



Inside **BricsCAD**[®]

V19

Copyright Information

Copyright © 2018 by upFront.eZine Publishing, Ltd.

All rights reserved worldwide.

This 10th edition is based on BricsCAD V19

28 December 2018

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.

Technical Writer **Ralph Grabowski**

Technical Editing **Bricsys Staff**

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.

Visit the *Inside BricsCAD* Web site at www.upfrontezine.com/ib8. At this Web page, editions of this book are available for BricsCAD V8 through V17.

Brief Contents

Introduction

1 A Quick Tour Through BricsCAD.....	1
2 Navigating the BricsCAD Interface	21

2D Tutorials

3 Setting Up A New Drawing.....	53
4 Creating Your First Drawing	81
5 Adding Details to Drawings.....	109
6 Making Changes to Drawings.....	143
7 Adding Notes and Dimensions.....	163
8 Bills of Material.....	199

Advanced Topics

9 Working with 2D Regions & Booleans	235
10 Direct 3D Modeling & Editing.....	255
11 Dimensional & Geometric Constraints	295

Appendices

A Concise Summary of Command Names.....	311
B Concise Summary of Variables & Settings	337
C What's New in BricsCAD V19	361

Table of Contents

1 A Quick Tour Through BricsCAD	1
How to Start BricsCAD	3
Starting BricsCAD on Windows.....	3
Windows 7.....	3
Windows 8.....	3
Windows 10.....	4
Starting BricsCAD on Linux.....	5
Starting BricsCAD on Mac MacOS.....	5
BricsCAD V19 User Interface	6
Getting Started	8
The BricsCAD Window.....	9
A Basic Tour of the User Interface.....	10
Crosshair and Arrow Cursors	10
Command Bar	11
All About Command Prompts.....	12
Undoing What You've Done: U.....	14
Seeing What You Did Before: Command History	15
Typing Less: Aliases and AutoComplete.....	16
UCS Icon.....	18
Online Help	19
SUPPORTED GRAPHICS BOARDS	20
Exiting BricsCAD	20

2 Navigating the BricsCAD Interface 21

IN THIS CHAPTER 21

Above the Drawing Area 22

Title Bar 23

Menu Bar..... 24

Toolbars 25

Toolbar Buttons and Macros 27

The Standard Toolbar 27

Flyouts 28

Droplists 29

Changing Properties..... 30

Other Panels 32

USER INTERFACE ELEMENTS OF PANELS 33

Ribbon Tabs and Panels 35

Drawing Tabs..... 36

In the Drawing Area..... 37

Shortcut Menus..... 37

quad cursor 37

Quick Properties 39

Scroll Bars 40

Prompt Menu..... 40

Look-From Control..... 40

The UCS Icon in 2D Drafting, 3D Modeling, and DUCS 42

Hotkey Assistant..... 45

Manipulator Widget..... 45

Below the Drawing Area 47

Layout Tabs..... 47

Status Bar 48

3 Setting Up A New Drawing..... 53

Before You Begin 55

Starting A New Drawing..... 56

Default Settings..... 57

HOW TO START NEW DRAWINGS 58

Selecting the Linear Units.....	60
Selecting the Angle Style.....	61
Choosing the Plot Style.....	62
Setting Entity Properties.....	63
Finishing the Wizard.....	63
Additional Important Settings.....	64
Setting Drawing Limits.....	64
Accessing and Changing Settings.....	67
Changing the Snap and Grid Spacing.....	69
Creating Layers.....	71
Naming Layers.....	72
QUICK SUMMARY OF LAYERS.....	73
Saving Drawings.....	75
QUICK SUMMARY OF THE SAVEAS COMMAND.....	76
Making Backups Automatic.....	77
QUICK SUMMARY OF ALTERNATIVE SAVE FORMATS.....	78
4 Creating Your First Drawing.....	81
Reopening Drawings.....	83
Drawing the Lot's Boundary.....	84
Planning the Next Steps.....	87
Changing Layers.....	87
Drawing the House Outline.....	89
QUICK SUMMARY OF THE PLINE COMMAND.....	90
Direct Distance Entry.....	91
QUICK SUMMARY OF ENTITY SNAP MODES.....	92
Dynamic Input.....	93
Moving the House into Position.....	95
Starting on the Driveway.....	97
Finishing the Driveway.....	98
Mirroring Entities.....	99
Putting Drawings to Paper.....	101
QUICK SUMMARY OF PRINT PREVIEW.....	103

Solving Printing Problems.....	104
Smudgy Prints with Bad Colors with Inkjets	105
Generating PDFs of Drawings.....	105
Specifying PDF Output Options.....	106

5 Adding Details to Drawings 109

Dividing the Lot 111

Smoothing Polylines.....	114
QUICK SUMMARY OF THE PEDIT COMMAND	115

Grips Editing 116

QUICK SUMMARY OF EDITING WITH GRIPS.....	118
Editing by Double-clicking.....	120
Hatching the Lawn	121
Modifying Hatching	123
QUICK SUMMARY OF DRAWING CIRCLES	124

Creating Symbols (Blocks) 124

Drawing Circles.....	125
Zooming in Real Time	125
Creating Arrays.....	126

Making Blocks..... 129

Adding Many More Trees (Insert)	130
---------------------------------------	-----

Drawing the Pond 132

QUICK SUMMARY OF DRAWING ELLIPSES.....	134
Array Paths.....	135
Drawing Splines.....	135
Zooming to Objects.....	136
QUICK SUMMARY OF SPLINE	137
QUICK SUMMARY OF POLYGON	138
Paving Stones from Polygons	138
Arraying Along a Path	139
QUICK SUMMARY OF ARRAYPATH.....	140
Editing Associative Arrays	141
QUICK SUMMARY OF ARRAYEDIT	141

6 Making Changes to Drawings	143
Changing the Look of Lines.....	145
Loading Linetypes	145
Changing Properties.....	147
QUICK SUMMARY OF PROPERTIES PANEL.....	150
Changing the Linetype Scale.....	150
Selecting Entities by Other Methods.....	151
Selecting Entities by Their Properties	151
QUICK SUMMARY OF SELECTION SET OPTIONS.....	153
Selecting an Entity by Tabbing.....	154
Changing Line Lengths	154
QUICK SUMMARY OF THE LENGTHEN COMMAND	155
Stretching the Pond	156
Moving Entities	157
Nudging Entities	158
Adding the Fence	159
Listing Data	160
7 Adding Notes and Dimensions.....	163
Adding Notes to Drawings	166
How to Determine the Size of Text	167
QUICK SUMMARY ON CALCULATING TEXT HEIGHTS	168
Creating Text Styles	169
Simple Text	171
QUICK SUMMARY OF TEXT JUSTIFICATION MODES.....	172
Changing Text	172
Placing Rotated Text.....	173
Placing Multiple Lines of Text.....	174
Finding and Replacing Text	175
Placing Dimensions in Drawings.....	177
Preparing Drawings for Dimensioning	177
Dimensioning the Yard	178
Vertical and Baseline Dimensions	180
Baseline Dimensioning.....	181

Aligned and Radial Dimensions	181
Radial Dimensions.....	182
Dim, the Super Dimension Command.....	183
Annotatively-Scaled Text and Dimensions	185
Using Annotative Scaling.....	187
A FEW EXTRA THINGS ABOUT ANNOTATIVE SCALING.....	187
Leaders and Multileaders.....	193
QUICK SUMMARY OF QLEADER	194
Single-line Leaders.....	195
Multi-line Leaders.....	196
QUICK SUMMARY OF MLEADER.....	197
8 Bills of Material	199
About Attribute Data.....	201
Creating Blocks with Attributes	202
Drawing Blocks.....	204
Defining Attributes.....	204
Adding More Attributes.....	207
Combining Entities and Attributes into Blocks.....	207
Inserting Blocks with Attributes.....	209
Alternatives to the Insert Command.....	210
Inserting Blocks with Drag'n Drop.....	211
Exporting Data from Drawings.....	212
Data Extraction	212
QUICK SUMMARY OF ADDITIONAL FORMATS.....	218
Importing Data Files into Spreadsheets.....	219
Placing Data in Drawings as Tables	221
Linking Data in Drawings with Tables.....	222
Editing a Datalink	225
Updating the Table.....	226
Automatic BOMs from 3D Components.....	227
Exploding Assemblies.....	227
QUICK SUMMARY OF BMEXPLODE.....	228
Generating BOMs.....	229

QUICK SUMMARY OF BMBOM	229
Attaching Balloons.....	231
QUICK SUMMARY OF BMBALLOON	231
9 Working with 2D Regions & Booleans	235
About Regions	237
How to create Regions	237
Differences Between Boundary and Region	238
QUICK SUMMARY OF BOUNDARY OPTIONS	239
What To Do with Leftovers	242
Tutorial: Creating Boundaries	242
QUICK SUMMARY OF BOOLEAN OPERATIONS	244
Boolean Operations	246
Tutorial: Creating a Waffle Shape.....	246
Measuring Regions.....	253
QUICK SUMMARY OF MASS PROPERTIES	254
10 Direct 3D Modeling & Editing	255
About 3D Solid Models	257
Planning Ahead.....	257
ELEMENTS OF 3D MODELS	258
Planning Ahead by Deconstructing 3D.....	259
Traditional Extrusion Method	259
Preparing the Drawing	260
Modeling a Box.....	262
QUICK SUMMARY OF EXTRUDE COMMAND.....	263
3D View Rotation	265
Extrusion by Direct Modeling.....	266
Applying Dynamic UCS	266
Extruding Directly with the Quad Cursor	269
Subtracting Solids To Make Holes	271
QUICK SUMMARY OF LOOKFROM COMMAND.....	272
PushPull Modeling.....	272
Aligning the UCS.....	275

Pulling, Instead of Pushing.....	276
Filleting 3D Solids.....	280
Joining Parts with Union.....	281
Generating 2D Drawings	282
QUICK SUMMARY OF VIEWBASE OPTIONS.....	285
Adding Section Views And Detailed Views.....	286
Resizing Views	286
QUICK SUMMARY OF VIEWSECTION OPTIONS.....	287
Making Section Views	288
QUICK SUMMARY OF VIEWDETAIL OPTIONS	290
Creating Detail Views	291

11 Dimensional & Geometric Constraints 295

Working with Constraints	297
About Dimensional Constraints.....	297
QUICK SUMMARY OF DIMENSIONAL CONSTRAINTS.....	298
Using Dimensional Constraints.....	300
QUICK SUMMARY OF 2D GEOMETRIC CONSTRAINTS.....	302
About Geometric Constraints.....	303
Identifying Geometric Constraints.....	303
Using Geometric Constraints	304

Appendices

A Concise Summary of Command Names..... 311

B Concise Summary of Variables & Settings 337

C What's New in BricsCAD V19 361

What's New in the User Interface	362
What's New in General Commands & Variables.....	365
What's New in Assemblies	388
What's New in Building Information Modeling.....	390
What's New in Communicator	397
What's New in Import & Export Commands.....	398
What's New in Generated Views	404
What's New in Modeling.....	405
What's New in Point CloudS.....	407
What's New in Sheet Metal.....	410
What's New in APIs.....	412
What's New in Licenses.....	414

CHAPTER 1

A Quick Tour Through BricsCAD

Welcome to BricsCAD V19!

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

In this chapter, 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 V19
- 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 to you from the operating system and from BricsCAD
- Flip screen** — switches between the drawing window and a text window
- Flyout** — shows a secondary toolbar when clicked
- Icon** — represents commands pictorially
- Layout** — determines 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	...
Ucslcon

* 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.

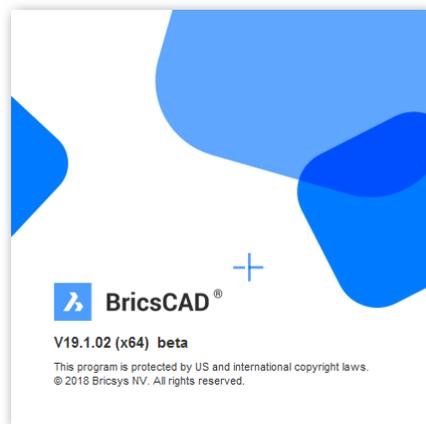
How to Start BricsCAD

If BricsCAD is not yet set up on your computer, do so. 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.



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 V19 works with Windows 7 and newer.

Windows 7

In Windows 7, you 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 V19** folder, and then click on **BricsCAD V19**.

The exact name you see depends on the language version you downloaded. For example, “BricsCAD V19 (x64) en_US” is the name of the 64-bit program for English speakers in the US dialect.

Windows 8

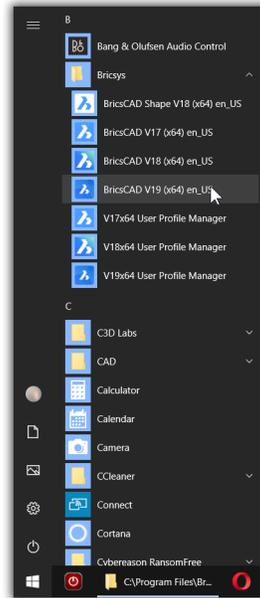
In Windows 8.x, you make these moves:

1. If necessary, switch to the **Start** screen.
2. In the Start screen, tap on the **BricsCAD V19** icon.

Windows 10

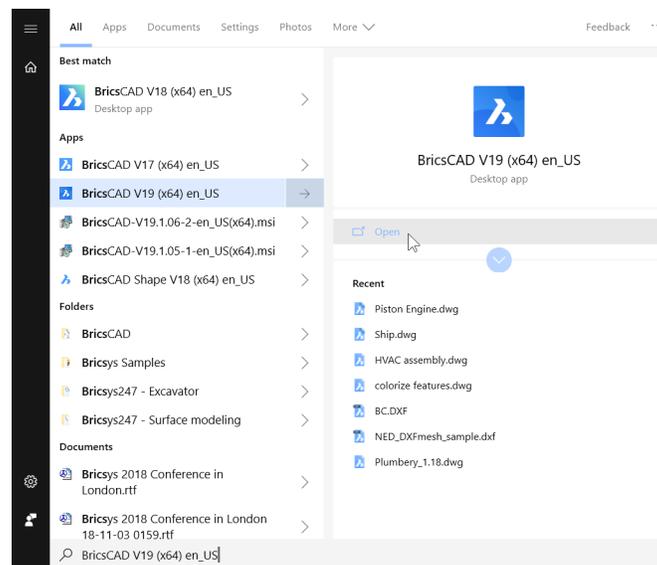
Under Windows 10, starting BricsCAD is like Windows 7:

1. Tap the **Start** button.
2. Choose **All Apps**.
3. In the **B** menu, tap **Bricsys** folder, and then tap the **BricsCAD V19** item.



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

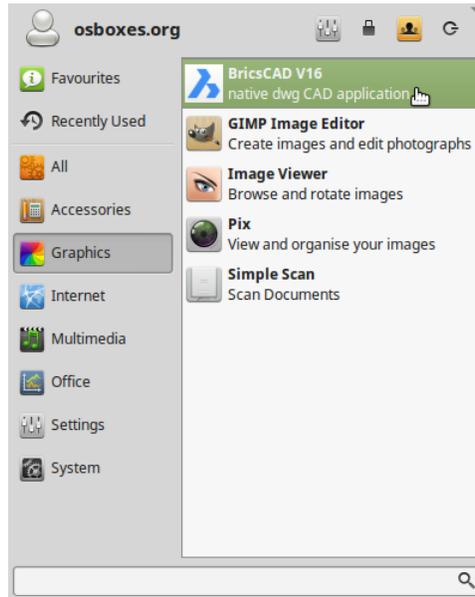
1. Tap the **Start** button.
2. Start typing "bricscad"
3. When you see **Bricsys V19**, tap it.



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 V19** icon:



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

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



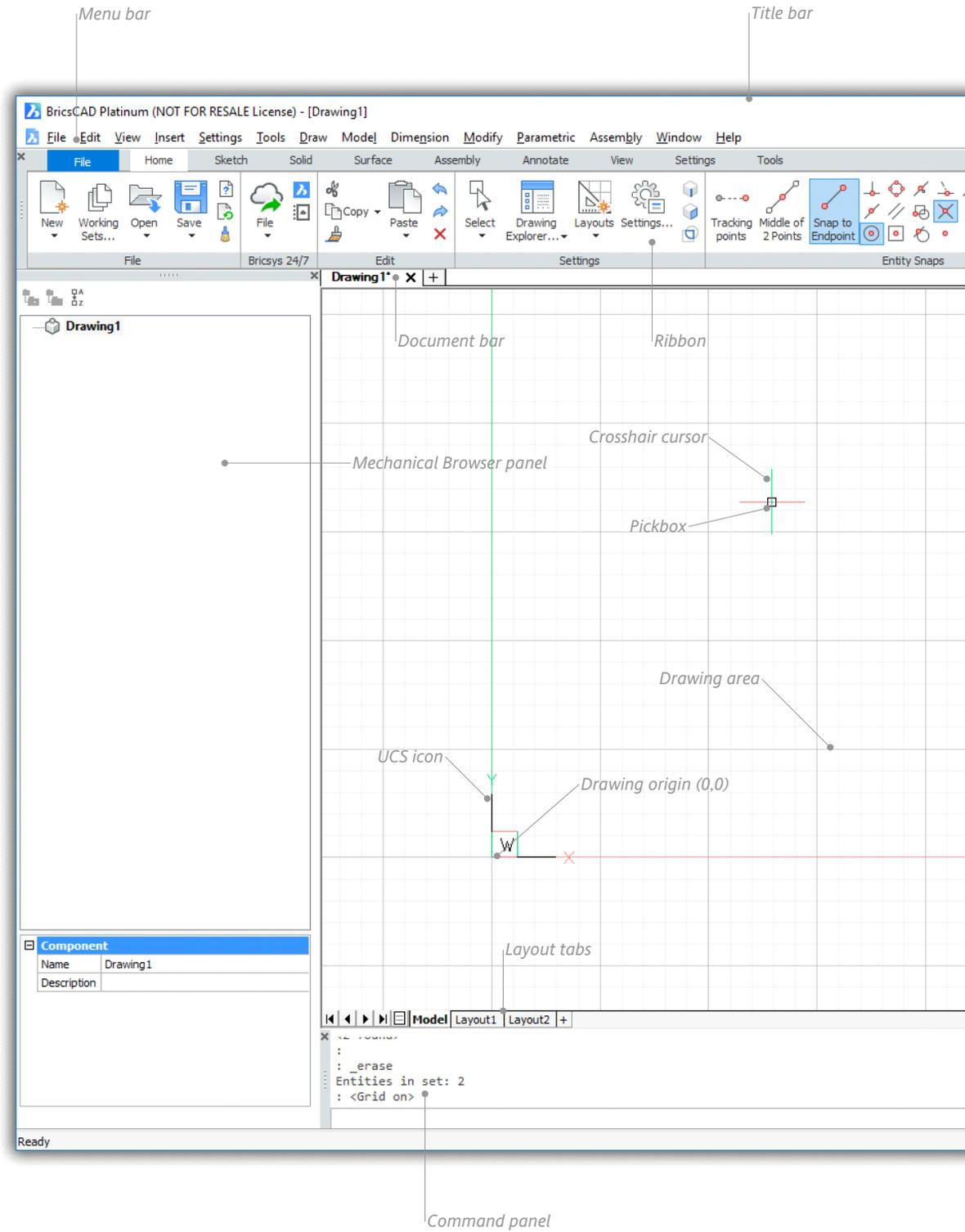
2. Find the **BricsCAD V19** icon, and then click it.



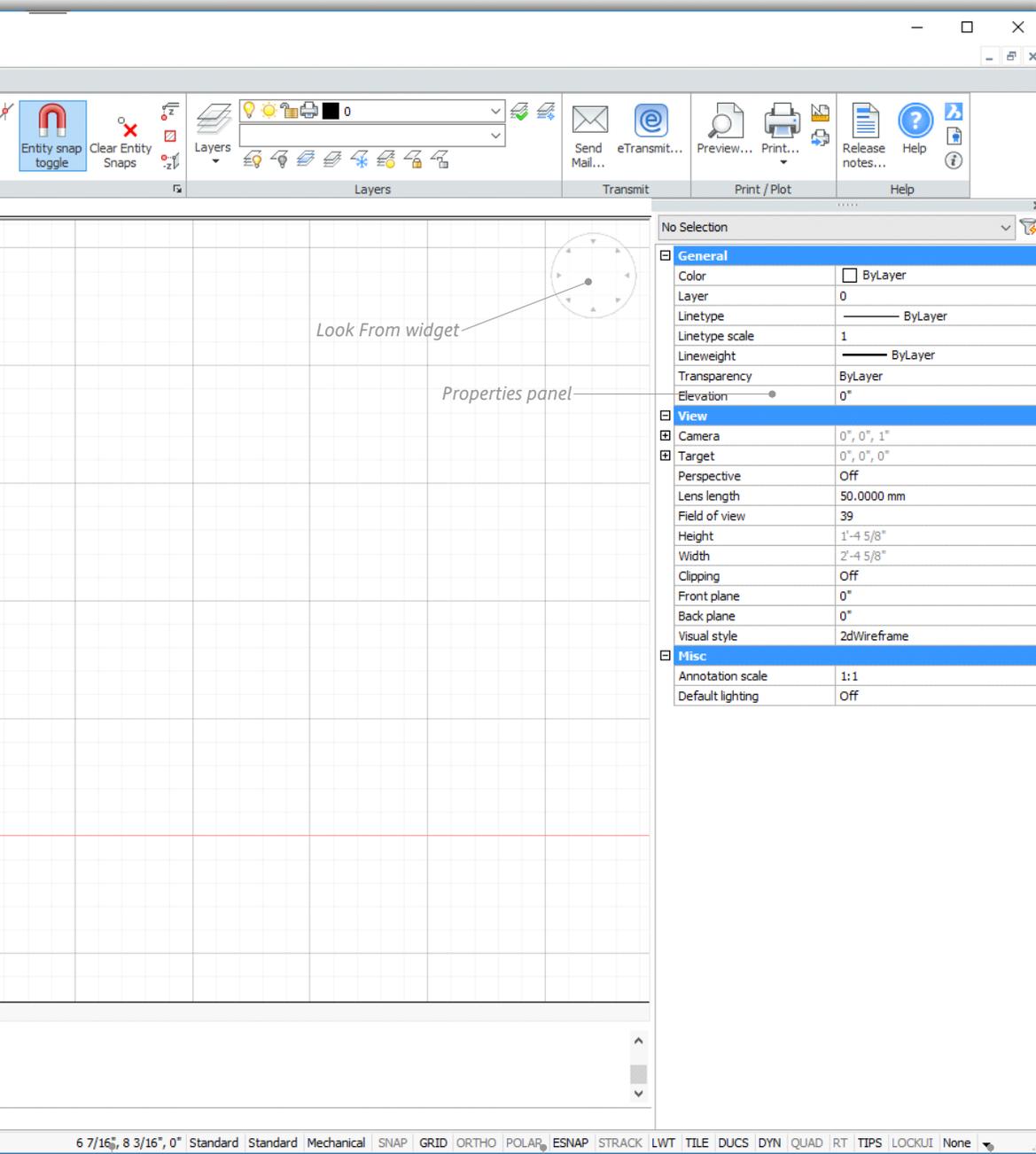
To keep the icon in the dock, right-click the BricsCAD icon. From the shortcut menu, choose **Options**, and then choose **Keep in Doc**.

BRICSCAD V19 USER INTERFACE

Illustrated is BricsCAD running on Windows 10.



No matter the operating system, BricsCAD looks pretty much the same in each one, as illustrated on these two pages. Instructions in this book specific to Linux and Mac are shown in gray text.



X, y coordinate and elevation (z)

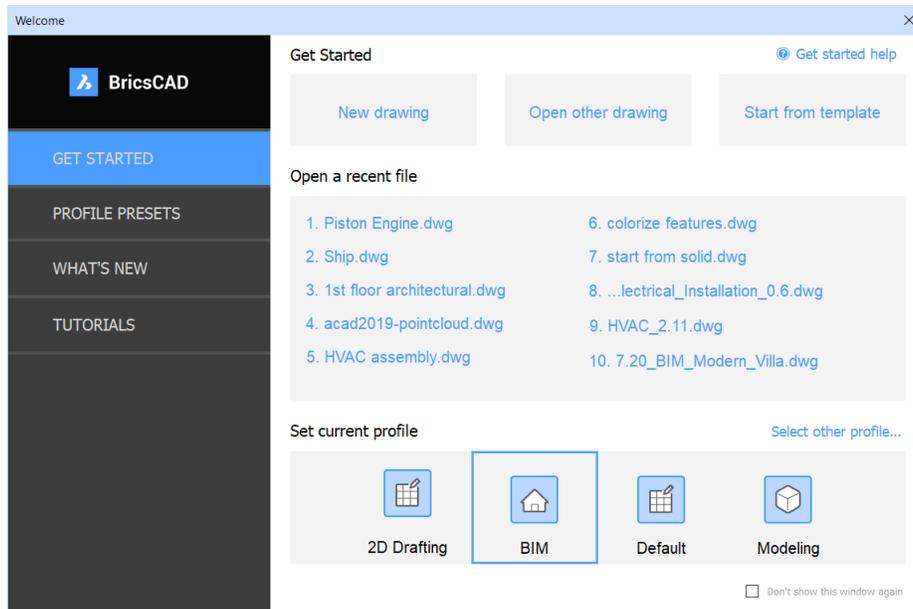
Status bar

Status bar options

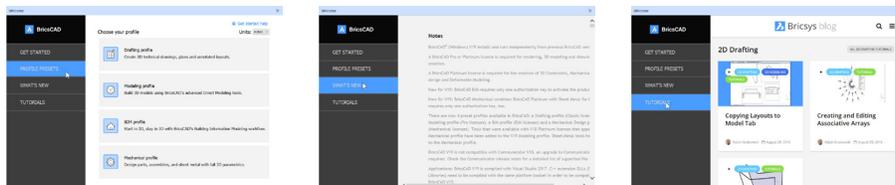
Resize window

Getting Started

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:



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

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 Blog hosted by Bricsys at <https://blog.bricsys.com/>

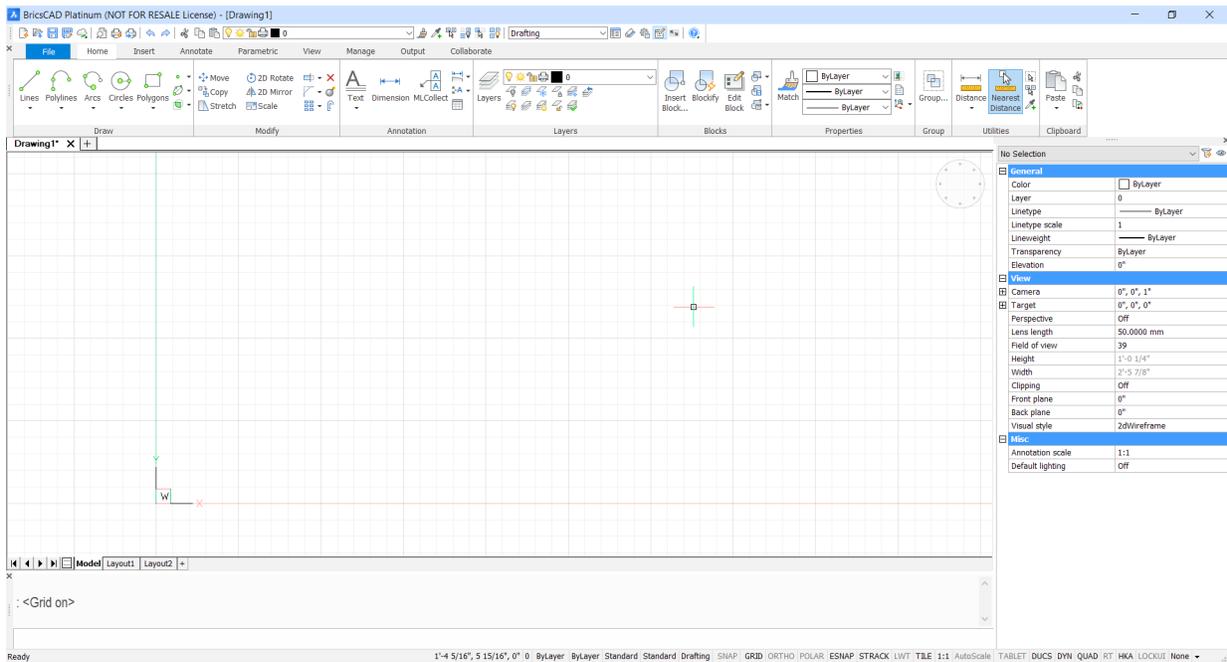
For now click **New Drawing** to enter BricsCAD.

