V17) Structure	...	Drawing structure browser
Tool Palettes	Toolpalettes, ToolpalettesClose	Tools palette

USER INTERFACE ELEMENTS OF PANELS

Panels or “palettes” are so important to BricsCAD that I am going to give you this quick tour of their user interface features. (Prior to V17 many of them were called “bars.”) Once you learn these for the Properties panel, you can use them with other panels, as well. (See Chapter 6 for how to use the Properties panel.)



	Drag the title bar to relocate the panel on the screen
	Click the x button to close the panel
	Click the droplist to access other entities (if any)
	Click the + node to expand a section
	Click the - node to collapse a section
	Click on a property value to modify it
	Drag the edge of the panel to change its size, smaller or larger

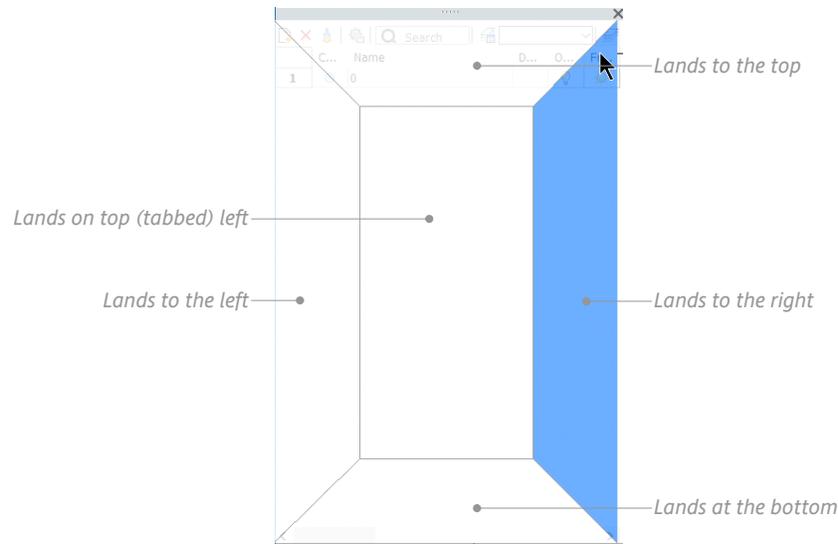
SHORTCUT KEYSTROKES FOR PANELS

Several panels and bars can be opened and closed using shortcut keystrokes. Here is a summary of them:

Action	Command	Shortcut Keystrokes
Open or close Command bar	CommandLine	Ctrl+9 Shift+F2
Open or close Properties panel	Properties	Ctrl+1 Ctrl+Shift+P
Open or close Ribbon bar	Ribbon	Ctrl+F2
Display or hide status bar	StatBar	Shift+F3
Display or hide scroll bars	ScrollBar	Shift+F4

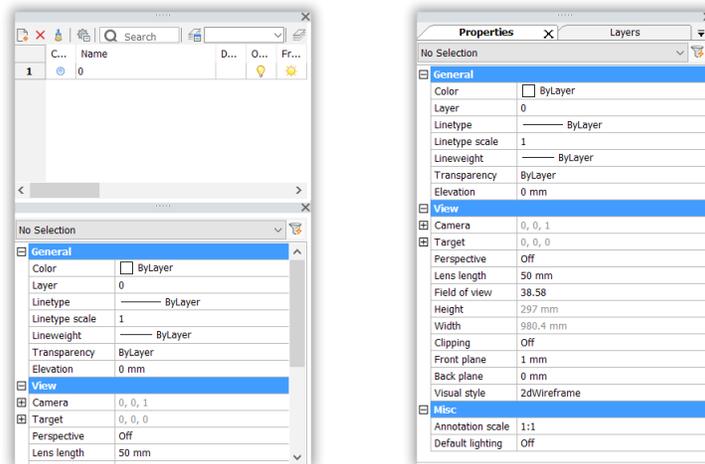
(NEW IN V17) Panels can be stacked on top of each other or else placed next to one another. To do so, follow these steps:

1. Drag a panel by its title bar on top of another panel. Don't let go of the mouse button!
2. Notice the five trapezoids. One is colored in with blue. If you let go of the mouse button, it will land at that spot:



Trapezoids indicating the land location for a panel

Here is what two panels look like when one panel lands above another (shown at left) or the panel is stacked on top of another (shown at right):



- › When above or beside each other, a drag bar appears between them that lets you change their size
- › When on stacked, tabs appear that let you switch between them

To unstick panels, grab one by its title bar and then drag it away from the others. To close a panel, click the x in its upper right corner.

These panels can be docked: Command Bar, BIM Compositions, Content Browser, Layers, Mechanical Browser, Properties, Render Materials, Ribbon, Sheet Sets, Structure, and Tool Palettes.

RIBBON TABS AND PANELS

The ribbon dominates the upper part of the user interface. It's kind of like a series of overlapping toolbars, where of *tabs* segregate the “toolbars” into groups of functions. Each tab has is further segregated by a series of *panels*, and each panel contains a group of buttons, flyouts, and droplists — just like toolbars.

If you do not see the ribbon, enter the **Ribbon** command:

: ribbon

Shown below is the ribbon's standard Home tab.



Home tab showing nine tabs of command groups

Each tab shows a different group of panels. Click the name of a tab to switch to it. Here is the **View** tab.



View tab

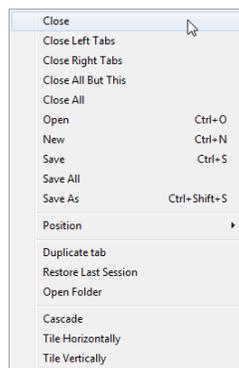
Because Bricsys wrote its own version of the ribbon interface, it is equally available on the Windows, Mac, and Linux versions, unlike all other CAD systems. You customize the ribbon through the **Customize** command.

DRAWING TABS

Drawing tabs let you switch quickly between open drawings, and provides a shortcut to file-related commands, such as Open and Close. BricsCAD calls it “Document Tab.”



To access the commands, right-click a drawing tab:



Most of these commands are familiar to you, but here are a couple that are unique to BricsCAD:

Close Left Tabs — closes all drawings to the left of this tab. This is useful for closing older drawings, ones that were opened earlier

Close All But This — closes all other drawings, except the current one. I could find this useful when I open an entire folder's worth of drawings, and then want to keep just one open

Save All — saves all drawings at once

Duplicate Tab — makes a copy of the current drawing, naming it *Copy_name.dwg*

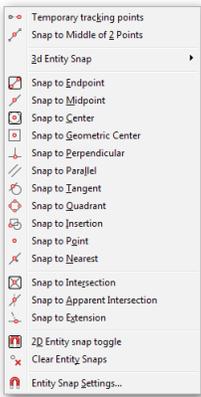
Open Folder — opens the folder from which the drawing was opened

There are two variables that control drawing tabs. **DocTabPosition** places the tab at the top, bottom, left, or right of the drawing area. **ShowDocTabs** turns the tab row on and off.

In the Drawing Area

The drawing area is in the center of the BricsCAD window, as shown above. Here I will tell you about some of the user interface elements you find in the drawing area.





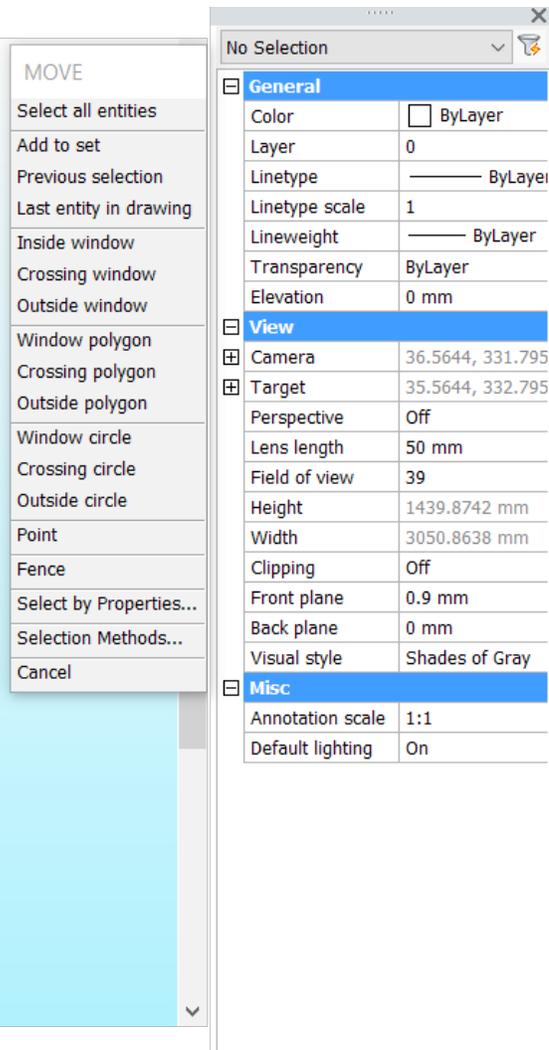
SHORTCUT MENUS

Scattered all about BricsCAD are accesses to shortcut menus. Nearly any place you right-click, a menu will pop up with a list of commands. The commands usually are relevant to the spot you right-click. Holding down the **Shift** or **Ctrl** keys sometimes displays other shortcut menus. In later lessons, I'll point out shortcut menus that are useful to the work at hand. I don't detail them all here, because there are so many. The figure shows what happens when you hold down the **Ctrl** key and click the right mouse button: you access the entity snap modes, as describe later in this book.

QUAD CURSOR

Status bar QUAD
Shortcut F12

The Quad is unique to BricsCAD in the way that it incorporates drawing and editing commands. This multifunction cursor takes its cue from the "heads-up" style of computer interface design, placing in the drawing area many useful commands.

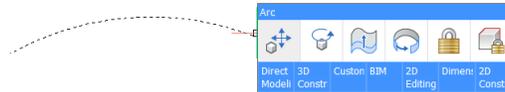


The Quad is normally not visible; most of the time, you see the standard tri-color crosshair cursor or arrowhead cursor. When you pass the cursor over an entity, however, the quad cursor appears, first as a single button; see figure below.

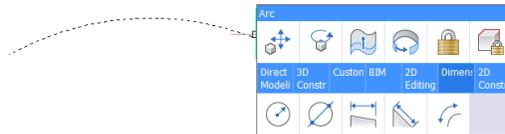


(If it does not appear, then turn it on by clicking **QUAD** on status bar or pressing function key **F12**.)

When you move the arrow cursor onto the sole button, the quad cursor expands to show additional buttons, usually for commands most commonly used with the nearest entity.

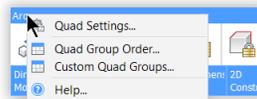


In addition, there are blue boxes for groups of additional buttons. Some groups are for common operations, while others are specific to the entity. To access the additional buttons, pass the cursor over a blue box. Click a button to execute its command.



The Quad changes its content, depending on the nearby entity and the workspace. BricsCAD comes with sets of predefined Quads for the various workspaces.

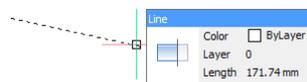
To customize the Quad, right-click it to access the following shortcut menu:



Quick Properties

When Quick Properties are turned on, the properties of an entity are displayed by the Quad. To turn this in, click **RP** on the taskbar. (RP is short for “rollover properties” — I know, the name keeps changing. Sigh.)

Hover the cursor over an entity, and the Quad displays some of the properties of it.



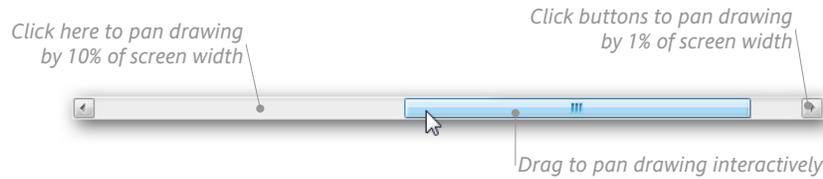
Quad displaying some properties of a line

If you want the Quad to display other properties, then use the **Customize** command’s Properties tab to specify the properties for every entity.

SCROLL BARS

A scroll bar rests at the right edge of the drawing area. It lets you pan the drawing up and down. A second one is along the bottom of the drawing area; it pans left and right. Normally, they are turned off, but I find them handy. To turn on scroll bars, enter the **ScrollBar** command.

Scroll bars are limited to panning left-right and up-down.

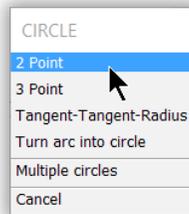


Using the scroll bar to pan the drawing

As a more flexible alternative, hold down the middle button (or the roller wheel in the middle of the mouse), and then drag the mouse around. This action pans the drawing in any direction during any command.

PROMPT MENU

The Prompt menu shows the prompts available for the current command. This menu appears in the upper right corner (usually) of the drawing areas. It is useful when the Command bar is turned off, because then you see what options are available for each command.



To select an option, just pick it from the list shown. When the command ends, the Prompt menu disappears.

LOOK-FROM CONTROL

BricsCAD has a LookFrom widget in the upper right corner of the drawing area. When you pass the cursor over the widget, small triangles appear, as does the the preview of a chair. Pausing the cursor over a triangle shows what the 3D view will look like:



Left: LookFrom control with no cursor interaction...; right: ...and when the cursor is over one of the small triangles

Clicking the triangle changes the 3D viewpoint.

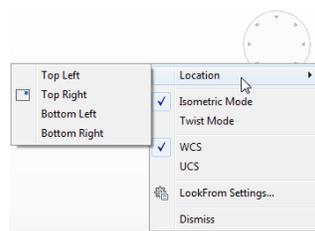


The green dot indicates the cursor position, kind of like a laser pointer:

TIPS To see the bottom views, hold down the **Ctrl** (or **Cmd** in Mac) key.

Click the center of the LookFrom control to return the view to its home view. This is particularly helpful in Twist mode.

There are two ways you can change the way the LookFrom control operates. The easier one is right-click the control, and then choose an option from the shortcut menu.



Most of the options in the shortcut menu are straight-forward, but I do want to explain the difference between Isometric and Twist modes:

- > **Isometric** mode is like using the Viewpoint or View commands
- > **Twist** mode is like using the RtRotF (real time view rotation) command



Left: LookFrom in isometric mode; right: And in twist mode

The other method is to enter the **LookFrom** command, from which you can turn off (and on) the control and access its settings:

```
: lookfrom  
LookFrom [ON/OFF/Settings] <ON>:
```

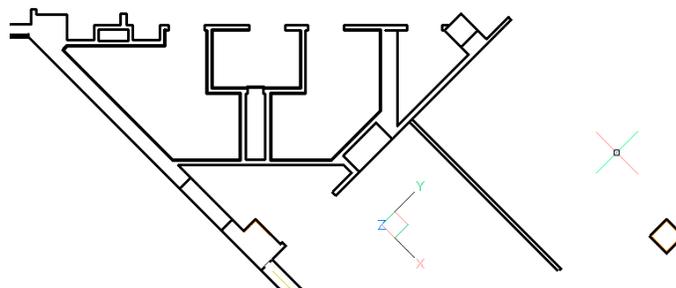
The **Settings** option opens the Settings dialog box at the LookFrom section. Here you can adjust the properties of the widget, such as its translucency and position. Of particular interest is the number of isometric viewpoints it can display, which is set through “Direction Mode” or the **LookFromDirectionMode** variable. The following table shows you the options:

LookFromDirectionMode		Number of Views
0		6 orthogonal views
1		14 views; no flat views of corners
2		18 views; top down corners
3		26 views; eight top down corners

THE UCS ICON IN 2D DRAFTING, 3D MODELING, AND DUCS

I described the function of the UCS (user-defined coordinate system) icon in the last chapter. Here I want to talk a bit about how the UCS is used — rarely in 2D drafting, but a lot in 3D modeling. In short, a UCS lets us locate twist the x,y,z axes in any direction in space. This makes it easier to draw and edit at unusual angles and on slanted surfaces.

Static UCSs for 2D. In 2D drafting, for example, when we draw things at an angle, such as the angled wing of a building, then we use a USC. Changing the coordinate system to match the angle of the building rotates everything else as well. In the figure below, I rotated the UCS icon to match the angled walls; notice that the crosshair cursor has matched the angle.

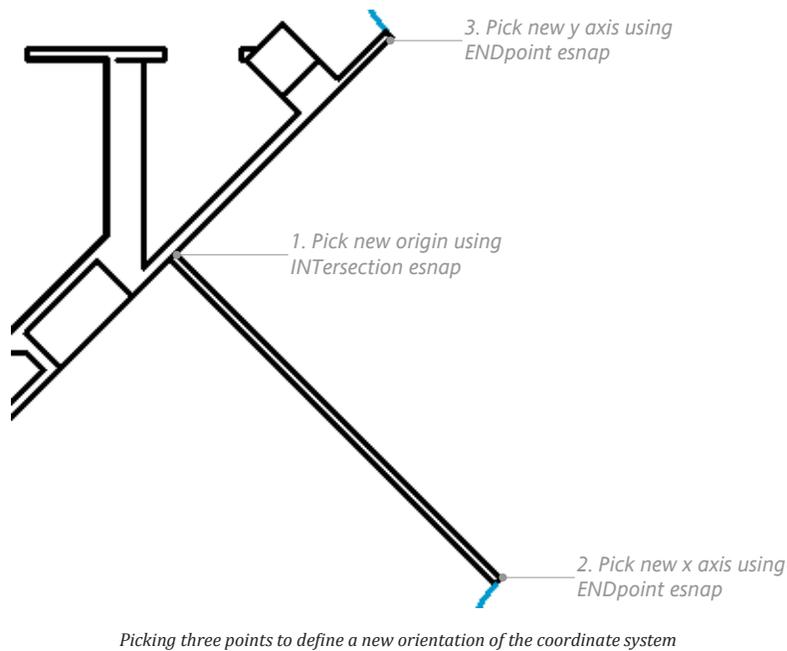


Rotated coordinate system allows easy drawing at an angle

To change the angle of the UCS icon, you tell the **UCS** command three things:

1. The new origin point (o,o)
2. The new direction of the x axis (determines the angle)
3. The new direction of the y axis (determines the orientation)

From these three, BricsCAD figures out the direction of the z axes using the right-hand rule. Here is how to use the UCS command in 2D drawings:



: **ucs**

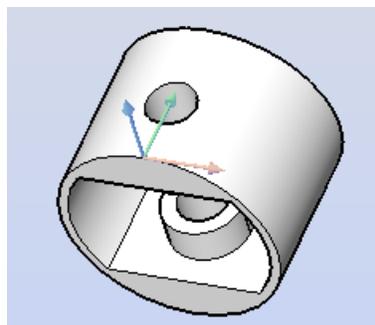
Specify origin of UCS or Face/Named/Entity/Previous/View/X/Y/Z/ZAxis/Move<World>: (Pick a point like the intersection of two lines; BricsCAD turns on INTersection entity snap automatically)

Point on X-axis or <Accept>: (Pick a point along the new x axis; BricsCAD turns on ENDpoint entity snap automatically)

Point on the XY-plane with positive Y value or <Accept>: (Pick a point along the new y axis; BricsCAD turns on ENDpoint entity snap automatically)

If you use the same UCS often, then you can give it a name. This lets you switch back to it quickly. Named UCSes can be managed through the **ExpUcs** command, which displays the Coordinate Systems section of the Drawing Explorer.

Dynamic UCSs for 3D. The same system works for manually changing the UCS in 3D modeling. Why would you want to change the UCS in 3D? Because CAD is still fundamentally 2D, even when in 3D. To draw on a face, the UCS needs to match the orientation in space of the face. BricsCAD, fortunately, can do this automatically through dynamic UCS. Here is a brief example. In the figure below, the UCS icon is in the “world” position:

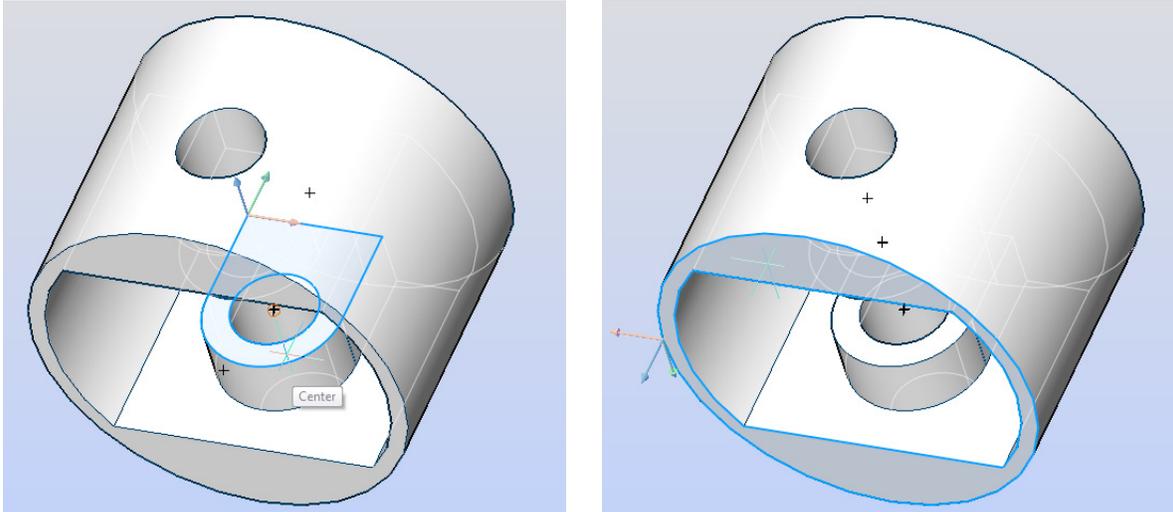


UCS icon at its home position

To, say, write some text on the bottom of the piston, the UCS needs to relocate to the flat face. To do so, first turn on DUCS (dynamic UCS) mode by clicking the **DUCS** button on the status bar (so that it looks black). Then start the **Text** command:

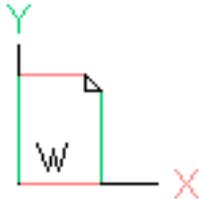
```
: text  
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: (Pick a face on which to write)
```

Now move the cursor around faces of the piston. Notice how the UCS jumps to match the orientation of the face. I show two of these in the figures below.



Left: UCS dynamically placed on one face...; right: ...and on another face

UCSs in Layout Mode. In paper space of layout modes, the UCS icon turns into something that looks like a sheet of paper.



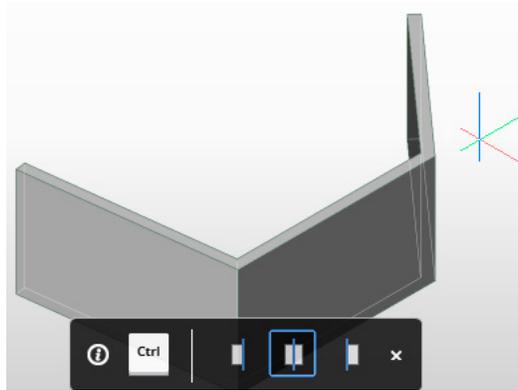
It serves no purpose. In the early days when layout mode was first introduced, the CAD interface was such that it could be difficult for users to determine if they were in model or paper space. So the UCS icon switched to a triangle in some CAD programs, and a piece of paper in BricsCAD. Today, however, the user interface makes layout mode quite clear, and so the paper icon remains primarily for sentimental reasons.

TIPS WIDGET

“Tips” are interactive tooltips that report options that might otherwise be unknown to users. For example, the following Tips widget appears during the Polysolid command.

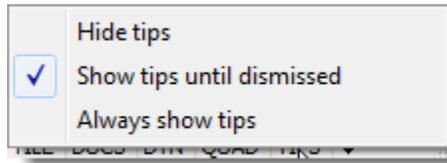
Notice that it shows several icons. The Ctrl “button” is a reminder that by holding down the **Ctrl** key during the command, you can change the justification of polysolids between left, centered, and

right. To get a brief description of the purpose, pause the cursor over the Tip.

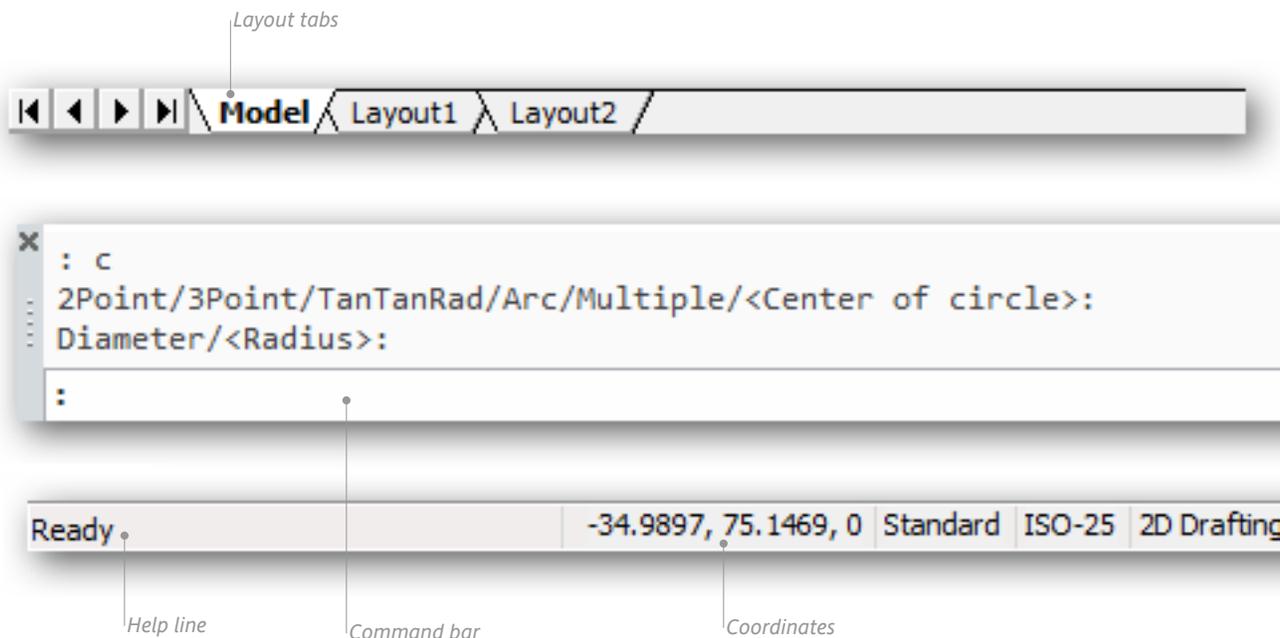


Tips widget appearing during the Polysolids command

To dismiss the Tip, click the small **x** at the right end. (The **Ctrl** button itself does nothing; neither does the *i* icon.) The display of the Tips widget is toggled through the **TIPS** button on the status bar. Right-click the button for options.



Options for controlling the Tips widget



Below the Drawing Area

At the bottom of BricsCAD are several more areas of information:

Layout tabs — switch between model space and layouts quickly

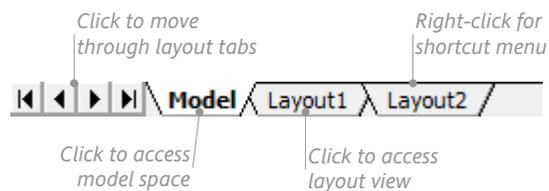
Command bar — accepts typed input and displays prompts; see Chapter 1 for details

Status bar — reports on the status of the drawing and provides on-off toggles

Let's examine the function of each area.

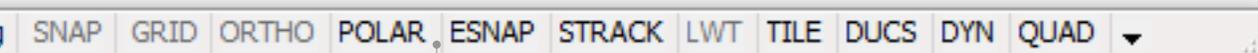
LAYOUT TABS

Under the drawing area are three tabs labeled Model, Layout1, and Layout2. You may have seen similar tabs in other Windows or Linux software, such as the sheet tabs in LibreOffice Calc or Excel.



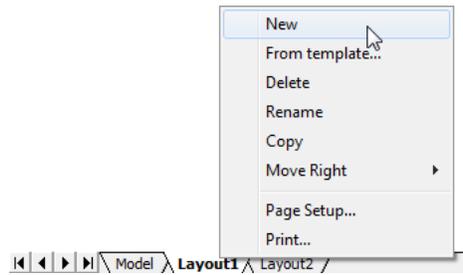
In BricsCAD, these tabs switch the drawing view among *layouts*. Layouts allow you to define how the drawing will look before it is plotted, such as positioning its location on the paper, adding a drawing border, or specifying shades of gray instead of colors. The arrow buttons (to the left of the tabs) are for when there are too many tabs to fit along the bottom of BricsCAD.

Horizontal scroll bar



Status bar

To access a shortcut menu of layout options, right-click any layout tab. The options let you create more layouts, rename them, or specify settings specific to a variety of plotter models.



Layouts are discussed in greater detail later in this book.

So, there are drawing tabs to switch between drawings, and layout tabs to switch between layouts inside each drawing.

Scroll Bar

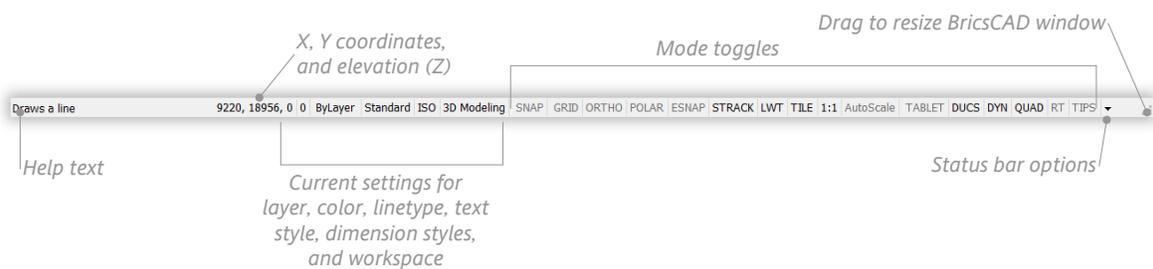
The scroll bar is described earlier in this chapter.

Command Bar

The Command bar is described in the previous chapter.

STATUS BAR

Below the command prompt area is the status bar, which reports the status of the drawing. From left to right, the status bar displays the following info:



Help text — displays a line of helpful text when the cursor is paused on a toolbar button or a menu item.

X,Y Coordinates — 2D coordinates of the cursor's current location in the drawing.

Elevation (Z) — the current setting of the elevation.

Layer — reports the currently active layer name (default = 0)

Color — reports the current color (default = 0; black)

Linestyle — reports the current line type (ByLayer)

Style — reports the current text style (Standard)

Dimension style — reports the current dimension style (ISO-25)

Workspace — reports the current workspace (2D Drafting).

Drawing Settings — reports the status of drawing settings. Each word is an on/off button called a “toggle:”

- > **Gray text** — the toggle is turned off, as for **SNAP** in the figure above.
- > **Black text** — the toggle is turned on, as for **POLAR**.

Click a button to turn the mode on and off.

SNAP — toggles the distance snap

GRID — toggles the display of grid lines

ORTHO — toggles orthographic drawing mode to draw at right angles

POLAR — toggles snapping at common polar angles, such as 15 degrees

ESNAP — toggles all entity snaps (a.k.a. object snaps)

STRACK — toggles snap tracking

LWT — toggles lineweight display

TILE / P:Layout — switches between paper and model modes

Annotation Scale — specifies the current scale factor for annotative entities

AutoScale — toggles whether annotation scales are added automatically

Tablet — toggles tablet mode on and off

DDUCS — toggles dynamic UCS mode for drawing on the faces of 3D entities

DYN — toggles dynamic dimensions mode

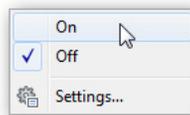
QUAD — toggles the cursor between crosshair and quad modes

RT — toggles the display of rollover tooltips

TIPS — toggles the tips widget

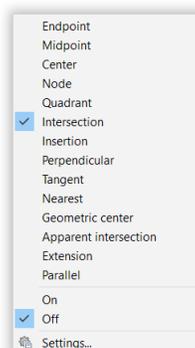
None — selects a GIS coordinate system

When you right-click any of status bar button, BricsCAD displays a shortcut menu. Many have the same set of options: On, Off, and Settings,

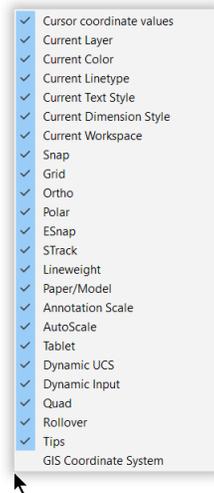


Settings is a shortcut to the dialog box that regulates the mode. For example, to change the settings for the grid, right-click the GRID button and then select Settings; BricsCAD displays the Snap/Grid section of the Settings dialog box.

When you right-click a few buttons, they present a longer list of options so that you can, for example, choose an entity snap.



Click the tiny black triangle  near the right end of the status bar to display a shortcut menu listing all possible toggles.



This lets you turn on and off toggles that you want (or don't want) to see.

Summary

You have learned all about BricsCAD's user interface, from the title bar at the top to the status bar along the bottom, and nearly everything in between

Next, you begin to start working with BricsCAD, beginning with starting a new drawing.

Setting Up A New Drawing

You now prepare the CAD environment for creating new drawings. By the end, you will know how to set up drawings, save your work to disk, and how to exit BricsCAD.

Before creating any drawing with BricsCAD, you must prepare it. Here we have the tutorials show you how to do this:

- Step 1: **Start** a new, blank drawing
- Step 2: **Name** the drawing
- Step 3: Specify the **units** of measurement
- Step 4: Set the **snap** and **grid** spacings
- Step 5: Indicate the drawing **limits**
- Step 6: Create **layers**
- Step 7: **Save** the drawing

IN THIS CHAPTER

- Preparing a drawing for first-time use with a wizard
- Specifying units, angle formats, and drawing settings
- Understanding how layers organize the drawing
- Saving the drawing
- Recognizing the importance of automatic backups

KEY TERMS IN THIS CHAPTER

- Default** — value of a setting defined by the program and unchanged by the user
- File name** — uniquely identifies drawing files and other documents
- Grid** — displays a grid of lines or array of dots as visual guides
- Layers** — organize drawings by segregating common elements
- Limit** — specifies the nominal limits of drawings, and constrains the range of grid marks
- Scratch** — brand-new drawing with no preset parameters
- Snap** — constrains cursor movement to discrete distances
- Template** — a drawing file read by BricsCAD that sets default parameters for new drawings
- Unit** — specifies units of measurement, such as metric, architectural, and engineering
- Wizard** — series of dialog box that step users through a procedure
- Working set** — set of one or more drawings that were previously open in BricsCAD or were saved for future use
- Workspace** — predefined user interface specific to tasks, such as 2D drafting or 3D modeling
- Zoom** — enlarges and reduces the visual size of drawings

USEFUL ABBREVIATIONS

' or ft	Feet
" or in	Inches (12 inches per foot)
ANSI	American National Standards Institute
BAK	Backups of BricsCAD drawing files
DWT	Portion of file names that identify them as template files
DWG	Portion of file names that identify them as BricsCAD drawings
mm	Millimeters (1000mm per meter)
m	Meters

NEW COMMANDS

Command	Alias	Menu Bar	Ribbon Tab
Layer	la	Settings Layers	Home Layers Layers
Limits	...	Settings Drawing Limits	...
NewWiz	ddnew	File New Wizard	...
Settings	rm	Settings Settings	Settings Settings Settings
SaveAs	...	File Save As	...

Before You Begin

To learn how to use BricsCAD, you will work with a drawing based on a landscape plan, creating and modifying the drawing of a yard around a home.

Before beginning this tutorial, you may want to measure your yard and locate major features, such as the house, driveway, and garden areas. If you'd rather not measure your yard or you don't have access to one, then you can follow along with the following sketch, which is the drawing used by the next several lessons.

The tutorial will be in metric units (millimeters and meters), and for North American readers I'll provide imperial units (feet and inches).

STARTING A NEW DRAWING

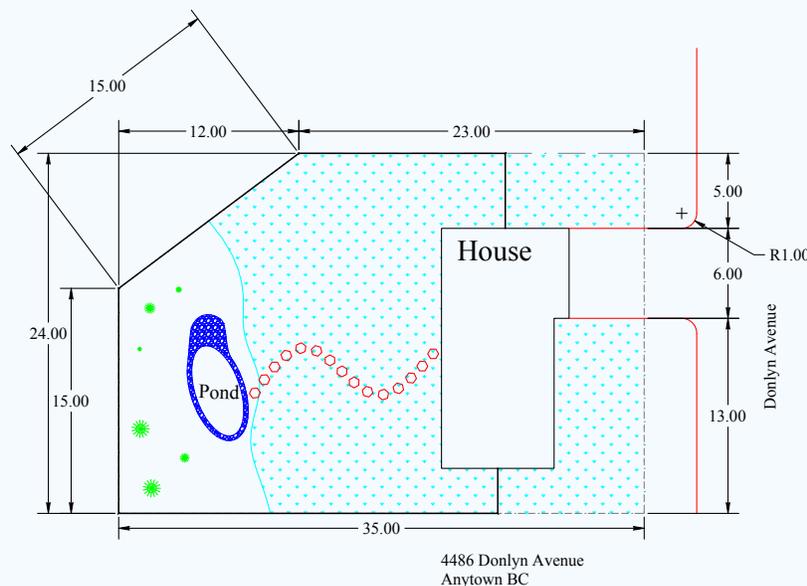
Command	NewWiz
Menu	File New Wizard

When you start a new drawing in BricsCAD, you can start in several ways:

- From “scratch” with a blank drawing
- With a template drawing that presets certain aspects
- Enlist the services of a software wizard — how you will start this tutorial

THIS SECTION'S DRAWING

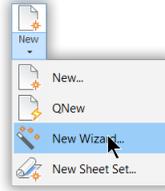
During this book's first few chapters, you create the following drawing. The completed *Yard.dwg* file can be downloaded from <http://www.upfrontezine.com/lb8>. (Use *Imp-Yard.dwg* if you draw with imperial units.)



The *wizard* is a series of dialog boxes that take you through the steps needed to set up a new drawing. It prompts you to select the units, grid, and so on.

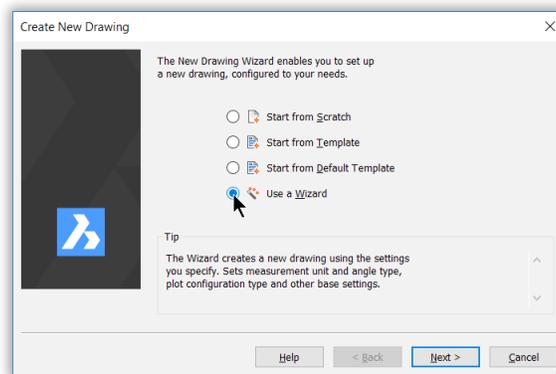
In this tutorial, you create the new drawing with the assistance of the wizard.

1. If BricsCAD is not running, start it now by double-clicking its icon on the desktop
2. Notice the Welcome dialog box. Click **New Drawing**.
3. Start the wizard by taking one of the following steps:
 - > In the ribbon's **Home** tab, click **New** and then **New Wizard**



- > From the menu bar, choose **File**, and then **New Wizard**
- > At the command line, enter the **NewWiz** command
: newwiz

Notice the Create New Drawing dialog box:



Selecting the Use a Wizard option

The dialog box supplies you with these options for starting new drawings or opening existing ones:

Start from Scratch — starts empty new drawings, using either imperial or metric units

Start from Template — starts new drawings based on a template file that you select from a long list with names like *default-mm.dwt* and *Mechanical-imperial.dwt*

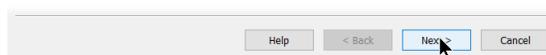
Start from Default Template — starts new drawings with the template file specified by the BaseFile system variable; the current default is *default-mm.dwt*

Use a Wizard — creates new drawings based on settings specified in a series of dialog boxes that follow

4. Select the **Use a Wizard** button by clicking the circle (called a “radio button”) next to it.



5. Click **Next**.



The wizard takes you through a series of dialog boxes that ask you to specify the units, format of angles, and other drawing settings. I'll explain what each of them mean.

Default Settings

First off, you choose the units. Well, the wizard calls them the “default settings.” This is because so many settings in a drawing are affected by the units — metric (decimal) or Imperial (feet and inches). Choosing one type of unit determines many other settings.

Here is how to decide which units to choose:

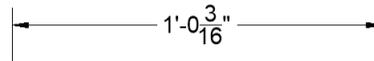
- ▶ If you measured the yard in feet and inches, then click the radio button next to **Imperial (feet and inches)**
- ▶ If you measured the yard in meters and centimeters, then click the radio button next to **Metric**



6. If you are following along with my tutorial drawing, then choose **Metric**, as shown above.
7. Click **Next**.

SELECTING THE LINEAR UNITS

Drafting uses *linear measurements* for things like lengths and areas, as illustrated by the dimension below. (The only other kind of measurement that you work with in CAD are *angular measurements*, and they are used to measure angles.) Linear measurements can be displayed in several different kinds of formats.



Linear dimension showing feet and inches

Look closely at the example measurement below to understand how BricsCAD displays each one. Note that the dash (-) separates feet from inches, while a space separates inches from fractional inches.

Unit Name	Measures In	Example
Architectural	Feet, inches, fractional inches	4'-6 1/16"
Engineering	Feet, inches, decimal inches	4'-6.0625"
Fractional	Inches, fractional inches	54 1/6"
Decimal	Units and decimal units	2128.4449
Scientific	Units, decimal units, and exponents	2.1284E+03

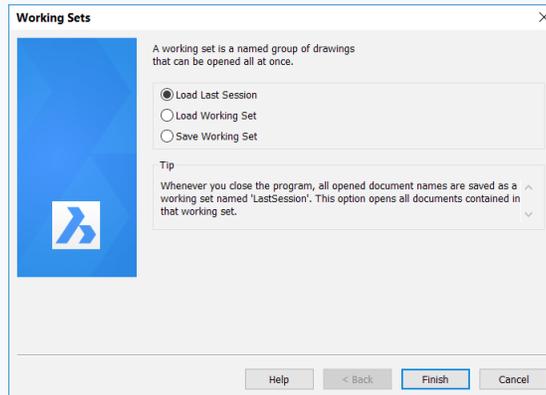
The default is called “Decimal,” which is often used for metric measurements, such as millimeters. BricsCAD shows metric measurement as unitless “decimals,” such as 1.2345. The “mm” for millimeter is not displayed.

Once you select a unit style, BricsCAD displays all measurements in that format. But don’t worry! You can always switch the measurement format to something else. You would do that with the **Units** command.

WAYS TO START NEW DRAWINGS

OPENING WORKING SETS

A “working set” is a list of previously-opened drawings, or a saved sets of drawing file names. To save and open sets of drawings, use the **Workset** command.



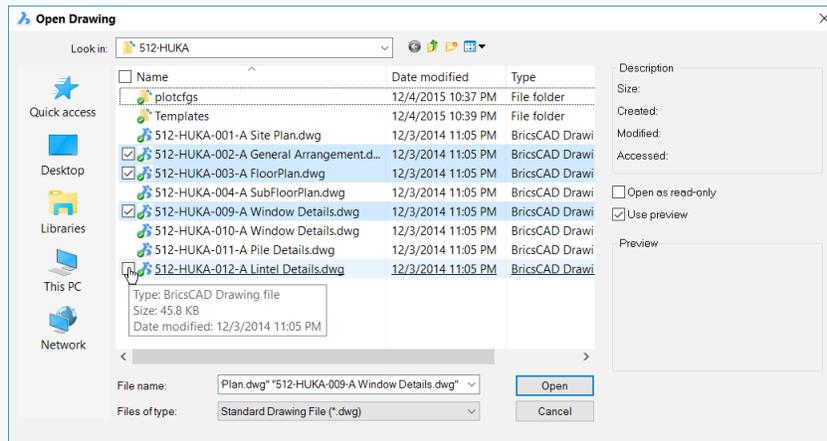
Dialog box for loading and saving sets of drawings

OPENING MORE THAN ONE DRAWING

To select one than one drawing at once, hold down the following keys as you pick them in the Open dialog box,:

Shift — select a contiguous range of .dwg files

Ctrl (Cmd on Macs) — to select non-contiguous files, as shown below



Choosing more than one drawing to open at once

STARTING FROM SCRATCH

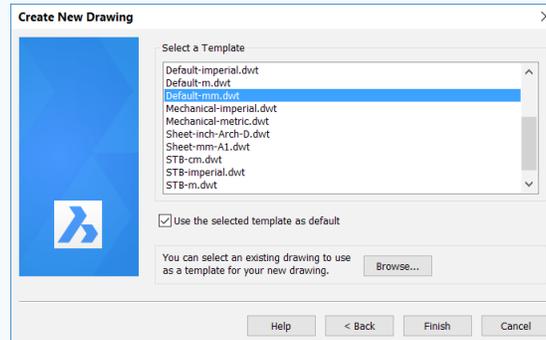
New “empty” drawings employ default settings for all variables, which are read from the following DWT template files:

default-mm.dwt — for drawings with metric units or unitless (millimeters)

default-imperial.dwt — for drawings with Imperial or British units (inches)

STARTING FROM TEMPLATES

New drawings are begun with a DWT template file that you select from the following list:



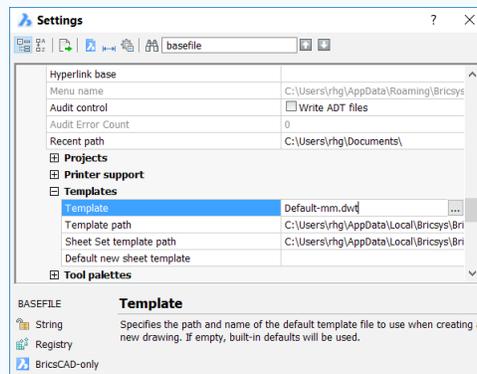
Selecting a template drawing on which to base the new drawing

You can create your own templates: edit the current drawing, and then use the **SaveAs** command. Choose “Drawing Template (*.dwt)” from the **Save As Type** droplist.

STARTING FROM THE DEFAULT TEMPLATE

Start new drawings with the template file specified by the **BaseFile** system variable, usually *default-mm.dwt* if none other is specified. This option is useful when you usually want to start with the same template, such as one specified by your firm or a client.

To change the value of BaseFile, enter the **Settings** command, search for “basefile,” and then click the  button in the Template field. From the Choose a File dialog box, select the .dwt file you wish to employ as the new default template.



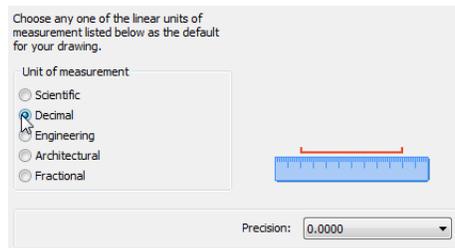
Specifying the default DWT template file name

USING A WIZARD

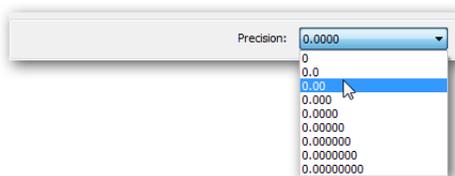
Creates new drawings based on settings specified in a series of dialog boxes. See the tutorial in this chapter.

(The reason you can switch between units is because internally BricsCAD keeps track of measurements with *unitless* real numbers accurate to 14 decimal places; it actually doesn't use metric or Imperial units. For the sake of us humans, however, BricsCAD displays distances and angles in the units we want; the software converts them on-the-fly.)

- For this drawing tutorial, measurements were made in centimeters. Under **Unit of Measurement**, ensure "Decimal" is selected.



- Centimeters are represented by two decimal places. To set the number of decimal places, follow these steps: Next to **Precision**, click the down arrow and then select "0.00."



(For imperial drawings, measurements made to the nearest 1/4-inch are accurate enough for this project. Select **Architectural**, and then choose **0'-0 1/4"**.)

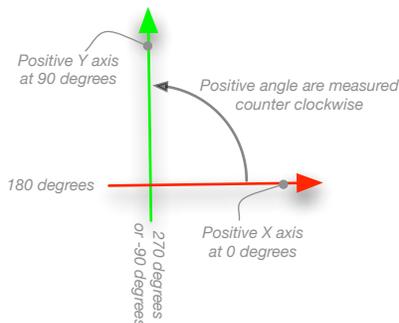
- Click **Next**. As we see next, BricsCAD displays the set of options for measuring and displaying angles.

TIP You are free to enter distances more accurate than a centimeter, because BricsCAD remembers distances to full accuracy. When BricsCAD displays coordinates, however, it will round them off to the nearest 0.01 units, because we told it to in step 9.

SELECTING THE ANGLE STYLE

Before we choose the angle measurement system, I must mention some facts about angles:

- Angles are measured counter clockwise from the positive x axis; see figure below
- 0 degrees is the direction of the positive x axis
- 90 degrees is at the positive y axis



The elements of axes and angles

The Angle dialog box lets you choose from the formats of angular units that BricsCAD should display.

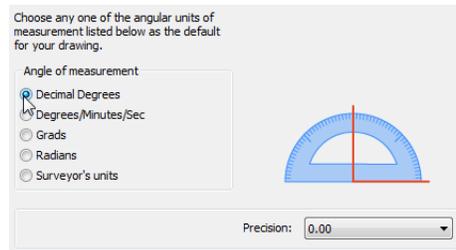
The options available are listed below:

Angle Name	Measures In	Example
Decimal degrees	Degrees and decimals	12.3456
Degrees/Minutes/Seconds (d ' ")	Degrees, minutes, seconds, and decimal seconds	12d34'56"
Grads (g)	Grads and decimals	123.45g
Radians (r)	Radians and decimals	1.23r
Surveyor's Units (N d ' " E)	North/South degrees towards East/West	N12d34'56"E

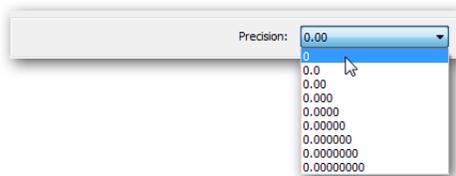
Note that there are 400 grads (used in Germany) and are 2π radians in a 360-degree circle.

If we were *real* land surveyors, we would use Surveyor format. But we're not, and so we're sticking with the more familiar decimal degrees. (Again, you can choose the change the angular format at anytime with the **Units** command without affecting the drawing.)

11. Ensure that the default, **Decimal Degrees**, is selected for the angle of measurement.



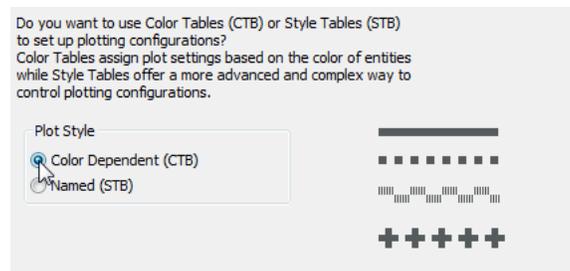
12. Change the default **Precision** to **0** degrees, as we are measuring angles to the nearest degree.



13. Click **Next**.

CHOOSING THE PLOT STYLE

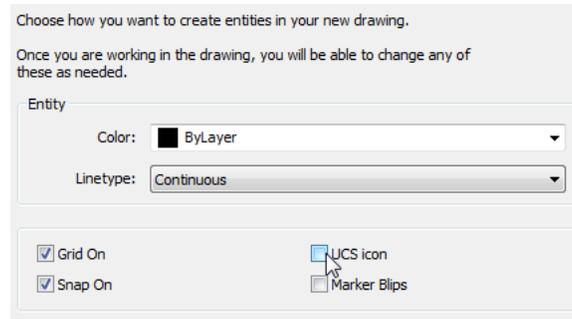
When it comes time to printing drawings, they can be formatted by BricsCAD using one of two methods. One method is known as a *color table* (a.k.a. "color dependent" styles or CTB), while the other method is through a *style table* (a.k.a. "named" styles or STB). They are, unhappily, mutually exclusive, meaning only one can be active at a time — meaning you have to choose one.



- Right now, all you need to know is that neither color tables nor style tables really matter when plotting regular 2D drawings on monochrome printers. And so, ensure that the **Color Dependent (CTB)** option is selected.
- Click **Next**.

SETTING ENTITY PROPERTIES

You come to the final dialog box of the wizard. Here, you set the defaults of some entity properties (a.k.a. “objects”). By default, all entities are drawn with a color named “ByLayer” and a linetype of the same name.



ByLayer means that layers determine the color and linetype of entities. If the layer is set to red, then all entities on that layer appear red; change the layer to green, and the entities follow suit. In this dialog box, you can overrule these default properties. But you won't, because changing colors and linetypes is best left for later and done with the Layers dialog box — i.e. “by layer.”

- The only change to make here is turning off the UCS icon, because it is unhelpful (by being visually intrusive) to the tutorial drawing. See figure above for how settings should look.
- Click **Finish** to close the dialog box.

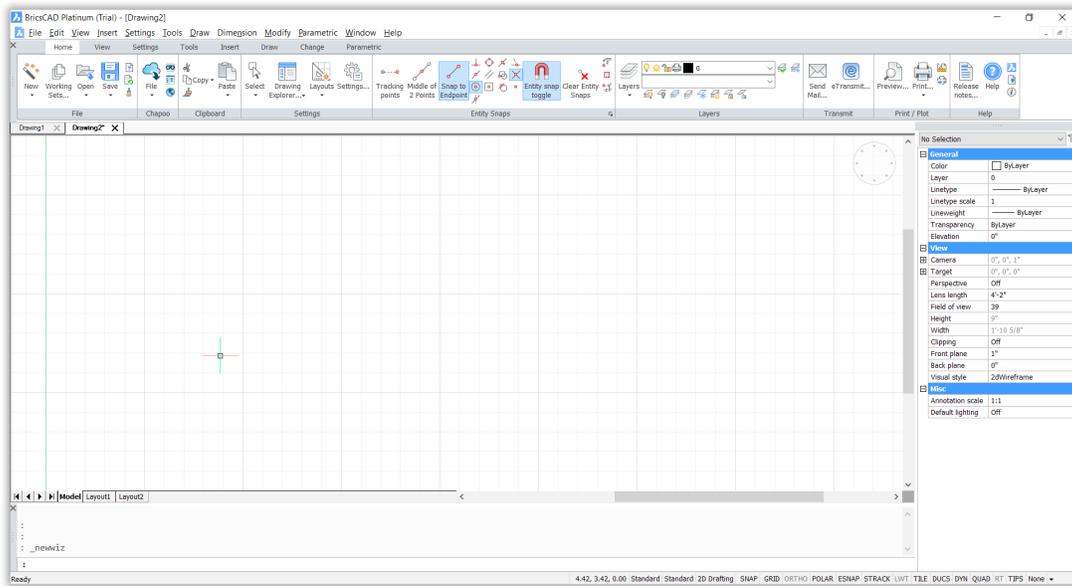
FINISHING THE WIZARD

When the wizard is done, BricsCAD's drawing area looks a little bit different from before. There is an array of lines in the drawing called the “grid” lines. There is one grid line that goes through the origin (at 0,0) that is red and another that is green — the same colors as the x and y crosshair cursor lines:

Red grid line — x axis (horizontal)

Green grid line — y axis (vertical)

The UCS icon is gone, because you turned it off and because it is unnecessary for this tutorial. And when you move the mouse, the coordinate display on the status bar changes in increments of 0.01 — this is due to setting the precision of units to 0.01.



BricsCAD's drawing area at the end of the wizard

The wizard does not adjust all settings, and you have a few more to change, as described next.

Additional Important Settings

The wizard does not, unfortunately, set up everything you need in new drawings, and so for the rest of this lesson you'll use commands to set other things, such as the area of the drawing's limits, the spacing of the snap and grid, and the names of layers. Along the way, I'll explain what each means.

SETTING THE DRAWING LIMITS

Command Limits
Menu Bar Settings | Drawing Limits

There is no limit to the size of drawings you can create with BricsCAD. You could, if you wanted, draw the entire solar system full size. Indeed, an early AutoCAD sample drawing showed that this was possible: from the orbit of Pluto we could zoom all the way down to the individual letters on a plaque mounted on a lunar lander in a crater on the Earth's moon.

When it comes time to printing drawings, however, the overall size of a drawing is constrained by the size of the paper that the printer can handle, often just 210 x 297mm or 8-1/2 x 11" — a far sight smaller than the solar system! One way to control the size is by setting the limits with the **Limits** command.

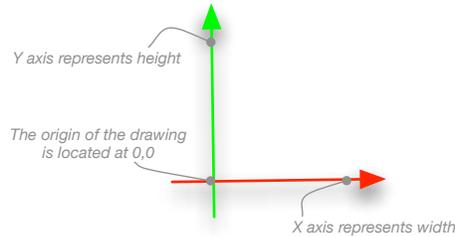
Limits show the nominal area of a drawing, and they perform two other functions that are useful:

- > Constrain the extents of the grid lines
- > Determine the area shown by the Zoom command's All option

More on these later. In the meantime, here is how you determine the limits and then set them:

1. Examine the size of the yard in the sketch. After leaving a bit of “breathing room” around the plan, the drawing will need about 40m (130 feet) of width and 30m (100 feet) of height.
 - > **Width** is represented by the **x** coordinate, which sometimes is shown in red
 - > **Height** by the **y** coordinate, shown in green usually

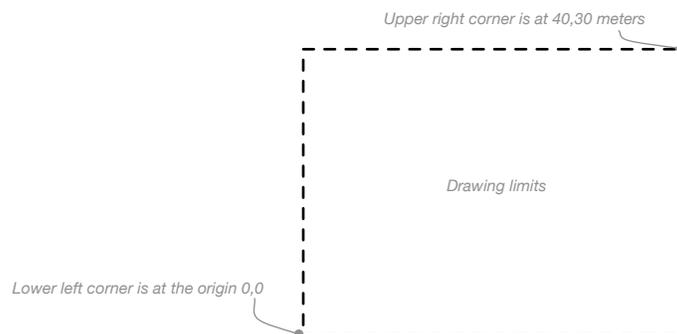
The *origin* of the drawing is where the lines of the x and y axes meet, at 0,0.



2. At the ':' prompt, enter the **Limits** command, as follows:
: **limits** (Press **Enter** to execute the command)
3. Notice that BricsCAD reports that the limits are turned off. Turn them on:
Limits are off: Set lower left corner or [turn limits ON] <0,0>: **on**
4. Restart the Limits command by pressing the spacebar. This is a shortcut to repeating the last-used command:
: (press the **spacebar**)
: LIMITS

Notice that BricsCAD reports the name of the command it is repeating, so you are sure it is the correct one.

5. Now specify the x,y coordinates of the lower left corner. The default value of 0,0 is good for this drawing, so just press **Enter** to leave the value as is:
Limits are on: Set lower left corner or [turn limits ON] <0,0>: (Press **Enter** to accept the default value of 0,0)
6. When BricsCAD asks for coordinates of the upper right corner, enter 40 and 30 (meters), as follows:
Upper right corner <12,9>: **40,30**



Warning! If you work with imperial units, it is important that you include the **apostrophe (')** when you enter measurements as feet, like 130'.

When you leave out the apostrophe of 130, BricsCAD assumes you mean 130 inches, which it translates to 10'-10.

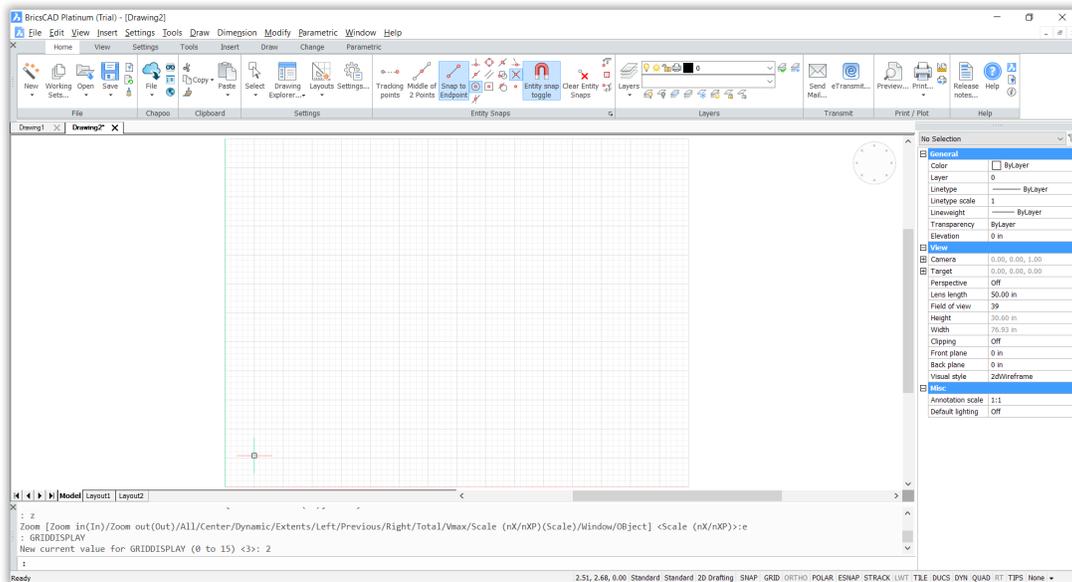
- I find it handy to restrict the grid's lines to the extent of the Limits. To do so, I change the value of the **Grid-Display** system variable to 2 like this:

```
: griddisplay
New current value for GRIDDISPLAY (0 to 15) <3>: 2
```

- Finally, use the **Zoom All** command to see all of the drawing. Enter the **Zoom** command, and then enter 'a' for the **All** option:

```
: zoom
Zoom: In/Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale/Window/<Scale (nX/nXP)>: a
```

The Zoom command lets you see the “big picture,” as well as zooming in for a detailed look.



The drawing area after the grid is restricted to the area of the limits

The grid is now constrained to the area specified by the limits. (NEW IN V17) Grids lines are fainter in BricsCAD V17 so that they do not obscure drawing elements.

When you began this tutorial, BricsCAD displayed an area of 12" by 9"; now BricsCAD displays an area of 40m by 30m (130' by 100'). When you move the cursor to the upper right corner of the drawing area, the status bar should report values near 40, 30 (meters).

ACCESSING AND CHANGING VARIABLES

Command	Settings
Ribbon	Settings Settings Settings
Menu Bar	Settings Settings
Alias	SE

Some of the values you entered at the command prompt can also be changed through the Settings dialog box. This is a special dialog box that reports on the values of *variables*. “Variables” hold settings that control and report on nearly all aspects of BricsCAD — and there are over 800 of them! I list them all in Appendix B. Variables are also known as “system variables,” or “sysvars” for short.

For example, GridDisplay is a variable. You used it to tell BricsCAD how to display the grid. This specific variable has four settings, each represented by a number (technically, an integer).

Here is what the GridDisplay variable controls:

GridDisplay	Meaning
1 (default)	Display beyond the area of the limits
2 (default)	Display adaptive grid
4	Allow subdivisions below grid spacing
8	Follow dynamic UCS

The numbers can be added together to turn on more than one option. The default value is 3, which means that (1) the grid is displayed beyond the limits and (2) the adaptive grid is displayed. In the tutorial, you changed the value to 2 so that the grid no longer displayed beyond the limits.

(*Adaptive grid* means that as you zoom out, the grid shows fewer lines so that the drawing area is not overwhelmed with too many closely-spaced lines. *Allow subdivisions* means that additional lines are displayed between the specified grid spacing. *Follow DUCS* means that the grid rotates to match the x,y-plane specified by the current UCS.)

Here is another example: When you used the Limits command, for example, BricsCAD stored the values you entered in two system variables, LimMin and LimMax:

LimMin — stores the x, y coordinates of the lower left corner.

LimMax — stores coordinates of the upper right corner.

These two system variables can be accessed directly at the command bar by entering their names, just like they were commands:

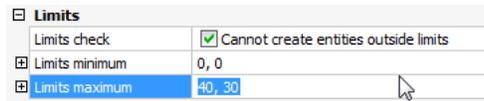
```
: limmin  
New value for LIMMIN <0,0>: (Press Enter to keep values as they are)  
  
: limmax  
New value for LIMMAX <40,30>: (Press Enter)
```

A problem lies in remembering the names of hundreds of system variables. And so the other method is to use the Settings dialog box, as follows:

1. Enter the **Settings** command:
: **settings**

Notice the Settings dialog box.
2. This dialog box contains hundreds of settings, and so the best way to access one of them is to use the **Find** feature, as follows:
 - a. Enter “limits” in the **Find** field.
 - b. Press **Enter**.

The Settings dialog box goes to the first setting with “limits” in its name. Click the  **Find Next** arrow until you arrive at the **Limits Minimum** and **Limits Maximum** settings:



3. There is no “Close” or “OK” button to dismiss the dialog box. Instead, when you are done just click the red X.

TIP When the **Find** field turns orange, it means that there is no setting by that name. This can occur when you misspell the name.

CHANGING THE SNAP AND GRID SPACING

One great advantage to drawing with CAD is that software permits you to create *very* accurate drawings. BricsCAD has several functions that help you draw with pretty much perfect accuracy.

One such feature you already met, the grid. It helps you visualize distances.

Another such function is known as “snap mode.” Snap can be thought of as setting the *mouse resolution*. It determines the distance the cursor moves in the drawing. For instance, by setting the snap distance to 0.5, the cursor moves at precise, 0.5-unit distances during drawing and editing commands. (You learn about other accuracy aids in later lessons of this book.)

During the wizard, you turned on snap mode, but the spacing of 0.5 meters is too wide. Follow these steps to change the snap and grid spacing to 0.1 meters:

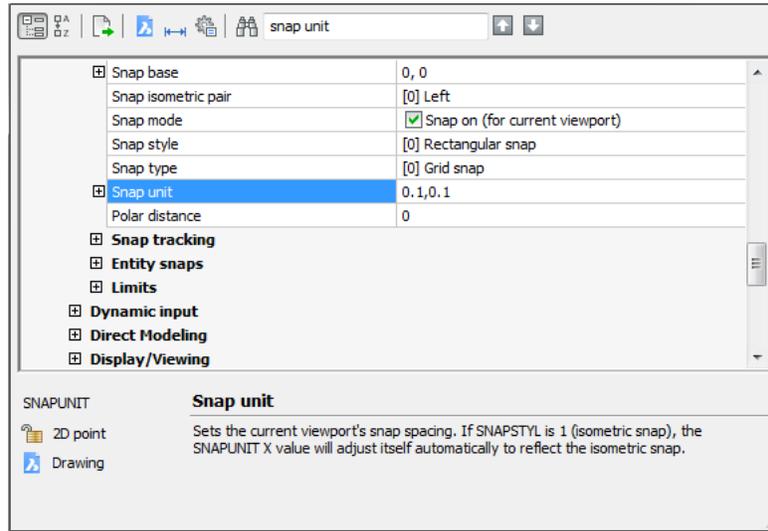
1. (If you are not in the Settings dialog box, then enter the **Settings** command.)
In the Find field, enter “snap mode” and then press **Enter**.



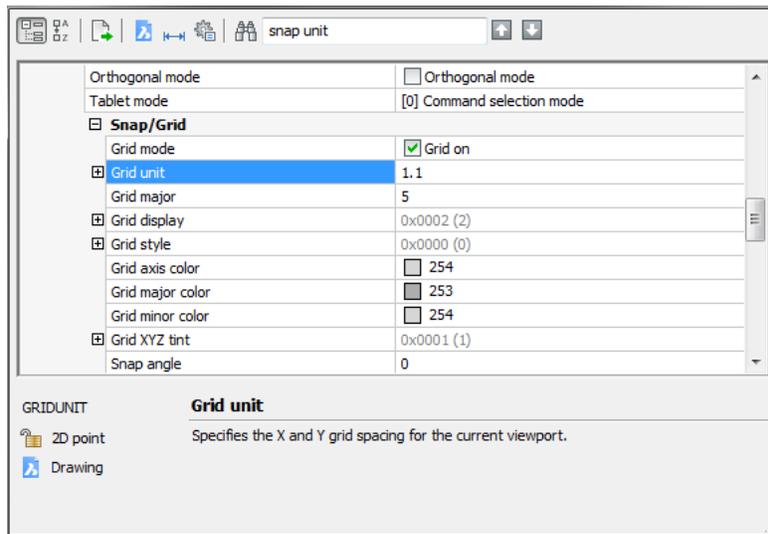
Annotations for the Settings dialog box:

- Enter text to search for (points to the search field)
- Click arrows to find next occurrence (points to the Find Next arrow)
- Click X to close dialog box and save changes (points to the close button)
- Click + to expand section (points to the plus icon next to Snap unit)
- Click - to collapse section (points to the minus icon next to Snap unit)
- Descriptive name of system variable (points to 'Limits maximum')
- Actual name of system variable (points to 'LIMMAX')
- Description of system variable (points to the description of the LIMMAX variable)
- Value of system variable (points to the value '40, 30')

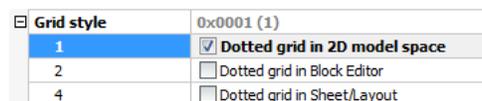
2. Notice that the listing in the dialog box jumps to the **Snap Unit** field. It specifies the distance that the cursor moves.
3. Change the X and Y values of **Snap Unit** to **0.1, 0.1** to represent 1 decimeter (or **1", 1"** in imperial units). Leave other snap settings at their default values, such as Snap Angle = 0.



4. Earlier, the wizard had turned on the grid as a visual guide. The grid is meant to guide you; the default spacing of 10 units is perhaps to far apart. (BricsCAD will in fact not display the grid when its lines or dots are too closely spaced.) Change the spacing to 1 meter by changing the value of **Grid Unit** to **1,1**.



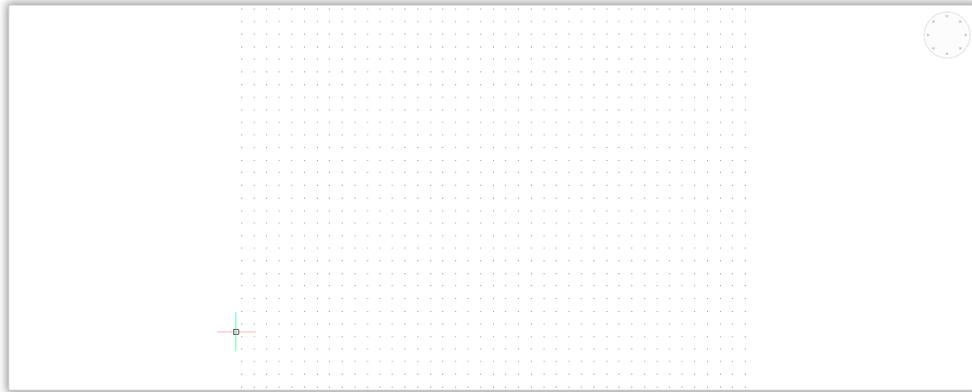
5. Finally, change the type of grid from lines to dots. Grid lines are useful in 3D modeling, but tend to cover up lines in 2D drafting. Here is how to do this:
 - a. Click the **+** node next to **Grid Style**.
 - b. Turn on the **Dotted Grid is 2D Model Space** option.



6. To close the dialog box, click the **X** at the top right end.

TIP A grid distance of 0" has a special meaning in BricsCAD. It means that the grid spacing matches the snap spacing, 1" in our case.

Notice that the drawing area is covered by a grid of dots.



TIPS Sometimes the snap function can get in the way of your drafting. You can turn the snap on and off at any time by pressing function key **F9**.

Similarly, the grid can be toggled with **F7**.

For many options in BricsCAD's commands, you need only type the first letter of the option. For example, type "a" as the abbreviation for the **All** option. When two options begin with the same letter, you need to type the first two characters of the option.

When entering text and numbers in a dialog box, pressing the **Tab** key is a quicker way to get to the next field. *Fields* are buttons, text entry boxes, list boxes, and other dialog box elements that you can change. To return to previous fields, press **Shift+Tab** (hold down the **Shift** key, and then press **Tab**).

When you change values in the Settings dialog box, they are shown in boldface text. And, the changes take effect immediately.

Creating Layers

Commands	Layer, layerpanelon
Ribbon	Home Layers Layers
Menu Bar	Settings Layers
Alias	LA

If you ever worked with overlay drafting, then you would be familiar with the concept of layers. In overlay drafting, drafters draw the base plan on one clear sheet of Mylar, a common form of drawing media made from plastic. The electrical plan would be drawn on another sheet, the structural on a third. Since the Mylar is transparent, drafters would overlay the two or all three of the drawings to create a single blueprint —depending on how much detail needed to be shown.

In CAD, *layers* operate in a similar manner. We draw different parts of drawings on different layers. The base on one layer, electrical on another, and so on. Then we turn layers off and on to display the drawing in different ways.

For example, the electrical contractor would be interested in seeing only the base plan layer with the electrical layer. While it is possible to create hundreds and thousands of layers in drawings, it is more common to work with a few dozen layers; in this book, we work with a mere half-dozen.

QUICK SUMMARY OF LAYERS

BricsCAD provides great control over layers through the Drawing Explorer. It lists the names of all layers and their properties. The names of properties are indicated by the names on the header bar.

Current	Layer Name	Color	Plot	Description	On/Off	Freeze	Locked	Lineweight	Plot Style	Linetype	Transparency	New VP	Material
1	0	White						Default	Color 7	Continuous	0		Global
2	Defpoints	White						Default	Color 7	Continuous	0		Global
3	House	White						Default	Color 7	Continuous	0		Global
4	Lawn	Cyan						Default	Color 4	Continuous	0		Global
5	Lot	8						Default	Color 8	Continuous	0		Global
6	Plants	Green						Default	Color 3	Continuous	0		Global
7	Pond	Blue						Default	Color 5	Continuous	0		Global
8	Road	Red						Default	Color 1	Continuous	0		Global

(To sort layers alphabetically, click a header such as Name or Linetype. Click a second time to sort in reverse order, Z to A.)

- — indicates the current layer.

Layer Name — specifies the name up to 255 characters long. You can use numbers, letters, and the following punctuation marks: dollar sign (\$), hyphen (-), underline (_), and spaces. Two layers cannot have the same name in the same drawing. There is no limit to the number of layers in a drawing.

On/Off (*obsolete*) — when on, displays entities; when off, entities cannot be seen nor are they plotted.

Freeze — freezes entities so that they cannot be seen, edited, or plotted. Also, frozen layers are not included when BricsCAD performs drawing regenerations and hidden-line operations. When thawed, the layers behave normally. (To thaw layers means to turn off their frozen status.) It is better to freeze layers than turn them off.

Locked — locks layers, which means entities are seen but cannot be edited. Unlock layers to make entities available for editing.

Color — specifies the color of entities on the layer. The default color is 7, which is displayed as white or black, depending on the background color. Entities drawn on the layer are displayed in this color, but the entity color can be overridden with the Color command.

Linetype — specifies the line type by which entities are displayed. The default linetype is Continuous (solid line). Before any other linetype can be used, its definition must be loaded into the drawing; can be overridden by the Linetype command.

Lineweight — specifies the line widths for entities on the layer. The default is 0.00mm; the maximum is 2.11 mm (about 0.08", or 6 points wide); can be overridden by the Lineweight command.

Plot Style — specifies the plot style with which entities plotted. Named plot styles defines colors, widths, and percentages of black during plotting. This option is not available for drawings created with color-dependent plot styles.

Plot — specifies whether the layer is plotted; when off, the layer is not plotted.

Material — specifies the material used during renderings.

LAYOUT MODE PROPERTIES

When drawings are in layout mode, this dialog box displays more columns. (VP is short for "viewport.")