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.)



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

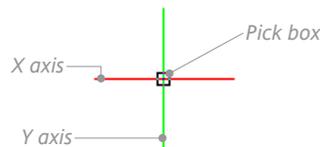
The user interface of any CAD program has many elements, and so it can be daunting to learn all of it at once. In this chapter, you look at just a few UI elements, the more important ones:

- ▶ Crosshair and arrow cursors
- ▶ Command bar
- ▶ UCS icon

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

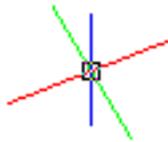
Crosshair and Arrow Cursors

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. They 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; you don’t see it 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 to select and edit them. Entities outside the pickbox will not be picked. You use the pickbox during Chapter 5, “Adding Details to Drawings.”

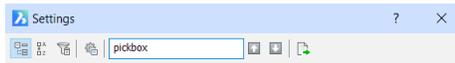
You can change the colors and the size of the crosshairs, as well as the size of the pick box, with the Settings command; see the tip coming up soon.

When you move the crosshair out of the drawing area, it changes its shape to an arrow — one with which you probably are familiar 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.

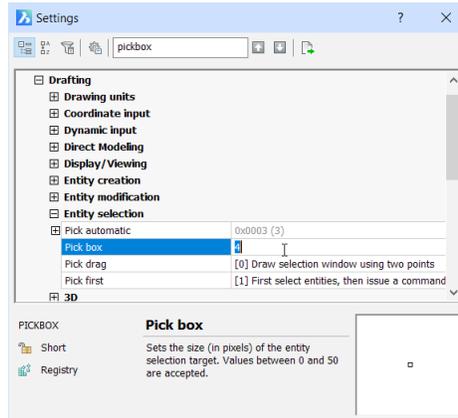


Left: Arrow cursor outside the drawing area, used to select UI elements;
right: Double-headed cursor used to change size of UI elements

Many user interface options are changed in BricsCAD with the **Settings** command. It provides you with over 1,000 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.

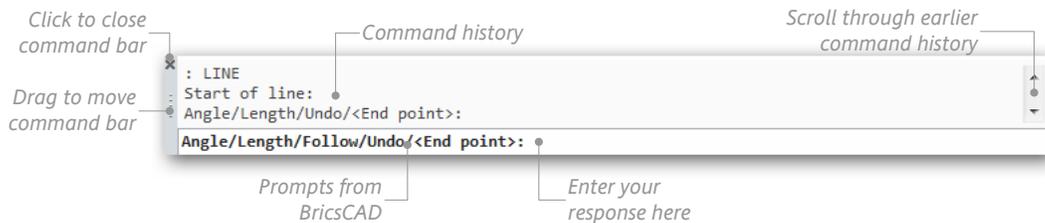


Changing the Crosshair Cursor Size. Should you find the crosshair cursor too small or too large, you change its size also through the Settings command: search for “crosshair.” The default value is 5, which means the length of the cross hairs is 5% of the screen’s size. When set to 100, the cross hairs 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 the one place where you enter the names of commands and specify their options. If you are a touch typist like me, then you’ll find that you 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! Here 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:

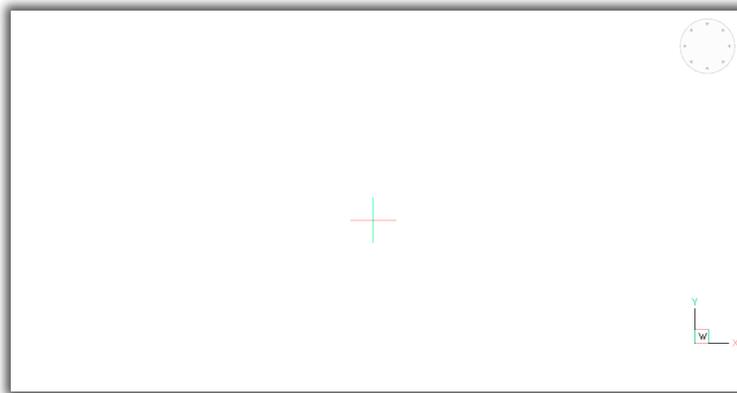
: **line** (Press **Enter**)

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:

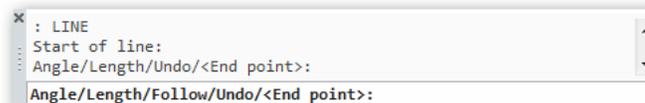
Start of line:

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.

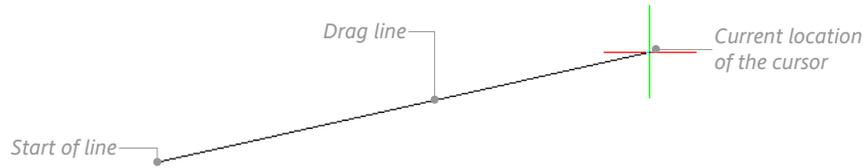


3. At any spot in the drawing area, pick a point on the screen by pressing the **first** button on your mouse. (The first button is the *left-most* one.) The left button is known universally in the CAD world as the "pick button." Notice that the Command bar changes the prompt wording by adding more options. I'll tell you their meanings later.

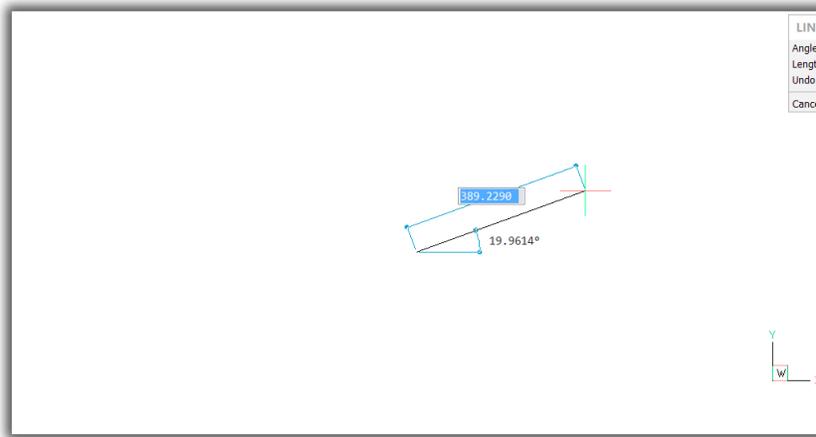
Set end point or [Angle/Length/Undo]: (Pick another point)



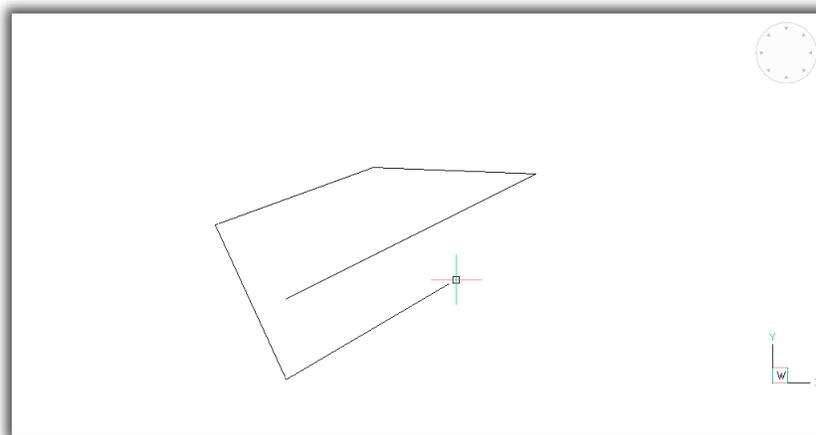
- As you keep moving the mouse, notice the line that stretches like a “rubber band” from the point you picked. This rubber band is also known as the “drag line.” It shows where the line would be located if you were to click now.



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 direct distance entry in a later chapter.



- Move the mouse some more, and then press the pick button again. There: you’ve drawn your first line with BricsCAD!
- 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.
Set end point or [Angle/Length/Follow/Close/Undo]: (Press **ESC**)

I want to point out that the prompt text — ‘**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.

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 at the prompt
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.

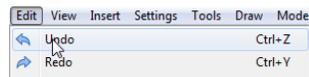
Undoing What You’ve Done: U

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

: u

Alternatively, you could access the U command in these other 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!

Trust me, Ctrl+Z will become your best friend!

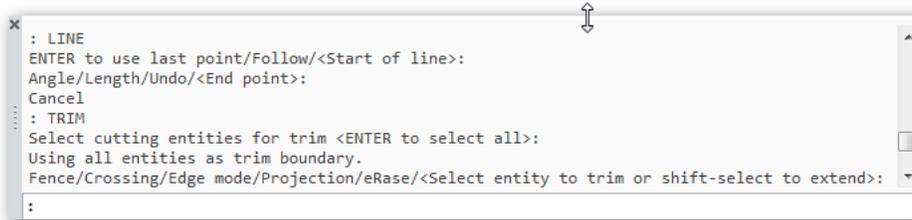
As you can see from the Undo example, BricsCAD provides several ways to perform the same action. In the days to come, you will probably find yourself using a combination of keyboard typing, toolbar or ribbon icons, menu picks, keyboard shortcuts — whichever one you find the most convenient.

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.

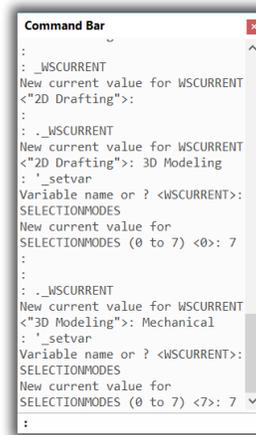
Seeing What You Did Before: Command History

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.



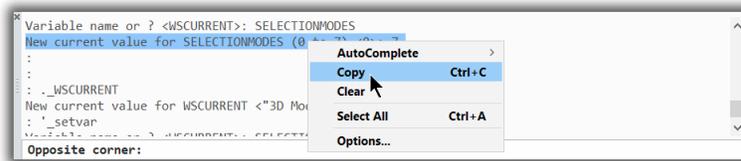
- ▶ Press **F2** to see the prompt window. BricsCAD display a second window on the computer screen. You can reposition it anywhere, such as on a second screen.



The Prompt History window displays the most recent 400 lines of command text. You can scroll back to earlier 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 typed 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\V19x64\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.



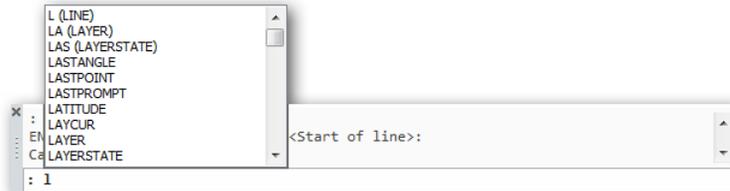
Typing Less: Aliases and AutoComplete

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.

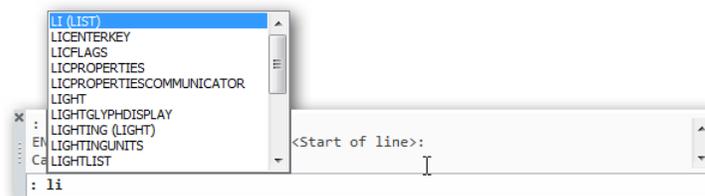
- ▶ **Aliases** — command abbreviations, such as 'l' for the Line command
- ▶ **Auto-complete** — lists the names of all commands that begin with the same letter(s) as you are typing

Aliases are described later in this book. 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 **i**, the second letter of the Line command. Notice that the suggestion list shortens to just the names that begin with 'li.'



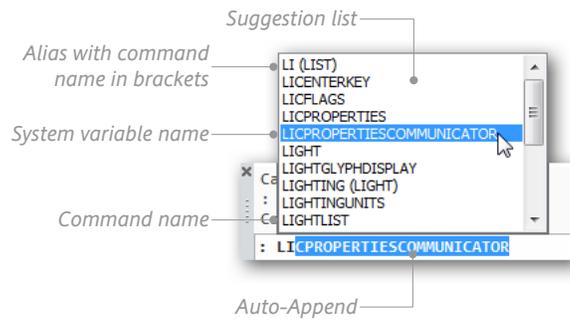
- a. You can keep typing letter of the command name to further reduce the suggestion list.



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 all the elements of auto-complete:



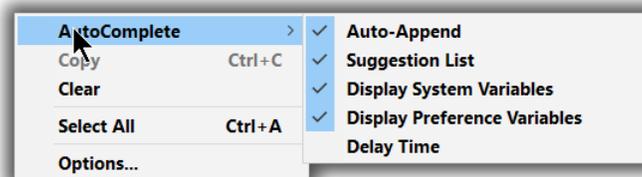
Suggestion list — lists the names commands and system variables; you can turn off the display of system variables to declutter the list, as described below.

Aliases — identified by appending the full command name in brackets, such as LI (LIST).

Systems variables — are 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 letters of the command name BricsCAD automatically completes 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 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 it and then choose **Options** from the shortcut menu.

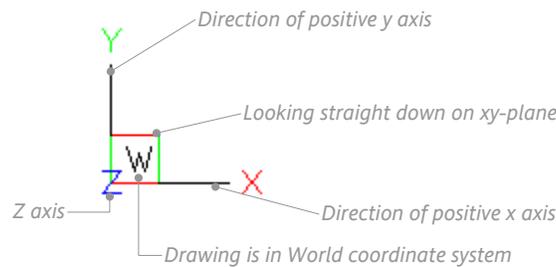


This brings up the Settings dialog box at the Command Bar section, where you can make changes to your heart's delight.

UCS ICON

Command	UcsIcon
Ribbon	...
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 whole bunch of information:



X,Y Axes — 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 — the 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,” or UCS, for short. A UCS is 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:

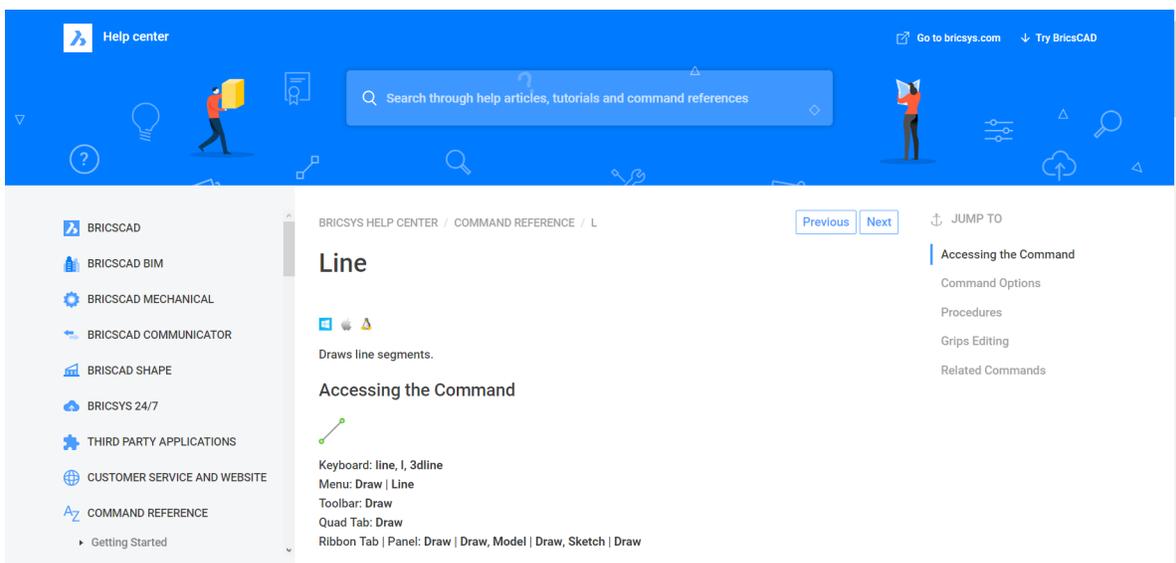
1. Type the **UcsIcon** command at the ':' prompt in the command bar, as follows:
: ucsicon (Press **Enter**)
Press **Enter** to execute the command.
2. Notice the next prompt displayed by BricsCAD:
[ucs icon ON/ucs icon OFF/display in All views/display at ORigin/display in Corner]
<ON>: **off** (Press **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
Alias	?
Menu Bar	Help Help
Ribbon	Home Help Help
Shortcut	F1

(**NEW IN V19**) To peruse help during a command, press **F1** and BricsCAD displays Help in your computer's Web browser, such as is illustrated below for the Line command. (New to V19 is that help appears in the Web browser.)

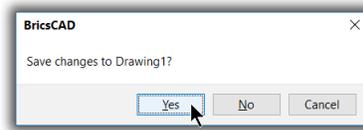


Icons indicate whether the command operates in each of the supported operating systems, Windows, MacOS, or Linux; most commands do. Other icons indicate which edition the command works with, such as Pro, Platinum, BIM, or Mechanical. For the complete list of commands, see Appendix A.

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.

SUPPORTED GRAPHICS BOARDS

BricsCAD works with whatever graphics board is built into your computer. When it comes to non-wire-frame 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

In this chapter, we continue the tour of BricsCAD's user interface (UI) by looking at it in greater detail. We will work our 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 nuances to the UI, such as entering aliases and working with relative coordinates.

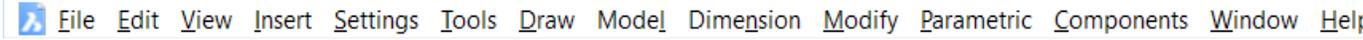
IN THIS CHAPTER

- 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)

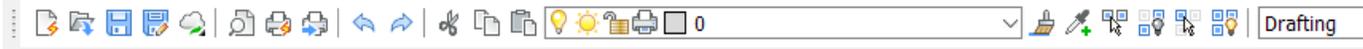
Title bar shows the name of the program and the current drawing, with program window controls at the far right:



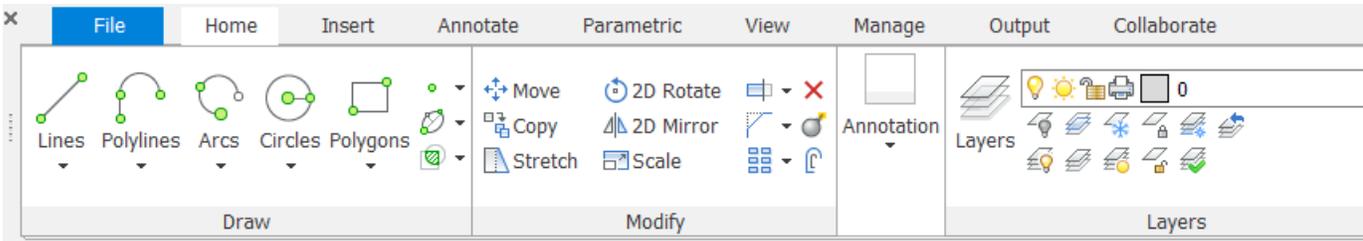
Menu bar arranges commands in logical groupings:



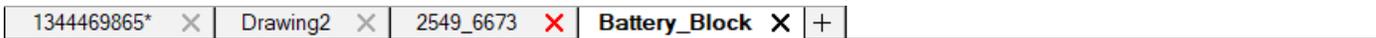
Toolbar collects buttons into groups, sometimes with flyouts and droplists:



Ribbon segregates commands into tabs and panels, with buttons, flyouts, and droplists:



Drawing tabs access all open drawings quickly:



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]"

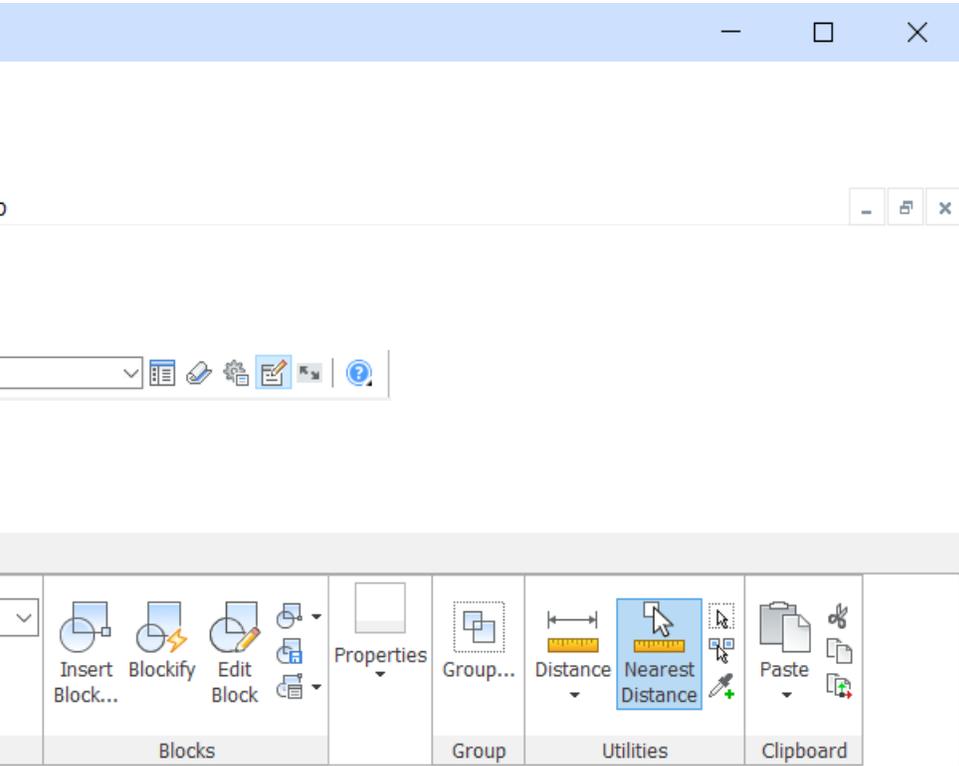
Menu bar — contains pull-down menus that list commands in related groups

Toolbars — collects buttons labeled with *icons* (miniature pictures) into bars of related 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 with which you are working, such as...

[Office Fixtures.dwg]

If the drawing is “read-only,” it cannot be saved by the same name to its original file location, usually because another copy of the drawing is already open elsewhere. In this case, 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:

Maximize the BricsCAD window quickly by *double-clicking* the title bar.

Restore the window by double-clicking the title bar a second time.

Open a drawing quickly by dragging it from the file manager to BricsCAD’s title bar.

Both ends of the title bar provide buttons and menus that control the size of the BricsCAD window. At the left end is a rarely used-menu that performs many of the same functions as the trio of buttons over on the right end: to — minimize, maximize or  restore the BricsCAD window, or else  exit BricsCAD.



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

MENU BAR

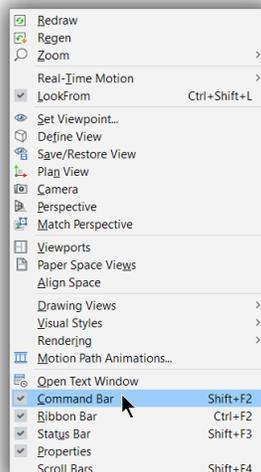
BricsCAD has a menu bar that operates identically to other programs running on Windows, MacOS, and Linux. If you do not see the menu in BricsCAD, enter the **MenuBar** variable, as follows:

: `menubar` (and then press the **Enter** key to execute the command)

Enter **1** to turn on the menu bar:

New current value for MENUBAR [1 for on(ON)]/ 0 for off(OF) <0 for Off>: 1

To access a menu, click a word on the menu bar, such as **View**, to reveal the dropdown menus, such as this one:



Here are some notes on what you see in the menus:

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

Menu use symbols to indicate the status of some commands, as shown by the table below:

Menu	Symbol	Example	Meaning
vt...	... (ellipsis)	Named Views...	Command will display a dialog box
▶	▶ (arrowhead)	Zoom ▶	Indicates the presence of a submenu
<input checked="" type="checkbox"/>	✓ (check mark)	✓ Clean Screen	The command is turned on
t+F	+ (plus)	Shift+F2	Uses the Shift key for command shortcuts
<u>C</u> o	_ (underline)	Command Bar	Uses the Alt key to access menu items

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. On the menu bar, notice that each word has a letter underlined. To access the View menu, for instance, hold down the **Alt** key, and then press the **v** key.



File Edit **View** Insert Settings Tools Draw Dimension Modify Parametric Window Help

(Alternatively, use the left and right cursor keys to move the blue highlight to the menu name.)

2. Notice that the View menu drops down. Press a key to execute a command, such as **c** key for **Command Bar**. (Again, you can manoeuvre to it with the up and down cursor keys.)

Advanced users who are interested in customizing menus and other parts of BricsCAD with the **Customize** command, shown consult the *Customizing BricsCAD* ebook, which is available to download as a PDF file free from <https://help.bricsys.com/hc/en-us> (scroll to the bottom of the Web page).

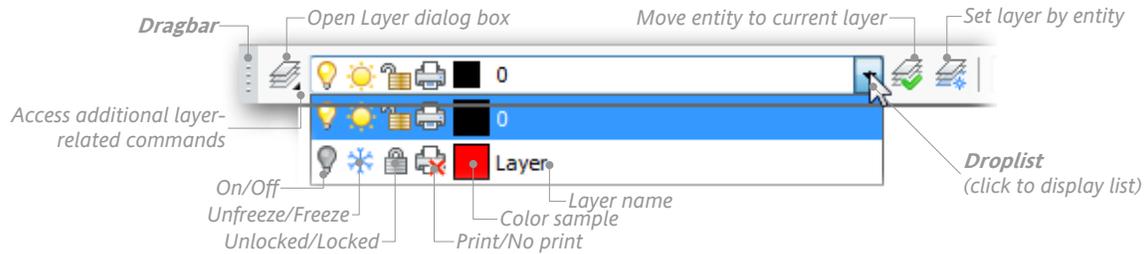
TOOLBARS

Below the menu bar might be several toolbars. I say “might be,” because toolbars can be turned on and off. *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.

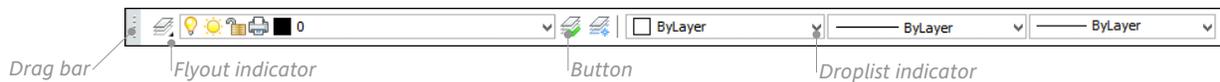


Left: Toolbar for drawing 2D entities; **right:** toolbar for connecting entities with 3D constraints

Each toolbar consists of a row of buttons, and sometimes list boxes that 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 controls you can expect to find on toolbars.



Toolbars employ visual elements to control them:

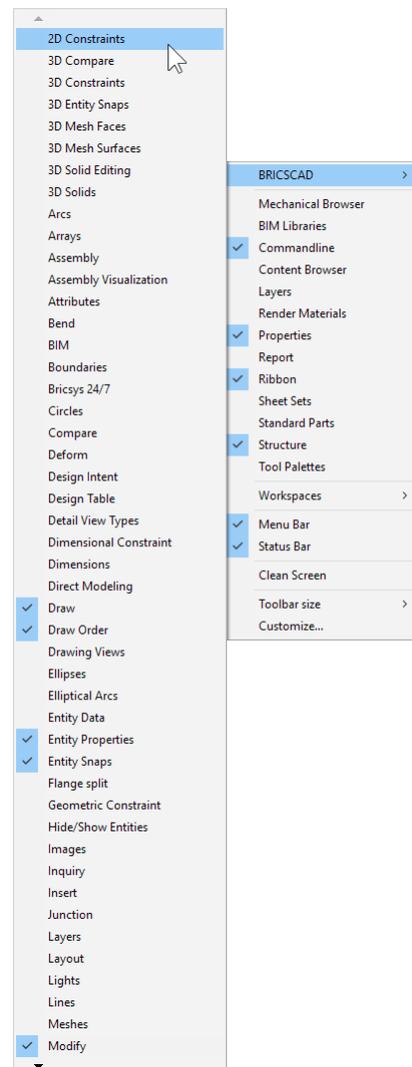


- ▶ *Drag bar* at the left end of a docked toolbar lets you move the toolbar around the screen. To drag a floating toolbar, grab it by its title bar.
- ▶ *Flyout* (small black triangle) displays an embedded toolbar; hold down on the button to see the flyout
- ▶ *Droplist* (a.k.a. list box) lists items;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

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.

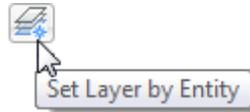
BricsCAD lets you change the look of the icons displayed by all toolbars, the function of the icons, and even the shape of the toolbar. These operations are described in the *Customizing BricsCAD* ebook.



Toolbar Buttons and Macros

Toolbar buttons have small pictures called “icons.” Icons are pictorial representations of commands. Because icons are pictures, sometimes their meaning is clear, and other times not. For example, a button shows this  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.

For this reason, BricsCAD also displays word descriptions for every icon. Pass the cursor over an icon, and then wait for a second. A small tag, called a “tooltip,” appears. The tooltip tells you the command executed by the button; in this case, it is “Set Layer by Entity” (LayMCur command).



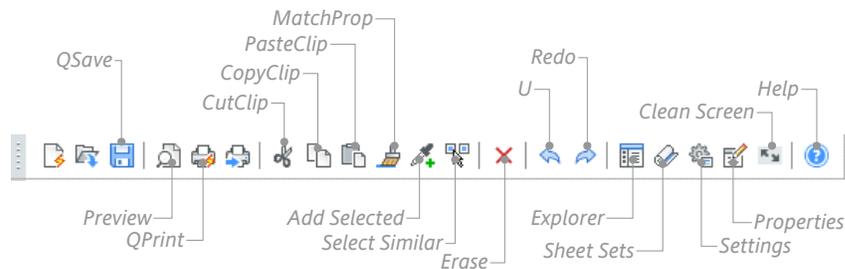
Keep the cursor over the icon, and then look down at the status bar at the very bottom of the BricsCAD window. It displays a one-sentence description of the button’s meaning.

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 most important one contains buttons with which you are probably already familiar from other Windows or Linux applications. It 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.

Save (QSave command) saves the current drawing.

Print Preview (Preview command) shows what the drawing will look before it is printed or plotted.

Print (QPrint command) 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.

Layer control for changing layers

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 removes entities from the drawing.

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

Redo redoes the last undo.

Drawing Explorer (Explorer command) 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.

Clean Screen (ClearScreenOn command) maximizes the drawing area by hiding most user interface elements

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

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



The extra-large icons are meant for very high-resolution computer screens, such as 4K. The size affects toolbar buttons only, and no other user interface element such as the ribbon.

Droplists

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

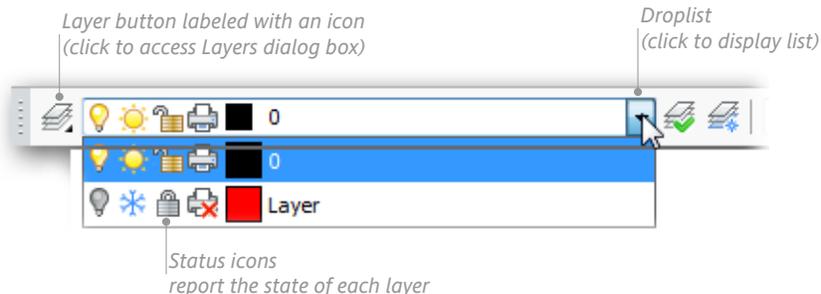


From left to right: Droplists access layer names and properties; colors; linetypes (patterns); and line weights (entity widths)

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.)

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:



Top: Layer settings turned on; **bottom:** settings turned off

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

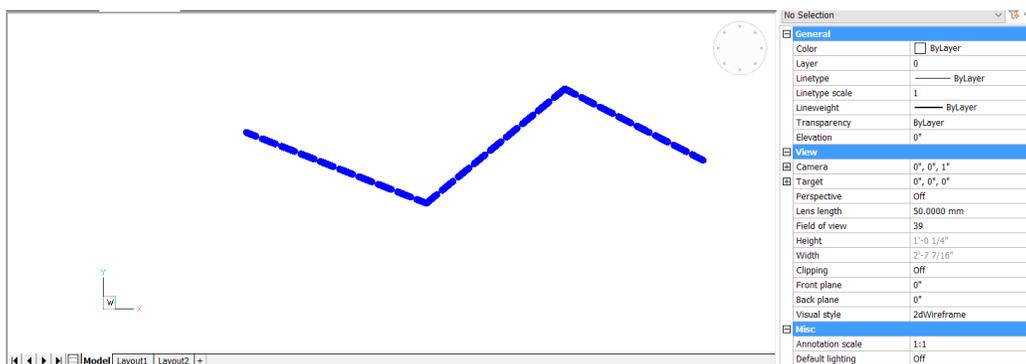
Square — specifies the color assigned to the layer (black in the figure above); click the color square to access the Colors dialog box, which lets you change the layer's color.

Name — specifies the name of the layer ("0" in the figure)

Changing Properties

The two big jobs in CAD are making and editing drawings, but following closely behind 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. In the figure below, I show a blue polyline with a dashed line pattern; its properties are displayed by the Properties panel.



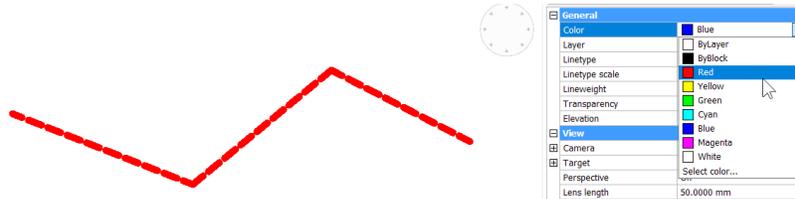
Here is how to use the Properties panel to view and change properties of the polyline:

When no entities are selected — 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 the Property panel in the

figure above.

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

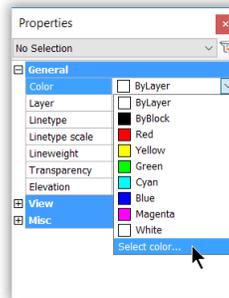
While an entity is selected — choose a different property from the Properties panel. The entity takes on the new property. For instance, select “Red” from the Color droplist, and the polyline turns red.



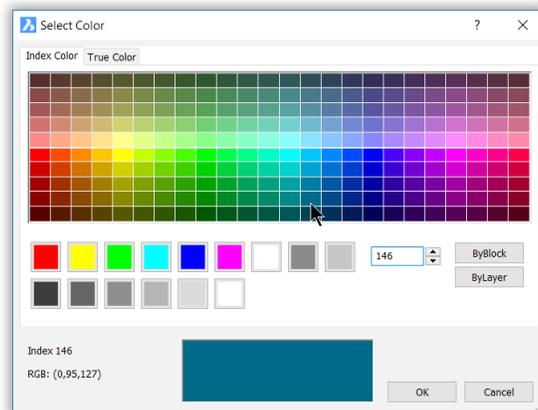
In the same way, you change other properties of the selected entity — elevation (distance in the z direction), transparency (level of see-through-ness), and even the coordinates of its position in the drawing. Some droplists in the Properties panel contain extra functions that are kind of hidden from view. Here is how to access them:

Adding Colors. Initially the **Color** droplist shows just nine colors. If you want more colors on the list, follow these steps:

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

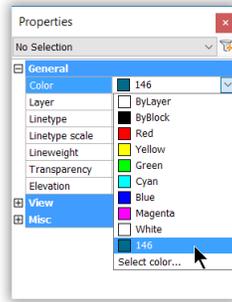


2. Notice the Select Color dialog box, Choose a color from one of 255 “index” colors (standard colors among BricsCAD and AutoCAD uses) or 17.6 million “true” colors (standard in the broader world of graphics).



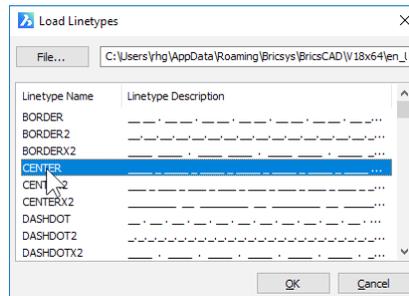
3. Click **OK**. Each color you choose from this dialog box is added to the droplist, so that you can reuse them in

the future.



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.

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

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

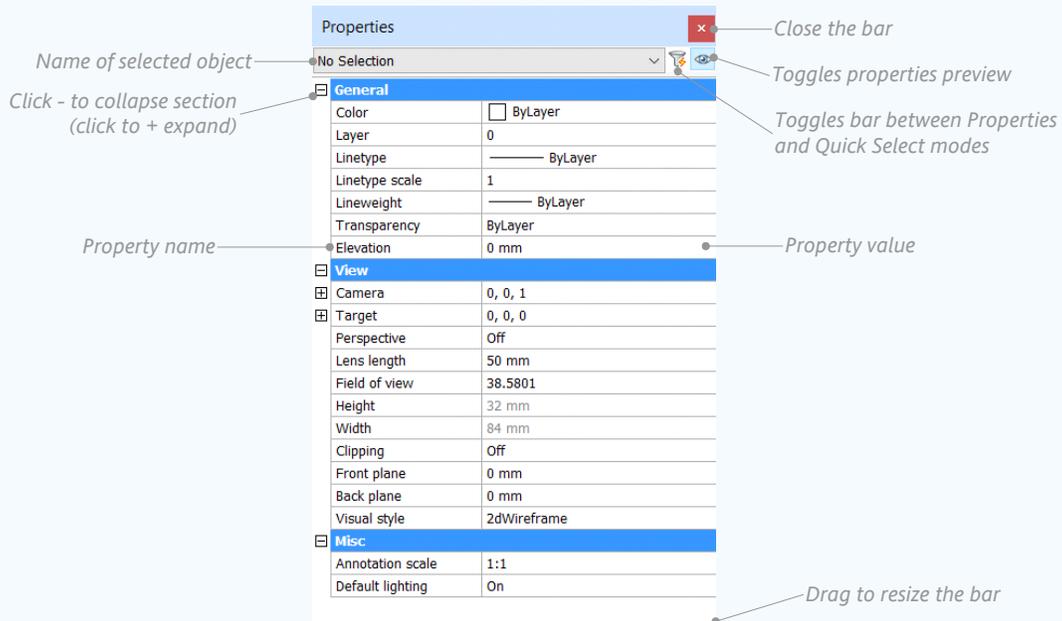
This is why you often see “ByLayer” in property droplists.

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. The actual list of panels you see depends on which edition of BricsCAD you are using. (The Classic version has the least; the Ultimate one the most.)

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.)

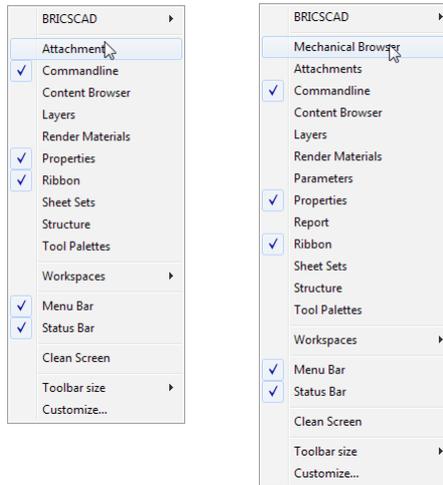


	Drag the title bar to relocate the panel on the screen
	Click the x button to close the panel
	Click the droplist to access other entities (if any)
	Click the + node to expand a section
	Click the - node to collapse a section
	Click on a property value to modify it
	Drag the edge of the panel to change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

SHORTCUT KEYSTROKES FOR PANELS

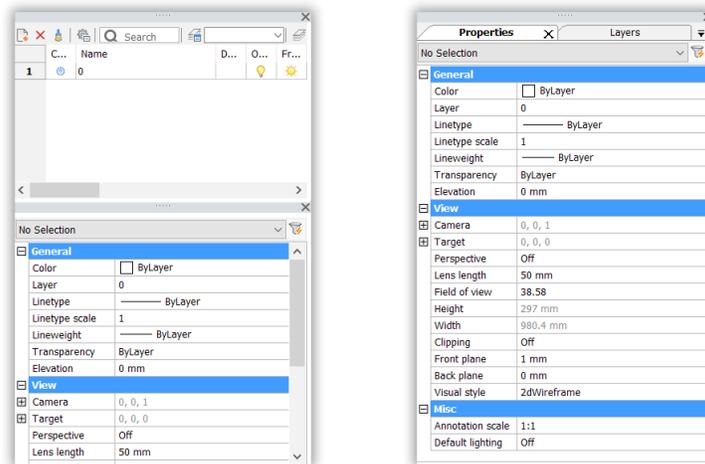
Several panels and bars can be opened and closed using shortcut keystrokes. Here is a summary of them:

Action	Command	Shortcut Keystroke
Open or close Properties panel	Properties	Ctrl+1
Open the Drawing Explorer	Explorer	Ctrl+2
Open or close Command bar	CommandLine	Ctrl+9



Left: Panels available in BricsCAD Classic; **right:** ...and in BricsCAD Ultimate

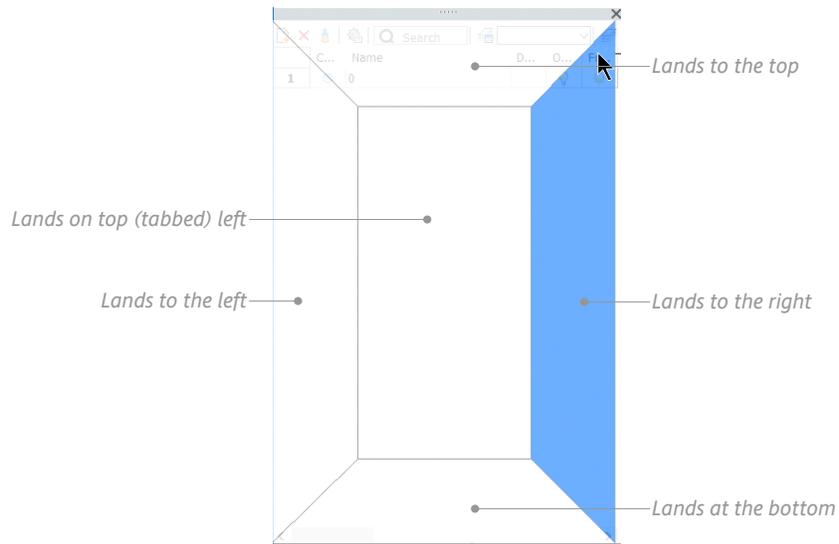
A blue check mark means the panel is open. Panels can be stacked on top of each other or else placed next to one another. Here is what two panels look like when one panel lands above another (shown at left) or the panel is stacked on top of another (shown at right):



- ▶ 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 move panels, 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:



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.

To keep a panel or toolbar from docking to a side of the drawing area, hold down the **Ctrl** key (**Cmd** for MacOS) while dragging the panel or toolbar.

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.

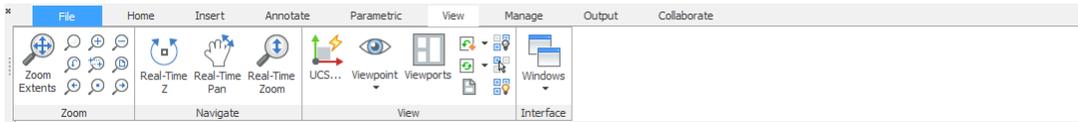
If you do not see the ribbon, enter the **Ribbon** command:

: ribbon

Shown below is the ribbon's standard Home tab. Other tab names include Insert and View.



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

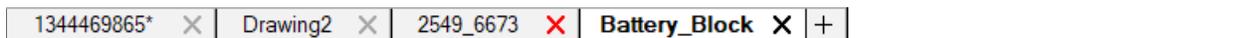


Because Bricsys wrote its own version of the ribbon interface, it is equally available on the Windows, Mac, and Linux versions, unlike all other CAD systems. You customize the ribbon through the **Customize** command.

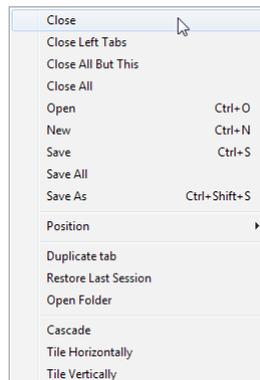
(The File “tab” is not a tab, but acts like a menu item to access file-related commands.)

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

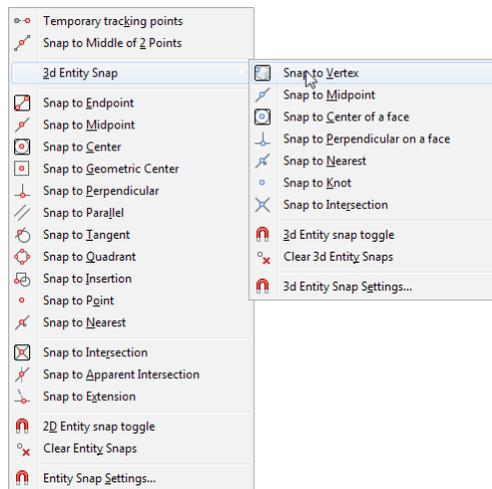
There are two variables that control drawing tabs. **DocTabPosition** places the tab at the top, bottom, left, or right of the drawing area. **ShowDocTabs** turns the tab row on and off.

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 above shows what happens when you hold down the **Shift** key and click the right mouse button: you access the entity snap modes, as describe later in this book.

QUAD CURSOR

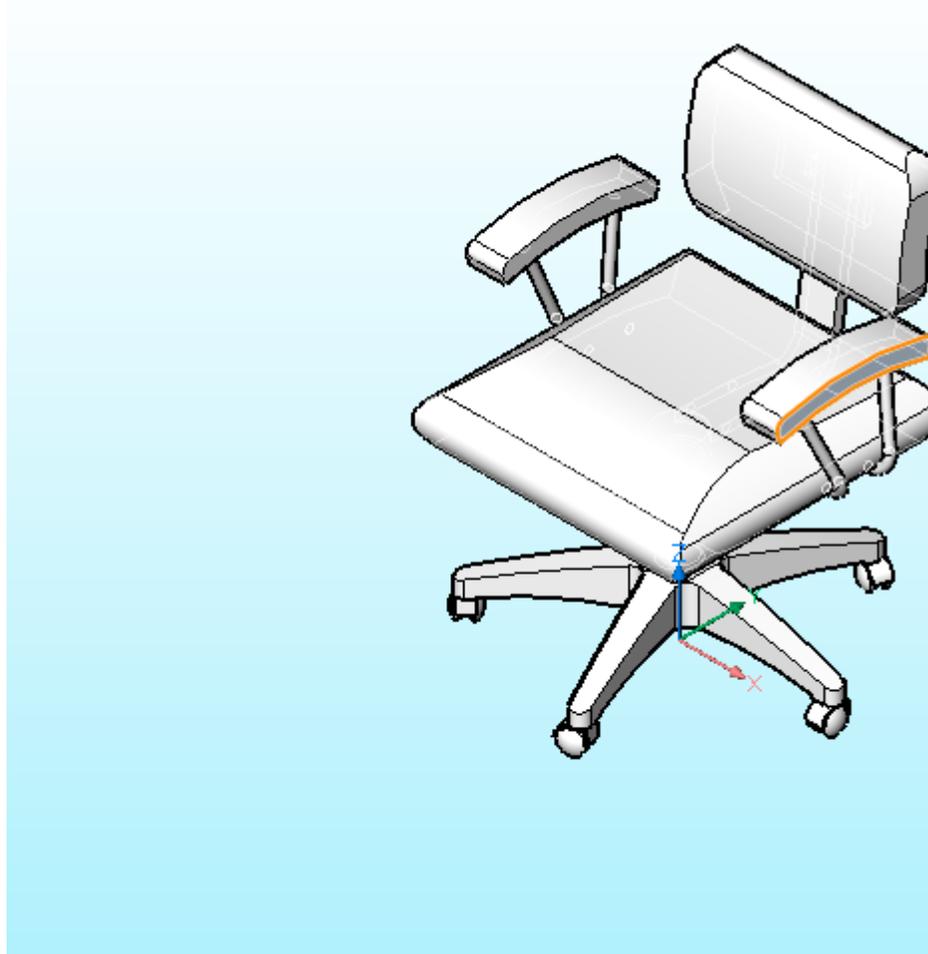
Status bar QUAD

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.

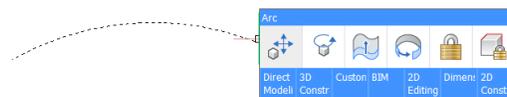
The Quad is normally not visible; most of the time, you see the standard tri-color crosshair cursor or arrowhead cursor. When you pass the cursor over an entity, however, the quad cursor appears, first as a single button; see figure below.



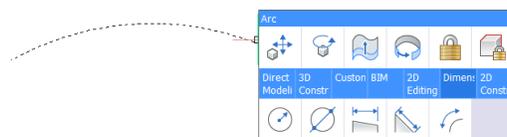


(If it does not appear, then turn it on by clicking **QUAD** on status bar or pressing function key **F12**.)

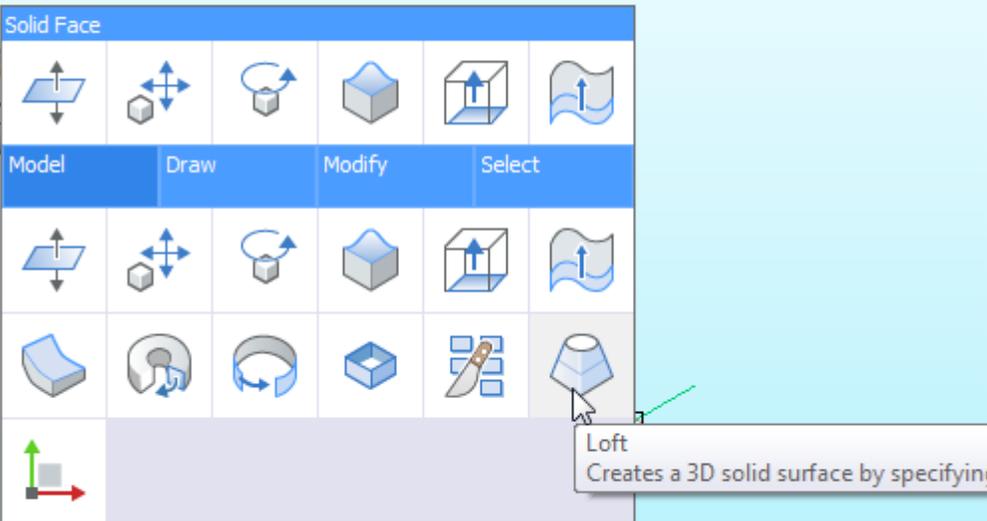
When you move the arrow cursor onto the sole button, the quad cursor expands to show additional buttons, usually for commands most commonly used with the nearest entity.



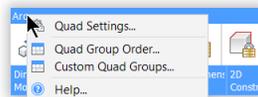
In addition, there are blue boxes for groups of additional buttons. Some groups are for common operations, while others are specific to the entity. To access the additional buttons, pass the cursor over a blue box. Click a button to execute its command.



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



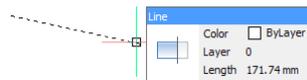
To customize the Quad, right-click it to access the following shortcut menu:



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.)

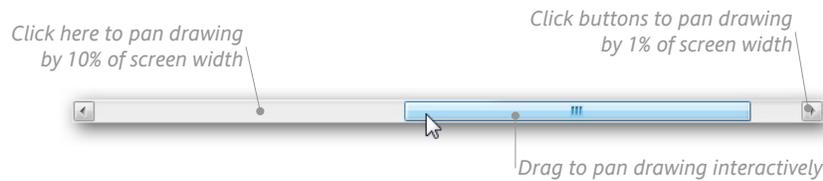
Hover the cursor over an entity, and the Quad displays some of the properties of it.



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

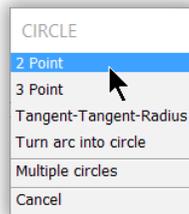
A scroll bar rests at the right edge of the drawing area. It lets you pan the drawing up and down. A second one is along the bottom of the drawing area; it pans left and right. Normally, they are turned off, but I find them handy. To turn on scroll bars, enter the **ScrollBar** command. Scroll bars are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



As a faster alternative, hold down the mouse's middle button (or roller wheel), 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.



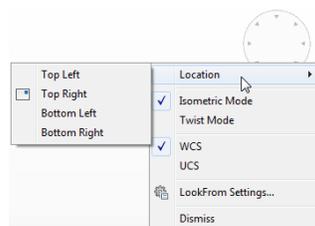
The green dot indicates the cursor position, kind of like a laser pointer:

To see the bottom view, 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.

Press the **Home** button on the keyboard to return the view to its normal position.