VP Freeze — freezes the layers in new viewports created in paper space

VP Color, VP Linetype, VP Lineweight, VP Plot Style — specifies the color, linetype, lineweight, and plot style for the layer in paper space viewports

The layers in this tutorial segregate the text, the yard, the road, and so on.

A further advantage to CAD layers is that global properties can be applied to entities. For example, all entities assigned to a layer called “grass” could be colored green — or maybe brown, depending on the time of year! Changing the color of the layer instantly changes the color of all entities assigned to that layer.

Assigning entities to layers is this simple: (a) set the name of a layer as current, and then (b) start drawing! The most common way to set a name as current through the Layer droplist. There are commands that let you move entities to other layers, should that be necessary.

NAMING LAYERS

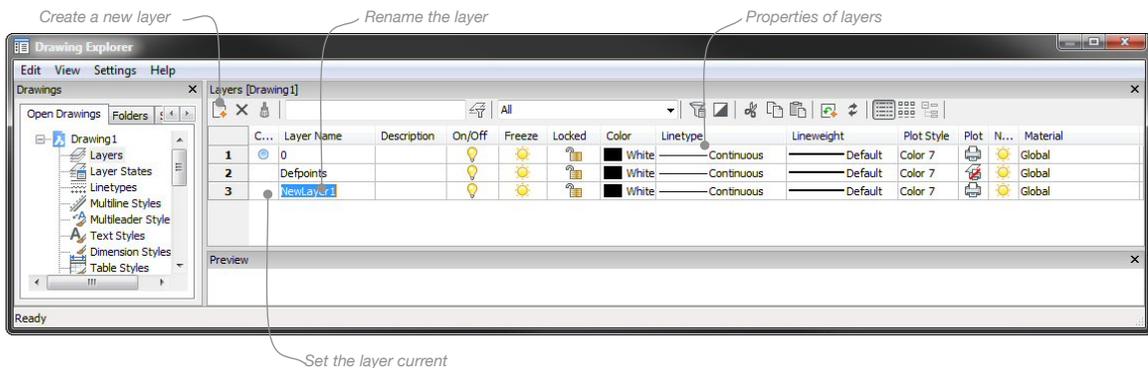
Layers are specified by name. BricsCAD lets you give layers names up to 255 characters long. Start setting up new layers with the **Layer** command, which displays a dialog box. (NEW IN V17) Layers can also be displayed in a panel, which can always stay open, with the **LayersPanelOpen** command.

1. Enter the **Layer** command:

: layer

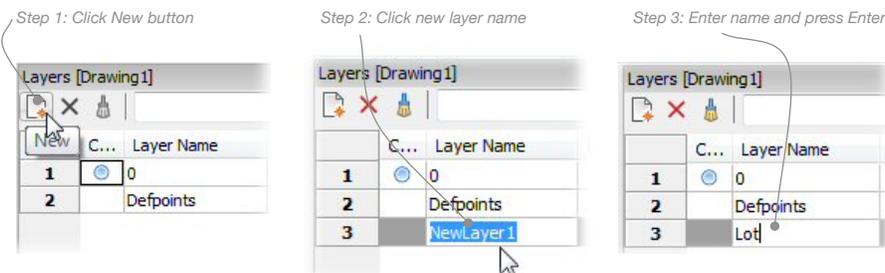
Notice the Drawing Explorer dialog box. This dialog box lets you control almost every aspect of layers in drawings.

The drawing already has one layer: 0. Every new BricsCAD drawing has one layer, called “0”, which you can never erase. Layer 0 has special properties that affect the creation of blocks, as discussed in greater detail later in this book.



2. Following the steps illustrated below, create a layer:

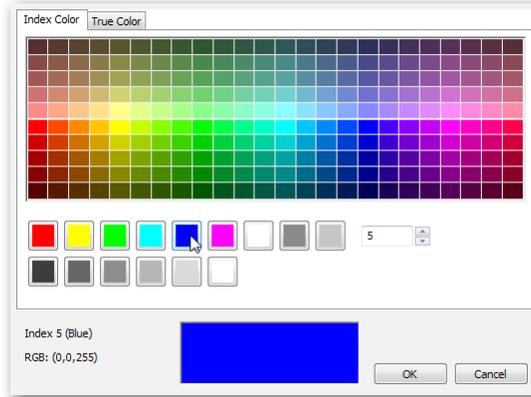
- a. Click the **New Layer** button. Notice that BricsCAD creates a new layer called “NewLayer1.”
- b. Change the name by clicking “NewLayer1.”
- c. Type **Lot**, and then press **Enter**.



3. Assigning a color to each layer makes it easier to determine which lines belong to which layers. Change the color of the Lot layer to blue, as follows:
 - a. Click the black square under the **Color** column across from the layer name **Lot**.



- b. The Select Color dialog box appears, displaying 255 colors. Which one to choose?



Above the set of black and gray squares are BricsCAD’s “standard colors,” which are the most-commonly used ones. Select the dark blue square. Notice that the number “5” appears in the Color text box, because this is color number 5 according to the color-numbering system used by many CAD systems.

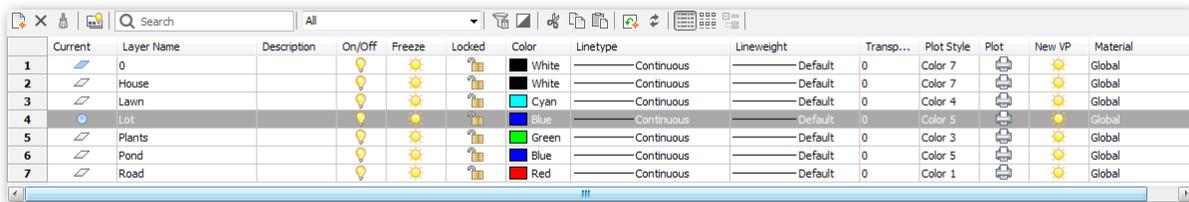
- c. Click **OK** to exit the Select Color dialog box. The color of the square across from layer Lot changes to blue.
4. Add the remaining layer names and colors, using the table as a guide. If you make a spelling mistake, just click the layer name and type the correction.

New Layer	Layer Color
Lot	Blue (color 5)
House	White (black)
Road	Red (color 1)
Lawn	Cyan (light blue, color 4)
Plants	Green (color 3)
Pond	Blue (color 5)

5. When you finish assigning colors to layer names, pick the **Lot** layer name, then click in the blank square between the 2 and Lot. Notice the **blue** ball that appears; it indicates the *current* layer.



For now, drafting takes place on the **Lot** layer — until you select another name as the current layer.



6. To exit the Drawing Explorer dialog box, click **X** at the top right of the dialog box.

Notice that the name of the layer on the Entity Properties toolbar changes from **0** to **Lot**. The color has changed from black to blue. When you click the layers droplist, you see all of the newly-created layer names and their properties. Later in this book, you make use of this droplist to control layer properties.



TIPS There can be some confusion over the color “white” — or is it black? BricsCAD switches white and black depending on the background color of the drawing area. When the background is black, BricsCAD displays white lines; when white, BricsCAD displays black lines. So, white can be black — at least in the world of CAD.

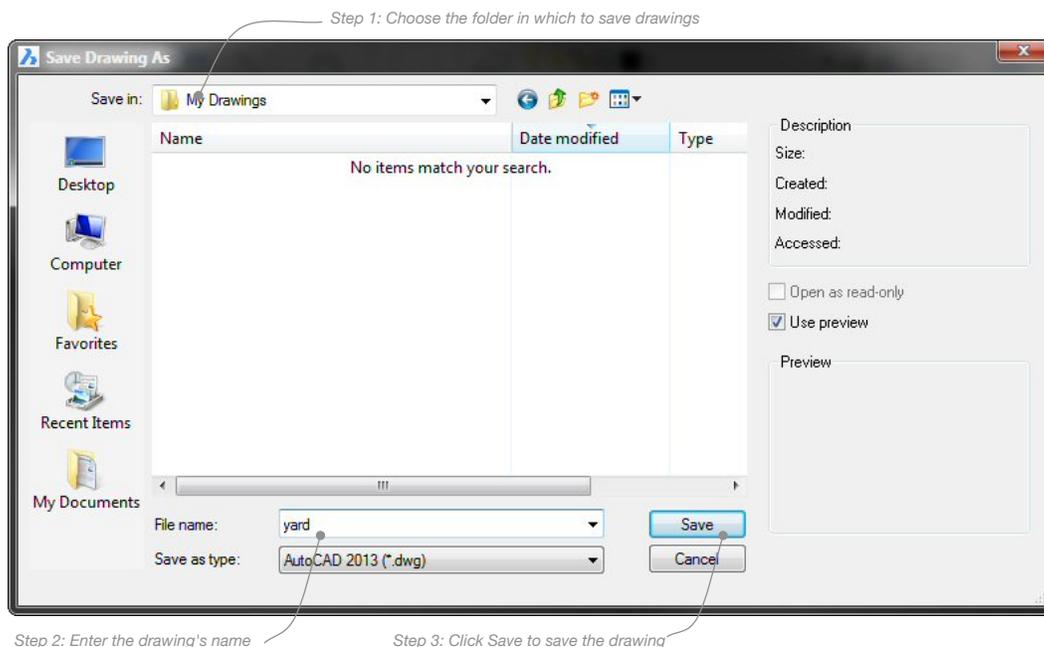
To set the background color in BricsCAD, from the Settings menu, select Settings. In the Find field, enter background color, and then press Enter. Select the color.

Saving Drawings

Command	Save
Ribbon	Home File Save
Menu Bar	File Save
Shortcut	Ctrl+S (Cmd+S on Macs)

As the final activity, save your important work with the **Save** command, as follows.

1. On the toolbar, click on the icon that looks like a diskette  (tooltip = Save). In the ribbon’s Home tab, click the Save button in the File panel.
2. Because this drawing has a generic name of “Drawing1.Dwg,” BricsCAD displays the Save Drawing As dialog box to give you a chance to give it a better name. Type the name “Yard” in the **File name** text entry box.



3. Click the **Save** button. BricsCAD saves the drawing with the name *yard.dwg*. From now on, you and BricsCAD refer to this drawing as “Yard.”

MAKING BACKUPS AUTOMATIC

For speed, BricsCAD keeps parts of the drawing in the computer's memory (RAM). The drawback is that when the operating system crashes or when the power is cut to the computer, you may lose some or all of your work. Because crashes are common in older versions of Windows especially, it is an excellent idea to save drawings every 10 or 15 minutes.

BricsCAD does this automatically, but also lets you specify the interval in which the drawings are saved — without you needing to repeatedly use the Save command. The default setting is **60** minutes, which is too long.

QUICK SUMMARY OF THE SAVEAS COMMAND

The Save As dialog box lets you save drawings in several dialects of DWG. Newer versions of BricsCAD can always read drawings created by older versions of BricsCAD, Ares, DraftSight, AutoCAD, AutoCAD LT, IntelliCAD, and other DWG-based drafting programs.

Older versions of BricsCAD cannot, however, read drawing files created by newer versions, and this is true of all other CAD programs, including AutoCAD. For example, BricsCAD V8 cannot read drawings created in BricsCAD V17. When working with users of older versions of CAD programs, you must make BricsCAD explicitly save drawings in earlier formats compatible with older software, as described by this table:

<u>Save As File Format</u>	<u>Saves Drawings in File Formats Used By These Versions</u>				
AutoCAD 2013	2013	2014	2015	2016	2017
AutoCAD 2010	2010	2011	2012		
AutoCAD 2007	2007	2008	2009		
AutoCAD 2004	2004	2005	2006		
AutoCAD 2000	2000	2000i	2002		
AutoCAD Release 14	Release 14 (released in 1997)				
AutoCAD Release 13	Release 13 (released in 1994)				
AutoCAD Release 11/12	Releases 11 and 12 (released in 1990 and 1992, respectively)				

When you regularly save drawings in older formats, then you can make one of them the default. In the Settings dialog box, search for “save format,” and then choose the file format you wish to make the default.

Should you need to go all the way back to AutoCAD 10 through 2.5, then use the DXF format, such as “DXF Release 10.”:

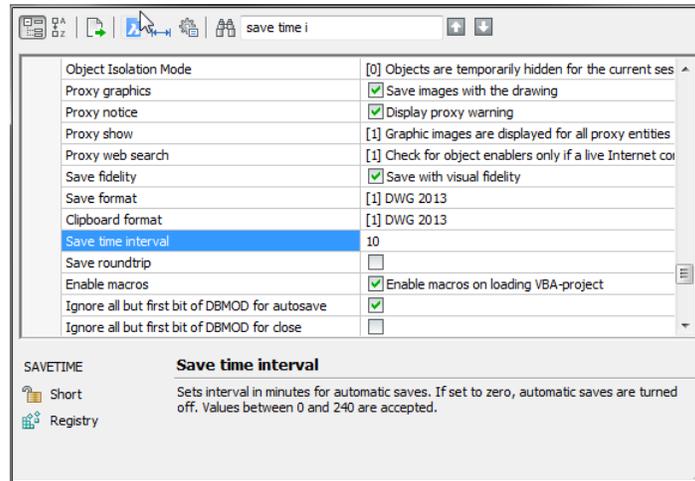
AutoCAD Release 10	R10 (released in 1988)
AutoCAD Release 9	R9 (released in 1987)
AutoCAD Release 2.6	R2.6 (released in 1987)
AutoCAD Release 2.5	R2.5 (released in 1986)

DXF is short for “drawing interchange format,” and is a (mostly) open file format read by many CAD and other programs. BricsCAD imports and exports drawings in DXF format.

Warning! BricsCAD may erase and alter some entities when it translates drawings to earlier formats of DWG and DXF. Check drawings after translation. This is not the fault of BricsCAD but due to the way that entities are represented in drawing files.

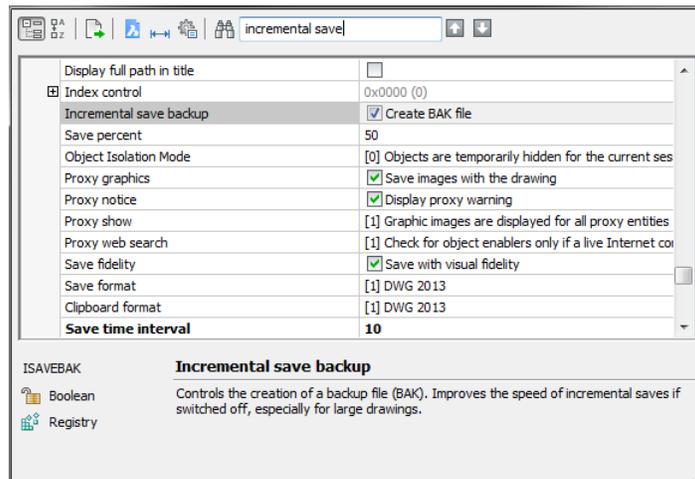
To ensure that automatic backups are enabled, and to make automatic saves more frequently, follow these steps:

1. Enter the **Settings** command.
2. In the **Find** field, enter “save time,” and then press **Enter**.
3. Change the value from **60** to **10**.



TIP The **Save Time Interval** setting specifies how much time elapses before BricsCAD saves drawings. Don't set this number too low, otherwise the computer spends so much time saving to disk that other work cannot be done.

4. If it isn't already, you might want to turn on **Incremental Save Backup**. This means that BricsCAD keeps the previous saved file as a backup copy — and that's a good thing. Automatic saves and backups are not made to the original files. Instead, during automatic saves, BricsCAD saves drawings with the extension of **.sv\$** and backed up files with **.bak**.



5. Click **X** to close the dialog box.
6. If you need to take a break at this point, use **File | Exit**. BricsCAD closes its window and you find yourself back at the desktop.

Although BricsCAD automatically saves drawings, it is still a good idea for you to save your work after finishing a significant amount of editing.

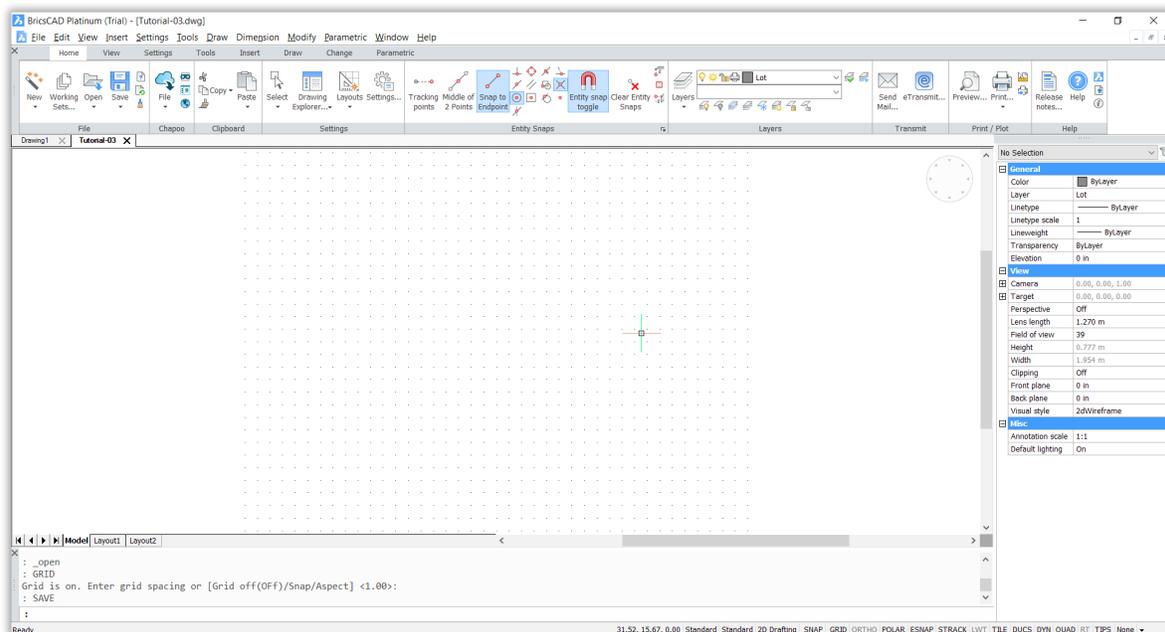
QUICK SUMMARY OF ALTERNATIVE SAVE FORMATS

In addition to saving drawings in AutoCAD's .dwg file format, BricsCAD can export drawings with the **File | Export** command in the following formats:

- ▶ ASCII and binary DXF (*.dxf)
- ▶ Template (*.dwt)
- ▶ Adobe Portable Document Format (*.pdf)
- ▶ Scalable Vector Graphics (*.svg)
- ▶ Windows Bitmap (*.bmp), Metaformat (*.wmf), and Enhanced Metaformat (*.emf)
- ▶ And many versions of DWF, such as 2D, 3D, binary, ASCII, and compressed ASCII

Summary

Let's review the drawing to this point. Although you haven't drawn anything yet, the drawing file contains a fair amount of information.



- ▶ On the toolbar, you see the color of the current layer is blue and its name is **Lot**.
- ▶ On the status line, you see that the coordinates are displaying in decimal units, or meters; and that drafting modes, like snap, grid, and model, are turned on.
- ▶ The grid is made of dots.
- ▶ The automatic backup feature saves your drawing six times an hour.

In the next lesson, you begin to draw!

Creating Your First Drawing

The point to 2D CAD is to draft drawings efficiently, and then print them — either on paper or electronically. Here you learn how to draw lines, accurately, how to make changes to them, and then produce a copy of your first drawing on a printer.

IN THIS CHAPTER

- Drawing with lines and polylines
- Understanding absolute and relative distances
- Using polar coordinates
- Modifying entities
- Plotting (printing) drawings

KEY TERMS IN THIS CHAPTER

- Absolute coordinate** — refers to measurements made relative to the drawing's origin
- Aperture** — refers to the area around the cursor in which BricsCAD searches for entities to snap to
- Direct distance entry** — specifies points by moving the mouse in a direction, then entering the distance
- Extent** — refers to the invisible rectangle that encompasses all entities in drawings
- Fillet** — rounds corners
- Mirror** — mirrors copies of entities
- Entity (entity) snap** — snaps the cursor to geometric features, such as to the ends of lines
- Origin** — refers to the location of $x=0, y=0$, usually at the lower-left corner of drawings
- Ortho** — constrains cursor movement to the vertical and horizontal; short for “orthographic”
- Pick cursor** — refers to the square cursor in which BricsCAD searches for entities to select
- Polar coordinate** — describes measurements specified by distances and angles
- Relative coordinate** — describes measurements made relative to the last point

USEFUL ABBREVIATIONS

- @** Specifies relative coordinates, such as @2,3
- #** Specifies absolute coordinates, such as #4,5
- <** Specifies angles, such as 10<45
- Forces BricsCAD to use the command-line version of a command, such as -layer
- [option]** Indicate command options, such as [Undo]
- <value>** Indicate the default (current) value, such as <LOT>
- x** Specifies the X coordinate along the horizontal axis
- y** Specifies the Y coordinate along the vertical axis

NEW COMMANDS

Command	Aliases	Menu Selection	Ribbon Tab
Cancel	Esc	... ¹	...
Fillet	f or fi	Modify Fillet	Change Modify Fillet
Mirror	mi	Modify 2D Mirror	Change Arrange 2D Mirror
Move	m	Modify Move	Change Arrange Move
Open	op or Ctrl+O	File Open	Home File Open
OSnap	os or F3	Settings Settings	Settings Settings
PLine	pl	Draw Polyline	Draw Draw Polyline
Plot	Ctrl+P	File Print	Home Print/Plot Print
QSave	Ctrl+S	File Save	Home File Save
Zoom	z	View Zoom	View Zoom Zoom

¹ The ellipsis (...) indicates no menu selection is available.

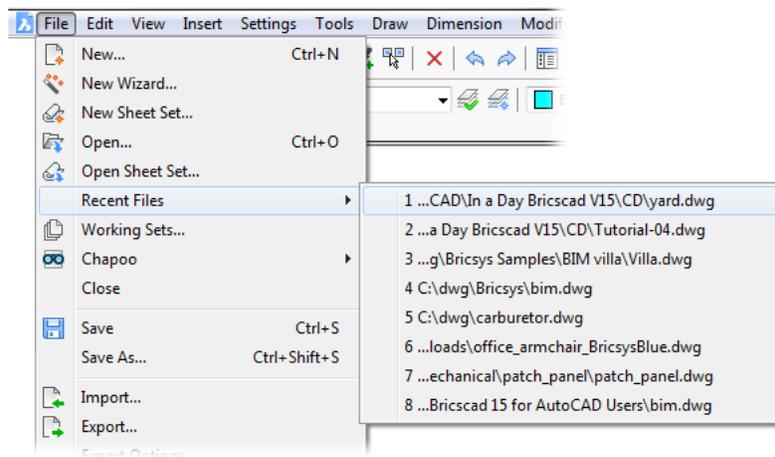
Reopening Drawings

Command	Open
Ribbon	Home File Open
Menu Bar	File Open
Shortcut	Ctrl+O

If you exited BricsCAD at the end of the last lesson, then you need to restart BricsCAD, and then load the Yard drawing.

TIP The Recent Drawings item of the File menu always holds the names of the last ten drawings you opened with BricsCAD. Here's how to do this:

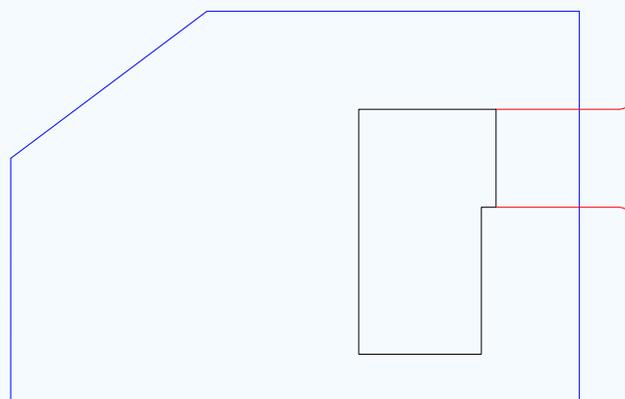
1. Click **File** on the menu bar.
2. Slide your cursor down to **Recent Drawings**, and then click on **yard.dwg**. See figure below.



Notice that when the *Yard* drawing opens in BricsCAD, it looks exactly the same as when you last saw it — that is to say blank, except for the grid lines.

THIS CHAPTER'S DRAWING

By the end of this lesson, your drawing should look like this one:



Tutorial-04 drawing file

Drawing the Lot Boundary

Command	Line
Ribbon	Draw Draw Line
Menu Bar	Draw Line
Alias	L

Let's get some lines on the screen! To orient yourself, the first thing to do is to draw the boundary of the yard. That helps you see the extents of the drawing.

The lines making up the lot boundary are drawn with the **Line** command. You begin drawing the lot lines at the lower-left corner, the origin (0,0), then work counterclockwise around the lot boundary, as shown by the illustration below.

1. First, ensure **DYN** is turned off on the status bar; the word should look gray. This turns off dynamic input, so that all prompts appear in the Command bar.



2. Enter the **Line** command, as follows:

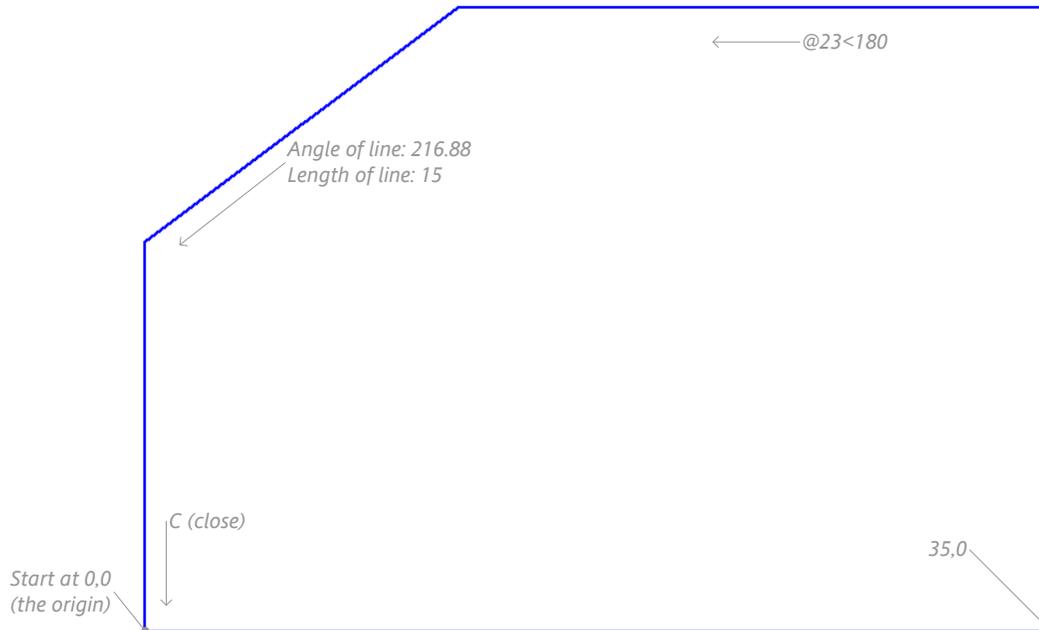
: **line**

3. Respond to the 'Specify first point:' prompt by typing the coordinates of the origin:

: **_line Start of line: 0,0**

4. To draw the lower boundary line 35m long, you need to tell BricsCAD that the far end of the line is located at the x,y coordinates of **35,0**. (In imperial units, this is **116',0**.)

Angle/Length/Undo/<End point>: **35,0**



TIP Notice that the prompt line **Angle/Length/Undo/** is separated by slashes. The slashes indicate that these words are *options* of the Line command. If you were now to type “undo”...

Angle/Length/Undo/<End point>: **undo**

...BricsCAD would un-draw the last line segment. “Angle” and “Length” are other options that specify the angle and length of line segments. Later, you will encounter another option, “Close.”

5. The next line is 24m north. Its endpoint is located at coordinates 35,24.
(In imperial units, draw to 116',80'. Remember to include the apostrophe ' with each distance; the apostrophe indicates feet; if you were to leave it out, BricsCAD would interpret the numbers as inches, and you would end up with a very small yard!)

Angle/Length/Follow/Undo/<End point>: **35,24**

6. You drew the first two lines with *absolute coordinates*, where you calculated the coordinates based on measurements relative to the origin at 0,0. BricsCAD, however, can do these calculations for you when you use *polar coordinates*, in which you specify the distance and angle. Continue drawing the lot boundary by combining relative and polar coordinates like this:

Angle/Length/Follow/Close/Undo/<End point>: **@23<180**

(In imperial units: End point: **@76'<180**)

With all that punctuation, entering **@23<180** can feel like quite a typing chore. I'm getting you to type coordinates for now so that later you will appreciate other, less cumbersome input methods!

TIPS When you tell BricsCAD to draw a line with the above relative polar coordinates, you enter a special notation that has the following meaning:

Notation	Meaning
@	Use relative coordinates
23	Distance is 23m from the current point
<	Draw the line at an angle...
180	...of 180 degrees

Lines are drawn relative to the current point; the angle, however, is measured in absolute degrees using the East-is-0-degrees convention. Using relative polar coordinates makes sense when you have many angled lines to draw.

In the meantime, if you make a mistake entering the coordinate notation, simply use the Undo option (type **U** at the prompt) and reenter the coordinates.

7. Enter the coordinates for the next endpoint using the **Angle** option, as follows:

Angle/Length/Follow/Close/Undo/<End point>: **a**
Angle of line: **216.88**
Length of line: **15**

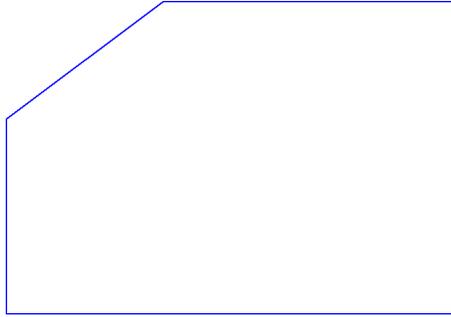
(In imperial units, Length of line: **50'**.) Notice that the **Angle** option removes the need to use the **@** and **<** characters.

The **Length** option works the same way, but prompts first for the length, then the angle.

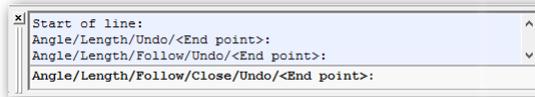
8. To finish the lot boundary, you use a shortcut. Type **c** (short for “Close”) to close the boundary — instead of typing the final coordinates (0,0):

Angle/Length/Follow/Close/Undo/<End point>: **c**

BricsCAD automatically draws a line from the current endpoint to the beginning of the first line.



As I mentioned earlier, slashes separate options names. All of the options for the Line command are **Angle/Length/Follow/Close/Undo/** :



Angle — asks you to enter the angle of the next line segment, followed by length

Length — asks you to enter the length of the next line segment, followed by angle

Follow — asked you to show the angle with the mouse, and then enter the length

Close — BricsCAD closes the polygon formed by the lines

Undo — BricsCAD undoes the drawing of the last line segment

9. To add space around the drawing, use the **Zoom** command to make the drawing 10% smaller, as follows:

```
: zoom  
In/Out/All/Center/Extents/Left/Previous/Right/Window/<Scale (nX/nXP)>: 0.9x
```

This zooms the drawing 90% as large as it was before.

Instead of drafting on paper, you have created your first digital drawing! More importantly, you have drawn it full size — even if it looks small on your computer screen. This is one of the most powerful aspects of CAD: everything is drawn full size. There is no need for a scale ruler or to divide distances by a scale factor as in manual drafting.

Planning the Next Steps

The next step is to draw the outline of the house. The lower-right corner of the house is located 3m up and 6m in from the lot corner.

There are several ways in BricsCAD to locate one entity relative to another, in this case the corner of the house relative to the property line. You could...

- > Calculate the coordinates of the house's corner
- > Use the **XLine** command to draw a pair of construction lines, then start drawing from their intersection
- > Use the **From** option to start drawing from an offset from the lot corner
- > Use *tracking* to offset the starting point (described in a later lesson)
- > Draw the house in the corner of the lot, then move the house into position

Please don't feel overwhelmed by this list of many options. I showed them to you purely to make the point that in CAD there are many ways to skin a cat.

In this tutorial, you use the last method in the list: draw, and then move. It demonstrates two of BricsCAD's most powerful commands, **PLine** and **Move**. But first, make a layer change.

CHANGING LAYERS

Command **-Layer**

Before you draw the house, you need to change the layer to "House." Continuing with the theme of many approaches to solving drafting problems, a significant number of BricsCAD's commands can be carried out in more than one way. Here you learn an alternative way to control layers.

In the previous chapter, you used the Drawing Explorer dialog box to create and color new layers and to set "Lot" as the current (or working) layer. Another method is to change the layer through a droplist on the ribbon, status bar, toolbar, or use the Layers panel. And then there is another method:

In this part of the tutorial, you change layers by typing the command and its options at the keyboard by entering the **-Layer** command. The hyphen (-) in front of Layer forces the command to display its prompts in the command bar.

1. To set the House layer with the **-Layer** command, enter the following:
: **-layer**
2. The **-Layer** command has more than a dozen options, most of which you ignore for now:
? to list/New/Make/Set/Color/Ltype/LWeight/TRansparency/MATerial/Plot/stAte/ON/OFF/Freeze/Thaw/LOck/Unlock: **s**

Enter **s**, and BricsCAD invokes the Set option. It sets another layer as active, whose name you specify.

3. BricsCAD prompts you to enter the name of the layer to make active. To change the working layer to **House**, type "house" and then press **Enter**.

New layer to make current: **house** (Press **Enter**)

4. The *House* layer is now current.
The **-Layer** command repeats its many-optioned prompt. Press **Esc** to cancel the command and return to the ':' prompt.
? to list/New/Make/Set/.../Unlock: (Press **Esc**)

TIP You can cancel commands at any time by pressing **Esc**. Sometimes, however, you may need to press **Esc** two or even three times in commands that have large numbers of sub-options, such as the **PEdit** command.

5. Look at the ribbon, status bar, or toolbar to confirm that BricsCAD has changed the working layer from layer Lot to layer House.



TIP For all of command options, BricsCAD capitalizes one (or more) letters of each option. To select an option, you need type only its capitalized character, such as **S** for the **Set** option.

When two (or more) options begin with the same first letter — such as **LType** and **LWeight** — then you need to enter the first two characters, as specified by the capitalization — **LT** and **LW** in this case.

There are a few times when the character that’s capitalized is in the middle of the option name, such as **A** for the **stAte** option. This is because the all the other letters in stAte were already taken up by other option abbreviations.

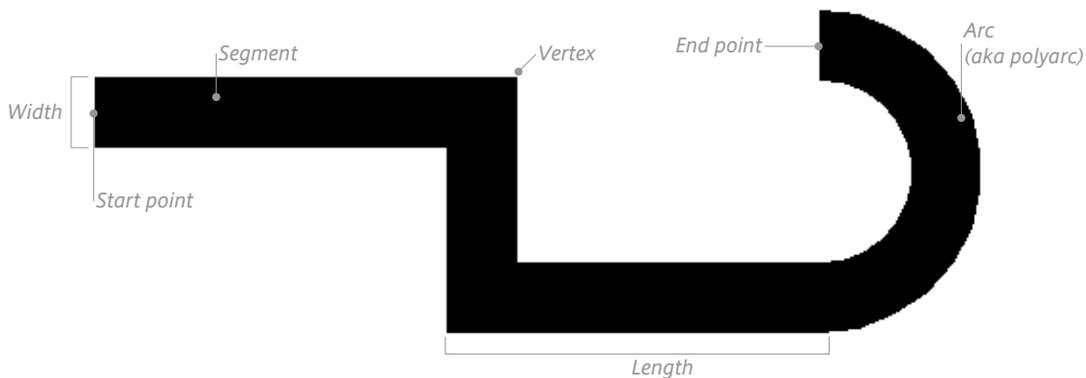
By capitalizing them, BricsCAD shows you the fewest characters you need to enter for it to know what you’re talking about. So you could enter **SE** for the **Set** option or **LTY** for **LType**.

Drawing the House Outline

Status Bar	ORTHO
Ribbon	Settings Settings Draw Orthogonal
Menu	Settings Draw Orthogonal
Shortcuts	Ctrl+L F8
Command	Pline
Ribbon	Draw Draw Polyline
Menu Bar	Draw Polyline
Alias	PL

Earlier, you drew the lot boundary as a polygon with the Line command. It looked like a continuous line that was made of several segments and *vertices* (corners). Each segment, however, is independent; they only *look* connected. This disconnected group of lines can be difficult to deal with when it comes to selecting something like the entire lot boundary at once.

To remedy this, BricsCAD offers a special kind of line called the “polyline.” Polylines are lines made up of many features, as suggested by the prefix *poly*. They can be made of lines and arcs, or of snaky-looking spline-like shapes; plines can have widths (which lines cannot), and even varying widths — all connected together as a single entity, as illustrated below.



1. Because the lines describing the house are all at right angles, I recommend using orthographic mode, known as “ortho” for short. The mode constrains cursor movement to the horizontal and vertical directions. Turn on ortho mode by clicking the **ORTHO** button on the status bar so that it turns black.



As confirmation, BricsCAD reports the following:
: <Ortho on>

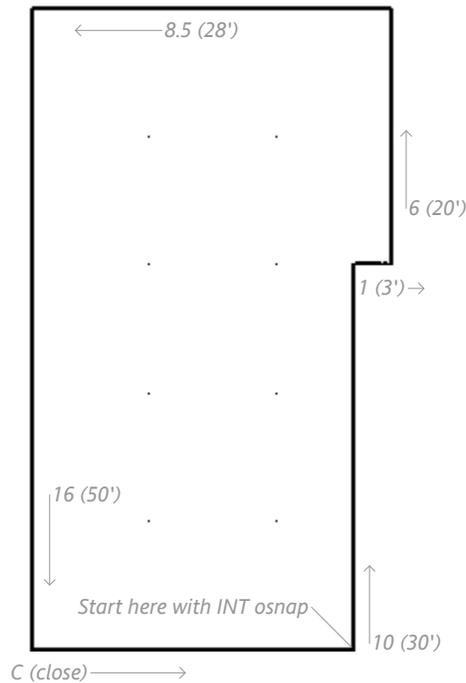
2. Draw the house's outline as a polyline. (The figure further below gives the dimensions for the outline of the house.) To do so, enter the **PLine** command.

: **pline**

BricsCAD prompts you for the point from which to begin drawing the polyline. But instead of specifying coordinates, you this time ask BricsCAD to find a geometric feature by using *entity snap*:

ENTER to use last point/Follow/<Start of polyline>: **int**

When you enter **int** (short for “intersection”), BricsCAD attempts to snap to the nearest intersection, rather than to the nearest 0.1m you specified earlier with the snap mode. *Entity snaps* override regular snap distance. Here, intersection snap overrides the 0.1m-snap. This is referred to as “entity snap override.”



QUICK SUMMARY OF THE PLINE COMMAND

A *polyline* consists of numerous lines and/or arcs connected together as a single entity. Polylines are drawn with the PLine command, and edited with the PEdit command. The PLine command has the following options for drawing polylines:

: **pline**

ENTER to use last point/Follow/<Start of polyline>:

Arc/Close/Distance/Follow/Halfwidth/Width/Undo/<Next point>:

Next point — specifies the location of the polyline's next vertex

Arc — draws polyarcs; you can switch between lines and arcs when drawing polylines

Close — joins the last endpoint with the starting point

Follow — draws a specific distance, in the direction of the cursor's angle

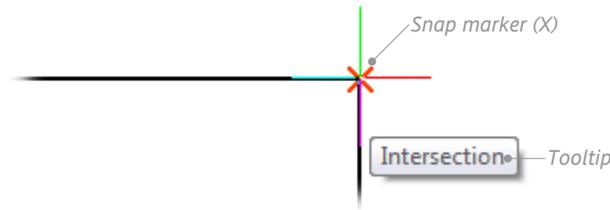
Halfwidth — specifies the width of polylines by the distance from their center line to their outside edge

Width — specifies the width of polyline segments; allows independent starting and ending widths for tapered polylines

Undo — undoes the last polyline drawing operation

Press **Esc** to exit the PLine command.

3. BricsCAD then prompts you to position the cursor near the intersection of two lines.
Snap to intersection of: (*Pick the Lower-right corner of the lot.*)



- a. Move the cursor until the aperture is over the lower-right corner of the lot boundary. The tooltip reports the entity snap found by BricsCAD.
- b. *Click* — press the left mouse button. BricsCAD “captures” the intersection as the starting point for the polyline.

DIRECT DISTANCE ENTRY

As CAD systems improve, they develop easier way to specify distances and angles. Let’s now look at a shortcut called “direct distance entry” (DDE). It combines cursor movement with keyboard entry — you directly show BricsCAD the angle, and then type in the distance. It’s a lot quicker than typing values of angles — and more intuitive, too.

Carrying on with the tutorial, here is how to use DDE:

4. The second line of the prompt displays many options, but don’t let them intimidate you; for now, you ignore all of them, except for the default, ‘Next point.’
Arc/Distance/Follow/Halfwidth/Width/<Next point>: (*Move cursor up*)

Here is how to do direct distance entry:

- a. First, move the cursor up towards the top of the drawing. The distance does not matter, only the direction. (Other ways of saying “move the cursor up” include “in the positive y direction,” or “to the North,” or “at 90 degrees”).
- b. Then, on the keyboard, type **10** and press **Enter**. (In imperial units, **30'**)

Notice that the line is drawn upwards, 10m long. Direct distance entry is like another form of relative coordinates: BricsCAD measures the 10m relative to the last entered point.

DYNAMIC INPUT

Let’s now move to another way of using direct distance entry, one that is even more interactive with the drawing area. It is known as “dynamic input,” because it displays distances and angles dynamically — right at the cursor in the drawing. Let’s see how it works:

5. To use dynamic input for the next segment, the 1m line, follow these steps:
 - a. On the status bar, ensure **DYN** is turned on. When **black** (instead of **gray**) then dynamic input is on.



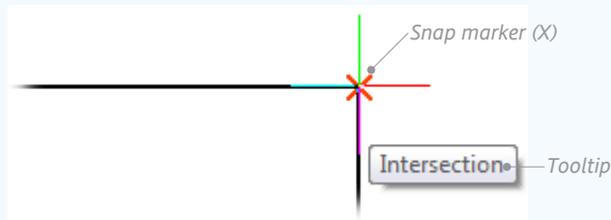
QUICK SUMMARY OF ENTITY SNAP MODES

BricsCAD has the following entity (object) snaps that look for specific geometric features on entities:

Mode	Command	Snaps to...
app	APParent	Apparent intersections of two entities
cen	CENter	Center of arcs, circles, and polyarcs
end	ENDpoint	Either end of lines, arcs, and other open entities
ext	EXTension	Extension of two entities
from	FROM	Offset distance from a specified point
gcn	GCENter	Geometric center of objects
ins	INSertion	Insertion point of blocks and text
int	INTersection	Intersection of lines, arcs, circles, and other entities
mid	MIDpoint	Middle of lines, arcs, and other open entities
nea	NEARest	Nearest point on the nearest entity
nod	NODE	Point entities
par	PARallel	Parallel to lines and other entities
per	PERpendicular	Perpendicular to lines, arcs, and other entities
qua	QUADRant	0-, 90-, 180-, and 270-degree points on arcs, circles, and polyarcs
qui	QUICK	First geometric feature found by BricsCAD
tan	TANgent	The tangents of arcs and circles

The **M2p** command is kind of like an entity snap, in that it finds the middle between two points.

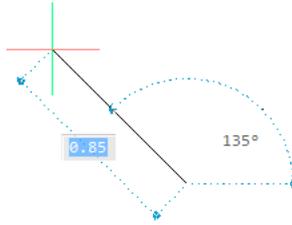
Because more than one entity snap can be active at a time, BricsCAD provides a number of visual cues to identify the current one. You can toggle the visual cues through the **Program Options | Display** section of the Settings dialog box.



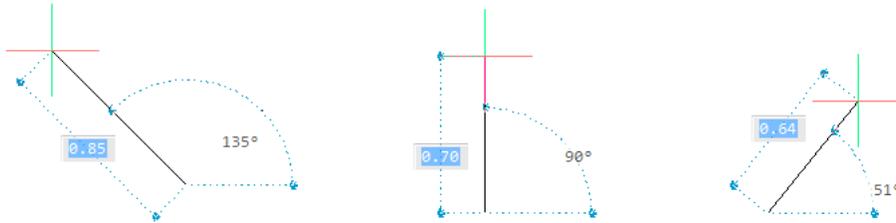
The table below illustrates on-screen markers associated with each entity snap mode.

<input type="checkbox"/> Endpoint	<input type="checkbox"/> Extension
<input type="checkbox"/> Midpoint	<input type="checkbox"/> Insertion
<input type="checkbox"/> Center	<input type="checkbox"/> Perpendicular
<input type="checkbox"/> Geometric Center	<input type="checkbox"/> Tangent
<input type="checkbox"/> Node	<input type="checkbox"/> Nearest
<input type="checkbox"/> Quadrant	<input type="checkbox"/> Apparent intersection
<input type="checkbox"/> Intersection	<input type="checkbox"/> Parallel

Notice that a couple of “dimensions” immediately appear in the drawing area:



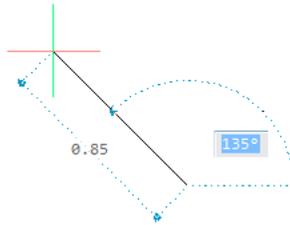
- b. These are known as “dynamic dimensions,” because they change as you move the cursor. Try it now: Move the cursor and watch how the length and angle values change.



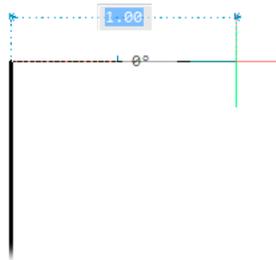
Left to right: Dynamic dimension values changing as the cursor moves

TIP When you enter numbers at the keyboard, they are entered into the blue field. Notice in the figures above that the **length** field is highlighted in blue.

To move between the length and angle fields, press the **Tab** key. In the figure below, the **angle** field (135 degrees) turns blue, as shown below.

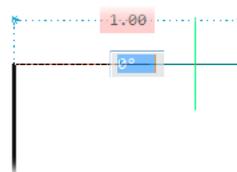


- c. Enter the length of the wall, **1** and then press **Tab**. (In imperial units: **3'**.)



Notice that the length field (1 or 3) turns **red**. This tells you that BricsCAD has accepted the value, and has locked it temporarily.

- d. When you pressed **Tab** in the last step, BricsCAD highlighted the angle field in **blue**. This means that it is ready for input: enter the angle of the wall, **0** (degrees)...

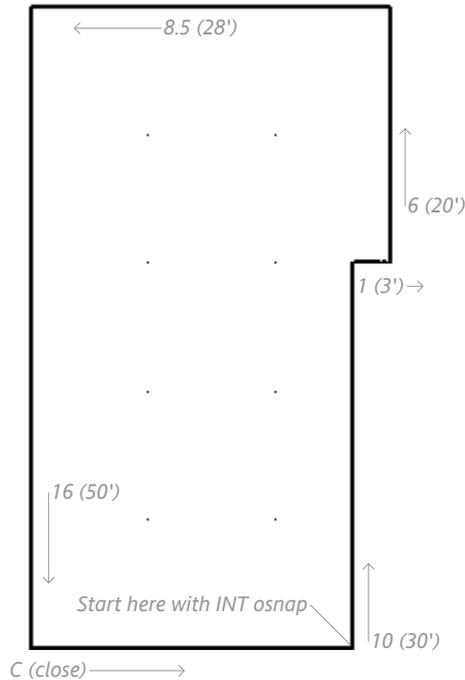


...and then press **Tab** again to lock the value. (To change the length and/or angle, press **Tab** to switch between the two fields.)

- e. Accept the length and angle values by pressing **Enter**.
6. Draw the remainder of the house outline by moving the cursor in the appropriate direction, and then entering the distances:

<u>Cursor Direction</u>	<u>Metric Distance</u>	<u>Imperial Distance</u>
East	1	3'
North	6	20'
West	8.5	28'
South	16	50'

7. Complete the polyline with the **C** option, as you did with the Line command.



When **Lot** was the working layer, the lines you drew showed up in blue. You changed the layer to **House**, and BricsCAD automatically drew the lines in black instead. This shows that lines take on the color specified by their layer. (You can, if you need to, change colors on-the-fly with the **Color** command.)

Moving the House into Position

Command	Move
Ribbon	Change Arrange Move
Menu Bar	Modify Move
Alias	M

Now that you've drawn the outline of the house, you need to move it into position. Entities are moved with the **Move** command.

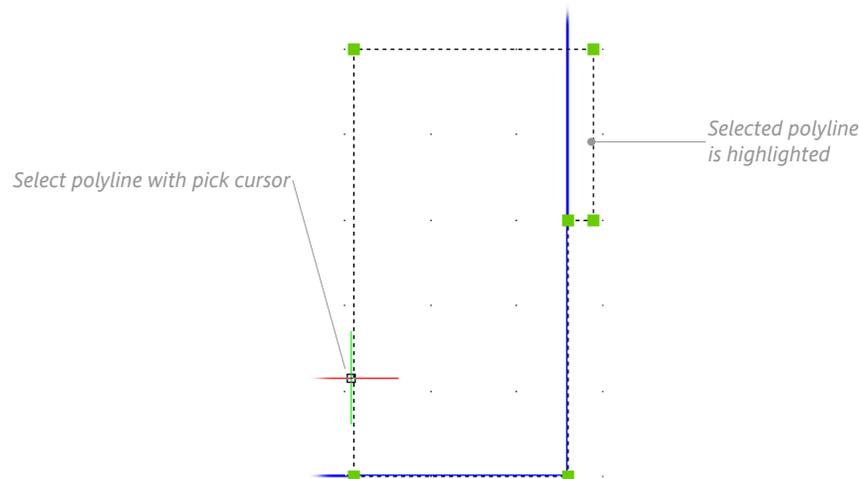
1. Enter **Move** at the 'Command:' prompt:
: move
2. Notice that BricsCAD asks what you want to move:
Select entities to move:

("Entities" is an older word that means the same as *objects*.)

At the same time, BricsCAD changes the crosshair cursor into a small square cursor, called the pick cursor.

3. Move the cursor to any part of the house's polyline, and then press the pick button.
Select entities to move: (*Pick the outline of the house*)

The entire house outline is highlighted. The highlighting shows as a dotted line, which is how BricsCAD lets you know it found the entity you picked.



4. BricsCAD reports the number of entity selected, and then lets you make additional selections, if need be. The prompt repeats:

Select entities to move: (*Press **Enter** to exit entity selection*)

Because you are moving just one polyline, press **Enter** to end the entity selection process.

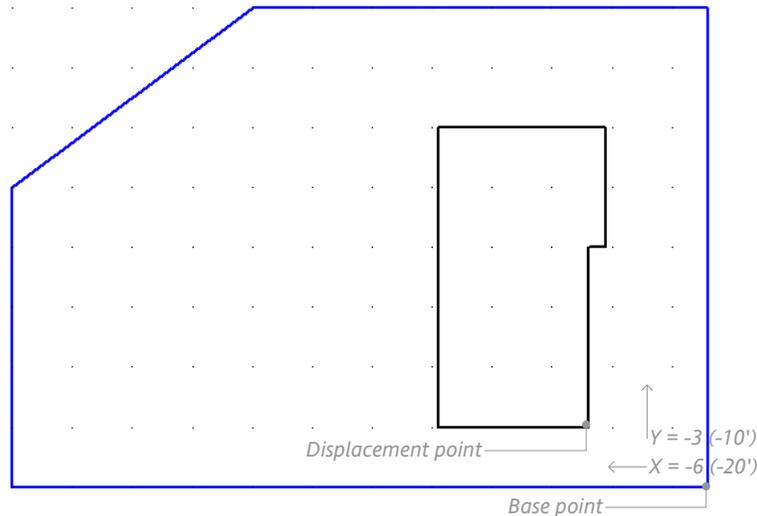
5. Just as when drawing lines, the Move command needs to know a *from*-point and a *to*-point. But here the *from*-point is named the "base point," as follows:

Enter base point: <Displacement>: **0,0**

(If you were to press Enter at the 'Enter base point <Displacement>' prompt, BricsCAD would ask for the displacement vector, which consists of a move in the x,y, and z directions.)

6. Now BricsCAD wants to know where you want to move the selected entities. The Move command calls the *to*-point the "displacement point," as follows:

Displacement point <ENTER to use base point as displacement>: **-6,3**



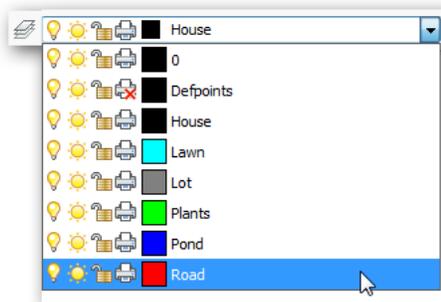
The coordinates -6,3 (in imperial units: **-20',10'**) tell BricsCAD to move the house left by 6m (x direction = **-6** meters) and up by 3m (y direction = **3**). BricsCAD instantly relocates the house much faster than a house mover. The Move command shows you a powerful aspect of CAD: no eraser dust!

7. It's a good idea to regularly save your work. Press **Ctrl+S**, and BricsCAD silently saves the drawing to disk. The only indication is that “:_qsave” appears on the command line.
Alternatively, you can click the diskette icon on the toolbar, enter the **QSave** command, or select **Save** from the **File** menu.

Starting on the Driveway

The final drafting for this project is to add the driveway and the street. Before drawing them, change the layer to **Road** using the technique described below. Once the layer is set correctly, you will draw the driveway and street outlines using a shortcut. Once the upper roadwork is drawn, you duplicate it with a single command to create the lower roadworks.

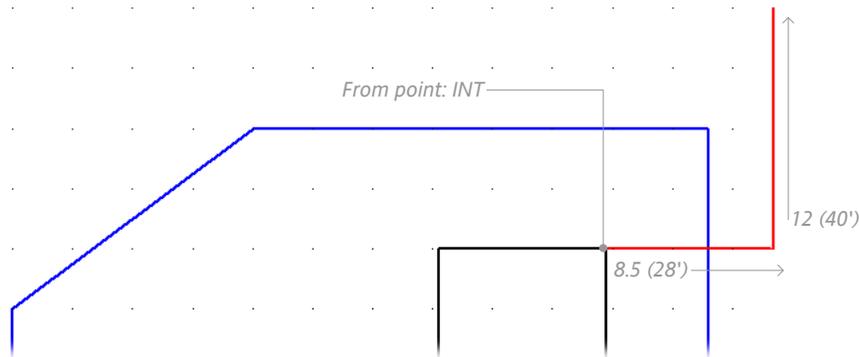
1. On the toolbar, status bar, or ribbon, click on the **House** layer name.
2. When the list box appears, select **Road**. BricsCAD changes the layer name from House to Road and the working color from black to red. Below, I show the toolbar version of the Layers droplist.



3. Ensure that ortho mode is still on by glancing at the status bar. The **ORTHO** button should be in black text.
4. Start the **Line** command:
: **line**

...and then draw the upper driveway and street line using direct distance entry, as follows:

```
ENTER to use last point/Follow/<Start of line>: int  
Snap to intersection of: (Pick upper-right corner of house)  
Angle/Length/Undo/<End point>: 8.5  
Angle/Length/Follow/Undo/<End point>: 12  
Angle/Length/Follow/Close/Undo/<End point>: (Press ENTER to end the command)
```



Recall that you can “back up” to undraw incorrect lines with **u** (short for “Undo”), as follows:

```
Angle/Length/Follow/Close/Undo/<End point>: 21  
Angle/Length/Follow/Close/Undo/<End point>: u  
Angle/Length/Follow/Close/Undo/<End point>: 12
```

FINISHING THE DRIVEWAY

Command	Fillet
Ribbon	Change Modify Fillet
Menu Bar	Modify Fillet
Alias	F

To add the *curb return* — the arc joining the driveway and street — use BricsCAD’s **Fillet** command. It draws arcs between intersecting lines. The lines don’t have to physically meet; BricsCAD takes care of extending (or trimming) the lines so that the arc is drawn between them.

You use the Fillet command by specifying the radius of the arc and then applying the fillet.

1. To start the command, enter the **Fillet** command, or else do as I do: just type **f** at the ‘:’ prompt.
: f
2. Notice that BricsCAD reports the current fillet radius. To change it, enter the hidden **r** option:
_fillet
Fillet (radius=0.50): Settings/Polyline/Radius/Trim/Undo/Multiple/<Select first entity>: r

TIP If you follow the official route by entering “s” for the **Settings** option, then BricsCAD opens the Settings dialog box at the section where the fillet radius is specified. Just another case of more than one way to skin cats.

3. Enter the fillet radius of one meter (three feet in imperial units):
Fillet radius <radius=0.50>: 1
4. With the fillet radius set to 1m, perform the filleting, as follows:
Fillet (radius=0.50): Settings/.../<Select first entity>: (Pick one Line)
Select second entity: (Pick the other Line)

BricsCAD automatically adjusts the two lines to fit the 1m arc between them.



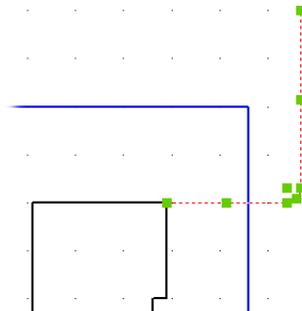
Mirroring Entities

Command	Mirror
Ribbon	Change Modify 2D Mirror
Menu Bar	Modify 2D Mirror
Alias	MI

Above, you used the Line and Fillet commands to create one part of the roadway. A very important concept behind computer-aided *anything* is that you should never have to draw the same line twice.

To illustrate the power of this concept, use the Mirror command to duplicate the lower driveway and street line without having to draw them! This command creates mirrored copies of entities, naturally: the copy is the reverse of the original.

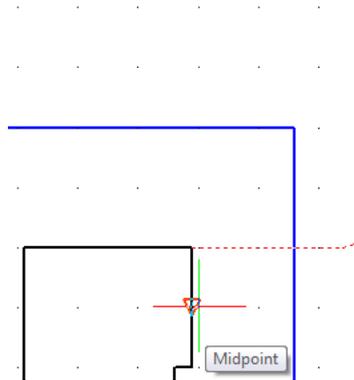
1. Enter the **Mirror** command:
: `_mirror`
2. BricsCAD asks you to select the entities you want to mirror. Use the cursor to pick the line and arc segments, as follows:
Select entities to mirror: (Pick the driveway Line)
Entities in set: 1
Select entities to mirror: (Pick the curb return)
Entities in set: 2
Select entities to mirror: (Pick the street line)
Entities in set: 3
Select entities to mirror: (Press **Enter** to end entity selection)



3. BricsCAD needs you to specify the mirror line, an imaginary line about which it mirrors the entities you just picked. Use **Midpoint** and **Perpendicular** entity snaps to help you:

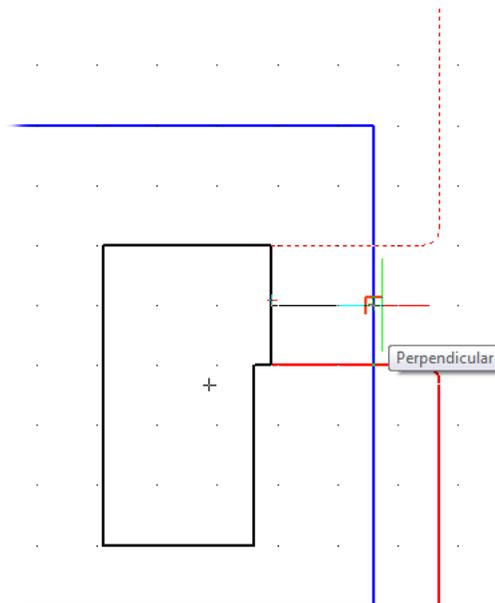
Start of mirror line: **mid**

Snap to midpoint of: *(Pick center of garage entrance)*



End of mirror line: **per**

Snap to perpendicular of: *(Pick right-hand Lot boundary)*



The length of the mirror line is not important, but its angle is crucial. For this reason, you used two new entity snap modes: **mid** to find the midpoint of the garage entrance and **per** to ensure the mirror line is perpendicular to the lot boundary.

4. At this point, BricsCAD gives you the option of erasing the old entities — the two lines and arc you picked. In most cases, as in this case, you *don't* want them erased:

Delete the original entities? <N> n

BricsCAD draws the lower driveway and street outline as a perfect mirror image of the upper set.

You have now drawn the outline of the lot, house, and driveway. The work you have done is valuable and it is important that you save the drawing to disk. Use the **QSave** command to store the drawing on disk.

Putting Drawings to Paper

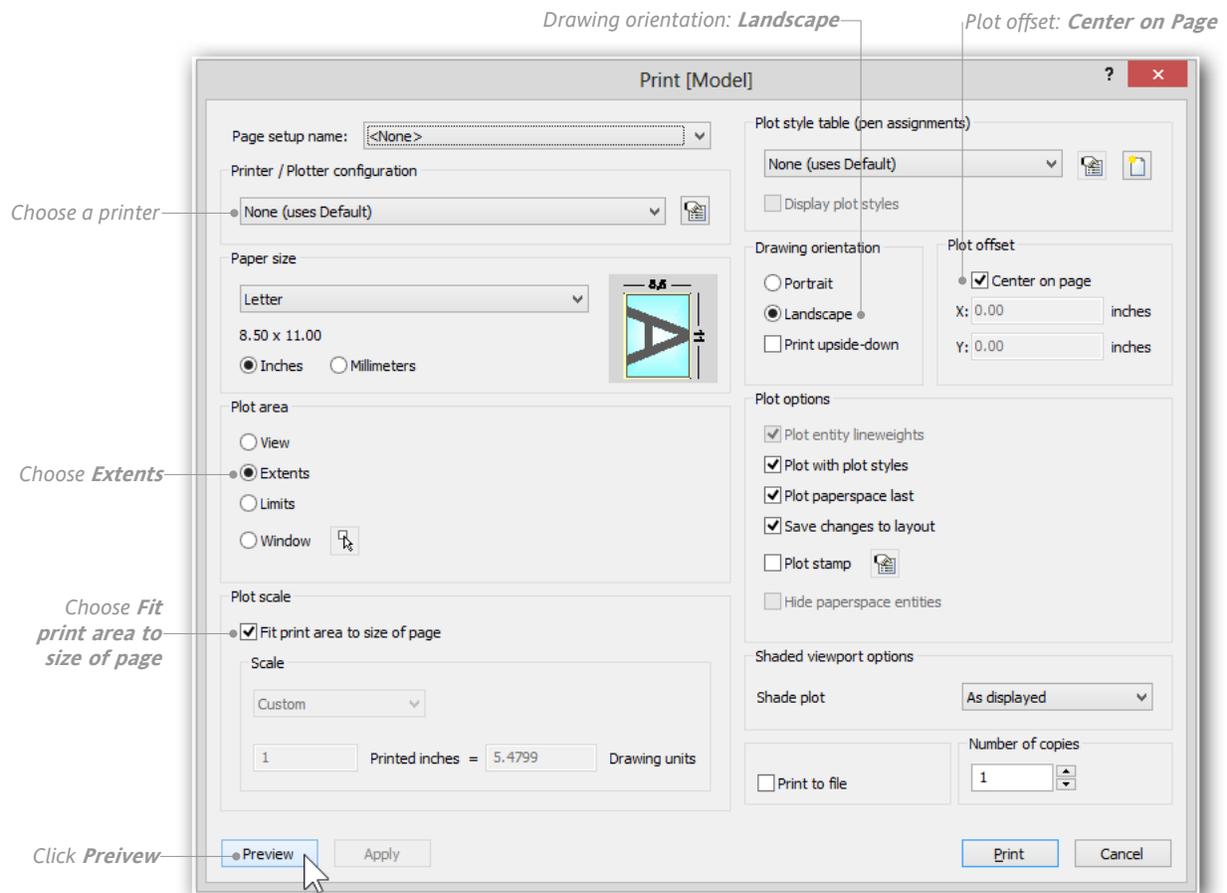
Command	Print
Ribbon	Home Print/Plot Print
Menu Bar	File Print
Shortcut	Ctrl+P

While it is efficient (and environmentally aware) to create and store drawings on computers and share them electronically via email or Web sites, you may want to print copies on paper. That lets you mark up the drawing with notes or show off your progress to friends and family!

The **Print** command sends the drawing to printers, plotters (oversize printers), and files. It is also known as the print command. Traditionally, “plotters” used motors to move the pen over the paper, while “printers” are what we mostly use today – laser printers and inkjet printers.

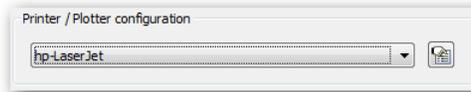
The following series of dialog boxes assumes you have a printer compatible with Windows or Linux attached to your computer. (The method for printing with Macs is roughly similar, but shows different dialog boxes.)

1. Press **Ctrl+P** or enter the **Print** command. Notice that BricsCAD displays the **Plot [Model]** dialog box. (Old timers might enter the command as “plot,” but this makes the command’s options appear in the Command bar, instead of in the dialog box.)



2. The dialog box has many options. Fortunately you can ignore most options, except for these:

Printer/Plotter Configuration



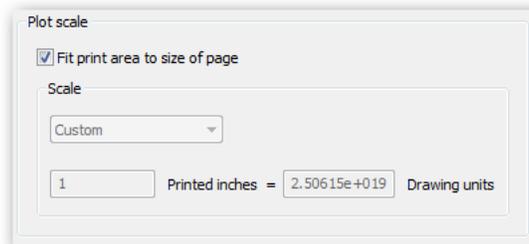
Select a printer from the **Name** droplist. BricsCAD supports local and networked printers.

Plot Area



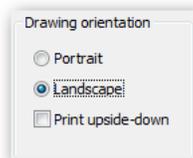
Select **Extents**. This option ensures *everything* in your drawing is plotted on the paper.

Plot Scale



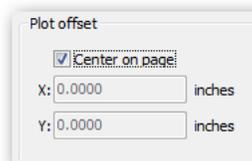
Check that **Fit Print Area to Size of Page** is selected. This ensures the entire drawing will be plotted, and that it won't be too large or too small.

Page Orientation



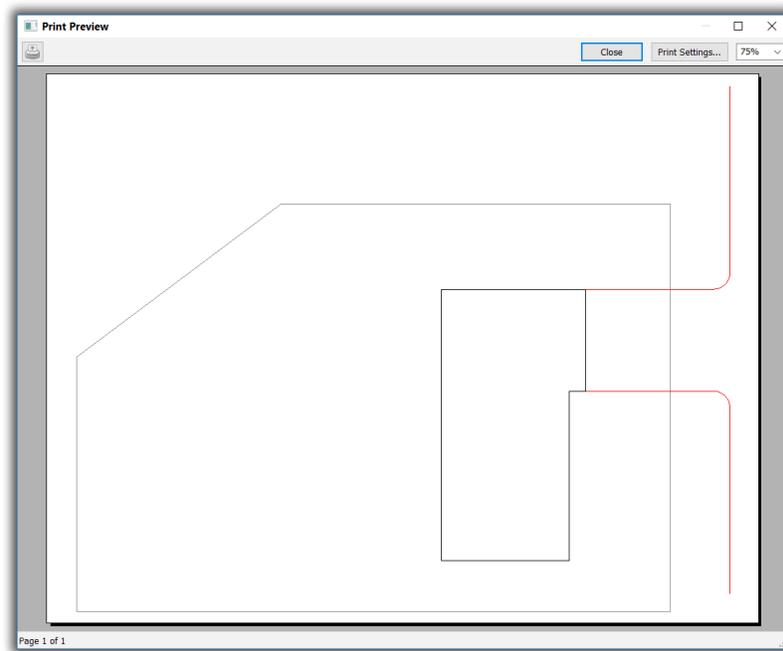
If the orientation of the paper doesn't match that of the drawing, then you need to change it. The paper is tall but the drawing is wide. To change the orientation of the paper, select **Landscape**.

Plot Offset



Turn on the **Center on Page** option to have the drawing nicely centered on the paper.

3. To check that the drawing fits the paper, click the **Preview** button. The drawing appears simulated on a sheet of paper.



QUICK SUMMARY OF PRINT PREVIEW

The Print Preview window has the following controls:



Print button prints the drawing as shown.

Close closes the window, and returns to the Print dialog box.

Print Settings also returns to the Print dialog box.

Zoom droplist changes the size of the preview image; scale ranges from 10% to 500%.

TIP If the **Preview** button is grayed out in the Print dialog box, this means you have not yet selected a printer for this drawing.



To make preview available, select a printer — any printer! — from the Printer/Plotter Configuration droplist:



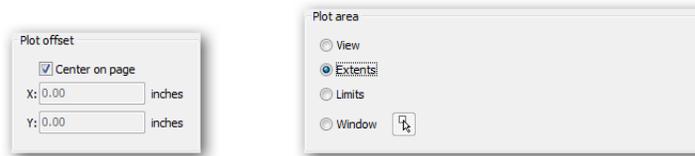
- If the drawing looks alright to you, then click **Print** to print the drawing.
(If, however, the preview shows a problem, click **Close** to exit, and return to the Print dialog box to adjust settings.)

BricsCAD sends the drawing to the printer, and it should appear a few seconds later looking exactly like the preview.

SOLVING PRINTING PROBLEMS

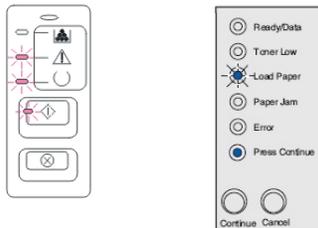
If the printer does not produce the drawing correctly, here are some items to check:

- Check in the Print dialog box that...
 - ...drawings are centered on the page
 - ...and check that the Extents of the drawings is printed.
 These two settings ensure that *all* of the drawing will be printed on the paper.



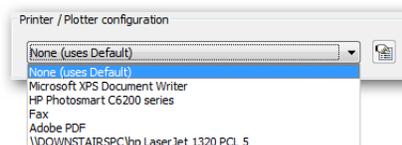
*Left: Print will be centered on page.
Right: Print will be made to drawing extents.*

- Check the printer:
 - Is it turned on?
 - Does it have paper?
 - Is the paper not jammed?
 - Are the toner or ink cartridges not low?
 - Are there any warning messages or lights?



Left and right: Warning lights from various printers.

- Check that the BricsCAD is printing to the correct printer. The operating system sometimes mistakenly lists names of printers that are not necessarily accessible by the computer.



- In some cases, the *spooler* acts up in Windows. The spooler is software that feeds data to the printer, allowing you to continue working with BricsCAD while the printer works in the background. Sometimes, the spooler plugs up, preventing the printer from receiving the data.

To fix the problem, follow these steps:

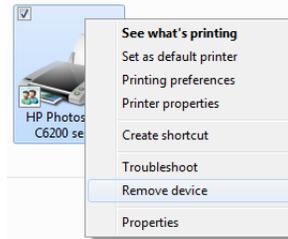
- Click **Start**, and then **Devices and Printers** in Windows 7.

(In Windows 8, press **Windows+W**, and then enter “printers” in the **Search** field. Choose **View Devices and Printers**.)

2. In the window, click **Add a Printer**. Install the same printer as the one that is giving you problems.



3. You now have two drivers for the same printer. Remove the first printer: right-click its icon, and then choose **Remove Device**.



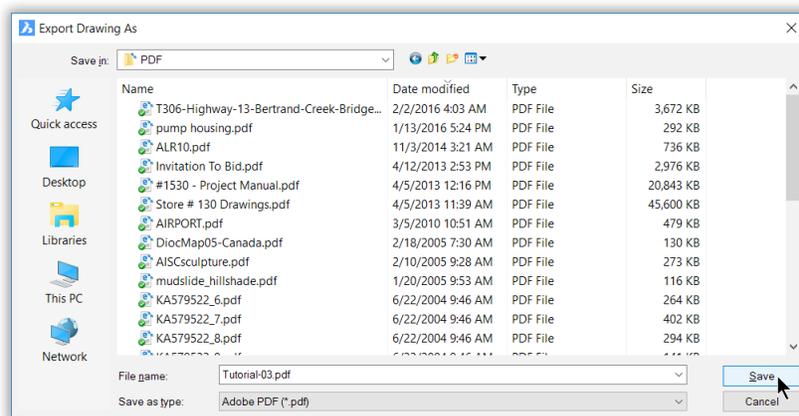
4. Printing should now work.

GENERATING PDFs OF DRAWINGS

The PDF format from Adobe is a popular way of sharing drawings *electronically*, so that you can send them by email or post them to Web sites. When drawings are saved in PDF format, they can be viewed but not altered. (NEW IN V17) In V17, BricsCAD moved the PDF function from the Export command to the ExportPDF command.

To save drawings in PDF format in BricsCAD, follow these steps:

1. Enter the **ExportPDF** command.
: **exportpdf**
2. Notice the Export Drawing As dialog box. You have the options of choosing a folder in which to save the file (“Save In”), and changing the file name (“File name”).



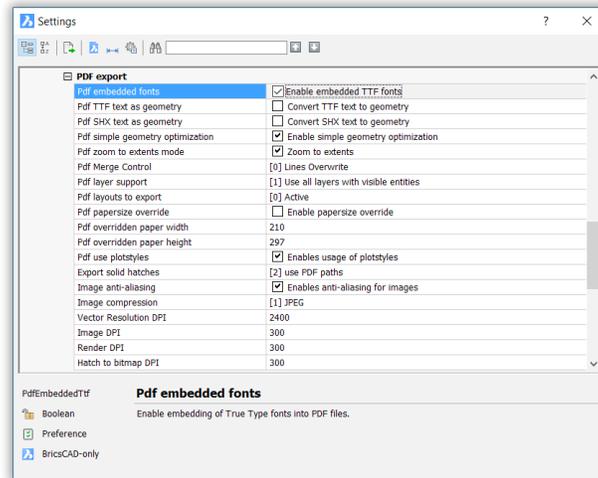
Click **Save**.

You can open the PDF file in Acrobat Reader or another program that displays *.pdf* files.

Specifying PDF Output Options

You control aspects of the PDF files produced by BricsCAD through the Settings dialog box. The quick way to access options specific to PDF files is with the **PdfOptions** command, which then opens the Settings dialog box at the section for PDF Export options, conveniently enough

Here is an explanation of the settings available, along with the default values in parentheses:



PDF Embed Fonts (on) embeds TTF font files in the PDF file:

- ▶ **PDF TTF Text as Geometry** (off) converts text made with TTF fonts into geometry (lines and arcs)
- ▶ **PDF SHX Text as Geometry** (off) converts text made with SHX fonts into geometry

PDF Simple Geometry Optimization (on) reduces the file size by combining individual line segments into polylines, and uses Bezier control points to define splines. This option does not affect the drawing, only the resulting PDF file.

PDF Zoom to Extents Mode (on) exports the drawing unscaled, so that its extents fit the page

(**NEW TO V17**) **PDF Merge Control** (0) determines the look of overlapping lines:

- 0 Overwrite; the topmost line obscures lines underneath it
- 1 Merge; overlapping lines are semi-transparent

PDF Layer Support (1) determines whether layers are included:

- 0 Exclude layers; export entities with no layer information (this is a form of security)
- 1 Export layers, except those turned off (default setting)
- 2 Export all layers, including those turned off

PDF Layouts to Export (1) specifies which layouts should be included in the PDF file:

- 0 Only the current “active” layout; this is like plotting a single layout (default setting)
- 1 All layouts; the PDF file becomes a multi-page PDF, with each layout on its own page

PDF Papersize Override — determines whether the PDF page should be of a custom size:

- ▶ **PDF Overridden Paper Width** (210) specifies the width of the PDF page in millimeters
- ▶ **PDF Overridden Paper Height** (297) specifies the height in millimeters

(**NEW TO V17**) **Export solid hatches** (2) as...