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>:
```

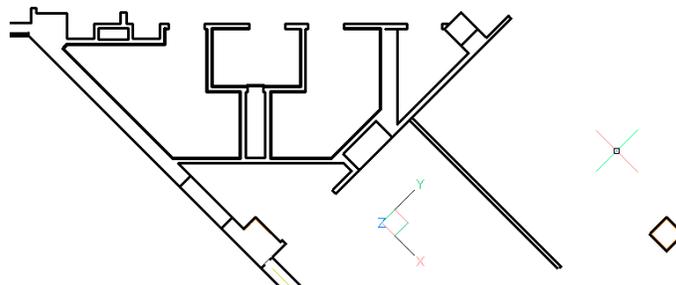
The **Settings** option opens the Settings dialog box at the LookFrom section. Here you can adjust the properties of the widget, such as its translucency and position. Of particular interest is the number of isometric viewpoints it can display, which is set through “Direction Mode” or the **LookFromDirectionMode**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

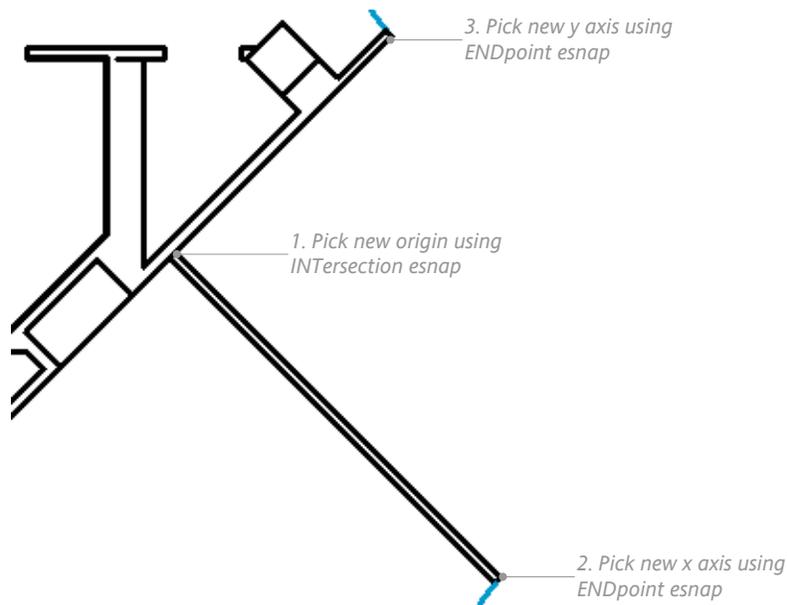
Static UCSs for 2D. In 2D drafting, for example, when we draw things at an angle, such as the angled wing of a building, then we use a USC. Changing the coordinate system to match the angle of the building rotates everything else as well. In the figure below, I rotated the UCS icon to match the angled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



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

1. The new origin point (0,0)
2. The new direction of the x axis (determines the angle)
3. The new direction of the y axis (determines the orientation)

From the 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

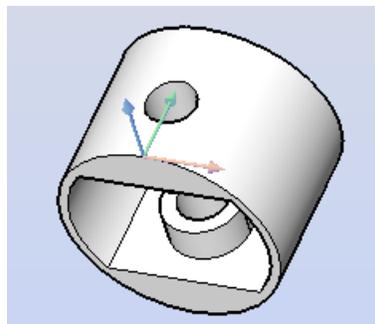
Specify origin of UCS or [Face/NAmed/Entity/Previous/View/X/Y/Z/Z Axis/Move/World]
<World>: (Pick a point like the intersection of two lines; BricsCAD turns on INTERsection entity snap automatically)

Point on X-axis/<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/<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 should give it a name. This lets you switch back to it quickly. Named UCSes are managed by the **ExpUcs** command.

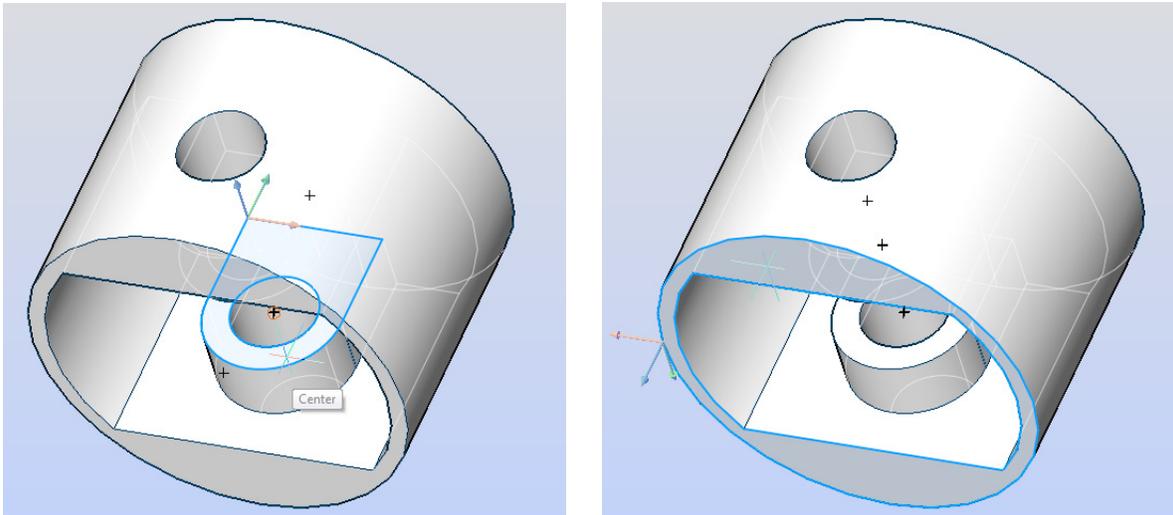
Dynamic UCSs for 3D. The same system works for manually changing the UCS in 3D modeling. Why would you want to change the UCS in 3D? Because CAD is still fundamentally 2D, even when in 3D. To draw on a face, the UCS needs to match the orientation in space of the face. BricsCAD, fortunately, can do this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



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:

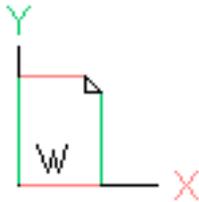
```
: text
Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: (Pick a face on which
to write)
```

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.

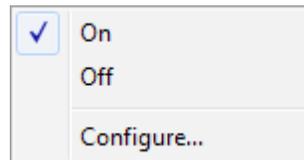
HOTKEY ASSISTANT

(NEW IN V19) Hotkey Assistants are interactive widgets that report options that might otherwise be unknown to users. (These were formally known as “Tips widgets.”) For example, the following widget appears during the Polysolid command.



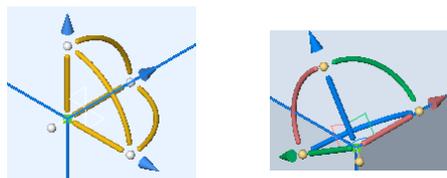
Notice that it shows several icons in a row. The **Ctrl** “button” reminds you to tap the **Ctrl** key to make the changes during the command. (The **Ctrl** button itself does nothing; neither does the **i** icon.) In this case, the next three icons explain the changes that occur each time you tap the key: the justification of the polysolid changes between left, centered, and right. To get a brief description of the purpose, pause the cursor over the Tip.

To dismiss the widget, click the small **x** at the right end. The display of the widget is toggled through the **HKA** button on the status bar. Right-click the button for options; the configure button leads you to a dialog box that lets you determine when the widget appears.



MANIPULATOR WIDGET

To manipulate entities, BricsCAD provides the manipulator widget. When the widget is turned on, it attaches to the entity you select. You drag the arrowheads or bars of the widget to rotate, move, mirror, or scale 2D and 3D entities along the x, y, or z axes or xy, xz, or zy planes. An alternative to dragging, you can enter values via dynamic dimensions.



Left: Manipulator in default colors; **right:** ...and in classic colors

The **Manipulator** variable determines when the manipulator widget is available:

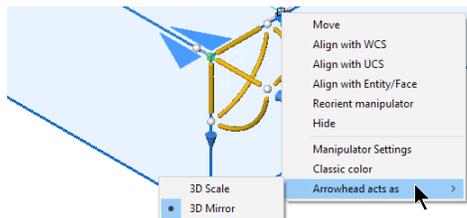
Manipulator	Meaning
0	Not displayed (default)
1	Display manipulator when entities are selected
2	Display manipulator when left mouse button is pressed longer than 250msec

Manipulator Action	Action
Drag a blue arrowhead	Scale (resize) or Mirror (default)
Drag a gold bar	Move
Hold Ctrl while dragging bar	Copy
Drag a yellow arcs	Rotate

You can move the widget about the entity to edit different parts of it:

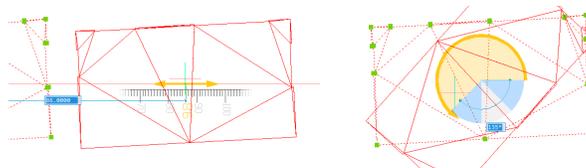
- ▶ **Relocate** the widget by dragging the white ball (found nearest to the origin); entity snaps are respected
- ▶ **Twist** the widget by dragging one of the three white balls adjacent to each arrowhead

Right-click the widget for the following options:



- ▶ **Move** moves the widget freely to another location
- ▶ **Align with WCS** or **UCS** or **Entity/Face** aligns the arms of the widget with the x,y,z-axes of the world, or a user-defined coordinate system, or to the nearest face of an entity.
- ▶ **Reorient manipulator** acts like reorienting the UCS icon.
- ▶ **Hide** hides the widget.
- ▶ **Manipulator Settings** opens the Settings dialog box at the Manipulator section.
- ▶ **Classic color** changes the arm colors so that the x arm is red, the y one is green, and z is blue.
- ▶ **Arrowhead Acts As** switches between 3D scaling (indicated by the double blue arrowhead) and 3D mirroring (indicated by the mirrored blue triangles).
- ▶

(NEW IN V19) As you drag the manipulator's arm, a ruler shows you the distance interactively. Or, if you are changing an angle, then a protractor illustrates increments of 45 degrees.



Left: Ruler showing the drag distance; right: protractor showing the angle rotation

Below the Drawing Area

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

Layout tabs — switch between model space and layouts quickly

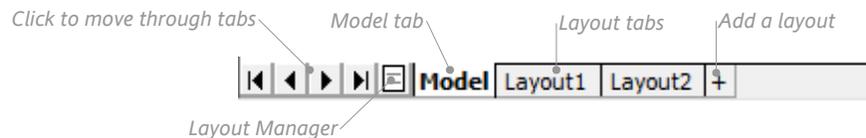
Command bar — accepts typed input and displays prompts; see Chapter 1 for details

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 as it is plotted. They let you position its location on the paper, adding drawing borders, specific views, and/or specifying shades of gray instead of colors.

Click the + plus sign to add more tabs.

The arrow buttons let you see more of them there are too many tabs to fit along the bottom of BricsCAD.

Layouts bar controls the display of layouts; also holds the horizontal scroll bar:



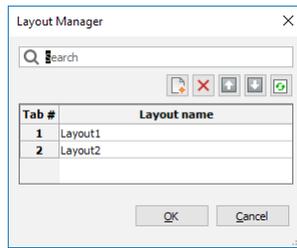
Command bar is where you enter commands and read prompts:



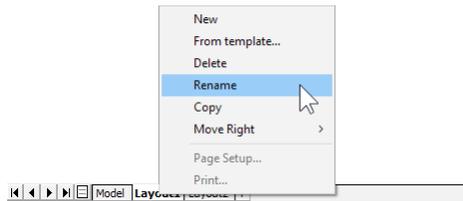
Status bar shows help text, reports the status of the drawing, and changes settings:



Click the **Layout Manager** button to display the Layout Manager dialog box, which lets you create, name, and organize more than layout at a time.



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

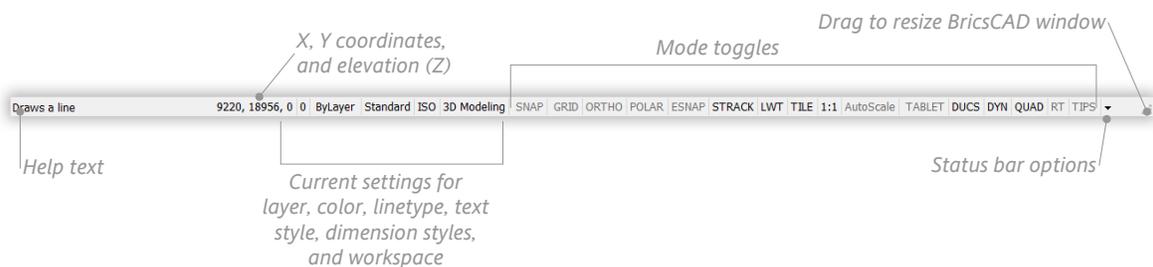
The scroll bar is described earlier in this chapter.

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

HKA — toggles the hotkey assistant widget

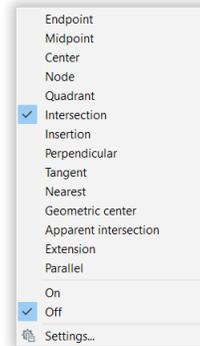
None — selects a GIS coordinate system

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,

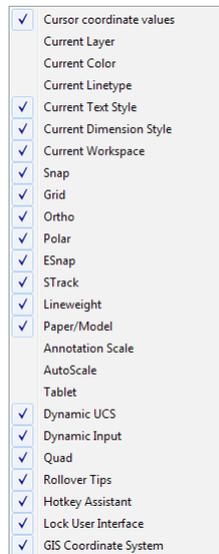


Settings is a shortcut to the dialog box that regulates the mode. For example, to change the settings for the grid, right-click the GRID button and then select Settings; BricsCAD displays the Snap/Grid section of the Settings dialog box.

When you right-click a few buttons, they present a longer list of options. For example, right-click **ENSAP** to choose entity snaps.



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.

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

- Step 1: **Start** new, blank drawings
- Step 2: **Name** the drawing file
- 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, you learn how to set up new drawings, save your work to disk, and then exit BricsCAD.

IN THIS CHAPTER

- Preparing a drawing for first-time use with a wizard
- Specifying units, angle formats, and drawing settings
- Understanding how layers organize drawings
- Saving drawings
- 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

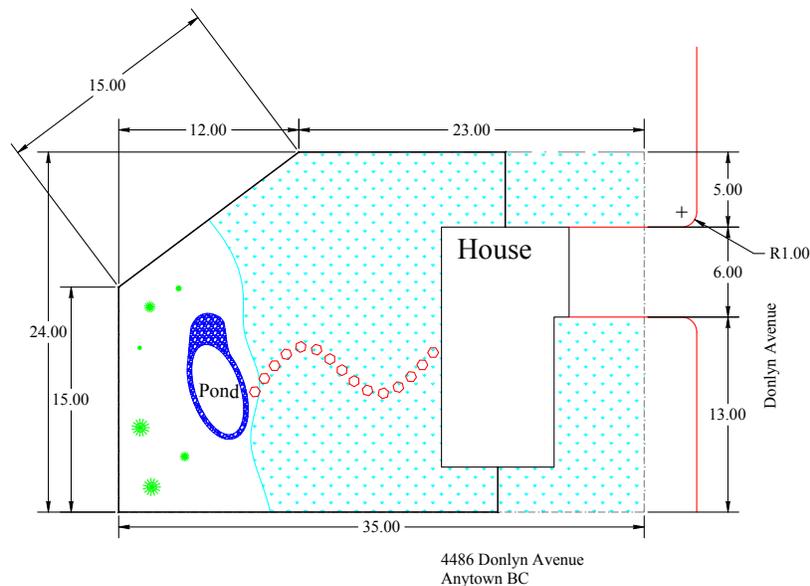
' or ft	Feet
" or in	Inches (12 inches per foot)
ANSI	American National Standards Institute
BAK	Backups of BricsCAD drawing files
DWT	Portion of file names that identify them as template files
DWG	Portion of file names that identify them as BricsCAD drawings
mm	Millimeters (1000mm per meter)
m	Meters

NEW COMMANDS

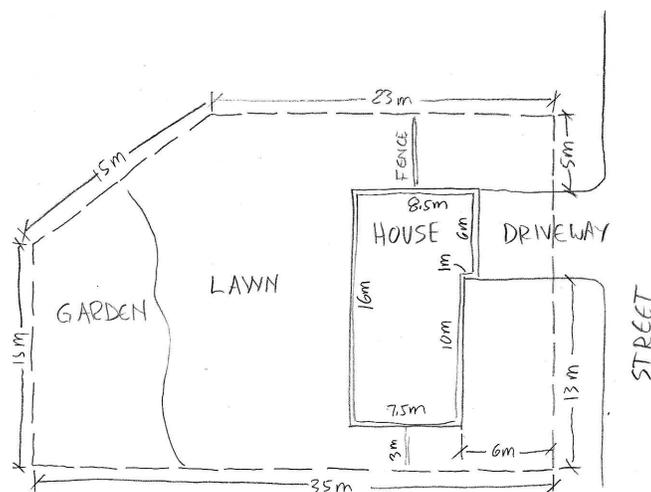
Command	Alias	Menu Bar	Ribbon Tab
Layer	la	Settings Layers	Home Layers Layers
Limits	..	Settings Drawing Limits	..
NewWiz	ddnew	File New Wizard	..
Settings	options	Settings Settings	Settings Settings Settings
Save	Ctrl+S	File Save	File Save
SaveAs	..	File Save As	..

Before You Begin

To learn how to draw with BricsCAD, you will work on a landscape plan, creating and modifying the drawing of a yard around a house. By the end of chapter 7, the drawing will look like this one:



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



The tutorial in this book will be done in metric units (meters), but for North American readers I'll include imperial units (feet and inches). The completed *Yard.dwg* file can be downloaded from <https://www.dropbox.com/s/ft6gzw3r408ndgd/Inside-BricsCAD-Tutorial-Files.zip>. If you draw with imperial units, then use *Imp-Yard.dwg* found inside the ZIP file. The ZIP file also holds the drawing as it appears at the end of each chapter.

STARTING A NEW DRAWING

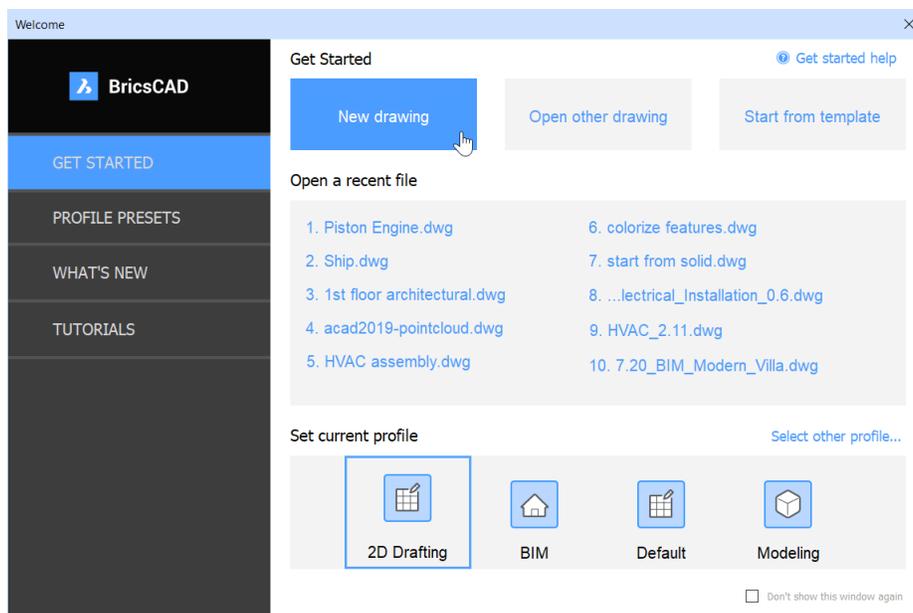
Command	NewWiz
Menu	File New Wizard
Ribbon	Home File New Wizard

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

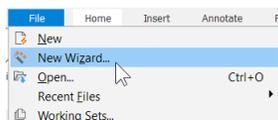
- ▶ From *scratch* — with a blank drawing
- ▶ Or, with a *template file* — that presets certain aspects of new drawings
- ▶ Or, enlist the services of a software wizard — this how you will start the first tutorial

In this chapter, you create the new drawing with the assistance of a *wizard*. The “wizard” is a series of dialog boxes that take you through the steps needed to set up new drawings. They prompt you to select the drawing units (such as Imperial or metric), turn on the grid, and so on. The wizard is meant to help you get through the initial few steps in preparing new drawings, which can be confusing to new users faced with a blank screen.

1. If BricsCAD is not running, start it now by double-clicking its icon on the desktop
2. Notice the Welcome dialog box. Choose “2D Drafting” as the current profile, and then click **New Drawing**.

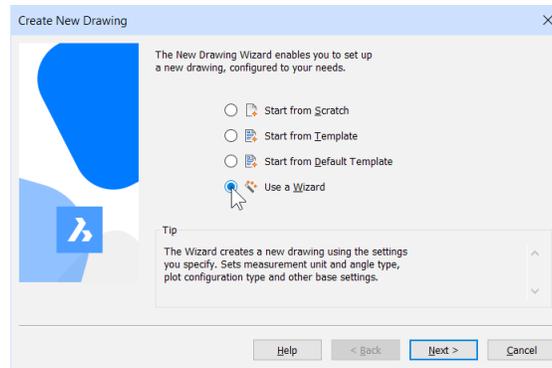


3. Start the wizard by taking one of the following steps:
 - ▶ In the ribbon's **Home** tab, click **New** and then **New Wizard**



- ▶ Or, from the menu bar, choose **File**, and then **New Wizard**
 - ▶ Or, at the command line, enter the **NewWiz** command
- : **newwiz**

Notice the Create New Drawing dialog box:



The dialog box supplies you with options for starting new drawings:

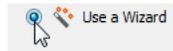
Start from Scratch — starts empty new drawings, using either imperial or metric units

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*

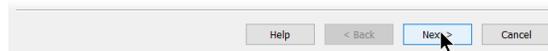
Start from Default Template — starts new drawings with the template file specified by the **BaseFile** system variable; the current default is *default-mm.dwt*

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 its label.



5. Click **Next**.

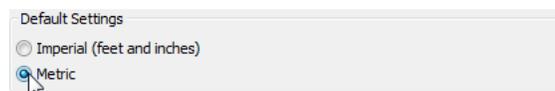


The wizard now takes you through a series of dialog boxes that ask you to specify the units, format of angles, and other drawing settings. I’ll explain to you what each one means.

Default Settings

First, choose the units. Well, the wizard calls them the “default settings.” This is because so many settings in drawings are affected by the type of units. Your choices are *metric* (decimal) or *Imperial* (feet and inches). Here is how to decide which units to choose:

- ▶ If you measured the yard in meters and centimeters, then click the radio button next to **Metric**
 - ▶ If you measured the yard in feet and inches, then click the radio button next to **Imperial (feet and inches)**
6. If you are following along with my tutorial drawing, then choose **Metric**, as shown below.

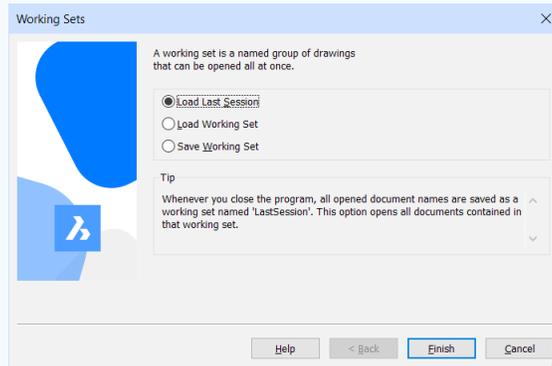


7. Click **Next**.

HOW 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.

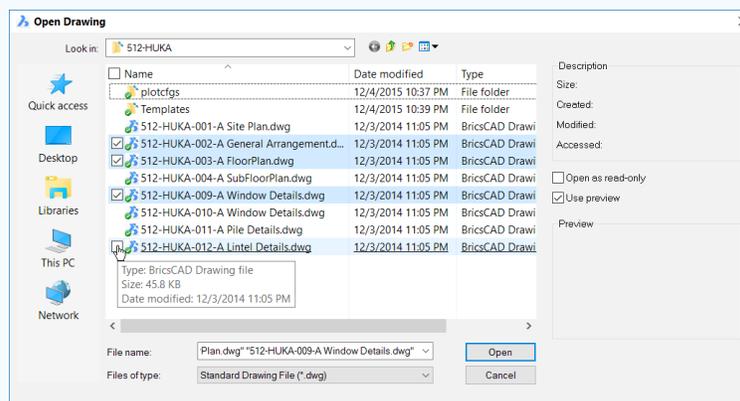


OPENING MORE THAN ONE DRAWING

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



STARTING FROM SCRATCH

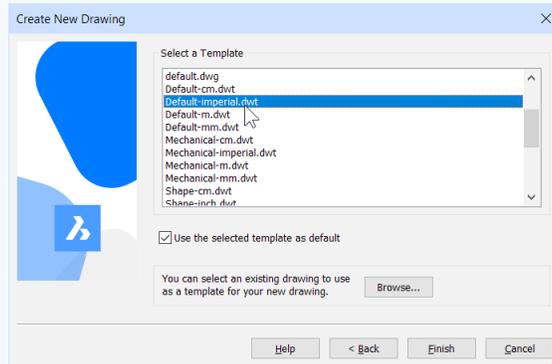
New “empty” drawings employ default settings for all variables, which are read from the following DWT template files:

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:

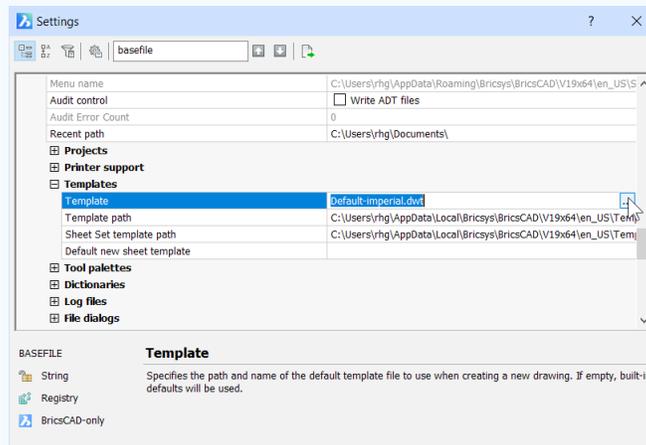


You can create your own templates: edit the current drawing, and then use the **SaveAs** command. Choose “Drawing Template (*.dwt)” from the **Save As Type** droplist.

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.

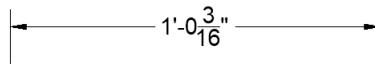


USING A WIZARD

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

SELECTING THE LINEAR UNITS

Drafting uses *linear measurements* for things like lengths and areas, as illustrated by the dimension below. The other kind of measurement is *angular measurements* for angles.



Linear measurements can be displayed in several different formats. Examine closely at the sample measurements shown in the table below to understand how BricsCAD displays each. Note that a *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 (default)	Units and decimal units	2128.4449
Scientific	Units, decimal units, and exponents	2.1284E+03

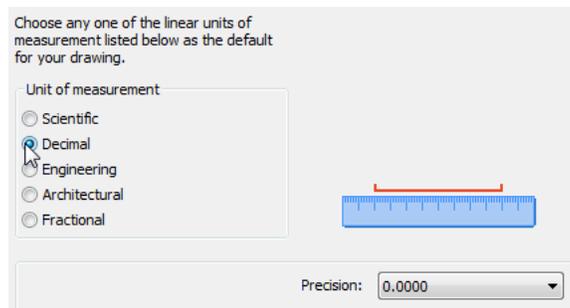
The default is called “Decimal,” which is often used for metric measurements, such as millimeters and meters. BricsCAD displays metric measurement as *unitless* decimals, such as 2128.4449 2128.4449; the “mm” for millimeter or “m” for meters is not displayed.

(How do you know if 2128.4449 displayed by BricsCAD is millimeters or meters? It’s whatever you start drawing with. If the first thing you draw is 2128.4449 meters long, then everything else is in meters, too.)

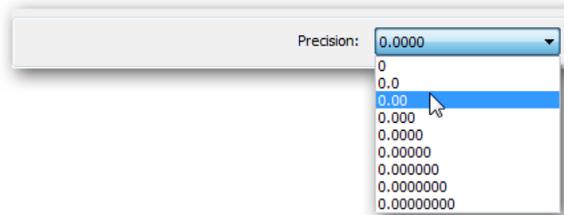
Once you select a style of unit, BricsCAD *displays* all measurements in that format. But don’t worry! You can always switch the measurement format to something else with the **Units** command.

(The reason you can switch between units is because internally BricsCAD actually ignores the type of units. It keeps track of measurements with unitless real numbers accurate to 14 decimal places, and then converts them on the fly to the units we want, for the sake of us humans.)

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



9. This drawing is measured in meters, to the nearest centimeter (0.01m). 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 in feet to the nearest 1/4-inch — accurate enough for this project. Select **Architectural** units, and then from the Precision droplist choose **0'-0 1/4"**.)

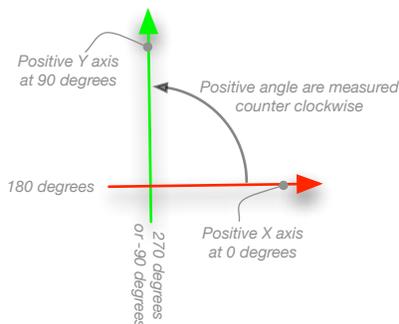
10. Click **Next**. As we see next, BricsCAD displays the set of options for measuring and displaying angles.

You are free to enter distances more accurately than a centimeter, such as 0.4441 because BricsCAD remembers distances to full accuracy. When BricsCAD displays coordinates, however, it will round them off to the nearest 0.01 units ("0.44"), because we told it to in step 9.

SELECTING THE ANGLE STYLE

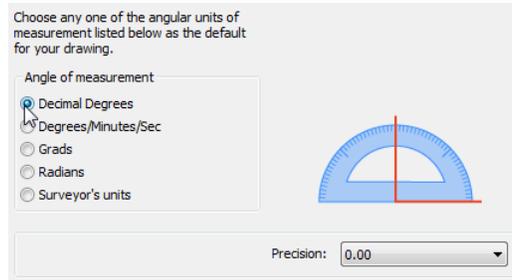
Before we choose the angle measurement system, I must mention some facts about angles:

- ▶ In CAD, angles are usually measured starting at the x-axis, which is at 0 degrees; see figure below
- ▶ Angles are usually measured counter-clockwise (backward from how a clock moves) from that positive x axis
- ▶ 90 degrees is at the positive y axis
- ▶ BricsCAD shows the x axis in red, and the y axis in green



The elements of axes and angles

The Angle dialog box is for choosing the formats of angular units that BricsCAD should display.



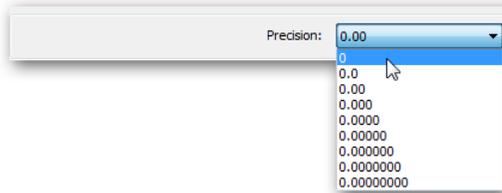
The options available are as listed below:

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

There are 400 grads (used in Germany) and 2π radians (or 6.2832 radians) in a 360-degree circle.

If we were *real* land surveyors, we would use the Surveyor format. But we're not, and so we will stick with familiar decimal degrees. (Again, you can choose the change the angular format at any time 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.

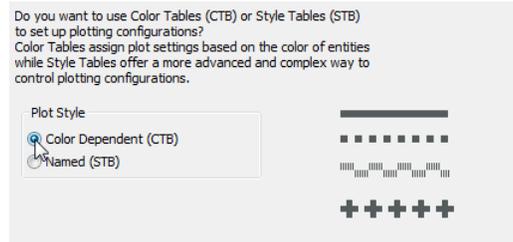


13. Click **Next**.

CHOOSING THE PLOT STYLE

When it comes time to printing drawings, they can be prepared by BricsCAD for the printer using one of two methods. One method uses *color tables* (a.k.a. “color dependent” styles or CTB), while the other method uses *style tables* (a.k.a. “named” styles or STB). They are, unhappily, mutually exclusive, meaning that you can choose only one, ahead of time.

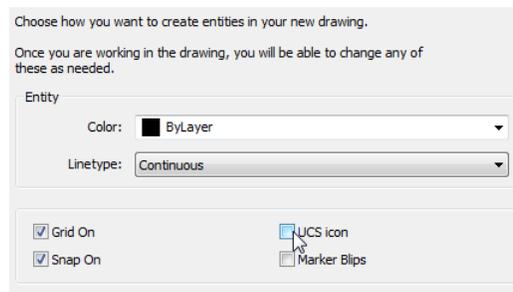
- Right now, all you need to know is that neither color tables nor style tables really matter when plotting regular 2D drawings on printers. So, ensure that the **Color Dependent (CTB)** option is selected, as it is the easier of the two to work with.



- Click **Next**.

SETTING ENTITY PROPERTIES

You now come to the final dialog box of the wizard. Here you set the *defaults* of some entity properties (a.k.a. “objects”). Default means the “initial setting.” Initially, 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, for instance, then all entities on that layer appear red; change the layer’s color to green and the entities follow suit.

In this dialog box, you could overrule the default properties. But you won’t, because changing colors and linetypes is best left for later, and is done with the Layers dialog box.

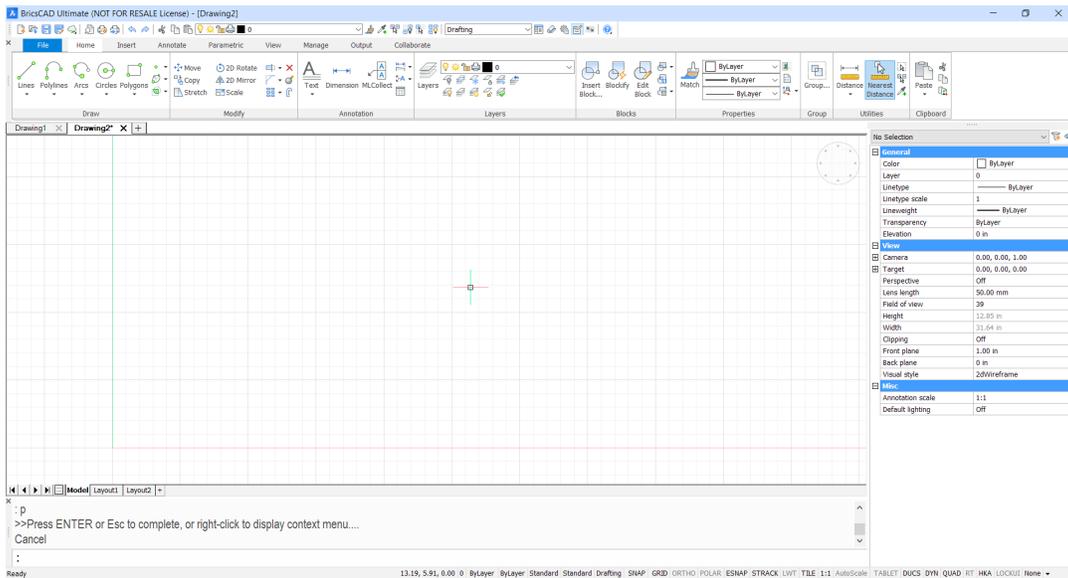
- The only change to make here is to turn off the UCS icon, because it is unhelpful (by being visually intrusive) to the tutorial drawing.
- See figure above for how settings should look. Click **Finish** to close the dialog box.

FINISHING THE WIZARD

When the wizard is done, the final dialog box closes, and you get to see BricsCAD’s drawing area. It looks a little bit different from before: there is an array of lines in the drawing called “grid” lines. One grid line that goes horizontally through the origin (at 0,0) is red and a vertical one 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. 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.



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

Additional Important Settings

The wizard, unfortunately, does not set up everything you need in new drawings, and so in the rest of this lesson you use commands to set other aspects, 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 aspect means.

SETTING DRAWING LIMITS

Command	Limits
Menu Bar	Settings Drawing Limits
Toolbar	Settings

There is no limit to the size of drawings you can create with BricsCAD. If you wanted, you could draw the entire solar system — full size! Indeed, an early AutoCAD sample drawing showed this is possible: from the orbit of Pluto we can 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 size of a drawing is severely constrained by the size of the paper the printer handles, 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 visually is by setting the limits. This is done with the **Limits** command.

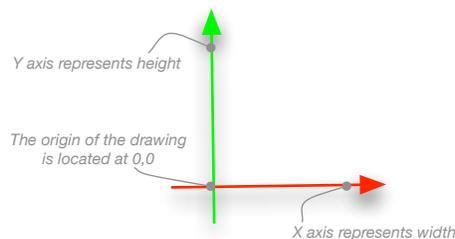
Limits show you the nominal area of a drawing, “nominal” because the drawing can be much larger, of course. Limits perform two more useful functions:

- ▶ 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 what the limits should be, and then set them:

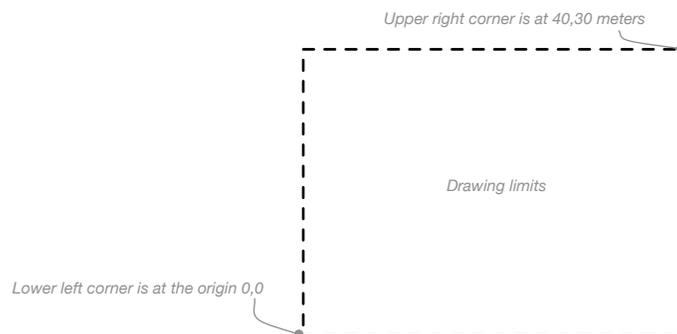
1. Examine the size of the yard in the sketch. After leaving a bit of “breathing room” around the plan, the drawing will need about 40m (130 feet) of width and 30m (100 feet) of height.
 - ▶ **Width** is represented by the **x** coordinate, which sometimes is usually in red
 - ▶ **Height** by the **y** 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. Pressing the spacebar is a handy 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 is actually 10'-10.

7. I find it handy to restrict the lines of the grid to the extents of the limits. To do this, I change the value of the **GridDisplay** system variable to 2 like this:

: **griddisplay**

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

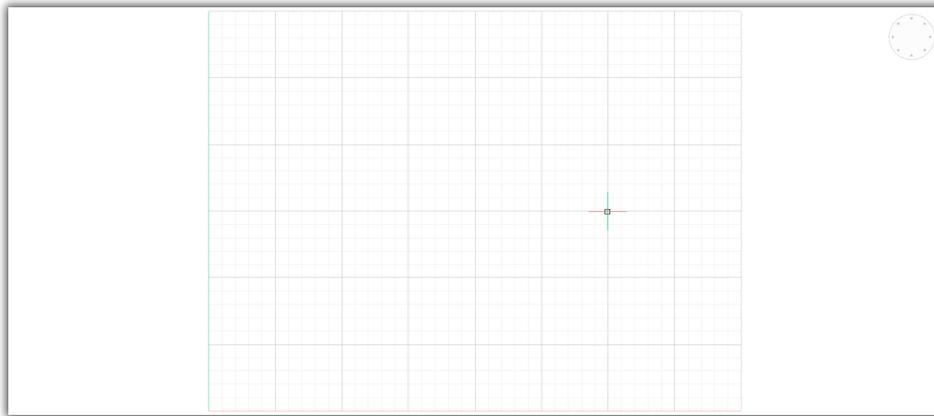
The 2 tells BricsCAD to crop the grid at the limits. (A 0 tells BricsCAD to turn off the grid, while other values tell BricsCAD to treat the grid in other ways.)

8. 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 [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/Object] <Scale (nX/nXP)>:**a**

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



The grid is now constrained to the area specified by the limits. Grids lines are shown faintly 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 SETTINGS

Command	Settings
Aliases	SE, options, ddrmodes, rm, dsettings, ddsetvar
Menu Bar	Settings Settings
Ribbon	Settings Settings Settings
Toolbar	Settings

Some of the values you entered at the command prompt can be changed through the Settings dialog box. This dialog box reports on the values of all *variables*. “Variables” hold settings that control and report on nearly all aspects of BricsCAD — and there are over 1,000 of them! I list them all in Appendix B. Variables are also known as “system variables,” or “sysvars” for short.

For example, **GridDisplay** is the name of a variable that tells BricsCAD how to display the grid. This variable has four settings, each represented by a number (technically, an integer). Here is what the value of the GridDisplay variable controls:

GridDisplay	Meaning
0	Turns off the display of the grid
1 (default)	Display grid beyond the area of the limits
2 (default)	Display adaptive grid
4	Allow sub-divisions below grid spacing
8	Follow dynamic UCS

The numbers 1 through 8 can be added together to turn on two or more options. 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)

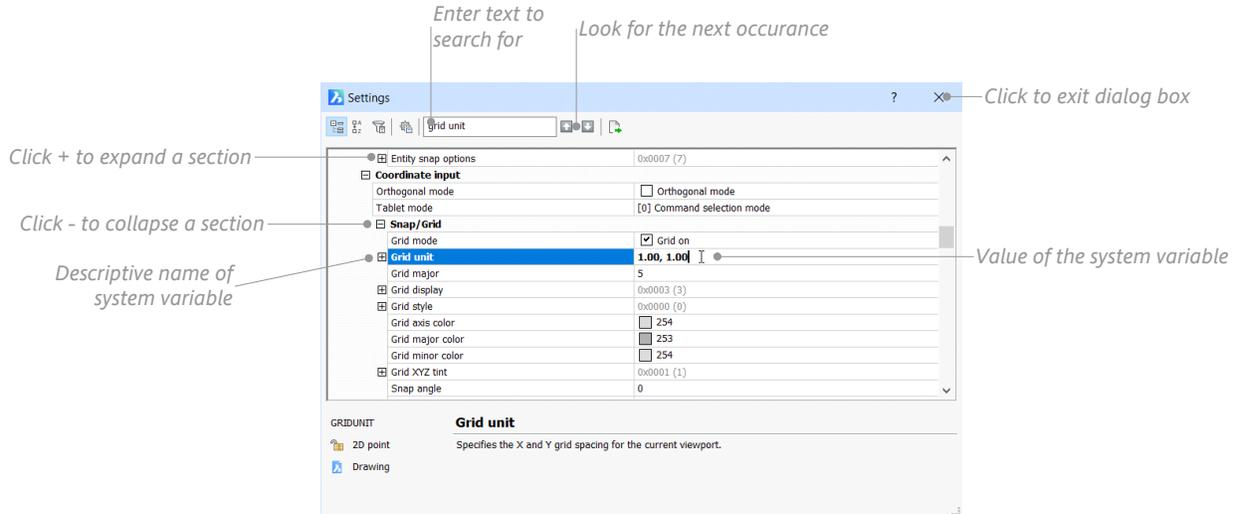
: limmax
New value for LIMMAX <40,30>: (Press Enter)
```