- 0 Bitmaps
- 1 Vectors
- 2 PDF paths (default)

PDF Use Plotstyles (on) determines whether the drawing is exported to PDF format using named plot styles; named plot styles must exist in the drawing

Image Anti-Aliasing (on) applies anti-aliasing (smoothing) to images requiring upscaling for export to PDF format

Image Compression (JPEG) specifies whether to compress raster images in drawings; 0 = off.

(NEW TO V17) **Vector Resolution** (2400) specifies the resolution of vector graphics.

Image DPI (300) specifies the minimum resolution of raster images

Render DPI (300) specifies the minimum resolution of rendered images

Hatch to bitmap DPI (300) exports hatch patterns as raster bitmaps, and specifies their resolution

Summary

Congratulations! You've drafted your very first drawing using a computer. You can save the drawing as a memento of your introduction to computer-aided drafting.

In the next lesson, you learn how to add details to drawings, such as hatching and symbols (blocks).

Adding Details to Drawings

So far you've learned how to draw lines to create the outlines of the lot, the house, and the driveway. Now it's time to add details, like the lawn, some trees, and a pond. Here you learn to use BricsCAD's intermediate commands, such as those that create ellipses, place hatch patterns, and draw offsets.

IN THIS CHAPTER

- Drawing circles and ellipses
- Editing with grips
- Applying hatch patterns to areas
- Arraying entities
- Making parallel offsets of entities
- Creating symbols (blocks)
- Performing real-time zooming and panning

KEY TERMS IN THIS CHAPTER

Drag — describes holding down the left mouse button while moving selected entities

Grip — describes the small square that indicates the editing points on a selected entity

Grips editing — refers to selecting entities, and then applying editing commands

Hatch pattern — displays a repeating pattern that indicates the material of entities

Palette or **bar** — describes a window that provides continuous information about BricsCAD

Real time — refers to an action in BricsCAD that occurs at the same time as you move the mouse

USEFUL ABBREVIATIONS

L last (used by entity selection)

W window (used by entity selection and zooming)

NEW COMMANDS

Command	Aliases	Menu Bar	Ribbon Tab
Array	ar	Modify 2D Array	Change Arrange 2D Array
Block	b	Tools Create Block	Tools Blocks Create Block
Circle	c	Draw Circle	Draw Draw Circle
Ellipse	el	Draw Ellipse	Draw Draw Ellipse
Hatch	h	Draw Hatch	Draw Hatch Hatch
Insert	i	Insert Block	Insert Blocks Insert Block
Offset	o	Modify Offset	Change Modify Offset
Pan	p, -p	View Pan	View Navigate Real-time Pan
PEdit	pe	Modify Edit Polyline	...
RtZoom	...	View Realtime Realtime Zoom	View Navigate Real-time Zoom
Zoom Window	z w	View Zoom Zoom Window	View Zoom Zoom Window

Dividing the Lot

The yard has both a lawn and a garden area. In this tutorial, you draw the boundary between the two areas with a polyline using the PLine command, and then smooth it with the PEdit (polyline edit) command.

If BricsCAD is not running, start it now. If you did not complete the previous lesson, open the *tutorial-04.dwg* file. You can download them from my public Dropbox folder at <https://dl.dropboxusercontent.com/u/28941239/Inside-BricsCAD-Tutorial-Files.zip>. (If you're doing the tutorials in imperial units, open *Imp-tutorial-04.dwg*.)

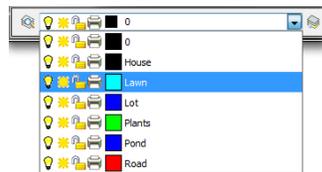
Before starting to draw, though, take these steps:

1. You won't be drawing at right angles, so turn off ortho mode by clicking the **ORTHO** button on the status bar. It should look gray to indicate it is turned off.



ORTHO

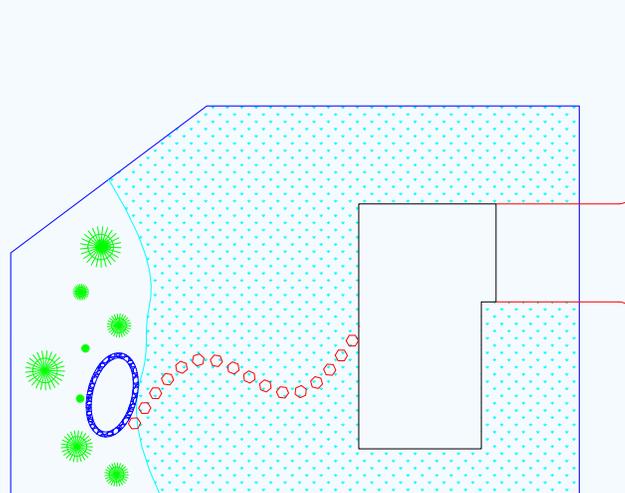
2. Change the working layer to **Lawn**; its color should be cyan (light blue). Here are several different ways to do this:
 - > In the toolbar or ribbon interfaces, choose "Lawn" from the Layers droplist (as shown below).



- > Or from the Properties panel, choose "Lawn" from the Layers droplist

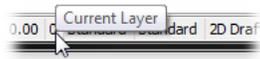
THIS CHAPTER'S DRAWING

By the end of this lesson, your drawing should look similar to this one:

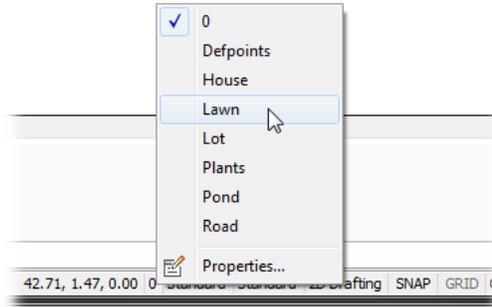


Tutorial-05 drawing file

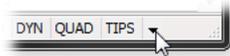
- Or in the status bar, right-click the current layer field...



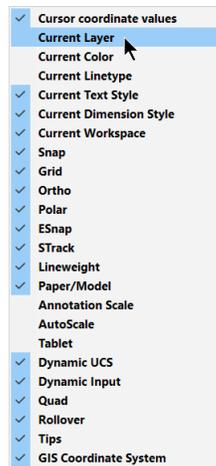
... and then choose “Lawn” from the list



- TIP If the Current Layer button is not visible on the status bar, then you can add the button like this:
- Click the black arrow at the end of the status bar



- From the shortcut menu, choose **Current Layer**.



- Or at the command prompt, you can enter the **CLayer** system variable. (Clayer is short for “current layer,” and is a quick way to change layers at the keyboard. In fact, this is what the Current Layer field on the status bar uses.)

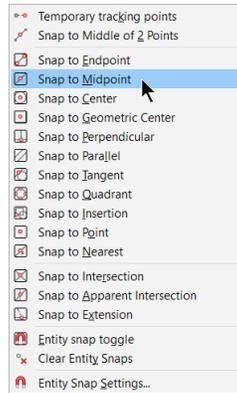
```
: clayer
New current value for CLAYER <"0">: lawn
```

Now the drawing is ready for you to draw the lawn-garden boundary.

- To draw the polyline, start the **PLine** command.

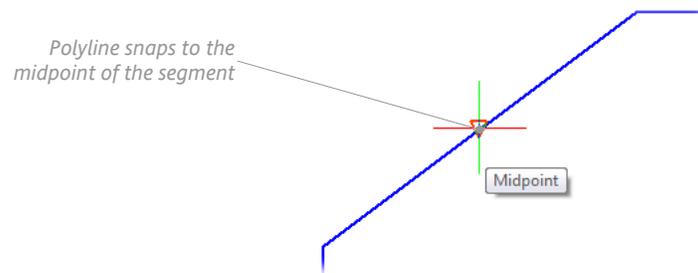

```
: pline
```
- To determine the polyline’s starting point, you use an *entity snap* mode. But this time, you will access entity snap modes differently than in last chapter’s method:
 - Hold down the **Shift** key on the keyboard.

- b. Press the right mouse button, and then let go of the **Shift** key. Notice the new menu that pops up on the screen. This is called a “shortcut menu.” The menu lists all of BricsCAD’s entity snap modes.



- c. Move the cursor down to **Snap to Midpoint**, and then click. On the command line, notice that BricsCAD reports that MIDpoint esnap is activated:
Start of polyline: `_midpoint`

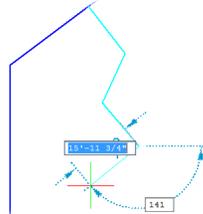
5. Move the cursor to any point on the diagonal portion of the lot line, and then select the line.
Snap to midpoint of: *(Pick diagonal Lot Line)*



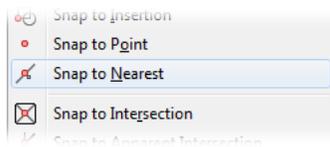
MIDpoint esnap forces to BricsCAD snap to the precise midpoint of the lot’s diagonal line.

6. Moving your way down toward the bottom of the yard line, pick a few more points at roughly 3m to 6m intervals. (In imperial units: 10’ to 20’ intervals.)

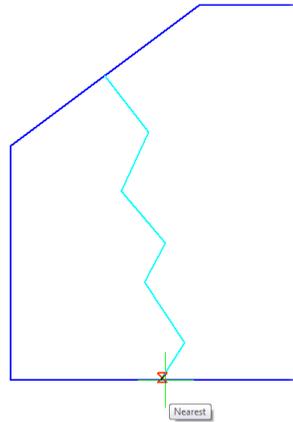
TIP If you are not sure how far 3m or 6m is (10ft or 20ft), keep an eye on the distance displayed by the coordinates on the status bar. Or else turn on **DYN** on the status bar, and watch the distance dynamically:



7. When you get to the bottom lot line, press **Shift**+right mouse button. From the shortcut menu, select **Nearest** entity snap.



- Pick anywhere the line crosses the cursor's aperture box. Again, BricsCAD snaps the polyline precisely to the lower lot line.



- Press **Esc** to end the PLine command.

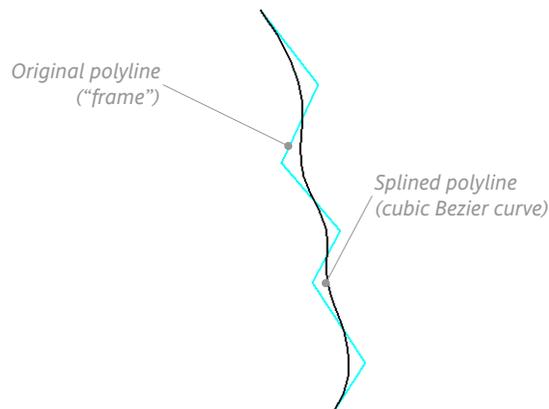
Here is the reason you created the boundary as a polyline: you can now use the **PEdit** command (short for “polyline edit”) to change the crooked polyline segments into a smooth flowing curve easily.

SMOOTHING POLYLINES

Command	PEdit
Shortcut	<i>double-click the polyline</i>
Menu Bar	Modify Edit Polyline
Alias	PE

The PEdit command's purpose is to change the look of polylines. It has many, many options; don't worry, you'll use just one of them.

- To edit the polyline, begin the **PEdit** command:
: **pedit**
- Pick the polyline.
Select polyline to edit: (*Pick the polyline, or type L to choose the Last-drawn entity*)



- Use the **Spline** option to smooth the straight lines into a flowing curve, as follows:
Edit polyline: Edit vertices/Close/Decurve/Fit/Join/Linetype-mode/Reverse/Spline/
Taper/Width/Undo/<eXit>: **s**

The straight lines disappear and are replaced by a smooth curve. (Technically, BricsCAD redrew the line segments as a *cubic Bezier curve* based on the polyline frame.)

- Exit the PEdit command by pressing **Esc**.

QUICK SUMMARY OF THE PEDIT COMMAND

The PEdit command edits polylines:

```
: pedit
Edit polyline: Edit vertices/Close/Decurve/Fit/Join/Linetype-mode/Reverse/Spline/Taper/
Width/Undo/<eXit>:
```

Edit vertices — edits the width and position of individual segments; inserts and removes vertices.

Close — closes an open polyline by drawing a segment between the starting and ending vertices; this prompt is displayed only when an open polyline is being edited.

Open — opens a closed polyline by erasing the last segment drawn; this prompt is displayed only when a closed polyline is edited.

Decurve — reverts a curve-fit or splined polyline to its original form.

Fit — applies a curve fit to the polyline.

Join — joins this polyline with another polyline, line, or arc; entities must be connected to form a single polyline (best done with entity snaps); this option fails when gaps exist between the entities to be joined.

Linetype-mode — determines if linetypes fit over the entire length of the polyline, or between vertices.

Reverse — reverses the direction of the polyline.

Spline — applies a Bezier spline to the polyline.

Taper — applies a varying width to segments.

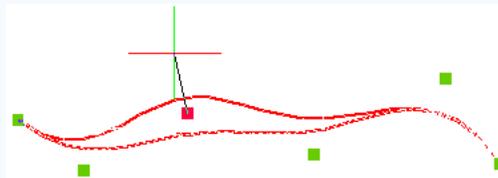
Width — applies a uniform width to all segments making up the polyline.

Undo — undoes the last polyline editing operation.

eXit — exits the command.

TIP While you can use the PEdit command's **Edit vertices** option to change the shape of polylines, it is much easier to use grips editing, like this:

1. Select a polyline.



2. Notice the green squares, called "grips." Pick one; it turns red.
 3. Drag the red grip. Notice that the polyline changes its curvature.
 4. When done editing the polyline, press **Esc**.
-

TIP The letter **L** is short for "last," and is shorthand notation for selecting the last entity drawn still visible on the screen. After you select one or more entities, BricsCAD reports the number selected: "1 found."

Grips Editing

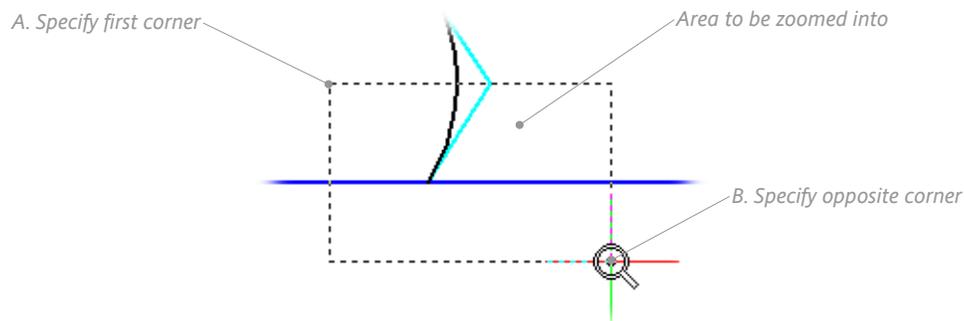
When you edited the polyline, you started the PEdit command, then selected the polyline to edit. BricsCAD can do this in reverse: select an entity, and then edit it. This is called “grips editing.”

Grips editing can sometimes be a faster and more direct way to edit CAD drawings. I say “sometimes,” because not all editing commands lend themselves to it. Here you use grips editing to change the shape of the splined polyline separating the garden from the lawn.

TIP If floating toolbars obscure an entity you want to edit, use the Pan command or a scroll bar to move the entity into view. Alternatively, hold down the roller button on the mouse and then move it to pan.

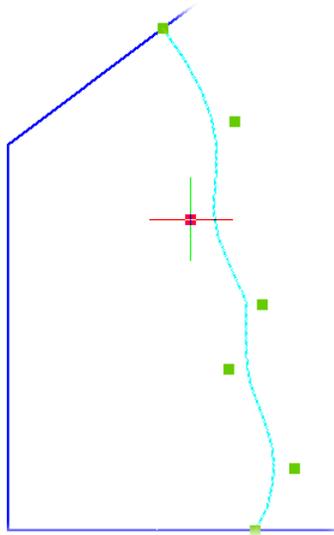
1. First though, enlarge the view of the splined polyline with the **Zoom Window** command:
 - a. Enter the **Zoom** command:
: **zoom**
 - b. Specify the Window option by entering ‘w’:
Specify corner of window, enter a scale factor (nX or nXP), or
[All/Center/Dynamic/Extents/Previous/Scale/Window/Entity] <real time>: **w**
Specify first corner: *(Pick a point)*
 - c. The Window option requires you to pick a rectangular area on the screen to magnify.
Specify opposite corner: *(Pick another point)*

When you pick points for the first and other corners, you specify the two opposite corners of the rectangle, as shown in the following figure.

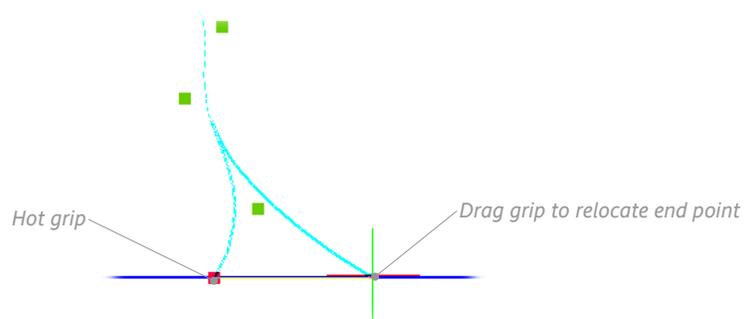


Now that the working area is larger, you can start to edit the polyline using grips. In an earlier lesson, I pointed out the small square at the center of the crosshair cursor. It is called the “pickbox.” When BricsCAD displays the pickbox, you can select entities without any command being active.

- Pick the polyline. Notice that the polyline changes from solid to dashed, and that small green squares appear along the splined polyline. The green squares are called “grips,” because they let you grip entities. The green grips that are not on the spline indicate vertices of the straight polyline segments you originally drew, before splining it.



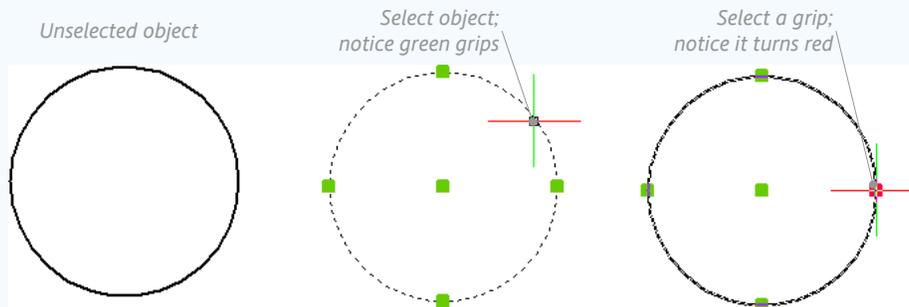
- Pick the green grip at the very bottom end of the polyline. BricsCAD prompts:
New location for control point: **nea**



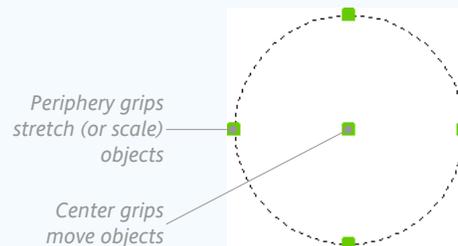
- Enter the **NEAr** entity snap mode, and then move the cursor. As you do, the last segment of the polyline curves and arcs to follow you. (The NEAr entity snap ensures the polyline ends precisely at the lot line.)
- Click along the lot line where you want the end of the polyline to move to. If you wish, feel free to interactively reshape the rest of the polyline, segment by segment.
- When you are finished reshaping the polyline, press **Esc** twice to exit grips editing.
- Enter the **Zoom Previous** command to see the entire drawing again.

QUICK SUMMARY OF EDITING WITH GRIPS

Grips editing is interactive: you select an entity, select one of its grip, and then manipulate the grip to modify the entity.



Most grips look the same, and so it can be hard to determine their function — until you start to drag one of them. Typically, an interior grip moves the entity, while grips on the periphery stretch or scale the entity.



GRIPS EDITING COMMANDS

During grips editing, BricsCAD makes available these editing commands: Stretch, Move, Rotate, Scale, Mirror, and Copy. (The six operate identically to their non-grips equivalents.) To see the commands, you need to press the Spacebar, repeatedly. Each time you do, BricsCAD displays another set of prompts:

```
** STRETCH **  
<Stretch to point>/Base point/Copy/Undo/eXit:  
  
** MOVE **  
<Move to point>/Base point/Copy/Undo/eXit:  
  
** ROTATE **  
<Rotation angle>/Base point/Copy/Undo/Reference/eXit:  
  
** SCALE **  
<Scale factor>/Base point/Copy/Undo/Reference/eXit:  
  
** MIRROR **  
<Second point>/Base point/Copy/Undo/eXit:
```

After **** MIRROR ****, BricsCAD repeats the cycle. The other editing options are:

Base point — specifies a base point other than the hot grip.

Copy — copies the entity, using the current grip as the base point.

Reference — changes the reference point from the current hot grip.

Undo — undoes the last editing operation.

eXit — exits non-modal editing mode (or press ESC).

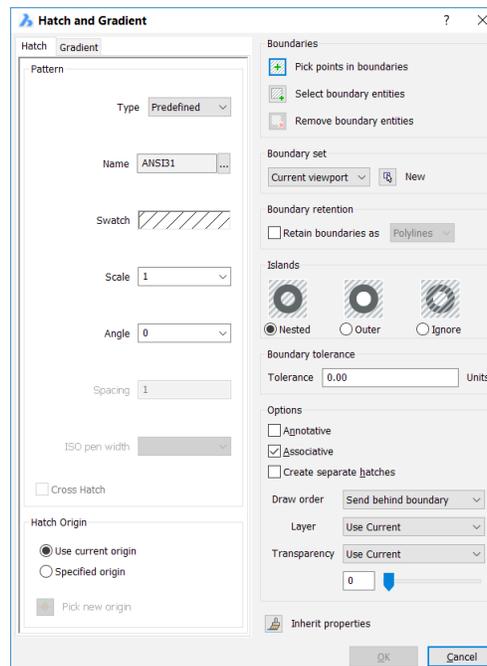
HATCHING THE LAWN

Command	Hatch
Ribbon	Draw Hatch Hatch
Menu Bar	Draw Hatch
Alias	H

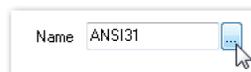
You created the boundary between the lawn and garden, but how do you show the difference between the areas of grass and dirt? One way is to add symbols that identify areas. In BricsCAD, this is done with the **Hatch** command, and in this tutorial you hatch the lawn with the symbol for grass.

1. First, turn off the grid, since its lines can interfere with seeing the hatch pattern. On the status bar, click **GRID** so that the word turns gray.
2. Enter the **Hatch** command to start the Hatch command:
: hatch

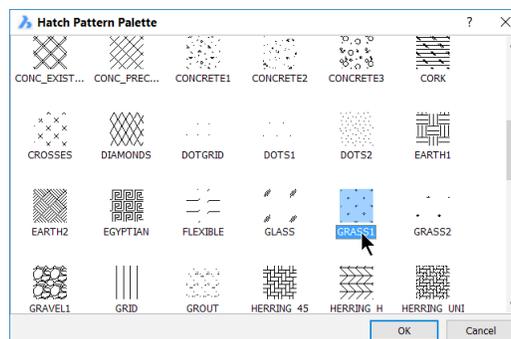
Notice the Hatch and Gradient dialog box:



3. To find the grass hatch pattern, click the  button to the right of “Name.”

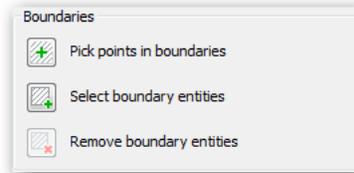


4. BricsCAD displays the Hatch Pattern Palette dialog box. The patterns are listed alphabetically; look for the pattern named “Grass1.”



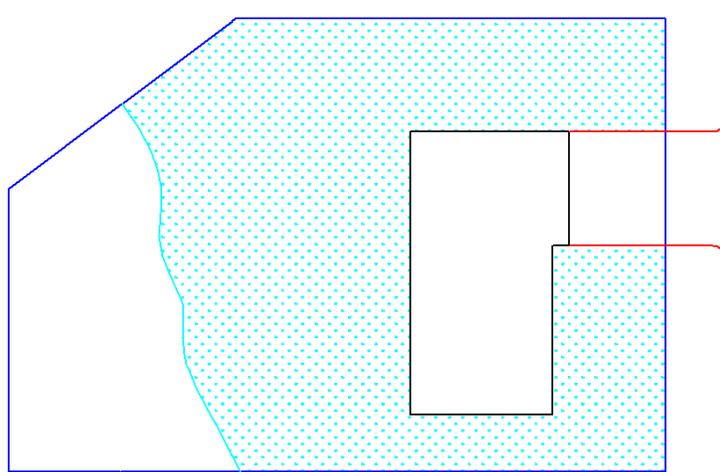
5. Select **Grass1**, and then click **OK** to return to the Hatch dialog box. Notice that **Name** changes to “Grass1.”
6. Ensure the **Scale** is set to **1** for metric drawings. (For imperial drawings, change the **Scale** to **25**.) If the scale factor were too small, BricsCAD would refuse to draw the hatch pattern, and then complain,

Hatch spacing too dense, or dash size too small.
Please increase pattern scale or MAXHATCH value.
7. BricsCAD has a useful feature that searches a contiguous area, no matter how many different borders the area has. To employ this feature, click the **Pick points in Boundaries**  button.



8. The dialog box disappears and BricsCAD prompts to pick a point. Do so anywhere in the lawn area.

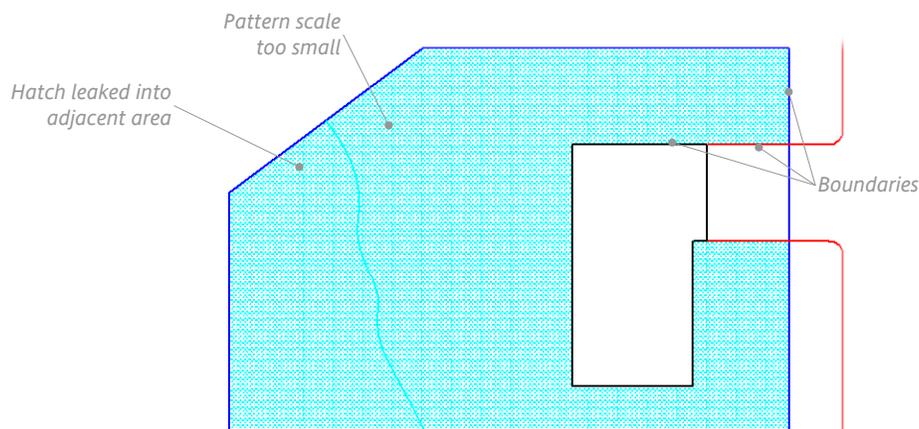
Select a point to define a boundary or hatch area: *(Pick a point inside the lawn area)*



Notice that BricsCAD immediately previews the hatch pattern for you. It shows that the scale factor is good enough. Note how precisely the hatch pattern is applied, and how it is automatically clipped along boundaries. Try doing a hatch pattern that neatly and that quickly by hand!

The hatch preview lets you check for two important things:

- > That the pattern floods the correct area. Look out for patterns that leak into unwanted areas, or don't appear at all
- > That the pattern is applied at a good scale factor. Look for one that are too large or too tightly spaced



TIPS When the scale is much too large, the hatch may seem invisible. When too small, the hatch pattern look like a solid and can takes a very time to display.

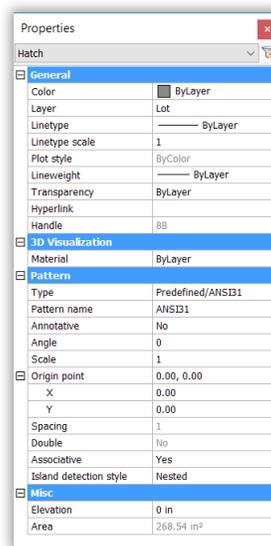
There is also a **-Hatch** command that operates at the command line. It is meant for use by macros and programming routines.

9. Press **Enter** to return to the dialog box:
Select a point to define a boundary or hatch area: (*Press Enter*)
10. After you press Enter, the Hatch dialog box reappears. Click **OK** to exit it.

With all this hard work on your drawing, it's a good idea to save the drawing to the computer's hard disk with the **Save** command... right now!

Once hatch patterns are in place, they are not cast in stone. You can change them in several ways:

- > You can move entities that make up the boundary of the pattern, and the hatching automatically updates itself.
- > The hatch does not need to stay in place. You can move it, copy, erase, and so on.
- > Click the hatch entity and BricsCAD brings up the Properties palette, illustrated below. You can use it to change the color, pattern, scale factor, rotation angle, and so on.



Creating Symbols

You've given the lawn area its grass. Now it's time to add trees and shrubs to the garden area. Instead of drawing complex things like trees, landscape architects typically draw simple representations, such as a circle with radiating lines.

And instead of drawing the trees and shrubs over again, designers draw one, and then insert repeating copies of it. The most efficient method is to use the Block command to create one copy, and then insert multiple copies with the Insert command.

DRAWING CIRCLES

Command	Circle
Ribbon	Draw Draw Circle
Menu Bar	Draw Circle
Alias	C

In this tutorial, you learn how to draw trees as simplified symbols, and then turn them into *blocks*.

1. Before starting to draw first tree, make sure the working layer is set to **Plants**. From a Layer droplist, select layer "Plants."
2. Next, draw a **0.15**-meter radius circle. Enter the **Circle** command, and then specify a radius of 0.15 units (in imperial units, **6"**):
: circle
2Point/3Point/RadTanTan/Arc/Multiple/<Center of circle>: (Pick a point anywhere in garden area.)
Diameter/<Radius>: **0.15**

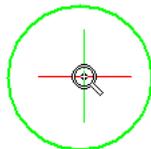
When the radius is 0.15, the diameter is 0.3, so don't be surprised when BricsCAD draws the circle larger than you expect. (Recall that the radius is half the diameter.)

Zooming in Real Time

Command	RtZoom
Shortcut	<i>scroll the mouse wheel</i>
Ribbon	View Navigate Real-time Zoom
Menu Bar	View Realtime Realtime Zoom
Command	Regen
Menu Bar	View Regen

The program lets you zoom in and out without needing to specify a zoom size or ratio. This is done with the mouse and the **RtZoom** command:

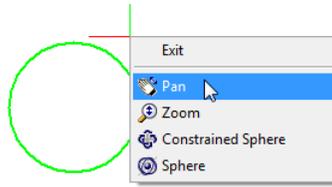
3. The 0.3m circle looks very small on the screen. The RtZoom command lets you see your work more clearly. The command name is short for "real time zoom."
: rtzoom
>>Press ENTER or Esc to complete, or right-click to display context menu....



- a. The cursor changes to a magnifying glass. Drag the cursor over the circle.
- b. As you move the mouse forward and backward, BricsCAD dynamically increases and decreases the zoom. (As an alternative, if you have a mouse with a wheel, roll the wheel forward to zoom in.)

NOTE To *drag* means to (a) hold down the left mouse button, (b) move the mouse, and then (c) let go of the mouse button.

4. If you find the tree going off the edge of the screen, right-click the drawing, and then choose **Realtime Pan** from the shortcut menu. This pans (shifts) the drawing in real time: as you drag the cursor, the tree-circle is moved back to the center of the drawing area.



5. When the tree-circle is at a satisfactory size, press **Esc** to get out of real-time pan-and-zoom mode:
>>Press ENTER or Esc to complete, or right-click to display context menu....(Press **ESC**.)

If the circle looks like an octagon or diamond, select **View | Regen** to clean it up, as follows:

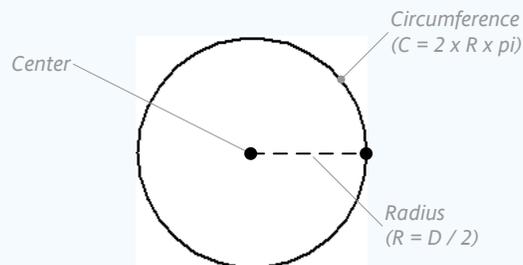
: _regen

Now that the circle looks rounder and larger, it is easier to work with.

QUICK SUMMARY OF DRAWING CIRCLES

BricsCAD provides several methods for drawing circles. Use the one that suits your need the best.

: **circle**
2Point/3Point/TanTanRad/Arc/Multiple/<Center of circle>:

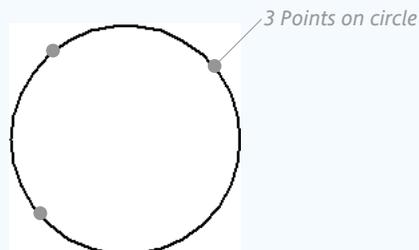


Center-Radius — pick the center point, and then specify the radius.

Center-Diameter — pick the center point, and then specify the diameter.

2-Points — pick two points to define the diameter.

3-Points — pick three points to define the circumference.



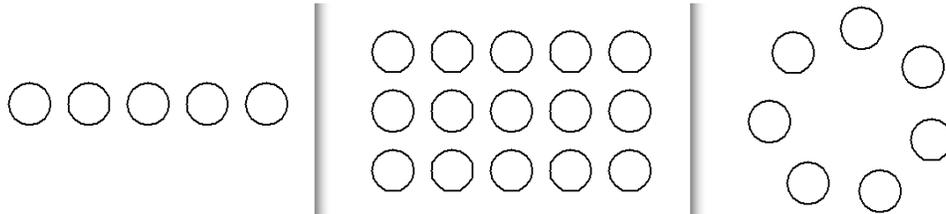
Radius-Tangent (RadTanTan) — pick two points of tangency to other entities, and then specify radius.

Convert Arc to a Circle — closes an arc to form a circle.

CREATING ARRAYS

Command	Array
Ribbon	Change Arrange 2D Array
Menu Bar	Modify 2D Array
Alias	AR

The tree branches are represented by an array of lines. To create the array, use the **Array** command to create the radiating lines. This command can create several types of arrays: linear, rectangular, and polar (circular). They are illustrated below.



Left to right: Linear, rectangular, and polar arrays.

For this tutorial, you draw one line and then array it to create ... more.

1. Enter the **Line** command:

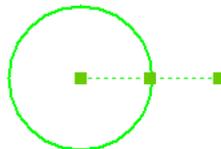
```
: line  
Specify first point: cen  
of (Pick center of circle)  
  
Specify next point or [Undo]: (Pick point anywhere outside of circle)  
Specify next point or [Undo]: (Press ENTER to end the command)
```

TIP To draw a perfectly horizontal or vertical line, hold down the **Shift** key. This action turns on ortho mode temporarily, for as long as you hold down the Shift key.

The **CENter** entity snap begins the line precisely at the center of the circle. The other end of the line extends beyond the edge of the circle.



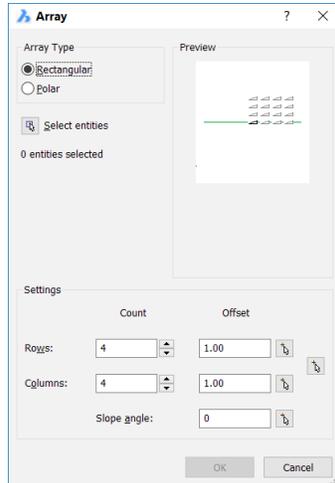
2. Select the line you just drew. Notice that it is highlighted.



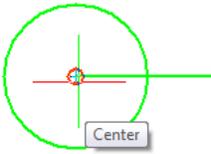
3. Enter the Array command:

```
: array
```

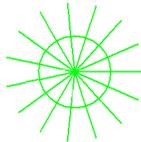
Notice the Array dialog box.



4. Entering array parameters in this dialog box is like filling out a form:

Array Parameter	Value
Array Type	Polar
Select Entities	Click  Select Entities , and then choose the line
Center	Click  Pick Center Point , and then choose the center of the circle like this: Base/Center of polar array: cen Snap to center of: <i>(Choose the circle)</i>
	
Number of items	15
Rotate items as copied	Yes

5. Ignore the other settings (leave them at their default values); see figure above. Click **OK**.



Later in this chapter, you turn this symbol into a block. But first, a detour to interactive arrays.

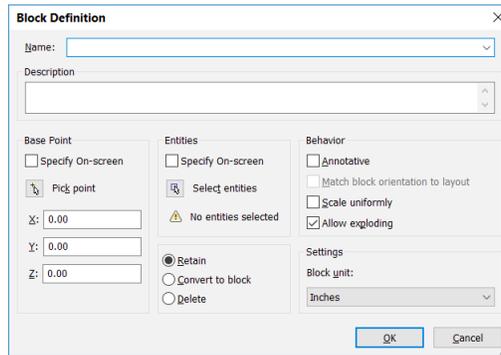
Making Blocks

Command	Block
Ribbon	Insert Blocks Create Block
Menu Bar	Tools Create Block
Alias	B

CAD draws symbols more quickly and accurately than you could by hand. The key is to turn the symbols into blocks, and then insert the blocks into drawings. In this section, you do this by adding trees to the garden area.

- To turn the tree symbol into a block, use the **Block** command:
: **block**

Notice that BricsCAD displays the Block Definition dialog box.

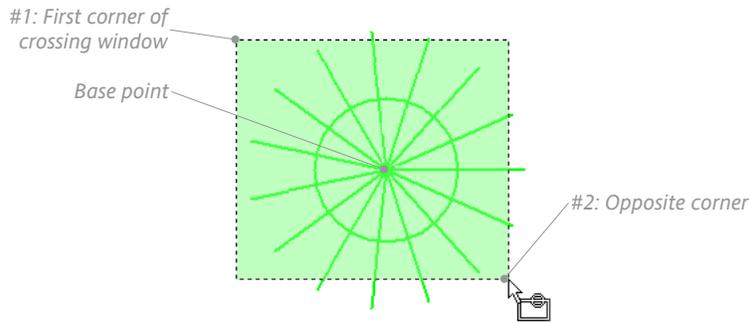


- Enter the following parameters:

Block Option	Value	Notes
Name	Tree	You can give the block any name you like, up to 255 characters long
Base point		Click Pick Point button, and then choose the center of the circle
Entities		Click Select Entities , and then choose all 16)
Convert to block	On	Converts the entities to a block when you click OK
Scale uniformly	On	Forces the circle to remain round

- The base point is used later by the Insert command; this point is also called the “insertion point” — the point where the block is inserted in the drawing. The center of the tree symbol is a logical spot base point, so use CENter entity snap:
Insertion point for new block: **cen**
Snap to centerpoint of: (*Pick the circle*)
- You need to tell BricsCAD which entities to turn into a block. BricsCAD lets you select entities by several different ways. So far, you have picked them one at a time with your mouse or with the L (last) option. Just as you windowed the zoomed-in view, you can window the entities you want to select with the **W** option (short for “Window”).
Select entities for block: **w**
- You pick the two corners of a rectangle that encompasses the circle and lines making up the tree symbol:
First corner of crossing window: (*Pick point 1, shown in figure below*)
Opposite Corner: (*Pick point 2, shown in figure*)

Entities in set: 30
 Select entities for block: (Press **Enter** to end entity selection)



3. Click **OK**. BricsCAD records the tree symbol as a block in the drawing file. In the next section, you place many trees as blocks.
4. Use the **Zoom Extents** command to see the entire yard.

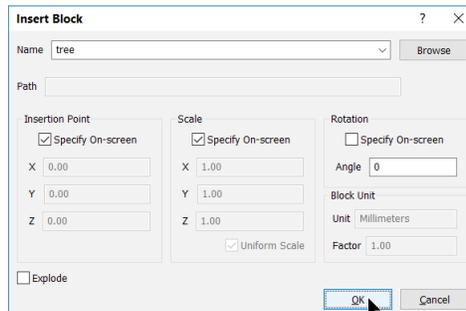
ADDING MANY MORE TREES

Command	Insert
Ribbon	Insert Blocks Insert Block
Menu Bar	Insert Block
Alias	I

With the bigger view in place, insert tree blocks in the drawing.

1. Start the **Insert** command:
:insert
 Notice that BricsCAD displays the Insert dialog box.
2. The **Tree** block name should be shown in the **Name** field.
 You can ignore most of the dialog box, other than to ensure that the options are set as follows:

Insert Options	Value
Insertion Point	Specify On-screen <input checked="" type="checkbox"/> (on)
Scale	Specify On-screen <input checked="" type="checkbox"/> (on)
Rotation	Specify On-screen <input type="checkbox"/> (off)
Angle: 0	

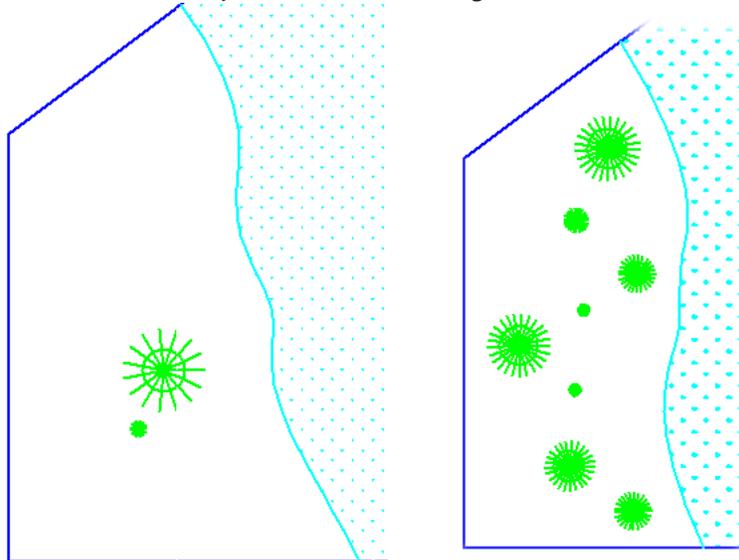


3. Click **OK**. BricsCAD now prompts you in the command bar, because of the Specify On-Screen options that were turned on in the dialog box.
 The first item of business is to pick a spot for the block, a suitable spot in the garden area.
 Insertion point for block: (Pick a point anywhere in the garden area.)

- The other item is the scale, which draws the block larger or smaller than the symbol original size:
 Corner/XYZ/X scale factor <1.000000>: 2

The x scale factor of 2 draws the block twice as large as the original. Because you drew the original symbol one meter in diameter, the newly inserted tree has a diameter of two meters. If you had entered a scale factor smaller than 1, such as 0.5, then the block would have been drawn half-size.

From this exercise, you can see that it makes sense to draw a symbol to *unit size* (to the nearest meter or foot), because that makes it easy to scale the block during insertion.



*Left: Two trees inserted at different scale factors
 Right: Many more tree blocks inserted into the drawing*

- Try a different way of repeating the **Insert** command. This time, right-click the drawing. Notice that the Insert command repeats, again displays the dialog box and prompts of the Insert command.
- Add several more trees around the garden area using the **Insert** command and different scale factors, such as 2, 1.2, 0.6, and 0.4 (or 6.0, 4.0, 2.0, and 1.5 in imperial units).

Use the **Mirror** command to double the number of trees.

Drawing the Pond

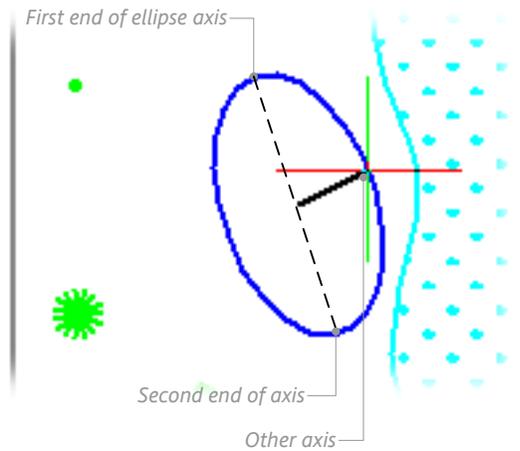
Command	Ellipse
Ribbon	Draw Draw Ellipse
Menu Bar	Draw Ellipse
Alias	EL
Command	Offset
Ribbon	Change Modify Offset
Menu Bar	Modify Offset
Alias	O

Drawing the garden pond illustrates another pair of useful commands: Ellipse draws oval shapes and Offset adds parallel copies. The oval will be the pond, the parallel offset the pond's edging.

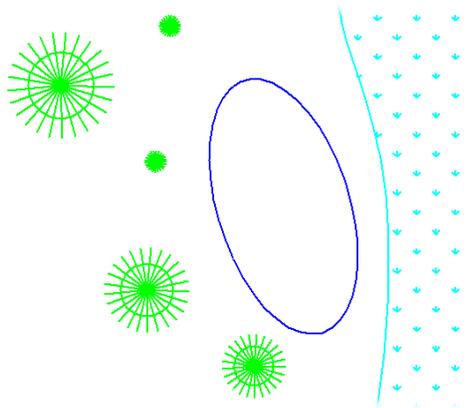
- Switch to the **Pond** layer via one of the layer droplists.
- The pond is to be an oval 4.5m tall by 1.5m wide (15 by 5 feet in imperial units). To draw the pond with the Ellipse command, follow these steps:
 : ellipse

Pick the starting point of the ellipse anywhere in the garden area.

Arc/Center/<First end of ellipse axis>: c
Center of ellipse: (Pick a point)



3. The other end of the pond is 4.5m away. Use direct distance entry by moving the cursor down and right:
Endpoint of axis: (Move the cursor) 4.5
4. And the pond is five feet wide:
Rotation/<Other axis>: 1.5

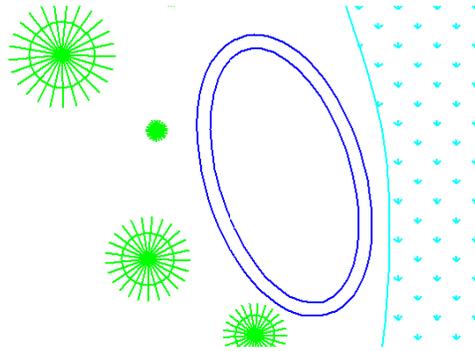


5. The pond is edged with rock. You could draw the edging of by repeating this command to draw a second, larger ellipse. The easier alternative is to use the **Offset** command, which creates parallel copies, precisely. Here, you use it to create a concentric ellipse:

: **offset**

- a. The offset distance is 0.3m (or one foot in imperial units):
Parallel: ENTER for Through point/<Distance>: 0.3
- b. Choose the ellipse; notice that only one entity can be offset at a time:
Select entity: (Pick the ellipse)
- c. Place the offset copy on the outside of the ellipse:
Both sides/<Side for parallel copy>: (Pick outside the ellipse)

- d. The command repeats itself to offset other entities. But to exit the command, press **Enter**.
 Select entity: (Press **ENTER**)

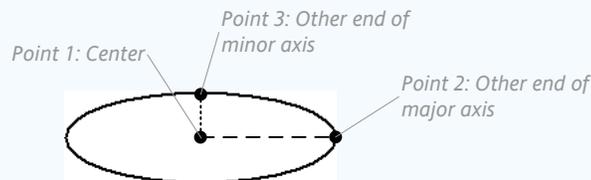


The Offset command also creates parallel lines, parallel polylines, and concentric circles and arcs.

QUICK SUMMARY OF DRAWING ELLIPSES

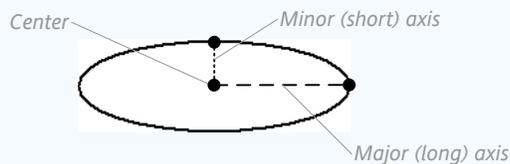
BricsCAD provides three methods for drawing ellipses (ovals):

: ellipse
 Arc/Center/<First end of ellipse axis>: (Enter an option.)



Center — specifies the ellipse's center point, and then the endpoints of the major and minor axes.

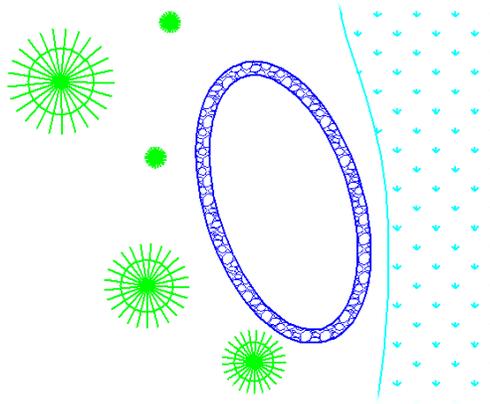
Axis, End — specifies the endpoints of the ellipse's major axis, and then the minor one.



Arc — draws elliptical arcs.

This command also draws isocircles, which are isometric circles. This option is available only when isometric drawing mode is turned on through the Snap or Settings commands.

If you wish, add the Gravel hatch pattern to the pond edging.



ARRAY PATHS

Another type of array lines objects along a path. This is called a “path array” appropriately enough. You use this construction technique to lay a path of paving stones across the lawn, from the house to near the pond.

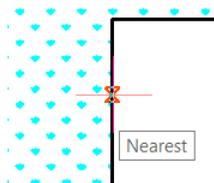
A path array needs a *guide path*, an entity along which to place the arrayed entities. In this part of the tutorial, you draw the guide path with a spline (Spline command), then array (PathArray command) the paving stones made of polygons (Polygon command), and finally edit the array (ArrayEdit command).

Drawing Splines

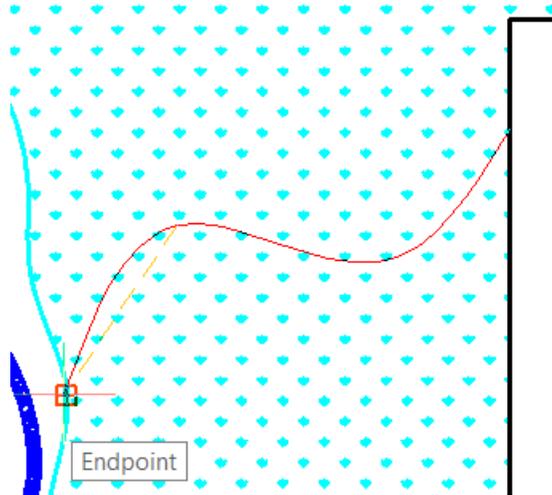
Command	Spline
Ribbon	Draw Polylines Spline
Menu Bar	Draw Spline
Alias	spl

To draw a spline with the **Spline** command, follow along with these steps:

1. First though, change the layer to “Road” using one of the methods you learned earlier. (My preferences is to use the “clayer road” shortcut shown below.) This means the paving stones are placed on the same layer as the driveway.
: **clayer**
New current value for CLAYER: **road**
2. Draw the guide path using a spline. *Splines* were first used in ship building for making hulls from smooth flowing strips of wood. If you ignore the options, then drawing a spline is as easy as drawing lines:
: **spline**
First point for spline: (*Pick a point on the house outline, as show below.*)



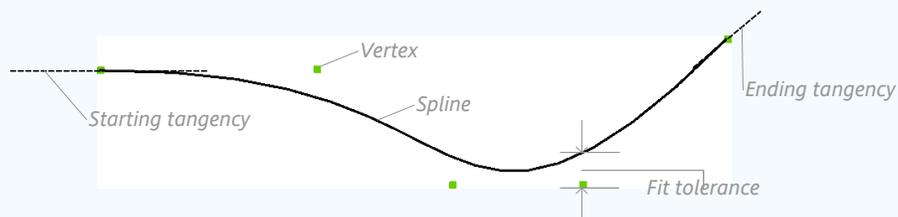
Second point (*Pick a point along the Lawn*)
 Close/Fit tolerance/<Next point>: (*Pick another point along the Lawn*)
 Close/Fit tolerance/<Next point>: (*Pick a point at the edge of the Lawn near the pond, as shown below*)



- To end the command, press **Enter** three times, once for each prompt:
 Close/Fit tolerance/<Next point>: (*Press Enter*)
 Starting point tangent point: (*Press Enter*)
 Enter tangent for ending point: (*Press Enter*)

QUICK SUMMARY OF SPLINE

```
: spline
First point for spline:
Second point:
Close/Fit tolerance/<Next point>:
Select starting tangent point:
Enter tangent for ending point:
```



First point for spline — specifies the start of the spline's tangency

Second point — locates the first vertex of the spline

Close — closes the spline, joining the start and points

Fit tolerance — specifies how closely the spline matches its vertices (also known as "fit points"); a value of 0 forces the spline to pass through the points you pick, and then the higher the number the further the spline is from those points.

Select starting tangent point — locates a tangency for the starting point; press Enter to ignore the option

Enter tangent for ending point — locates a tangency for the ending point; press Enter to ignore the option

TIP Another way to draw splines is with polylines. Use the **PLine** command to draw the polyline, and then use the **PEdit** command's **Spline** command to convert it to a spline. This can be an easier approach than using the **Spine** command directly.

Zooming to Objects

The **Zoom** command has a handy option that lets you zoom into a single entity. Here is how it works:

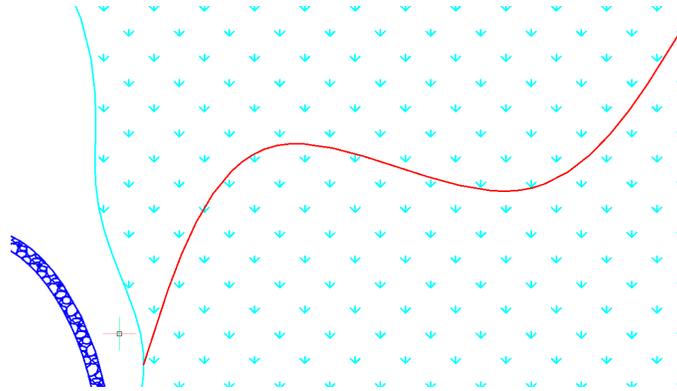
- To zoom into the spline, use the **Object** option of the **Zoom** command, as follows:

```
: zoom
Zoom: In/Out/.../Object/<Scale (nX/nXP)>: ob
Select an entity: 1
```

By entering 'L' at the select prompt, you ask the program to select the *last*-drawn entity, the spline. Using **Object** with **Last** makes it easy for you to pick the spline out of the grass pattern.

- Notice that the program zooms into the spline, so that it fills the screen. Zoom back out by 10% (i.e., 0.9x) so that you have some working space (see figure below):

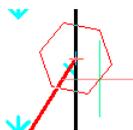
```
: zoom .9x
```



Paving Stones from Polygons

Command	Polygon
Ribbon	Draw Polygons
Menu Bar	Draw Polygon
Alias	pol

- The paving stones are hexagonal (six-sided) in shape, and so you draw them best with the **Polygon** command. This command, unfortunately, is an unintuitive one and so tricky to execute; follow along:
 - polygon**
 - The first thing you do is specify the number of sides. For a hexagon, you specify six sides:
Polygon: Multiple/Width of line/<Number of sides>: <4>: 6
 - Then you pick a center point for the polygon. For this tutorial, pick one end of the spline using **ENDpoint** entity snap:
Specify by: Edge/<Center of polygon>: end
Snap to endpoint of: (*Pick end of the spline near the house*)
 - Now you specify the size of the hexagon, which is sometimes not easy to do. For this tutorial, just drag the cursor away from the center point, as shown by the figure.
Specify by: Vertex/<Midpoint of side>: (*Move cursor away from center of polygon*)



Arraying Along a Path

Command	ArrayPath
Ribbon	Change Array Path Array
Menu Bar	Modify Array Path Array
Alias	...

With one paving stone (polygon) in place, you use the **ArrayPath** command to add all of the others along the path (spline). But first, a word of explanation.

ArrayPath is one of a trio of commands that create associative arrays; the other two are ArrayRect (for associative rectangular arrays) and ArrayPolar for polar ones. *Associative* means that all the elements of the array act as a single entity, yet can be individually edited — like an associative dimension (see chapter 7). When you change the path, the array changes to suit.

1. Start the **ArrayPath** command, and then select the polygon, as follows:

```
: arraypath  
Select entities to array: (Pick the polygon)
```

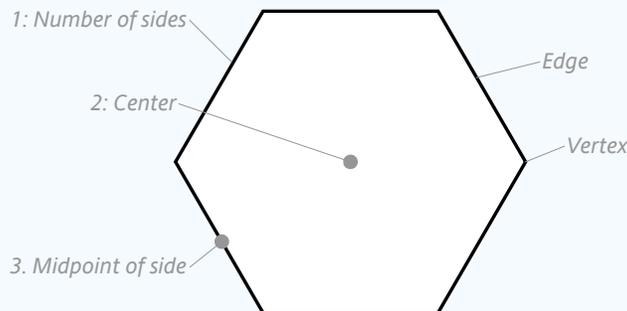
2. Press **Enter** to exit the selection process, and then pick the spline:

```
Entities in set: 1  
Select entities to array: (Press Enter)  
  
Type = Path, Associative = Yes  
Select the path curve: (Pick the spline)
```

QUICK SUMMARY OF POLYGON

This command draws regular polygons, which means that every side has the same length. (To draw irregular ones, use another command such as Rectang or PLine.) The minimum and maximum sides is 3 and 1024.

```
: polygon  
Polygon: Multiple/Width of line/<Number of sides> <4>:  
Specify by: Edge/<Center of polygon>:  
Specify by: Vertex/<Midpoint of side>:
```



Multiple — repeats the command to draw additional polygons

Width of line — specifies the width of the lines, because polygons are made of polylines

Number of sides — specifies the number of sides between 3 and 1024

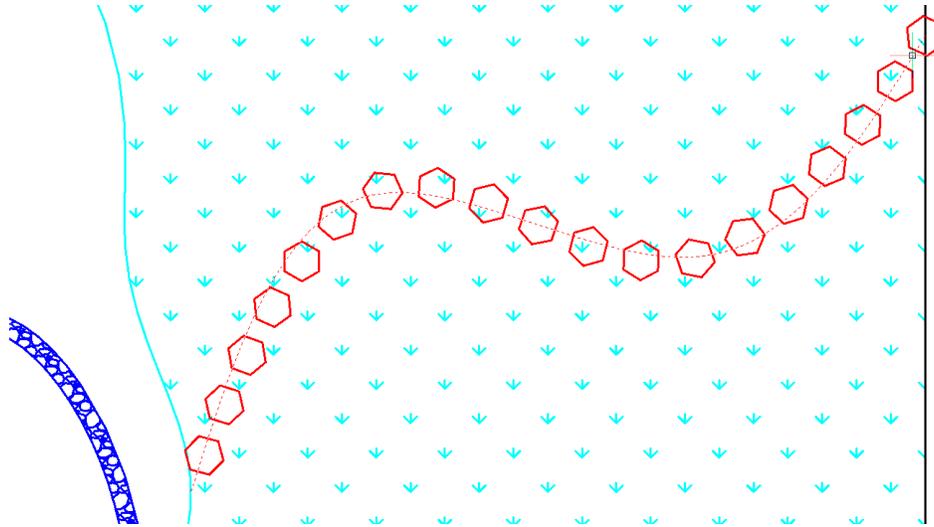
Edge — specifies the length of one side

Center of polygon — specifies the center point of the polygon

Vertex — specifies the location of a vertex (corner of the polygon)

Midpoint of side — specifies the midpoint of one of the sides

3. Now you see many paving stones along the path. Press **Enter** to exit the command.



4. Use the **Erase** command to remove the spline from the drawing.

QUICK SUMMARY OF ARRAYPATH

: arraypath

Select entities to array:

Select path curve: (ASsociative/Method/Base point/Tangent direction/Items/Rows/Levels/
Align items/Z direction/<eXit>:

Select path curve — you can use any linear (open) entity for the path, such as a line, polyline, arc, circle, ellipse, elliptical arc, spline, helix, or 3D polyline.

ASsociative — toggles the array between associative and non-associative

Method — determines how the entity is arrayed along the path:

Measure	Places entities at a specified distance apart
Divide	Places a specified number of entities along the path

Base point — specifies a new base point from which the entity is arrayed; useful for when you don't want the array on top of the path

Tangent direction — determines how the entity is aligned along the path in 3D, tangent or normal.

Items — specifies the (Measure mode) distance between items, or (Divide mode) number of items to place along the path; a maximum number of items can be specified to make the array shorter than the path.

Rows — specifies the number of rows for stacked arrays

Levels — specifies the number of levels and their spacing for 3D arrays

Align items — determines whether the entity is aligned along the path

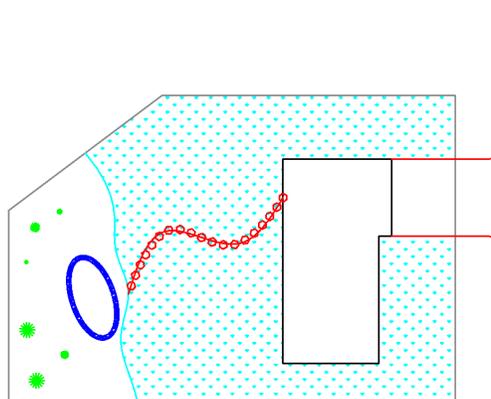
Z direction — toggles the z orientation between upright or banked along path

eXit — ends the command

To count the number of paving stones you need to order from the lumber store, select the array and then look at the **Items** field in the Properties pane: 19. Your number may differ, depending on how you drew the spline and the polygon stepping stones.

General	
3D Visualization	
Geometry	
Base	21.50, 16.00, 0.00
Path length	16.83 mm
Misc	
Method	Measure
Fill entire path	Yes
Items	19
Item spacing	0.90 mm
Start offset	0.00 mm
Rows	1
Row spacing	0.98 mm
Row elevation increment	0.00 mm
Levels	1
Level spacing	1.00 mm
Align items	Yes

To see the progress you are making in learning BricsCAD, plot your drawing with the **Print** command. At this point, your drawing should look similar to the following:



Remember to save your work with the **Save** command.

Summary

You've added a number of details to the drawing. In the next lesson, you learn how to make changes to them — how to edit entities.

Making Changes to Drawings

Previously, you added details to the drawing, such as the lawn, trees, and a pond. Here, you learn how to change parts of the drawing and how to extract information. This when CAD software shows itself to be powerful: effortlessly making changes and finding out the data held by drawings.

IN THIS CHAPTER

- Modifying properties of entities
- Applying linetypes
- Changing the length of open and closed entities
- Finding information about entities in drawings

KEY TERMS IN THIS CHAPTER

Extend — extends open entities to boundaries that are defined by other entities

Filter — creates specified subsets of entities based on their properties

Linetype — displays line patterns made of dashes, dots, gaps, and symbols

Properties — lists all the characteristics of an entity

Trim — cuts entities at cutting edges that are defined by other entities

USEFUL ABBREVIATIONS

ISO International Organization for Standardization

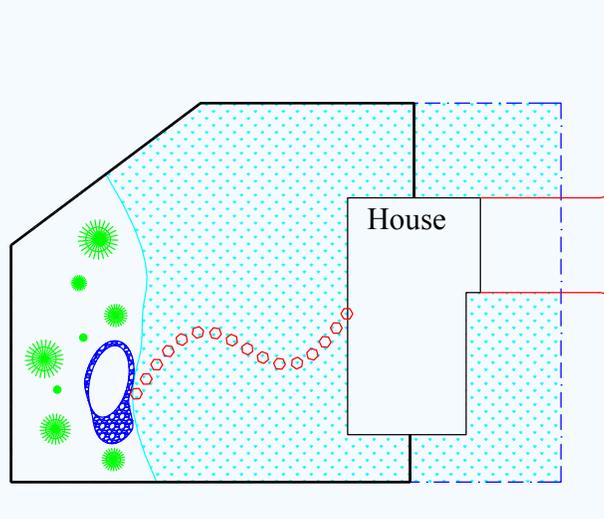
.lin Linetype definition file

NEW COMMANDS

Command	Aliases	Menu Selection	Ribbon Tab
Dist	di	Tools Inquiry Distance	Tools Inquiry Distance
Lengthen	len	Modify Lengthen	...
Linetype	lt	Settings Linetypes	...
List	li	Tools Inquiry List Entity Info	Tools Inquiry List
Properties	pr	Modify Properties	...
Stretch	s	Modify Stretch	Change Modify Stretch

THIS CHAPTER'S DRAWING

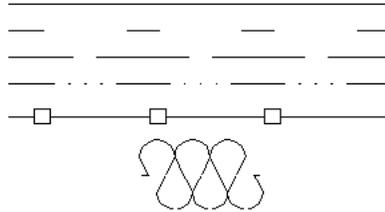
By the end of this lesson, your drawing will look like this one:



Tutorial-06 drawing file

Changing the Look of Lines

When you drew the lot lines in an earlier lesson, they showed on the screen as solid lines. Lot lines, however, are usually shown by a dashed pattern. Just as BricsCAD comes with several hatch patterns, it also includes a number of line patterns called “linetypes.” Here is what linetypes look like:



To change a line from looking solid (called “Continuous” by BricsCAD) to looking dashed, you load the linetype definition into the drawing, and then change the line(s) to the new type.

LOADING LINETYPES

Command	Linetype
Menu Bar	Tools Linetypes
Status Bar	Linetype
Alias	LT

Start BricsCAD, and then open your copy of *yard.dwg*. (If necessary, open the *tutorial-05.dwg* file or *Imp-tutorial-05.dwg* in imperial units. You can access them from my public Dropbox folder: <https://dl.dropboxusercontent.com/u/28941239/Inside-BricsCAD-Tutorial-Files.zip>.)

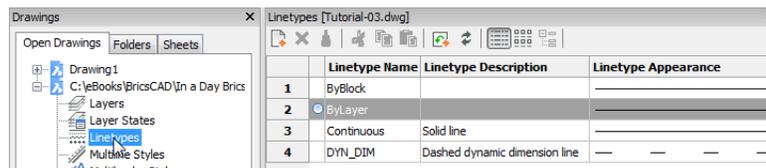
1. If necessary, use the **Zoom** command’s **All** option so that you see the entire drawing on the screen”
: **zoom**
Zoom: In/Out/... <Scale (nX/nXP)>: **all**

Follow this by an 80% zoom that adds breathing space around the drawing:

```
: zoom  
Zoom: In/Out/... <Scale (nX/nXP)>: 0.8x
```

2. Before you can apply linetypes, you must *load* their definitions from a file into the drawing. Linetype definitions are stored in files separately from BricsCAD and drawings. You recognize the file by the *.lin* found at the end of its name. Here is how to load linetypes into the drawing.
 - a. Enter the linetype command:
: **linetype**

Notice that BricsCAD displays the Drawing Explorer:

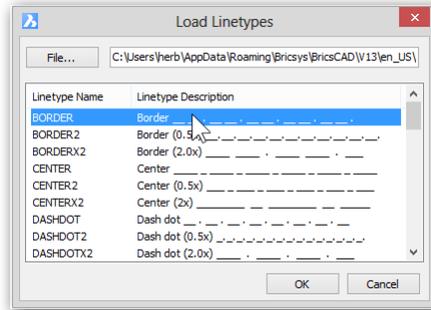


All new drawings initially contain some linetypes, and here is what they mean:

- > **ByLayer** — entities take the linetype assigned to the layer
- > **ByBlock** — entities take the linetype of their block
- > **Continuous** — lines are solid, with no gaps
- > **DYN_DIM** — lines have dashes; used by dynamic dimensions

TIP Although BricsCAD uses terms like *linetypes* and *line weights*, these apply to nearly any kind of entity, such as circles and arcs — not just to lines. They do not apply to text or 3D entities, however.

- To add a linetype to this drawing, click the  **New** button. BricsCAD doesn't actually create new linetypes, as the name of this button suggests. Instead, this button leads to the dialog box from which you load pre-defined linetypes. Notice the Load Linetypes dialog box.



The dialog box lists in alphabetical order the names of all linetype definitions available to you. (They are stored in the *default.lin* file.) Scroll through the list to see what is available. The names of linetypes use codes to identify some of them:

- > Linetype names postfixed with **X2** are double scale
- > Linetype names postfixed with **2** or **4** or **8** are half or quarter or eighth-size (0.5x, 0.25x, 0.125x scale)
- > Linetype names prefixed with ISO match the ISO (International Standards Organization) standard

- Load the Border linetype like this:
 - Select **BORDER**.
 - Click **OK** to close the dialog box.
- You are back in the Drawing Explorer dialog box. Set the Border linetype as *current*, like this:
 - Click **BORDER** to select it.
 - Under the **Current** column, click the blank square so that the  blue dot appears.
 - Click **X** to dismiss the dialog box.

	Current	Linetype Name
1	<input checked="" type="checkbox"/>	BORDER
2	<input type="checkbox"/>	ByBlock

If the Properties panel or Entity Properties toolbar are open, notice that the default linetype changes to Border.



CHANGING PROPERTIES

Command	Properties
Menu Bar	Modify Properties
Ribbon	...
Alias	PR

The Properties panel (or bar or palette) is handy for changing several properties of one or more entities. It provides three kinds of information:

- No Selection** — Properties panel describes the way an entity will look when drawn next, with the color, layer, linetype, and so on that are shown. The figure below shows that entities will be drawn with color ByLayer (red), on layer “Road,” and so on.

General	
Color	ByLayer
Layer	Road
Linetype	ByLayer
Linetype scale	1
Lineweight	ByLayer
Transparency	ByLayer
Elevation	0.00 mm
View	
Camera	0.00, 0.00, 1.00
Target	0.00, 0.00, 0.00
Perspective	Off
Lens length	50.00 mm
Field of view	39
Height	31.62 mm
Width	66.83 mm
Clipping	Off
Front plane	0.00 mm
Back plane	0.00 mm
Visual style	2dWireframe
Misc	
Annotation scale	1:1
Default lighting	On

- Entity selected** — properties describes the properties of a single entity that is selected. The General section is common to all entities. The Geometry section changes depending on the type of entity selected. In the figure at left below, you see that a line is selected and that the Properties panel shows the properties of the line.

General		General	
Color	ByLayer	Color	ByLayer
Layer	Lot	Layer	*Varies*
Linetype	ByLayer	Linetype	ByLayer
Linetype scale	1	Linetype scale	1
Plot style	ByColor	Plot style	ByColor
Lineweight	ByLayer	Lineweight	ByLayer
Transparency	ByLayer	Transparency	ByLayer
Hyperlink		Hyperlink	
Thickness	0.00 mm	Thickness	0.00 mm
3D Visualization		3D Visualization	
Material	ByLayer	Material	ByLayer
Geometry		Geometry	
Start point	12.00, 24.00, 0.00	Start point	*Varies*
X	12.00	X	*Varies*
Y	24.00	Y	*Varies*
Z	0.00	Z	0.00
End point	0.00, 15.00, 0.00	End point	*Varies*
X	0.00	X	*Varies*
Y	15.00	Y	*Varies*
Z	0.00	Z	0.00
Delta	-12.00, -9.00, 0.00	Delta	*Varies*
X	-12.00	X	*Varies*
Y	-9.00	Y	*Varies*
Z	0.00	Z	0.00
Length	15.00 mm	Length	149.00 mm
Angle	217	Angle	*Varies*

Left: Properties panel when one line is selected; right: More than one line selected

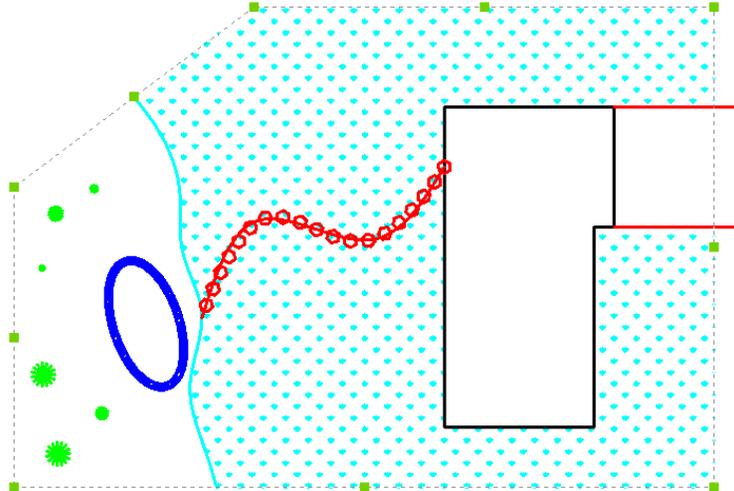
- Multiple entities** — when two or more entities are selected, the Properties panel reports only those properties that are in common; when properties differ between entities, the panel reports “*Varies*”, as shown above at right.

Use the Properties panel to change the lot lines from ‘Continuous’ to ‘Border’, like this:

- If the Properties panel is not displayed, enter the **Properties** command to open it now:
: properties

Notice that the Properties panel opens.

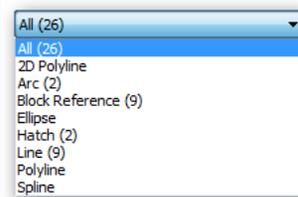
2. In the drawing, pick the five blue lot lines directly. As you pick them, notice that BricsCAD highlights them: they are shown as dashed lines, with green grip squares appearing on each line.



At the top of the Properties panel, notice that the list box reports **Line (5)**. It tells you that five lines were selected, and that the properties listed belong to the five. Because more than one line is selected, many properties are reported as **Varies**, because properties (such as the x coordinate) differs for each line.

QUICK SUMMARY OF PROPERTIES PANEL

Properties describe entities, such as their color, linetype, starting and ending coordinates, and thickness. When two or more entities are selected, then the Properties panel (bar or palette) can display a subset of entities. Click the droplist, and then choose a subset, such as two arcs. (NEW IN V17) BricsCAD changed the name from “bar” to “panel.”

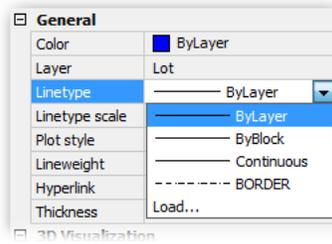


The Properties palette does more than display properties; it also lets you to *change* values of properties — usually. Properties marked **Varies** can also be changed; in this case, all selected entities take on the same property. If you do this by accident, use the U command to undo the change.

There are three ways to change property values:

- › Enter new values by typing them in
- › Select preset values from droplists (when available)
- › Click buttons (when available) to select values from drawings or dialog boxes

- Next to Linetype in the Properties panel, click **ByLayer**. Notice that BricsCAD lists the names of the linetypes in the drawing, ByLayer, ByBlock, Continuous, Dyn_Dim, and Border.



- Select the **Border** linetype by clicking its name.
- Press **Esc** to clear the grips.

Notice that the lines change their look and take on the dashed linetype. If, however, you see no change to the lines, then read the next section to learn how to fix this problem.

CHANGING THE LINETYPE SCALE

Command	LtScale
Menu Bar	...
Ribbon	...
Alias	LTS

The reason linetypes sometimes look continuous is that they are sensitive to scale, just like hatch patterns. When the lines should show gaps, but don't, then the solution is to change the scale (size) of the linetype.



Top: Incorrect linetype scale; above: Correct linetype scale.

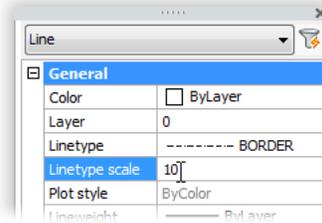
All linetypes start with a scale factor of 1.0, which usually is too small to display the pattern of dots and dashes correctly.

Warning! Setting linetype scale is one of the trickiest aspects of BricsCAD. Lintypes look continuous when the scale is too large — and when too small!

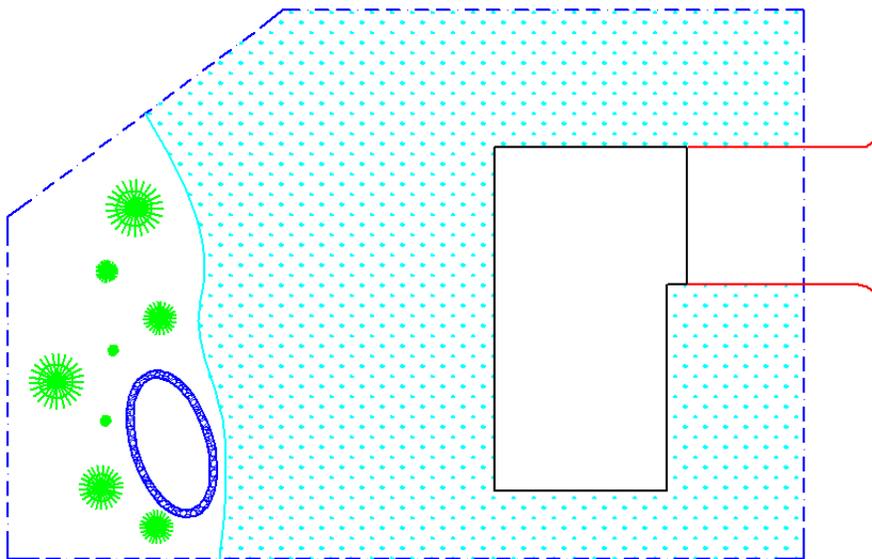
One method to change the scale of linetypes is to use **LtScale**; another is to use the Properties panel and its Linetype Scale field, as follows:

- Continuing in the Properties panel, click the field next to **Linetype Scale**.

- Change 1.0 to something like 10.



- Check whether the linetype pattern becomes visible. If not, try another scale value, such as 0.01.



- It may be necessary for you to use the **Regen** command to clean up the display.

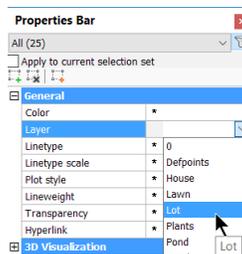
SELECTING ENTITIES BY OTHER METHODS

I find that it can be tricky trying to select entities in crowded drawings. No matter how tiny the pick box, sometimes I repeatedly pick the wrong entity. The program provides two workarounds that are useful to know.

Selecting Entities by Their Properties

In the previous tutorial I had you pick the five lot lines one by one. I did that on purpose so that I could show you easier ways! You can select entities instead by their properties, as follows:

- Press **Esc** to ensure no entities are selected.
- In the Properties pane, click the  **Quick Select** button. Notice that the pane changes its look:



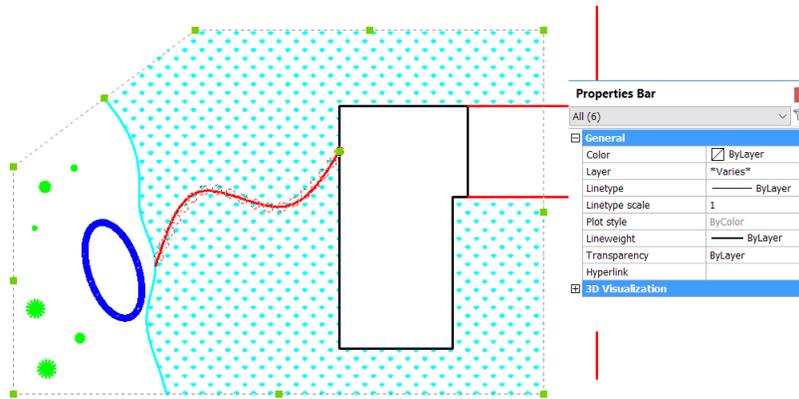
Notice the following about the new look:

- > All entities are selected, as shown by “All (25)”
- > Three buttons populate the toolbar
- > Basic properties show ‘*’ (asterisk), which means that all properties are selected.



When using Quick Select, you are modifying filters that apply to the current selection set, which starts off as all entities in the drawing. A *filter* changes the selection set by properties. For instance, when you select “Red” as the **Color**, you filter out all entities that are not red.

3. In the **Layer** droplist, choose “Lot.”
4. Click the **Add to Selection Set** button. Notice that only the lot lines are highlighted. You can now change other properties of the lot lines, such as the linetype or color.



TIP The * is not the only wildcard used by Quick Select. To see the full list, click the asterisk to uncover this shortcut menu. (That’s left click, not right click.)

- * Select All
- = Equals
- ≠ Not Equal
- < Smaller
- ≤ Smaller Or Equal
- > Greater
- ≥ Greater or Equal

This lets you, for example, select all lines that have a Length property of less than 10m.

Selecting an Entity by Tabbing

When two or more entities overlap, you can press the **Tab** key to force the program to cycle through them. This trick works only when the **SelectionPreview** variable is turned on.

CHANGING LINE LENGTHS

Command Lengthen
Menu Bar Modify | Lengthen

Once you've drawn stuff in a drawing, they aren't static; you can change their size and position. When it comes to changing size, it matters if the entities are open or closed. *Open* entities are like lines, arcs, and open polylines; *closed* ones are like circles, ellipses, and polygons. Some commands work only with open ones, some with only closed, and some with both kinds.

One way to change lengths is using grips, as described earlier. A more precise method is with the **Lengthen** command. To see how it works, in this tutorial you extend the edge of the street line to the bottom of the screen. If necessary, first zoom in on the area around the pond with the Zoom Window command.

1. Start the **Lengthen** command:
 : **lengthen**
2. At the 'Select entity to list length' prompt, pick the lower road line:
 Edit length: DYNamic/Increment/Percent/Total/<Select entity to list length>: (*Pick the Line.*)

QUICK SUMMARY OF THE LENGTHEN COMMAND

The **Lengthen** command provides you with the following options for changing the length (longer or shorter) of open entities, such as lines, arcs, and polylines:

```
: lengthen  
Edit length: DYNamic/Increment/Percent/Total/<Select entity to list length>:
```

DELta — changes the length to an absolute amount; you show the delta by picking a point the required distance from the endpoint; default = 0. (*Delta* is the Greek word used by mathematicians to indicate change.)

DYNamic — changes the length by interactive dragging.

PERcent — changes the length relative to 100%, as follows:

- ▶ Less than 100%, such as 50%, shortens the entity.
- ▶ More than 100%, such as 200%, lengthens the entity.

TOTAL — changes the length to an absolute amount; default = 1 unit.

UNdo — undoes the last change.

As is common in BricsCAD, there is more than one way to change the length of lines and other open entities. You can also use grips and these commands:

Change command — changes the length of a group of lines to a common endpoint.

Extend command — extends a line to a boundary entity.

Trim command — cuts back a line to a cutting edge entity.

Notice that BricsCAD reports its length in the command bar:

```
Current length: 11.00
```

This means that the line is 11m long (about 37', imperial units).

3. Lengthen the road to 12m by specify the **Total** option, and then entering **12** as the new length, as follows:
Edit length: DYnamic/Increment/Percent/Total/<Select entity to list length>: **t**
Angle/<Enter total length (11.00)>: **12**
4. Curiously, BricsCAD asks you a second time to select the entity; for some reason, it does not automatically change the length of the line you choose first:
Mode/<Select entity to change>: (*Pick the same Line*)
5. Notice that the line lengthens. Press **Enter** to end the command.
Mode/Undo/<Select entity to change>: (*Press ENTER*)

Stretching the Pond

Command	Stretch
Ribbon	Change Modify Stretch
Menu Bar	Modify Stretch
Alias	S

You've used several editing commands to change entities, such as PEdit to modify polylines, grips editing, Properties to change linetype and scale, and Lengthen to alter the length of lines.

One of BricsCAD's most powerful editing commands is **Stretch**. This command lets you take parts of entities and stretch them wider and thinner, longer and shorter.

Here you apply it to change the shape of the pond. You may find it helpful to first zoom in to the area around the pond.

1. Start the **Stretch** command.
: **stretch**
Select entities to stretch by crossing-window or crossing-polygon: **C**

The Stretch command has an odd limitation, in that entities are initially selected only by two methods, both of them *crossing* selection modes. You can specify either Crossing (which forms a rectangular selection window) or CPolygon, which forms a polygonal selection window. The reason for the limitation is that it allows Stretch to know which entities to stretch — those crossing the selection window — and which to not stretch (those outside the selection window). Entities fully inside the selection window are moved.

TIP Some entities cannot be stretched, like text and 3D solids. When they cross the selection window, the Stretch command moves them.

QUICK SUMMARY OF SELECTION SET OPTIONS

Selection has many options for selecting entities, as shown in the following table. I find that I typically use just some of them, such as pick, W, C, L, P and Enter. To see the full list in BricsCAD, enter the **Select** command, and then press **?**, like this:

```
: select
Select entities to include in set: ?
Select entities: ALL/Add+/Remove-/Previous/Last/Window/Crossing/Outside/WPolygon/CPolygon/OPolygon/WCircle/CCircle/OCircle/Box/POint/Fence/AUto/Multiple/Single/PROperties/Dialog/Undo:
```

Mode	Abbreviation	Selects
Entity	<i>[pick]</i>	A single entity
All	ALL	All non-frozen entities; can also press Ctrl+A
Previous	P	Most recently selected entity
Last	L	Most-recently created entity visible on the screen
Window	W	All entities inside a rectangular window
Crossing	C	All entities crossing and within a rectangular window
Outside	O	All entities outside the rectangular window
Window Polygon	WP	All entities within a polygonal window
Crossing Polygon	CP	All entities crossing and within a polygon
Outside Polygon	OP	All entities outside a polygonal window
Window Circle	WC	All entities inside a circular window
Crossing Circle	CC	All entities crossing and inside a circular window
Outside Circle	OC	All entities outside a circular window
Box	B	All entities depending on how direction selection window is drawn: <ul style="list-style-type: none"> > Right to left: selects by Crossing mode > Left to right: selects by Window mode
Fence	F	All entities along a fence polyline
Auto	AU	Single pick selects one entity; otherwise acts like Box option
Single	S	Selects the first entity encountered, and ends the command

SELECTION MODIFICATIONS

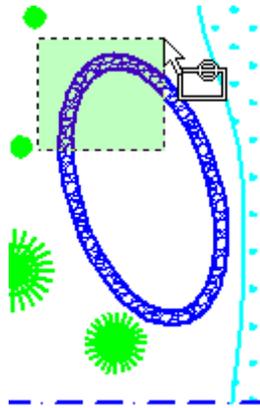
Add <i>or</i> +	A <i>or</i> +	Enters add-entities mode
Remove <i>or</i> -	R <i>or</i> -	Enters remove-entities mode
Multiple	M	Select entities without highlighting them; faster in complex drawings
Undo	U	Removes most recent selection group
Dialog	D	Displays the Entity Selection section of the Settings dialog box
End	<i>press</i> [Enter]	Ends entity selection
Cancel	<i>press</i> [Esc]	Cancels entity selection

SELECTION BY PROPERTIES

The **PROperties** option selects entities by common properties:

Properties	Abbreviation	Selects
Color	C	All entities of the same color name or number such as "red" or "122"
Layer	LA	All entities with the same layer name
Ltype	LT	All entities of the same linetype
Name	N	All entities of the same name
Thickness	TH	All entities of the same thickness
Type	TY	All entities of the same entity type, such as "circle"
Value	V	All entities of the specified handle
Width	W	All entities of the specified width
Location	LO	Switches back to original select prompt

2. Select entities by picking two points to specify a crossing window, as follows:
 First corner of crossing window: *(Pick a point)*
 Opposite Corner: *(Pick another point)*

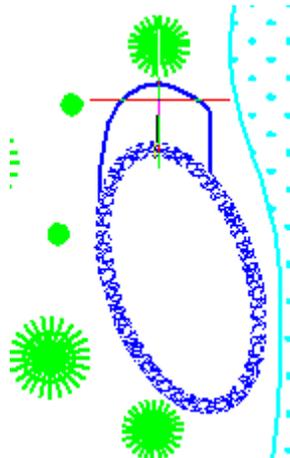


C is short for “crossing,” an entity selection mode similar to the window mode you used earlier with the Zoom command. In this case, BricsCAD selects all entities within the selection rectangle and all entities crossing or touching the rectangle.

3. After picking two corners of a rectangle that covers part of the pond, press **Enter**:
 Select entities to stretch by crossing-window or crossing-polygon: *(Press ENTER to end entity selection)*

TIP If the pond were entirely inside the entity selection rectangle, the Stretch command would **move** the pond, not stretch it. If you accidentally select entities other than the pond, type the R option (short for “remove from selection set”), and then select the entities to remove.

4. To tell BricsCAD how much you want the pond stretched, pick two points that indicate the distance:
 Base point of displacement: *(Pick a point near the pond)*
 Second point of displacement: *(Pick a point away from the pond)*



You have now created a whole new look to your pond! (Your pond will probably look different from the one illustrated in this book.) Note that you cannot create this effect using grips editing.

5. If you don’t like it, you can undo the stretch with the U command, as follows:
 : **u**
 U: STRETCH

... and try stretching the pond again.

MOVING ENTITIES

Command	Move
Ribbon	Change Modify Move
Menu Bar	Modify Move
Alias	M

If the pond isn't exactly where you want it, you can relocate it with the **Move** command.

1. Start the Move command by entering the letter 'm', as follows:

: m

Here you used another of BricsCAD's shortcuts. **M** is the *alias* (abbreviation) for the Move command. (Appendix A lists all command aliases that you can use in BricsCAD.)

2. Continue the Move command by specifying Crossing selection mode to capture all three parts of the pond: two ellipses and hatch pattern.

Select entities to move: c

First corner of crossing window: *(Pick a point near the pond)*

Opposite Corner: *(Pick another point to encompass the pond)*

Select entities: *(Press ENTER to end entity selection)*

3. Just as with the Stretch command, you need to specify two points. They indicate the direction and distance to move the entities:

Vector/<Base point>: *(Pick edge of pond.)*

Displacement point: *(Pick new location for pond.)*

4. Notice that the pond has moved. If necessary, clean up the screen with the **Redraw** command, using its alias, as follows:

: r

5. Save your work.

TIP BricsCAD lets you specify commands by typing just one or two letters at the ':' prompt. Shortcut keystrokes are listed at the start of each chapter. The complete list of command name abbreviations (called "aliases") is stored in the *default.pgp* and is provided in Appendix A.

Adding the Fence

Say you've decided to add a fence to the backyard. You can use BricsCAD to help plan the materials needed: after drawing the fence (as a polyline), you can find out from the drawing the length of the fence.

The fence is drawn as a polyline, 0.1m-wide (4"-wide). Along the way, you'll use a variety of entity snap modes and tracking mode.

1. First, switch the working layer to **House** by selecting its name from the toolbar. Use the **Zoom Previous** command to see the full drawing.
2. If you turned off the **INTERsection** entity snap, turn it back on using this method: at the command prompt, enter **Intersection** like a command.

: **intersection**

3. Begin by entering the PLine command:

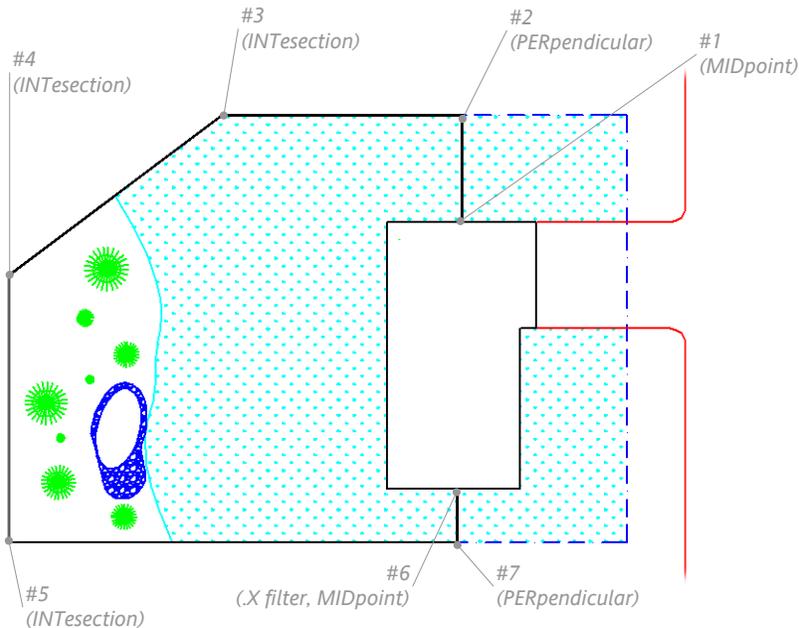
: **pline**

4. Start the polyline at the middle of the upper house line. This point is shown as #1 in the figure below). Use the **MIDDLE** entity snap to precisely locate the polyline's starting point at the middle of the house line. "Mid" is

the abbreviation for MIDDLE entity snap.

```
:_pline  
Start of polyline: Arc/Distance/Halfwidth/Width/<Next point>: mid  
Snap to midpoint of: (Pick upper house Line.)  
Current line-width is 0
```

- To change the width of the polyline from zero to 10cm (4"), use the PLine command's **w** (Width) option, as follows:
Start of polyline: Arc/Distance/Halfwidth/Width/<Next point>: **w**
Starting width <0>: **0.1**
Ending width <0.1>: (Press **Enter** to accept the new default)



Note that you can specify different starting and ending widths, which would produce tapered polylines.

- Now that the starting point and width are set, continue drawing the fence. Follow the path shown by numbers 2 through 5 in the figure above.
- Pause the picking action at point #5. When you get to the bottom of the house, you get into a bit of tricky geometry. You want the fence to end at the same relative location as its starting point. You're not sure of the x coordinate, which is located somewhere along the bottom line of the lot. Fortunately, you can find that point with *point filters*.

Normally, when you pick a point, you are providing BricsCAD with the x and y coordinate. Point filters capture a single coordinate, such as just the x or the y.

- Here is how to get just the x coordinate:
Arc/Distance/Follow/Halfwidth/Width/<Next point>: **.x**
- BricsCAD asks you to pick a point ('Select X of' prompt), from which it obtains that x coordinate. Tell it that you want the middle of the line:
Select X of: **mid**
Snap to midpoint of: (Pick the Lower Line of the house at #6)

Notice that you can combine point filters and entity snaps to get very specific coordinates!

- And then BricsCAD asks you for the y coordinate. When it asks 'Still need YX,' BricsCAD is asking for the y and z coordinates. Well, the z you don't care about, but here is how to get the y with the perpendicular entity snap:

```
Still need YZ of: per  
Snap to perpendicular of: (Pick the property Line at #7)
```

You completed the fence by drawing the last polyline segment using **PERpendicular** entity snap at #7.

- Exit the **PLine** command:
...<Next point>: (Press **ENTER** to exit the command)

LISTING DATA

Command List
Ribbon Tools | Inquiry | List
Menu Bar Tools | Inquiry | List Entity Info
Alias LI

Command Dist
Ribbon Tools | Inquiry | Distance
Menu Bar Tools | Inquiry | Distance
Alias DI

Now that you've drawn the fence, you can use the **List** command to tell you its length.

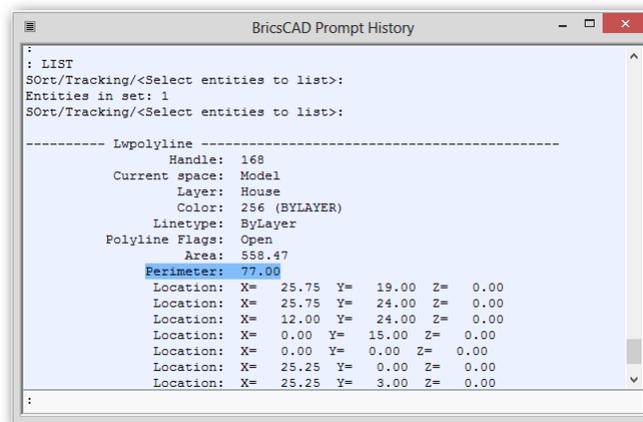
1. Enter the List command:

```
: list  
SOrt/Tracking/<Select entities to list>: (Pick the fence's polyLine)  
SOrt/Tracking/<Select entities to list>: (Press Enter to end entity selection)
```

BricsCAD flips to the Prompt History and lists lines of information. The List command tells you every piece of information about the polyline (technically known as a "lwpolyline") that BricsCAD has stored in its database. Most of the information is about its vertices ("at point").

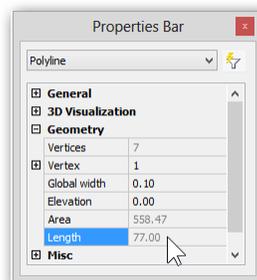
The total length of the polyline is shown in the middle of the listing:

Perimeter 77.0



So, 77 meters (roughly 260 feet). Now you know how much fencing you would need.

2. Press function key F2 to flip back to the graphics window. The Properties panel also reports area and length information:



3. You can measure distances directly on the drawing with the **Dist** command (short for “distance”). To find the shortest distance from the house to the pond with the Dist command, from the **Tools** menu bar, select **Inquiry | Distance:**

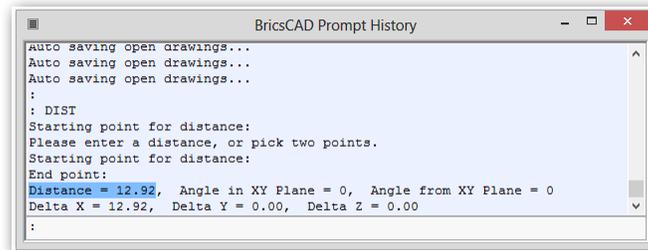
: _dist

Starting point for distance: **nea**

Snap to nearest of: (*Pick inside edge of pond.*)

End point: **per**

Snap to perpendicular of: (*Pick house wall.*)



The beeline distance from house to pond is 13m (around 43 feet). The value on your drawing may differ, depending on where you located the pond.

Summary

You’ve done a lot of editing, so remember to save the work you have done on the drawing. You may also want to plot out the drawing.

Next, you learn how to add text and dimensions to the drawing.

Adding Notes and Dimensions

With the yard plan nearly finished, now is the time to add *callouts* (or text) to describe the parts of the yard. Here you learn how to add these finishing touches to the drawing.

IN THIS CHAPTER

- Understanding drawing scales
- Placing text in drawings
- Defining styles and justification modes for text
- Learning about annotative scaling
- Editing text
- Toggling text and quick text
- Attaching a variety of dimensions to entities in the drawing

KEY TERMS IN THIS CHAPTER

Anotative — refers to entities that scale themselves to match the layout's scale factor

Associative — refers to entities, such as dimensions and hatches, that update automatically when associated entities are updated

Bounding box — describes the invisible rectangle that determines the word wrap of paragraph text

Callouts — describes the text attached to lines that describe significant parts of drawings

Dimension — indicates a measured distance, usually consisting of lines, arrowheads, and text

Justification — determines the relative positioning of text, such as right, left, and centered

Layout — presents editable print previews

USEFUL ABBREVIATIONS

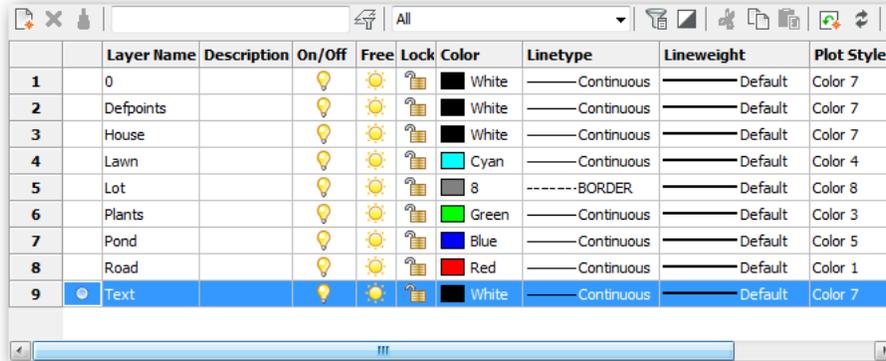
dims	Dimensions
dimvars	Dimension variables
dimstyle	Dimension style

NEW COMMANDS

Command	Shortcuts	Menu Bar	Ribbon Tab
DdEdit	ed	...	Draw Dimensions Edit
DimAligned	dal, dimali	Dimension Aligned	Draw Dimensions Aligned
DimBaseline	dba, dimbase	Dimension Baseline	Draw Dimensions Baseline
DimContinue	dco, dimcont	Dimension Continue	Draw Dimensions Continue
DimEdit	ded, dimed	Modify Edit Dimension Text	...
DimLinear	dli, dimlin	Dimension Linear	Draw Dimensions Linear
DimRadius	dra, dimrad	Dimension Radius	Draw Dimensions Circle
MText	t	Draw Multiline Text	Draw Text Multiline Text
QText	qt
Rotate	ro	Modify Rotate	Change Arrange 2D Rotate
Style	st	Settings Text Style	Home Drawing Explorer
Text	tx	Draw Text	Draw Text Text

Before starting the exercise, take these steps:

1. Start BricsCAD, and then open  the `yard.dwg` file. (If you were unable to complete the exercises of previous lessons, open the `tutorial-06.dwg` file; for imperial units, open the `Imp-tutorial-06.dwg` file. You can access them from my public Dropbox folder: <https://dl.dropboxusercontent.com/u/28941239/Inside-BricsCAD-Tutorial-Files.zip>.)
2. Do a **Zoom All**  to make the full drawing visible.
3. With the **Layer**  command, create a new  layer, and then name it **Text**.
4. Assign color “White” to the layer, and then make the layer current by clicking in the **Current** column, so that the blue dot appears next to the name, “Text.”

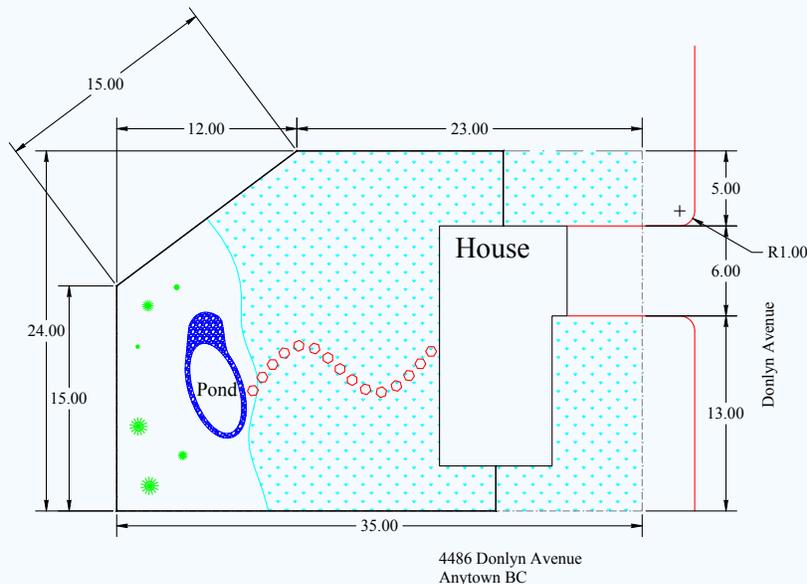


	Layer Name	Description	On/Off	Free	Lock	Color	Linetype	Lineweight	Plot Style
1	0					White	Continuous	Default	Color 7
2	Defpoints					White	Continuous	Default	Color 7
3	House					White	Continuous	Default	Color 7
4	Lawn					Cyan	Continuous	Default	Color 4
5	Lot					8	BORDER	Default	Color 8
6	Plants					Green	Continuous	Default	Color 3
7	Pond					Blue	Continuous	Default	Color 5
8	Road					Red	Continuous	Default	Color 1
9	Text					White	Continuous	Default	Color 7

You are now ready to tackle text in drawings.

THIS CHAPTER'S DRAWING

By the end of this lesson, your drawing will look like this one:

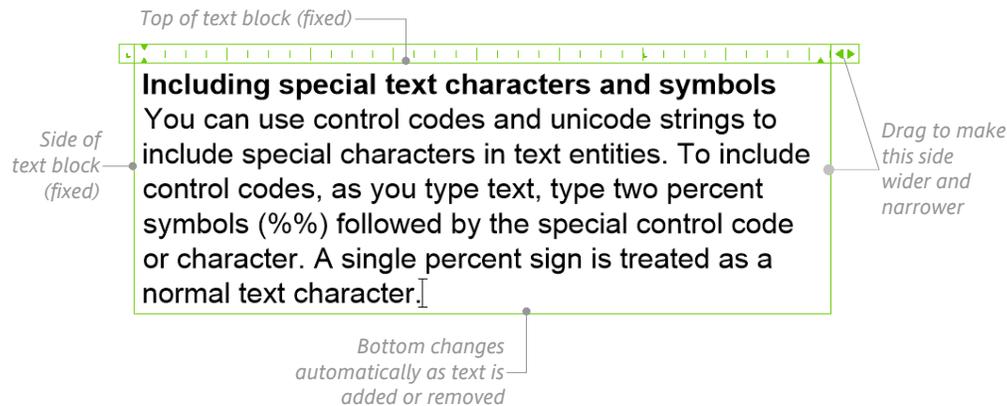


Adding Notes to Drawings

Command	MText
Ribbon	Draw Text Multiline Text
Menu Bar	Draw Multiline Text
Alias	T

You add callouts with the **MText** command, short for “multiple-line text” and also known as *paragraph text*.

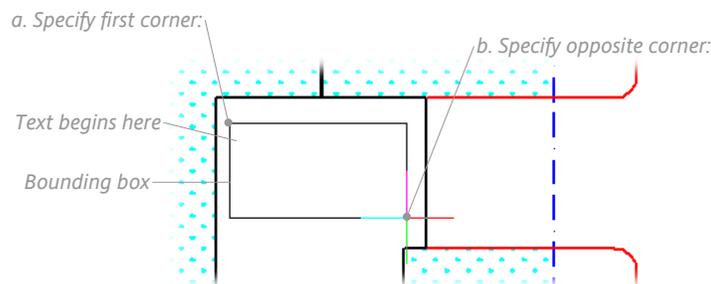
Mtext constrains text within a rectangular block. The top and two sides of paragraphs are constrained, but not the bottom; as you add text, it flows down. You can, however, drag the sides to make the block wider and thinner (see figure below); this is handy for ensuring the text fits a particular spot, such as notes or in tables. You can move the block of mtext to any location in the drawing.



You can format individual words and letters of the text with a variety of fonts and colors, as well as effects such as **boldfacing** and *italics*. Special symbols can be included, such as the center line symbol and the degree symbols. You can define how far in margins are indented and where tabs are set. All these features make mtext very flexible, but somewhat complex. (Later you meet the simpler but less capable, Text command.)

Place some mtext in the drawing now, like this:

1. Start the **MText** command:
: mtext
2. The first thing it wants is a starting point; pick it inside the house, as shown below:
Multiline Text: First corner for block of text: (Pick a point within the house outline)



3. Drag the cursor, and as you do notice that a rectangle is being formed. Pick a second point to define the width of the text:

Specify opposite corner or [Height/Justify/Line spacing/Rotation/Style/Width]: (Pick a second point)

The rectangle forms a *boundary box* that constrains the text. The box is visible only when the MText command is active; otherwise, it is neither displayed nor printed.

BricsCAD will start the first line of text at the top of the box. The right edge of the box determines where the text *wraps* — flows onto the following lines. (For right-to-left text like Hebrew, the *left* edge determines the word wrap location.) This is just like setting margins with word processing software.

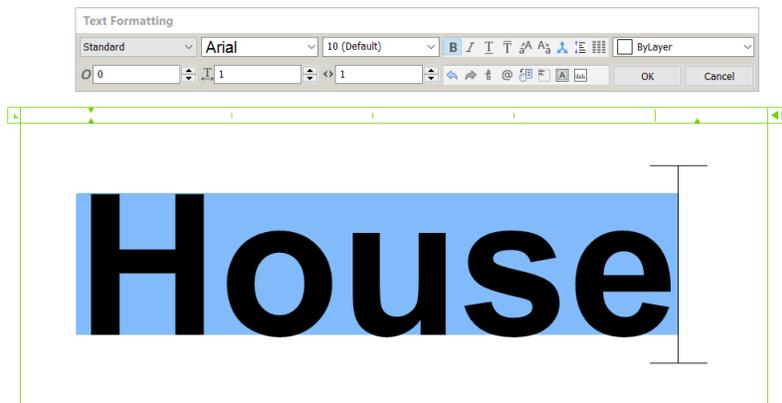
The bottom of the boundary is ignored. When there is too much text for the box, BricsCAD automatically extends it downwards. The bottom line is used only when the direction of the text flows upwards.

DETERMINING THE SIZE OF TEXT

Now you need to pause to do some multiplication and division, because you have to work out the height of the text. The drawing standard for text height is 10mm (or 3/8" in imperial units). But if text were placed at that size, it would be nearly invisible; after all, a few millimeters is tiny next to a 15-meter house! You need the text to be big enough to be legible, and so it has to be larger, much larger.

To figure out how tall to make the text, read the text box, *Quick Summary on Calculating Text Heights*. Based on the size of this drawing, the text height needs to be of 400mm tall (or 18" in imperial units). Continue with the MText command to specify the text height and other font parameters:

4. After picking the second point, BricsCAD displays the Text Formatting dialog box, in which you specify font settings and enter the text. First, change the height, as follows:
 - a. In the Font droplist, ensure “Arial” is specified.
 - b. In the **Height** droplist, highlight 0.2, and then enter **0.4**.
 - c. Press **Enter**.



QUICK SUMMARY ON CALCULATING TEXT HEIGHTS

Here are the steps to calculating the height of text appropriate for any size of drawing.

IN METRIC UNITS

Step 1: Work out the related drawing and paper widths, which will specify the scale factor:

$$\frac{\text{Width of drawing}}{\text{Width of paper}} = \frac{40\text{m}}{0.297\text{m}}$$

Step 2: Determine the scale factor by dividing the two values:

$$\frac{40}{0.297} = 134$$

The scale factor is 134. That means text in the drawing should be about 140 times taller to plot big enough to be legible on the page. The standard for text heights in drawing is 3mm (0.003m), and for title text is 10mm.

Step 3: Multiply the standard text size by the scale factor:

$$0.003\text{m} \times 134 = 0.4\text{m}$$

In BricsCAD's MText and Text commands, specify a text height of 0.4m.

IN IMPERIAL UNITS

Step 1: Work out the related drawing and paper widths, which will specify the scale factor:

$$\frac{\text{Width of drawing}}{\text{Width of paper}} = \frac{130 \text{ ft.}}{11 \text{ in.}}$$

Step 2: To make the units consistent, convert the feet to inches:

$$\frac{130 \text{ ft} \times 12 \text{ in./ft}}{11 \text{ in.}} = \frac{1,560 \text{ in.}}{11 \text{ in.}}$$

Step 3: Determine the scale factor by dividing:

$$\frac{1,560 \text{ in.}}{11 \text{ in.}} = 142$$

The scale factor is 142. That means text in the drawing should be about 140 times taller to plot big enough to be legible on the page. The standard for text heights in drawing is 1/8" (0.125"), and for title text 3/8" (0.375").

Step 4: Multiply the standard text size by the scale factor:

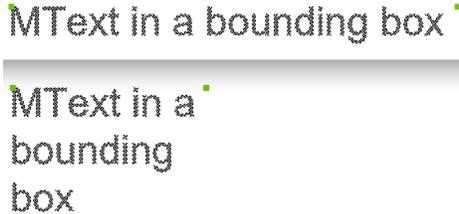
$$0.125" \times 142 = 18"$$

In BricsCAD's MText and Text commands, specify a text height of 18".

The reason for the difference between the metric scale factor of 134 and the imperial scale factor of 142 is that metric A4 paper is 18mm (0.7") wider than imperial A-size paper. See <http://www.papersizes.org/a-paper-sizes.htm>.

5. Type **House**, and then press **Enter**. Notice that the text appears in the drawing.
6. Click **OK** to exit the MText command.

- TIP** You can change the width of the mtext boundary box using grips. Here is how to do this:
1. Select the text. Notice the two handles (green grips), one at each corner of the top of the boundary box.
 2. Select a grip; notice it turns red.
 3. Drag a grip wider and narrower. Notice that the word wrap (paragraph lengths) change.



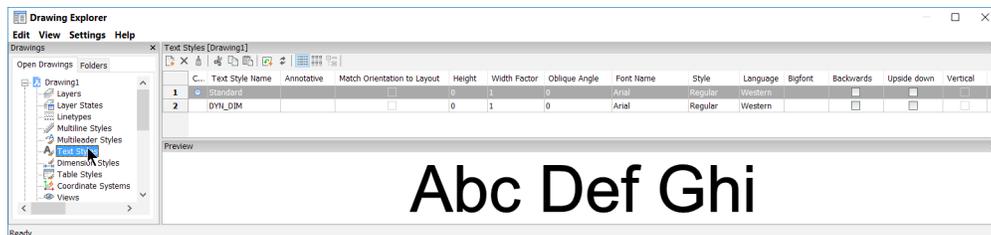
CREATING TEXT STYLES

Command	Style
Ribbon	Home Drawing Explorer Text Styles
Menu Bar	Settings Text Styles
Alias	ST

You can change the font used with text, but in CAD this is done indirectly with *styles*. “Styles” determine the look of the text, including which font should be used — just like in a word processor. The difference is that in CAD, styles must be used, where as in word processors they are optional.

Just as linetypes must be loaded into BricsCAD drawings, text fonts must be loaded before they can be assigned to styles. This is done through the **Style** command.

1. Enter the **Style** command. Notice that the Drawing Explorer dialog box appears on the screen.
: style

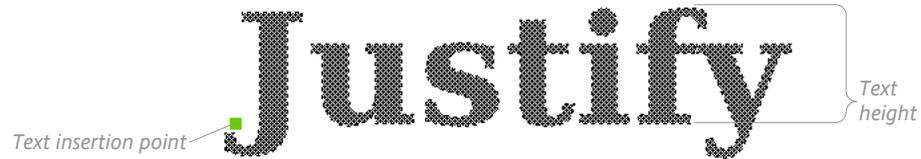


2. Make the following changes to the Standard style:

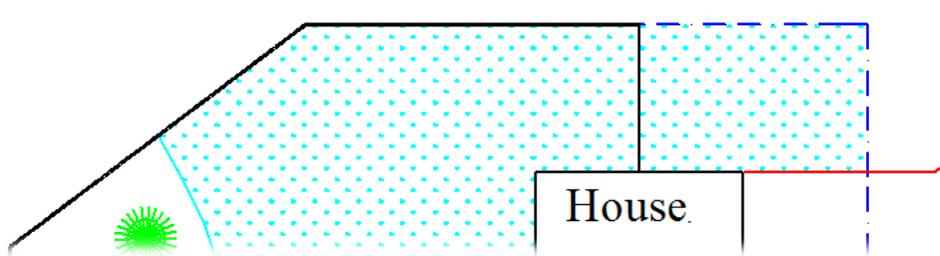
Property	Old Value	New Value
Font Name	Arial	Times New Roman
Height	0	.75

(For imperial drawings, enter **26** as the height.) Ignore the other settings, as you don’t need the text to be backwards, upside-down, or vertical — at least for now.

TIP The height of **o** has a special meaning: it means that the text height is not predefined and so must be specified during the Text command.



3. Click **X** to close the dialog box. Notice that the House text changes immediately to the new font:



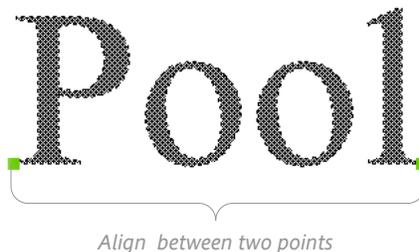
From now on, all text you place uses the Times New Roman font and is 0.75m tall (imperial units: 26 inches tall) — text that is half as tall as before.

SIMPLE TEXT

Command	Text
Menu Bar	Draw Text
Ribbon	...
Alias	TX

BricsCAD has a second command for placing text: **Text** places single lines of text. In some ways, it is easier than MText. To try it out, first zoom in on the pond.

1. Now that you see the pond area more clearly, start the **Text** command with **Draw | Text | Text**.
: _text
2. Select the **Align** justification mode, which places the text fitted between two points that you pick:
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: a
Text start point: (Pick one end of the pond.)
Text end point: (Pick other end of the pond.)

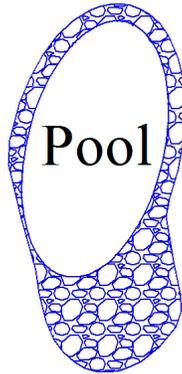


Because the Align option's pick points define the width and angle of the text, Text doesn't ask for the rotation angle, as it normally would.

- The Text command goes straight to the prompt:

Text: **Pool**

Notice that the word “Pool” is drawn with the Times New Roman text font specified by the Standard style.



QUICK SUMMARY OF TEXT JUSTIFICATION MODES

BricsCAD can specify justification (alignment) for text in many different ways:



Justification	Meaning
Start point	Baseline left
Align	Fitted between two points
Fit	Fitted with constant text height
Center	Baseline center
Middle	Exact center of text
Right	Baseline right
TL	Top left
TC	Top center
TR	Top right
ML	Middle left
MC	Middle center (equivalent to Middle)
MR	Middle right
BL	Baseline left (equivalent to Start point)
BC	Baseline center (equivalent to Center)
BR	Baseline right (equivalent to Right)

4. Press **Enter** at the 'Text:' prompt to exit the command.
Text: (Press **ENTER**)

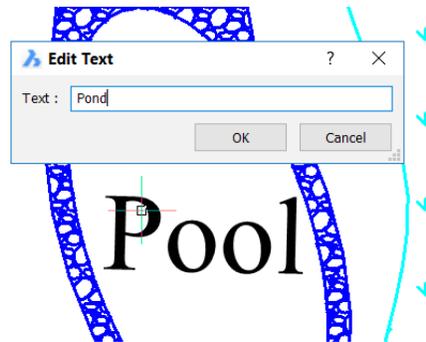
Changing Text

Command	DdEdit
Shortcut	<i>double-click text</i>
Alias	ED

Oops! "Pool" should read "Pond."

To change the wording, use the **DdEdit** command. You could enter the command name at the : prompt, but there is another way:

1. Double-click the word **Pool**. This action activates the **DdEdit** command (short for "dynamic dialog editor"). Notice the Edit Text dialog box.
2. Replace Pool with **Pond**.



3. Click **OK**. Notice that the word is corrected in the drawing.



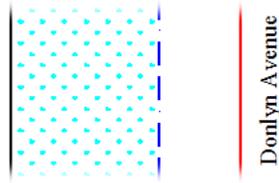
PLACING ROTATED TEXT

Now we'll add some more text. This time you place the text sideways to see that BricsCAD can place rotated text. First, though, perform a **Zoom All** to see the entire drawing.

Restart the **Text** command, and then follow these steps:

1. Enter **Text**:
: text
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: (Pick a point on street)

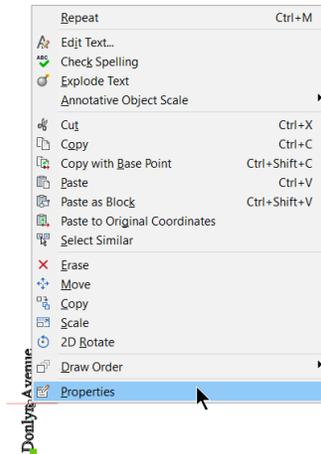
2. Specify a rotation angle of 90 degrees, and then enter the text:
 Rotation angle of text <0>: **90**
 Text: **Donlyn Avenue**
 Text: (Press **Enter**)



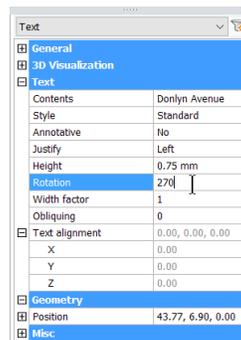
By specifying a rotation angle of 90 degrees, BricsCAD draws the text sideways. You can, of course, place text at any angle — from 0 degrees to 359 degrees.

3. If you picked the wrong rotation angle, you can rotate the text after the fact. Use the **Zoom Object** command to take a closer look.
4. Pick the text...
 : (Pick text)

..and then right-click the text. From the shortcut menu, choose **Properties**.



5. In the Properties palette:
 - a. Look for **Rotation** in the Text section.
 - b. Change 90 to **270**, and then press **Tab**.



6. Notice that BricsCAD rotates the text by 180 degrees. Press **Esc** to exit grips editing.

PLACING MULTIPLE LINES OF TEXT

Place more callouts throughout the drawing. First, though, use **Zoom All** to see the entire drawing, and then follow these steps:

1. Use the **Text** command to add multiple callouts to the drawing.
: **text**
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: *(Pick near the bottom of the drawing)*
2. Remember to change the rotation angle back to 0 degrees.
Rotation angle of text <90>: **0**
3. Enter two lines of text, as follows:
Text: **4486 Donlyn Avenue** *(Press Enter)*
Text: **Anytown BC** *(Press Enter)*
Text: *(Press Enter)*

Notice how BricsCAD places the second line of text precisely below the first.

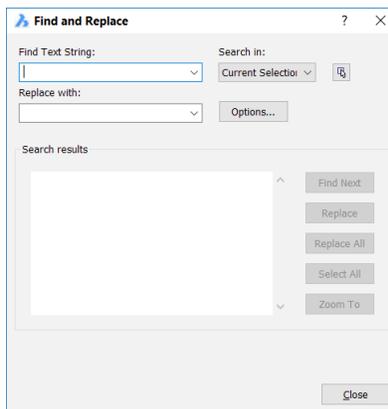
4. Repeat the command to add labels to trees, such as Birch, Aspen, Yellow Pine, or Western Red Cedar.

At this point, it is a good idea to save your work using the **Save** command.

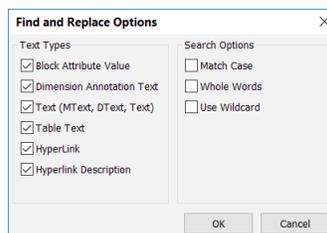
SEARCHING AND REPLACING TEXT

Command	Find
Ribbon	Draw Text Find
Menu Bar	Edit Find

To find text in drawings and/or replace it with different text, use the **Find** command (**Edit | Find**). This displays the following dialog box. In the **Find Text String** field, enter the text for which you are looking, and then click **Find Next**:



By clicking the **Options** button, you can narrow the search by BricsCAD to certain kinds of text:



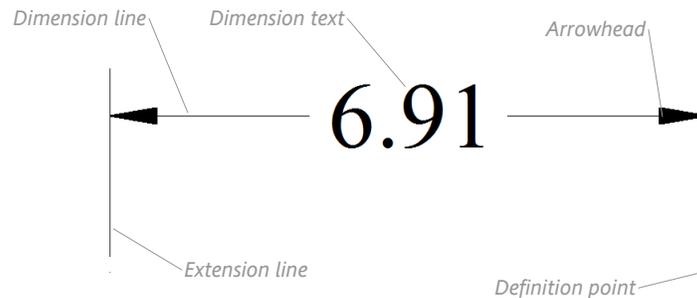
- > **Block Attribute Value** — attribute text found in blocks
- > **Dimension Annotation Text** — text in dimensions, including leaders
- > **Text (MTert, DText, Text)** — all other kinds of text, including field text
- > **Table Text** — text found in the cells of tables
- > **Hyperlink** — text in URLs (links)
- > **Hyperlink Description** — text found in the descriptions of hyperlinks

And then there are these additional options for narrowing down the search further:

- > **Match Case** — the text must match the pattern of upper and lower case
- > **Whole Words** — the text must have spaces on either side, and so not be part of a larger word
- > **Use Wildcard** — the text contains characters wildcards, such as * (any number of characters) and ? (any single character)

Placing Dimensions in Drawings

With callouts firmly placed in the drawing, let's turn to dimensioning the lot with commands that start with Dim (short for "dimensioning"). Some of the elements of linear dimensions are illustrated below.



The bad news is that there are dozens and dozens, and dozens of settings for dimensions to adhere to the drafting standards used by various countries; the good news is that BricsCAD comes pre-configured with most settings at reasonable values, for both metric and imperial units.

The settings are known as *dimensioning variables*, or "dimvars" for short. You can change dimvars in two ways: with the Dimension Styles node (in the Drawing Explorer dialog box), or by directly entering their names at the ':' prompt. In the following tutorials, you use both methods.

PREPARING THE DRAWING FOR DIMENSIONING

System Variable DimScale

Like text, hatch patterns, and linetypes, the scale of dimensions is relative to the printed size. Otherwise, the arrowheads and text will be too small to read. Before drawing dimensions, you should set the scale, as follows:

1. To set the dimension scale, enter the **DimScale** dimvar as a command:
: **dimscale**
2. The same scale factor that you calculated for text works here: **4** for metric units (or 142 for imperial units).
Enter new value for DIMSCALE <1>: **4**

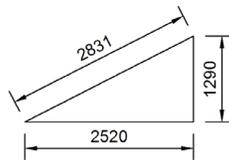
TIP The height of dimension text is controlled by **Style**, and is unaffected by the **DimStyle** scale factor.

3. Create a new layer called **Dims** (black color, continuous linetype), and then make it current.
4. If you wish, freeze unnecessary layers, such as Text, Lawn, Plants, and Pond. This keeps them from getting in the way. (I left some of these layers on for the illustrations below so that it is easier for you to see the locations in the drawing where the dimensions are being applied.)
5. Close the Drawing Explorer dialog box.
6. Because dimensioning often takes place at intersections, turn on **INTersection** entity snap mode, as follows:
: intersection

DIMENSIONING THE YARD

Command	DimLinear
Ribbon	Draw Dimensions Linear
Menu Bar	Dimension Linear
Alias	dimlin
Command	DimContinue
Ribbon	Draw Dimensions Continuous
Menu Bar	Dimension Continuous
Alias	dimcont

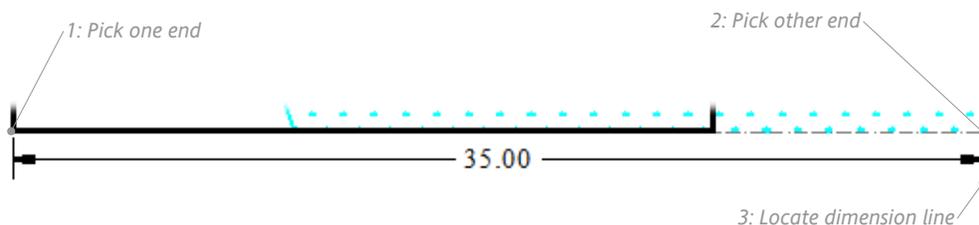
Begin dimensioning with the most used dimensioning command, DimLinear. It draws horizontal, vertical, and rotated linear dimensions — depending on how you move the cursor. By tradition, most dimensions are either horizontal or vertical. In rare cases they are placed at an angle (“rotated”). BricsCAD determines how to construct the dimension from the relative positions of your pick points in the drawing.



- **For horizontal dimensions** — pick two points roughly horizontal, and BricsCAD draws the dimension perfectly horizontal
- **For vertical dimensions** — pick two points roughly vertical, and BricsCAD draws the dimension exactly vertical.
- **For rotated dimensions** — at the prompt, enter **R** to force the dimension to be placed at an angle. BricsCAD prompts you to specify the angle. A more useful command, however, is DimAligned, which determines the angle automatically.

1. To begin, select **Linear** from the **Dimensions** menu.
: _dimlinear
2. Dimension the lower lot line horizontally, as follows:
ENTER to select entity/<Origin of first extension line>: (Pick intersection near #1)
Origin of second extension line: (Pick other intersection at #2)

Angle/Text/Horizontal/Vertical/Rotated: (Locate dimension line by picking a spot below the lot line, near #3 in the figure below)



BricsCAD determines the length of the line from your pick points, and then places all the elements of the

dimension for you, including determining the length between the two extension lines.

TIP BricsCAD automatically draws all of the components of a dimension: both extension lines, the dimension line, both arrowheads, and the dimension text.

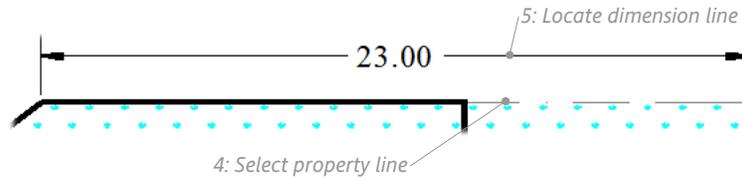
3. Try another horizontal dimension of the top lot line. This time, use BricsCAD's entity dimensioning, in which you pick the entity and BricsCAD dimensions it. The method uses just two picks, instead of three.

Press the **Spacebar** to repeat the **DimLinear** command:

```
: (Press Spacebar to repeat command) _dimlinear  
ENTER to select entity/<Origin of first extension line>: (Press Enter to select entities)
```

```
Select entity to dimension: (Pick property line at #4.)
```

```
Angle/Text/Orientation of dimension line: Horizontal/Vertical/Rotated: (Pick #5)
```



4. You can continue drawing horizontal dimensions with the **DimContinue** command. This also is a way to dimension with fewer picks. When there is a lot of dimensioning to do, you want to minimize the amount of picking you have to do! From the **DIMENSION** menu, select **Continue**:

```
: _dimcontinue
```

```
Continue: ENTER to select starting dimension/<Origin of next extension line>: (Pick #6)
```

TIP The **DimLinear** command has several options
Angle/Text/Orientation of dimension line: Horizontal/Vertical/Rotated:

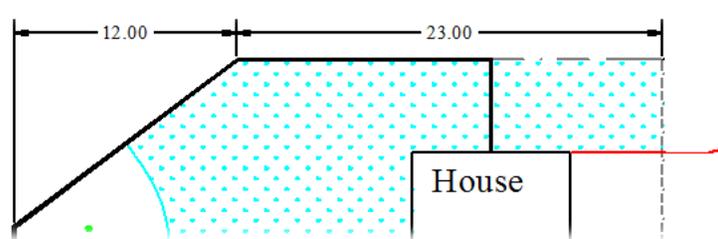
Angle — rotates the dimension text.

Text — lets you modify or replace the dimension text.

Orientation — forces the dimension line to be horizontal, vertical, or rotated.

5. To end the command, it is important that you press **Esc**, because pressing Enter makes BricsCAD prompt you to choose the starting dimension.

```
Continue: ENTER to select starting dimension/<Origin of next extension line>: (Press ESC  
to end the command)
```



Because BricsCAD knows where the last extension line was, all it now needs to know the location of the *next* extension line to draw in the second dimension.

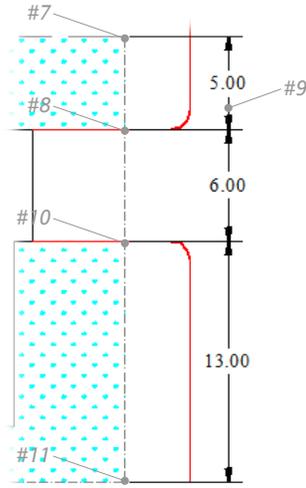
VERTICAL AND BASELINE DIMENSIONS

As noted earlier, the DimLinear command draws vertical dimensions in addition to horizontal dimensions. When you move the cursor in a vertical direction, BricsCAD knows to draw the dimension vertically.

1. To draw vertical dimensions, select **Dimension | Linear**, and then follow the numbered pick point from the illustration below.

```
: _dimlinear
ENTER to select entity/<Origin of first extension line>: (Pick #7)
Origin of second extension line: (Pick #8)

Angle/Text/Orientation of dimension line: Horizontal/Vertical/Rotated: (Pick #9)
```



2. Use **DimContinue** to continue the vertical dimensions along the right side of the lot at points 10 and 11.

```
: dimcont
Continue: ENTER to select starting dimension/<Origin of next extension line>: (Pick #10)
Continue: ENTER to select starting dimension/<Origin of next extension line>: (Pick #11)
Continue: ENTER to select starting dimension/<Origin of next extension line>: (Press Esc)
```

Baseline Dimensioning

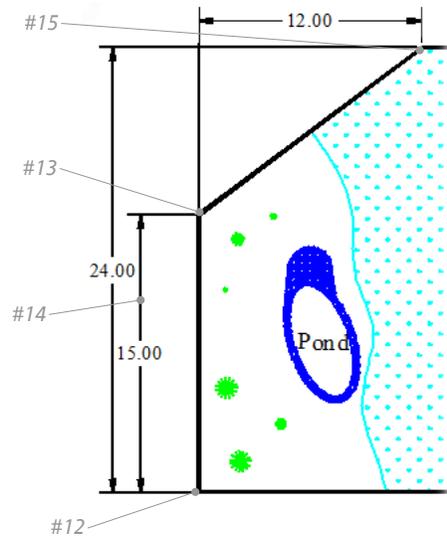
Command	DimBaseline
Ribbon	Draw Dimensions Baseline
Menu Bar	Dimensions Baseline
Alias	dimbase

A variation on the DimContinue command is **DimBaseline**. Rather than continuing a dimension from the *previous* extension line, DimBaseline dimensions from the *original* extension line (a.k.a. “the base line”). To see how it works, first place a vertical dimension:

3. With the **DimLinear** command, place the first extension line at #12. Place the second extension line at #13, and then the dimension line at #14.
4. Now try out the DimBaseline command. From the **Dimension** menu, select **Baseline**.

```
: _dimbaseline
Baseline: ENTER to select starting dimension/<Origin of next extension line>: (Pick #15)
Baseline: ENTER to select starting dimension/<Origin of next extension line>: (Press Esc)
```

Pressing **Esc** exits the command.



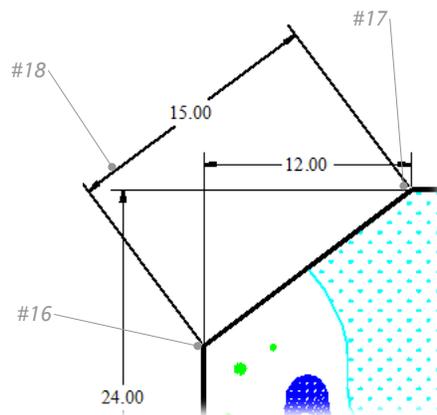
The **DimBaseline** and **DimContinue** commands also work with other linear and angular dimensions.

ALIGNED AND RADIAL DIMENSIONS

Command	DimAligned
Ribbon	Draw Dimensions Aligned
Menu Bar	Dimensions Aligned
Alias	dimali

So far, you have dimensioned the straight and angled portions of the lot line with horizontal and vertical dimension commands. To dimension an angled line, you use the **DimAligned** command.

1. Enter the **DimAligned** command:
: **dimaligned**
ENTER to select entity/<Origin of first extension line>: (Pick #16.)
Origin of second extension line: (Pick #17.)
Angle/Text/<Location of dimension line>: (Pick #18.)



The dimension you draw may look different, depending on how the angled line was drawn.

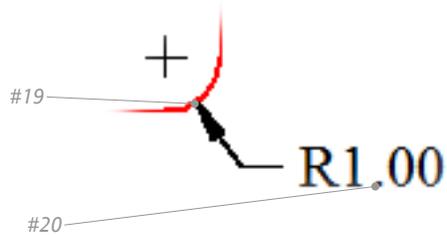
Radial Dimensions

Command	DimRadius
Ribbon	Draw Dimensions Radius
Menu Bar	Dimensions Radius
Alias	dimrad

So far, all dimensioning commands have presented pretty much the same prompts. Now try one that's a bit different. The **DimRadius** command dimensions arcs and circles. The related DimDiameter command places dimensions across diameters of circular entities.

- (Zoom in for a closer look, if necessary.) Enter the **DimRadius** command, and then follow the prompts:
: dimradius
Select arc or circle: (Pick #19)
Dimension text = 1
Angle/Text/<Location of dimension line>: (Pick #20)

The DimRadius command gives you some flexibility as to where you want to place the dimension text. As you move the cursor, BricsCAD ghosts in the leader and text.



- Save your work, and then print out a copy.

Annotatively-Scaled Text and Dimensions

In this and the previous chapter, you needed to work out the scale factor for *annotative* entities, such as text, dimensions, hatch patterns, and linetypes. The scale of these entities takes into account the size of paper on which the drawing will be printed. If the paper size never changes, then all is fine.

If the paper size will change, such as printing sometimes on A- (A4) and then on D-size (A1) paper, then you need a handy way to change the size of annotative entities easily. BricsCAD fortunately provides this through annotative scaling.

When you turn on the annotative scaling option, BricsCAD determines the correct height to use for text, the correct scale factor for hatch patterns, and so on. Indeed, it can store several annotative scales per entity, hiding those scale(s) that are currently unneeded.

Think of annotative scaling as a “master scale factor” that affects only annotative entities, and only in three areas of drawings: with annotative entities, in viewports, and on plots. To understand this, I have put together this rule:

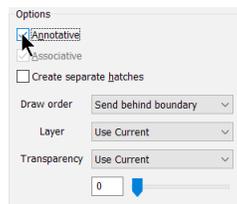
$$\text{Annotative scale} = \text{Viewport scale} = \text{Plot scale}$$

The scale at which you plot the drawing determines the scale factor of annotative entities. (If your drawing uses viewports, the same scale applies.)

To use annotative scaling, you turn it on in styles controlled by the Drawing Explorer. You click the Annotative check box in styles for text, multilines, dimensions, multilines, and blocks. Below, I show that creating an annotative text style is as easy as clicking the Annotative property; I happened to name the text style “Annotative.”

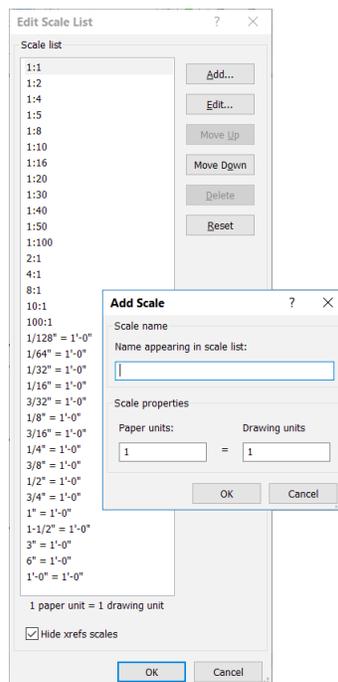
	C...	Text Style Name	Annotative	Match Orientation to Layout	Height
1		Standard	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.75
2		DYN_DIM	<input type="checkbox"/>	<input type="checkbox"/>	0
3		Annotative	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.75

For entities that have no styles, their dialog box has the Annotative check box, such as for hatches and attributes. See the Annotative option in the fragment of the Hatch and Gradient dialog box illustrated below.



Linetypes are a bit different, in that they use the MsLtScale variable to match the current annotative scale factor. (Annotative scales are applied to entities, not to layers.)

You can't just type one in an annotative scale factor; you are limited to the ones provided by Brics-CAD. You can, however, edit the list with the ScaleListEdit command to add and remove scale factors.



USING ANNOTATIVE SCALING

So far, I've been doing a lot of talking. Let's see how annotative scaling works for real. I'll have you switch back and forth between model and layout modes. *Layout mode* shows you what the drawing looks like on the paper you'll be printing it on; it is like an interactive print preview mode. (I've frozen the Dims layer so that the dimensions do not clutter the drawing.)

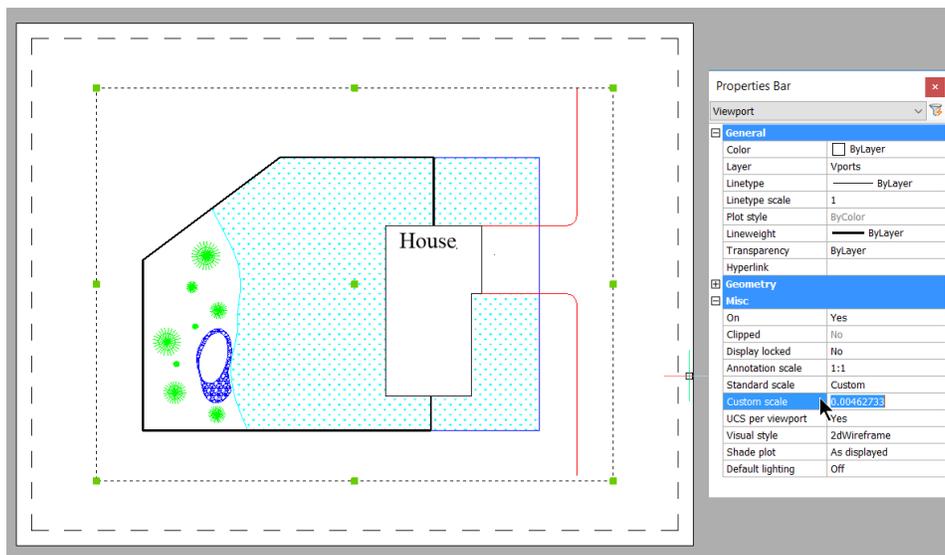
You find tabs for switching between the model and layout modes at the bottom of the drawing area.

1. Open the *Imp-Tutorial-06.dwg* file.
2. Switch to layout mode by clicking the **Layout1** tab.



In the figure below:

- > White area represents the paper
- > Black rectangle is the *viewport*, in which you see the model; the viewport has green grips and dashed lines, because I have selected it
- > Dashed rectangle is the *margin*, the area at the edge of the paper on which the printer cannot print



3. When the drawing is first opened in a viewport, it is most likely displayed at a non-standard scale. BricsCAD fits the model's extents inside the area of the viewport, and so this drawing has an initial scale of 0.00462733 (or 1:216.1073, expressed as a *scale factor*). To see the model's scale, select the viewport border, and then check the **Custom Scale** value in the Properties panel.

Because 1:216.1073 is a non-standard scale factor, you have to change it to meet these two criteria:

- > The entire drawing must fit the viewport (usually somewhat smaller than the viewport)
- > The drawing must be at a standard scale

TIP To convert the *scale* into the *scale factor*, divide 1 by the scale. In this case

$$0.00462733 = 1/0.00462733 = 1:216.1073$$

- You change the oddball scale to a standard one through a button on the status bar or the Properties panel. On the status bar, the annotative scale button appears near the right end next to DUCS, and probably looks like 1:1 right now. (If you do not see it, click the arrow at the right end of the status bar, and then turn on the **Annotative Scale** button.)

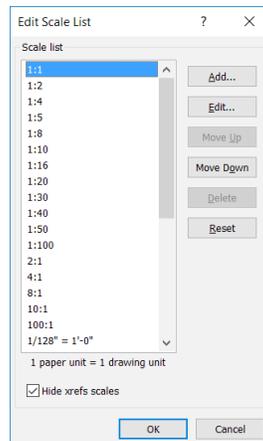


The annotation scale button does double-duty of setting the scale for the viewport and for annotative objects.

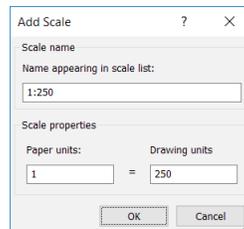
- Enter the model area by double-clicking inside the viewport border (the black rectangle), because annotative scaling works only in model space.
- Use the **Zoom Extents** command to ensure that the drawing fits the viewport snugly. This gives you the largest scale permissible.
- For the scale factor, pick the next largest whole number to 1:216.1073, because standard scale factors are always whole numbers, such as 1:250. Choosing the next largest number makes the drawing slightly smaller in the viewport. You don't want some of the drawing hidden by the edge of the viewport! (If the scale factor were something like 1:7.25, then pick 1:8.)

There is, however, no scale factor listed at 1:250, so you need to add it with the ScaleListEdit command, as follows:

- Enter the **ScaleListEdit** command. Notice the Edit Scale List dialog box.
: **scalelistedit**

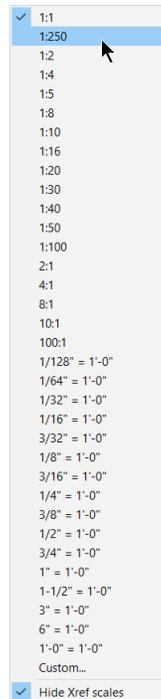


- In the Edit Scale List dialog box, click **Add**.
- For the 'Name appearing in scale list' you can type anything. For this tutorial, enter **1:250**.



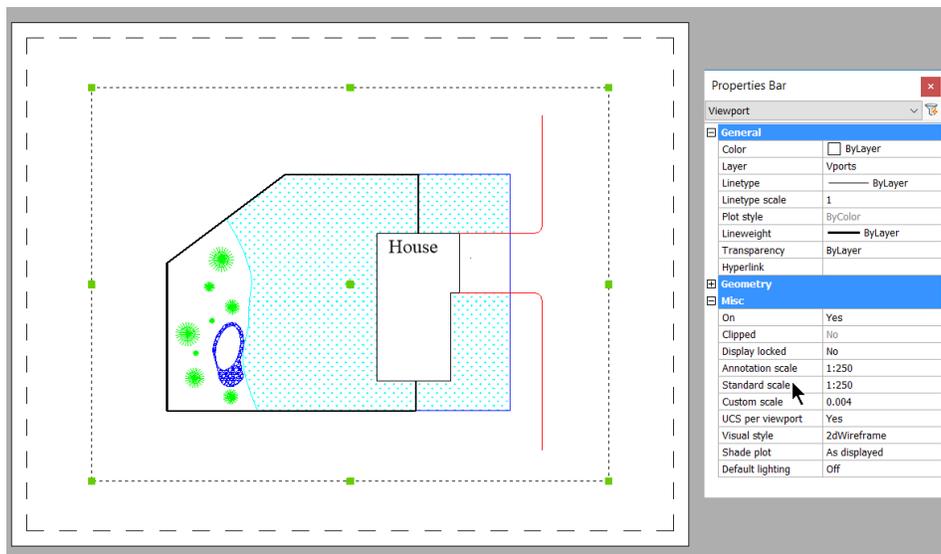
- For 'Drawing Units' enter **250**. Leave the Paper Units set to 1.
- Click **OK**.

8. To pick the annotative scale factor, right-click the **Annotation Scale** button (1:1), and then select **1:250**.



You know that you have picked the correct scale when the drawing becomes slightly smaller, but not too much smaller. (See figure below.) Notice also that the reported scale matches the annotation factor you selected. **From now on, the viewport scale is locked to the annotation scale.** This means that any change you make to the annotation scale changes the viewport scale automatically.

TIP If the drawing does not change its size in the viewport automatically, you will need to make the change manually. In the Properties panel, change the value of **Viewport Scale** to match **Annotation Scale**, or 1:4 for this tutorial.



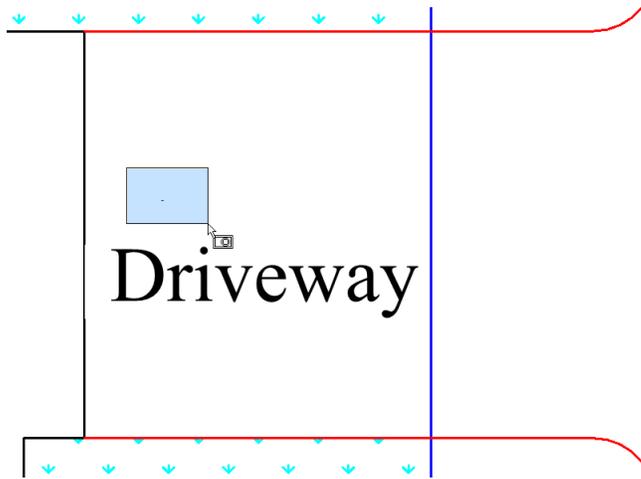
9. With the annotative scale factor set up, test the annotation scaling feature by placing two kinds of text in Model tab, standard and annotative. First the standard text:
- Click the **Model** tab.

TIP Why place the annotative text in model tab? (Why not in the layout tab?) My CAD-expert friend Bill Fane explains, “If you zoom and pan while entering annotative text in a layout tab, you mess up the viewport’s scale; when you go back into model space, the annotations are missing, because their scale no longer matches the scale of the viewport. For this reason, it is best to enter annotative text in Model tab.”

- b. Enter the **Text** command to place regular text with the “Standard” style (which is non-annotative) at a height of 1/8" (0.125"):

```
: text
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: (Pick a point)
Height of text <0.2000>: 0.125
Rotation angle of text <0>: (Press Enter)
Text: Driveway
Text: (Press Enter)
```

Notice that the text looks very tiny. It is so tiny that I needed to highlight it with a blue selection rectangle; look for the short dash in the center. It is so tiny because it is placed at the size we normally use for text in drawings, 1/8". Recall from earlier in the chapter that we would normally scale it up by 250x and so draw it 31.25" high!



10. Now place the annotative text:
- a. Restart the **Text** command, and then use the **Style** option to change the text style to an annotative one.
- ```
: text
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: s
Text style to use (or '?'): Annotative
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: (Pick a point)
```
- b. Again place text with a height of 1/8".
- ```
Height of text <0.2000>: 0.125
Rotation angle of text <0>: (Press Enter)
Text: Driveway
Text: (Press Enter)
```

Notice that the text looks a reasonable size. BricsCAD did the scale calculation for us and determined that the text should be 31.25" tall.

- TIP If the drawing does not contain an annotative text style, then you can create one like this:
1. Enter the **Style** command.
 2. Make a copy of an existing style by clicking the **New** button.
 3. Click the **Annotative** option. It's that simple!



	C...	Text Style Name	Annotative	Match Orientation to Layout	Height	Width Factor	Oblique Angle	Font Name	Style	Language
1		DYN_DIM		<input type="checkbox"/>	0	1	0	Arial	Regular	Western
2		Standard		<input type="checkbox"/>	0	1	0	Times New Roman	Regular	Western
3		Annotative	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1	0	Times New Roman	Regular	Western

4. Give the style a clever name, such as “Annotative,” and then exit the Drawing Explorer.

A FEW EXTRA THINGS ABOUT ANNOTATIVE SCALING

Annotative scaling is tricky enough that some users avoid applying it to their drawings. Here are some things to watch out for.

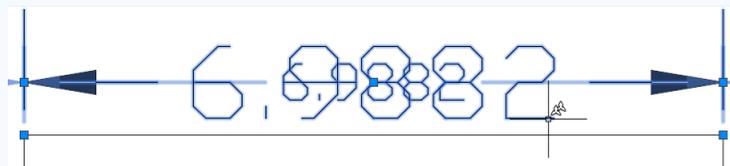
WHAT TO DO WHEN ANNOTATIVELY-SCALED ENTITIES GO MISSING

Annotative scaling can cause objects to disappear. The problem occurs when the viewport scale no longer matches the entities' annotative scale. This, however, makes sense: BricsCAD drawings can have multiple layouts, each with a different scale factor so that drawings can be printed on a variety of paper sizes. (This is why all drawings have just one Model tab but many Layout tabs: each Layout tab is normally assigned a different plot scale and/or a different view of the drawing.)

Annotatively-scaled entities only appear when the correct scale is set in model space.

VIEWING ALL ANNOTATIVE SCALES IN DRAWINGS

There is one exception to disappearing entities, and that's when multiple copies appear in the drawing. This occurs when more than one annotative scale is applied to entities. When the **AnnoAllVisible** system variable is turned on ($= 1$), then all variations appear, as illustrated below.