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

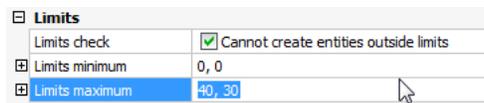
Notice the Settings dialog box.



2. This dialog box contains hundreds of settings, and so the best way to access one of them is to use the **Find** function, as follows:

- a. In the **Find** field, enter “limits”
- 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. Make changes to settings. When you change a setting, its text become **boldfaced**.
4. When you are done, just click the red **X**. There is no “Close” or “OK” button to dismiss this dialog box.



When the **Find** field turns orange, it means that there is no setting by that name. This occurs when you misspell the name, or when a variable of that name does not exist in BricsCAD.

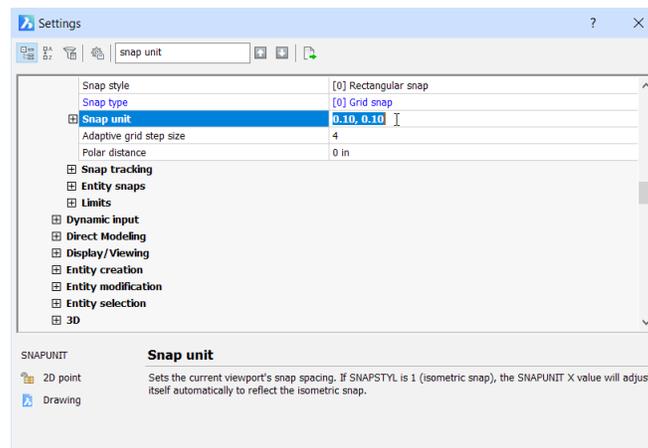
CHANGING THE SNAP AND GRID SPACING

One great advantage to drawing with CAD is that software permits you to draw very accurately. BricsCAD has several functions that help you draw with perfect accuracy. One you've already met, the grid. It helps you visualize distances.

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

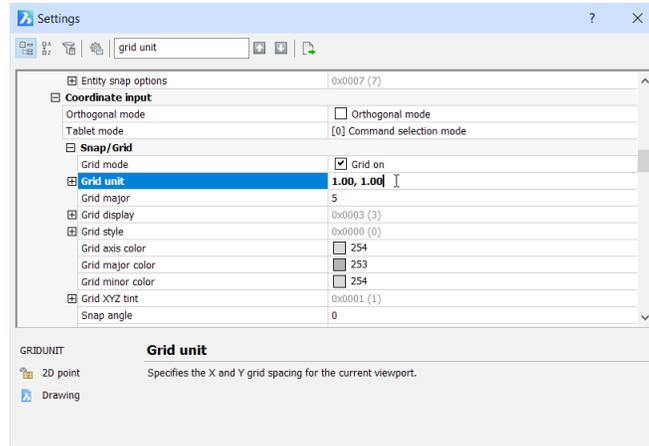
Recall that during one of the wizard's steps, you turned on snap mode. The default (initial) spacing of 0.5 meters is, however, too wide. Follow these steps to change the snap and grid spacing to 0.1 meters, which is more suitable for our drawing:

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**. See figure below.
2. Notice that the listing jumps to the **Snap Unit** field.
3. Change the X and Y values of **Snap Unit** to **0.1, 0.1** to represent 1 decimeter in each direction of x and y (or **1",1"** in imperial units). This is the distance that the cursor moves. Leave other snap settings at their default values, such as Snap Angle = 0.

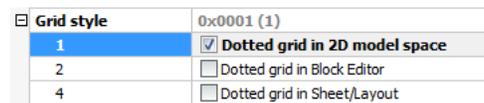


4. Earlier, the wizard had turned on the grid as a visual guide. The grid is meant to guide you; the default spacing of 10 units is perhaps too far apart. (BricsCAD will in fact not display the grid when its lines or dots are too closely spaced.)

Change the spacing to 1 meter by changing the value of **Grid Unit** to **1,1**.



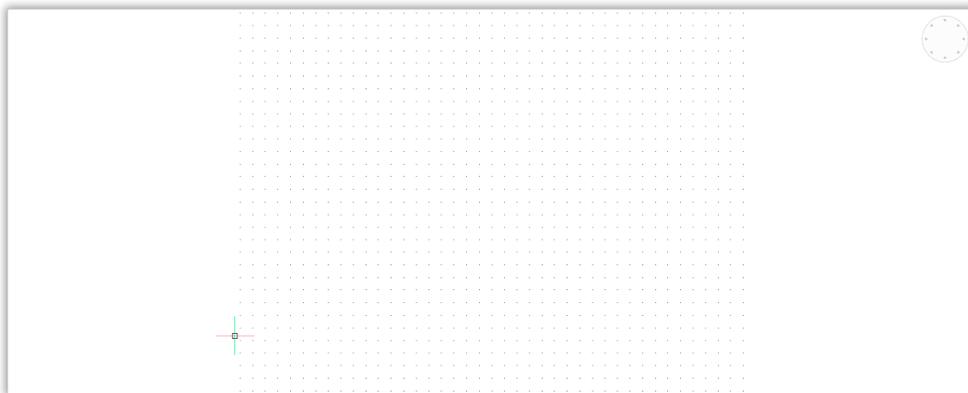
5. Finally, change the *type* of grid from lines to dots. Grid lines are useful in 3D modeling, but tend to cover up lines in 2D drafting, and so I prefer the more discrete dots. Here is how to do this:
 - a. Click the **+** node next to **Grid Style**.
 - b. Turn on the **Dotted Grid is 2D Model Space**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



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

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.



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
Alias	LA
Menu Bar	Settings Layers Tools Drawing Explorer Layers
Ribbon	Home Layers Layers
Toolbar	Settings Layers

If you ever worked with *overlay drafting*, then you may 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 medium made of plastic. The electrical plan would be drawn on another sheet, the structural on a third. Since the Mylar is transparent, drafters overlay two or three of the drawings to create a single blueprint — depending on how much detail was needed to be shown.

In CAD, *layers* operate in a similar manner. We draw different parts of drawings on different layers. The base is on one layer, electrical on another, and so on. Then we turn layers off and on to display the drawing with different amounts of detail. For example, the electrical contractor is usually interested in seeing the base plan layer with only the electrical layer.

While it is possible to create hundreds or even thousands of layers in drawings, it is more common to work with a few dozen layers; in this book, we work with a half-dozen layers to segregate the text from the lines showing the yard, the roads, and so on.

A further advantage to CAD layers is that they apply global properties to entities. For example, when a layer is green, then all entities assigned to that layer are colored green. Changed the color of the layer, such as to brown, and the color of all entities assigned to that layer change to brown instantly.

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. BricsCAD has commands that let you move entities to other layers, should that be necessary.

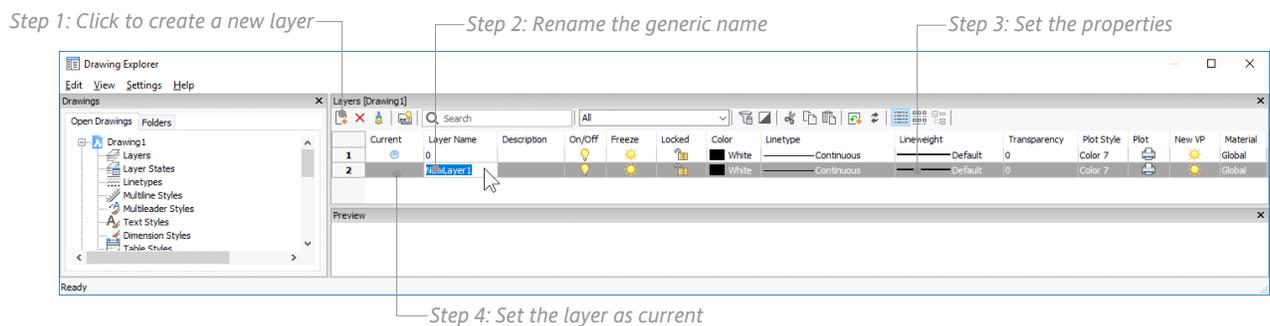
NAMING LAYERS

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

1. Enter the **Layer** command:

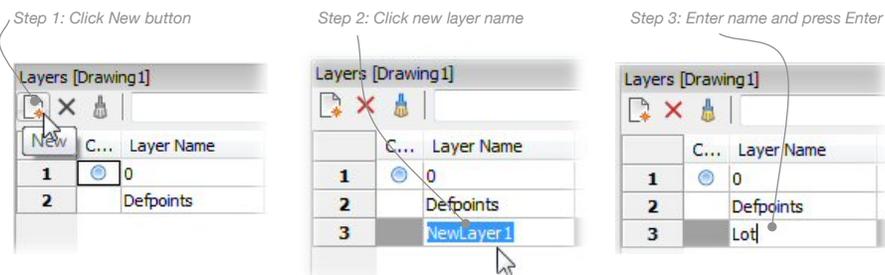
: layer

Notice the Drawing Explorer dialog box. This dialog box lets you control nearly all aspects of layers in drawings.



The drawing already has one layer: 0. Every new BricsCAD drawing has that layer "0", which you can never erase. Layer 0 has special properties that affect the creation of blocks, as discussed later.

2. Follow these steps to create and name a layer:
 - a. Click the **New Layer** button. Notice that BricsCAD creates a new layer called "NewLayer1." See figure below.
 - 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**.



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	Description	On/Off	Freeze	Locked	Color	Linetype	Lineweight	Transp...	Plot Style	Plot	New VP	Material
1		0					White	Continuous	Default	0	Color 7			Global
2		NewLayer1					Red	Continuous	1.00 mm	90	Color 1			Global
3		NewLayer2					Green	Continuous	0.40 mm	50	Color 3			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.)

Current — 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.

Description — describes the purpose or content of the layer; optional.

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

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 2.11mm (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.

If some of the drawing does not plot, check the setting of the **Plot** property.

Transparency — specifies the translucency of the layer; the default of 0 means all entities are displayed opaque.

New VP — specifies whether the status of new viewports, visible or frozen.

Material — specifies the material used during renderings.

...continued

...continued

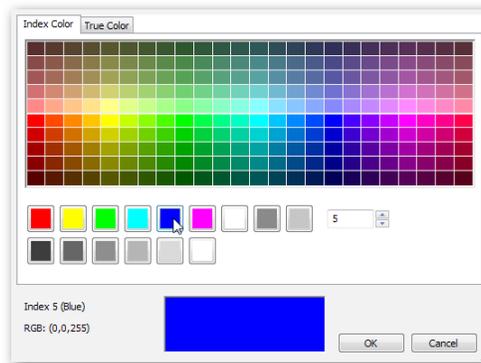
LAYOUT MODE PROPERTIES

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 Transparency, and VP Plot Style — specifies the color, linetype, lineweight, transparency, and plot style for the layer in paper space viewports

- b. Notice that the Select Color dialog box appears. It displays 255 colors: which one to choose?

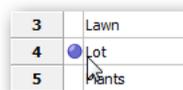


Above the row of black and gray squares is the row of BricsCAD’s “standard colors.” Red, yellow, green, and so on are the ones used most-commonly. Select the dark blue square. Notice that the number “5” appears in the **Color** text box, because this is color number 5, according to the color-numbering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

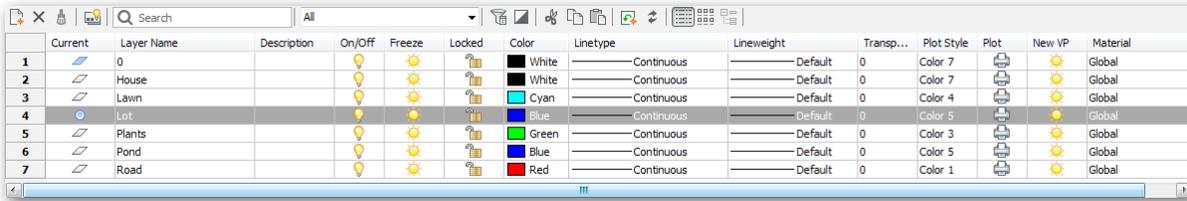
- c. Click **OK** to exit the Select Color dialog box. Notice that the color of the square across from layer Lot changes to blue.
4. Add the remaining layer names and colors, using the table below as a guide. If you make a spelling mistake, just click the layer name and type the correction.

New Layer	Layer Color	Color Number
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 layer 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 in the Entity Properties toolbar or ribbon 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.

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

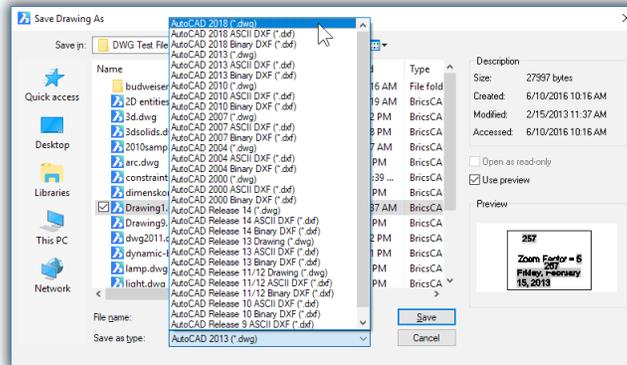
The most important task in your CAD career is to save your work. Use the **Save** command, as follows.

1. In the ribbon’s **Home** tab, click the **Save** button in the File panel.
Or, on the toolbar, click the icon that looks like a diskette  (tooltip: Save).