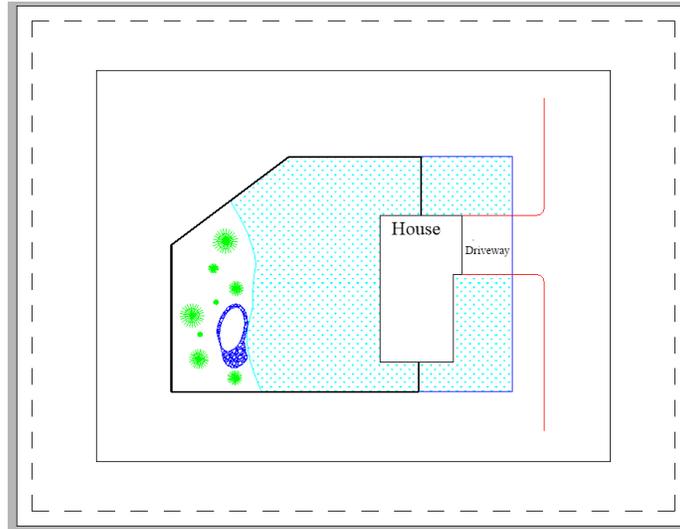


This can lead to some confusion as multiple-sized versions of text and other entities appear. The variable is meant for debugging: it's handy when you need to see all scales assigned to an annotative entity. To keep from seeing in double or triple vision, BricsCAD leaves AnnoAllVisible turned off by default.

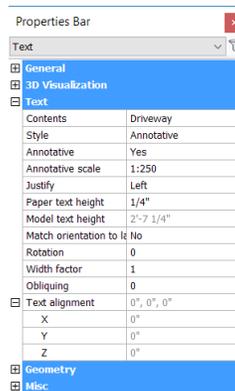
HOW TO SPOT ANNOTATIVELY SCALED ENTITIES

An entity affected by annotative scaling shows the  icon when you pass the cursor over it. When the entity has been assigned more than one annotative scale, then the double version  of the icon is displayed. The icon represents the end view of an engineer's scale ruler, which allows engineers to work with multiple scales.

11. Any text you place now in Model tab will be scaled appropriately for the viewport and the scale factor. To see that the text appears at the correct size in the viewport, switch to **Layout1**. The text placed with the Standard style is effectively invisible, but the annotative text "Driveway" is correctly visible.



12. Select the annotative text, and notice that the Properties panel reports two scales:
 - ▶ Paper text height **1/4"**
 - ▶ Model text height **2'-7 1/4"** (same as 31.25")



TIP Should you need to plot the drawing at a second scale, then I recommend you set up a second layout tab, set the new annotation scale, and then go back to model tab. Select the text and other annotations needing scaling, and then apply the new annotation scale factor.

Summary

This lesson showed how to add text and dimensions to drawings.

This tutorial is now complete. The following lessons describe other 2D CAD topics in greater detail, such as working with attribute data, and constructing 2D regions.

Bills of Material

One of the most powerful aspect of CAD is its ability to generate information from drawings. You saw a hint of this in an earlier lesson when you used the Dist and List commands to find distances, lengths, areas, and totals.

Here you learn how to extract information stored by attributes in blocks, and then how to create a bill of materials (BOM) in the BricsCAD drawing and externally in a spreadsheet.

IN THIS CHAPTER

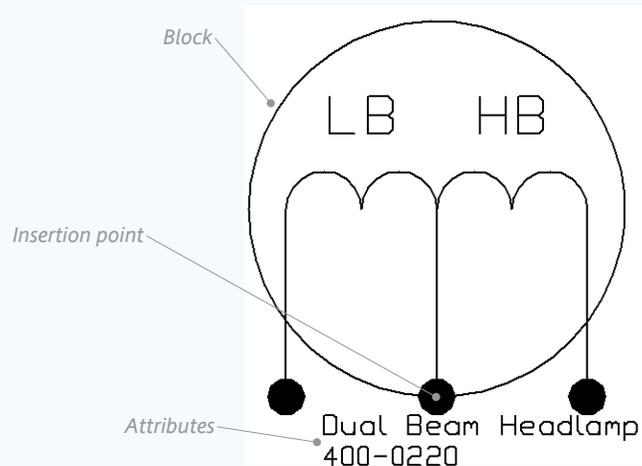
- Defining attributes (data)
- Attaching attributes to blocks
- Creating templates for formatting attribute and block data
- Exporting attributes and block data from drawings
- Importing the data into spreadsheets for analysis
- Importing spreadsheets into drawings

KEY TERMS IN THIS CHAPTER

Attribute — consists of text data attached to blocks

Attribute extraction — describes a process that exports attribute and block data to files

Block — describes BricsCAD's term for a symbol. (Other CAD software packages call blocks “components,” “cells,” “shapes,” “symbols,” or “parts.”)



Donut — consists of a solid-filled circle

Insertion point — specifies the place at which blocks are inserted in drawings

Prompt — refers to the text displayed by BricsCAD during block insertions

Tag — identifies attributes by name

Template file — defines the format of data in export files

Value — specifies the default values of attribute data

USEFUL ABBREVIATIONS

CDF	Comma-delimited format
DXF	Drawing interchange format
ODF	Open Document file
SDF	Space-delimited format
TXT	Extension for an ASCII text file
XLS	Excel spreadsheet file

NEW COMMANDS

Command	Aliases	Menu Bar	Ribbon Tab
AttDef	at	Tools Attributes Define Attributes	Insert Block Define Attribute
AttDisp	ad	Tools Attributes Attribute Display	Insert Block Attribute Display
AttExt	ax	Tools Attributes Extract Attributes	Insert Block Extract Attributes
Donut	do, doughnut	Draw Donut	Draw Draw Polygons Donut
Table	...	Draw Table	Draw Tables Table

About Attribute Data

CAD can be used for many different kinds of drawings, such as floor plans, building construction, mechanical design, and electrical layouts. Drawings are usually made using standard parts, such as desks, windows, screws, and switches. When the parts contain attribute data, then the drawings become truly useful.

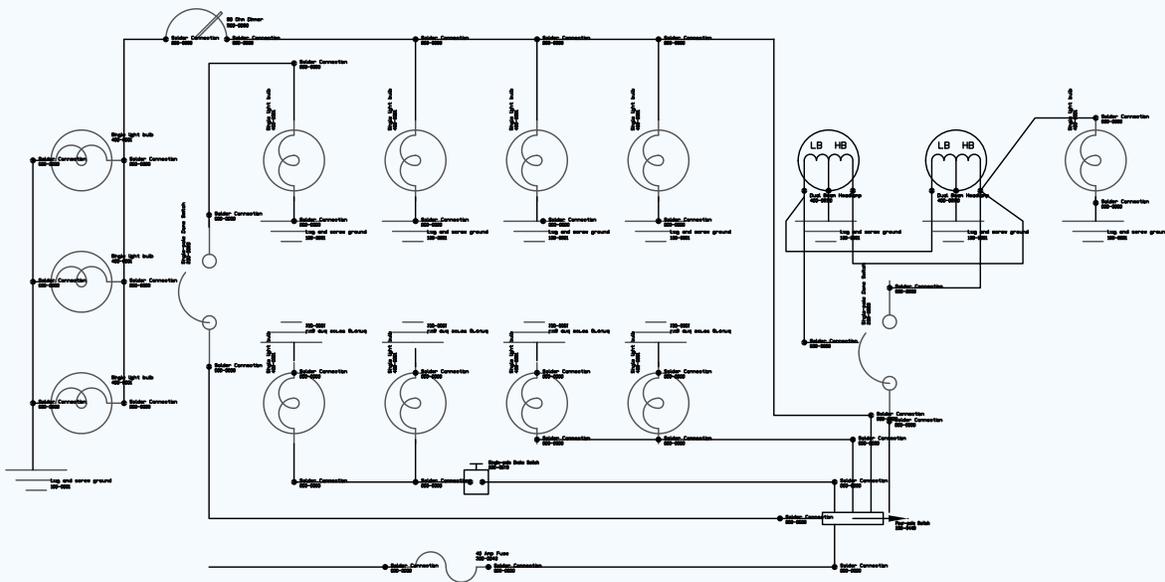
In this chapter, you create the schematic drawing for an automobile electrical system. The drawing contains numerous components, such as a battery and a fuse, several kinds of light bulbs, and quite a few ground and solder connections. You could count these components by hand, but I'd wager that you would miss a few and end up with a wrong total. (Well, *you* might not miscount, but I know I would.)

It's faster and 100% accurate to let BricsCAD do the counting. Part of the counting process is called "attribute extraction," because BricsCAD *extracts* attribute data from the drawing and then stores it in a data file. The data file can be imported into a spreadsheet to count the components, determine component costs, and so on. The parts count can be placed in drawings as tables.

This is very powerful stuff, and is known in many disciplines as BOM (short for "bill of materials") or in architecture as FM (short for "facilities management").

THIS CHAPTER'S DRAWING

During this lesson, you work with the following drawing file, *electric.dwg*.



Electric.dwg can be downloaded from <http://www.upfrontezine.com/lb8>

The attribute extraction process progresses through these steps:

- Step 1: **Template** — define an attribute extraction template file
- Step 2: **Extract** — use the AttExt command to extract attributes to a data file
- Step 3: **Import** — import the data file into a spreadsheet program

AttExt is the name of the command that extracts attributes from drawings. It is a very old command, one that goes back to 1985 in some CAD packages — which explains why it is somewhat ornery. Over the decades it has remained essentially unchanged, except for the addition of the dialog box front-end. While other programs may boast “coaches” and “wizards” to step you through complicated procedures, BricsCAD does not assist you in this area. And so this lesson exists to guide you.

In the first tutorial, you create a headlamp block, and then attach attribute data to it. In the following tutorials, you open a copy of the completed *electric.dwg* file, and then extract the attribute data from it. Later, you open the attribute data file in a spreadsheet program to count the parts, and finally you bring that data back into BricsCAD as a table.

Creating Blocks with Attributes

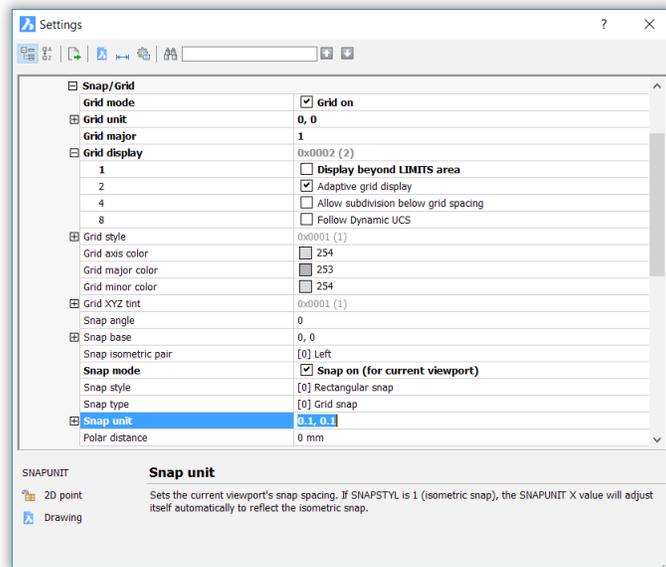
The example drawing for this lesson is part of the electrical schematic of an antique automobile. See the next page. (Wondering which auto? It’s adapted from the electrical system that helped power my first car, a bright-orange 1965 Volkswagen Beetle.) If you have repair manuals or other schematic drawings laying around, you can use them to reproduce your own schematic with BricsCAD. If not, then follow along with the schematic sketch I provide with this ebook.

To begin with, you create one block and attach attributes to it. Working with attributes is kind of tedious, which is why I won’t get you create *all* of the blocks. I’ve got them all in a drawing that you can download.

1. Start BricsCAD with a new drawing using the 2D Drawing workspace (in metric units) or the *Default-cm.dwt* template file.
2. Because you will be drawing the block to an accuracy of 0.1 unit, it makes sense to change the snap spacing to this value and to turn on the grid. Follow these steps:
 - a. On the status bar, right-click **SNAP**, and then choose **Settings** from the shortcut menu.
 - b. In the Settings dialog box, make these changes:

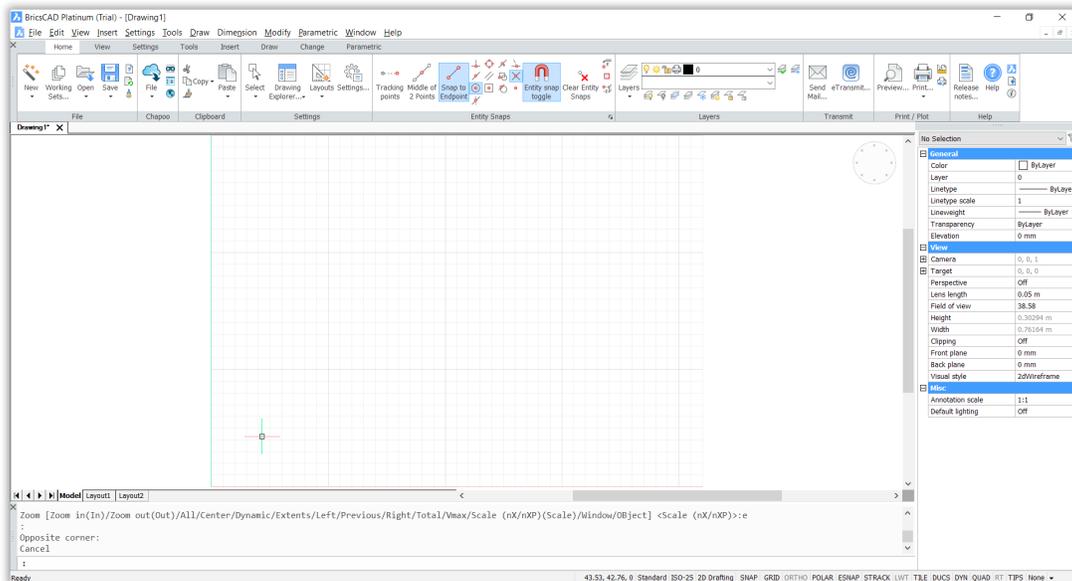
Grid Mode	✓ Grid on
Grid Unit	0.1,0.1
Grid Major	1
Grid Display	Turn off Display beyond LIMITS area
Snap Mode	✓ Snap on
Snap Unit	0.1,0.1

The result should be like the bold-faced entries in the Settings dialog box illustrated below.



- c. Click **X** when done to close the dialog box. Notice that the screen fills with a fine grid of lines. (Recall that the the extents of the grid display are determined by the limits.)
3. A couple more steps, and you are ready to draw:
 - a. There is no need for the UCS icon, so turn it off with the **Ucslcon** command:


```
: ucslcon
ON/OFF/All/ORigin/Corner/<ON>: off
```
 - b. Perform a **Zoom All** to center the drawing on the screen.



Notice that there are some light gray grid lines and some that are dark gray. The dark gray lines are called “major grid lines” and are spaced 100 units apart; the light gray ones are called “minor grid lines” and are spaced 10 units apart.

DRAWING BLOCKS

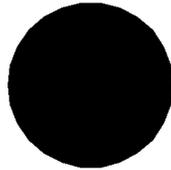
Command	Donut
Ribbon	Draw Draw Polygons Donut
Menu Bar	Draw Donut
Alias	doughnut

The symbol used for solder connections is one that's really easy to draw. It is just a fat dot, and it is best drawn with the **Donut** command. This command creates solid looking circles out of wide polyline arcs, and are specified with outer and inner diameters.

1. Start the **Donut** command.
: **donut**
2. The solder connection is 0.1 units in diameter. To make a solid-filled donut, specify an inside diameter of zero. Enter the following sizes at the prompts:
2Point/3Point/RadTanTan/<Inside diameter of donut> <0.5000>: **0**
Outside diameter of donut <1.0000>: **0.1**
3. BricsCAD prompts you to place the donut, as follows:
Center of donut: *(Pick anywhere in the screen)*
Center of donut: *(Press ENTER to exit the command)*

The Donut command repeats itself until you exit it by pressing **Enter**.

4. That 0.1-unit donut sure is tiny. Enlarge the drawing area with the **Zoom Extents** command:
: **zoom**
Zoom: In/Out/All/Center/.../<Scale (nX/nXP)>: **e**



(If the donut looks like it has straight sides, then use the **Regen** command to smooth its edges.)
: **regen**

DEFINING ATTRIBUTES

Command	AttDef
Ribbon	Insert Blocks Define Attributes
Menu Bar	Tools Attributes Define Attributes

With the solder connection drawn as a donut, you can now create the attribute data. The process takes these steps:

- Step 1: **AttDef** — define attribute definitions with the **AttDef** command (as described in this tutorial)
- Step 2: **Block** — attach attribute definitions to blocks with the **Block** command (in tutorials later)
- Step 3: **Insert** — define attribute data during the **Insert** command; optionally, edit data with **AttEdit**

Attributes are customized data that store information, such as part numbers and prices. Drawings are useful to showing you how to assemble the electrical system, but the bill of materials (derived from the attributes) tells you *how many* parts are needed for the assembly.

TIP Attributes attach only to blocks. It is possible to add data to other kinds of entities, but the process is more technical, because it involves *xdata* (short for “extended entity data”). Nevertheless, BricsCAD can export all data (such as properties) from all entities to CSV files for further processing.

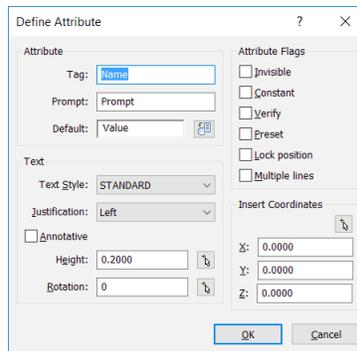
A single block can contain one or more attributes. Just as blocks are graphical descriptions, attributes are textual descriptions. And only text or numbers can be used; graphical data, such as images and other entities, cannot be included in attributes.

Attributes can describe the block’s part number, manufacturer, price, and any other text-based information you want to include. These attributes can have descriptive labels, such as “Product name,” “Manufacturer,” “Model number,” “Stock number,” “Serial number,” and “Material.”

You could include “Price” as an attribute field; but since prices tend to change, it is better to deal with the price later in the spreadsheet program. Instead, I suggest that if you need prices in attributes that you use a code instead, which is later substituted with the latest price.

Attributes are created with the **AttDef** command (short for “attribute definition”).

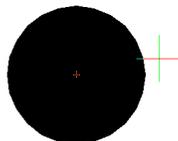
1. Start the **AttDef** command. Notice the Define Attribute dialog box.
: **attdef**



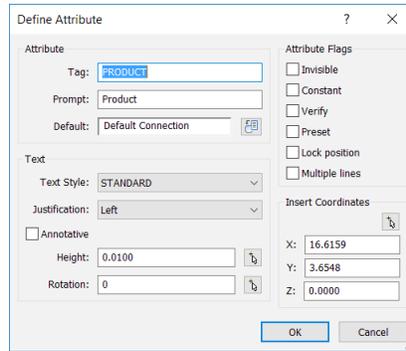
2. In the Attribute section, enter the following text:

Attribute Field	Value
Tag	Product
Prompt	Product
Default	Solder Connection

3. In the Text section of the dialog box, set the **Height** (of the text) to a tiny, unobtrusive value, such as **0.01**. You don’t actually want to see these attributes in drawings; you work with them through dialog boxes.
4. The Insert Coordinates section defines the point in the drawing at which the attribute text starts. A logical location would be on or near the block. To obtain the value of the **Insert Coordinates**, follow these steps:
 - a. In the Insert Coordinates section, click the  button. The dialog box disappears temporarily.
 - b. In the command bar, notice that BricsCAD prompts you for the location of the insertion point.
Specify insertion point: (Pick a point to the right of the donut)
 - c. Pick a point to the right of the connection block:



As soon as you do, the dialog box returns. Its fields should now be filled out with values that look similar to the following:



5. Click **OK**. Notice that BricsCAD shows the tag next to the donut.



TIP Although blocks can contain multiple attributes, the **AttDef** command creates just one attribute at a time. You have to repeat the command to define additional attributes, up to a maximum of 245.

ADDING MORE ATTRIBUTES

For the Stock Number, add a second attribute directly below the first:

1. Press the spacebar to repeat the **AttDef** command:
: (Press spacebar)
_attdef

And notice that the Define Attribute dialog box reappears.

2. Fill in the following data in the Attribute section:

Attribute Field	Value
Tag	Stockno
Prompt	Stock No.
Value	000-0000

Ensure the tag name “Stockno” is a single word, with no spaces.\

3. Click the  button, and then pick a point below the previous tag. (You find this easier to do if you turn off snap mode for now.)
4. Back in the dialog box, click **OK**. Notice that BricsCAD adds the second attribute below the first.



The two attributes are identified by their *tags*, “PRODUCT” and “STOCKNO.” The text will change when the attributes are later combined with the donut to create a block, as you see in the next tutorial.

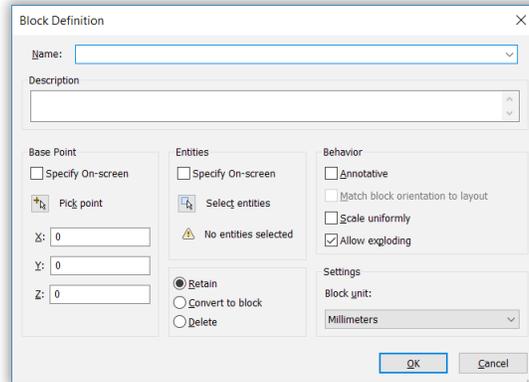
COMBINING ENTITIES AND ATTRIBUTES INTO BLOCKS

Command	Block
Ribbon	Insert Blocks Create Block
Menu Bar	Tools Create Block
Alias	b

You now have a donut with two attributes. The next step is to combine the donut and attributes into a single entity. This is done with the Block command.

1. Start the **Block** command.
: block

Notice the Block Definition dialog box.



2. The first thing to do is enter a name for the block. In this case, enter “Connect” in the **Name** field:
Name: **Connect**

You can ignore the **Description** field.

3. The most important aspect to blocks is their base point. The *base point* is the spot where the block will be inserted in drawings — either with coordinates or a cursor pick. The default is at the origin of the drawing at 0,0,0. However, other points might be more convenient, as described in the boxed text on the next page. For this block, change the base point to the center of the donut:
 - a. In the **Base Point** section, click the **Pick Point** button. The dialog box disappears temporarily.
 - b. In the command bar, notice that BricsCAD prompts you for the location of the insertion point. Pick a point to the right of the connection block.
Insertion point for new block: (Pick the center of the donut)



As soon as you pick the point, the dialog box returns.

4. BricsCAD needs to know which entities will be part of the block:
 - a. In the **Entities** section, click the **Select Entities** button. The dialog box disappears temporarily.

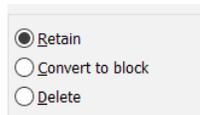
- b. In the command bar, notice that BricsCAD prompts you to select the entities that you want to be part of the block. Select the donut and the text.
 Select entities for block: (Select donut and two attribute entities)



You can press **Ctrl+A** (or **Cmd+A** on Macs) to select all entities in the drawing, or else use windowing, like I did in the figure above.

- c. Press **Enter** to end entity selection and then return to the dialog box.
 Select entities for block: (Press **ENTER**)

5. Choose the **Retain** option. This determines what happens to the entities after you click **OK**.



Option	Meaning
Retain	Retains the entities after defining them as a block
Convert to Block	Creates a block from the entities, and then inserts them in the drawing; this option combines two steps: erases the entities and inserts the block
Delete	Erases the entities after defining them as a block; this option saves using the Erase command to later remove the entities

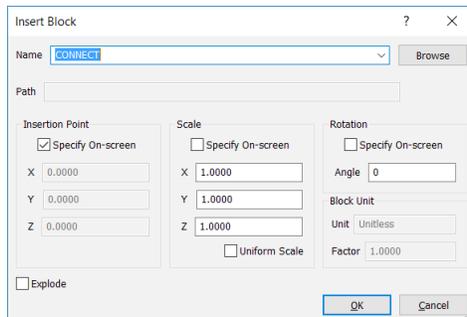
6. Click **OK**. The drawing looks no different, but behind the scenes BricsCAD has converted the three entities into a *block definition*. Block definitions are stored in the .dwg file, and are used to insert them into the drawings, making them visible. See the next tutorial.)
7. Save the drawing with **Ctrl+S**, giving it the name “Symbols.”

Inserting Blocks with Attributes

Command	Insert
Ribbon	Insert Blocks Insert Block
Menu Bar	Insert Insert Block
Alias	i

In an earlier lesson, you placed blocks with the Insert command. To see how attributes work, you will use the same command with the Connect block.

1. Enter the **Insert** command:
: insert
2. Notice the Insert Block dialog box. Check that **Name** field contains “Connect.” If not, select it from the droplist.



3. Change the following options:

Attribute Field	Value
Insertion Point	✓ Specify On-screen
Scale X	(off; scale = 1)
Rotation	(off; angle = 0)

4. Click **OK**.

Notice that the cursor is located at the block’s center. This is the base point you defined earlier with the Block command; now it is known as the “insertion point.”



5. In the command bar, BricsCAD asks for the location of the block:
Insertion point for block: (Pick a point)

Notice that BricsCAD does not prompt you for the scale or rotation angle, because you specified these parameters in the dialog box (scale = 1, rotation = 0).

6. BricsCAD does, however, prompt you to enter the attribute values. You can press **Enter** to accept the defaults:

Product <Solder Connection>: (Press **Enter**)
Stock No. <000-0000>: (Press **Enter**)



Notice that the attribute values appear next to the connection block, in exactly the same location where you placed the tags.

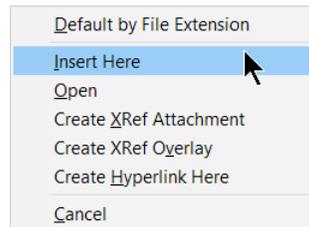
ALTERNATIVES TO THE INSERT COMMAND

In addition to the Insert command, BricsCAD has several other methods for placing blocks in drawings:

- > **Drawing Explorer** accesses blocks stored in other drawings and inserts them into drawings
- > **-Insert** command displays insertion options in the command bar, and is meant for use in scripts and macros
- > **Drag'n drop** to drag *.dwg* files from the operating system's file manager right into the drawing

When you use drag'n to drop *.dwg* files from the file manager into the drawing, BricsCAD reacts differently depending on which mouse button you hold down during the action:

- > Hold down the **left** mouse button — file opened as a drawing, like using the Open command
- > Hold down the **right** mouse button — BricsCAD gives you options through this shortcut menu:



The functions of these options are as follows:

Default by File Extension — places known file types with appropriate commands:

File Extension	Command Activated	Result
Drawing <i>.dwg</i>	Open command	Inserted as a block
Raster images	ImageAttach command	Attached as a raster image
Unknown file types	InsertObj command	Inserted as an OLE entity (Windows only)

Insert Here — places the file as a block through the -Insert command

Open — opens the file as a drawing in a new window through the Open command

Create XRef Attachment — places the file as an externally-referenced drawing through the -XRef Attach command

Create XRef Overlay — places the file as an overlain xref through the -XRef Overlay command

Create Hyperlink Here — prompts you to select one or more entities, and then attaches a URL to them that points to the file through the -HyperLink command

Cancel — cancels the operation

Exporting Data from Drawings

Adding attributes to blocks is tedious work, and so to save you time and trouble, I have prepared for you the drawing used by the next set of tutorials. It already contains all the needed blocks and attributes. Get your copy of the *electric.dwg* file from my public Dropbox folder: <https://dl.dropboxusercontent.com/u/28941239/Inside-BricsCAD-Tutorial-Files.zip>.

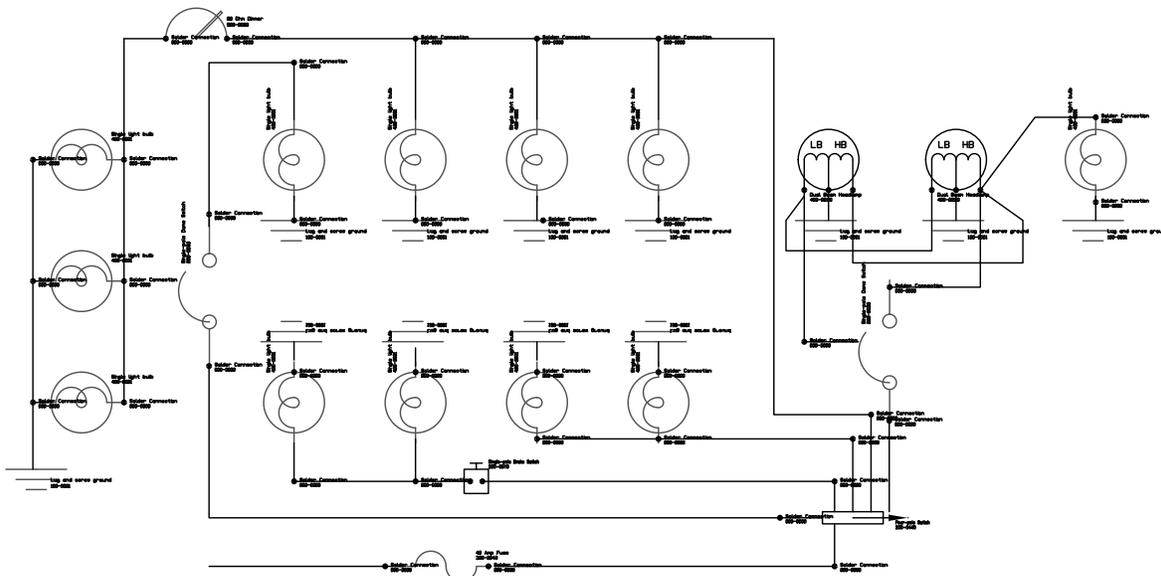
DATA EXTRACTION

Command	DataExtraction
Ribbon	...
Menu Bar	Tools Attributes Extract Data

To export data from drawings, you use the DataExtraction command. It saves the data in a CSV file, short for “comma delimited values,” which can then be read into spreadsheet and database programs for further processing. (This command is a more convenient replacement to the AttExt command, because it does not need template files.)

To extract the attribute data from the drawing, follow these steps:

1. Launch BricsCAD, and then open the *electric.dwg* file.



2. To see *all* of the attributes, including invisible ones, use the **AttDisp** command. This step is not necessary for extracting attributes, because BricsCAD extracts both visible and invisible attributes. But turning them on makes it easier for you to monitor the task.

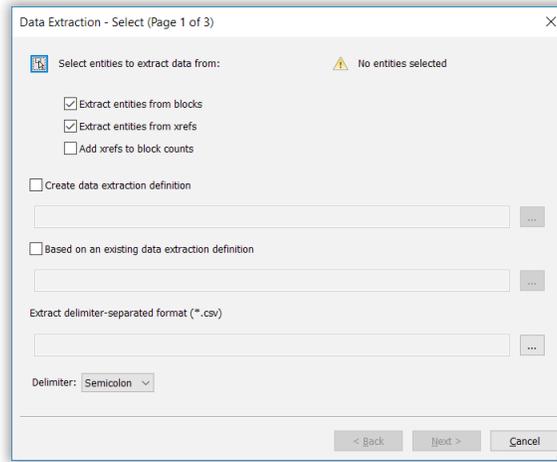
```
: attdisp  
Attribute display mode: ON/OFF/Normal <Normal>: ON
```

TIP The **AttDisp** command changes the way BricsCAD displays attribute text:

- Normal** — displays attributes, except for those set to Invisible mode.
- On** — displays all attributes, including invisible ones.
- Off** — hides all attributes.

3. Start the attribute extraction entering the **DataExtraction** command.
: **dataextraction**

Notice the Data Extraction dialog box.



4. Click the  **Select** button, and then select all of the entities in the drawing like this:
Select entities: **all**
Entities in set: 124
Select entities: (Press **Enter** to return to dialog box)

The number of entities found may differ in your drawing from mine, depending on how you drew it.

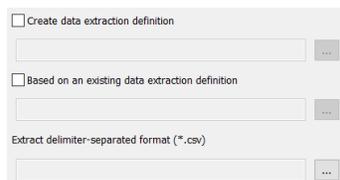
5. Back in the dialog box, notice the first set of options:



You are only interested in extracting attribute data from blocks, so turn off all options. The section now looks like this:



6. The second set of options specify files that are needed. Let's go through them one by one:



Create Data Extraction Definition — I recommend turning this option on.

When this option is on, BricsCAD saves the options you chose to a DXD file (short for “data extraction definition”). The next time you run this data extraction command, you use the following option (“Based on an existing data extraction definition”) to reuse the options, saving you from setting things up all over again. You see the benefit of this when you get to the next dialog box!

Based on an Existing Data Extraction Definition — Reuse the DXD file from a previous session; this option does not apply the first time you run this command.

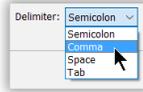
Extract Delimiter-separated Format — Specify the file name and folder location for the extracted data. Data is saved in CSV format, which separates data values with commas: one record per row, with fields separated by commas. You have to fill out this option, as BricsCAD prevents you from moving further without it by keeping the Next button deactivated.

For ease of reference, specify the following names for each file:

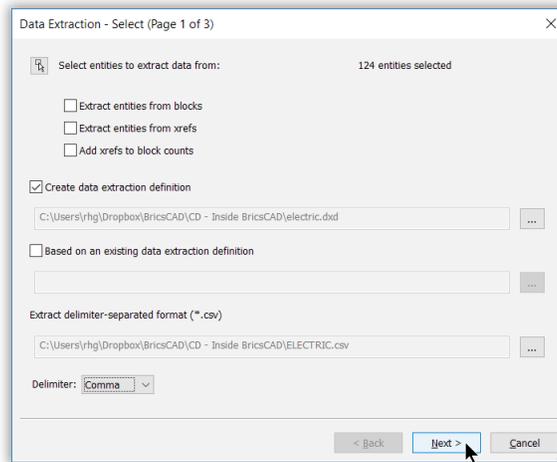
Create Data Extraction Definition	electric.dxd
Based on an Existing Data Extraction Definition	(ignore this time around)
Extract Delimiter-separated Format	electric.csv

For two of the options, click the **...** **Browse** button, and then choose a convenient folder location. Specify the 'electric' file names.

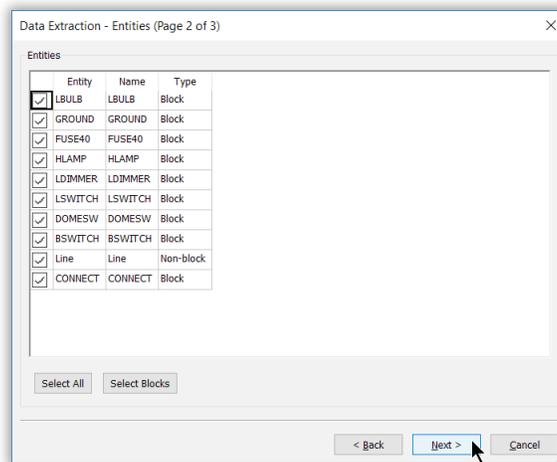
7. (NEW TO V17) Click the **Delimited** droplist, and change it to 'Comma.' A *delimiter* is punctuation that separates data values in the CSV file. It can be a semi-colon, a comma, a space, or a tab. I recommend using the comma.



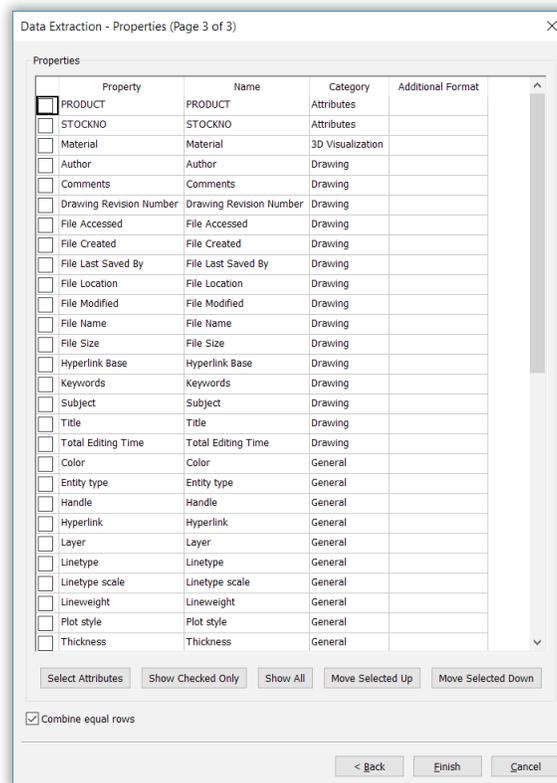
When you are done, the dialog box should look something like this.



8. Click **Next**. Notice that the Page 2 dialog box lists the blocks and line entities. You want data about all of them extracted, so leave everything as it is. Even the line entities, from which you can get the length of each; the total length tells you the amount of wiring needed. Here is the meaning of the options:
 - > **Check boxes** — to exclude an item, click the check box
 - > **Select all** — selects all items in the list (turns on all check boxes)
 - > **Select blocks** — selects just blocks, and avoids non-block entities (like the lines)



9. Click **Next** to carry on. When the Page 3 dialog box appears, it looks scary as it lists every property of every entity, as well as attributes.



The options in this dialog box have the following meaning:

- > **Checkbox** — selects an item to be exported to the CSV file
- > **Select Attributes** — selects only attributes in the list, and excludes other items
- > **Show Checked Only** — shows only the items that are selected, and hides the rest from the list
- > **Show All** — shows all items, checked and unchecked
- > **(NEW TO V17) Move Selected Up / Down** — moves the selected item(s) up and down the list; this affects the order in which items appear in the CSV file

TIPS To turn off (or on) all properties at once, select the first property in the list, hold down the **Shift** key, select the last property, and then click the checkbox.

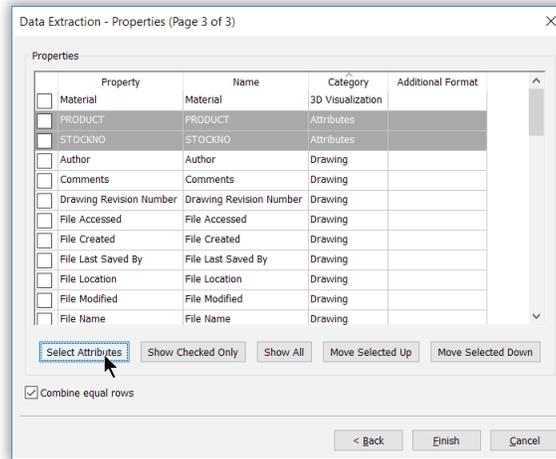
To sort the columns alphabetically, click the header of each one. When you click the invisible header of the checkboxes column, it sorts by which boxes are and are not checked.

Your big job now is to turn off all properties, except for the following ones:

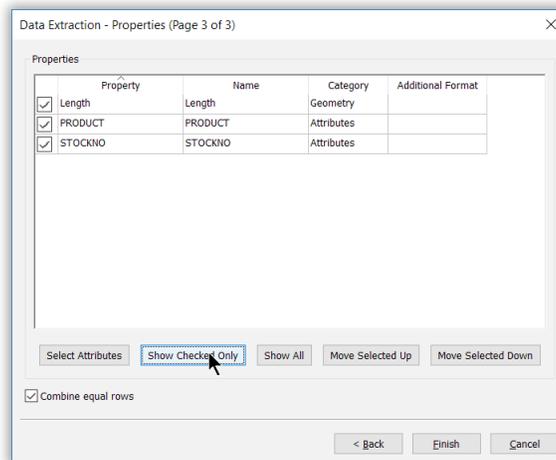
Length
Product
StockNo

To do this, follow these steps:

- a. Click the **Select Attributes** button. This highlights the attributes in the list.

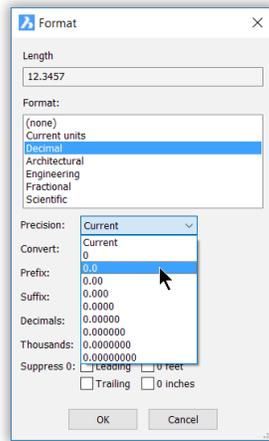


- b. Click one of the checkboxes in front of an attribute name. This action selects both of them.
- c. Click the **Property** header. This sorts the items alphabetically by property name.
- d. Go through the list until you find **Length**, and then click its checkbox.
- e. Click **Show Checked Only** to ensure that you have the correct ones.

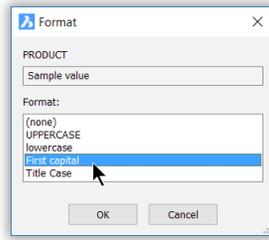


10. The dialog box has a column labeled **Additional Format** column. Its purpose is for specifying different formats for values. You don't want too much formatting, because that might confuse the spreadsheet program. Nevertheless, here is how to use it:
 - a. Next to "Length," click the blank area under **Additional Format**. Notice the dialog box.

The **Length** field shows you what the format looks like, either the default or else after you make changes. In the figure shown, I changed the “Format” to **Decimal** and the “Precision” to **0.0**. I was tempted to specify **mm** for “Suffix,” but then I realized that would make the spreadsheet unhappy — it would think the number was text.



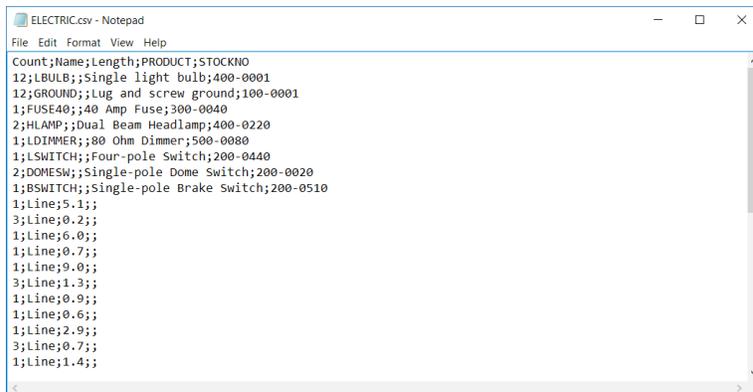
- b. The Product and StockNo attributes are text, and I changed the “Format” to **First capital**.



- c. Click **OK** to exit the Format dialog box.

TIP The formatting codes used by the DataExtraction command are the same ones used for field text.

11. Click **Finish**. After a second or two, the results are deposited in the *electric.csv* file.
12. To see the result, open the *electric.csv* file with a text editor. The content should look something similar to that illustrated below.



For each block and line in the Electric drawing, BricsCAD lists the following pieces of data:

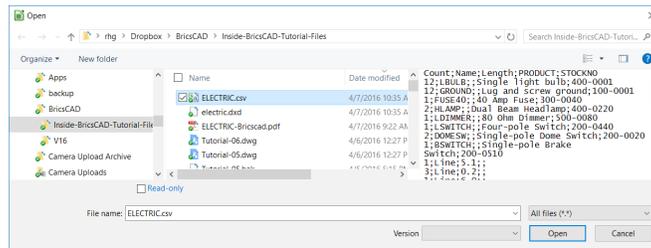
Field	Meaning	Example Value
Count	Number of occurrences in the drawing	12 (light bulbs)
Name	Name of the block or entity	LBULB
Length	Length of line (does not apply to attributes)	5.1
Product	Block's attribute value (does not apply to lines)	Single light bulb
StockNo	Block's attribute value (does not apply to lines)	400-0001

You've created a rudimentary bill of material, which can be printed out on your printer or imported to a spreadsheet — as described next. Later you learn how to bring the spreadsheet into the drawing as a bill of materials.

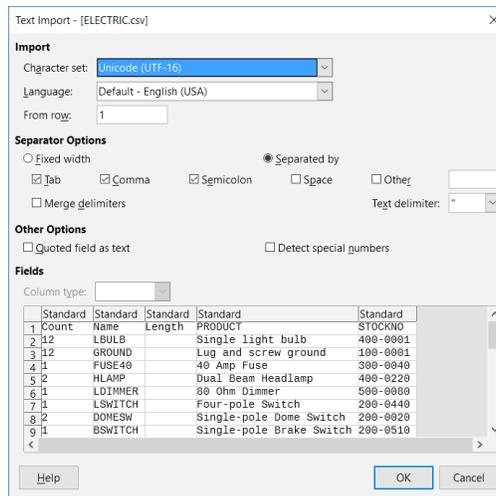
IMPORTING DATA FILES INTO SPREADSHEETS

To format the raw data and perform calculations, you can import the exported data into a spreadsheet. Shown in this tutorial is LibreOffice Calc, part of a free software package available from <https://www.libreoffice.org/discover/calc/>. Other spreadsheet programs, such as Excel, may have analogous steps.

1. Launch the spreadsheet program.
2. To open the data file correctly, please follow these steps:
 - a. From the spreadsheet's **File** menu, select **Open** to display the Open dialog box.
 - b. Navigate to the folder holding the *electric.csv* file.



- c. Select the *electric.csv* file, and then click **Open**.
3. Notice that LibreOffice displays the Text Import dialog box, which allows you to specify the format of files being imported.

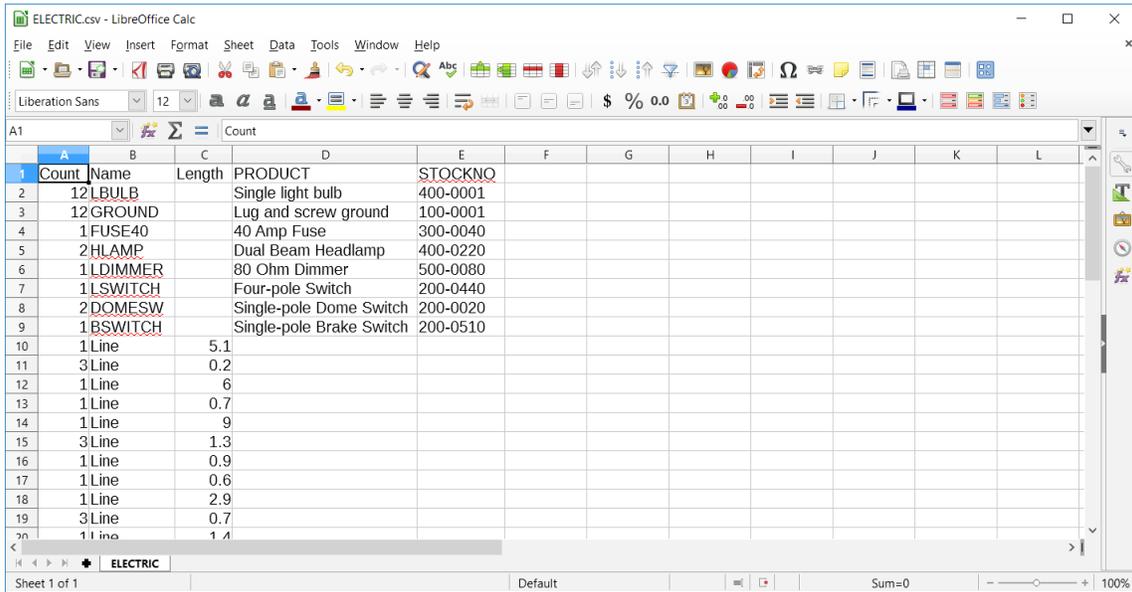


All the default options are sufficient, but make sure the following ones are set:

Import
From row: **1**

Separator Options
Separated by: **Tab Comma Semicolon**

- Click **OK**. Notice that Excel loads the *electric.csv* file, and then displays each field in its own column. If it is necessary to adjust the column widths, select the four columns, then choose from the menu bar **Format | Column | Optimal Width**.



- If you wish, add price and extension fields, a totals row, and format the text for lovely output. For example, I used the SUM() function to find the total length of wires (lines).

Count	Name	Length	PRODUCT	STOCKNO
26	1 Line	3.2		
27	1 Line	5.8		
28	4 Line	0.5		
29	1 Line	5.2		
30	2 Line	1.2		
31	3 Line	1.6		
32	1 Line	6.2		
33	3 Line	1.5		
34	7 Line	0.4		
35	3 Line	0.3		
36	1 Line	0.9		
37	1 Line	2.8		
38	2 Line	1		
39	1 Line	2.4		
40	1 Line	0.9		
41				
42	Total Length	83.6		
43				

- Remember to save the spreadsheet file.

Summary

You learned how to export attribute data to spreadsheets, and then bring formatted spreadsheet data back into BricsCAD. You also learned about creating, populating, and editing tables.

Next, you learn about a more sophisticated 2D entity known as the region.

Modeling 2D Regions and Booleans

Here, you learn how to increase your CAD efficiency by working with regions and Boolean operations. These two BricsCAD features let you construct complex 2D shapes, and then analyze them — and BricsCAD does this far faster than if you were to perform the calculations by hand.

(The Region command is available in the Pro and Platinum editions of BricsCAD only.)

IN THIS CHAPTER

- Converting collections of entities into regions
- Using point filters to find coordinates
- Applying Boolean operations to regions
- Finding the mass properties of regions

KEY TERMS IN THIS CHAPTER

Boolean — refers to logical operations, such as AND, OR, and NOT

Mass property — reports the properties of a mass, such as area, centroid, and radius of gyration

Point filter — returns a single coordinate

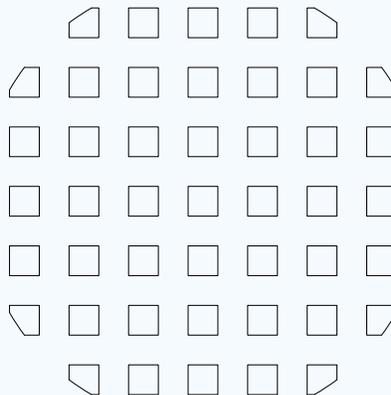
Region — consists of a closed 2D area

NEW COMMANDS

Command	Shortcut	Menu Bar	Ribbon
DelObj
Intersect	in	Modify Solid Editing Intersect	Solid Edit Intersect
MassProp	...	Tools Inquiry Region/Mass Properties	Tools Inquiry Mass Properties
Region	reg	Draw Region	...
Subtract	su	Modify Solid Editing Subtract	Solid Edit Subtract
Union	uni	Modify Solid Editing Union	Solid Edit Union

THIS CHAPTER'S DRAWING

By the end of this lesson, your drawing will look like this one:

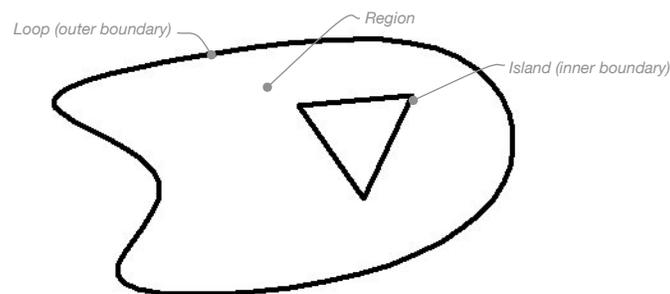


RegionTutorial drawing

About Regions

BricsCAD works with *regions*. These are closed 2D areas of almost any shape. Regions are interesting, because two or more can be combined using *Boolean* operations. These operations join, intersect, and subtract one region from the other. This lets you create a single entity with holes in it, and analyze its physical properties.

Technically, regions are made of ACIS entities, the same technology used for 3D solid models. The outer boundary of the region goes by the name of a “loop.” The holes inside of regions are called “islands.” The boundaries and islands can have any kind of shape, from a as simple as triangle to flowing curves.



HOW TO CREATE REGIONS

In a curious twist of fate, BricsCAD cannot create regions from scratch, such as like using the PLine command to directly draw polylines. Rather, BricsCAD converts existing closed areas into regions. To do this, BricsCAD provides you with two commands, **Boundary** and **Region**. They are subtly different:

- > **Boundary** command prompts you to pick a point inside a closed area, and then generates a region or a polyline; it is a subset of the Hatch command and displays a dialog box
- > **Region** command prompts you to select entities and/or boundaries that make up a closed area, and then generates a region entity; it operates at the command line

In the end, both create a region entity out of a closed area (made of one or more entities). So it takes two steps to create a region:

Step 1: Draw the shape using drawing commands such as **PLine**, **Arc**, and **Circle**. To create a region successfully, the shape must be closed. Self-intersecting curves are made into multiple regions. See figure below for examples. Left to right, you end up with one region, no region, and three regions.



Left: Closed non-self-intersecting shape; **center:** Not-closed shape cannot become a region; **right:** Self-intersecting curve becomes multiple regions

Step 2: Convert the shape into a region with the **Boundary** or **Region** commands. The result of the two commands differs for the shapes in the figure shown below:

Command	Closed Entity	Open Entity	Self-intersecting Entity
			
Boundary:	1 region	0 or 1 regions *	1 region per pick
Region:	1 region	0 regions	3 regions with 1 pick

The Boundary command can make a region from an open area if the gap tolerance is sufficiently large.

The two commands do not combine non-overlapping entities into a single region entity. For instance, select three circles, each an inch apart, and they become three regions. (This behavior may change in a future release of the software.)

Both commands leave the source entities in place. Use the **DelObj** (short for “delete objects”) variable to change this behavior in the Region command. Set it to -2 or 2 to have Region erase the source entities automatically:

DelObj	Meaning
-2	Prompts if all entities should be erased
-1	Prompts if profile entities should be erased during commands like Extrude, Revolve, and Sweep
0	Leaves source entities in place
1 (default value)	Erases entities only during commands like Extrude, Revolve, and Sweep
2	Erases source entities with other commands, such as Region

TIPS The **U** command can be used to change region entities back to their original components.

Regions are always closed; there are no open regions.

After entities are converted to regions, they look no different, but they do have different properties, of which you learn about later. In the following tutorials, you first use the Boundary command and then the Region command to see how they work. Later, you apply boolean operations.

TUTORIAL: CREATING BOUNDARIES

Available in all versions of BricsCAD

Command	Boundary
Ribbon	Draw hatch Boundary
Menu Bar	Draw Boundary Polyline
Alias	B0

The Boundary command draws a boundary around the inside of selected areas and/or entities. It draws them as polylines or regions — your choice, although in this tutorial you want regions.

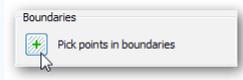
(This command is a subset of the Hatch command. To place hatch patterns correctly, the Hatch command first draws invisible boundaries around the areas to be hatched; the hatching is applied inside the boundary, and then the boundary is optionally erased. The Boundary command draws the same kind of outline, but keeps it.)

QUICK SUMMARY OF BOUNDARY OPTIONS

The Boundary dialog box contains options that control how boundaries are created.

BOUNDARIES

The Pick Points button dismisses the dialog box temporarily, and the prompts you to pick a point in the drawing: 'Select a point to define a boundary or hatch area.' You can pick inside more than one boundary.



BOUNDARY SET

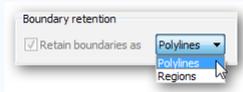
The Boundary Set section determines where BricsCAD should look for entities from which to create boundaries.



The default is the entire current viewport, but you can change this with the **New** button to smaller areas, such as the existing selection set or the current viewport. For instance, after you click the **New** button and choose a number of entities, the boundary seeking algorithm limits its work to the entities you selected.

BOUNDARY RETENTION

The Boundary Retention option makes the boundary from a polyline or region.



ISLANDS

When entities contains other entities (a.k.a. "islands"), then you can tell the boundary-seeking algorithm to include or ignore them.



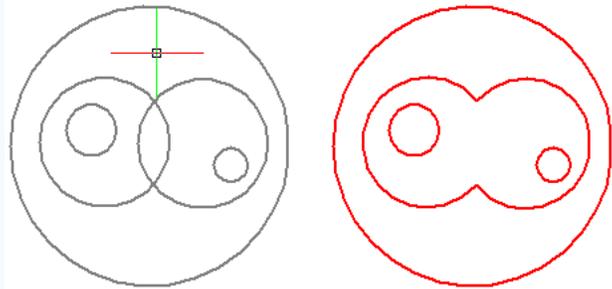
The circles shown below illustrate the difference between the Nested, Outer, and Ignore options. Notice the location of the cursor, and that the pick point is the same in each case.

TIP Entities created by Boundary can look invisible, because they are created on the current layer. To see them, create a new layer with a different color.

Continued...

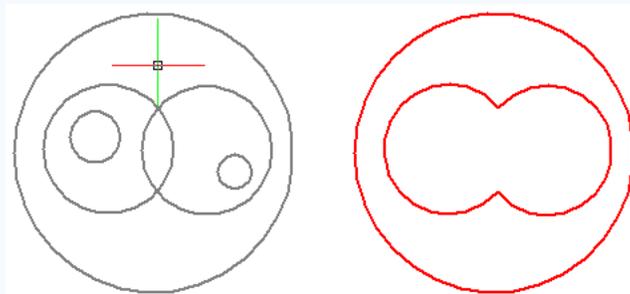
...continued.

Nested — all entities (the circles) form boundaries, as shown in red at the right.



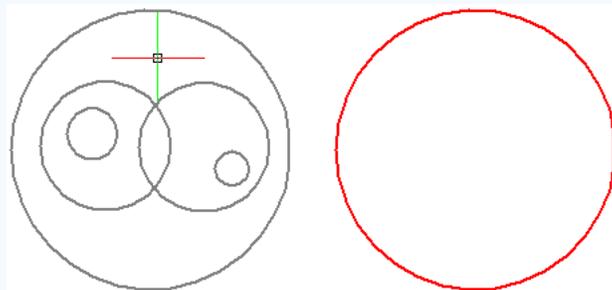
Left: Original circles
Right: All circles turned into boundaries

Outer — the outermost circles form boundaries (shown in red below); the innermost circles are ignored.



Left: Original circles
Right: Circles nearest to pick point turned into boundaries

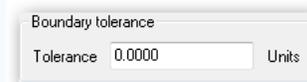
Ignore — only the outermost circle is used to form the boundary; the others are ignored.



Left: Original circles
Right: Only outermost circle turned into a boundary; other circles ignored

BOUNDARY TOLERANCE

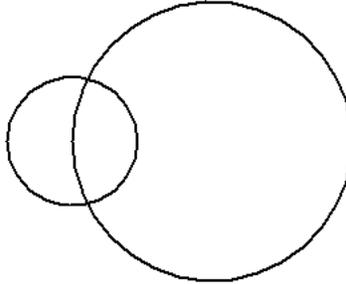
The Boundary Tolerance option specifies the acceptable size of gaps. For example, when you set the tolerance to 0.1", then gaps as large as 0.1" will be ignored (or bridged) by the boundary-seeking algorithm.



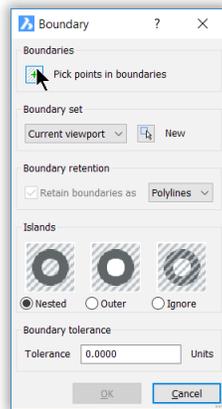
The Boundary command displays a dialog box that looks like a simplified version of the Hatch and Gradient dialog box — the hatching components are missing.

To create boundaries as regions, follow these steps:

1. Start BricsCAD with a new drawing. The workspace or template does not matter.
2. Use the **Circle** command to draw two overlapping circles, as illustrated below. The size does not matter.

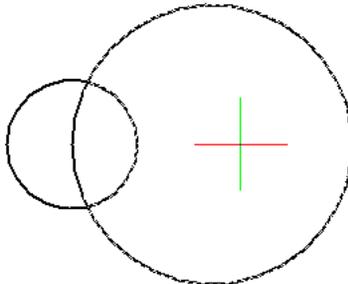


3. Start the **Boundary** command. Notice the Boundary dialog box; the boxed text on a nearby page explains its functions.

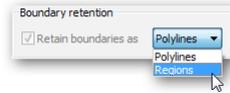


4. Click the **Pick Points in Boundaries**  button. It lets you pick a point inside the circles; BricsCAD then determines the boundary(ies) automatically, based on your pick point.
5. Notice that the dialog box disappears (temporarily), and that you are prompted at the command bar.
Pick a point to define a boundary or hatch area or [Select entities/Undo]: (Pick inside one of the circles)

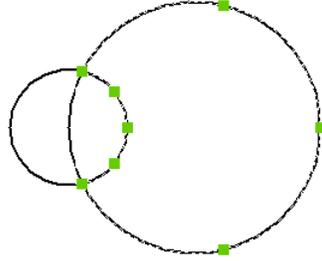
Pick a point *inside* one of the circles, as illustrated below. It doesn't matter which one, just don't select a circle itself.



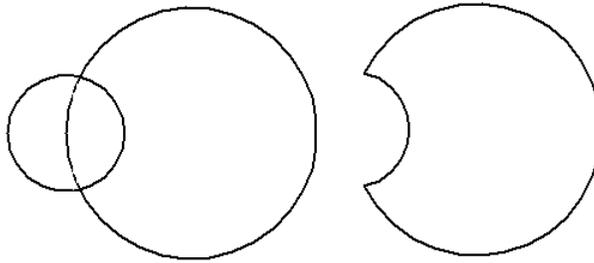
6. Press **Enter** to return to the dialog box:
Pick a point to define a boundary or hatch area or [Select entities/Undo]: (Press **ENTER**)
7. In the **Boundary Retention** area, change the boundary type from Polyline to “Regions”:



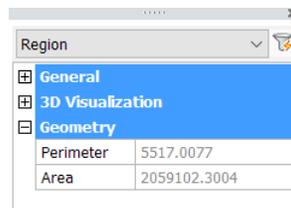
8. And then click **OK** to exit the dialog box.
9. The circles look no different, yet BricsCAD has drawn an entity inside one of them. To see it, drag it away from the circles, like this:
 - a. Click the circle that surrounds your pick point made during the Boundary command. Notice that when BricsCAD highlights it, the highlight looks like a crescent moon. This moon is the region entity.



- b. Drag the region away from the two circles. Notice that it is an independent entity; the two original circles remain in place.



10. Use the Properties panel to determine that the entity is in fact a region. Notice that the Properties panel also reports the area and perimeter of the region, a very useful side effect!



As you can see from the result of this tutorial, regions are a handy way to create unusual shapes as a single entity. In the next tutorial, you use the Region command.

TIPS After the boundary (region or polyline) is created, the Boundary command does delete source entities, even when the **DelObj** variable (short for “delete entity”) is set to 2 to force deletion.

The command line version is **-Boundary**, and is meant for use by scripts and programming routines.

Yes, you can use Boundary on regions, and Region on polylines made by the Boundary command, and Region on regions. In all cases, an identical copy is made of the region or polyline.

Boolean Operations

Available only in the Pro and Platinum versions of BricsCAD

When drawings contain two or more regions or 3D solids, you can perform *boolean* operations on them. These operations allow you to combine separate regions and 3D solids and make them one — or to separate one into many. “Boolean” is a term from mathematics that describes the logical operations that you may have learned in school, such as AND, OR, and NOT.

BricsCAD provides three boolean commands to combine and separate regions and solids. See the details in the text box on a nearby page. Here are the commands and what they do:

- ▶ **Union** command combines two or more regions into one region (AND operation)
- ▶ **Intersect** command finds the area common to two or more overlapping regions (OR operation)
- ▶ **Subtract** command removes the area of one or more regions from other overlapping ones (NOT operation)

Technically, these commands employ ACIS modeling to perform their work, which is why they are unfortunately not available in BricsCAD Classic.

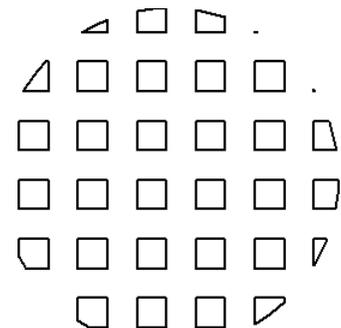
Together with Boundary and Regions commands, these boolean commands are be employed to construct complex shapes with BricsCAD. Indeed, boolean operations work only with region entities and 3D solids. They do not work with regular 2D and 3D entities like lines and 3D meshes; if you want to, you need to first convert these regular entities into regions or 3D solids.

The trickiest of the boolean operations is subtraction, which is why get to employ it in the following tutorial.

TUTORIAL: CREATING A WAFFLE SHAPE

Commands	Array and -Array
Ribbon	Change Array
Menu Bar	Modify 2D Array
Aliases	AR and -AR
Command	Region
Ribbon	Model Solids Region
Menu Bar	Draw Region
Alias	REG
Command	Subtract
Ribbon	Model SOLID Editing Subtract
Menu Bar	Modify Solids Subtract
Alias	SU

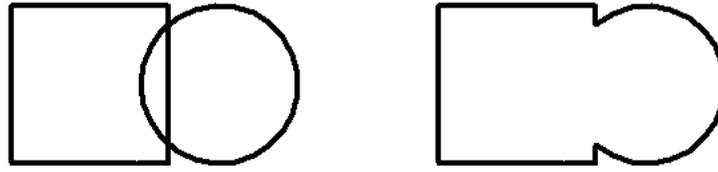
In this tutorial, you create a waffle shape as illustrated below. After drawing some intersecting rectangles, you will apply the Subtract command to form the waffle shape. To determine the net area, you find the mass properties of the shape; this would be very difficult to do without regions.



QUICK SUMMARY OF BOOLEAN OPERATIONS

UNION COMMAND

Union joins two or more regions into one. This allows you to create complex entities from simple ones.



Left: Original objects (two regions); right: square unioned with circle as a single object

In boolean terms, the union operation returns everything in region #1 OR in #2.

INTERSECT COMMAND

Intersect removes all but the overlapping portions of two or more regions. This allows you to find the areas in common between entities.



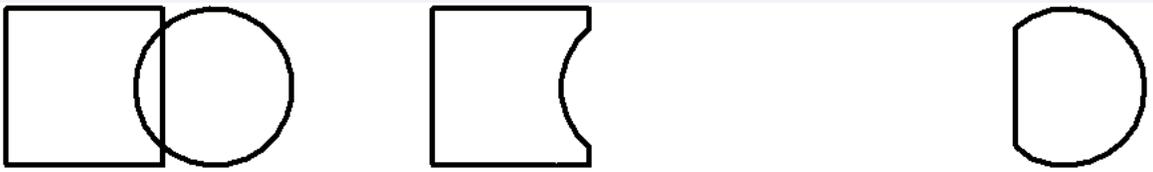
Left: Original objects (two regions); right: circle intersected with square

In boolean terms, the intersection operation returns everything that's in region #1 AND in region #2.

SUBTRACT COMMAND

Subtract subtracts one set of regions from another. This allows you to remove parts of entities.

There are two outcomes possible from subtraction. The outcome depends on the order in which you select the regions, as shown by the figure below. In the center result, the circle was removed from the square; in the right result, the opposite occurred: the square was removed from the circle. (Selection order does not matter for the union and intersect operations.)



Left: Original objects (two regions); center: circle subtracted from square; right: square subtracted from circle

In boolean terms, the subtraction operation returns everything that's in region #1 but NOT in region #2.

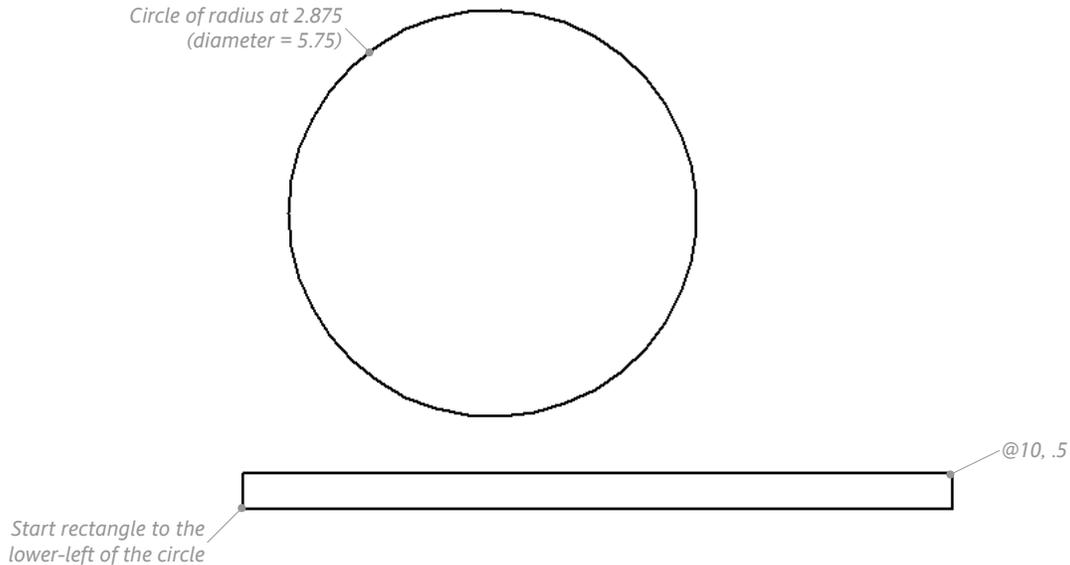
1. Start BricsCAD with a new drawing.
2. Using the **Circle** command, draw a circle with a radius of **2.875** units at the origin (0,0), as follows:

```

: circle
2Point/3Point/RadTanTan/Arc/Multiple/<Center of circle>: 0,0
Diameter/<Radius>: 2.875

```

(If necessary, use the **Zoom Extents** command to see the entire circle, and then employ **Zoom 0.5x** to get some space around it.)



3. Now draw a rectangle with the **Rectang** command:

```

: rectang

```

The rectangle's first corner is not crucial, except that it should be to the lower-left of the circle. See the figure above.

```

Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/<Select first corner of
rectangle>: qua
Snap to quadrant of: (Pick a point to the Lower-Left of the circle.)

```

Enter relative coordinates to position the other corner. The width (x) does not matter, but the height (y) should be 0.5 units. I'm going to use a width of 10 units. I'll specify the other corner using relative coordinates, as follows:

```

Other corner of rectangle: @10, .5

```

4. The next step is to convert the circle and rectangle to region entities using the **Region** command:

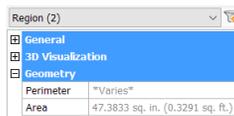
```

: region
Select entities: all
Select entities: (Press ENTER to end entity selection.)
2 region(s) created

```

After the entities are converted to regions, they look no different. The circle and rectangle are, however, now *region* entities that are circular and rectangular in shape. The Boolean operations that you carry out later in this tutorial would not work on actual circles and rectangles, and so they had to be converted to regions.

If necessary, use the Properties panel to convince yourself the entities are now regions!



5. Create copies of the rectangle region to cover the circle.
Now, you could use the Copy command for this task, but it is much faster to deploy the **-Array** command. In this tutorial, you array the rectangle twice. The first time you array the rectangle vertically; the second time, you rotate one by 90 degrees (with the Mirror command) and then array it horizontally.
To start the -Array command, enter the **-ar** alias:
: -ar

6. Choose the rectangle:
Select entities to array: (Pick the rectangle)
Select entities to array: (Press **Enter** to end entity selection)

This command can make arrays that are polar (circular) or rectangular, but you want rectangular:

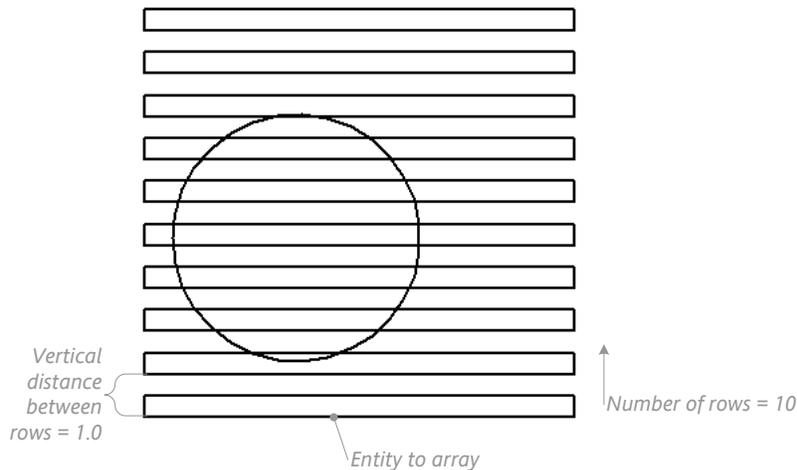
Type of array: Polar/<Rectangular>: (Press **Enter** to accept default, Rectangular)

The number of rows in the first array is somewhat arbitrary, because you want to cover the entire circle with rectangles. If there are too many, you can just erase the extras; if too few, then it's a pain to restart the Array command, so draw too many in the first place!

Number of rows in the array <1>: **10**
Number of columns <1>: (Press **Enter** to accept default, 1)

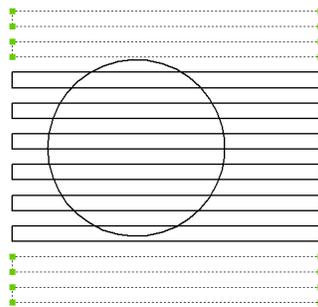
The distance between rows is 1 unit:
Vertical distance between rows, or spacing rectangle: **1**

Notice that BricsCAD instantly creates ten copies of the rectangle.



(The reason I use command-line oriented -Array command instead of the dialog box-toting Array is because it can be faster to enter values at the command line than hunting through a dialog box.)

7. Use the **Erase** command to remove superfluous rows, in other words, those that don't cover the circle.



8. Now you want a second set of rectangles at 90 degrees. Create the first one by mirroring an existing rectangle with the **Mirror** command:

: **mirror**

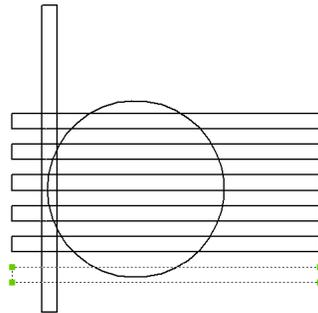
Select the lowest rectangle, like this:

Select entities to mirror: *(Pick the rectangle, indicated by the figure below)*
 Select entities to mirror: *(Press ENTER to end entity selection)*

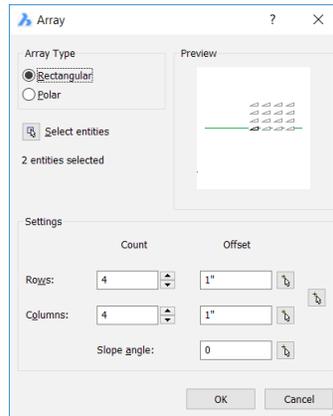
BricsCAD needs an imaginary line called the “mirror line” about which to mirror the rectangle. The placement of the mirror line is not crucial, as long as it is at 45 degrees to the rectangle. The easy way to do this is to employ the following relative polar coordinates: @1<45 as the mirror line’s end point:

Start of mirror line: **0,0**
 End of mirror line: **@1<45**

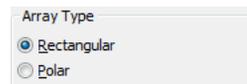
Delete the original entities? <N> *(Press ENTER to accept default, N)*



9. With the vertical rectangle in place, start the **Array** command. This is the dialog box version of the command.



- a. For Array Type, ensure Rectangular is chosen:



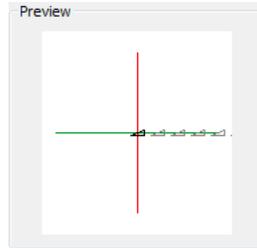
- b. Click  **Select Entities** and then enter ‘L’ to select the last-chosen entity:
 Select entities to array: **1**
 Entities in set: **1**
 Select entities to array: *(Press Enter to return to the dialog box)*

- c. Specify the number of copies to make:

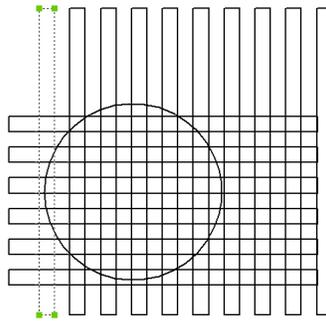
	Count	Offset
Rows:	1	1
Columns:	10	1
Slope angle:	0.0000	

Rows Count: 1
Columns Count: 10
Column Offset: 1

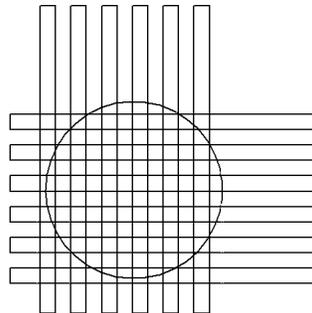
Notice that the preview window shows several columns, one row high.