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 V19. 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:



Save As File Format	Saves Drawings in File Formats Used By These Versions				
AutoCAD 2018	2018	2019			
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.”:

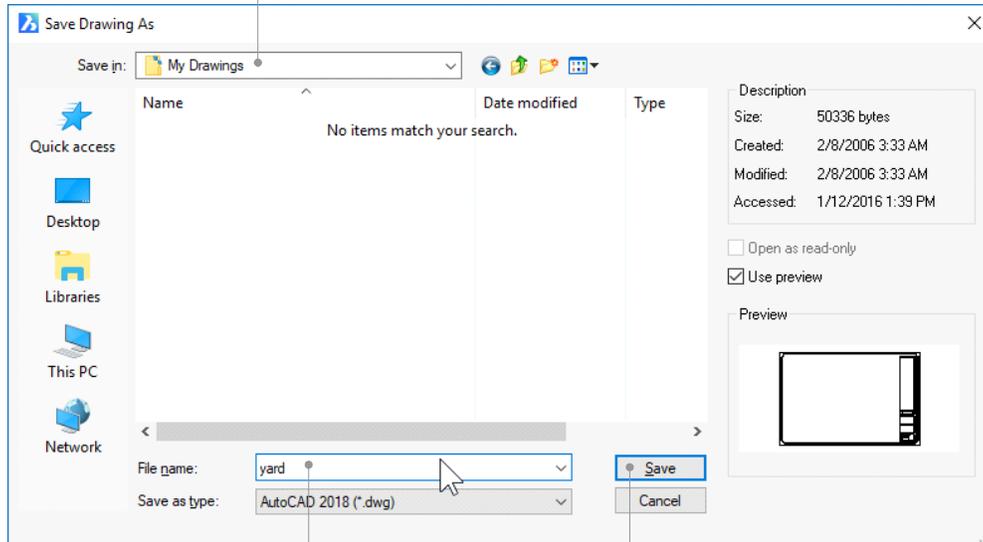
AutoCAD Release 10	R10 (released in 1988)
AutoCAD Release 9	R9 (released in 1987)
AutoCAD Release 2.6	R2.6 (released in 1987)
AutoCAD Release 2.5	R2.5 (released in 1986)

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.

- Because new drawings have a generic name, such as “Drawing1.Dwg,” BricsCAD automatically 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.

Step 1: Choose drive and folder in to save the file



Step 2: Name the drawing file

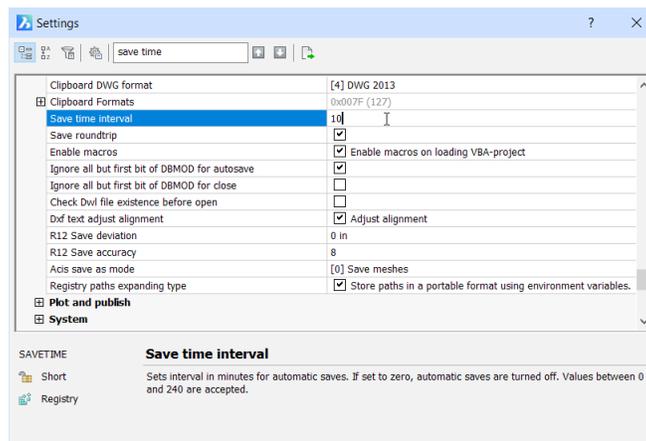
Step 3: Click Save

- 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 few 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 far too long.



QUICK SUMMARY OF ALTERNATIVE SAVE FORMATS

In addition to saving drawings in AutoCAD's .dwg file format, BricsCAD Classic, Pro, and Platinum 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)
- ▶ Industry Foundation Classes (*.ifc) for other BIM programs
- ▶ Mudbox in ASCII and binary formats (*.fbx) and Collada (*.dae) for rendering software
- ▶ STL (stereolithography) for 3D printers (*.stl)
- ▶ Windows Bitmap (*.bmp), Metaformat (*.wmf), and Enhanced Metaformat (*.emf)
- ▶ And many versions of DWF, such as 2D, 3D, binary, ASCII, and compressed ASCII

To enable automatic backups and to make automatic saves 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**.

The **Save Time Interval** setting specifies how much time elapses before BricsCAD saves drawings. Don't set this number too low, such as 1 minute, 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 manually after finishing a significant amount of editing.

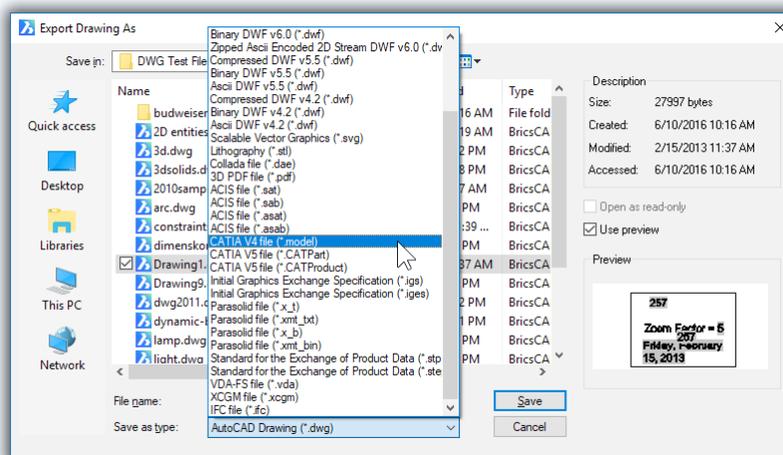
COMMUNICATOR TRANSLATOR

When you need to translate drawings to and from other formats, then you may want enlist Communicator. This is an optional, extra-cost add-on for BricsCAD Pro and Platinum that translates files in additional formats.

IMPORT		
File format	Direct translator	Parasolid-based translator
ACIS	R1 – 2018 1.0	
CATIA V4	4.1.9 – 4.2.4	
CATIA V5	V5R8 – V5–6R2017	
3DExperience(CATIA V6)	up to V6 R2017x	
IGES	up to 5.3	
Inventor	V6 – V2017	
JT	8.x, 9.x and 10.x	8.x, 9.x and 10.x
NX	NX 1 – NX 11	11 – NX 11
Parasolid	9.0.x – 29.0.x	9.0 – 29.0.137
Pro/E / Creo	16 – Creo 4.0	
Solid Edge	V18 – ST9	V18 – ST9
SolidWorks	2003 – 2017	98 – 2017
STEP	AP203, AP214, AP242	
STL	All (Graphical data only)	
VDA–FS	1.0 – 2.0	
XCGM	R2012 – 2018 1.0	

EXPORT		
File format	Direct translator	Parasolid-based translator
3D PDF	1.7	
ACIS	2018 1.0	
CATIA V4	4.2.4	
CATIA V5	V5–6R2017	
IGES	5.3	
STEP	AP203, AP214, AP242 (*)	
VDA–FS	2.0	
XCGM	2017 1.1	
Parasolid	N/A	29.0.137

You access the additional formats with the **Import** and **Export** commands, after the software is installed on your computer.



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.

In this chapter, 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
- Exporting drawings in PDF format

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

- @** At symbol specifies relative coordinates, such as @2,3
- #** Pound sign specifies absolute coordinates, such as #4,5
- <** Angle bracket specifies angles, such as 10<45
- Dash forces BricsCAD to use the command-line version of a command, such as -layer
- [option]** Square brackets indicate command options, such as [Undo]
- <value>** Angle brackets indicate the default (current) value, such as <LOT>
- x** x specifies the X coordinate along the horizontal axis
- y** 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.

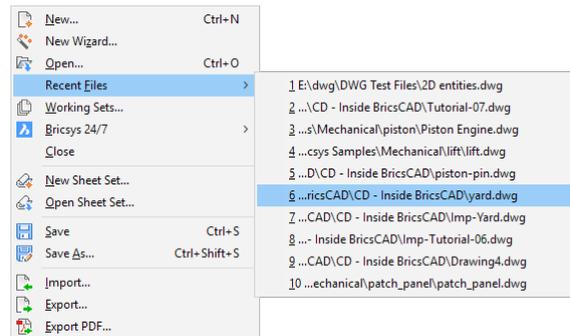
Reopening Drawings

Command	Open
Alias	op
Menu Bar	File Open
Ribbon	Home File Open
Shortcut	Ctrl+O
Toolbar	Standard

If you exited BricsCAD at the end of the last lesson, then you need to start the program again, and then load the *Yard* drawing.

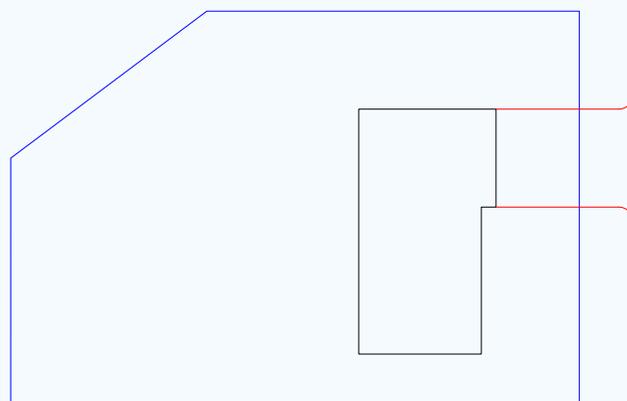
The **Recent Drawings** item of the **File** menu always holds the names of the last ten drawings you opened with BricsCAD. Here's how to do this:

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



THIS CHAPTER'S DRAWING

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



Tutorial-04 drawing file

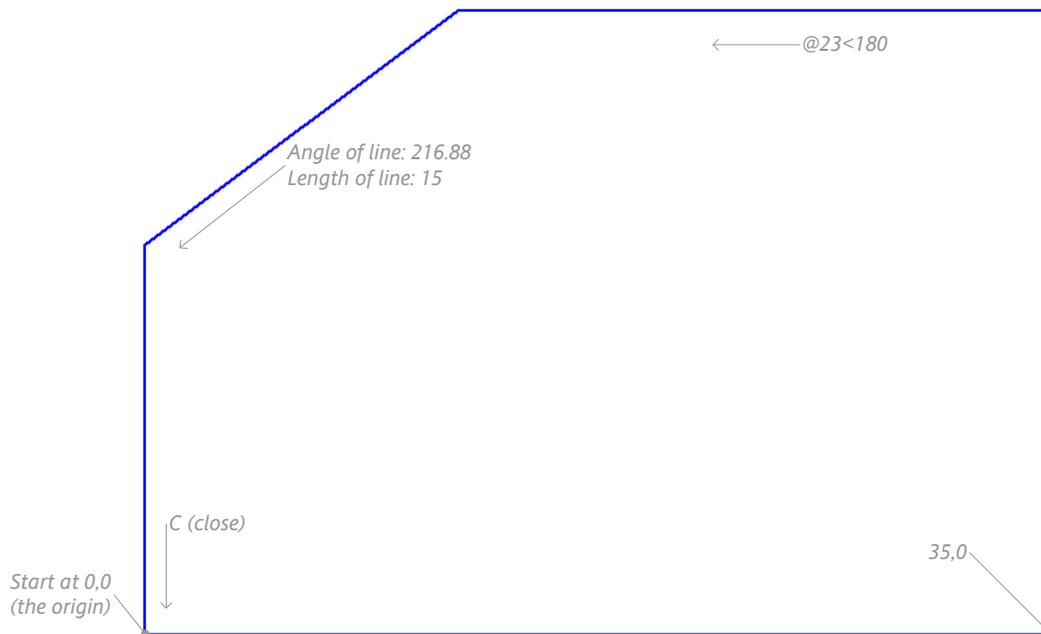
Notice that when the *Yard* drawing opens in BricsCAD, it looks exactly the same as when you last saw it — that is to say blank, except for the grid. Let's get some lines on the screen!

Drawing the Lot's Boundary

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

The first thing to do is to draw the boundary of the yard, as this helps orient yourself on the screen and lets you see the extents of the drawing.

You draw the lot boundary with lines, and so you use the **Line** command to do that. You begin drawing at the lower-left corner, starting at the origin (0,0), and then working your way counter-clockwise around the lot, 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.

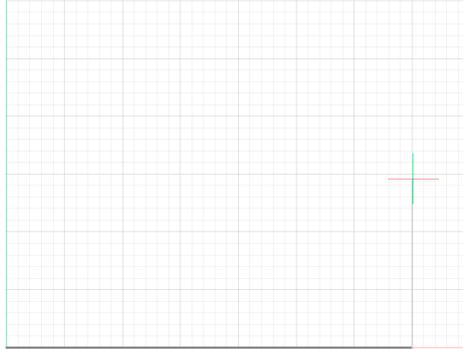


2. Enter the **Line** command, as follows:

: **line**

- Respond to the 'Specify first point:' prompt by typing the coordinates of the origin:
Start of line: **0,0**
- 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"**.)
Set end point or [Angle/Length/Undo]: **35,0**

See the figure below for what the first line should look like.



- The next line is 24m north (up). 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!)
Set end point or [Angle/Length/Follow/Undo]: **35,24**

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

Set end point or [Angle/Length/Follow/Undo]: **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."

- 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:
Set end point or [Angle/Length/Follow/Close/Undo]: **@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!

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:

Set end point or [Angle/Length/Follow/Close/Undo]: **a**

Angle of line: **216.88**

Length of line: **15**

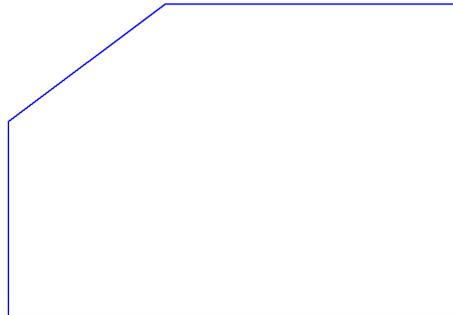
(In imperial units, Length of line: **50'**.) Notice that the **Angle** option removes the need to use the @ and < characters.

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):

Set end point or [Angle/Length/Follow/Close/Undo]: **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. 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**

```
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/  
Window/Object] <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, or
- ▶ Use the **XLine** command to draw a pair of construction lines, then start drawing from their intersection, or
- ▶ Use the **From** option to start drawing from an offset from the lot corner, or
- ▶ Use *tracking* to offset the starting point (described in a later lesson), or
- ▶ Draw the house in the corner of the lot, then move the house into position

Please don't feel overwhelmed by this long list of options. I show 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 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:

```
Layer [ ? to list/New layer/Make new current layer/Set layer as current/Color/Linetype/  
LineWeight/TRansparency/MATerial/Plot/stAte/turn layer ON/turn layer 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**.

```
Layer to set as 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.