- d. Click **OK**, and BricsCAD instantly creates ten copies of the vertical rectangle.



10. Erase the rectangles that don't lie on the circle.

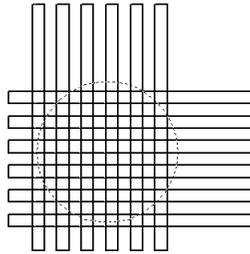


11. Now you finally get to use a Boolean. With the **Subtract** command, remove the rectangular regions from the circular region, as follows:

: **subtract**

Select ACIS entity to subtract from: (Pick circle.)

Select ACIS entity to subtract from: (Press **ENTER** to end "From" entity selection.)

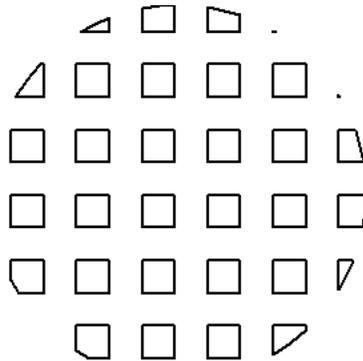


To pick all of the rectangles more quickly, follow these steps:

- a. First, use **All** selection mode to select everything in the drawing.
Select ACIS entities to subtract: **all**
- b. Second, use the **R** (remove) option to remove just the circle from the selection set.
Select ACIS entities to subtract: **r**
Subtract entities from selection set: (Pick circle.)
Subtract entities from selection set: (Press **ENTER** to end the command.)

That's just two selections, instead of 12!

Notice the result: the **Subtract** command removes the overlapping regions, producing the waffle effect — yet those 32 parts are in reality a *single* entity. This again shows you one benefit of working with regions.

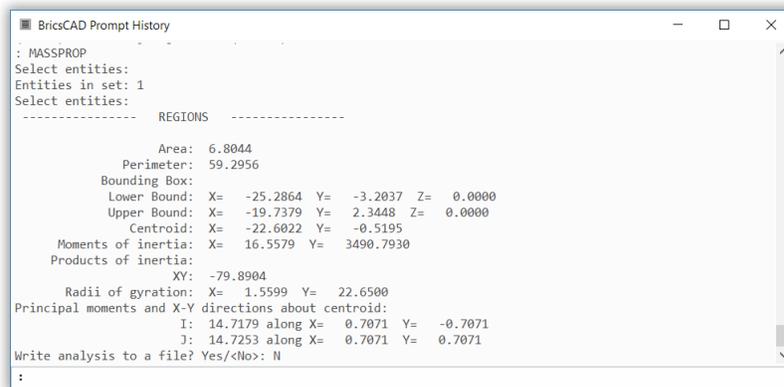


Measuring Regions

Command MassProp
Menu Bar Tools | Inquiry | Mass Properties

You've seen in the first two tutorials how region entities are unique in BricsCAD, by consisting of unusual shapes and even a series of seemingly disconnected shapes. Another benefit we get from regions is that we can easily determine their physical properties — especially easy for ones made of many separate parts, like the waffle shape. If the waffle were instead made of 32 regular 2D entities, then we would have to add up the areas of all individual parts, and then find the total. In contrast, the waffle region is a single entity, and so easy to measure.

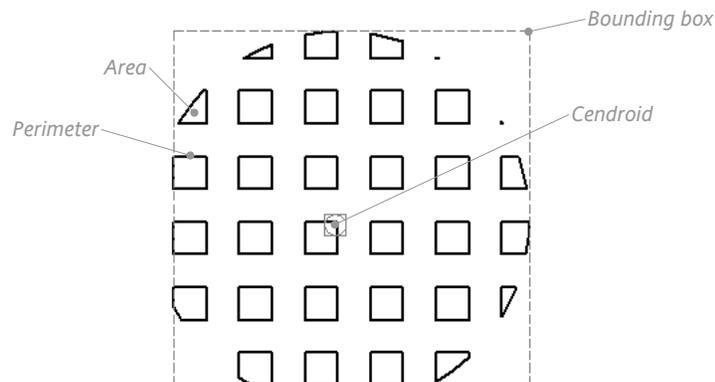
1. To find the total area of the waffle pattern, use the **MassProp** command, as follows:
: massprop
Select entities: (Pick waffle shape)
Select entities: (Press ENTER)
2. BricsCAD displays the results of the analysis in the text window. If necessary, press **F2**:



So now you know: The area of the waffle pattern is 6.8044 square units, precisely. The value you see for your region may differ due to the way you placed the rectangles.

ABOUT MASS PROPERTIES

The MassProp command uses terms that have the following meaning:



- > **Area** — cumulative area of all 24 waffles.
- > **Perimeter** — cumulate perimeters of all 24 waffles.
- > **Bounding Box: Lower Bound and Upper Bound** — rectangular limits of the region; the coordinates describe the lower-left and upper-right corners of the corners of an imaginary rectangle that tightly encloses the region.
- > **Centroid** — center of mass of the region's 24 waffles, excluding the open areas.
- > **Moments of inertia** — measure of the region's resistance to angular acceleration.
- > **Product of Inertia: XY** — measures the region's resistance to change in rotation.
- > **Radius of Gyration** — distance from the axis that the entire region can be concentrated to obtain the same mass moment of inertia.
- > **Principle moments and X-Y directions about centroid** — equivalent to torque.

Summary

This lesson introduced you to regions and Boolean operations. These concepts extend to working with 3D models in BricsCAD Professional and other 3D software packages.

Direct 3D Modeling & Editing

Here, you learn how to create 3D models using traditional and modern editing techniques. Traditional techniques use commands to create and manipulate 3D models; modern ones manipulate 2D and 3D entities directly. Direct modeling and editing is available in the Pro and Platinum editions of BricsCAD.

IN THIS CHAPTER

- Learning about 3D solid models
- Extruding profiles (2D entities) into bodies
- Rotating 3D viewpoints
- Subtracting 3D entities from one another
- Modeling with direct editing
- Aligning UCSes
- Adding fillets to 3D edges

KEY TERMS IN THIS CHAPTER

Direct modeling and editing — creates and edits 3D parts directly, without entering commands

Profiles — describes 2D entities that define 3D parts.

Quad — multi-tiled cursor with common commands

Shell — refers to a hollowed out 3D solid model

Sub entities — describes entities that make up 3D solid models, such as faces and edges

Union — joins two or more solid models into a single body

USEFUL ABBREVIATIONS

QUAD Quad cursor

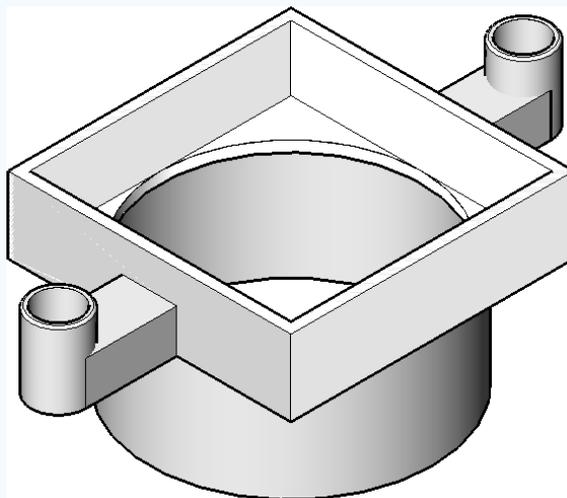
SUB Sub-entity selection

NEW COMMANDS

Command	Shortcut	Menu Bar	Ribbon Tab
DelObj
Extrude	Ext	Model 3D Solids Extrude	Solid Create
QuadDisplay	F12
SolidEdit	...	Model 3D Solid Editing Shell	Solid Edit
UCS)

THIS CHAPTER'S MODEL

By the end of this lesson, your 3D model will look like this one:



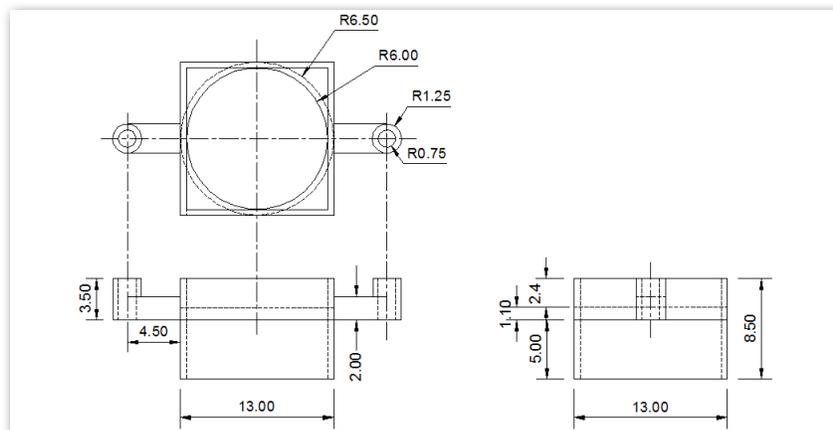
Focus-bracket drawing 3d-part.dwg

About 3D Solid Models

BricsCAD Pro and Platinum allow you to create 3D models from *solids*. Solids are 3D entities that are solid on the inside, which distinguishes them from other 3D entities that are not solid, such as surfaces and polyface meshes. Because they are solid through and through, models made from 3D solids accurately mimic real world entities. For instance, using third-party tools like FEA (finite element analysis) and CFD (computational fluid dynamic), they can be tested on the computer to ensure they work properly once manufactured.

BricsCAD has several methods for creating 3D models from solids, and here you learn about two of them: using traditional commands and direct modeling.

In these tutorials, you model the focus ring mount for a small webcam. The 2D plans are illustrated below and finished 3D model is shown on the facing page. The 2D drawings illustrate a further benefit to 3D: it is easier for non-experts to visualize products in 3D than in 2D. (The 2D drawing is *Tutorial-10-2D.dwg* and the completed 3D model is *3d-part.dwg*, and both are available through my public Dropbox folder: <https://dl.dropboxusercontent.com/u/28941239/Inside-BricsCAD-Tutorial-Files.zip>.)



PLANNING AHEAD

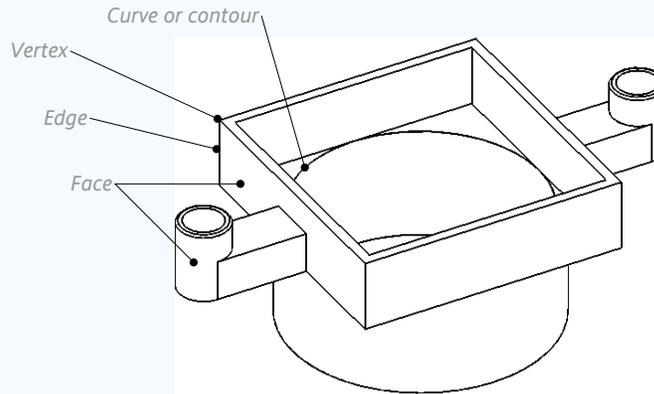
There are several differences between 2D drafting and 3D modeling. One difference is that 3D has a limited number of ways to create the parts that make up a 3D model, whereas there are many ways to draw in 2D. For instance, to create a box shape in 3D, you can use the Box command or else extrude a rectangle, and that's pretty much it; in 2D, by contrast, you can draw a rectangle with lines, polylines, or traces, or with the Rectangle or Polygon commands, and so on.

Another difference is that in 3D modeling you must plan ahead how to construct the 3D model; this differs from 2D, where you can simply start drawing. Now, there is a bit of a Catch-22 here: you can only become effective in planning ahead after you gain experience with 3D modeling. So this chapter shows you some of the tricks in putting together 3D models.

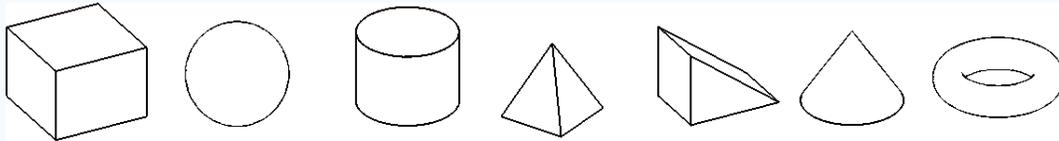
ELEMENTS OF 3D MODELS

Three-dimensional modeling employs a jargon of its own. Here is a visual reference to some common terms.

PARTS OF A 3D MODEL

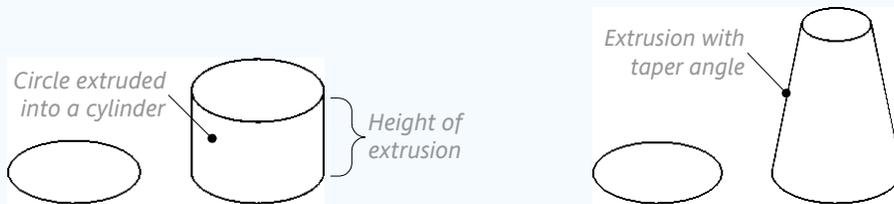


Primitive (basic) 3D parts are made with (left to right) Box, Sphere, Cylinder, Pyramid, Wedge, and Torus commands.



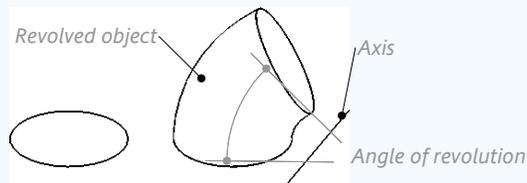
CREATING 3D MODELS FROM 2D

Applying the **Extrude** command to 2D entities:



Left: Extrude command applied to a 2D circle; **right:** Extrude with 5-degree taper

Applying the **Revolve** command to 2D entities:



Revolve command extrudes circle by 45 degrees

Planning Ahead by Deconstructing 3D

The way to plan ahead is to deconstruct the model. Here are some tips:



- ▶ Look at it to see what kinds of *sub-parts* exist. “Sub-parts” are portions of the model that look like boxes and cylinders; holes are formed from cylinders.
- ▶ Figure out which common operations can be applied. For instance, parts can be repeated (made once then copied), mirrored (made once then copy-mirrored), or are unique (made once).
- ▶ There are some rounded edges that could be made with the Fillet command.
- ▶ Use 2D drawings to determine the dimensions of the part.

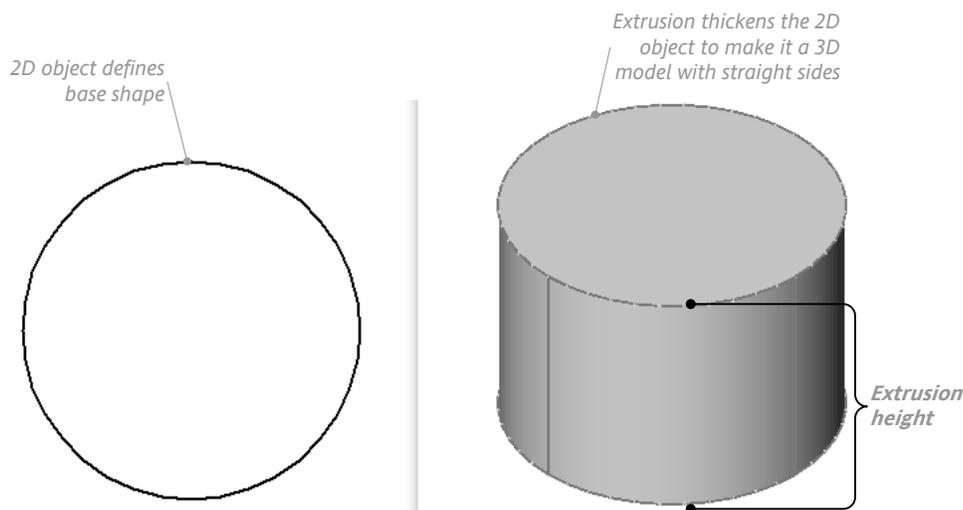
For example, look at the photograph and identify sub-parts (a.k.a. “primitives”) that could be modeled first. Notice that there are some square ones and some round ones. How would these be modeled? (With the Box and Cylinder commands.)

Because the sides are parallel, much of this part could be constructed from *extrusions*. An “extrusion” takes a 2D outline and then thickens to make it 3D.

Traditional Solid Modeling Commands

In this set of tutorials, you use variations on commands to model the focus ring holder of a webcam. For instance, in the first tutorial you use the traditional Extrude command, which in CAD dates back to the late 1980s; in a later tutorial, you use a newer approach to do the same thing. It is known as “direct modeling.”

To create an extrusion, you first draw the outline of the part in 2D (on the x,y -plane), and then use the Extrude command to thicken it in the z -direction. See the figure below for how this works. Extrusions always have *straight* or slanted sides. (If you want a part to have curved sides, then you would *revolve* the 2D entity with the Revolve command.)



Left: 2D circle defining the diameter of the cylinder; **right:** Circle extruded to become a 3D cylinder

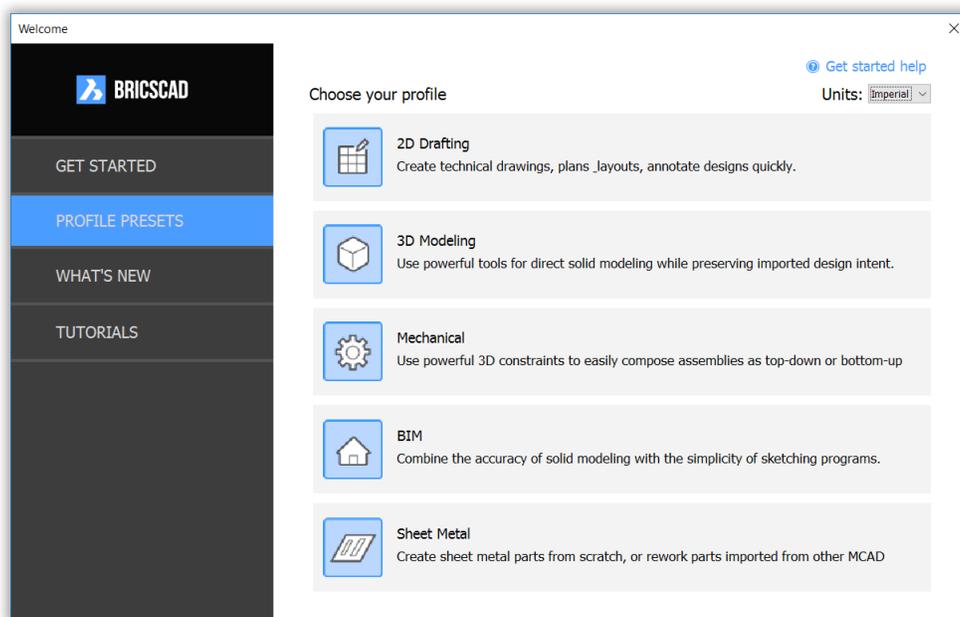
TRADITIONAL EXTRUSION METHOD

Command	Extrude
Ribbon	Solid Create Extrude
Menu	Models 3D Solids Extrude
Alias	EXT
Command	SolidEdit
Ribbon	Solid Edit Shell
Menu	Model 3D Solid Editing Shell
Alias	...

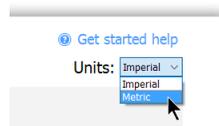
Preparing the Drawings

Here are the steps you need to take to prepare the drawing for 3D modeling.

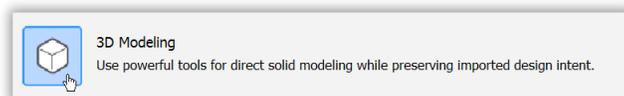
1. Start BricsCAD.
2. Notice the Get Started dialog box. Click **Profile Presets**.



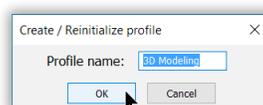
3. Change the **Units** to “Metric.”



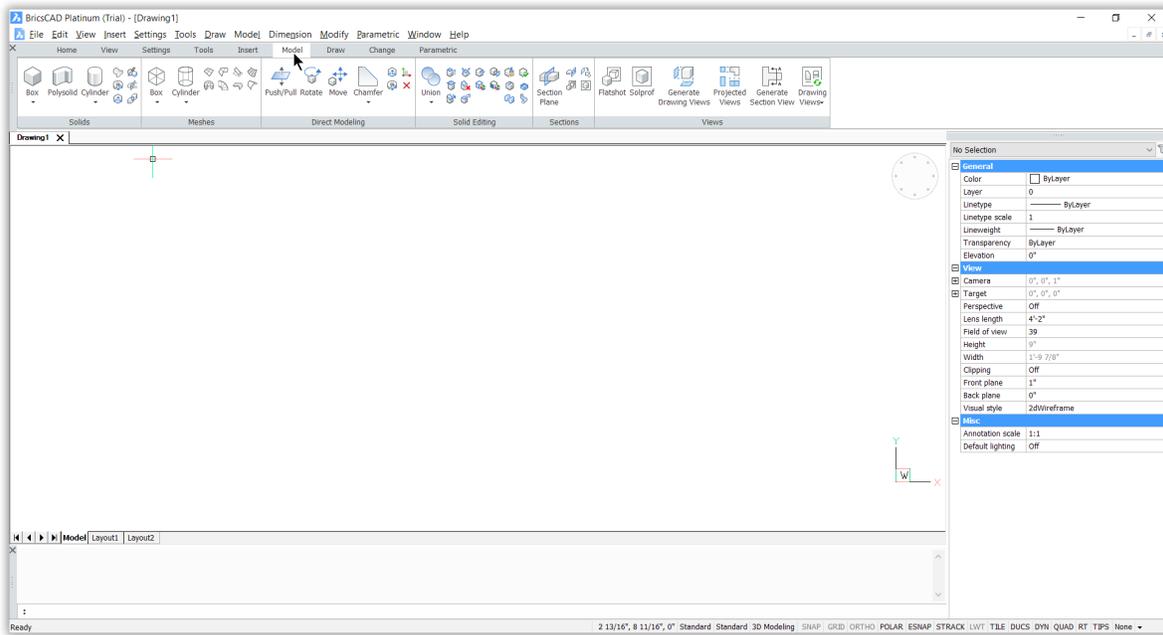
4. Click **3D Modeling**.



5. When the Create/Reinitialize Profile dialog box appears, click **OK**.



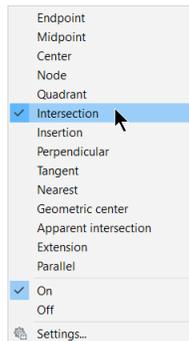
6. Notice that BricsCAD opens in the 3D Modeling workspace. On the ribbon, click the **Model** tab.



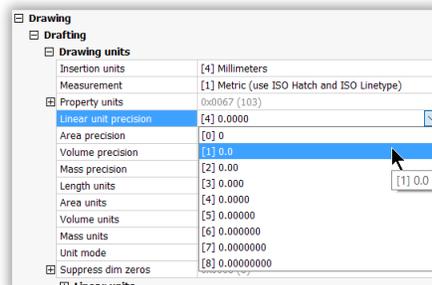
7. Prepare this drawing by changing the following settings in the status bar:

Setting	Toggle	Comments
SNAP	Off	You use entity snaps and dynamic dimensions in place of snap spacing
GRID	Off	The grid is not useful for this project
ESNAP	On	Turn on INT ersection snap; turn off all others

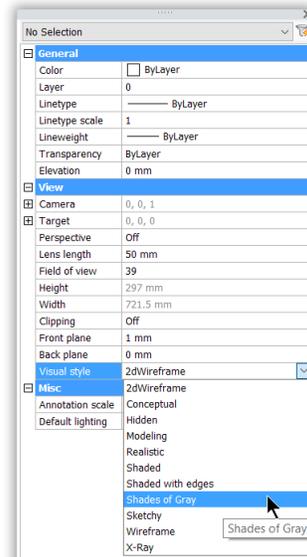
To turn on Intersection entity snap, right-click **ESNAP** on the status bar, as shown below. Ensure all other ensaps are turned off, as shown below.



8. Use the **Units** command to change the display precision of decimal places (linear units) to 1. You will be working with dimensions to the nearest 0.5mm.



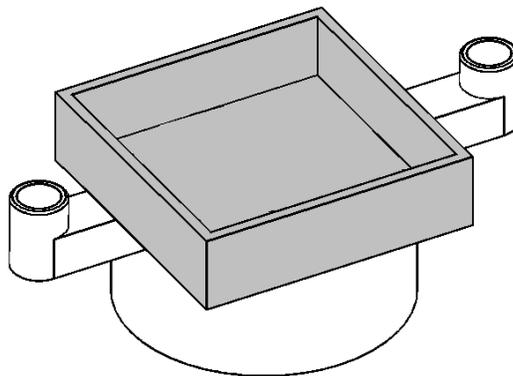
9. Close the Settings dialog box by clicking the **X**.
10. Change the visual style to “Shades of Gray,” which I find I like the best for 3D modeling. You can do this through the Properties panel.
 - a. If it is not open, enter the **Properties** command.



- b. In the View section, click the **Visual Style** droplist
 - c. Choose “Shades of Gray.” (By the way, there are 256 shades of gray, not just 50.)
11. Also, change the value of **DelObj** to **0**. Recall from a previous lesson that this system variable determines what happens to 2D entities after you convert them to 3D models. (When set to 1 or 2, the program erases them; I find it useful to keep them around.)

Modeling a Box

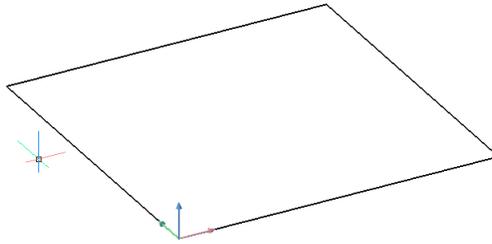
The top of the parts is a square open box. It is shown in gray in the figure below.



To make it, you’ll work through three commands, in this order:

- **Rectangle** command defines the size and base of the square
- **Extrude** command changes the 2D square into a solid 3D box
- **SolidEdit** command hollows the 3D box by shelling it

1. First, use the **Rectangle** command's **Dimension** option to draw a 2D square sized 13x13mm:



```
: rectangle
Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/Area/Dimensions/<Select first
corner of rectangle>: d

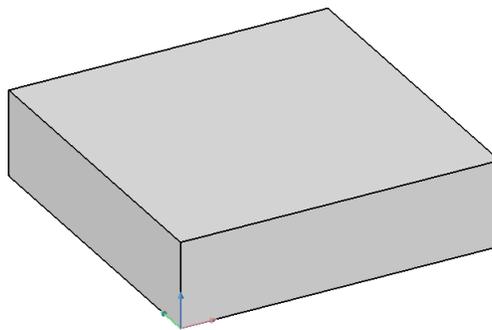
Length to use for rectangles <13.0>: 13
Width to use for rectangles <13.0>: 13

Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/Area/Dimensions/<Select first
corner of rectangle>: 0,0
Other corner of rectangle: (Pick a point.)
```

2. If necessary, use the **Zoom Extents** to see it better.
3. Now use the **Extrude** command to convert the square into a short 3D box. Specify a height of 3.5mm, as follows:

```
: extrude
Select entities: (Pick the square.)
Select entities: (Press ENTER to end entity selection.)

Specify height of extrusion or [Direction/Path/Taper angle] <1.0>: 3.5
```



QUICK SUMMARY OF EXTRUDE COMMAND

BricsCAD Pro and Platinum include the **Extrude** command:

```
: extrude
Select entities: (Choose one or more 3D solids or 2D regions.)
Select entities: (Press ENTER to end entity selection.)
Specify height of extrusion or [Direction/Path/Taper angle] <1.0>: (Enter an option.)
```

Height — specifies the height of the extruded entities; positive values extrude in the positive z direction, negative values go “downwards”

Direction — controls the direction of the extrusion, up or down along the z axis

Path — allows curved extrusions by specifying an entity that determines the path

Taper angle — gives sloping sides to the extrusion; positive angle slope inwards, negative angles outwards

4. Follow these steps to make the box hollow:
 - a. From the ribbon's **Model** tab, look in the **Solid Editing** panel, and then click **Shell**. (It's a bit hard to find it, but Shell is the last icon in the middle row.) "Shell" is an option of the SolidEdit command that turns solid entities into "walls" — it uniformly removes the insides. Because the SolidEdit command has many options, it is faster to get to the Shell option via the ribbon.:


```

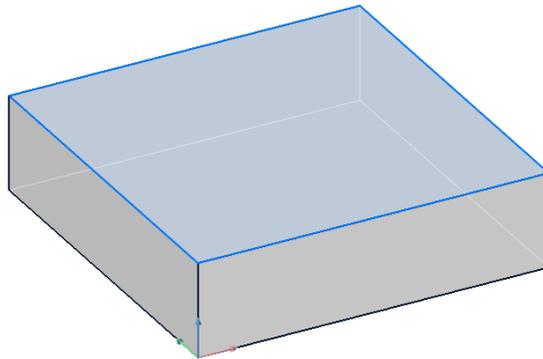
          : _solidedit
          Enter a solids editing option Face/Edge/Body/Undo/eXit: _body
          Enter a body editing option Imprint/seParate/Shell/cLean/Check/Undo/eXit: _shell
          
```
 - b. As prompted, select the extruded box:


```

          Select 3d solid: (Pick the box)
          
```
 - c. Pick the top face to remove it entirely. It is shown in blue in the figure below. BricsCAD does not give any indication when you choose the face, and so you work somewhat blindly here.


```

          Select face to remove or Add/Undo/ALL: (Pick the face on top of the box)
          Select face to remove or Add/Undo/ALL: (Press Enter to exit entity selection)
          
```



- d. Specify the thickness of the walls, **0.5mm**, for the remaining sides:

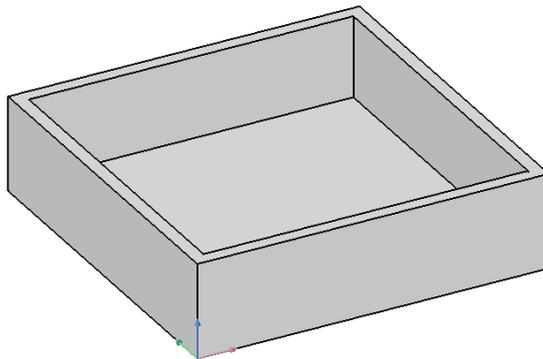

```

      Enter the shell offset distance: .5
      
```
- e. And finally press **Enter** twice to exit the command:


```

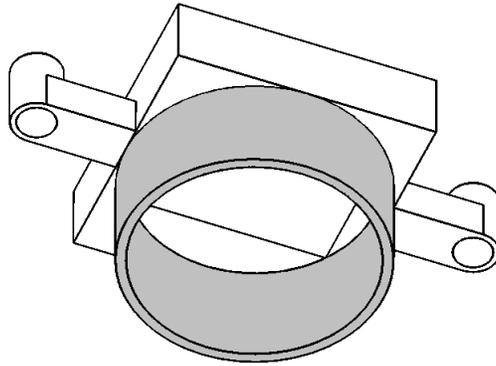
      Enter a body editing option Imprint/seParate/Shell/cLean/Check/Undo/eXit: (Press Enter)
      Enter a solids editing option Face/Edge/Body/Undo/eXit: (Press Enter)
      
```

Notice that the result: a hollow box with no top.



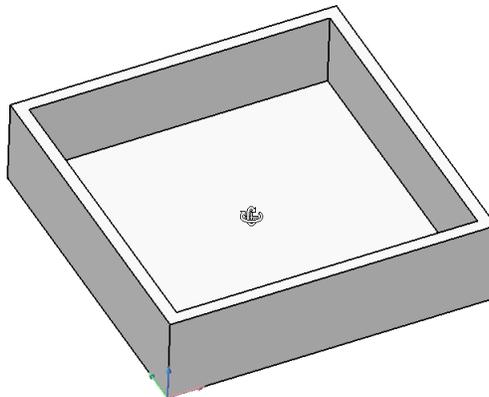
3D View Rotation

The next step is to add the round part underneath the box, shown in gray below.

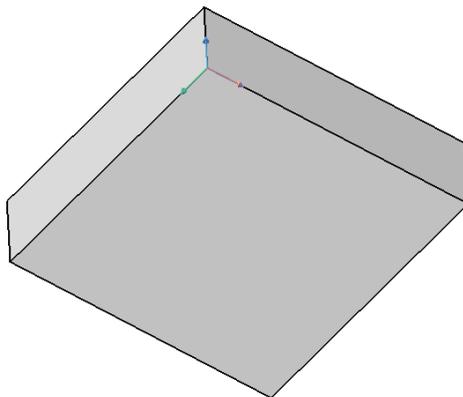


But to do this, you need to spin the model so that you can see the bottom of the box. To do so, there are several methods you could employ, but the easiest is the most direct way: real-time rotation.

1. Position the cursor in the center of the box. The location of the cursor determines the center point about which the model will rotate. This means it is important for you to locate the cursor at roughly the right point.
2. Hold down the **Shift** key. (If you don't hold down Shift, then the model will pan instead of rotate.)
3. Hold down the middle mouse button, and then move the mouse. Notice the real-time rotation  cursor.



4. As you drag the cursor, notice that the entire model rotates. Once you see the bottom of the box, let go of the mouse button and **Shift** key.



5. Press **ESC** to exit the command.
6. If necessary, use the **Zoom Extents** command to see the entire model.

EXTRUSION BY DIRECT MODELING

The bottom of the part has a hollow cylinder. In this part of the tutorial, you draw two circles and then extrude them using *direct modeling* — this means you employ no 3D commands to do the work.

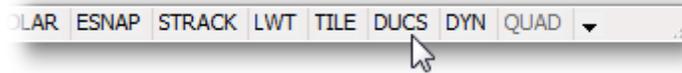
But first, this word of explanation of how to draw in 3D space...

Applying Dynamic UCS

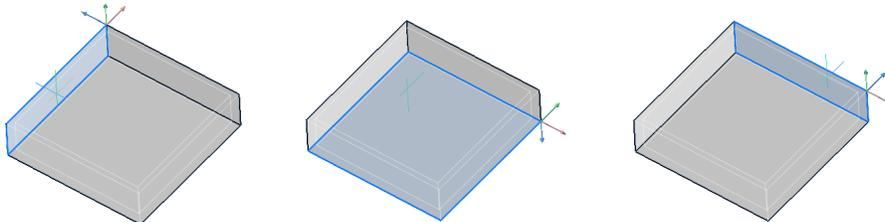
A crucial aspect to direct modeling is working with the correct plane. No matter which 3D CAD package you use, they all fall back to working on the x,y-plane. In 3D, a box has six planes, one for each side. CAD, however, can only draw on one side at a time: the side that is in the current x,y-plane.

Fortunately, CAD systems make it easy to rotate the x,y-plane so that it matches the area on which you are working. In BricsCAD, this is called *dynamic UCS*. Here is how it works.

1. On the status bar, ensure **DUCS** is turned on. The letters should look black, not gray.



2. To draw the circle on the bottom of the box, start the **Circle** command with the **2P** option, as follows:
: circle
2Point/3Point/TanTanRad/Arc/Multiple/<Center of circle>: 2p
3. Now you get to see the effect of dynamic UCS: move the cursor around the visible faces of the box. Notice that two things are happening: the face turns blue, and the UCS icon jumps to a corner of the face. This is dynamic UCS at work: BricsCAD is automatically relocating the x,y drawing plane to the face that you pick.



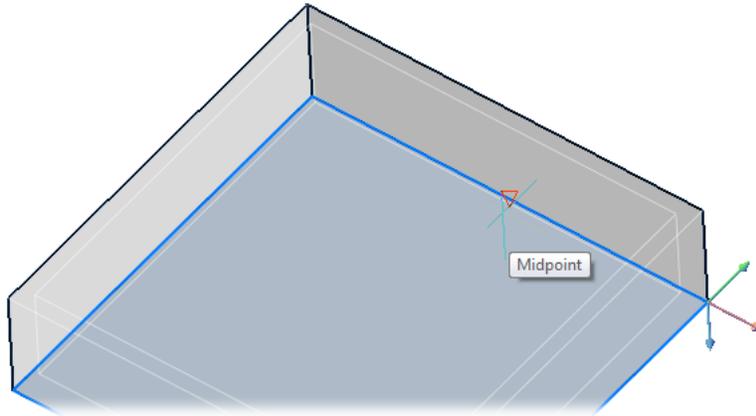
Left to right: As the cursor passes over a face (in blue), BricsCAD dynamically relocates the UCS (tri-color icon) to the corner of the face, making the face a temporary x,y drawing plane; the UCS icon is positioned at the origin

In summary: As the cursor passes over a face during DUCS, the face turns blue to tell you that it is the current, temporary x,y drawing plane. The UCS icon is positioned at the current, temporary origin (0,0,0). DUCS is available only during drawing and editing commands, because most of them operate only on an x,y plane.

TIPS DUCS works only during drawing and editing commands.

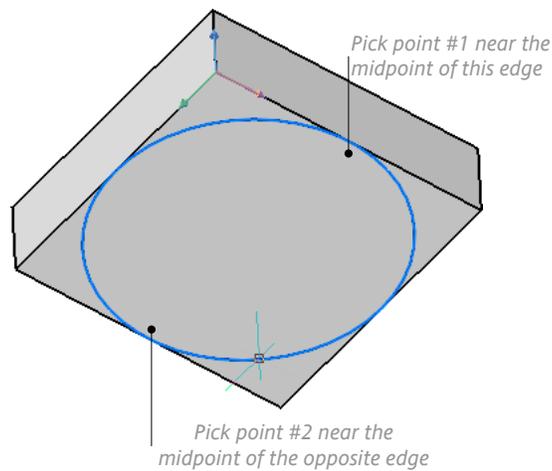
If you want to see a more dramatic effect of dynamic UCS at work, turn on the grid display. (Click **GRID** on the status bar.) As you move the cursor from face to face, the grid reorients itself to match the face.

4. To locate the circle, pick two points on the edges of the box, right at the middle of two facing edges. Use the **MIDpoint** ensap to assist you.
 - a. Position the cursor over the bottom face of the box. Notice that it turns blue.
 - b. Move the cursor close to the edge illustrated below, but do not go beyond the edge! (Were you to go beyond the edge, then the DUCS would jump to the adjacent face, telling BricsCAD in effect that you want to draw the circle on a different face.)

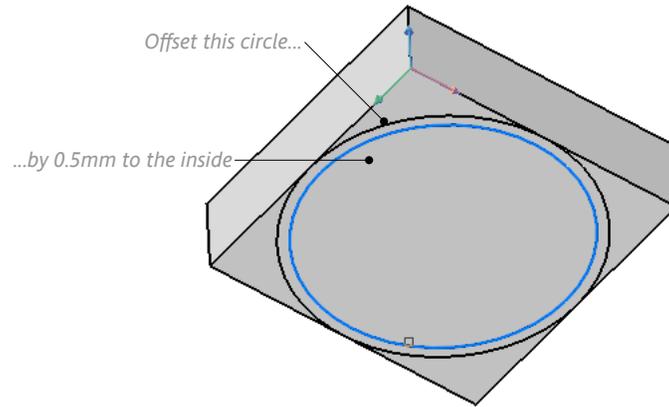


The crosshair cursor is hard to see in this figure, but it is near the edge while still on the bottom (blue) face

- c. Enter the MIDpoint entity snap mode:
First point on diameter: **mid**
- d. Pick a point near the edge:
Snap to midpoint of: *(Pick near point 1.)*
- e. Repeat the MID ensap mode, and pick a point near the opposite edge:
Second point on diameter: **mid**
Snap to midpoint of: *(Pick near point 2)*



5. A second circle is needed to define the thickness of the 0.5mm wall. Instead of drawing it, you will make a copy using the **Offset** command. The offset distance of **0.5mm**.



```

: offset
Offset: Through point/Erase/<Distance> <0.5>: .5

ENTER to stop/<Select entity>: (Select the black circle)
Both sides/<Side for parallel copy>: (Pick a point inside the circle)

ENTER to stop/<Select entity>: (Press Enter to end the command)

```

Extruding Directly with the Quad Cursor

With the pair of 2D circles in place, you extrude them to create a 5mm-tall hollow cylinder. This time, you use the Quad to extrude. The Quad is unique to BricsCAD, and gives you fast access to commands right at the cursor. In many cases, commands suitable to the highlighted element are presented to you. (See the boxed text for more on using the Quad.)

Cylinders are extruded from circles. Making a hollow cylinder takes these steps:

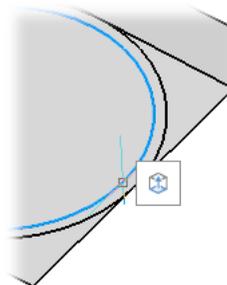
- > a. Extrude the inside circle into a cylinder that is more than 5mm in height
- > b. Extrude the outside circle by exactly 5mm
- > c. Subtract the taller cylinder from the shorter to make the hole.

You extrude the inside circle by a distance longer than 5mm, because its height as a cylinder does not matter; indeed, in a later step, you “erase” it to make the hole. Follow these steps:

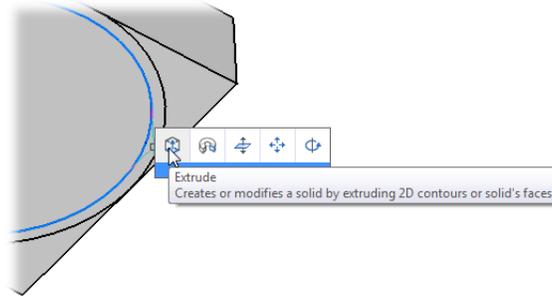
1. On the status bar, ensure that **QUAD** is turned on.



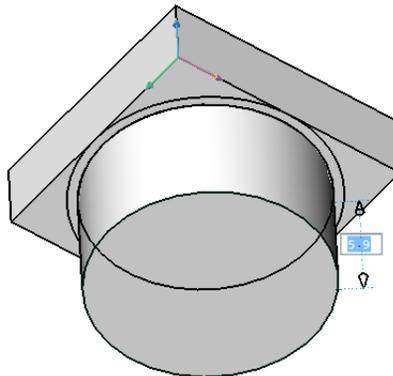
2. Move the cursor over the smaller circle. Notice that it turns blue, and that a second cursor appears. This is the Quad cursor, sporting a single icon (for now).



- The icon  on the cursor indicates the Extrude command. To confirm, pass the cursor over the icon. Notice that the Quad cursor expands to show more commands; after a moment the tooltip appears explaining the name and purpose of the button.

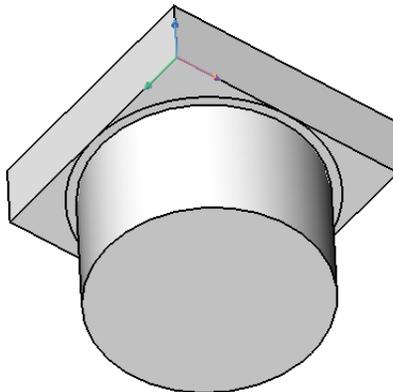


- Click the **Extrude**  button, and then move the cursor. As you do, notice that the circle extrudes into a cylinder (i.e., thickens in 3D). The direction you move the cursor determines the direction the circle extrudes, upwards or down.
- Notice the *dynamic dimension* next to the cylinder: a pair of arrowheads and text with blue background. It allows you to enter precise distances; in this case, however, you don't care about the *precise* height of this cylinder, because its only purpose is to be cut out of the larger one to form a hole. But it is useful to ensure the cylinder ends up longer than 5mm.

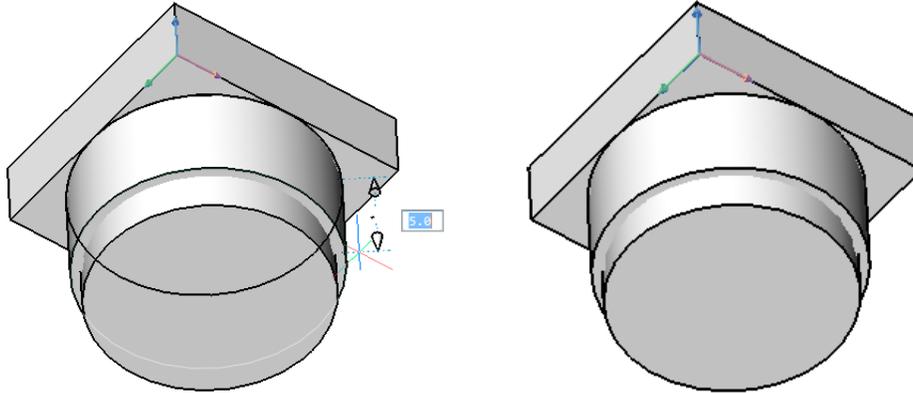


Watching the *dynamic dimension*, drag the extrusion so that it is longer than 5mm — like 7 or 8mm.

- Then click. The cylinder is formed — without entering commands.



- Repeat the direct modeling procedure for the larger circle. This time, however, enter 5 in the direct dimension.



Left: Extruding the outer circle into a 5mm cylinder; right: Two cylinders in place

- Save your work with **Ctrl+S**, naming it “Focus-Bracket.dwg.”

Subtracting Solids To Make Holes

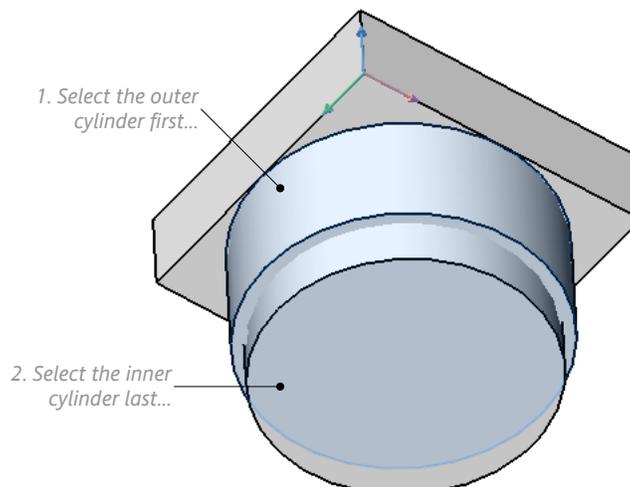
Command: Subtract
Ribbon: Solid | Edit | Subtract
Menu: Model | 3D Solids Editing | Subtract
Alias: SU

You learned about Boolean operations in a previous lesson, where it was applied to 2D regions. The same operations can be applied to 3D solids: union, subtract, and intersect. In this tutorial, you create a hole by subtracting one cylinder from another.

TIP The **Subtract** command is sensitive to the order in which entities are selected. Make sure you pick the correct ones in the correct order: first the one that stays, and then the one that will be removed (subtracted).

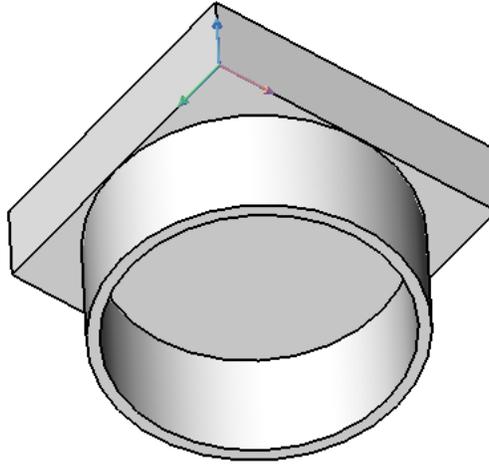
- With the two cylinders looking somewhat like a wedding cake, it is time to subtract the smaller one from the larger one. Start the **Subtract** command, and then choose the cylinders in correct order:

: subtract
Select ACIS entity to subtract from: (*Select the fatter, shorter cylinder*)
Select ACIS entity to subtract from: (*Press Enter to end entity selection*)



2. Now pick the cylinder to be removed (erased):
 - Select ACIS entities to subtract: (*Select the smaller, longer cylinder*)
 - Select ACIS entities to subtract: (*Press **Enter** to end the command*)

After you press **Enter**, the Subtract command ends and hollows out the cylinder successfully.



PushPull Modeling

A hole needs to be punched through the wall between the cylinder and the box. While developing this tutorial, I tried a number of approaches to make the opening, and found the only good way was to again draw a circle and use PushPull to turn it into a hole.

1. The easiest way to draw a circle (and other 2D entities) is in plan view. The quick way to switch between static viewpoints — such as the plan and isometric viewpoints — is to use the Look From widget. To use it, follow these steps:
 - a. Locate the Look From widget. Usually, it is in the upper right corner of the drawing area.

TIP If the Look From widget is turned off, you can turn it on with the **LookFrom** command:

```
: lookfrom  
LookFrom [ON/OFF/Settings] <ON>: on
```

- b. Move the cursor into the center of the widget. Notice that the icon changes to show the top view of a chair.



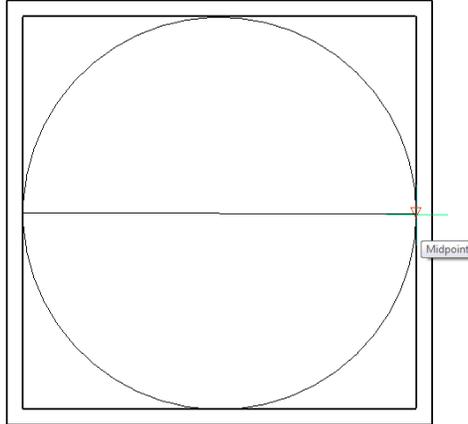
- c. Click in the center of the widget. Notice that the viewpoint changes.
 - d. You may need to do a **Zoom Extents** to see the entire model.

- To draw the circle, use the same technique as before: employ the **Circle** command with the **2P** option and **MIDpoint** esnaps:

```

: c
2Point/3Point/TanTanRad/Arc/Multiple/<Center of circle>: 2p
First point on diameter: mid
Snap to midpoint of: (Pick one edge)
Second point on diameter: mid
Snap to midpoint of: (Pick the opposite edge)

```



- Using the Quad cursor, punch out the circle using direct modeling. You may find it easier to control the extrusion by clicking the **Top Front Left** position on the Look From widget.



QUICK SUMMARY OF LOOKFROM COMMAND

```

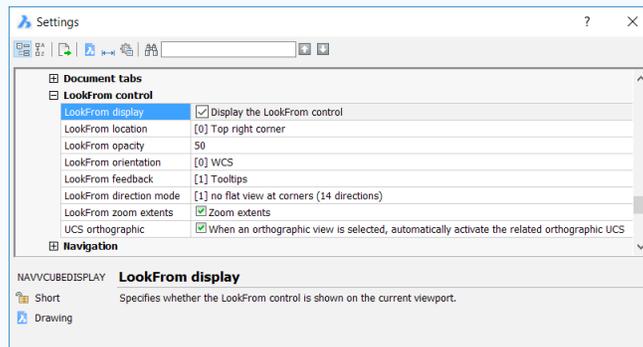
: lookfrom
LookFrom [ON/OFF/Settings] <ON>: on

```

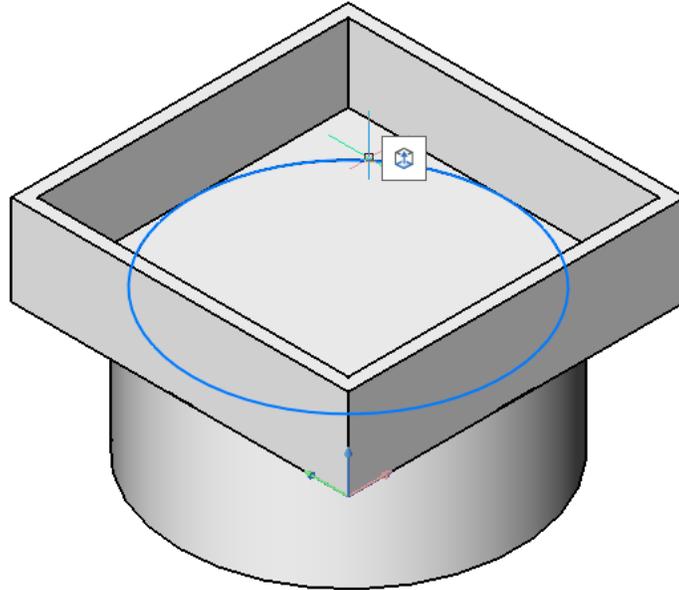
ON — turns on the Look From widget

OFF — turns off the Look From widget

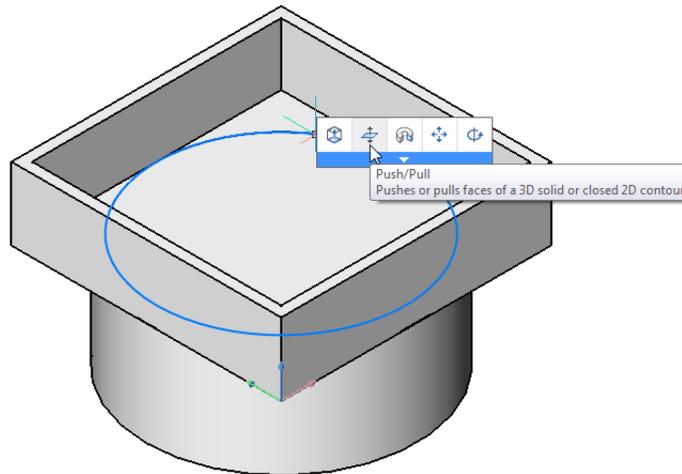
Settings — displays the Settings dialog box at the Look From Control section



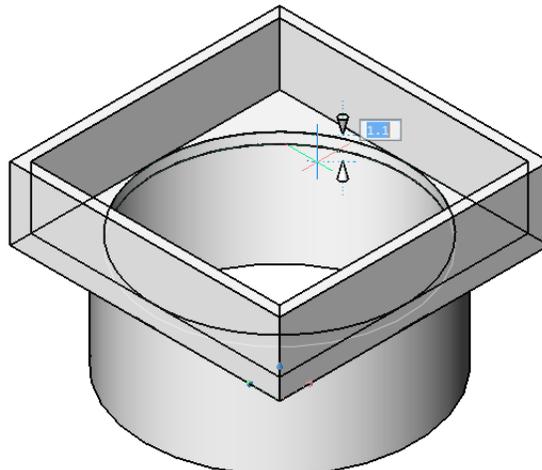
- a. Pass the cursor over the newly added circle. Notice that it turns blue.



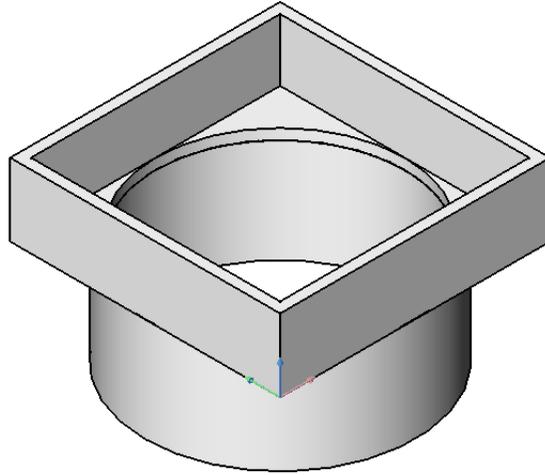
- b. Move the crosshair cursor into the Quad cursor. Notice that it expands to show more commands. Choose the **PushPull**  button.



- c. Drag the cursor down so that a hole appears. The distance you drag does not matter; all you need is for the hole to appear.



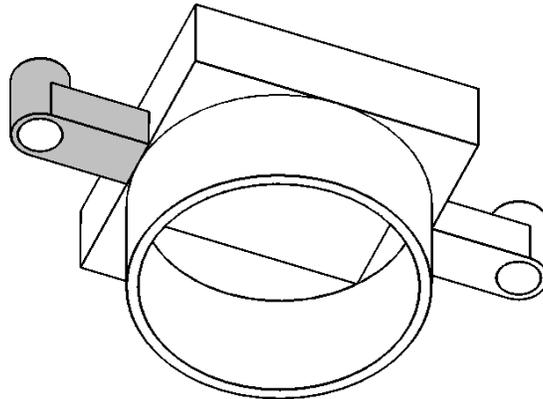
- d. Click to finish the hole. This was quicker and easier than using the Extrude command!



Aligning the UCS

Command	UCS Face
Alias	...

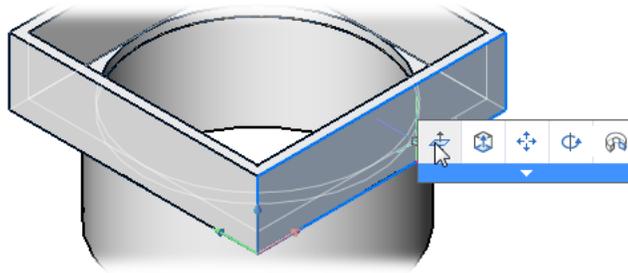
The next step is to draw the rectangle needed for the arms that hold the two screw holes, one of which is shown in gray in the figure below. This time you use PushPull to create an entity, instead of a hole.



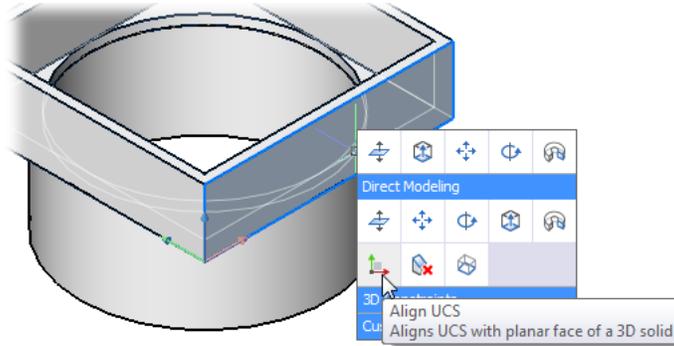
In this case, the 2D entity is a rectangle. Before drawing the rectangle, however, you align the UCS on the face by another method: align UCS. This permanently reorients the UCS plane to a face — permanent until you change it again. This is different from the temporary, dynamic UCS alignment you used in the earlier part of this tutorial.

1. Ensure **QUAD** is turned on in the status bar, and then move the crosshair cursor over a rectangular face. Notice that it is highlighted in blue, and that the Quad cursor appears.

2. Move the crosshair cursor into the Quad cursor. The crosshair turns into an arrow cursor.

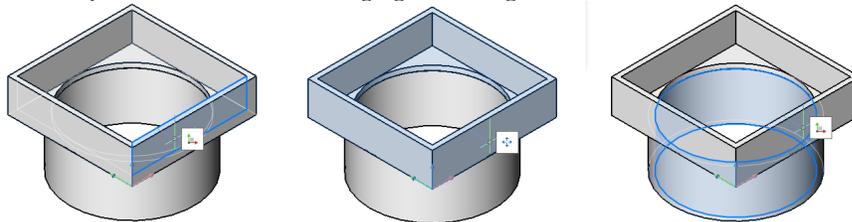


3. Move the arrow cursor into the blue Direct Modeling title bar. Notice that the Quad cursor expands to show more icons.



4. Choose the **Align UCS**  button, and then press **Enter**. The UCS plane is now fixed on the face.

TIP If BricsCAD does not highlight the feature in which you are interested, then press the **Tab** key. Each time you press Tab, a different feature laying under the cursor is highlighted. Press Tab enough times, and the selection cycles around to the first one highlighted. See figure below.



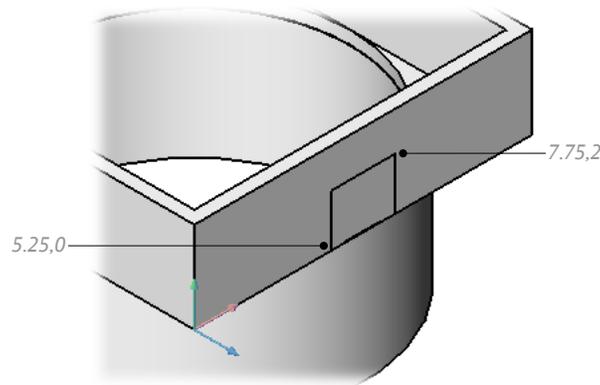
Pulling, Instead of Pushing

Command	dmPushPull
Ribbon	Solid Edit Push/Pull
Menu	Model Direct Modeling Push/Pull

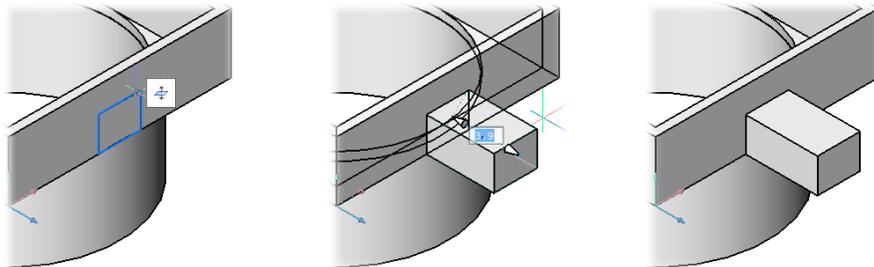
Draw a rectangle on the face, then pull it out with the direct PushPull operation. When it comes to the screw holes, I want you to make a mistake so that I can show you that Bricsys also performs direct editing — in this case, changing the diameter of a 3D hole without using commands.

1. The arm has a rectangular cross-section of **2.5mm** wide (length) by **2.0mm** high (width). Draw its profile with the **Rectang** command. I'll give you the dimensions:

```
: rectang  
Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/Area/Dimensions/  
<Select first corner of rectangle>: 5.25,0  
Other corner of rectangle: 7.75,2
```

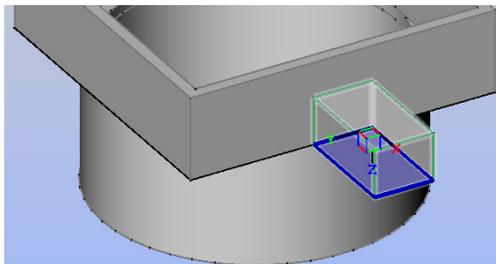


2. Use direct modeling to pull the profile to a length of **4.5mm**.



Left to right: Choose the rectangle (a.k.a "profile"); select PushPull command and specify extrusion distance of 4.5mm; done!

3. The screw hole hangs on the wing. It is made of a pair of cylinders, which (again) are made from a pair circles. First, though, align the UCS to the bottom face of the wing, as illustrated below:



Remember to press **Tab**, if you find you have difficulty selecting the bottom face.

4. Drawing 2D entities in 3D can be tricky when the viewpoint is the plan view. By default, Bricsys will snap to the nearest geometric feature; in 3D, this can too easily be the one you don't want. To solve the problem, turn on the **OsnapZ** system variable:

```
: osnapz  
New current value for OSNAPZ (Off or On) <Off>: on
```

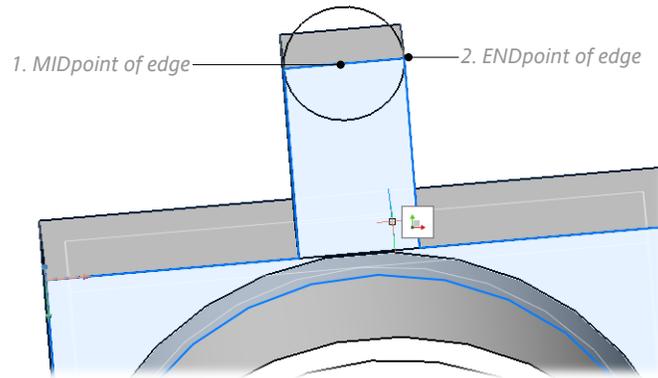
When on, all entity snaps set $z = 0$, so that the snapping takes place on the x,y -plane — and not at some other higher or lower location.

OsnapZ	Meaning
Off	Snaps to the nearest z coordinate
On	Snaps to the elevation setting (usually 0; $z = \text{elevation}$)

5. Using the Isometric Views toolbar, change the viewpoint to the top (or plan) view.

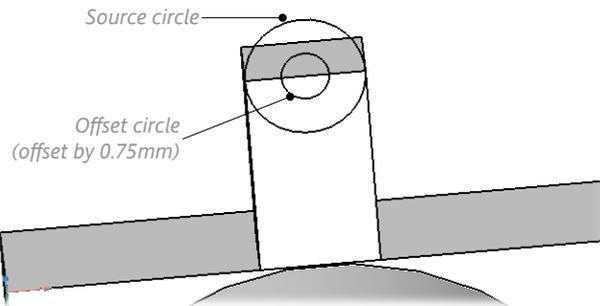
6. Use the Circle command to draw the round profile:

```
: c  
2Point/3Point/TanTanRad/Arc/Multiple/<Center of circle>: mid  
Snap to midpoint of: (Pick the midpoint of the edge.)  
  
Diameter/<Radius> <1.3>: end  
Snap to endpoint of: (Pick the end of the edge)
```



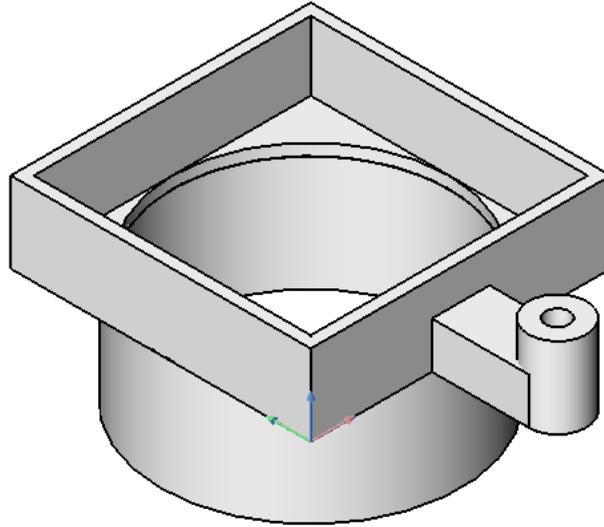
When you rotate the view back to isometric, you see the circle placed on the bottom face — thanks to OsnapZ.

7. Use the **Offset** command to place a second circle, offset by **0.75mm**. (This value is deliberately incorrect, and you correct it later with direct editing.)

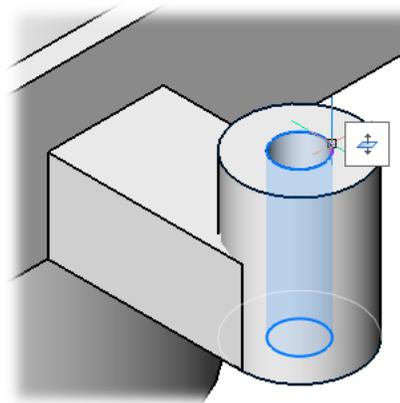


8. As in the earlier tutorials, use direct modeling to extrude the two circles into cylinders:
a. Use Extrude to turn the outer circle in a cylinder with a height of precisely 3.5mm.

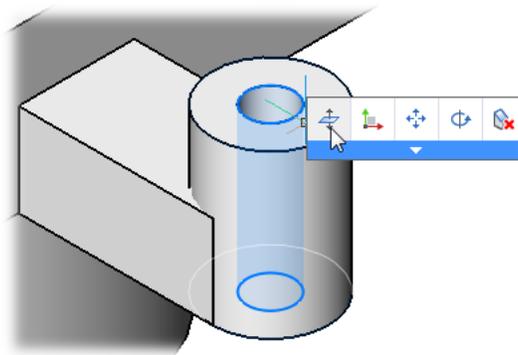
- b. Use PushPull to turn the inner one to any height taller than 3.5mm; notice that it creates the hole automatically — no need to use the Subtract command!



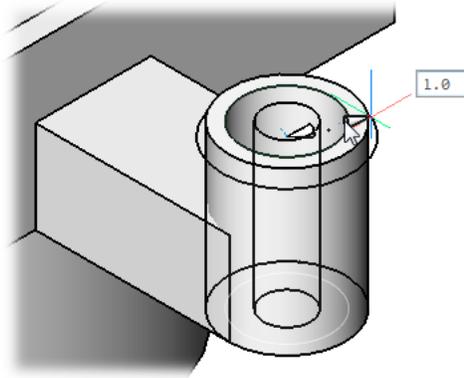
9. The diameter of the hole is too large. It should be 1mm, but is currently 0.5mm. BricsCAD can edit solid models using direct editing. Here is how to correct the diameter of the hole:
- a. Move the cursor over the edge of the hole. If the hole does not highlight in blue, then press the **Tab** key until it does.



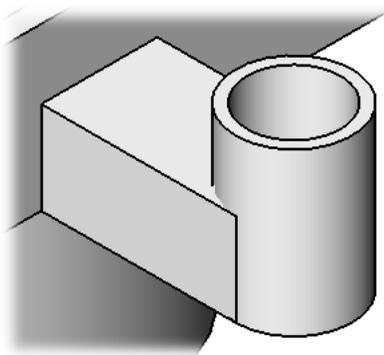
- b. In the Quad cursor, choose the **PushPull**  button.



- c. The hole must be changed to a diameter of 1.0mm. Drag the hole dynamically, or enter 1.0 in the dynamic dimension.



The hole is the correct size. Press Ctrl+S to save your work.

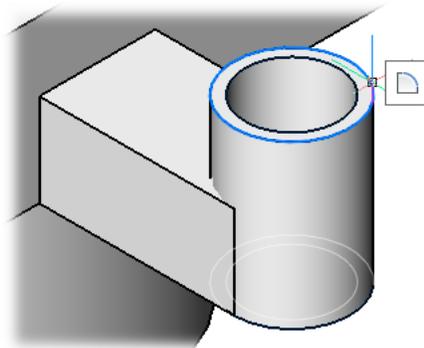


TIP To change the UCS back to “normal,” enter the **UCS** command, and then choose the **World** option:
: **ucs**
Specify origin of UCS or Face/NAmed/Entity/Previous/View/X/Y/Z/ZAis/Move<World>: **w**

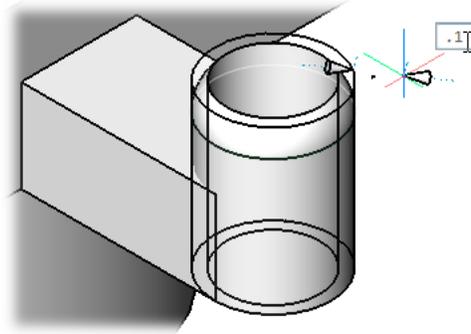
Filleting 3D Solids

The outer edge of the screw hole has a small fillet around the edge. With direct editing, you can apply fillets interactively. Here’s how:

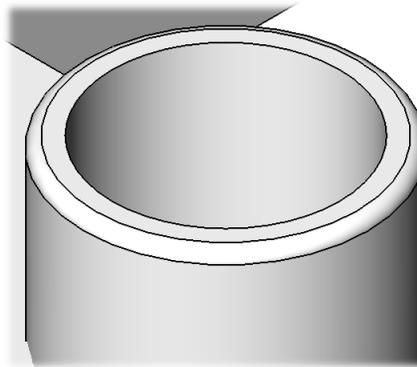
1. Move the cursor over the edge of the cylinder. Remember: if you cannot select the outer edge, then press **Tab** until BricsCAD highlights it. See the figure below.



2. Choose the  **Fillet** button on the Quad cursor, and then enter **0.1** as the fillet radius.



...and then press **Enter**.



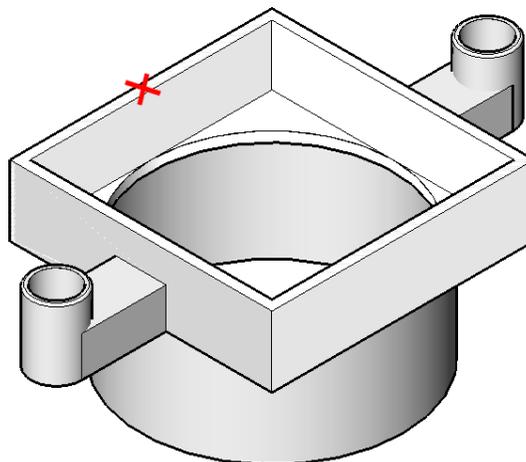
3. Mirror the two new parts to place them on the other side, like this:

```
: mirror
Select entities to mirror: (Select the wing part and screw hole)
Select entities to mirror: (Press Enter to continue)

Start of mirror line: mid
Snap to midpoint of: (Pick middle of rectangular part, as illustrated below)
End of mirror line: mid

End of mirror line: mid
Snap to midpoint of: (Pick a point on the other side)

Delete the original entities? <N> n
```



Joining Parts with Union

The 3D model consists of six solids. You can leave them as individuals, or join them into a single body with the Union command.

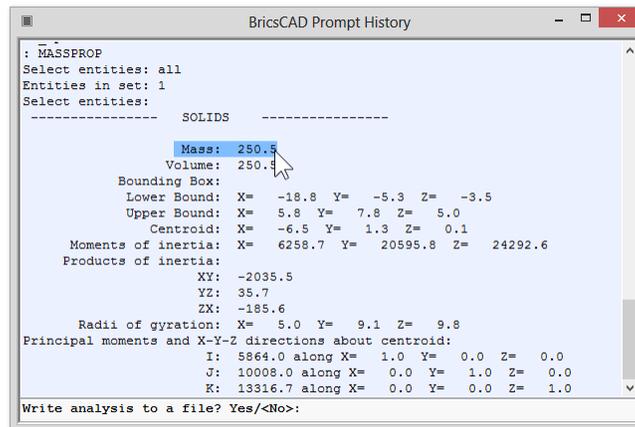
1. Use the **Union** command to merge all solids into one:
: union
Select ACIS entities to union: all
Select ACIS entities to union: (Press Enter to end the command)

TIP You can continue to directly edit the model, even after all its parts are joined by the **Union** command.

2. Now that the model is one unified part, you can apply commands like **MassProp** and **Properties** to find out the properties of this focusing bracket.

```
: massprop  
Select entities: all  
Select entities: (Press Enter to continue)  
Write analysis to a file? Yes/<No>: y
```

The report you see in the text window can be saved to a .mpr (mass properties report) file.



```
BricsCAD Prompt History  
: MASSPROP  
Select entities: all  
Entities in set: 1  
Select entities:  
----- SOLIDS -----  
Mass: 250.5  
Volume: 250.4  
Bounding Box:  
Lower Bound: X= -18.8 Y= -5.3 Z= -3.5  
Upper Bound: X= 5.8 Y= 7.8 Z= 5.0  
Centroid: X= -6.5 Y= 1.3 Z= 0.1  
Moments of inertia: X= 6258.7 Y= 20595.8 Z= 24292.6  
Products of inertia:  
XY: -2035.5  
YZ: 35.7  
ZX: -185.6  
Radii of gyration: X= 5.0 Y= 9.1 Z= 9.8  
Principal moments and X-Y-Z directions about centroid:  
I: 5864.0 along X= 1.0 Y= 0.0 Z= 0.0  
J: 10008.0 along X= 0.0 Y= 1.0 Z= 0.0  
K: 13316.7 along X= 0.0 Y= 0.0 Z= 1.0  
Write analysis to a file? Yes/<No>:
```

3. Save the drawing.

Generating 2D Drawings

Command	ViewBase
Ribbon	Model Views Generate Drawing Views
Menu	View Generated Views Generate Drawing Views

With the 3D model completed, it is time to generate engineering drawings from it. These are 2D plans that are used to check dimensions and give instructions to the manufacturer. BricsCAD generates these drawings semi-automatically. Let's see how it works.

1. Switch your focus to the **Views** panel of the ribbon's Model tab. It contains the commands you need for this tutorial.

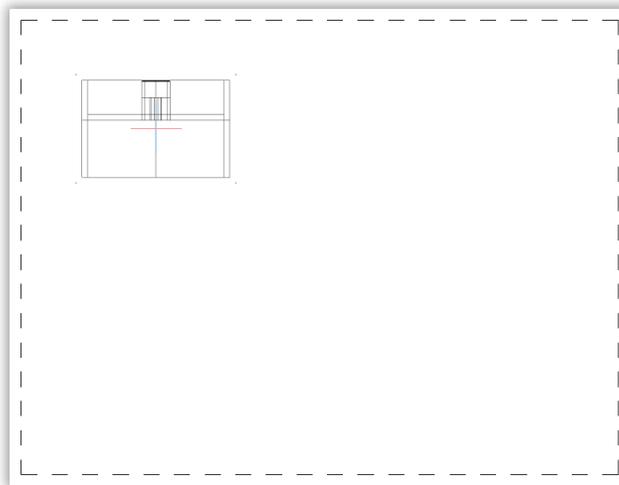


Click the  **Generate Drawing Views** button. (It runs the ViewBase command). This command automatically generates 2D views of the 3D model — front, side, top, isometric — in a new layout; all you need to do is pick the points where you want the views placed.

2. Press **Enter** to make drawings from the entire model:
: _viewbase
Preset: "None", View scale: "Adapt to paper size"
Select objects or [Entire model/preseTs] <Entire model>: (Press **Enter**)
3. At the next prompt, press **Enter** to use Layout1 for the drawings:
Enter new or existing layout name to make current <Layout1>: (Press **Enter**)
4. Notice that BricsCAD switches to Layout1 automatically, and then previews the 2D drawing being generated from the 3D model.
(If it does not switch automatically, then click the **Layout1** tab at the bottom of the drawing area.)



As you move the cursor, the 2D preview moves and changes its projection to match the location.



In this step of the the tutorial, position the cursor in the upper left quadrant, and then click.
Select position for base view [Scale/Tangent edges/Orientation/Projection type/Isometric style/sElect] <Cancel>: (Move the cursor to upper left, and then click)

QUICK SUMMARY OF VIEWBASE OPTIONS

: **viewbase**

Select objects or [Entire model/presets] <Entire model>:

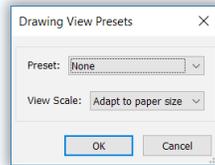
Enter new or existing layout name to make current <LayoutName>:

Select position for base view [Scale/Tangent edges/Orientation/Projection type/Isometric style]<Cancel>:

Select position for projected view <Done>:

PRESETS OPTION

Displays the DrawingView Presets dialog box:



Preset droplist offers these options:

- > **None**
- > **Architectural Views** — generates Front, Right, Back, Left (elevations) and Top views; three section views (two horizontal section plans); and one vertical section.
- > **Mechanical Views** — generates standard Front, Top and Left (first-angle projection) or Right (third-angle projection) views. The projection type: first or third angle. The projection type held by the ProjectionType variable.

View Scale droplist offers automatic scaling or one of the scales presided over by the ScaleListEdit command.

SCALE OPTION

Sets the scale of all views to fit the following arrangements:

- > **fit 4 views** — adjusts scale to fit four orthographic views: Front, Top, Left, and Right. The Front view (base view) is defined by the Orientation option.
- > **9 views** — adjusts scale to fit five orthographic views and four isometric views.
- > **5 views** — adjusts scale to fit five orthographic views: Front (*), Top, Left, Right, and Back.
- > **10 views** — adjusts scale to fit six orthographic views and four isometric views.
- > **Standard scales** — displays the scales lorded over by the ScaleListEdit command.
- > **Custom** — prompts for any scale factor.

TANGENT EDGES OPTION

Determines whether tangent edges between tangent faces are displayed or not; tangent edges are always displayed in isometric views:

ORIENTATION OPTION

After you specifies the main view, BricsCAD rotates the 3D model so that the main view is projected on the vertical projection plane. Choose from Front, Back, left, right, top, or bottom views.

Continued...

Continued...

PROJECTION TYPE OPTION

Determines the projection type: first angle (or European projection) or third angle (American projection).

ISOMETRIC STYLE OPTION

Determines the style for isometric views: rendered using Conceptual visual style, wireframe, or any other visual style.

The quadrant of the layout in which you click determines the viewpoint generated by BricsCAD:

Quadrant	2D View Generated
Upper left	Top view
Lower left	Front view
Upper right	Side view
Lower right	Isometric view

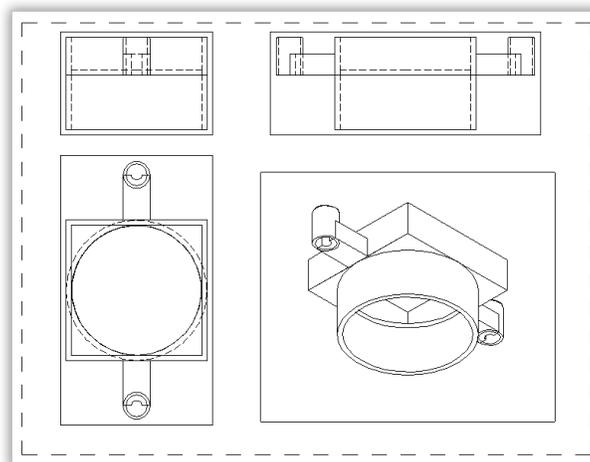
- The command repeats its prompt. Repeat the process: move the cursor to a quadrant of the layout, and then click to position the view.

Select position for projected view [Done] <Done> *(Move the cursor to Lower Left quadrant, and then click to place the front view)*

Select position for projected view [Done] <Done> *(Move the cursor to upper right quadrant, and then click to place the side view)*

Select position for projected view [Done] <Done> *(Move cursor to Lower right quadrant, and then click to place the isometric view)*

The result should look like the figure below. BricsCAD automatically adjust the scale factor so that the resulting viewports nicely fit the page.



- When you are finished placing view, press **Enter** to end the command. Don't press Esc, because all your carefully placed views are removed!

Select position for projected view [Done] <Done>: *(Press Enter)*

SLICING SECTIONS AND ADDING DETAILED VIEWS

Command ViewSection
Ribbon Model | Views | Generate Section View
Menu View | Generated Views | Generate Section View

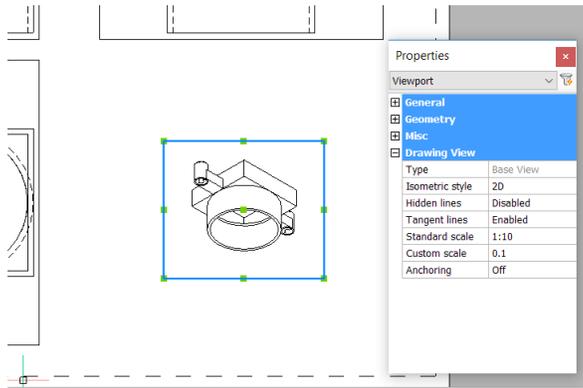
Command ViewDetail
Ribbon Model | Views | Detail View
Menu View | Generated Views | Detail View

With the 2D drawings in place, it is common to add sections that show the insides of models, which are used to indicate its materials with hatch patterns. When necessary, detail views are also added to show complex areas greatly enlarged.

Resizing Views

Before you can add a cross-section view, you have to adjust the existing views, because the page is full — no room for another view. One solution is to make the existing views smaller. Here is how to do this:

1. Select a viewport border.
2. In the Properties panel, click **Custom Scale**.



3. Enter a new value, such as .1.
4. Notice that the viewport is smaller. Move the viewport by dragging it (with its center grip) or through the Move command.

Another options is to erase unwanted views. To erase a view, select the viewport border, and then press the **Del** key on the keyboard.

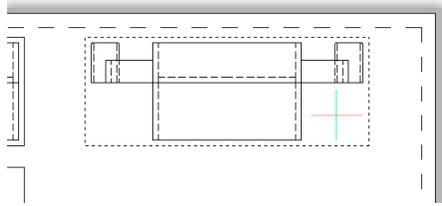
Making Section Views

Okay, with space freed up, you can go ahead to make the section view:

1. From the ribbon's **Model > Views** panel, click the  **Generate Section View** button (or else enter the **ViewSection** command).

2. BricsCAD prompts you to choose a drawing view. The section view will be generated from the view you pick, so choose carefully! For this tutorial, move the cursor to the upper left quadrant, and then click inside the view (side view).

: **_viewsection**
Select drawing view: *(Pick inside a viewport)*

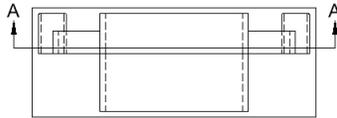


Notice that BricsCAD highlights the selected viewport with dashed lines.

TIP When selecting the view, don't select the viewport's rectangle, as BricsCAD will just repeat the prompt. Click *inside* the viewport, as shown by the crosshair cursor above.

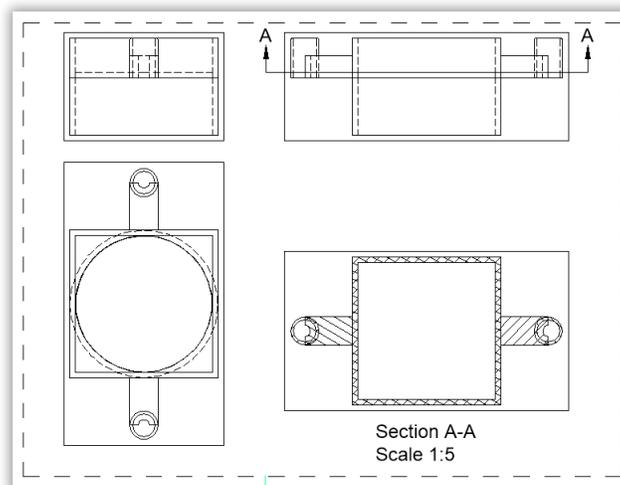
3. With the drawing view selected, it's time to show BricsCAD through which part of the drawing that you want the section to cut. At the next prompt, pick two points outside the viewport, shown by the arrowheads in the figure below:

Specify start point of section line or [Type] <Type>: *(Pick a point outside the viewport)*
Specify end point of section line: *(Pick a second point on the other side of the viewport)*



TIP I recommend holding down the **Shift** key to enforce ortho mode. This results in a section line that's precisely horizontal.

4. The final step is to position the section view:
Select position for section view: *(Pick a point in the drawing)*



QUICK SUMMARY OF VIEWSECTION OPTIONS

: **viewsection**

Select drawing view:

Specify start point of section line or [Type] <Type>:

Specify end point of section line:

Select position for section view:

Select option [Scale/Hidden lines/Tangent lines/anChor/Isometric style/Annotation/Depth/Projection] <Cancel>:

TYPE OPTION

Prompts for one of these options:

- > **Full** — draws a section of the entire viewport
- > **Half** — draws a section of half the viewport
- > **Offset** — draws a section specified by points that you pick
- > **Aligned** — draws a section perpendicular to the section line

SCALE OPTION

Sets the scale of the section view; choose from the standard scale factors, specify a custom scale, or inherit the scale of the parent view:

HIDDEN LINES OPTION

Toggles hidden line display of the section view on and off, or inherited from the parent view.

TANGENT LINES OPTION

Toggles display of tangent lines, on or off.

ANCHOR OPTION

Determines what happens to the section view when it is updated after the 3D model changes:

- > **Yes** — anchors the center of the viewport so that the viewport grows and shrinks around that point.
- > **No** — fixes the position of the geometry; the AutoVpFitting variable specifies whether the viewport size is adjusted to fit the extents of the 3D model automatically (default = on).

ISOMETRIC STYLE OPTION

Determines the style for isometric views: rendered using Conceptual visual style, wireframe, or any other visual style.

ANNOTATION OPTION

Prompts you to override that automatic annotations of section view: .

- > **Identifier** — specifies the view identifier, such as A.
- > **Label** — toggles the display of the view label.

DEPTH OPTION

Specifies clipping of the section view, full or limited. You can define the depth with your mouse.

PROJECTION OPTION

Toggles between normal and orthographic projection of the view.

BricsCAD automatically labels the section using the standard method of “A-A.” Engineers use the A-A to see what is referenced elsewhere in the drawing. It indicates the scale factor of the section, 1:5 — also automatically determined. Hatching is applied automatically to the cross section.

Hatching reports two types of information:

- ▶ Presence of hatching shows areas that are solid. Areas without hatching are empty air.
- ▶ Style of hatching indicates the type of material. Here you see “ANSI31” hatching, because it is the default pattern and it represents iron.

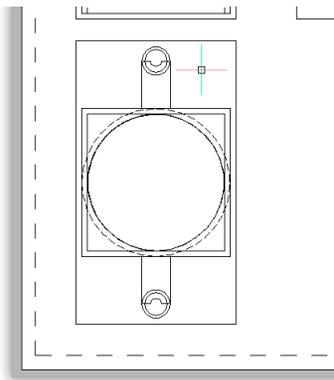
Here is the meaning of the ANSI hatch patterns included with BricsCAD:

Pattern Name	Example	Pattern Meaning
ANSI31		Iron, general purpose hatch
ANSI32		Steel
ANSI33		Bronze, brass, copper, composites
ANSI34		Rubber, plastic, electrical insulation
ANSI35		Defined by the legend
ANSI36		Defined by the legend
ANSI37		White metal, zinc, lead, babbit, and alloys of them
ANSI38		Magnesium, aluminum, and alloys of them

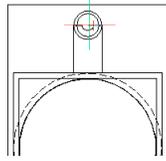
Creating Detail Views

A detail view is an enlargement of one part of a drawing. It makes it easier to see what is going on with complex parts. Details are placed a larger scale factor by the **ViewDetail** command. Here you make a detail of one of the small round parts:

1. Enter the enter the **ViewDetail** command:
: **viewdetail**
2. BricsCAD prompts you to choose a drawing view. This is the drawing view from which the detail will be taken. For this tutorial, click inside the lower left quadrant’s viewport.
Select drawing view: (*Pick inside a viewport*)

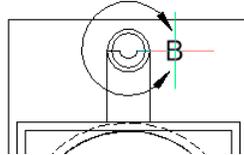


- Pick the center of the detail, as BricsCAD will use a circle as the view's boundary:
Specify detail center on source view: *(Pick the center of what will be detailed)*

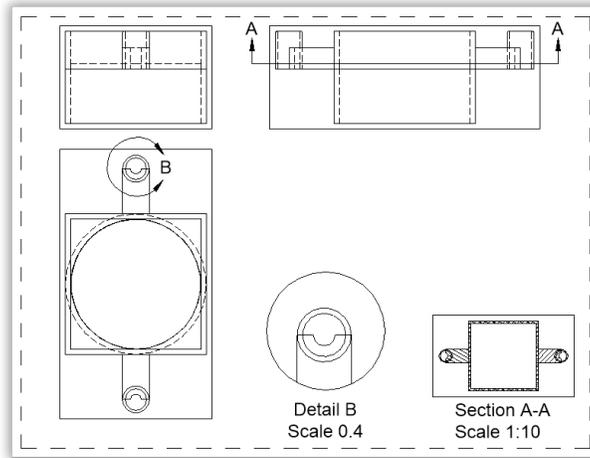


- Now size the circle to indicate the extent of the detail. You can see the circle in the figure above, it has the "B" reference attached.

Select radius of detail view: *(Drag the circle to size the detail area)*



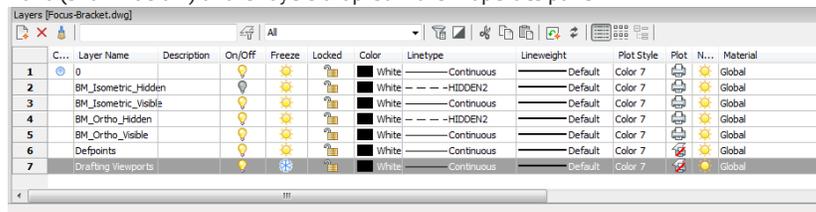
- Position the detail view somewhere in the drawing:
Select position for detail view [Scale/Exit] <Exit>: *(Pick a point)*



BricsCAD automatically labels the detail using the standard method of "B," which is used to refer to the source of the detail drawing. It also indicates the scale factor of the section, 4:1. You can use the Scale option to make the detail larger or smaller.

TIPS You can make sections of details, and details of sections, and details of details...

To hide the rectangles that make up the viewports, freeze the "Drafting Viewports" layer. You can use the Layer command (shown below) or the Layers droplist in the Properties panel.



QUICK SUMMARY OF VIEWDETAIL OPTIONS

: **viewdetail**

Select drawing view:

Specify detail center on source view:

Select radius of detail view:

Select position for detail view [Scale] <Cancel>:

Select option [Scale/Hidden lines/Tangent lines/Anchor/Annotation/Boundary/model Edge] <Cancel>:

SCALE OPTION

Sets the scale of the detail view; choose from the standard scale factors, specify a custom scale, or inherit the scale of the parent view:

HIDDEN LINES OPTION

Toggles hidden line display of the detail view on and off, or inherited from the parent view.

TANGENT LINES OPTION

Toggles display of tangent lines, on or off.

ANCHOR OPTION

Determines what happens to the detail view when it is updated after the 3D model changes:

- > **Yes** — anchors the center of the viewport so that the viewport grows and shrinks around that point.
- > **No** — fixes the position of the geometry; the `AutoVpFitting` variable specifies whether the viewport size is adjusted to fit the extents of the 3D model automatically (default = on).

ANNOTATION OPTION

Prompts you to override automatic annotations of detail views:

- > **Identifier** — specifies the view identifier, such as A.
- > **Label** — toggles the display of the view label.

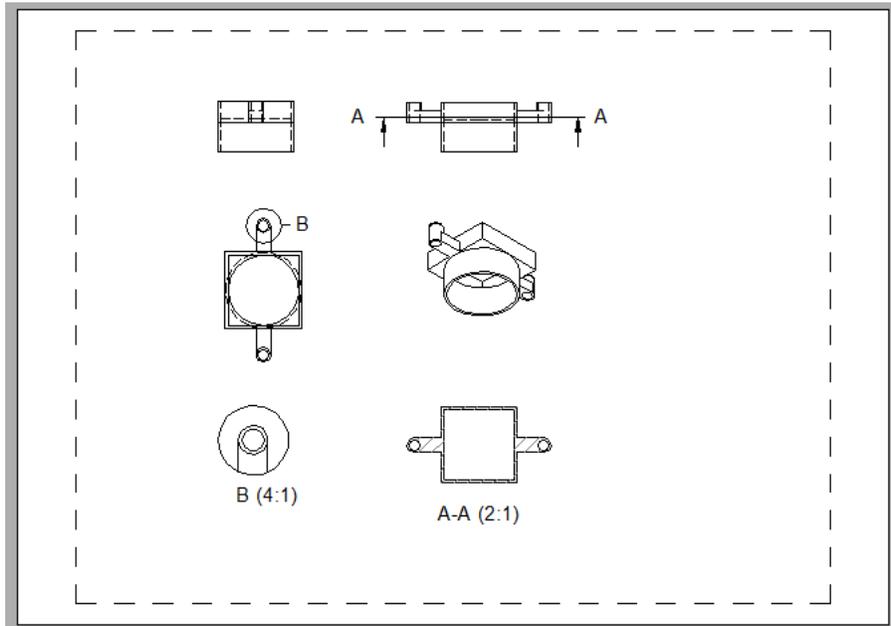
BOUNDARY OPTION

Determines if the boundary of the detail view is a circle or a rectangle.

MODEL EDGE OPTION

Specifies how the connection line is drawn between the two views (detail and in the parent view), smooth with the border or with the connection line.

Here is how the final 2D plan could look. The viewports are turned off and some viewports are moved (with the Move command) and resized (with the Custom Scale property). The next step would be to dimension the parts.



Summary

You learned how to construct 3D models using traditional and modern commands for creating and editing bodies. You also saw how to generate 2D drawings from the model, semi-automatically.

Next, you learn how to control the size and positioning of entities through constraints and parameters.

Dimensional & Geometric Constraints

Here you learn how to create drawings that are *constrained*, where dimensions determine the sizes of entities, and where geometry is locked, thereby determining the locations of entities relative to each other. Together with constraints, parameters determine the positions of entities through formulae. This chapter is for users of BricsCAD Pro and Platinum editions.

IN THIS CHAPTER

- Applying dimensional constraints
- Using 2D geometric constraints
- Controlling constraints through parameters

Working with Constraints

BricsCAD allows you to control the size of entities in drawings through constraints. There are two types of constraints: one of the types determines the *size* of entities, the other determines *position*.

- > **Dimensional constraints** regulate the sizes of entities, and the distances between them
- > **Geometric constraints** determine the position of entities relative to others

BricsCAD Classic and Pro provide 2D constraints. BricsCAD Platinum adds 3D constraints, which are applied to faces and edges of 3D bodies but do not work with 2D entities. (Three-dimensional constraints are not covered by this book.)

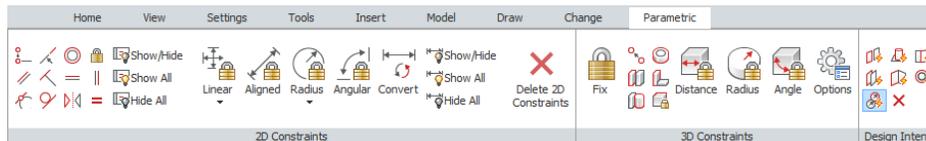
There are a number of commands for applying and removing constraints, but I find it easiest to just use the 2D Constraints toolbar or the Parametric tab on the ribbon.

To see the 2D Constraints toolbar, right-click any toolbar or ribbon, and then choose **BRICSCAD | 2D Constraints**:



Center group of 12 red buttons: geometric constraints; right group of 8 yellow buttons: dimensional constraints

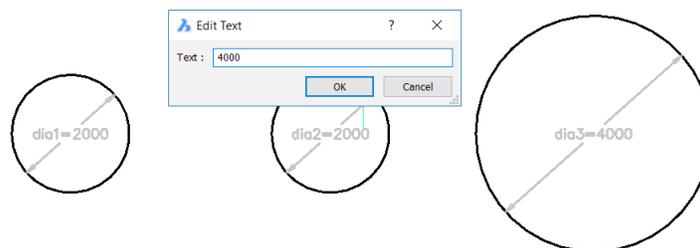
To see the tab on the ribbon, switch to the “3D Modeling” workspace, and then choose the ribbon’s **Parametric** tab:



Left side of ribbon: 2D constraints; right side: 3D constraints and design intent (Platinum edition only)

ABOUT DIMENSIONAL CONSTRAINTS

The great thing about dimensional constraints is that you use them to control the size of entities. For instance, apply a diameter dimensional constraint to a circle. Increase the value of the constraint, and BricsCAD forces the circle to become larger. To edit the value, you just double-click the dimension text.



Left: Circle with diameter of 2000 units; center: Editing the value of the dimensional constraint; right: New 4000-value of parameter forces circle to grow larger

QUICK SUMMARY OF DIMENSIONAL CONSTRAINTS

There are two sets of commands for applying constraints. One is the all-purpose **DimConstraint** command, which is useful for seeing a list of all modes. The other set consists of the individual commands, one for each dimensional constraint:

DIMCONSTRAINT COMMAND

: **dimconstraint**

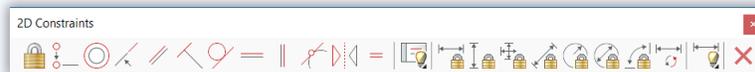
Select associative dimension to convert or [LLinear/Horizontal/Vertical/Aligned/ANgular/Radial/Diameter] <LLinear>: (Enter an option.)

DIMENSIONAL CONSTRAINT COMMANDS

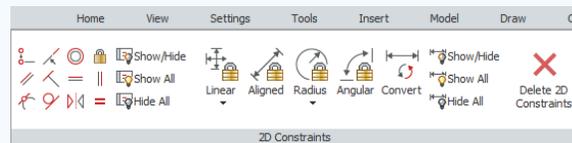
The other set of commands consists of ones specific to each constraint mode, as listed in the table below.

Icon	Type of Constraint	Command	Constraining Action
	Linear	DcLinear	Horizontal or vertical distance
	Horizontal	DcHorizontal	Horizontal (X) distance between two points
	Vertical	DcVertical	Vertical (Y) distance between two points
	Aligned	DcAligned	Distance between two points at any angle
	ANgular	DcAngular	Angle between two lines or linear polyline segments; angle of an arc or polyline arc;
	Radial	DcRadius	Radius of a circle, arc, or polyline arc
	Diameter	DcDiameter	Diameter of a circle, arc, or polyline arc
	...	DimConvert	Converts associative dimensions to dimensional constraints

Dimensional constraints are applied in the current UCS. Icons for dimensional constraints are found at the right end of the 2D Constraints toolbar or ribbon tab:



Above: 2D constraints on right half of the toolbar; below: 2D constraints on ribbon

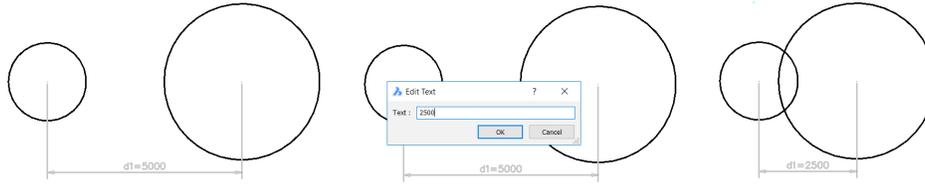


CONVERTING DIMENSIONS TO CONSTRAINTS

Dimensional constraints are very much like associative dimensions and, in fact, existing associative dimensions can be converted to constraints with the DimConstraint command's **Select Associative Dimension to Convert** option — or the DimConvert command.

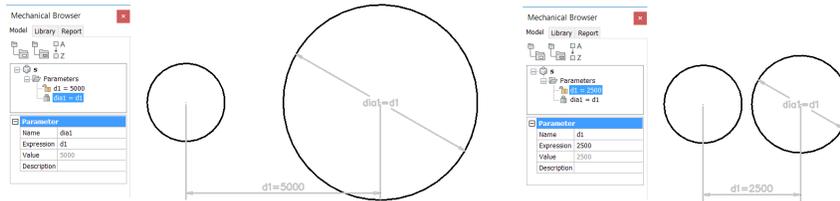
The command converts like to like. For instance, linear dimensions become linear constraints, radial dimensions become radial constraints, and so on.

You can place linear dimensional constraints between entities to control how far apart they are. Decrease the value of the constraint, and BricsCAD moves them closer together; increase it, and they are moved further apart. Again, to edit the value, just double-click the dimension text.



Left: Two circles separated by 5000 units; **center:** Distance being edited; **right:** Circles now separated by 2500 units

Better yet, the values of constraints can be determined by other constraints, and even by formulas. For instance, you can make the diameter of a circle equal to the linear distance between two entities. When you change one dimension, the linked ones follow suit. For example, the figure below shows that I made the value of *dia1* (diameter of one circle) equal to *d1* (distance between circles) using a simple formula, $dia1=d1$.

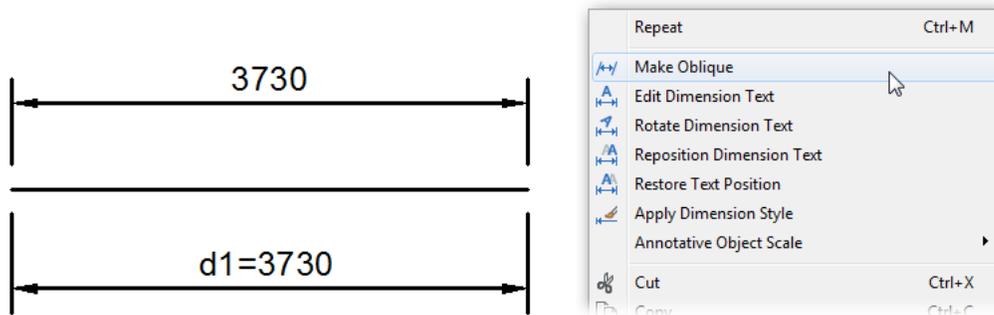


Left: Diameter of first circle = distance between circles; **right:** Changing the distance to 2500 units makes the second circle smaller

To change the values of dimensional constraints, open the Mechanical Browser with the **bmBrowser** command. To remove a dimensional constraint, simply select it and then press **Del** (or else use the **Erase** command). A concise reference to all dimensional constraints is found in the boxed text on the facing page.

Using Dimensional Constraints

Using dimensional constraints in drawings is very much like placing regular dimensions. Except for the text, they look exactly like each other; you can apply dimension styles to them. The only visual difference between the two is the 'd1=' text that prefixes the constraint value; this is how you identify dimensions that are constraints.



Left: Associative dimension (top) and dimensional constraint (bottom) applied to the same line; **right:** Editing options for dimensional constraints are the same as for associative dimensions

The difference between the two types of dimensions is this: whereas associative dimensions are controlled *by* the entity, dimensional constraints *do* the controlling. They specify the sizes of entities, overruling what ever you may have drawn. Here is a tutorial to illustrate the differences.

1. Start BricsCAD with a new drawing.
2. Draw a line with the **Line** command. The length is not critical.
3. Dimension the line with the **DimLinear** command.
4. Use the **dcLinear** command to apply a dimensional constraint to the same line. ('dc' is short for dimensional constraint.)

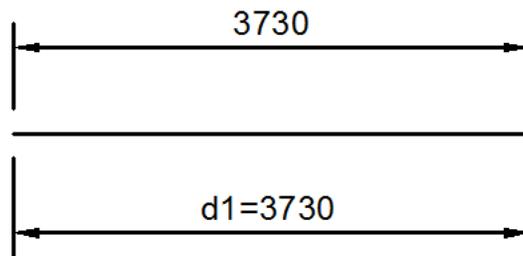
: **dcLinear**

Specify first constraint point or [Entity] <Entity>: (Press **Enter**)

Select an entity: (Choose the line.)

Specify dimension line location: (Move the cursor, and then click)

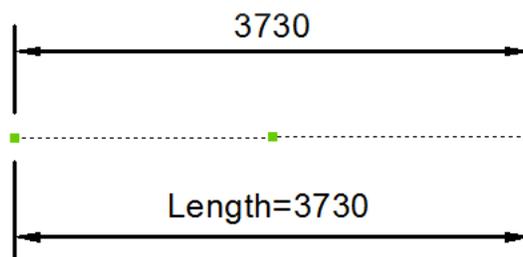
Dimension text <3730>: (Press **Enter**)



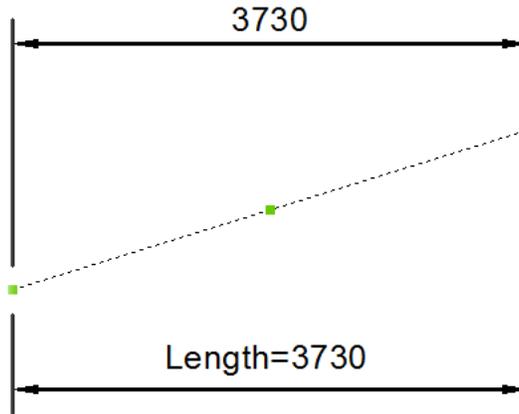
In the figure above, the two linear dimensions look identical — other than the 'd1=' name. The 'd' is short for "distance," and the '1' indicates this is the first distance constraint placed in the drawing. You can change this name through the Properties panel.

Continuing with the tutorial, you now edit the line and the dimensions to see how one affects the other.

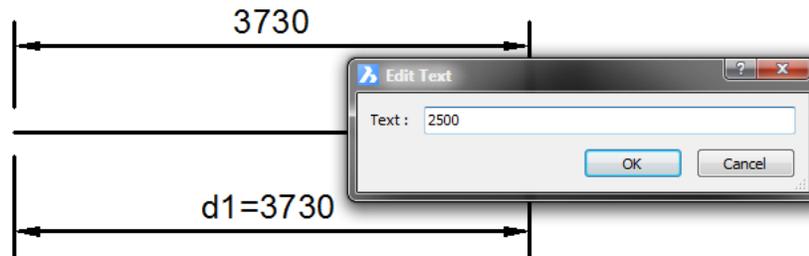
5. Edit the line by stretching one of its end grips; see figure below. Notice that you cannot. This is because the length of the line is "locked" by the dimensional constraint.



6. You can, however, use the center grip to move the line and its dimension as a unit. As well, you can stretch the line vertically and it will change its angle, because the position and the height are not locked — only the horizontal length is locked by dLinear.

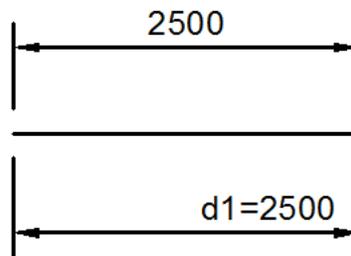


7. Now change the value of the constraint to see what happens to the line:
 a. Enter the **DdEdit** command, and then select the constraint value.



- b. Enter a new value, such as **2500**, and then twice press **Enter** to exit the dialog box.

Notice that the line changes its length to match the new value of the constraint, as does the distance measured by the associative dimension.



With it comes to dimensional constraints, their values control the entities' sizes, and so the entities cannot be edited directly.

QUICK SUMMARY OF 2D GEOMETRIC CONSTRAINTS

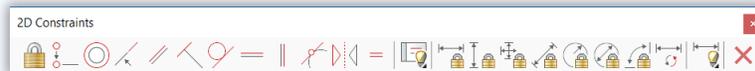
As with dimensional constraints, there are two sets of commands for applying geometric ones. One is the all-purpose **GeomConstraint** command, which is useful for seeing a list of all modes:

```
: geomconstraint
Enter constraint type [Horizontal/Vertical/Perpendicular/PARallel/Tangent/SMooth/Coinci-
dent/CONcentric/COLLinear/Symmetric/Equal/Fix] <CONcentric>:
```

The other set of commands consists of ones specific to each constraint mode, as listed in the table below.

Icon	Type of Constraint	Command	Constraining Action
	Horizontal	GcHorizontal	Keeps entities horizontal (parallel to the x-axis)
	Vertical	GcVertical	Keeps entities vertical (parallel to the y-axis)
	Perpendicular	GcPerpendicular	Keeps entities perpendicular to one other
	PARallel	GcParallel	Keeps entities parallel to one another
	Tangent	GcTangent	keeps circular and straight entities tangent
	SMooth	GcSmooth	Keep splines smooth with splines, lines, arcs, polylines
	Coincident	GcCoincident	Keeps end points attached, such as of two lines
	CONcentric	GcConcentric	Keeps circles, arcs, ellipses, and elliptical arcs centered
	COLLinear	GcCollinear	Makes linear entities to lie in the same line
	Symmetric	GcSymmetric	Keeps entities or points symmetric about mirror lines
	Equal	GcEqual	Gives curved entities the same radius; open entities the same length
	Fix	GcFix	Keeps entities fixed in-place in the drawing

Note that some geometric constraints, such as Horizontal, are applied relative to the current UCS.



Geometric constraints on the left half of the toolbar

ACCEPTABLE GEOMETRY

You can apply constraints to the following entities: lines, segments of polylines, circles, arcs, ellipses, elliptical arcs, and splines. Constraint points can be applied to the following geometric features:

Entity	Constraint Points
Arcs	Endpoints, center points, and midpoints
Block insertions	Insertion point
Circles	Center points
Ellipses	Center points
Elliptical arcs	Endpoints, center points, and midpoints
Lines	Endpoints and midpoints
Polyline line and arc segments	Endpoints, midpoints, center points of arc segments
Splines	Endpoints
Tables	Insertion point
Text, mtext, and attributes	Insertion point
Xref attachments	Insertion point

ABOUT GEOMETRIC CONSTRAINTS

Whereas dimensional constraint control the size of entities and distances between them, geometric constraints control their positions in the drawings. They act like semi-permanent entity snaps. Whereas esnaps are in effect only at the time that you create or edit an entity, geometric constraints remain in effect forever — until you remove them.

For instance, *horizontal* constraints force entities to be horizontal. Draw a line at any angle, and then apply the horizontal constraint: it snaps flat; see figure below.

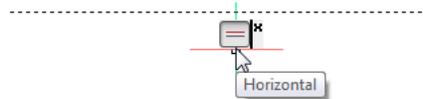


Left: Line drawn at an arbitrary angle; right: Line snapped flat by horizontal constraint

When you try to rotate the entity, you can't. It can be moved, shortened (or lengthened), but not rotated, because it is constrained. When you copy a constrained entity, the copy takes on the same constraint(s).

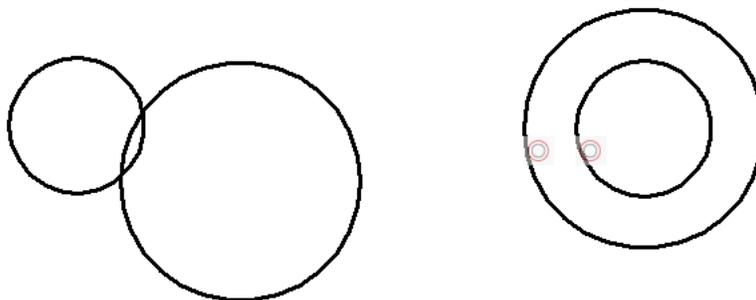
Identifying Geometric Constraints

Geometric constraints are identified by the small icon that floats near the associated entity. Pass the cursor over the icon and three things appear:



- > Tooltip identifies the name of the constraint, "Horizontal"
- > **X** appears next to the tooltip; clicking the X makes the icon disappear (the constraint remains in effect)
- > The associated entity is highlighted

Here is another for-instance: **concentric** constraints force circles to be concentric. Move one, and the other moves with it.



Left: Two circles placed arbitrarily in the drawing; right: Larger circle made concentric to the first one

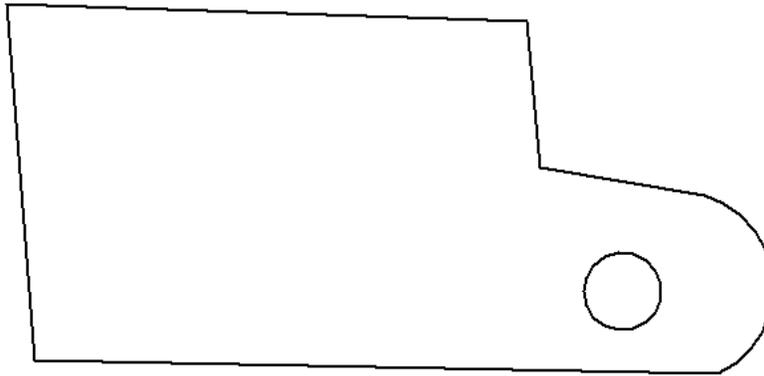
The gcConcentric command is order-dependent: the first entity you select is the master, meaning that the second entity you select follows it. (Unlike dimensional constraints, geometric constraints do not take formulae.)

To remove a geometric constraint, you have to use the **DelConstraint** command; you cannot simply “erase” geometric constraints.

All the kinds of dimensional constraints found in BricsCAD are listed in the boxed text on the facing page.

Using Geometric Constraints

The easiest way to understand geometric constraints is to work through a tutorial. Below, I drew a rough sketch of a bracket. I drew it badly deliberately to show off the beneficial effect of constraints. You can access the *bracket-ragged.dwg* file from my public Dropbox folder: <https://dl.dropboxusercontent.com/u/28941239/Inside-BricsCAD-Tutorial-Files.zip>.)

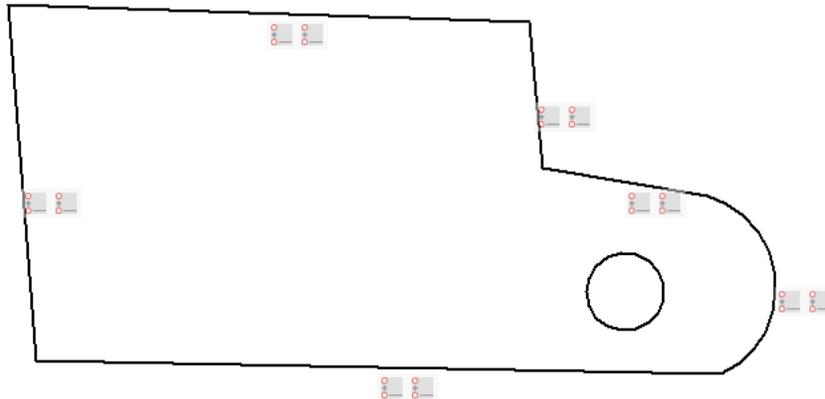


1. Start BricsCAD with a new drawing.
2. Draw a shape similar to the one illustrated above with the **Line**, **Arc**, and **Circle** commands. The exact size and exact placement of entities is not critical.
3. The 2D Constraints toolbar assists you in assigning geometric constraints to entities. Open it, as follows:
 - a. Right-click any toolbar.
 - b. From the shortcut menu, choose **BRICSCAD**, and then choose **2D Constraints**.



4. The first step, usually, is to connect all lines with the **gcCoincident** command. This particular constraint mode forces the endpoints of lines and arcs to stick together. The good news is that you don't need to pick entities individually:
 - a. From the 2D Constraints toolbar, choose the  Coincident button.
: `_gccoincident`
 - b. Use the command's **Autoconstrain** option to connect all lines in one fell swoop:
Select first point or [Entity/Autoconstrain] <Entity>: `a`
Select entities: `all`
 - c. Press **Enter** to end the command:
Select entities: (*Press Enter*)

5. Notice that tooltip-like  tags appear all over the drawing. They are known as “constraint bars,” and report the types of geometric constraints applied to entities. In this case, the tags are all the same, because of the coincident constraint.



6. Me, I find these bars annoying, because they clutter the drawing, and so I close them. You can, too:

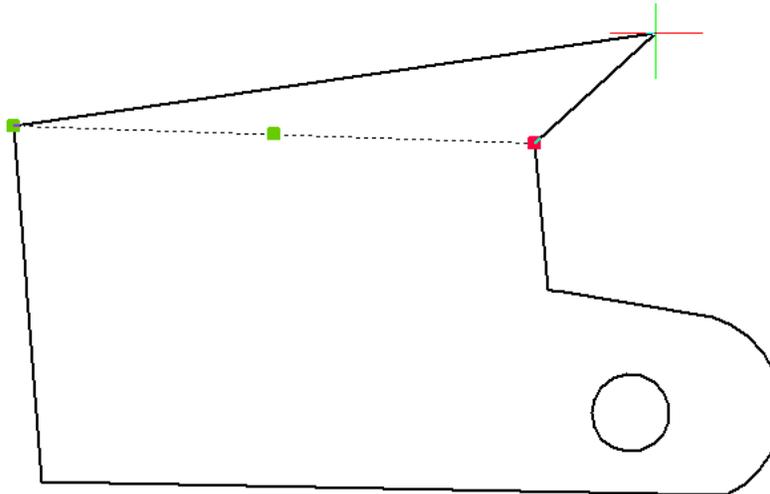
- > To turn off bars individually, click the  that appears in the upper right corner of each bar
- > To turn them off all at the same time, use the **ConstraintBar** command's **Hide** option, as follows:

```

: constraintbar
Select entities: all
Select entities: (Press Enter to continue)
Select option to [Show/Hide/Reset] constraints: h

```

7. To see the effect of the Coincident constraint, drag a line. Notice the other lines that are connected to it. This shows that Coincident is like a sticky bit of glue, making lines act like polylines.



8. Enter the **U** command to return the drawing as it was before.
9. To straighten out lines that are supposed to be horizontal, use the **gcHorizontal** command on the nearly horizontal ones, as follows:
 - a. In the 2D Constraints toolbar, click the  **Horizontal** button, and then follow the prompts in the command bar:

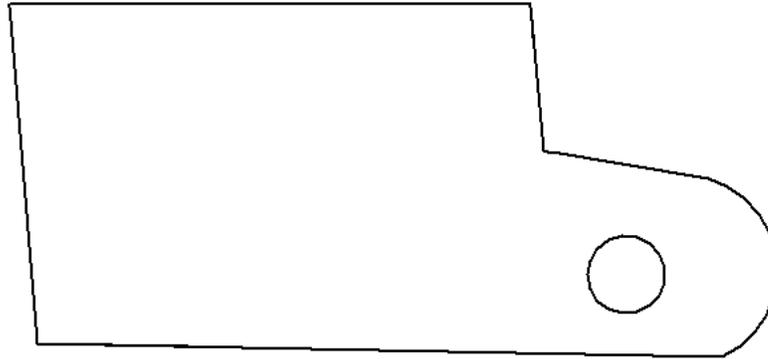
```

: _gchorizontal

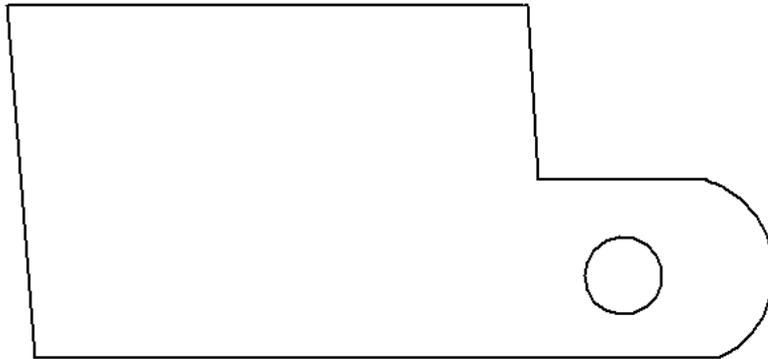
```

- b. Select a nearly-horizontal line:
Select an entity or [2Points] <2Points>: (Choose a Line)

Notice that it immediately goes precisely horizontal. The line changes its length as needed, so that it remains attached to its neighbors.



- c. Repeat the command to make other lines horizontal.



Now you need to straighten out the vertical lines. Here you have some options. You could apply any of these constraints:

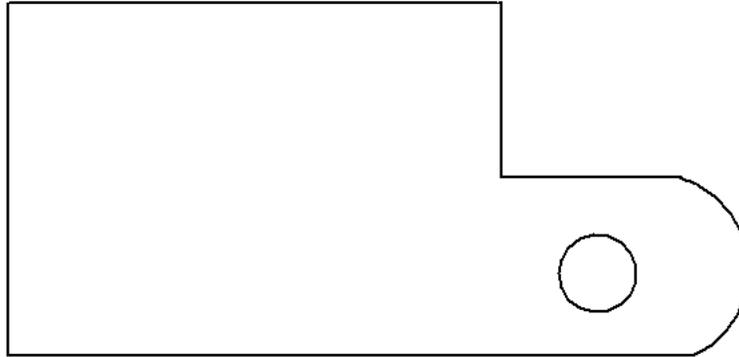
- > Two **vertical** constraints to make all the near-vertical lines truly vertical
- > Two **perpendicular** constraints to make vertical lines perpendicular to the horizontal ones
- > One **perpendicular** and one **parallel** constraint to make one vertical line perpendicular to one horizontal one, and then to make the remaining vertical one parallel to the first

Which approach you choose depends on your intent for the design: how do you want the lines to relate to each other?

10. I consider approach #2 is best, because I want vertical lines to all be perpendicular to the horizontals. Here is how to do this:
- a. Click the  Perpendicular button to start the **gcPerpendicular** command.
: `_gcperpendicular`
 - b. For this type of constraint, the selection order *matters*. Essentially, the second entity selected is made perpendicular to the first one. (The first is the master, the second is the slave.) Select a horizontal line:
Select first entity: (Choose a horizontal Line)
 - c. And then select an attached nearly-vertical one:
Select second entity: (Choose the vertical one)

Notice that the vertical line straightens out, because it is forced to be perpendicular to the horizontal one.

11. Repeat for the other nearly-vertical lines. The result should look similar to the figure below.

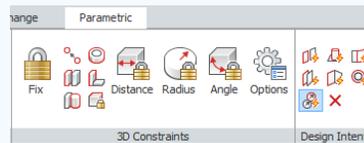


QUICK SUMMARY OF 3D CONSTRAINTS

The Platinum Edition of BricsCAD provides dimensional and geometric constraints work in three dimensions, as illustrated by the 3D Constraints toolbar and the 3D Parametric ribbon tab:



Above: 3D Constraints toolbar; below: 3D Constraints on the ribbon



The 3D constraints are applied with the **dmConstraint3d** command. ('dm' is short for direct modeling.)

dmconstraint3d

Select type of 3D constraint [Fix/Coincident/CONcentric/Parallel/PERpendicular/Tangent/RIGidSet/Distance/Radius/Angle]:

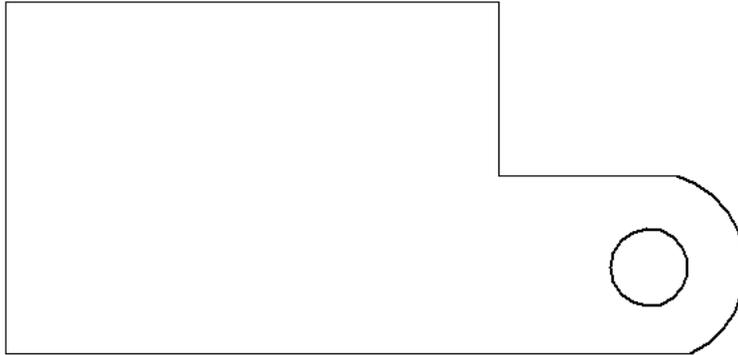
Icon	Type of Constraint	Command	Constraining Action
3D Geometric Constraints			
	Fix	DmFix3d	Fixes solids, or edges or faces of solids
	Coincident	DmCoincident3d	Fixes edges, faces, or an edge and a face of different solids
	Concentric	DmConcentric3d	Fixes two cylindrical, spherical or conical surfaces
	Parallel	DmParallel3d	Fixes two faces of a solid, or of different solids
	Perpendicular	DmPerpendicular3d	Fixes two faces of a solid, or of different solids
	Tangent	DmTangent3d	Fixes a face and a curved surface of different solids
	Rigid Set	DmRigidSet3d	Define a set of entities or sub-entities as a rigid body
3D Dimensional Constraints			
	Distance	DmDistance3d	Distance between two sub-entities of a solid or different solids
	Radius	DmRadius3d	Radius of cylindrical surfaces or circular edges
	Angle	DmAngle3d	Angle between the faces of a solid or of different solids

12. The circle needs a geometric constraint to fix its location. Use  **gcConcentric** to match the center point of the circle to that of the arc, as follows:

```

: _gcconcentric
Select first entity: (Pick the circle)
Select second entity: (Pick the arc)

```



Normally, selection order matters with Concentric constraint: the second entity shifts its position to be concentric with the first one selected. In this particular case, however, selection order did not matter, because the arc is fixed in place (through the earlier Coincident constraint), and so BricsCAD forced the circle to move.

13. Notice that the connection between the arc and the two tangent lines is not smooth. To force the arc's two ends smoothly meet the lines, use the **gcTangent** constraint, as follows:

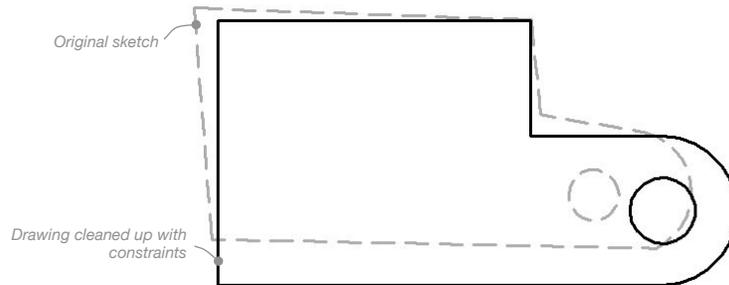
```

: _gctangent
Select first entity: (Pick the arc)
Select second entity: (Pick a Line)

```

Repeat for the second arc-line connection.

In the figure below, I overlapped the constrained drawing (shown in black) with the original rough sketch (shown in gray). It is easy to see how constraints fixed it up!



14. If you wish to see all the constraint bars again, turn them on like this:

```

: constraintbar
Select option to [Show/Hide/Reset] constraints: S
Select entities or ENTER to select all: (Press Enter)

```

15. With all the geometry fixed in place relative to one another, you can use dimensional constraints to size the bracket. The drawing illustrated below provides the values of the dimensions.

As you enter the values, notice that parts stretch their sizes to accommodate.

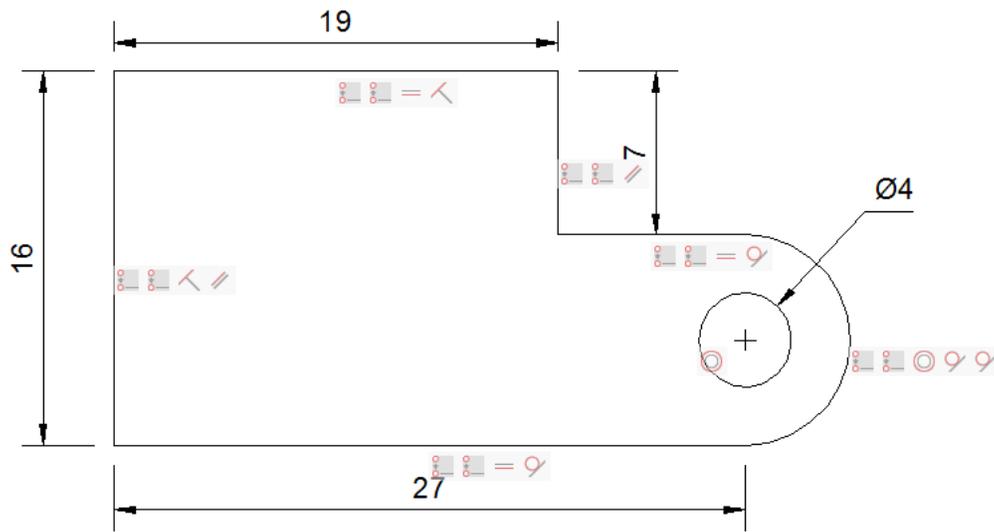
- > Use  **dcHorizontal** for horizontal dimensional constraints


```

: _dchorizontal
Specify first constraint point or [Entity] <Entity>: (Press Enter)
Select an entity: (Pick a Line)
Specify dimension line location: (Pick a point)
Dimension text <17.993713>: 19

```

- > Use  **dcVertical** for vertical dimensional constraints
- > Use  **dcDiameter** for diameter dimensional constraints



Summary

This concludes the tutorials on using constraints with BricsCAD. You learned how constraints can be used to clean up drawings, as well as to quickly change the sizes of them.

Concise Summary of Command Aliases

The fastest way to operate BricsCAD commands is through *aliases* — if you can remember them, because BricsCAD has aliases for nearly three hundred of its commands! Aliases are shortcuts for command names. For instance, **l is the alias for Line, while **la** is for Layer.**

On the following pages, aliases are twice listed alphabetically: first, in order of alias name, and then by command name.

While most aliases are true abbreviations, others provide continuity from history. For instance, the Color command has these aliases: Colour, because AutoCAD includes the British spelling for this command; DdColor and DdColour as the names for the first dialog box version; and SetColor as the IntelliCAD name for this command.

One-Letter Aliases

Even if you can learn just a few of them, then your drafting will proceed faster. The table below summarizes all of the one-letter aliases, and so could be considered to also be a list of some of the most important commands.

Command	Alias	Command	Alias
arc	a	move	m
block	b	mtext	t
circle	c	new	n
dimstyle	d	offset	o
erase	e	pan	p
explode	x	redraw	r
fillet	f	stretch	s
grid	g	u	u (<i>not an alias</i>)
hatch	h	view	v
insert	i	wblock	w
join	j	zoom	z
line	l		

BRICSCAD ALIASES SORTED BY ALIAS NAME

Alias _____ Command Name _____

A Aliases

a	Arc
aa	Area
ad	AttDisp
al	Align
ap	Aperture
ar	Array
array3d	3dArray
at	AttDef
-at	-AttDef
-ate	AttEdit
ate	EAttEdit
ax	AttExt
-ax	-AttExt

B Aliases

b	Block
-b	-Block
ba	Base
backgrounds	Background
bh	Hatch
-bh	-Hatch
bm	Blipmode
bo	Boundary
-bo	-Boundary
bpoly	Boundary
br	Break

C Aliases

c	Circle
cfg	Options
-ch	Change
ch	Properties
cha	Chamfer
cl	CopyLink
clip	XClip
closeall	WCloseAll
co	Copy
col	Color
-col	-Color
colour	Color
-colour	-Color
config	Options
cp	Copy
cui	Customize
cyl	Cylinder

Alias _____ Command Name _____

D Aliases

ddattdef	AttDef
ddatttext	AttExt
ddchprop	Properties
ddcolor	Color
ddcolour	Color
ddesnap	OSnap
ddinsert	Insert
ddlmodes	Layer
ddltype	Linetype
ddmodify	Properties
ddnew	NewWiz
ddosnap	OSnap
ddrename	Rename
ddrmodes	DSettings
ddstyle	Style
dducs	ExpUcs
dducsp	SetUcs
ddunits	Units
ddview	View
delete	Erase
di	Dist
div	Divide
do	Donut
doughnut	Donut
dr	Draworder
dv	DView
dwfout	Export
dx	DxfOut

Dimension Aliases

d	DimStyle
dal	DimAligned
dan	DimAngular
dba	DimBaseline
dce	DimCenter
dco	DimContinue
ddi	DimDiameter
ddim	DimStyle
ded	DimEdit
dimali	DimAligned
dimang	DimAngular
dimbase	DimBaseline
dimcont	DimContinue
dimdia	DimDiameter
dimed	DimEdit
dimension	Dim
dimhorizontal	DimLinear
dimlin	Dimlinear

Alias	Command Name
dimord	DimOrdinate
dimover	DimOverride
dimrad	DimRadius
dimrotated	DimLinear
dimsty	DimStyle
dimted	DimTEdit
dimvertical	DimLinear
dli	DimLinear
dor	DimOrdinate
dov	DimOverride
dra	DimRadius
ds	DimStyle
dst	DimStyle
-dst	-DimStyle
expdimstyles	DimStyle
setdim	DimStyle

E Aliases

e	Erase
ed	DdEdit
editlen	Lengthen
editplane	PEdit
el	Ellipse
esnap	-OSnap
ex	Extend
exit	Quit
exp	Export
expfonts	Style
expimages	Image
explayers	Layer
expltypes	Linetype
expstyle	Style
expstyles	Style
expviews	View
expxrefs	XRef
ext	Extrude

F Aliases

f	Fillet
face	3dFace
fi	Filter
finish	Materials
fog	RenderEnvironment
font	-Style
freehand	Sketch

G Aliases

g	Grid
geo	GeographicLocation
gr	DdGrips

Alias	Command Name
h	Hatch
-h	-Hatch
he	HatchEdit
hi	Hide

H Aliases

h	Hatch
-h	-Hatch
he	HatchEdit
hi	Hide

I Aliases

i	Insert
-i	-Insert
iad	ImageAdjust
iat	ImageAttach
icl	ImageClip
idpoint	Id
im	Image
imp	Import
in	Intersect
inf	Interfere
inpline	Xline
insal	InsertAligned
io	InsertObj
is	Isoplane
isolate	IsolateObjects

L Aliases

l	Line
la	Layer
-la	-Layer
las	LayerState
le	Leader
lead	Leader
len	Lengthen
li	List
lighting	Light
ll	Lightlist
ls	List
lt	Linetype
-lt	-Linetype
lts	LtScale

M Aliases

m	Move
ma	MatchProp
mat	Materials
matb	MatBrowserOpen
mesh	3dMesh
mi	Mirror
ml	Mline
mo	Properties
ms	Mspace
msnapshot	Mslide
mt	MText
mv	MView

Alias Command Name

N Aliases

n QNew
navvcube LookFrom

O Aliases

o Offset
op Open
or Orthogonal
ortho Orthogonal
os OSnap
-os -OSnap

P Aliases

p Pan
-p -Pan
pa PasteSpec
pe PEdit
pl PLine
plane Solid
planviewint Apparent
po Point
pol Polygon
polyline PLine
ppreview Preview
pr Properties
prc PropertiesClose
pre Preview
preferences Options
prefs Options
props Properties
ps PSpace
pso PolySolid
pu Purge
-pu -Purge
pyr Pyramid

Q Aliases

qt QText

R Aliases

r Redraw
ra RedrawAll
re Regen
rea RegenAll
rec Rectang
rect Rectang
rectangle Rectang
reg Region
ren Rename
-ren -Rename
rendscr RenderWin

Alias Command Name

rev Revolve
ri Reinit
rm DSettings
rmat Materials
ro Rotate
roptions RenderPresets
rr Render

S Aliases

s Stretch
sa Save
sc Scale
scr Script
se DdSelect
sec Section
selgrip SelGrips
set SetVar
setcolor Color
setesnap OSnap
setlayer LayMCur
setrender RpPref
setuv MaterialMap
setvpoint DdVpoint
sha Shade
sl Slice
sn Snap
so Solid
sp Spell
spe SplineEdit
spl Spline
ssm SheetSet
st Style
su Subtract
sun SunProperties

T Aliases

t MText
-t -Text
ta Tablet
th Thickness
ti Time
tol Tolerance
tor Torus
tr Trim
tx Text

U

uc ExpUcs
ucp SetUcs
un Units
-un -Units
undelete Oops
unerase Oops

Alias	Command Name
unhideobjects	UnisolateObjects
unhide	UnisolateObjects
unisolate	UnisolateObjects
uni	Union

V Aliases

v	View
-v	-View
vba	Vbalde
viewctl	DdVpoint
viewpoint	VPoint
-viewpoint	VPoint
viewports	VPorts
vl	VpLayer
vp	DdVpoint
-vp	VPoint
-vpoint	VPoint
vport	VPorts
vs	VSlide
vscurrent	ShadeMode
vsnapshot	VSlide
vw	VPorts

W Aliases

w	WBlock
we	Wedge
wi	Wmfln
wo	WmfOut

Alias	Command Name
-------	--------------

X Aliases

x	Explode
xa	XAttach
xb	ExpBlocks
-xb	XBind
xl	XLine
xr	XRef
-xr	-XRef

Z Alias

z	Zoom
---	------

3 Aliases

3a	3dArray
3dline	Line
3dmirror	Mirror3d
3drotate	Rotate3d
3f	3dFace
3m	Mirror3d
3p	3dPoly
3r	Rotate3d

BRICSCAD ALIASES SORTED BY COMMAND NAME

Command Name Alias(es)

A Commands

Align	al
Aperture	ap
Apparent	planviewint
Arc	a
Area	aa
Array	ar
AttDef	at, ddatdef
-AttDef	-at
AttDisp	ad
AttEdit	-ate
-AttEext	-ax
AttExt	ax, ddattext

B Commands

Background	backgrounds
Base	ba
Blipmode	bm
Block	b
-Block	-b
Boundary	bo, bpoly
-Boundary	-bo
Break	br

C Commands

Chamfer	cha
Change	-ch
Circle	c
Color	col, colour, ddcolor, ddcolour, setcolor
-Color	-colour,-col
Copy	co, cp
CopyLink	cl
Customize	cui
Cylinder	cyl

D Commands

DdEdit	ed
DdGrips	gr
DdSelect	se
DdVpoint	setvpoint, viewctl, vp
Dist	di
Divide	div
Donut	do
Donut	doughnut
Draworder	dr
DSettings	ddrmodes, rm
DView	dv
DxfOut	dx

Command Name Alias(es)

Dimension Commands

Dim	dimension
DimAngular	dimang, dan
DimAligned	dal, dimali
DimBaseline	dba, dimbase
DimCenter	dce
DimContinue	dco, dimcont
DimDiameter	ddi, dimdia
DimEdit	ded, dimed
DimLinear	dimhorizontal, dimlin, dimrotated, dimvertical, dli
DimOrdinate	dimord, dor
DimOverride	dimover, dov
DimRadius	dimrad, dra
DimStyle	expdimstyles, d, ddim, dimsty, ds, dst, setdim
-DimStyle	-dst
DimTEdit	dimted

E Commands

EAttEdit	ate
Ellipse	el
Erase	delete, e
ExpBlocks	xb
Explode	x
Export	dwfout, exp
ExpUcs	dducs, uc
Extend	ex
Extrude	ext

F Commands

Fillet	f
Filter	fi

G Commands

GeographicLocation	geo
Grid	g

H Commands

Hatch	bh, h
-Hatch	-bh,-h
HatchEdit	he
Hide	hi

I Commands

Id	idpoint
Image	expimages, im

Alias	Command Name
ImageAttach	iat
ImageAdjust	iad
ImageClip	icl
Import	imp
Insert	ddinsert, i
-Insert	-i
InsertAligned	insal
InsertObj	io
Interfere	inf
Intersect	in
IsolateObjects	isolate
Isoplane	is

L Commands

Layer	ddlmodes, explayers, la
-Layer	-la
LayerState	las
LayMCur	setlayer
Leader	le, lead
Lengthen	editlen, len
Light	lighting
Lightlist	ll
Line	3dline, l
Linetype	ddltype, expltypes, lt
-Linetype	-lt
List	li, ls
LookFrom	navvcube
LtScale	lts

M Commands

MatBrowserOpen	matb
MatchProp	ma
MaterialMap	setuv
Materials	finish, mat, rmat
Mirror	mi
Mirror3d	3dmirror, 3m
Mline	ml
Move	m
MSlide	msnapshot
MSpace	ms
MText	mt, t
MView	mv

N Command

NewWiz	ddnew
--------	-------

O Commands

Offset	o
Oops	undelete, unerase
Open	op
Options	cfg, config, preferences, prefs
Orthogonal	or, ortho

Alias	Command Name
OSnap	ddesnap, ddosnap, os, setesnap
-OSnap	esnap,-os

P Commands

Pan	p
-Pan	-p
PasteSpec	pa
PEdit	editpline, pe
PLine	pl, polyline
Point	po
Polygon	pol
PolySolid	pso
Preview	ppreview, pre
Properties	ch, ddchprop, ddmodify, mo, pr, props
PropertiesClose	prc
PSpace	ps
Purge	pu
-Purge	-pu
Pyramid	pyr

Q Commands

QNew	n
QText	qt
Quit	exit

R Commands

Rectang	rec, rect, rectangle
Redraw	r
RedrawAll	ra
Regen	re
RegenAll	rea
Region	reg
Reinit	ri
Rename	ddrename, ren
-Rename	-ren
Render	rr
RenderEnvironment	fog
RenderPresets	roptions
RenderWin	rendscr
Revolve	rev
Rotate	ro
Rotate3d	3drotate, 3r
RpPref	setrender

S Commands

Save	sa
Scale	sc
Script	scr
Section	sec
SelGrips	selgrip
SetUcs	dducsp, ucp
SetVar	set

Alias	Command Name
Shade	sha
ShadeMode	vscurrent
SheetSet	ssm
Sketch	freehand
Slice	sl
Snap	sn
Solid	plane, so
Spell	sp
Spline	spl
SplinEdit	spe
Stretch	s
Style	ddstyle, expfonts, expstyle, expstyles, st
-Style	font
Subtract	su
SunProperties	sun

T Commands

Tablet	ta
-Text	-t
Text	tx
Thickness	th
Time	ti
Tolerance	tol
Torus	tor
Trim	tr

U Commands

Union	uni
UnisolateObjects	unhideobjects, unhide, unisolate
Units	ddunits, un
-Units	-un

V Commands

Vbalde	vba
View	ddview, expviews, v
-View	-v
VpLayer	vl
VPoint	viewpoint,-viewpoint,-vp,-vpoint
VPorts	viewports, vport, vw
VSlide	vs, vsnapshot

Alias	Command Name
W Commands	
WBlock	w
WCloseAll	closeall
Wedge	we
Wmfln	wi
WmfOut	wo

X Commands

XAttach	xa
XBind	-xb
XClip	clip
Xline	inflight
XLine	xl
XRef	expxrefs, xr
-XRef	-xr

Z Command

Zoom	z
------	---

3 Commands

3dArray	3a, array3d
3dFace	3f, face
3dMesh	mesh
3dPoly	3p

TIP Some editing actions require no commands or aliases at all. Click an entity to move it, or its grip points; double-click to change its properties.

Concise Summary of System Variables and Settings

BricsCAD uses variables to store and report settings affecting the program and drawings. There are two types of variables: *system* variables that mimic the names and values from AutoCAD, and *preference* variables unique to BricsCAD. You access and change variables through a dialog box (**Settings** command) or directly on the command line (**SetVar** command).

This appendix lists in alphabetical order all 875 variable names found in V17.1.02.

UPPERCASE text indicates the name is also found in AutoCAD as a system variable

MixedCase text means the variable is a *preference*, and so is unique to BricsCAD

Blue text indicates that the variable is new since the last edition of this book

StrikeThrough text indicates the variable was removed from BricsCAD

userid refers to your computer login name

When you see **R/O** (read-only), it means that you cannot change the variable's value. The **Format** column reports the format in which values are saved:

- > **bool** Boolean (true or false, yes or no, 1 or 0)
- > **int** or **short** Integer number (a number without decimal point with a maximum value of 32768)
- > **long** Long integer (integers larger than 32,768)
- > **pt2d** 2D point (x,y)
- > **pt3d** 3D point (x,y,z)
- > **real** Real number (a number with decimal point)
- > **str** String (text)

Variable Name	Format	R/O	Default Value
ACADLSPASDOC	bool		0
ACADPREFIX	str	read-only	"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\Support\; C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Support\; C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Fonts\; C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Help\en_US\"
ACADVER	str	read-only	"20.0 BricsCAD"
AcisHlrResolution	real		-1
ACISOUTVER	int		70
AcisSaveAsMode	int		0
AFLAGS	int		0
AllowTabExternalMove	bool		1
AllowTabMove	bool		1
AllowTabSplit	bool		1
ANGBASE	real		0
ANGDIR	bool		0
ANNOALLVISIBLE	int		1
ANNOAUTOSCALE	short		-4
AnnoSelected	int	read-only	0
ANNOTATEDWG	bool		0
AntiAliasRender	short		2
AntiAliasScreen	short		1
APBOX	bool		0
APERTURE	int		10
AREA	real	read-only	0
AREAPREC	short		-1
AREAUNITS	string		"in ft mi µm mm cm m km"
ARRAYASSOCIATIVITY	bool		1
ARRAYEDITSTATE	bool	read-only	0
ATTDIA	bool		0
ATTMODE	int		1
AttractionDistance	int		3
ATTREQ	bool		1
AUDITCTL	bool		0
AuditErrorCount	int	read-only	0
AUNITS	int		0
AUPREC	int		0
AUTOCOMPLETEDELAY	real		0.3
AUTOCOMPLETEMODE	int		47
AUTOMENULOAD	bool		1
AutoResetScales	short		0
AutosaveChecksOnlyFirstBitDBMOD	bool		1
AUTOSNAP	int		119
AutoTrackingVecColor	int		171
AutoVpFitting	bool		1
AXISMODE	bool		0
AXISUNIT	pt3d		X= 0 Y= 0 Z= 0

B Variables

BACKGROUNDPLOT	int		2
BACKZ	real	read-only	0
BASEFILE	str		"Default-mm.dwt"
BINDTYPE	bool		0

Variable Name	Format	R/O	Default Value
BKGCOLOR	int		7
BKGCOLORPS	int		7
BLIPMODE	bool		0
BLOCKEDITLOCK	bool		0
BLOCKEDITOR	bool		0
BlocksPath	str		"C:\Users\userid\Documents\"
bmAutoUpdate	int		1
bmForceUpdateMode	int		0
bmReportPanel	bool		0
BndLimit	long		1000
BVMODE	long		0

C Variables

CACHELAYOUT	bool		1
CAMERADISPLAY	bool		0
CAMERAHEIGHT	real		0
CANNOSCALE	string		"1:1"
CANNOSCALEVALUE	real	read-only	1
CDATE	real	read-only	20160211.15522
CECOLOR	str		"ByLayer"
CELTSCALE	real		1
CELTYPE	str		"ByLayer"
CELWEIGHT	int		-1
CETRANSPARECNY	str		"ByLayer"
CGEOCS	str	read-only	""
CHAMFERA	real		0
CHAMFERB	real		0
CHAMFERC	real		0
CHAMFERD	real		0
CHAMMODE	int		0
ChapooLog	int		0
ChapooLogVerbose	boolean		0
ChapooOnModified	int		1
ChapooServer	str		"https://my.chapoo.com/"
ChapooTempFolder	str		"C:\Users\userid\AppData\Local\Temp\Chapoo\"
ChapooUploadDependencies	short		1
ChapooWebsite	str		"http://www.chapoo.com/"
CheckDwlPresence	long		0
CIRCLERAD	real		0
CLAYER	str		"0"
ClipBoardFormat	int		1
CLIPBOARDFORMATS	long		127
CLISTATE	int	read-only	1
CloseChecksOnlyFirstBitDBMOD	bool		0
CMATERIAL	str		"ByLayer"
CMDACTIVE	int	read-only	1
CMDDIA	bool		1
CMDECHO	bool		1
CmdLineEditBgColor	str		"#fefefe"
CmdLineEditFgColor	str		"#202020"
CmdLineFontName	str		"Consolas"
CmdLineFontSize	int		10
CmdLineListBgColor	str		"#ecf1ff"
CmdLineListFgColor	str		"#000000"

Variable Name	Format	R/O	Default Value
CMDLNTEXT	str		":"
CMDNAMES	str	read-only	"SETTINGS"
CMLEADERSTYLE	str		"Standard"
CMLJUST	int		0
CMLSCALE	real		1
CMLSTYLE	str		"Standard"
COLORX	int		11
COLORY	int		112
COLORZ	int		150
COMAcadCompatibility	bool		0
COMPASS	bool		0
CONSTRAINTBARDISPLAY	short		3
ContinuousMotion	bool		0
COORDS	int		1
COPYMODE	int		0
CLOTSTYLE	str		"ByColor"
CPROFILE	str	read-only	"Default"
CREATEVIEWPORTS	bool		1
CROSSINGAREACOLOR	int		91
CTAB	str		"Model"
CTABLESTYLE	str		"Standard"
Ctrl3DMouse	short		1
CTRLMOUSE	int		1
CURSORSIZE	int		3
CVPORT	int		2

D Variables

DATE	real	read-only	2456335.6613464
DBCSTATE	bool	read-only	0
DBLCLKEDIT	bool		1
DBMOD	int	read-only	0
DCTCUST	str		""
DCTMAIN	str		"en_US.dic"
ddBetweenKnots	int		2
ddFastMode	bool		0
ddGridAspectRatio	real		0
ddMaxFacetEdgeLength	real		0
ddMaxNumGridLines	long		10000
ddNormalTol	real		15
ddPointsPerEdge	long		0
ddSurfaceTol	real		0
ddUseFacetRES	bool		1
DEFAULTLIGHTING	bool		0
DefaultLightShadowBlur	short		8
DefaultNewSheetTemplate	str		""
DEFPLSTYLE	str		"Normal"
DEFPLSTYLE	str		"ByColor"
DeleteTool	bool		1
DELOBJ	int		1
DEMANDLOAD	int		3
DGNFRAME	int		2
DGNOSNAP	bool		1
DIASAT	bool	read-only	0
DisplaySnapMarkerInAllViews	bool		0

Variable Name	Format	R/O	Default Value
DisplayTooltips	bool		1
DISPPAPERBKG	bool		1
DISPPAPERMARGINS	bool		1
DISPSILH	bool		0
DISTANCE	real	read-only	0
DMAUTOUPDATE	bool		1
DmExtrudeMode	short		0
DMRECOGNIZE	int		0
DockPriority	int		1
DocTabPosition	short		0
DONUTID	real		0.5
DONUTOD	real		1
DRAGMODE	int		2
DragModeHide	short		0
DRAGMODEINTERRUPT	short		1
DRAGOPEN	int		1
DRAGP1	int		10
DRAGP2	int		25
DRAGSNAP	bool		0
DrawingPath	str		"C:\Users\userid\Documents\"
DrawingViewPreset	str		"none"
DrawingViewPresetScale	str		""
DRAWORDERCTL	int		3
DWFFRAME	int		2
DWFOSNAP	bool		1
DwfVersion	int		2
DWGCHECK	int		0
DWGCODEPAGE	str	read-only	"ANSI_1252"
DWGNAME	str	read-only	"Drawing1.dwg"
DWGPREFIX	str	read-only	"C:\Program Files (x86)\Bricsys\BricsCAD V17x64\"
DWGTITLED	bool	read-only	0
DXEVAL	int		12
DxfTextAdjustAlignment	bool		0
DYNCONSTRAINTMODE	bool		1
DYNDIGRIP	int		31
DynDimColorHot	int		142
DynDimColorHover	int		142
DynDimDistance	real		1
DynDimLineType	int		1
DYNDIVIS	int		1
DynInputTransparency	int		65
DYNMODE	int		3

Dimensions Variables

DIMADEC	int		0
DIMALT	bool		0
DIMALTD	int		2
DIMALTF	real		25.4
DIMALTRND	real		0
DIMALTTD	int		2
DIMALTTZ	bool		0
DIMALTU	int		2
DIMALTZ	int		0
DIMANNO	bool	read-only	0

Variable Name	Format	R/O	Default Value
DIMAPOST	str		""
DIMARCSYM	int		0
DIMASO	bool		1
DIMASSOC	int		2
DIMASZ	real		0.18
DIMATFIT	int		3
DIMAUNIT	int		0
DIMAZIN	int		0
DIMBLK	str		""
DIMBLK1	str		""
DIMBLK2	str		""
DIMCEN	real		0.09
DIMCLRD	int		0
DIMCLRE	int		0
DIMCLRT	int		0
DIMDEC	int		4
DIMDLE	real		0
DIMDLI	real		0.38
DIMDSEP	str		"0"
DIMEXE	real		0.18
DIMEXO	real		0.0625
DIMFIT	int		3
DIMFRAC	int		0
DIMFXL	real		1
DIMFXLON	bool		0
DIMGAP	real		0.09
DIMJOGANG	real		0.7853981634
DIMJUST	int		0
DIMLDRBLK	str		""
DIMLFAC	real		1
DIMLIM	bool		0
DIMLTEX1	str		""
DIMLTEX2	str		""
DIMLTYPE	str		""
DIMLUNIT	int		2
DIMLWD	int		-2
DIMLWE	int		-2
DIMPOST	str		""
DIMRND	real		0
DIMSAH	bool		0
DIMSCALE	real		1
DIMSD1	bool		0
DIMSD2	bool		0
DIMSE1	bool		0
DIMSE2	bool		0
DIMSHO	bool		1
DIMSOXD	bool		0
DIMSTYLE	str	read-only	"Standard"
DIMTAD	int		0
DIMTDEC	int		4
DIMTFAC	real		1
DIMTFILL	int		0
DIMTFILLCLR	str		"BYBLOCK"
DIMTIH	bool		1

Variable Name	Format	R/O	Default Value
DIMTIX	bool		0
DIMTM	real		0
DIMTMOVE	int		0
DIMTOFL	bool		0
DIMTOH	bool		1
DIMTOL	bool		0
DIMTOLJ	int		1
DIMTP	real		0
DIMTSZ	real		0
DIMTVP	real		0
DIMTXSTY	str		"Standard"
DIMTXT	real		0.18
DIMTXTDIRECTION	bool		0
DIMTZIN	int		0
DIMUNIT	int		2
DIMUPT	bool		0
DIMZIN	int		0

E Variables

EDGEMODE	bool		0
ELEVATION	real		0
EnableAttraction	bool		1
EnableHyperlinkMenu	bool		1
EnableHyperlinkTooltip	bool		0
ERRNO	int		0
EXPERT	int		0
ExplnsAlign	bool		0
ExplnsAngle	real		0
ExplnsFixAngle	bool		1
ExplnsFixScale	bool		1
ExplnsScale	real		1
EXPLMODE	bool		1
EXPORTMODELSPACE	int		0
EXPORTPAGESETUP	int		0
EXPORTPAPERSPACE	int		0
EXTMAX	pt3d	read-only	-1.0000E+20,-1.0000E+20,-1.0000E+20
EXTMIN	pt3d	read-only	1.0000E+20,1.0000E+20,1.0000E+20
EXTNAMES	bool		1

F Variables

FACETRATIO	int		0
FACETRES	real		0.5
FEATURECOLORS	bool		1
FIELDDISPLAY	bool		1
FIELDDEVAL	int		31
FILEDIA	bool		1
FILLETRAD	real		0.5
FILLMODE	bool		1
FLATLAND	bool		Off
FONTALT	str		"simplex.shx"
FONTMAP	str		"default.fmp"
FRAME	int		3
FRONTZ	real	read-only	0

Variable Name	Format	R/O	Default Value
FULLOPEN	int	read-only	1

G Variables

GDIOBJECTS	long	read-only	3768
GENERATEASSOCVIEWS	bool		0
GEOLATLONGFORMAT	int		1
GEOMARKERVISIBILITY	bool		1
GetStarted	bool		1
GLSWAPMODE	int		2
GradientColorBottom	str		"#d2d2d2"
GradientColorMiddle	str		"#fafafa"
GradientColorTop	str		"#ffffff"
GradientMode	str		"0"
GRIDAXISCOLOR	int		252
GRIDDISPLAY	int		3
GRIDMAJOR	int		5
GRIDMAJORCOLOR	int		253
GRIDMINORCOLOR	int		254
GRIDMODE	bool		0
GRIDSTYLE	int		0
GRIDUNIT	pt2d		1/2",1/2"
GRIDXYZTINT	int		1
GRIPBLOCK	bool		0
GRIPCOLOR	int		72
GRIPDYNCOLOR	int		140
GRIPHOT	int		240
GRIPHOVER	int		150
GRIPOBJLIMIT	int		100
GRIPS	bool		1
GRIPSIZE	int		4
GRIPTIPS	bool		1
GsDeviceType	int		0
GsDeviceType2D			0
GsDeviceType3D			1

H Variables

HALOGAP	int		0
HANDLES	bool	read-only	1
HANDSEED	str		"64"
HIDEPRECISION	int		0
HIDETEXT	int		1
HIDEXREFSCALES	bool		1
HIGHLIGHT	bool		1
HIGHLIGHTCOLOR	int		142
HIGHLIGHTEFFECT	int		0
HomeGradientColorBottom	short		210,210,210
HomeGradientColorMiddle	short		250,250,250
HomeGradientColorTop	short		White
HomeGradientMode	str		"0"
HorizonBkg_Enable	bool		1
HorizonBkg_GroundHorizon	str		"#878787"
HorizonBkg_GroundOrigin	str		"#5F5F5F"
HorizonBkg_SkyHigh	str		"#239BFF"

Variable Name	Format	R/O	Default Value
HorizonBkg_SkyHorizon	str		"#FFFFFF"
HorizonBkg_SkyLow	str		"#FAFAFF"
HPANG	real		0
HPANNOTATIVE	bool		0
HPASSOC	bool		1
HPBOUND	int		1
HPBOUNDRETAIN	short		0
HPDOUBLE	bool		0
HPDRAWORDER	int		3
HPGAPTOL	real		0
HPLAYER	str		""
HPLINETYPE	bool		0
HPNAME	str		""
HPOBJWARNING	long		10000
HPORIGIN	pt2d		0",0"
HPSCALE	real		1
HPSEPARATE	bool		0
HPSPACE	real		1
HPSTYLE	int		0
HPTRANSPARENCY	str		""
HYPERLINKBASE	str		""

I Variables

ImageCacheFolder	str		"C:\Users\userid\AppData\Local\Temp\ImageCache\"
ImageCacheMaxMemory	short		160
ImageDiskCache	bool		1
IMAGEFRAME	int		1
IMAGEHLT	bool		0
ImageNotify	bool		0
ImportCuiFileExists	short		0
IncludePlotStamp	bool		1
INDEXCTL	int		0
INETLOCATION	str		"http://www.bricsys.com"
INSBASE	pt3d		0",0",0"
INSNAME	str		""
INSUNITS	int		1
INSUNITSDEFSOURCE	int		0
INSUNITSDEFTARGET	int		0
INTERFERECOLOR	str		"BYLAYER"
INTERFEREOBJS	str		""
INTERFEREVPVS	str		""
INTERSECTIONCOLOR	int		257
INTERSECTIONDISPLAY	bool		0
ISAVEBAK	bool		1
ISAVEPERCENT	int		50
ISOLINES	int		4

L Variables

LASTANGLE	real		0
LASTPOINT	pt3d		0",0",0"
LASTPROMPT	str	read-only	": SETTINGS"
LATITUDE	real		37.795
LAYERPMODE	bool		1

Variable Name	Format	R/O	Default Value
LAYLOCKFADECTL	short		50
LAYOUTREGENCTL	int		2
LENGTHUNITS	stre		""
LENSLENGTH	real		50
LicExpDays	short		31
LICFLAGS	int		7
LICKEY	str	read-only	"7897-9999-0000-99999-0000"
LightGlyphColor	int		30
LIGHTGLYPHDISPLAY	bool		1
LIGHTINGUNITS	int		0
LightWebGlyphColor	int		1
LIMCHECK	bool		0
LIMMAX	pt2d		1',9"
LIMMIN	pt2d		0",0"
LINEARBRIGHTNESS	short		0
LINEARCONTRAST	short		0
LISPINIT	int		1
LOCALE	str		"en_US"
LocalRootFolder	str		"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V17x64\en_US\"
LOCALROOTPREFIX	str	read-only	"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V17x64\en_US\"
LOFTANG1	real		1.5707963268
LOFTANG2	real		1.5707963268
LOFTMAG1	real		0
LOFTMAG2	real		0
LOFTNORMALS	int		1
LOFTPARAM	int		7
LOGFILEMODE	bool		0
LOGFILENAME	str	read-only	""
LOGFILEPATH	str	read-only	"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V17x64\en_US\"
LOGINNAME	str	read-only	"userid"
LONGITUDE	real		-122.394
LookFromDirectionMode	short		1
LookFromFeedback	short		1
LookFromZoomExtents	bool		1
LTSCALE	real		1
LUNITS	int		4
LUPREC	int		4
LWDEFAULT	int		25
LWDISPLAY	bool		0
LWDISPSCALE	real		0.55
LWUNITS	int		1

M

MACROREC	bool		0
MACROTRACE	bool		0
MassPropAccuracy	real		0.01
MASSUNITS	str		"oz lbs stone mg g kg tonne"
MAXACTVP	int		64
MAXHATCH	int		100000
MAXSORT	int		200
MAXTHREADS	int		0
MBUTTONPAN	int		1
MEASUREINIT	int		0
MEASUREMENT	int		0

Variable Name	Format	R/O	Default Value
MENUBAR	bool		1
MENUCTL	bool		1
MENUJECHO	int		0
MENUNAME	str	read-only	"C:\Users\userid\AppData\Roaming\Bricsys\Bricscad\V17x64\en_US\Support\default.cui"
MESHTYPE	int		1
MiddleClickClose	bool		1
MILLISECS	long	read-only	436750804
MIRRTXT	bool		1
MLEADERSCALE	real		1
MODEMACRO	str		""
MSLTSCALE	short		1
MSOLESCALE	real		1
MTEXTCOLUMN	short		0
MTEXTED	str		""
MTEXTFIXED	int		2
MTFLAGS	int		0

N Variables

NAVVCUBEDISPLAY	bool		3
NAVVCUBELOCATION	int		0
NAVVCUBEOPACITY	int		50
NAVVCUBEORIENT	int		1
NavVCubeSize	short		4
NFILELIST	int		10
NOMUTT	bool		0
NORTHDIRECTION	real		0

O Variables

OBJECTISOLATIONMODE	short		0
OBSCUREDCOLOR	int		257
OBSCUREDLTYPE	int		0
OFFSETDIST	real		-1
OFFSETERASE	bool		0
OFFSETGAPTYPE	int		0
OLEFRAME	int		2
OLEHIDE	int		0
OLEQUALITY	int		0
OLESTARTUP	bool		0
OPMSTATE	int	read-only	1
ORTHOMODE	bool		0
OSMODE	int		4133
OSNAPCOORD	int		2
OSNAPZ	bool		0
OSOPTIONS	short		1

P Variables

PanBuffer	bool		1
PAPERUPDATE	bool		0
PARAMETERCOPYMODE	short		1
PdfEmbeddedTtf	bool		1
PdfExportSolidHatchType	short		2
PDFFRAME	int		1
PdfHatchToBmpDpi	short		300

Variable Name	Format	R/O	Default Value
PdfImageAntiAlias	bool		1
PdfImageCompression	short		1
PdfImageDPI	short		300
PdfLayersSetting	int		1
PdfLayoutsToExport	int		0
PdfMergeControl	long		0
PdfNotify	bool		0
PDFOSNAP	bool		1
PdfPaperHeight	int		297
PdfPaperSizeOverride	bool		0
PdfPaperWidth	int		210
PdfRenderDPI	short		300
PdfShxTextAsGeometry	bool		0
PdfSimpleGeomOptimization	bool		1
PdfTextIsSearchable	bool		0
PdfTtfTextAsGeometry	bool		0
PdfUsePlotStyles	bool		1
PdfZoomToExtentsMode	bool		1
PDMODE	int		0
PDSIZE	real		0
PEDITACCEPT	bool		0
PELLIPSE	bool		0
PERIMETER	real		0
PERSPECTIVE	bool		0
PFACEVMAX	int		4
PICKADD	bool		1
PICKAUTO	bool		3
PICKBOX	int		4
PICKDRAG	bool		0
PICKFIRST	bool		1
PICKSTYLE	int		1
PictureExportScale	real		1
_PKSER	str	read-only	""
PlacesBarFolder1	int		0
PlacesBarFolder2	int		1
PlacesBarFolder3	int		3
PlacesBarFolder4	int		5
PLATFORM	str	read-only	"Microsoft Windows NT Version 6.2"
PLINECACHE	bool		0
PLINECONVERTMODE	short		0
PLINEGEN	bool		0
PLINETYPE	int		2
PLINEWID	real		0
PlotCfgPath	str		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\PlotConfig\"
PLOTID	str		""
PlotOutputPath	str		""
PLOTROTMODE			2
PlotStylePath	str		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\PlotStyles\"
PLOTTER	int		0
PLOTTRANSPARENCYOVERRIDE	short		1
PLQUIET	bool		0
POLARADDANG	str		""
POLARANG	real		90
POLARDIST	real		0

Variable Name	Format	R/O	Default Value
POLARMODE	int		0
POLYSIDES	int		4
POPUPS	bool	read-only	1
PreviewDelay	int		30
PREVIEWEFFECT	int		2
PREVIEWFILTER	int		5
PreviewTopdown	bool		0
PREVIEWTYPE	int		0
PreviewWndInOpenDlg	bool		1
PrintFile	str		""
PRODUCT	str	read-only	"Bricscad"
PROGBAR	bool		1
PROGRAM	str	read-only	"BRICSCAD"
PROJECTIONTYPE	short		0
PROJECTNAME	str		""
ProjectSearchPaths	str		""
PROJMODE	int		1
PROMPTMENU	int		3
PromptMenuFlags	int		1
PromptOptionFormat	short		2
PromptOptionTranslateKeywords	bool		1
PROPUNITS	short		103
PropUnitsVersion	bool		1
PROXYGRAPHICS	bool		1
PROXYNOTICE	bool		1
PROXYSHOW	int		1
PROXYWEBSEARCH	int		1
PSLTSCALE	int		1
PSOLHEIGHT	real		4
PSOLWIDTH	real		0.25
PSTYLEMODE	int	read-only	1
PSTYLEPOLICY	int		1
PSVPSCALE	real		0
PUBLISHALLSHEETS	bool		1
PUCSBASE	str		""

Q

QAFLAGS	int		0
QTEXTMODE	bool		0
QuadAperture	short		20
QuadCommandLaunch	short		1
QuadCommandSort	short		0
QuadDisplay	bool		0
QuadExpandDelay	short		110
QuadExpandTabDelay	long		50
QuadExpandGroup	short		0
QuadGoTransparent	bool		0
QuadHideDelay	int		1000
QuadHideMargin	short		40
QuadIconSize	short		32
QuadIconSpace	short		1
QuadPopupCorner	short		1
QuadShowDelay	short		150
_QuadTabFlags	short		12

Variable Name	Format	R/O	Default Value
QuadToolipDelay	short		1200
QuadWarpPointer	short		4
QuadWidth	short		5

R Variables

R1ZSaveAccuracy	int		8
R1ZSaveDeviation	real		0
RASTERPREVIEW	bool		1
RE_INIT	int	read-only	0
RealTimeSpeedUp	int		5
REALWORLDSCALE	bool		1
RecentPath	str		"C:\Users\userid\Documents\"
RedHiliteFull_Edge_Alpha	int		100
RedHiliteFull_Edge_Color	str		"#007AFF"
RedHiliteFull_Edge_ShowHidden	bool		0
RedHiliteFull_Edge_Smoothing	bool		1
RedHiliteFull_Edge_Thickness	real		2
RedHiliteFull_Face_Alpha	int		10
RedHiliteFull_Face_Color	str		"#007AFF"
RedHilitePartial_SelectedEdgeGlow_Alpha	int		75
RedHilitePartial_SelectedEdgeGlow_Color	str		"#FFFFFF"
RedHilitePartial_SelectedEdgeGlow_Smoothing	bool		1
RedHilitePartial_SelectedEdgeGlow_Thickness	real		3
RedHilitePartial_SelectedEdge_Alpha	int		100
RedHilitePartial_SelectedEdge_Color	str		"#007AFF"
RedHilitePartial_SelectedEdge_ShowGlow	bool		1
RedHilitePartial_SelectedEdge_Smoothing	bool		1
RedHilitePartial_SelectedEdge_Thickness	real		2
RedHilitePartial_SelectedFace_Alpha	int		10
RedHilitePartial_SelectedFace_Color	str		"#007AFF"
RedHilitePartial_UnselectedEdge_Alpha	int		20
RedHilitePartial_UnselectedEdge_Color	str		"#007AFF"
RedHilitePartial_UnselectedEdge_ShowHidden	bool		1
RedHilitePartial_UnselectedEdge_Smoothing	bool		1
RedHilitePartial_UnselectedEdge_Thickness	real		1
RedHilite_HiddenEdge_Alpha	int		50
RedHilite_HiddenEdge_Color	str		"#FFFFFF"
RedHilite_HiddenEdge_Smoothing	bool		1
RedHilite_HiddenEdge_Thickness	real		1
RefeditLockNotInWorkset	bool		0
REFEDITNAME	str	read-only	""
REGENMODE	bool		1
REMEMBERFOLDERS	int		1
RenderMaterialPath	str		"C:\ProgramData\..."
RenderMaterialStaticPath	str		"C:\Program Files\..."
RenderUsingHardware	bool		1
RevCloudArcStyle	int		0
RevCloudMaxArcLength	real		0.375
RevCloudMinArcLength	real		0.375
RIBBONDOCKEDHEIGHT	short		120
RIBBONSTATE	bool	read-only	0
RoamableRootFolder	str	read-only	"c:\users..."
ROAMBLEROOTPREFIX	str	read-only	"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\"

Variable Name	Format	R/O	Default Value
ROLLOVEROPACITY	short		100
ROLLOVERTIPS	short		1
RTDISPLAY	int		1
RTRotationSpeedFactor	real		1
RTWalkSpeedFactor	real		1
RunAsLevel	short		2

S Variables

SaveChangeToLayout	bool		1
SAVEFIDELITY	bool		1
SAVEFILE	str	read-only	""
SAVEFILEPATH	str		"C:\Users\userid\AppData\Local\Temp\"
SaveFormat	int		1
SAVENAME	str	read-only	""
SAVEROUNDTRIP	bool		1
SAVETIME	int		60
SCREENBOXES	int	read-only	26
SCREENMODE	int	read-only	1
SCREENSIZE	pt2d	read-only	145'-8",73'-3"
SCRLHIST	int		256
SDI	int		0
SELECTIONANNODISPLAY	bool		1
SELECTIONAREA	bool		1
SELECTIONAREAOPACITY	int		25
SelectionModes	short		0
SELECTIONPREVIEW	int		3
SELECTSIMILARMODE	int		130
SHADEDGE	int		3
SHADEDIF	int		70
SheetNumberLeadingZeroes	int		1
SheetSetAutoBackup	bool		1
SheetSetTemplatePath	str		"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V17x64\en_US\Templates\Sheet Sets\"
SHORTCUTMENU	int		18
SHORTCUTMENUDURATION	long		250
ShowDocTabs	bool		1
ShowFullPathInTitle	bool		0
SHOWLAYERUSAGE	bool		0
ShowScrollButtons	bool		1
ShowTabCloseButton	bool		0
ShowTabCloseButtonActive	bool		0
ShowTabCloseButtonAll	bool		1
ShowTabControls	bool		1
ShowWindowListButton	bool		1
SHPNAME	str		""
SingletonMode	bool		0
SKETCHINC	real		0.1
SKPOLY	bool		0
SKYSTATUS	int		0
SMTARGETCAM	str		""
SNAPANG	real		0
SNAPBASE	pt2d		0",0"
SNAPISOPAIR	int		0
SnapMarkerColor	int		20

Variable Name	Format	R/O	Default Value
SnapMarkerSize	int		6
SnapMarkerThickness	int		2
SNAPMODE	bool		0
SNAPSTYL	int		0
SNAPTYPE	int		0
SNAPUNIT	pt2d		1/2",1/2"
SOLIDCHECK	bool		1
SORTENTS	int		127
spaAdjustMode	int		0
spaGridAspectRatio	real		0
spaGridMode	int		1
spaMaxFacetEdgeLength	real		0
spaMaxNumGridLines	long		512
spaMinUGridLines	long		0
spaMinVGridLines	long		0
spaNormalTol	real		15
spaSurfaceTol	real		-1
spaTriangMode	int		1
spaUseFacetRES	bool		1
SPLFRAME	bool		0
SPLINESEGS	int		8
SPLINETYPE	int		6
SRCHPATH	str		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\Support\; C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Support\; C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Fonts\; C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Help\en_US\"
SSFOUND	str		""
SSLOCATE	bool		1
SSMAUTOOPEN	bool		1
SSMPOLLTIME	short		15
SSMSHEETSTATUS	short		2
SSMSTATE	int		0
StampFontSize	real		0.2
StampFontStyle	str		"Arial"
StampFooter	str		""
StampHeader	str		""
StampUnits	int		0
STARTUP	int		1
STEPSIZE	real		6
STEPSPERSEC	real		2
StructureTreeConfig	str		"mechanical.cst"
SURFTAB1	int		6
SURFTAB2	int		6
SURFTYPE	int		6
SURFU	int		6
SURFV	int		6
SvgBlendedGradients	int		0
SvgDefaultImageExtension	str		".png"
SvgGenericFontFamily	int		0
SvgHiddenLineRemoving	int		0
SvgImageBase	str		""
SvgImageUrl	str		""
SvgLineWeightScale	real		1
SvgOutputHeight	int		768

Variable Name	Format	R/O	Default Value
SvgOutputWidth	int		1024
SvgPrecision	int		6
SYSCODEPAGE	str	read-only	"ANSI_1252"

T Variables

TabControlHeight	int		25
TABMODE	bool		0
TabsFixedWidth	bool		0
TARGET	pt3d		0",0",0"
TDCREATE	real	read-only	2456335.5399919
TDINDWG	real	read-only	0.121354456
TDUCREATE	real	read-only	2456335.8733252
TDUUPDATE	real	read-only	2456335.5399919
TDUSRTIMER	real	read-only	0.121354456
TDUUPDATE	real	read-only	2456335.8733252
TemplatePath	str		"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V17x64\en_US\Templates\"
TEMPPREFIX	str		""
TestFlags			0
TEXTANGLE	real		0
TEXTEVAL	int		0
TEXTFILL	int		1
TEXTQLTY	int		50
TEXTSIZE	real		0.2
TEXTSTYLE	str		"Standard"
TextureMapPath	str		"C:\Program Files (x86)\Bricsys\BricsCAD V17x64\Textures\1\"
THICKNESS	real		0
THUMBSIZE	short		1
TILEMODE	int		1
TILEMODELIGHTSYNCH	bool		1
TIMEZONE	int		-8000
Tips	short		1
ToolbarIconSize	short		16
TOOLPALETTEPATH	str		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V17x64\en_US\Support\ToolPalettes\"
TOOLTIPS			1
TPSTATE	int	read-only	0
TRACEWID	real		0.05
TRACKPATH	int		0
TRANSPARENCYDISPLAY	bool		1
TREEDEPTH	int		3020
TREEMAX	long		10000000
TRIMMODE	bool		1
TSPACEFAC	real		1
TSPACETYPE	int		1
TSTACKALIGN	int		1
TSTACKSIZE	int		70
TTFTEXT	int		3

U Variables

UCSAXISANG	int		90
UCSBASE	str		""
UCSDETECT	bool		0
UCSFOLLOW	bool		0
UCSICON	int		3

UCSICONPOS	int		0
UCSNAME	str	read-only	""
UCSORG	pt3d	read-only	0",0",0"
UCSORTHO	bool		1
UCSVIEW	bool		1
UCSVP	bool		1
UCSXDIR	pt3d	read-only	1",0",0"
UCSYDIR	pt3d	read-only	0",1",0"
UNDOCTL	int	read-only	5
UNDOMARKS	int	read-only	0
UNITMODE	bool		0
USER1 <i>thru</i> USER15	int		0
USERR1 <i>thru</i> USERR5	real		0
USERS1 <i>thru</i> USERS5	str		""
UseSheetMetal	short		2
UseStandardOpenFileDialog	bool		0

V Variables

VbaMacros	bool		1
VENDORNAME	str	read-only	"Bricsys"
_VERNUM	str	read-only	"16.2.05 (UNICODE)"
VersionCustomizableFiles	str		"243"
VIEWCTR	pt3d	read-only	10 7/16",4 1/2",0"
VIEWDIR	pt3d	read-only	0",0",1"
VIEWMODE	int	read-only	0
VIEWSIZE	real	read-only	297
VIEWTWIST	real	read-only	0
VIEWUPDATEAUTO	short		1
VISRETAIN	int		1
VOLUMEPREC	short		-1
VOLUMEUNITS	str		"in ft mi µm mm cm m km"
VPROTATEASSOC	bool		1
VSMAX	pt3d	read-only	-1.0000E+20,-1.0000E+20,-1.0000E+20
VSMIN	pt3d	read-only	1.0000E+20,1.0000E+20,1.0000E+20

W Variables

WarningMessages	int		65535
WHIPARC	int		1
WHIPTHREAD	int		0
WINDOWAREACOLOR	int		150
WIPEOUTFRAME	short		1
WMFBKGND	bool		0
WMFFOREGND	bool		0
WNDLMAIN	int		2
WNDLSCRL	bool		0
WNDLSTAT	bool		1
WNDLTABS	bool		1
WNDLTEXT	int		1
WNDPMAIN	pt2d		0",0"
WNDPTEXT	pt2d		3'-4",3'-4"
WNDMAIN	pt2d		101'-2",66'-11"
WNDSTEXT	pt2d		118'-4",86'
WorkspaceSecurity	bool		1
WORLDUCS	bool		1

Variable Name	Format	R/O	Default Value
WORLDVIEW	int		1
WRITESTAT	bool	read-only	1
WSAUTOSAVE	bool		1
WSCURRENT	str		"2D Drafting"

X Variables

XCLIPFRAME	int		2
XDwgFadeCtl	short		70
XEDIT	bool		1
XFADECTL	int		50
XLOADCTL	int		1
XLOADPATH	str		"C:\Users\userid\Documents\"
XNotifyTime	short		5
XREFCTL	bool		0
XRefNotify	bool		1
XREFOVERRIDE	bool		0

Z Variables

ZOOMFACTOR	int		60
ZOOMWHEEL	short		0

Variables

3DCOMPAREMODE	short		3
3DOSMODE	short		11
3dSnapMarkerColor	short		5

Concise Summary of Command Names

THIS APPENDIX LISTS THE NAMES OF COMMANDS FOUND IN BRICSCAD V17. THE LIST OF 700 commands is sorted alphabetically by name, as well as in groupings of common commands as follows:

AI Commands	page 318
BIM Commands	page 319
BM (BricsCAD Mechanical) Commands	page 320
Chapoo Commands	page 322
DIM (Dimension) Commands	page 323
DC (Dimensional Constraint) Commands	page 324
DM (Direct Modeling) Commands	page 324
GC (Geometric Constraint) Commands	page 326
Layer Commands	page 328
SM (Sheet Metal) Commands	page 334
VBA Commands	page 336
ViewBase Commands	page 336

“Bim-” BIM and “Sm-” sheet metal commands are available through optional extra-cost add-on modules.

When a command has a hyphen prefix, such as -Color, the command runs at the command prompt.

Command names added since the initial V16 edition of this ebook are shown in **blue**. Command names specific to the demo, Pro, and Platinum versions of BricsCAD are shown in **boldface**; these commands are not available in the Standard version. Commands specific to Platinum version are noted as (PLATINUM ONLY) and those specific to Windows as (WINDOWS ONLY).

A Commands

About displays information about the program.

AcisIn imports 3D solids in SAT format (**SAT** is short for “save as text”).

AcisOut exports 3D solids and surface entities in SAT format.

AddInMan displays the VBA COM Add-In Manager dialog box (WINDOWS ONLY).

AddSelected creates a new entity of the same type as an existing entity.

Align aligns entities with other entities in 2D and 3D space.

AlignSpace adjusts viewport angle, zoom factor, and pan position based on alignment points specified in model space and paper space; operates in paper space only.

AniPath makes movies from views generated by a camera moving through 3D scenes.

AnnReset resets all scale representations to the entity’s original positions

AnnUpdate updates annotative scale factors to match updates made with Style and DimStyle commands.

Aperture sets selection area for snapping to entities.

Apparent toggles Apparent intersection entity snap; snaps to the intersections of entities, even when they only appear to intersect in 3D space.

AppLoad loads DRX, LISP, and SDS applications to run inside BricsCAD; Mac and Linux load only LISP and SDS.

Arc draws arcs.

Area determines the area and perimeter of closed 2D objects; the area and length of open polylines and splines as if they were closed; the lengths only of lines, sketches, arcs, and elliptical arcs; and the areas of faces of 3D objects.

Array and **-Array** creates dynamic polar, path, and rectangular arrays of entities.

ArrayClassic runs the dialog box-based version of the Array command.

ArrayClose and **-ArrayClose** end the array editing session.

ArrayEdit edits entities and source entities of arrays.

ArrayEditExt edits entities in arrays.

ArrayPath distributes entity copies evenly along a path into multiple rows and levels.

ArrayPolar distributes entity copies evenly in a circular pattern about a center point or axis of rotation, using multiple rows and levels.

ArrayRect distributes entity copies into any number of rows, columns, and levels.

AttDef and **-AttDef** defines attributes for blocks.

AttDisp toggles the display of attributes through all, none, or those normally visible.

AttEdit edits the values and properties of attributes.

AttExt and **-AttExt** exports data from attributes to text files.

AttRedef redefines blocks and updates associated attributes.

AttSync synchronizes attribute definitions in all references to a specified block definition.

Audit repairs open drawings in case of data corruption.

AutoComplete sets the options for autocomplete mode on the command line.

Ai Commands

Ai_Box draws 3D boxes as mesh surfaces.

Ai_CircTan draws a circle tangent to three entities.

Ai_Cone draws 3D cones as mesh surfaces.

Ai_Cylinder draws 3D cylinders as mesh surfaces.

Ai_DeSelect unselects all selected entities.

Ai_Dish draws 3D dishes as mesh surfaces.

Ai_Dome draws 3D domes (half-spheres) as mesh surfaces.

Ai_DrawOrder changes the display order of overlapping entities.

Ai_Fms switches to the first layout tab and enters model space of the first viewport.

Ai_Molc makes the layer current of the selected entity (short for “make object layer current”).

Ai_MSpace switches to model tab.

Ai_PSpace switches to the first layout tab.

Ai_Pyramid draws 3D pyramids as mesh surfaces.

Ai_SelAll selects all non-frozen entities in the current space, like Ctrl+A.

Ai_Sphere draws 3D spheres as mesh surfaces.

Ai_TileModel sets TileMode variable to 1 and then switches to model tab.

Ai_Torus draws 3D tori as mesh surfaces.

Ai_Wedge draws 3D wedges as mesh surfaces.

AiMleaderEditAdd adds leader lines to multi-leaders.

AiMleaderEditRemove removes leader lines from multi-leaders.

B Commands

Base changes the drawing’s insertion point when it is inserted into other drawings.

BAttMan manages the attributes of block definitions (short for Block Attribute Manager).

BHatch and **-BHatch** fills closed areas with repeating patterns, solid colors, or gradients.

BlipMode enables and disables display of marker blips.

Block and **-Block** groups entities into blocks (symbols).

BmpOut exports the current viewport as a BMP (bitmap) file.

Boundary and **-Boundary** draws a polyline that forms a boundary around the inside closed areas.

Box draws three-dimensional solid boxes.

Break removes portions of entities.

Browser opens the default Web browser.

BIM Commands

(Available for Platinum edition only; bim = building information modeling)

bimAttachComposition attaches BIM compositions to solids.

bimAttachSpatialLocation locates the drawing in mapping references.

bimCheck checks the validity of the BIM model.

bimClassify classifies an entity as a building element with a name and an internal ‘guid’ (globally unique identifier).

bimConnect creates L-connections between faces of two solids.

bimDrag drags faces of solids; when dragging major faces, it preserves connections with minor faces; when dragging minor faces,

it optionally connects minor faces to major faces of other solids.

bimExport exports the current BIM model to an IFC file.

bimFlip flips the starting face from which the layers of a composition are set out.

bimGetStatisticalData reports statistics data of BIM objects in the current drawing.

bimIfcImport imports IFC files; IFC is short for “industry foundation classes.”

bimInsert and **-bimInsert** insert windows and doors in solids.

bimList list names and properties of BIM entities in the current drawing.

bimPatch reserves an of a BIM model for editing with the RefEdit command.

bimReposition repositions inserts in face of solids.

bimRoom defines room areas with markers.

bimSection creates BIM section entities.

bimSectionOpen opens the drawing file related to a BIM section entity; or the 3D BIM model related to a BIM section drawing.

bimSectionUpdate updates and exports BIM sections.

bimSkpImport imports SKP files with optional stitching; SKP is short for SketchUp.

bimSplit splits segmented solids into separated solids automatically; splits solids using cutting faces.

bimUpdateRoom updates data about the selected room.

bimUpdateThickness re-applies the overall thickness of a composition to the solid.

bimWindowPrint prints a specified area of the BIM model.

bimWindowUpdate updates openings made by windows or doors in solids in case the opening did not updated correctly automatically.

ClipDisplay toggles the clipped display property of a section plane or a BIM section entity.

BricsCAD Mechanical Commands

(Available in Platinum edition only; bm = BricsCAD mechanical)

bmBom inserts bill of material (BOM) tables in the current drawing.

bmBrowser toggles the visibility of the Mechanical Browser window.

bmDependencies lists all files, containing component definitions inserted in the assembly, in the command window.

bmDissolve dissolves a mechanical component inserted in the current drawing.

bmExternalize converts local components to external components.

bmForm creates a new mechanical component and inserts it into the current drawing; if necessary, run **bmMech** to initialize the mechanical structure in the current drawing.

bmHardware and **-bmHardware** insert standard hardware parts as a mechanical component in the current drawing.

bmHide hides the visibility of mechanical components; hidden inserts are taken into account by commands such as **bmBom** and **bmMassProp**.

bmInsert and **-bmInsert** insert an existing mechanical component as a virtual component into the current drawing.

bmLocalize converts external components to local components.

bmMassProp computes mass properties for the current model using densities assigned to the components (defined by the Density property of the components and subcomponents).

bmMech converts the current drawing into a mechanical component.

bmNew creates a mechanical component as a new drawing file.

bmOpen opens the source drawing of external mechanical components.

bmOpenCopy opens a copy of a component insert as a new drawing.

-bmParameters lists and edits parameters of inserted components.

bmRecover recovers broken mechanical structures.

bmReplace replaces a component insert.

bmShow shows previously hidden mechanical components.

bmUnmech converts the current mechanical component into a plain drawing.

bmUpdate reloads all referenced components from external files and updates BOM tables.

bmVStyle applies visual styles to mechanical component inserts.

bmXConvert converts X-Hardware solids in the current drawing to mechanical components.

C Commands

Cal displays the operating system's Calculator program.

Callout places callouts; can be used only from the SheetSet panel.

Camera changes the viewpoint to perspective.

Center toggles Center entity snap; snaps to the center of circles, arcs, and other circular entities.

Chamfer bevels entities.

Change changes the position and properties of entities: endpoint, color, elevation, layer, linetype, linetype scale, lineweight, and thickness.

ChProp changes just the properties of entities.

ChSpace moves entities from paper space to model space and vice versa.

Circle draws circles.

CleanUnusedVariables clears unused variables from memory.

Close exits the current drawing, but not the program.

Color and **-Color** specifies the color for entities.

CommandLine and **CommandLineHide** open and close the command bar.

Commands reports the names of all commands supported by the program.

Cone draws three-dimensional solid cones.

ContentBrowserClose and **ContentBrowserOpen** close and open the Content Browser panel.

ConvertCtb converts older CBT (color-based plot tables) files to newer STB (style-based plot tables) files.

ConvertOldLights converts old light definitions to the current format.

ConvertOldMaterials converts old material definitions to the current format.

ConvertPoly converts lightweight polylines to classic polylines (2D polylines) and vice versa.

ConvertPStyles converts drawings to from CTB (color-based plotting) to STB (plot styles).

Copy duplicates entities.

CopyBase copies entities with a specified reference point to the Clipboard.

CopyClip copies entities to the Clipboard.

CopyEData Copies extended entity data from one entity to others.

CopyHist copies the command history to the Clipboard.

CPageSetup edits the page setup of the current layout or model space.

CuiLoad and **CuiUnload** load and unload CUI and CUIX (user interface customization), MNU (menu), MNS (LISP code), and ICM (IntelliCAD menu) files.

Customize customizes user interface elements, such as menus, toolbars, and shortcuts.

CutClip copies entities to the Clipboard and deletes the entities.

Cylinder draws three-dimensional solid cylinders.

Chapoo Commands

ChapooAccount reports the status of the Chapoo account at the command bar.

ChapooDownload downloads drawings from the Chapoo project to a local folder.

ChapooLogoff logs off from the Chapoo project.

ChapooLogon logs on to Chapoo.

ChapooOpen opens a drawing after downloading it from Chapoo.

ChapooProject opens the Chapoo project in the default browser.

ChapooUpload uploads the current drawing to Chapoo.

ChapooWeb connects to the Chapoo website.

D Commands

DataExtraction exports entity properties, block attributes and drawing information to CSV (comma separated values) file.

DbList lists information about all entities in the drawing (short for “database listing”).

DdAttE edits the values of attributes through a dialog box (short for “dynamic dialog attribute editor”).

DdEdit edits single-line text, multi-line text, attribute definitions, and attribute text (short for “dynamic dialog editor”).

DdEModes sets default values for creating entities (short for “dynamic dialog entity modes”).

DdFilter creates a selection set of the entities selected.

DdGrips specifies the properties of grips through the Settings dialog box.

DdPType specifies the look and size of point entities, through the Settings dialog box (short for “dynamic dialog point type”).

DdSelect specifies the properties for selecting entities, through the Settings dialog box.

DdSetVar displays the Settings dialog box to change the values of variables.

DdSTrack Sets the properties for snap tracking, through the Settings dialog box (short for “snap tracking”).

DdVPoint sets 3D viewpoints or plan view

Delay delays execution of the next command; for use with scripts only.

DeIEData deletes extended entity data from the selected entity (short for “delete entity data”).

Dish draws dishes (bottom half-sphere) from polygon meshes.

Dist reports the distance and angle between two points.

Distantlight places distant lights.

Divide places points or blocks along entities.

Dome draws domes (top half-sphere) from polygon meshes.

Donut draws circular polylines with width.

DragMode controls the appearance of objects while being dragged.

DrawOrder changes the display order of overlapping entities.

DrawOrderByLayer controls the draw order of overlapping objects through layer names.

DSettings displays the Settings dialog box for drafting settings (short for “drafting settings”).

DView changes the 3D viewpoint interactively, and turns on perspective mode (short for “dynamic view”).

DwgCodePage changes the code page for text in drawings.

DwgProps opens the Drawing Properties dialog box, showing the general information and user defined properties stored with a drawing.

Dxfln and **DxfOut** imports DXF files (short for “drawing exchange format”) and exports drawings in ASCII or binary DXF format.

Dimension Commands

(Dim = dimension)

Ai_Dim_TextAbove moves text above the dimension line.

Ai_Dim_TextCenter centers text on the dimension line.

Ai_Dim_TextHome moves text to its home position, as defined by the dimension style.

AiDimFlipArrow mirrors arrowheads on dimension lines.

AiDimPrec changes the precision of dimension text.

Dim places and edits dimensions at the ‘Dimensioning command:’ prompt.

DimI executes a single dimension command at the ‘Dimensioning command:’ prompt.

DimAligned draws dimensions parallel to (aligned with) selected entities; works with lines, polylines, arcs, and circles.

DimAngular dimensions angles.

DimArc places arc length dimensions.

DimBaseline places multiple linear or angular dimensions starting at the same base point; command can only be used when at least one other dimension is already in the drawing.

DimCenter places center marks at the center points of circles and arcs.

DimContinue continues linear and angular dimensions from the endpoint of the previous dimension.

DimDiameter dimensions the diameter of circles and arcs, and places a center mark.

DimDisassociate removes associativity from selected dimension entities.

DimEdit changes wording and angle of dimension text; changes the angle of extension lines.

DimLeader draws leaders.

DimLinear places linear dimensions horizontally, vertically, or rotated.

DimOrdinate measures x and y ordinate distances from a common origin, specified by the current UCS origin.

DimOverride overrides the values of the current dimension style.

DimRadius dimensions the radii of arcs and circles.

DimReassociate reassociates or associates dimensions to entities or points on entities.

DimRegen updates associative dimensions (short for “dimension regeneration”).

DimStyle and **-DimStyle** creates and modifies dimension styles through the Drawing Explorer.

DimStyleSet reports the current dimension style in the command bar.

DimTEdit changes the position of dimension text.

Dimensional Constraint Commands

(*dc = dimensional constraint*)

CleanUnusedVariables purges variables not used by constraint expressions and not linked to dimensions.

dcAligned constrains the distance between two defining points on entities.

dcAngular constrains the angle between three constraint points on entities; or between two lines; or between two polyline segments; or constrains the angles of arcs or polyline arcs.

dcConvert converts an associative dimension to a dimensional constraint.

dcDiameter constrains the diameters of circles, arc, or polyline arcs.

dcDisplay shows and hides dimensional constraints.

dcHorizontal constrains the horizontal distance between two defining points on entities.

dcLinear constrains horizontal or vertical distance between two defining points on entities.

dcRadial constrains the radius of circles, arcs, or polyline arcs.

dcVertical constrains the vertical distance between two defining points on entities.

DelConstraint removes all dimensional (and geometrical) constraints from an entity.

DimConstraint applies a dimensional constraint to an entity or between constraint points on entities; converts associative dimensions to dynamic dimensions.

Direct Modeling Commands

(*Available for Pro or Platinum editions only; dm = direct modeling*)

dmAngle3D applies angle constraints between the faces of a solid or of different solids.

dmAudit checks and fixes 3D models.

dmChamfer creates an equal distance chamfer between adjacent faces.

dmCoincident3D applies coincident constraints between two edges, two faces, or an edge and a face of two different solids (PLATINUM ONLY).

dmConcentric3D applies concentric constraints between two cylindrical, spherical, or conical surfaces (PLATINUM ONLY).

dmConstraint3D applies geometric relationships and dimensional constraints between sub-entities (such as faces, surfaces, and edges) of 3D entities (PLATINUM ONLY).

dmDeformCurve deforms one or more connected faces of a 3D solid/surface by replacing their edges with given curves (PLATINUM ONLY).

dmDeformMove deforms one or more connected faces of a 3D solid/surface by moving and rotating their edges (PLATINUM ONLY).

dmDeformPoint deforms as smoothly as possible (using G1 or G2 continuity) a region, one or more connected faces of a 3D solid or a surface by moving a point lying on one of them in arbitrary 3D direction. (PLATINUM ONLY)

dmDelete deletes faces and solids.

dmDistance3D applies a distance constraint between two sub-entities of a solid or of different solids (PLATINUM ONLY).

dmExtrude creates 3D solids by extruding closed 2D entities, regions or closed boundaries.

dmFillet creates a smooth fillet between adjacent faces sharing a sharp edge.

dmFix3D applies a fixed constraint to a solid or to an edge or a face of a solid (PLATINUM ONLY).

dmGroup creates new groups, edits them, and dissolves groups.

dmMove moves the selected solids, or faces or edges of a solid using a vector.

dmParallel3D applies a parallel constraint between two faces of a solid or of different solids (PLATINUM ONLY).

dmPerpendicular3D applies a perpendicular constraint between two faces of a solid or of different solids (PLATINUM ONLY).

dmPushPull adds or removes volume from a solid by moving a face.

dmRadius3D applies a radius constraint to cylindrical surfaces or circular edges (PLATINUM ONLY).

dmRepair fixes inconsistencies in 3D geometry supported by ACIS kernel (3D solids, surfaces).

dmRevolve creates 3D solids by revolution of closed 2D entities or regions about an axis.

dmRigidSet3D defines a set of entities or sub-entities as a rigid body (PLATINUM ONLY).

dmRotate rotates faces of a solid around an axis.

dmSelect selects edges and faces of 3D solids or surfaces based on their geometric properties.

dmSelectEdges selects faces and edges of 3D solids.

dmSimplify simplifies the geometry and topology of 3D solid entities by removing unnecessary edges and vertices, merges seam edges, and replaces the geometry of faces and edges by analytic surfaces and curves, if possible within the user-specified tolerance. Run this command on imported 3D solid geometry.

dmStitch converts a set of region and surface entities that bound a watertight area to a 3D solid.

dmTangent3D applies a tangent constraint between a face and a curved surface of different solids (PLATINUM ONLY).

dmThicken creates 3D solids by thickening (i.e. adding thickness to) surfaces, their faces, and faces of 3D solids.

dmTwist twists 3D solids by an angle.

dmUpdate forces 3D constraints to update (PLATINUM ONLY).

E Commands

EAttEdit edits the value and most properties of attributes (short for “enhanced attribute editor”).

EdgeSurf creates a 3D Coons mesh surface patch between four lines, forming a closed shape (short for “edge surface”).

EditEData creates and edits extended entity data (short for “edit entity data”).

Elev changes the default elevation and thickness.

Ellipse draws ellipses and elliptical arcs.

Endpoint toggles endpoint entity snap; snaps to the ends of open entities, such as line, arcs, and open polylines.

Erase erases selected entities from drawings; alternatively, press the Del key.

eTransmit creates a package of a drawing file and all its dependencies, such as external references, images, font files, plot configuration files, plot style tables and font map files.

ExpBlocks opens the Blocks section of the Drawing Explorer dialog box (short for “explorer blocks”).

ExpFolders opens the Drawing Explorer on the Folders tab.

Explode breaks complex objects into their component entities.

Explorer opens the Drawing Explorer dialog box, which controls Layers, Layer States, Linetypes, Multiline Styles, Multileader Styles, Text Styles, Dimension Styles, Table Styles, Coordinate Systems, Views, Visual Styles, Lights, Materials, Render Presets, Blocks, External References, Images, PDF Underlays, Dependencies, Page Setups, and Section Planes.

Export saves entities in other file formats.

ExportLayout exports visible objects from the current layout to model space of new drawings.

ExportPDF exports the current layout to a PDF file.

ExpUcs creates, modifies, and deletes named UCSes through the Drawing Explorer (short for “explore user-defined coordinate systems”).

Extend extends entities to bounding edges defined by other entities.

Extension toggles extension entity snap, which snaps to the point where a line extended would intersect another entity.

Extrude extrudes closed entities as 3D solids and open ones as 3D surfaces.

F

Field inserts text that is updated automatically when system variables change.

FileOpen opens drawing (DWG), template (DWT), and interchange (DXF) files from the command line.

Files opens the operating system's file manager, such as Windows Explorer or Finder.

Fill fills areas with a solid color or color gradient

Fillet rounds entities.

Find finds and replaces text in notes, annotations, and dimension text.

Flatshot creates a hidden line representation of all 3D solids in model space as a block or a new drawing.

Flatten flattens 2D objects with thickness and allows to convert splines to polylines.

G Commands

GCE snaps to the geometric center of entities.

GenerateBoundary creates closed polylines from faces of 3D solids, as well as from boundaries detected when the Enable Boundary Detection of SelectionModes is activated.

GeographicLocation sets the geographic location of the drawing.

Gradient fills closed areas with gradient fills of one or two colors.

GradientBkgOff and **GradientBkgOn** turn off and on the gradient displayed in the working area.

GraphScr switches from the text windows to the graphics windows (short for "graphics screen").

Grid turns the grid display on or off and sets other grid options.

Group and **-Group** creates and modifies named groups of entities.

Geometric Constraint Commands

(For 3D constraints, see Direct Modeling Commands section; gc = geometric constraints)

ConstraintBar shows, hides, and resets the display of geometric constraint icons.

DelConstraint removes all geometrical (and dimensional) constraints from an entity.

gcCoincident constrains points on entities coincidentally; or constrains a point on an entity to another entity.

gcCollinear constrains lines collinearly.

gcConcentric constrains the center points of arcs, circles, ellipses, and/or elliptical arcs to be coincident.

gcEqual constrains lines to have the same length, or arcs and circles to have the same radius.

gcFix constrains points on entities to fixed positions.

gcHorizontal constrains lines or linear polyline segments, or pairs of points on entities to be parallel to the x axis in the current coordinate system.

gcParallel constrains two lines or linear polyline segments to be parallel to each other.

gcPerpendicular constrains two lines or linear polyline segments to be perpendicular to each other.

gcSmooth constrains a spline to be fluidly continuous to another spline, or arc, or line, or polyline.

gcSymmetric constrains two entities, or two points on entities, to be symmetric about a line of symmetry.

gcTangent constrains one entity tangent to another.

gcVertical constrains lines or linear polyline segments, or pairs of points on entities to be parallel to the y axis in the current coordinate system.

GeomConstraint acts as a universal command that applies all available geometric constraint points.

H Commands

Hatch and **-Hatch** fills a selected boundary with a pattern.

HatchEdit and **-HatchEdit** edits hatch patterns and gradient fills.

HatchGenerateBoundary generates a boundary around a hatch or gradient fill.

HatchToBack sets the draw order of all hatch entities in the drawing to display behind all other entities.

Helix draws 2D spirals or 3D helices.

Help displays online help.

HelpSearch prompts for searching through the help files at the command prompt.

Hide removes hidden lines from 3D entities until the **UnisolateObjects** command is used.

HideObjects temporarily hides selected entities.

Hyperlink and **-Hyperlink** adds hyperlinks to entities or modifies existing hyperlinks.

HyperlinkOptions controls the display of the hyperlink cursor, shortcut menu, and tooltips.

I Commands

Id reports the x,y,z coordinates of a picked point.

Image inserts raster images in drawings through the Drawing Explorer.

ImageAdjust adjusts the properties of images through the Properties palette.

ImageAttach and **-ImageAttach** attaches raster images to the drawing like xrefs.

ImageClip clips images.

ImageFrame toggles the frame around images.

ImageQuality determines the display quality of images attached to the drawing.

Import displays a dialog box for importing files into the drawing: DWG, DXF, DWT, and DAE (Collada) files. Platinum edition also imports IFC and SKP (SketchUp) files. Additional formats can be imported when the optional Communicator modules is purchased.

Imprint imprints 2D entities onto planar faces of 3D solids and surfaces; allows to create additional edges on planar faces.

Insert and **-Insert** inserts blocks or another drawing into the current drawing.

InsertAligned inserts blocks repeatedly, and inserts mirrored blocks.

Insertion toggles Insertion entity snap; snaps to the insertion point of text and blocks.

InsertObj displays data from other programs in drawings, such as text documents, spreadsheets, and images (windows only).

Interfere checks interferences between solid models.

Intersect creates regions or 3D solids from the intersection of regions or 3D solids.

Intersection Toggles Intersection entity snap; snaps to the intersections of entities.

IsolateObjects hides all other entities from view.

Isoplane controls the isometric plane (left, right, or top) when isometric snap is used.

J Command

Join joins lines, lwpolylines, 2D polylines, 3D polylines, circular arcs, elliptical arcs, splines and helices at common endpoints.

L Commands

Layer: see Layer Commands below.

Layout creates, copies, renames, and deletes layouts.

Leader draws leader lines that connect annotations to drawing entities.

Lengthen changes the length of open objects, such as lines and arcs.

LicenseManager provides access to all Bricsys software licenses, as shown below.

LicEnterKey enters the license key number (short for “licence enter key”).

LicProperties reports the BricsCAD license information; modifies and deactivates single user and volume license keys.

LicPropertiesCommunicator reports license information for the optional extra-cost Communicator add-on.

Light places lights in drawings.

LightList displays the lighting palette.

Limits sets the extents of the drawing and the grid.

Line draws straight line segments.

LineType and **-LineType** creates, loads, and sets linetypes.

List lists the properties of selected entities at the command line.

LiveSection toggles the **Live Section** property of a section plane.

Load loads compiled SHX shape files into the drawing.

Loft creates 3D solids passing through two or more cross sections.

LogFileOff and **LogFileOn** turn off and on log file recording.

LWeight sets lineweight options.

Layer Commands

LayCur moves the selected entities to the current layer.

Layer and **-Layer** controls layers and layer properties.

LayerP undoes previously applied changes to layer settings when LayerPMode is on (short for “layer previous”).

LayerPMode controls the tracking of changes made to layer settings.

LayersPanelClose and **LayersPanelOpen** closes and open the Layers panel.

LayerState saves and restores the properties of layers.

LayFrz and **LayThw** freeze and thaw the layers associated with entities selected in the drawing.

LayIso and **LayUnIso** isolate and restore layers associated with entities selected in the drawing; locks or turns off all other layers (short for “layer isolate”).

LayLck and **LayUlk** lock and unlock the layers of selected entities.

LayMCur changes the working layer to that of a selected entity (short for “layer make current”).

LayOff and **LayOn** turn off and on layers associated with entities selected in the drawing; off layers cannot be seen.

M Commands

Mail attaches the current drawing to a new message with your computer’s default email client.

MapConnect sets up a connection with a Web Map Service, after the GeographicLocation command defines the geographic location in the drawing.

MassProp reports the area, perimeter, and other mathematical properties of 3D solids and 2D regions (short for “mass properties”).

MatBrowserClose and **MatBrowserOpen** close and open the materials browser.

MatchPerspective changes the viewpoint in perspective mode to match a background image.

MatchProp assigns the properties of one entity to one or more other entities (short for “match properties”).

MaterialMap maps material definitions onto the surfaces of objects, with presets for boxes, planes, spheres, and cylinders.

Materials creates materials and edits their properties through the Drawing Explorer.

MatLib displays the Rendering Materials panel.

Measure places points or blocks along entities.

Menu loads menu files to modify the user interface.

MenuLoad and **MenuUnload** load and unload CUIX and CUI (user interface customization), MNU (menu), MNS (LISP code), and ICM (IntelliCAD menu) files.

Midpoint toggles Midpoint entity snap; snaps to the middle of lines, arcs, and other open entities.

MInsert inserts a block as a rectangular array; combines the -Insert and Array commands (short for “multiple insertion”).

Mirror draws mirror image copies of entities.

Mirror3D draws mirror images of entities about a plane in 3D space.

MLeader creates multileader entities using the current multileader style.

MLeaderEdit adds leader lines to and removes leader lines from a multileader entity.

MLeaderEditExt adds and removes leader lines, adds and removes vertices from a multileader entity.

MLeaderStyle creates and manages multileader styles through the Drawing Explorer.

MLine draws multilines.

MLStyle creates and edits multiline styles.

ModelerProperties and **-ModelerProperties** controls the various settings of the ACIS modeler through the Settings dialog box.

Move displaces entities a specified distance in a specified direction.

MoveEData moves extended entity data from one entity to another.

MSlide makes SLD (slide) files from the current view.

MSpace switches to model space inside a viewport of layout tab.

MText and **-MText** opens the multi-line text editor interface for placing paragraph text.

Multiple command prefix forces commands to repeat themselves automatically.

MView creates viewports in layout tab.

MvSetup prepares sets of paper space viewports; superseded by the ViewBase command.

MTP snaps to the midpoint between two points.

N Commands

Nearest toggles Nearest entity snap mode; snaps to the nearest geometry on entities.

NetLoad loads .NET applications.

New starts new drawing files.

NewSheetSet creates a new sheet set.

NewWiz starts new drawings with the New Drawing Wizard.

Node toggles Node entity snap mode; snaps to point entities.

None turns off all entity snap modes.

O Commands

ObjectScale and **-ObjectScale** adds or removes supported scales for annotative entities.

Offset offsets linear entities in parallel orientation.

OleLinks adjusts links of OLE entities embedded in or linked to drawings (short for “object linking and embedding”). (WINDOWS ONLY).

OleOpen opens OLE objects for modification (WINDOWS ONLY).

OnWeb opens the Bricsys home page in your computer’s default Web browser.

Oops un-erases the last erased entity, including those erased by the Block command.

Open opens an existing drawing file.

OpenSheetSet and **-OpenSheetSet** open an existing sheet set.

Options configures program operating parameters.

Orthogonal constrains the pointer so it moves parallel to the axes of the current coordinate system.

OSnap and **-OSnap** sets entity snaps through the Settings dialog box or the command line (short for “object snap”).

Overkill and **-Overkill** deletes duplicate entities and overlapping lines, arcs or polylines and unifies partly overlapping or contiguous ones.

P Commands

PageSetup creates and edits page setups for plotting drawings in the Drawing Explorer.

Pan and **-Pan** moves the drawing display in the active view tile.

Parallel turns on parallel entity snap.

-Parameters create and edit constraint expressions and values.

PasteBlock inserts data from the Clipboard as block.

PasteClip inserts data from the Clipboard.

PasteOrig pastes entities from the clipboard at the coordinates from the source drawing.

PasteSpec pastes entities from the clipboard, after the user specifies the format.

PdfAdjust adjust the fade, contrast and monochrome settings of PDF underlays.

Pdfattach and **-PdfAttach** attaches PDF files as underlays into the drawing.

PdfClip clips PDF underlays.

PdfLayers controls the display of layers in PDF underlays.

PdfOptions controls the exporting of drawings in PDF format through the Settings dialog box.

PEdit edits polylines, 3D polylines, and 3D meshes (short for “polyline edit”).

PEditExt edits vertices and segments of a polyline.

Perpendicular toggles perpendicular entity snap mode.

PFace draws 3D multi-sided meshes; meant for use by programs (short for “polyface mesh”).

Plan sets plan view to construction plane.

PLine draws polyline lines, arcs, and splines with optional width (short for “polyline”).

Plot and **-Plot** both execute the plot command at the command line.

PlotStyle sets the current plot style; works only when plot styles are enabled in drawings.

PlotterManager creates customized parameter PC3 files for printers and other output devices; executes the PlotConfig.exe utility program.

Point draws point entities.

PointLight places point lights in drawings.

Polygon draws equi-sided polygons from polylines of 3 to 1,024 sides.

PolySolid creates 3D wall-like solids.

Preview shows a preview before printing the drawing.

Print plots the drawing to a plotter, printer, or file.

ProfileManager sets current, create, copy, delete, import and export user profiles.

ProjectGeometry projects geometry like curves, and edges onto regions, surfaces, and 3D solids.

Properties displays the Properties palette to change drawing entity properties.

PropertiesClose closes the Properties palette.

PSetupIn and **-PSetupIn** imports page setup definitions from another drawing.

PSpace switches from model to paper space (short for “paper space”).

Publish and **-Publish** prints sheet lists of model space or paper space layouts; saves a sheet list to a file.

Purge and **-Purge** remove unused named entities from drawings, such as unused layers and linetypes.

Pyramid draws three-dimensional solid pyramids.

Q Commands

QLeader draws leaders; specifies properties through a dialog box.

QNew opens new drawings in BricsCAD (short for “quick new”).

QPrint prints the drawing with the default plot configuration, without displaying the Print dialog box (short for “quick print”).

QSave saves the drawing without displaying the Save dialog box (short for “quick save”).

QSelect composes a selection set using filters.

QText toggles the display of text as rectangles (short for “quick text”).

Quadrant toggles snaps to quadrant points of circles, arcs, and polyarcs.

Quick toggles snaps to the first entity geometry found; used together with at least one other entity snap mode.

Quit ends BricsCAD; optionally saves unsaved drawings.

R Commands

Ray draws semi-infinite construction lines

ReAssocApp associates extended entity data with applications (short for “reassociate application”).

Recover repairs damaged drawings.

RecScript records keystrokes to an SCR file for playback with the Script command (short for “record script”).

Rectang draws a rectangular polyline.

Redefine restores built-in commands that have been undefined using the Undefine command.

Redo reverses the effects of a previous U command.

Redraw refreshes the display of the active view tile.

RedrawAll refreshes the display of all currently-open view tiles.

RedSdkInfo reports on rendering related hardware and driver specifications (short for “Red software development kit information”).

RefClose closes the in-situ block and xref editor.

RefEdit and **-RefEdit** edits blocks and externally-referenced drawings (short for “reference editor”).

RefSet adds and removes entities from the block or external reference being edited.

Regen regenerates the current viewport.

RegenAll regenerates all viewports.

RegenAuto determines when BricsCAD regenerates the drawing automatically.

Region converts an entity enclosing an area into a region.

Relnit reloads the PGP alias file (short for “re-initialize”).

Rename and **-Rename** changes the names of objects.

Render and **-Render** generates photorealistic renderings of 3D models using materials and lights.

RenderPresets creates and edits rendering presets, and to set the current render preset.

ResetBlock resets dynamic blocks to their default values.

Resume resumes an interrupted script.

RevCloud draws revision clouds commonly used for red-lining drawings.

Revolve draws 3D solids or surfaces by revolving 2D objects about an axis.

RevSurf creates 3D mesh surfaces by revolving open entities around a axis (usually a line).

Ribbon displays the ribbon user interface.

RibbonClose closes the ribbon.

Rotate rotates entities about a base point.

Rotate3D moves entities about a 3D axis.

RScript reruns the currently loaded SCR script file (short for “repeat script”).

RtLook moves the viewpoint through a 3D scene (short for “real time looking”).

RtPan pans the view in real time.

RtRot, **RtRotCtr**, or **RtRotF** rotate the viewpoint in real time.

RtRotX, **RtRotY**, or **RtRotZ** rotates the 3D viewpoint about the x, y, or z axis in real time.

RtUpDown tilts the viewpoint up, down, left, or right in real time.

RtWalk walk lefts, right, forward or backward through 3D scenes in real time.

RtZoom zooms into the drawing in real time.

RuleSurf draws ruled surfaces between two curves.

S Commands

Save saves the drawing under the current file name or a specified name.

SaveAll saves all open drawings.

SaveAs saves an unnamed drawing with a file name or renames the current drawing.

SaveAsR12 saves drawings in DWG R12 format.

Scale enlarges or reduces specified entities equally in the X,Y, and Z directions.

ScaleListEdit and **-ScaleListEdit** edits the list of scale factors used by annotative scaling, sheet scales, and plot scales.

Script loads and runs SCR script files.

Scrollbar toggles the display of the horizontal and vertical scroll bars.

Section creates a cross section based on the intersection of a plane and 3D solids.

SectionPlane creates a section entity that creates sections of 3D solids.

SectionPlaneSettings defines the properties of section plane entities in the Drawing Explorer.

SectionPlaneToBlock saves the selected section plane as a 2D cross section / elevation block or a 3D cutaway section block

Security determines whether VBA macros can run automatically; not available in the 64-bit version.

SecurityOptions sets a password to protect the drawing (WINDOWS ONLY).

Select places selected entities in the 'Previous' selection set.

SelectAlignedFaces selects all faces in a model which are coplanar with a selected face.

SelectAlignedSolids Selects all solids in a model of which a face is coplanar with a selected face.

SelectConnectedFaces selects all faces in a model which are connected to a selected face.

SelectConnectedSolids selects all solids in a model which are connected to a selected face.

SelectSimilar selects entities of the same type and properties.

SelGrips prompts to select entities and then displays grips.

Settings displays the Settings dialog box for changing the values of variables.

SettingsSearch opens the Settings dialog box at the specified category, variable name, or user preference.

SetUCS sets the UCS to a viewpoint specified through a dialog box.

SetVar displays and changes the values of system variables (short for "set variables").

Sh and **Shell** open the Windows command prompt window; runs other applications (short for "shell").

Shade shades the drawing mode.

ShadeMode sets the current visual style at the command line, such as Realistic, Conceptual, Edges, and X-ray.

-ShadeMode sets the old type of shade modes: 2D, 3D, Hidden, Flat, Flat with Edges, Gouraud, and Gouraud with edges.

Shape places shapes from SHX files in drawings.

SheetSet and **SheetsetHide** manage sheet sets, and closes the Sheet Set pane.

Singleton toggles whether multiple copies of BricsCAD can run at the same time.

Sketch draws freehand lines.

Slice slices 3D solids with a plane or surface.

Snap restricts pointer movements and pointing in the drawing to specified intervals.

Solid draws solid-filled 2D faces.

SolidEdit edits 3D solids and 2D regions.

SolProf creates hidden line representations of 3D solids in a layout viewport.

Spell checks the spelling of text in the drawing.

Sphere draws three-dimensional solid spheres.

Spline draws quadratic or cubic non-uniform rational Bezier spline (NURBS) curves.

SpotLight inserts spot lights into drawings.

Start runs operating system applications.

StatBar toggles the display of the status bar.

Status reports status of the drawing's settings in the Text window.

StdOut export 3D models in STL format for 3D printing (short for "stereolithography").

StopScript stops recording of scripts begun with the RunScript command.

Stretch moves or stretches entities.

Style and **-Style** creates and edits text styles through the Drawing Explorer.

StylesManager creates and attaches plot style files.

Subtract creates a composite region or a 3D solid by subtraction.

SunProperties edits sun properties through the Drawing Explorer.

SupportFolder opens the C:\Users\<login>\AppData\Roaming\Bricsys\BricsCAD\VI 7x64\en_US\Support folder.

SvgOptions controls the output as SVG files.

Sweep creates solid primitives or surfaces by sweeping two dimensional entities along a path.

SysWindows arranges windows.

Sheet Metal Commands

(Available for Platinum edition; requires an additional license; sm = sheet metal)

LicPropertiesSheetmetal reports the license state of the sheet metal module.

smBendCreate converts hard edges (sharp edges between flange faces) into bends.

smBendSwitch converts bends to lofted bends.

smConvert automatically recognizes flanges and bends in a 3D solid.

smDelete removes a bend or a junction by restoring the hard edge between two flanges; removes a flange with all the bends adjacent to it.

smDissolve removes sheet metal data from the selected features.

smExport2D exports unfolded representations of sheet metal bodies as 2D profiles in DXF or DWG files.

smExportOSM exports sheet metal solids to OSM files (short for "Open Sheet Metal") used by CADMAN-B CAM systems.

smFlangeBase creates base (initial) flanges of sheet metal parts from closed 2D entities.

smFlangeBend bends existing flanges along a line, taking into account the k-factor.

smFlangeConnect closes gaps between two arbitrarily oriented flanges.

smFlangeEdge creates one or more flanges to a sheet metal part by pulling one or more edges of an existing flange.

smFlangeRotate rotates a selected flange of a sheet metal part with automatic selection of the rotation axis depending on the design intent.

smFlangeSplit splits a flange along a line drawn on its face.

smForm adds forms to sheet metal.

smJunctionCreate converts hard edges (sharp edges between flange faces) and bends into junctions.

smJunctionSwitch changes symmetrical junction features to overlapping faces.

smLoft creates sheet metal part with lofted bends and flanges from two non-coplanar curves.

smReliefCreate creates proper corner (three or more adjacent flanges) and bend reliefs (at the start and end of a flange edge).

smReliefSwitch converts corner reliefs a circular, rectangular or V-type relief. Allows to change the parameters of existing corner reliefs.

smRepair restores the 3D solid model of a sheet metal part by thickening one of its sides: all thickness faces become perpendicular to flange faces.

smReplace replacing form features with ones from libraries.

smRethicken restores the 3D solid model of a sheet metal part by thickening one of its sides (all thickness faces become perpendicular to flange faces).

smSelectHardEdges selects all hard edges on sheet metal parts.

smUnfold generates unfolded 2D or 3D representations of sheet metal parts.

T Commands

Table and **-Table** draws tables in drawings.

Tableedit edits text in table cells.

TableExport exports the contents of a table entity to CSV (command separated values) files.

TableMod modifies the properties of table cells.

TableStyle creates and manages table styles through the Drawing Explorer.

Tablet configures and calibrates tablets, and toggles tablet mode (WINDOWS MODE).

TabSurf draws tabulated surfaces from a path curve and a direction vector.

Tangent toggles tangent entity snap; snaps to the tangency of circles, arcs, ellipses and elliptical arcs.

TemplateFolder opens the C:\Users\<login>\AppData\Local\Bricsys\BricsCAD\VI7x64\en_US\Templates folder.

Text and **-Text** places lines of text in the drawing.

TextScr displays the text window showing command history (short for “text screen”).

TextToFront sets the draw order of all texts and dimensions in the drawing to display in front of all other entities.

Time reports on the time spent in the drawing.

Tolerance draws tolerances (datum indicators and basic dimension notation).

Toolbar and **-Toolbar** displays and hides toolbars.

ToolPalettes opens the Tool Palettes bar.

ToolPalettesClose Closes the Tool Palettes bar.

-ToolPanel opens tool panels by name at the command bar.

Torus draws three-dimensional torrid solids.

TpNavigate opens tool palettes or group at the command bar.

Trace draws traces.

Transparency toggles the transparency of monotone images; has nothing do with the transparency property

Trim trims entities at a cutting edge defined by other entities.

TxtExp explodes text into polyline segments.

U Commands

U reverses the most recent command.

Ucs creates and displays named UCSes through the command bar (short for “user-defined coordinate system”).

UcsIcon toggles the display of the UCS icon.

Undefine disables built-in commands.

Undo restores deleted entities.

Union creates composite regions or solids by addition.

UnisolateObjects makes entities visible again following the IsolateObjects and HideObjects commands

Units and **-Units** sets coordinate and angle display formats and precision.

UpdateField forces the values of field text to update.

Url opens the default Web browser (short for “uniform resource locator”).

V Commands

View and **-View** saves, restores, and manages user-defined model and sheet views, and presets views.

ViewLabel adds labels to views; available through the Sheet Set manager only.

ViewRes sets the view resolution and toggles fast-zoom mode (short for “view resolution”).

VisualStyles and **-VisualStyles** creates and edits visual style definitions in the Drawing Explorer or at the command line.

VmlOut exports drawings in VML format embedded in Web pages (short for “vector markup language”).

VpClip clips viewports in layouts (short for “view port clipping”).

VpLayer changes the properties of layers in the current paper space viewport (short for “view port layer”).

VPoint Changes the 3D viewpoint through a dialog box.

VPorts and **-VPorts** create one or more viewports in model space (short for “viewports”).

VSlide displays images saved as SLD or WMF files (short for “view slide”).

Vba Commands

(Available in Pro and Platinum editions only; requires a separate download as of V17; vba = Visual Basic for Applications)

Vbalde opens the VBA editing window; short for “integrated development environment” (WINDOWS ONLY).

VbaLoad and **-VbaLoad** loads VBA projects (WINDOWS ONLY).

VbaMan manages VBA projects; short for “manager” (WINDOWS ONLY).

VbaRun and **-VbaRun** runs, creates, edits, and deletes VBA macros (WINDOWS ONLY).

VbaSecurity sets the security level for running VBA macros.

VbaUnload unloads VBA projects (WINDOWS ONLY).

ViewBase Commands

(Available in Pro and Platinum editions only)

ViewBase generates associative orthographic and standard isometric views of a 3D solid model in a paper space layout.

ViewDetail creates a detail view of a portion of a standard generated drawing at a larger scale.

ViewDetailStyle specifies the visual format of detail views and detail symbols.

ViewEdit changes the scale and hidden line visibility of drawing views; works in paper space only.

ViewExport exports the content of drawing views to Model space or to a new drawing; operates in paper space only.

ViewProj generates additional projected views from an existing drawing view.

ViewSection creates cross section views based on standard drawing views generated by the ViewBase command in a paper space layout.

ViewSectionStyle specifies the visual format of section views and section lines.

ViewUpdate updates drawing views.

W Commands

WBlock and **-WBlock** export blocks, selected entities, or the entire drawing as a DWG file.

WCascade, **WClose**, **WCloseAll**, **WNext**, and **WPrev** cascade the windows, close the current window, close all windows, and switch to the next or previous windows.

Weblight places Web lights.

Wedge draws three-dimensional solids with a sloped face tapering along the X axis.

WhoHas reports the ownership of a drawing file.

WhTile, **WiArrange**, and **WvTile** tiles windows horizontally, arranges tiled windows in an overlapping manner, or tiles them vertically.

WipeOut creates blank areas in drawings.

WmfOut exports the drawing in WMF (WIndows meta file), EMF (enhanced meta file), or SLD (slide) format.

WorkSets creates and loads named sets of drawing files.

Workspace sets the current workspace; creates, modifies, and saves workspaces.

WsSaves saves the current user interface by name.

WsSettings opens the Customize dialog box at the Workspace tab.

X Commands

XAttach attaches externally-referenced drawings.

XClip clips externally-referenced drawings.

XEdges extracts edges from 3D solids as lines.

XLine draws infinitely long lines.

XOpen opens externally-referenced drawings in a new window.

Xplode explodes entities, and provides control over the resulting entities.

XRef and **-XRef** attaches DWG files to the current drawing through the Drawing Explorer or the command line.

Z Commands

Zcenter toggles the 3D center entity snap; snaps to the center of planar or curved 3D faces.

Zknot toggles the 3D knot entity snap; snaps to a knot on a spline.

Zmidpoint toggles the 3D midpoint snap; snaps to the midpoint of a face edge.

Znearest toggles the 3D nearest entity snap; snaps to a point on the face of a 3D entity that is nearest to the cursor.

Znone disables all 3D snap modes.

Zoom increases or decreases the visible part of the drawing.

Zperpendicular toggles the 3D perpendicular entity snap; snaps to a point perpendicular to a face.

Zvertex toggles the 3D vertex entity snap; snaps to the closest vertex of a 3D entity.

Commands

? displays the Help window.

2dIntersection toggles apparent intersection entity snap; snaps to the intersections of entities, even when they only appear to intersect in 3D space.

3D draws 3D polygon mesh objects: boxes, cones, cylinders, dishes, domes, pyramids, spheres, tori, wedges, or meshes.

3DArray constructs 3D rectangular arrays and rotated polar arrays.

3DCompare compares the 3D content of two drawing files.

3DConvert converts 3D solids to polyface meshes.

3DFace draws 3D 4-edged faces with optional invisible edges.

3DIntersection toggles Intersection entity snap; snaps to the intersections of entities.

3DMesh draws 3D surface meshes.

3DOsnap and **-3DOsnap** sets the entity snap modes for 3D entities through the Settings dialog box.

3DPoly draws 3D polylines.