```
Layer [ ? to list/New layer/Make new current layer/Set layer as current/Color/Linetype/  
LineWeight/TRansparency/MATerial/Plot/stAte/turn layer ON/turn layer OFF/Freeze/Thaw/  
LOck/UnLock]: (Press Esc)
```

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**.



For all 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**.

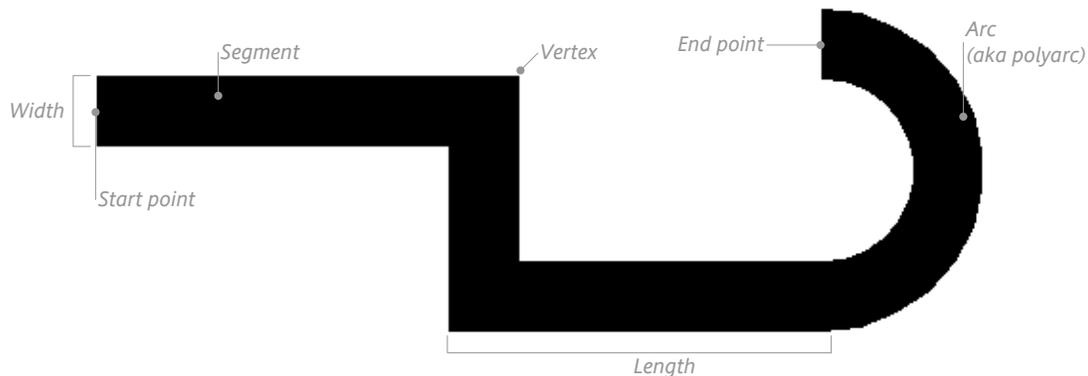
Drawing the House Outline

Command	Orthogonal
Aliases	ortho, or
Menu	Settings Draw Orthogonal
Ribbon	Settings Settings Draw Orthogonal
Toolbar	Settings
Shortcuts	Ctrl+L, F8, Shift
Status Bar	ORTHO

Command	Pline
Aliases	pl, polyline
Menu Bar	Draw Polyline
Ribbon	Draw Draw Polyline
Toolbar	Draw Polylines

Earlier, you drew the lot boundary as a polygon with the Line command. It looks 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 of many features, as suggested by the prefix *poly* (from the Greek word *polloi* for “many”). 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>

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

Start of polyline:

Set next point or [draw Arcs/Close/Distance/Follow/Halfwidth/Width/Undo]:

Next point — specifies the location of the polyline's next vertex

draw Arcs — draws polyarcs; you can switch between lines and arcs when drawing polylines

Close — joins the last endpoint with the starting point

Distance — specifies the length and angle of the next segment

Follow — draws a specific distance, in the direction of the cursor's angle

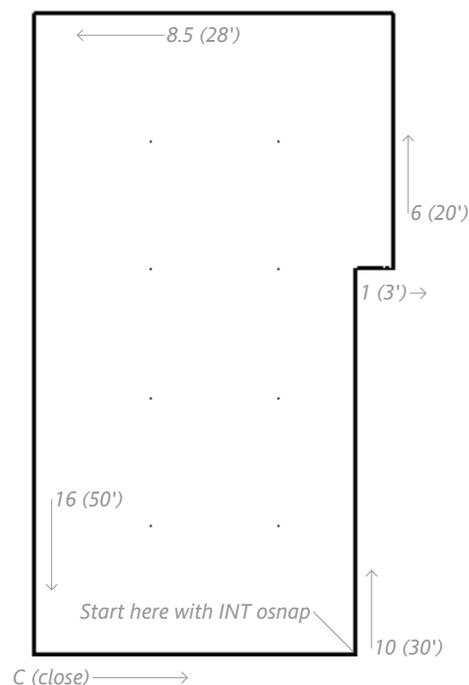
Halfwidth — specifies the width of polylines by the distance from their center line to their outside edge

Width — specifies the width of polyline segments; allows independent starting and ending widths for tapered polylines

Undo — undoes the last polyline drawing operation

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

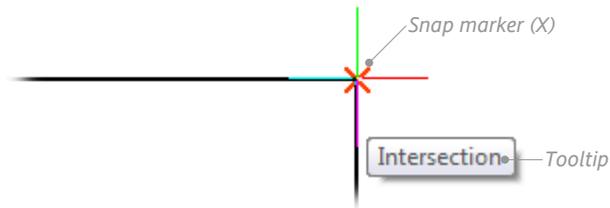


BricsCAD prompts you for the point from which to begin drawing the polyline. But instead of specifying coordinates, you this time ask BricsCAD to find a geometric feature by using *entity snap*:

Select start of polyline or [Follow] <Last point>: **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.”

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.’

Set next point or [draw Arcs/Distance/Follow/Halfwidth/Width]: (*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.

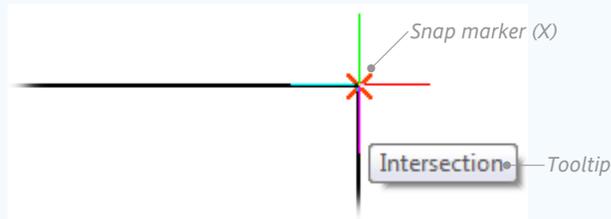
QUICK SUMMARY OF ENTITY SNAP MODES

BricsCAD has the following entity (object) snaps that look for specific geometric features on entities:

Mode	Command	Snap 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
gcen	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 Mzp 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

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:

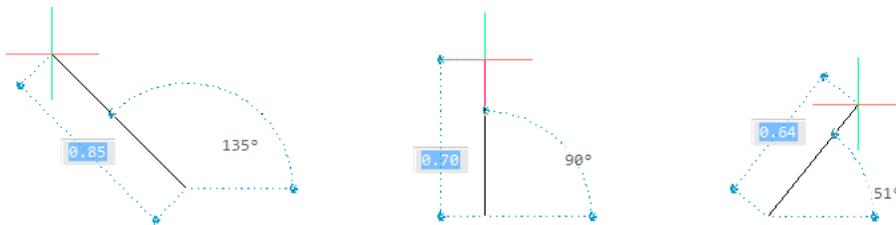
5. To use dynamic input for the next segment, the 1m line, follow these steps:
 - a. On the status bar, ensure **DYN** is turned on. When **black** (instead of **gray**) then dynamic input is on.



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.



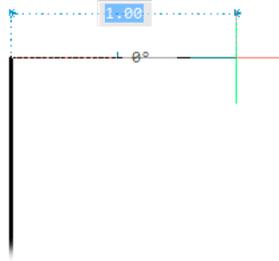
Left to right: Dynamic dimension values changing as the cursor moves

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.

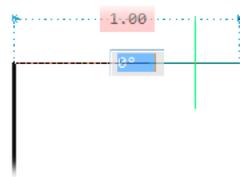


- c. Enter the length of the wall, **1** and then press **Tab**. (In imperial units: **3'**.)



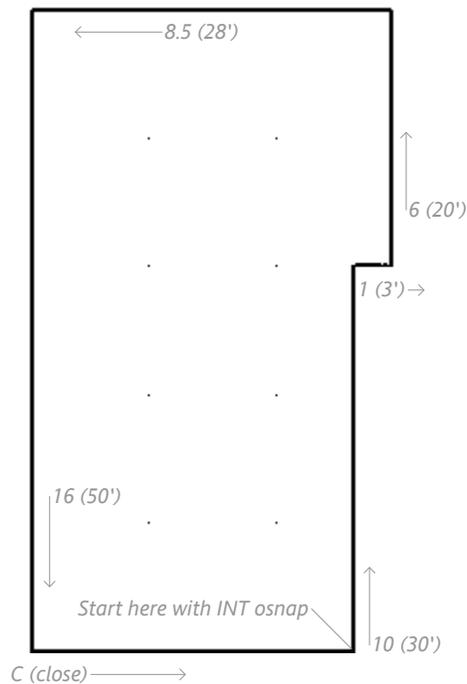
Notice that the length field (1 or 3) turns **red**. This tells you that BricsCAD has accepted the value, and has locked it temporarily.

- 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**.



- Draw the remainder of the house outline by moving the cursor in the appropriate direction, and then entering the distances:

Cursor Direction	Metric Distance	Imperial Distance
East	1	3'
North	6	20'
West	8.5	28'
South	16	50'

- 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 so 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

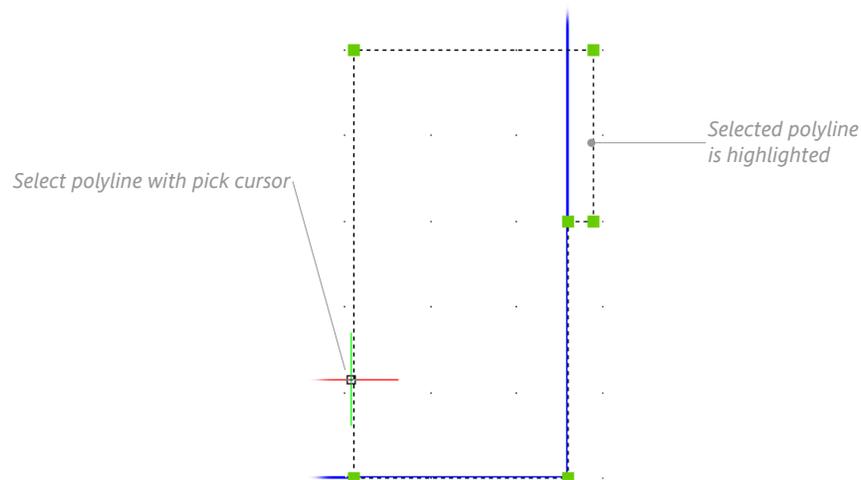
Command	Move
Alias	m
Menu Bar	Modify Move
Ribbon	Change Arrange Move
Toolbar	Modify Move

Now that you've drawn the outline of the house, you need to move it into position. Entities are moved with the **Move** command.

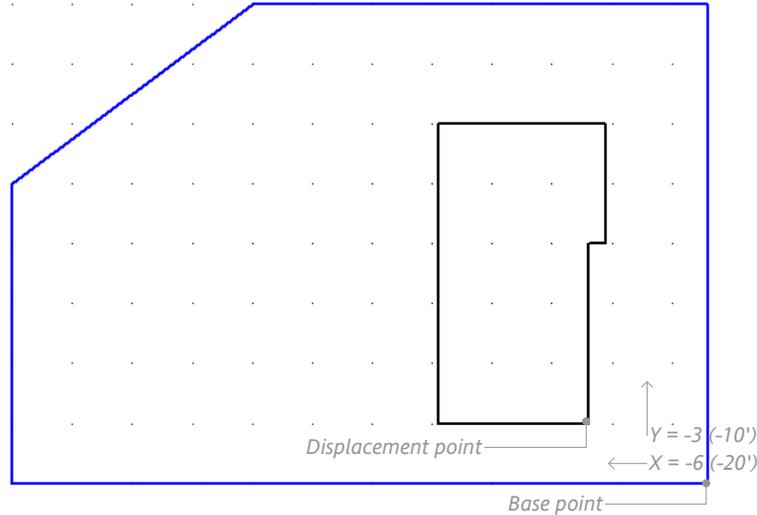
- Enter **Move** the 'Command:' prompt:
: move
- Notice that BricsCAD asks what you want to move:
Select entities to move:

("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] <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:
 Enter second point <Use base point as displacement>: **-6,3**

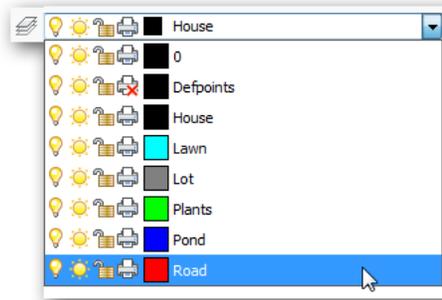


- The coordinates -6,3 (in imperial units: **-20',10'**) tell BricsCAD to move the house left by 6m (x direction = **-6** meters) and up by 3m (y direction = **3**). BricsCAD instantly relocates the house much faster than a house mover. The Move command shows you a powerful aspect of CAD: no eraser dust!
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:

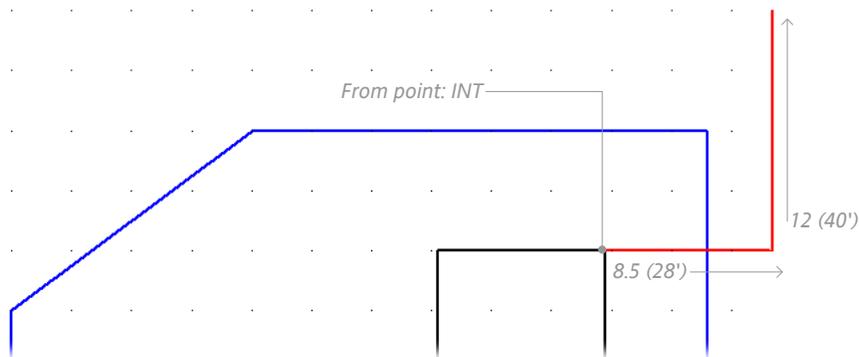
Start of line or [Follow] <Last point>: **int**

Snap to intersection of: (Pick upper-right corner of house)

Set end point or [Angle/Length/Undo]: **8.5**

Set end point or [Angle/Length/Undo]: **12**

Set end point or [Angle/Length/Follow/Close/Undo]: (Press **Enter** to end the command)



Recall that you can “back up” to undraw incorrect lines with **u** (short for “Undo”), as follows:

Set end point or [Angle/Length/Follow/Close/Undo]: **21**

Set end point or [Angle/Length/Follow/Close/Undo]: **u**

Set end point or [Angle/Length/Follow/Close/Undo]: **12**

FINISHING THE DRIVEWAY

Command	Fillet
Alias	f
Menu Bar	Modify Fillet
Ribbon	Change Modify Fillet
Toolbar	Modify Fillet

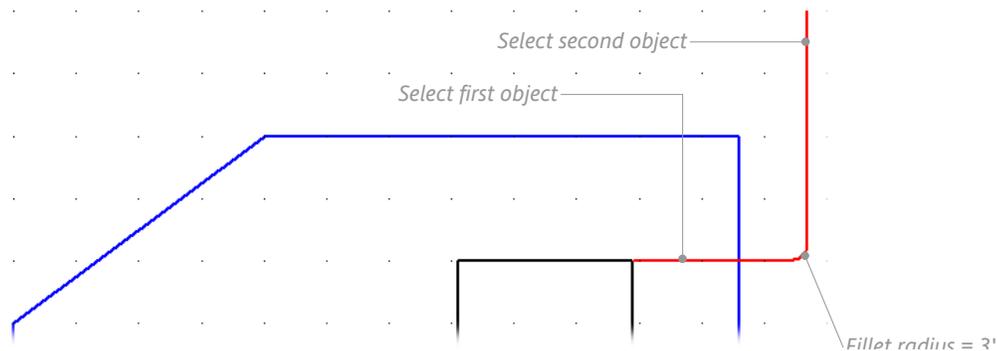
To add the *curb return* — the arc joining the driveway and street — use BricsCAD’s **Fillet** command. It draws arcs between intersecting lines. The lines don’t have to physically meet; BricsCAD takes care of extending (or trimming) the lines so that the arc is drawn between them.

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 **r** option:
Fillet (radius=0.5): Select first entity or [Settings/Polyline/Radius/Trim/Undo/Multiple]: r

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=1): Select first entity or [Settings/Polyline/Radius/Trim/Undo/Multiple]:
(Pick one Line)
Select second entity (select with pressed SHIFT to make corner): (Pick the other Line)
BricsCAD automatically adjusts the two lines to fit the 1m arc between them.



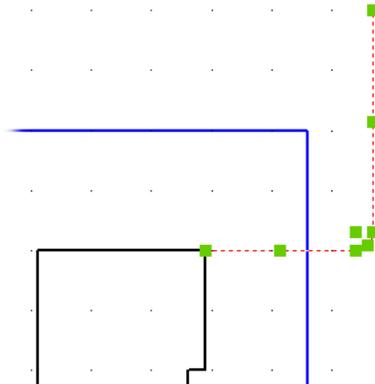
Mirroring Entities

Command	Mirror
Alias	mi
Menu Bar	Modify 2D Mirror
Ribbon	Change Modify 2D Mirror
Toolbar	Modify Mirror

Earlier, 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.

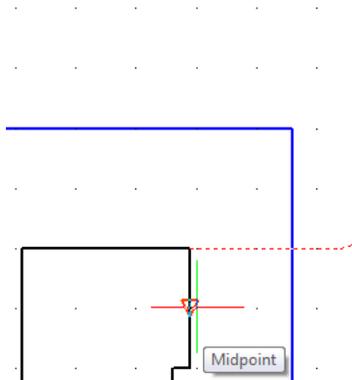
1. Enter the **Mirror** command:
: `_mirror`
2. BricsCAD asks you to select the entities you want to mirror. Use the cursor to pick the line and arc segments, as follows:
Select entities to mirror: (*Pick the driveway line*)
Entities in set: 1
Select entities to mirror: (*Pick the curb return*)
Entities 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:

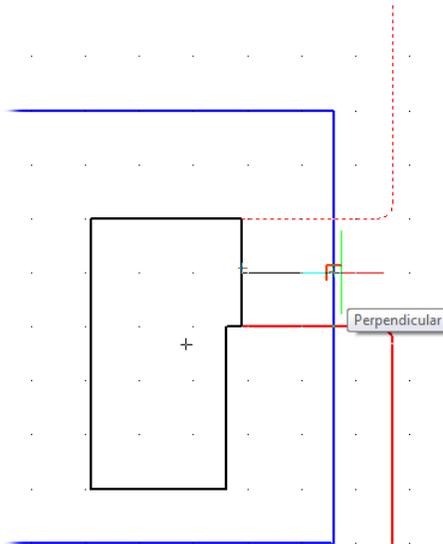
Start of mirror line: **mid**

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? [Yes-delete entities/No-keep entities] <No-Keep entities>: **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.

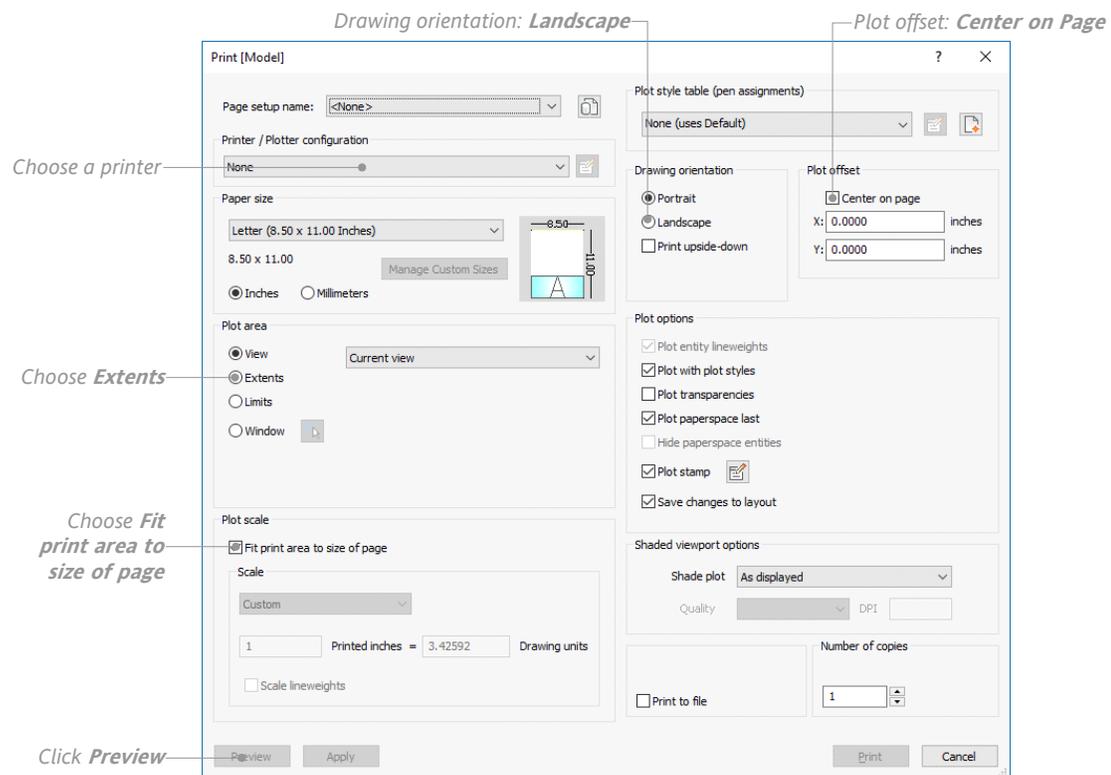
Putting Drawings to Paper

Command	Print
Menu Bar	File Print
Ribbon	Home Print/Plot Print
Shortcut	Ctrl+P
Toolbar	Standard Print

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.)

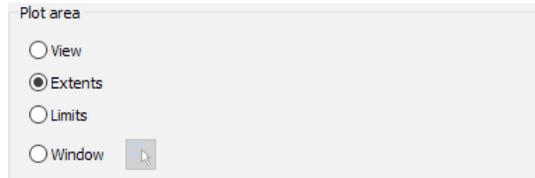
2. The dialog box has many options. Fortunately you can ignore most options, except for these:

Printer/Plotter Configuration selects the printer you want to use:



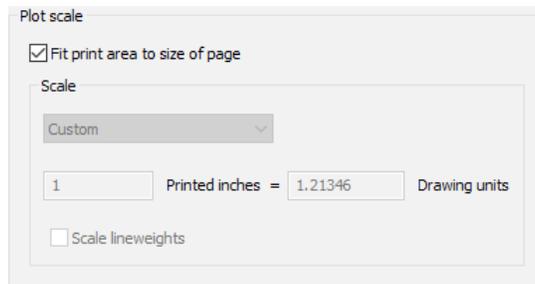
Select a printer from the **Name** droplist. BricsCAD supports local and networked printers.

Plot Area determines which part of the drawing is printed:



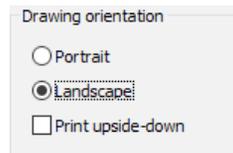
Select **Extents**. This option ensures *everything* in your drawing is plotted on the paper.

Plot Scale determines how large the drawing is printed on the paper:



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 turns the drawing by 90 degrees on the paper:



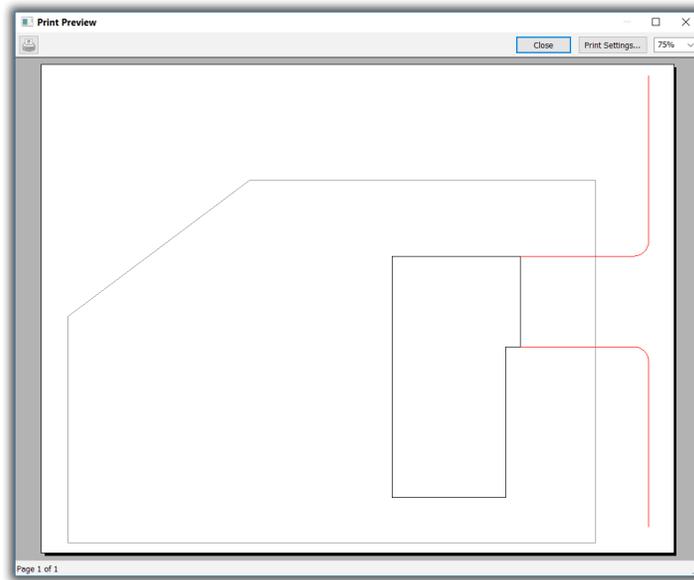
If the orientation of the paper doesn't match that of the drawing, then you need to change it. The paper is tall but the drawing is wide. To change the orientation of the paper, select **Landscape**.

Plot Offset moves the drawing to one side of the paper or the other:



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.



4. 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.

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%.

If the **Preview** button is grayed out in the Print dialog box, this means you have not yet selected a printer for this:



To make preview available, click the Printer/Plotter Configuration button in the Printer/Plotter Configuration dialog box:



SOLVING PRINTING PROBLEMS

If the printer does not produce the drawing correctly, here are some items to check:

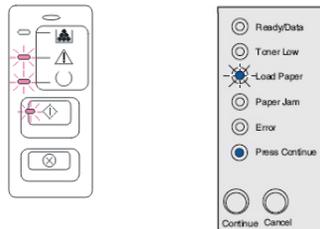
- ▶ Check in the Print dialog box that...
 - ...drawings are centered on the page
 - ...and check that the Extents of the drawings is printed.

These two settings ensure that *all* of the drawing will be printed on the paper.



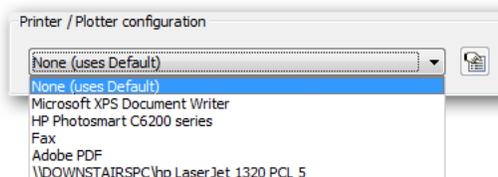
Left: Print will be centered on page; *right:* Print will be made to drawing extents.

- ▶ 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.

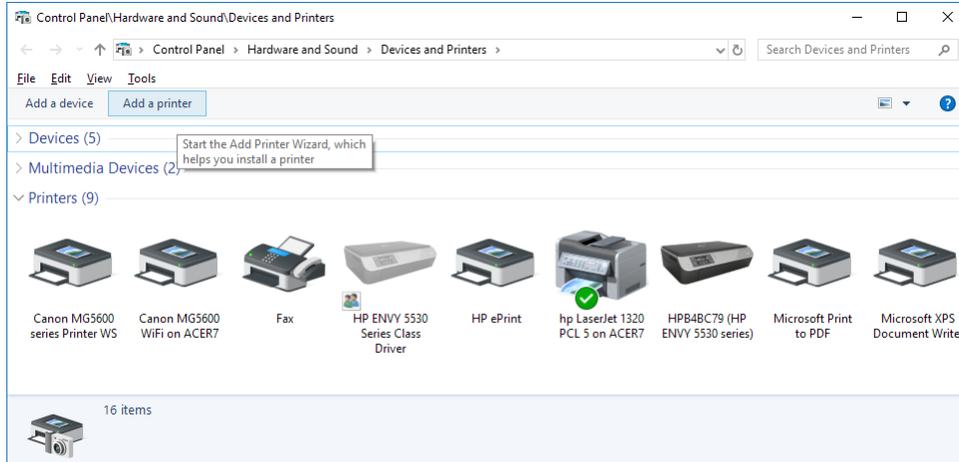


- ▶ In some cases, the *spooler* acts up in Windows. The spooler is software that feeds data to the printer, allowing you to continue working with BricsCAD while the printer works in the background. Sometimes, the spooler plugs up, preventing the printer from receiving the data.

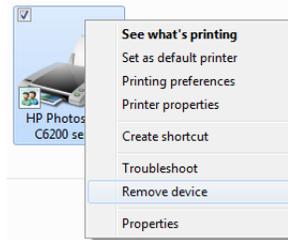
To fix the problem, follow these steps:

1. Click **Start**, and then **Devices and Printers** in Windows 7. (In Windows 8.x and 10, click the **Start** button, and then type “printers.” Choose **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.

Smudgy Prints with Bad Colors with Inkjets

Most printers use inkjet technology, which offers additional problems over laser printers. Here are solutions to some problems:

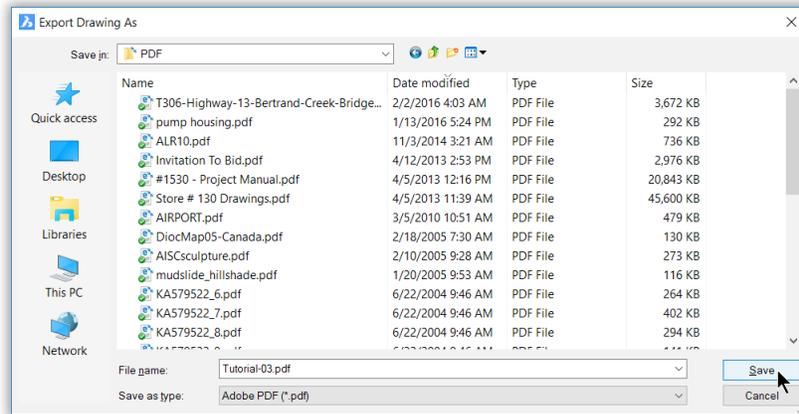
- ▶ When the prints come out smudgy, with ink that rubs off on your hand, the paper may have been put in the cartridge upside-down. The print side contains clay that absorbs ink; the underside does not.
- ▶ When prints of renderings or photographs have no detail in dark areas, this means that the black cartridge is low on ink. Cartridges low on ink do not squirt enough ink on the paper.
- ▶ When color prints are predominantly one color, this means that one or more colors of a multi-color cartridge are empty.

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. (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").



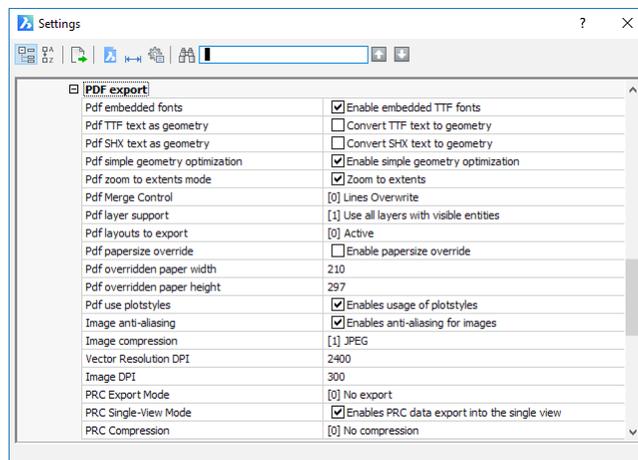
Click **Save**.

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 it extends to fit the page

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

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.

Vector Resolution (2400) specifies the resolution of vector graphics.

PRC Export Mode (0) determines if a 3D model is exported as a BREP (boundary representation), mesh, or not at all.

PRC Single-View Mode (1) determines whether the 3D model is exported as a single view.

PRC Compression (0) specifies whether the PRC file is exported with medium, high, or no compression.

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 that represent the outlines of the lot, house, and driveway. Now it's time to add details, like the lawn, some trees, and a pond.

In this chapter, you learn to use BricsCAD's intermediate commands, such as those that create ellipses, place hatch patterns, makes arrays, 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
ArrayClassic	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 back of the yard has lawn and garden areas. For 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. (If you're doing the tutorials in imperial units, open *Imp-tutorial-04.dwg*.) You can download them from my public Dropbox folder at <https://www.dropbox.com/s/ft6gzw3r408ndgd/Inside-BricsCAD-Tutorial-Files.zip?dl=0>.

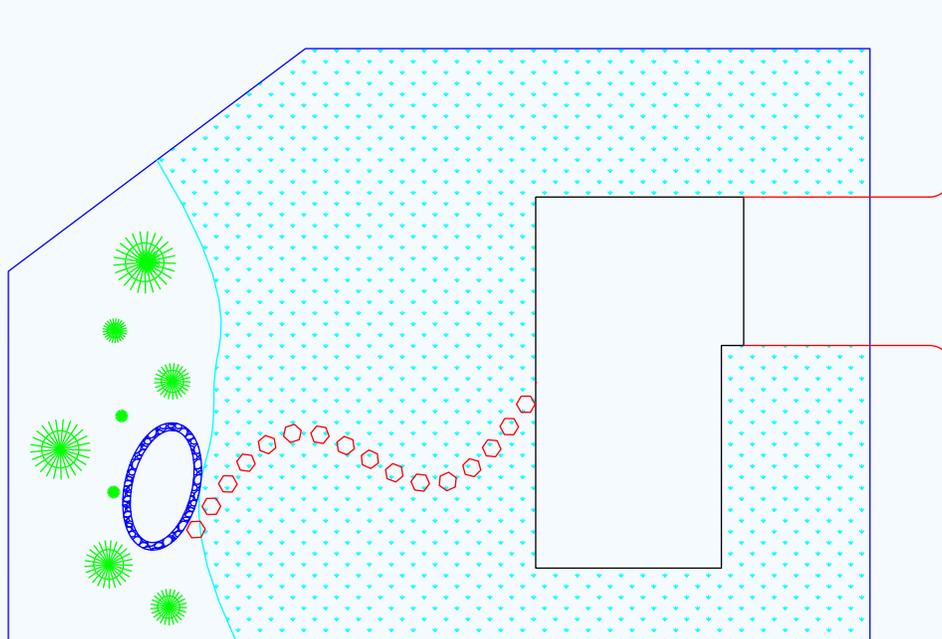
Before starting to draw, though, take these steps:

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.

ORID | ORTHO | POL

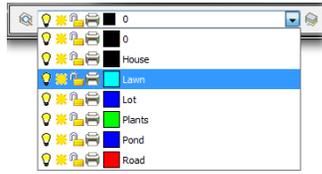
THIS CHAPTER'S DRAWING

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

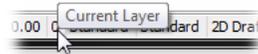


Tutorial-05 drawing file

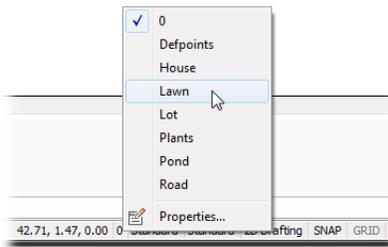
- Change the working layer to **Lawn**; its color should be cyan (light blue). There are several different ways to do this:
 - In the toolbar or ribbon interfaces, choose “Lawn” from the Layers droplist (as shown below).



- Or, from the Properties panel, choose “Lawn” from the Layers droplist
- Or in the status bar, right-click the current layer field...



... and then choose “Lawn” from the list



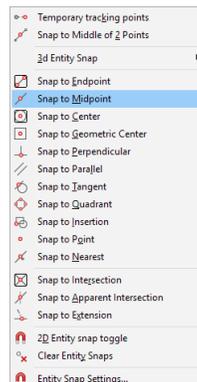
- Or at the command prompt, you can use my favorite: enter the **CLayer** system variable. (Clayer is short for “current layer.”) This is a quick way to change layers when you know their names. In fact, this is the method used by the status bar.

: clayer

New current value for CLAYER <"0">: **lawn**

Now the drawing is ready for you to draw the boundary between the lawn and the garden .

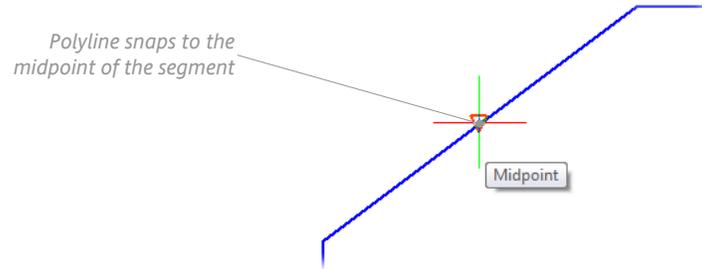
- To draw the polyline, start the **PLine** command.
 - : pline**
- To determine where the polyline starts, you use an entity snap mode. But this time, you access entity snap in a different manner from last chapter’s method. Here’s how:
 - Hold down the **Shift** key on the keyboard.
 - Press the right 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 to **Snap to Midpoint**, and then click. On the command line, notice that BricsCAD reports that MIDpoint esnap is indeed activated:

Start of polyline: `_midpoint`

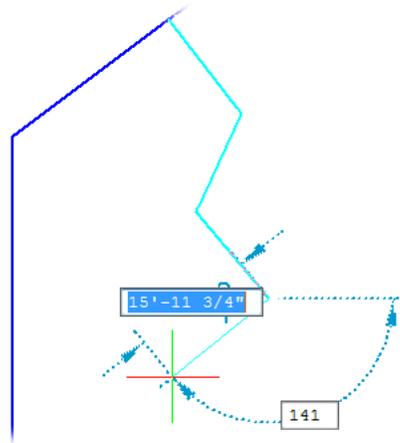
5. Move the cursor to *any* point on the diagonal portion of the lot line, and then select the line.
Snap to midpoint of: (*Pick a point on the diagonal Lot Line*)



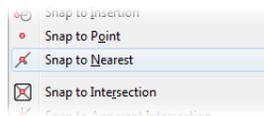
The MIDpoint esnap forces to BricsCAD snap to the precise midpoint of the lot's diagonal line.

6. Continue with the PLine command by moving your way down toward the bottom of the yard line, picking a few points along the way. Space them at roughly 3m to 6m intervals. (In imperial units, that's 10' to 20' intervals.)

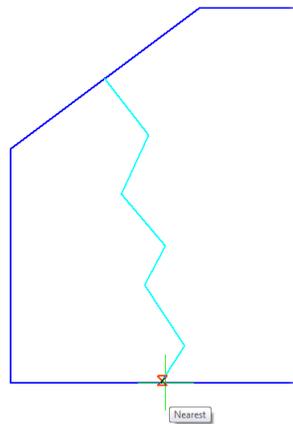
If you are not sure how far 3m or 6m is (10ft or 20ft), turn on **DYN** on the status bar, and then watch the distance dynamically, as shown below:



7. When you get to the bottom lot line, you need to bring the polyline to a precise ending. Press **Shift**+right mouse button to again display the shortcut menu. This time, select **Nearest** entity snap.



- Pick anywhere the line crosses the cursor's aperture box. BricsCAD snaps the polyline precisely to the lower lot line.



- Press **Esc** to end the PLine command.

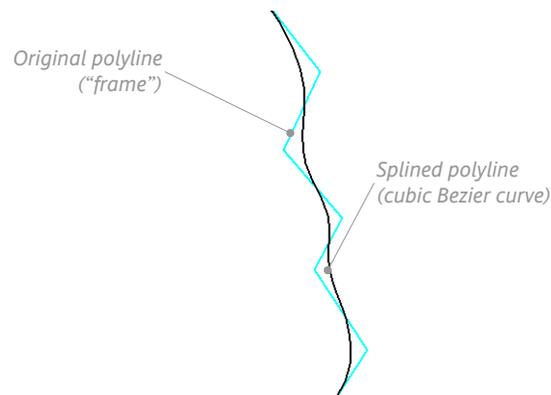
Here is the reason you created the boundary as a polyline: you now use the **PEdit** command (short for “polyline edit”) to easily modify the crooked segments into a smooth flowing curve.

SMOOTHING POLYLINES

Command	PEdit
Alias	PE
Menu Bar	Modify Edit Polyline
Mouse	Double-click a polyline

The PEdit command's purpose is to change the look of polylines. It has many, many options; don't worry, this time around you use just one of them.

- To edit the polyline, begin the **PEdit** command:
: **pedit**
- Pick the polyline.
Select polyline to edit [Multiple]: *(Pick the polyline, or type L to choose the Last-drawn entity)*



QUICK SUMMARY OF THE PEDIT COMMAND

The PEdit command edits polylines:

: **pedit**

Select polyline to edit [Multiple]:

Edit polyline [Edit vertices/Close/Decurve/Fit/Join/Linetype mode/Reverse direction/Spline/Taper/Width/Undo] <eXit>:

Multiple — turns lines, arcs, circles, and/or splines into a single polyline.

Edit vertices — edits width and position of individual segments; inserts and removes vertices.

Close / Open — closes (or opens) an open (or closed) polyline by drawing (or removing) a segment between the starting and ending vertices; this Close prompt is displayed when an open polyline is being edited, the Open prompt when 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 direction — reverses the direction of the polyline.

Spline — applies a Bezier spline to the polyline.

Taper — applies a varying width to segments.

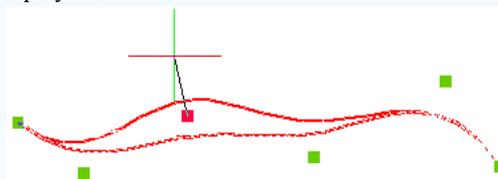
Width — applies a uniform width to all segments making up the polyline.

Undo — undoes the last polyline editing operation.

eXit — exits the command.

While you can use the **PEdit** command's **Edit vertices** option to change the shape of polylines, it is much easier to edit with grips, 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**.
-

3. Use the command's **Spline** option to smooth the jagged lines into a flowing curve, as follows:
 Edit polyline [Edit vertices/Close/Decurve/Fit/Join/Linetype mode/Reverse direction/
Spline/Taper/Width/Undo] <eXit>: **s**
 The straight lines disappear and are replaced by a smooth curve. (Technically, BricsCAD redraws the line segments as a *cubic Bezier curve* based on the polyline frame.)
4. Exit the PEdit command by pressing **Enter**:
 Edit polyline [Edit vertices/Close/Decurve/Fit/Join/Linetype mode/Reverse direction/
 Spline/Taper/Width/Undo] <eXit>: (*Press Enter*)

Grips Editing

To edit the polyline, you took two steps in this order: (1) you started the PEdit command, then (2) you selected the polyline to edit. BricsCAD can do this in reverse — select an entity, and then edit it — which sometimes is a more convenient way to approach editing.

There are two ways to do this:

- Grips editing — click the entity, and then move grips found on it
- Double-click editing — double-click the entity, which accesses the command best suited for editing it

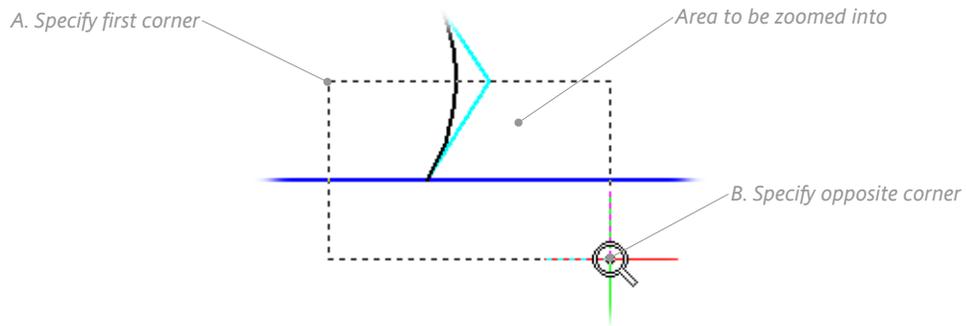
Grips editing is usually the fastest and most direct way to edit entities in CAD drawings — sometimes. I say “sometimes,” because not all editing commands lend themselves to it. In the case of entities like polylines, however, it does.

Here you use grips editing to change the shape of the splined polyline that you drew to separate the garden from the lawn.

If you cannot see the entity you want to edit, hold down the roller button on the mouse and then move it to pan the drawing. Alternatively, use the **Pan** command or a scroll bar to move the entity into view.

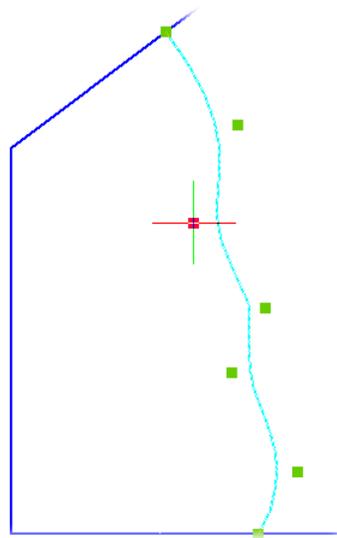
1. First, enlarge the view of the splined polyline with the **Zoom Window** command:
 - a. Enter the **Zoom** command:
 : **zoom**
 - b. Specify the **Window** option by entering 'w':
 Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale
 (nx/nxp)/Window/Object] <Scale (nX/nXP)>: **w**
 - c. The Window option requires you to pick a rectangular area on the screen to magnify.
 Specify first corner: (*Pick a point*)
 Specify opposite corner: (*Pick another point*)

When you pick points for the first and other corners, you specify the two opposite corners of a rectangle, as shown in the following figure.

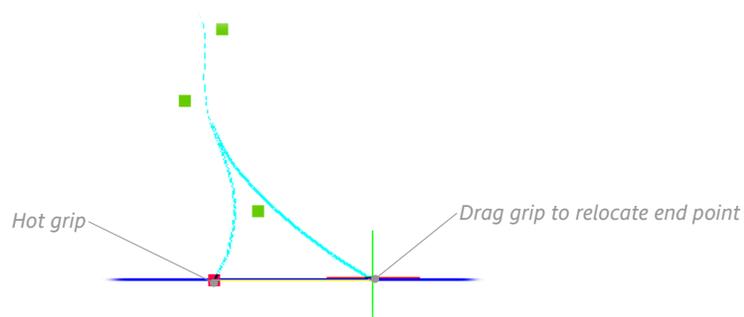


Now that the working area is larger, you can edit the polyline with grips. In an earlier lesson, I pointed out the small square at the center of the crosshair cursor. It is called the “pickbox.” When BricsCAD displays the pickbox, you can select entities without any command being active.

2. Use the pickbox to pick the polyline. Notice that the polyline changes its look from solid to dashed, and that small green squares appear along the splined polyline. These 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 you splined it.)

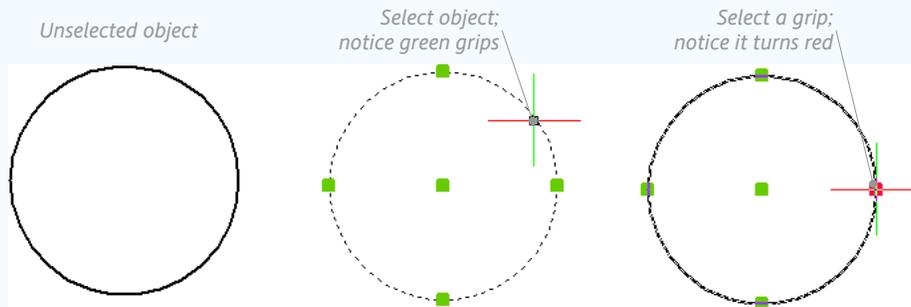


3. Pick the green grip at the very bottom end of the polyline. BricsCAD prompts:
New location for control point: **nea**

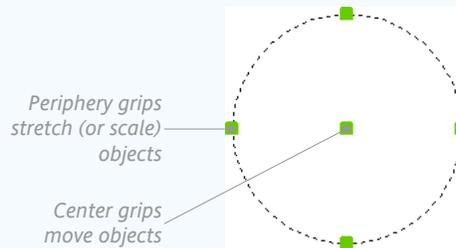


QUICK SUMMARY OF EDITING WITH GRIPS

Grips editing is interactive: you select an entity (green grips appear), select one of the grips (grip turns red), and then manipulate the red grip to modify the entity.



Most grips look the same (squares), and so it can be hard to determine their function — until you start to drag one of them. Typically, an interior grip moves an entity, while grips on the periphery stretch or scale the entity.



Some entities sport triangular entities, such as polylines: this grip moves the segment.



When **DYN** is turned on, then dynamic dimensions appear during grips editing:

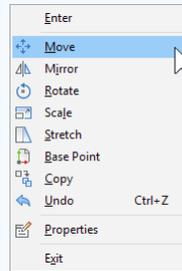


...continued

...continued

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 right-click for a shortcut menu, or else press the **Spacebar** repeatedly.



Each time you press the Spacebar, BricsCAD displays another set of prompts:

```
** STRETCH **  
<Stretch to point>/Base point/Copy/Undo/eXit: (Press spacebar)  
  
** MOVE **  
<Move to point>/Base point/Copy/Undo/eXit: (Press spacebar)  
  
** ROTATE **  
<Rotation angle>/Base point/Copy/Undo/Reference/eXit: (Press spacebar)  
  
** SCALE **  
<Scale factor>/Base point/Copy/Undo/Reference/eXit: (Press spacebar)  
  
** MIRROR **  
<Second point>/Base point/Copy/Undo/eXit: (Press spacebar)
```

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).

4. 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.)
5. 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.
6. When you are finished reshaping the polyline, press **Esc** twice to exit grips editing.
7. Enter the **Zoom Previous** command to see the entire drawing again.

When you move the cursor over a polyline, the Hotkey widget appears. It gives you the option to edit the entire polyline or just the segment over which the cursor is currently hovering — using grips editing. Press the **Ctrl** key to switch the editing type.



The Hotkey widget also appears for other entities.

Editing by Double-clicking

BricsCAD also lets you edit entities by double-clicking them. This is a fast way to do it, because you don't need to enter a command — or even remember the correct command name!

The table below lists the entities that react to double-clicking with a specific command. For all other entities, BricsCAD opens the Properties panel when you double-click them. You can use the **Customize** command to change how double-clicking works with entities. See *Customizing BricsCAD*.

Entity	Command Executed by Double-clicking	Notes
Array	ArrayEdit	Edits associative arrays
Attribute Definition	DdEdit	
Attribute Block	EAttEdit	
Block	RefEdit	
Dimension	DdEdit	Edits text of dimensions
Hatch	HatchEdit	
Image	ImageAdjust	Adjusts brightness, contrast, fade
Polyline	PEdit	
Multileader	DdEdit	Edits text of mleaders
Section	ClipDisplay	Adjusts clipping planes
Spline	SplinEdit	
Text	DdEdit	
Tolerance	DdEdit	Edits tolerance text
XRef	RefEdit	

HATCHING THE LAWN

Command	Hatch
Aliases	H, BH, BHATCH
Menu Bar	Draw Hatch
Ribbon	Draw Hatch Hatch
Toolbar	Draw Hatch

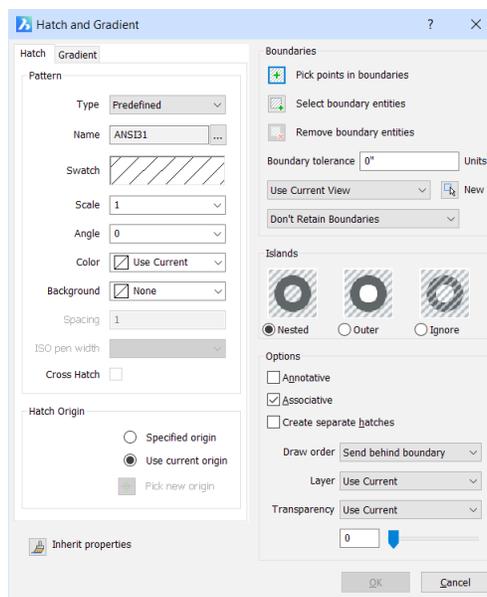
You created a boundary between the lawn and garden, but how do you show the difference between what's grass and what's dirt? One way is to add a repeating pattern that identifies areas; the repeating pattern is called *cross-hatching* or a *hatch pattern*. It is a shorthand used by drafters to indicate material, just like the purpose of linetypes is to identify lines.

In BricsCAD, hatching is done with the **Hatch** command. In this tutorial you hatch the lawn with the symbol used by drafters for grass.

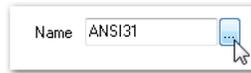


1. First, turn off the grid, since its lines can interfere with seeing the hatch pattern: on the status bar, click **GRID** so that the word turns gray.
2. Enter the **Hatch** command to start the Hatch command:
: hatch

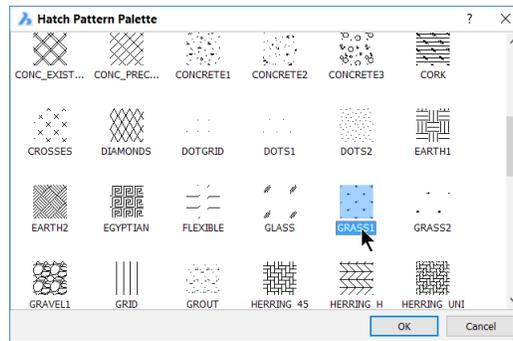
Notice the Hatch and Gradient dialog box:



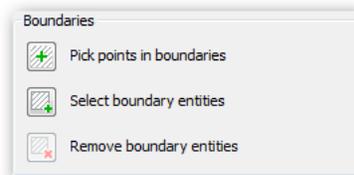
- To find the grass hatch pattern, click the  button to the right of “Name.”



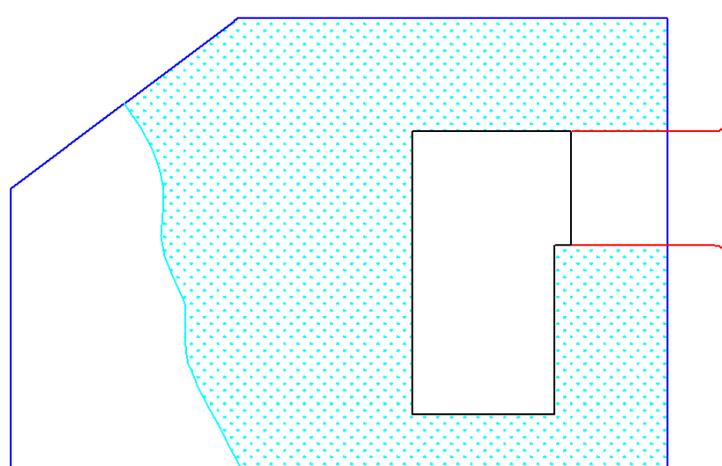
- BricsCAD displays the Hatch Pattern Palette dialog box. The patterns are listed alphabetically; look for the pattern named “Grass1.”



- Select **Grass1**, and then click **OK** to return to the Hatch and Gradient dialog box. Notice that **Name** changes to “Grass1.”
- 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 complain:
Hatch spacing too dense, or dash size too small.
Please increase pattern scale or MAXHATCH value.
- 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.



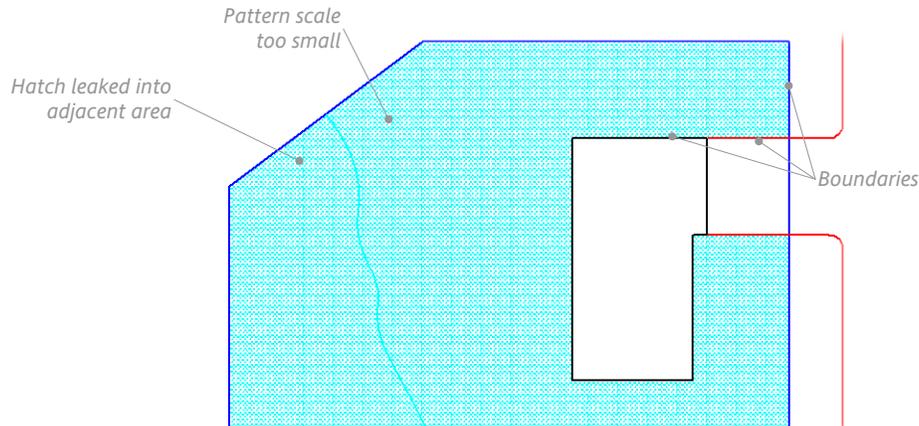
- The dialog box disappears and BricsCAD prompts you to pick a point. Do so anywhere in the lawn area. Select a point to define a boundary or hatch area: *(Pick a point inside the lawn area)*



Notice that BricsCAD immediately previews the hatch pattern for you, showing you that the scale factor is good enough. Note how precisely the hatch pattern is applied: it is clipped along boundaries automatically. Try doing a hatch pattern that neatly and that quickly by hand!

The hatch preview gives you a chance to correct errors, such as these two possible ones:

- Does the pattern flood incorrect areas — look out for patterns that leak into unwanted areas, or don't appear at all
- Is the pattern applied at the wrong scale factor — look for one that is too large or too tightly spaced. When the scale is much too large, the hatch may seem invisible; when too small, the hatch pattern looks like a solid and can take 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 Hatch and Gradient dialog box:
Select a point to define a boundary or hatch area: (*Press Enter*)
10. After you press Enter, the Hatch and Gradient dialog box reappears. Click **OK** to exit it. Notice that the hatching is applied in the color of the layer.

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!

Modifying Hatching

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 a pattern, and the hatching automatically updates itself.
- You can move, copy, and erase hatch patterns, because it acts like a block; it does not need to stay in one place.
- You can click the hatch entity to bring up the Properties panel. You use it to change the color, pattern, scale factor, rotation angle, and so on.
- You can double-click the hatch pattern to bring up the Hatch Edit dialog box. It looks identical to the Hatch and Gradient dialog box, except that it displays all the settings of the current hatch pattern.

Creating Symbols (Blocks)

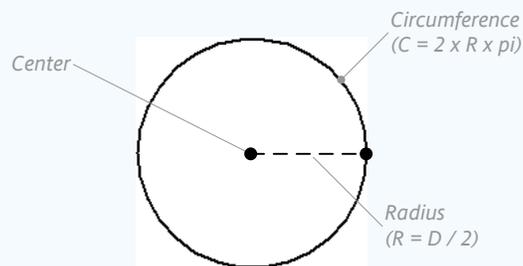
You've given the lawn its grass. Now it's time to add trees and shrubs to the garden. Instead of drawing complex things like trees with branches and leaves, landscape architects typically draw simple representations of them, such as a circle with radiating lines.

QUICK SUMMARY OF DRAWING CIRCLES

BricsCAD provides several methods for drawing circles. Use the one that suits your need the best.

: circle

Select center of circle or [2Point/3Point/TTR/Arc/Multiple]:

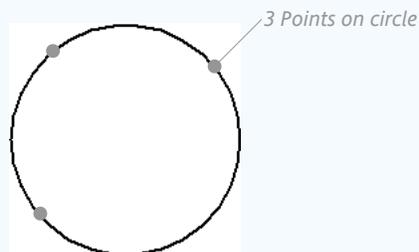


Center, Radius — pick the center point, and then specify the radius.

Center, Diameter — pick the center point, and then specify the diameter.

2-Point — pick two points to define the diameter.

3-Point — pick three points to define the circumference.



TTF — (Tangent, Tangent, Radius) pick two points of tangency to other entities, and then specify radius.

Arc — closes an arc to form a circle.

Multiple — repeats the command to draw more circles, until you press **Esc**.

And instead of drawing lines and circles over and over again, designers typically draw just one, and then repeatedly make copies of them. While you could use the Copy command, the most efficient method is to use the Block and Insert commands. **Block** creates a stencil-like object of which you place multiple copies with the **Insert** command.

DRAWING CIRCLES

Command	Circle
Alias	C
Menu Bar	Draw Circle
Ribbon	Draw Draw Circle
Toolbar	Draw Circle

In this tutorial, you learn how to draw a single tree as a simplified symbol, and then turn it into *blocks* to be used over and over again.

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

Select center of circle or [2Point/3Point/TTR/Arc/Multiple]: *(Pick a point anywhere in garden area.)*

Set Radius or [Diameter]: **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 a radius is half the diameter.)

Zooming in Real Time

Command	RtZoom
Menu Bar	View Realtime Realtime Zoom
Ribbon	View Navigate Real-time Zoom
Shortcut	<i>roll the mouse wheel</i>
Toolbar	View Real-Time Motion

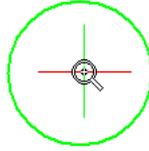
Command	Regen
Menu Bar	View Regen
Ribbon	View View Regen
Toolbar	View Regen

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

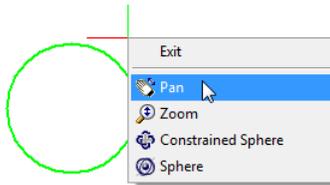
>>Press ENTER or Esc to complete, or right-click to display context menu...



- 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.)

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.



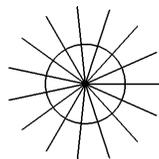
5. When the tree-circle is at a satisfactory size, press **Esc** to get out of real-time pan-and-zoom mode:
>>Press ENTER or Esc to complete, or right-click to display context menu....(Press **ESC**.)
If the circle looks like an octagon or diamond, select **View | Regen** to clean it up, as follows:
: _regen

Now that the circle looks rounder and larger, it is easier to work with.

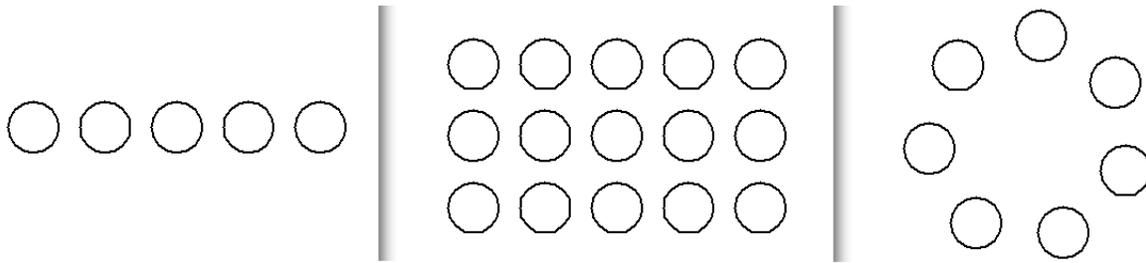
CREATING ARRAYS

Command ArrayClassic
Menu Bar Modify | Array | Array Dialog Box

With the circle in place representing the trunk, the tree branches can be added by an array of lines.



To create the array, use the **ArrayClassic** command to create the radiating lines. This command creates several types of arrays: linear, rectangular, and polar (circular). They are illustrated below.



Left to right: Linear, rectangular, and polar arrays.

(The Array command operates at the command line, but the ArrayClassic command displays a dialog box, which is easier to use!)

For this tutorial, you draw one line and then array it to create... more.

1. Enter the **Line** command:

: line

Start of line or [Follow] <Last point>: **cen**

of (*Pick center of circle*)

Set end point or [Angle/Length/Undo]: (*Pick point anywhere outside of circle*)

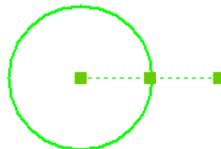
Set end point or [Angle/Length/Follow/Close/Undo]: (*Press Enter to end the command*)

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.



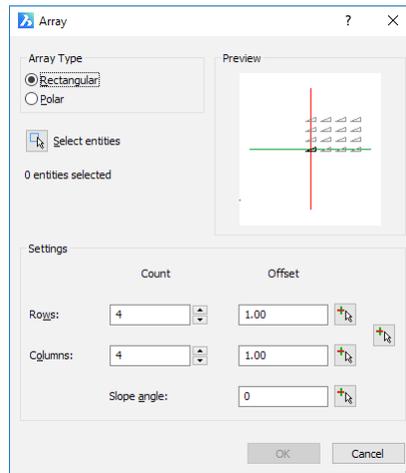
2. Select the line you just drew. Notice that it is highlighted.



3. Enter the ArrayClassic command:

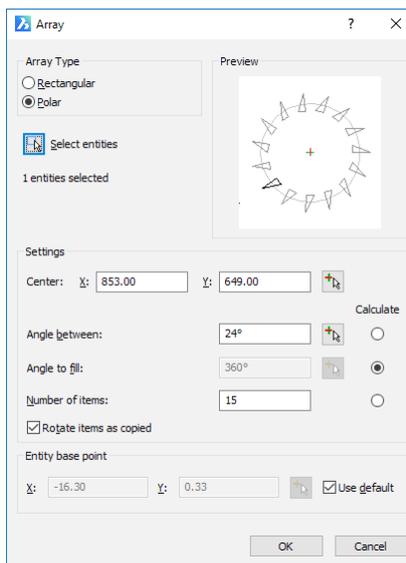
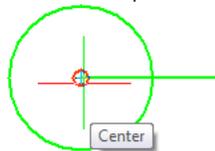
:arrayclassic

Notice the Array dialog box.

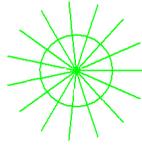


4. Entering array parameters in this dialog box is just like filling out a form:

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: (Choose the circle)
Number of items	15
Rotate items as copied	Yes



- Ignore the other settings (leave them at their default values). Notice that the preview window gives you can idea of what the array will look like. Click **OK**.



Making Blocks

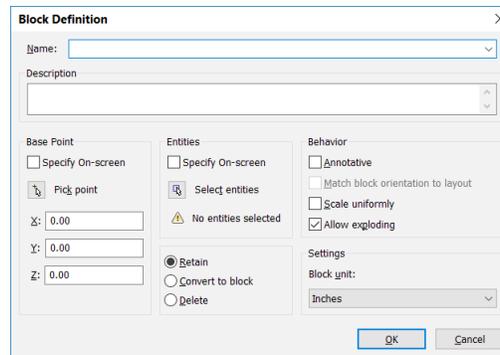
Command	Block
Alias	B
Menu Bar	Tools Create Block
Ribbon	Insert Blocks Create Block
Toolbar	Tools Block

CAD draws symbols more quickly and accurately than you can by hand. The key is to turn symbols into blocks, and then insert the blocks into drawings. In this section, you do this by adding tree blocks 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” — this is the precise point where the block is inserted in the drawing.

The center of the tree symbol is a logical spot base point, so use **CEN**ter entity snap:

Insertion point for new block: **cen**

Snap to centerpoint of: *(Pick the circle)*

- b. 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**

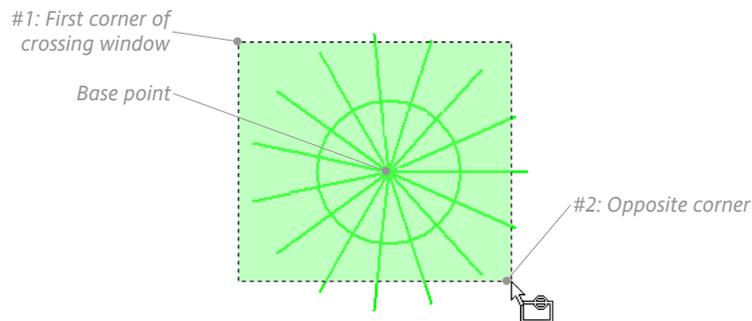
- c. You pick the two corners of a rectangle that encompasses the circle and lines making up the tree symbol:

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 (INSERT)

Command	Insert
Alias	I
Menu Bar	Insert Block
Ribbon	Insert Blocks Insert Block
Toolbar	Insert Block

With the bigger view in place, insert tree blocks in the drawing.

1. Start the **Insert** command:

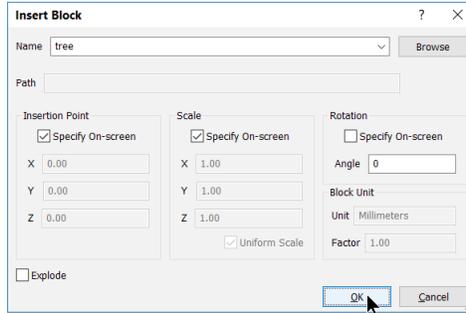
: **insert**

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 ✓(on)
Scale	Specify On-screen ✓(on)
Rotation	Specify On-screen (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.

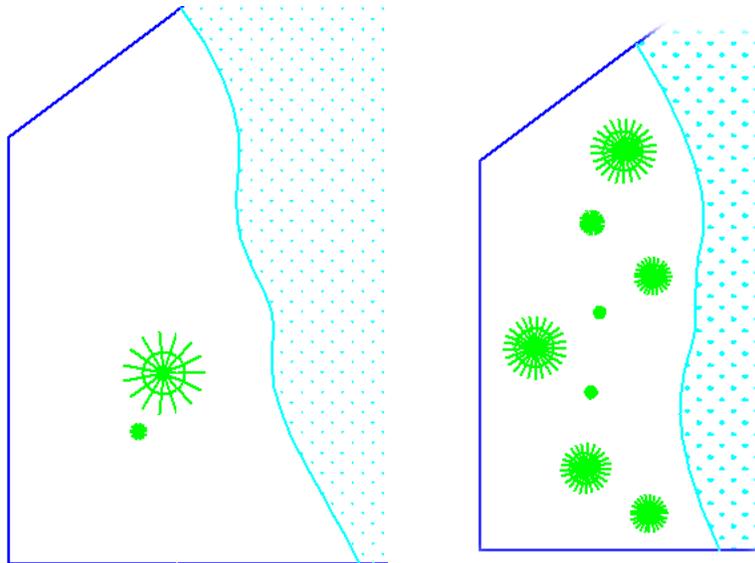
Insertion point for block: (Pick a point anywhere in the garden area)

4. The other item is the scale, which draws the block larger or smaller than the symbol original size:

Corner/XYZ/X scale factor <1.00000>: 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 enter a scale factor smaller than 1, such as 0.5, then the block is 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

5. 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.
6. 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
Alias	EL
Menu Bar	Draw Ellipse
Ribbon	Draw Draw Ellipse
Toolbar	Draw Ellipse

Command	Offset
Alias	O
Menu Bar	Modify Offset
Ribbon	Change Modify Offset
Toolbar	Modify Offset

Drawing the garden pond illustrates another pair of useful commands: **Ellipse** draws oval shapes and **Offset** adds parallel copies. The pond will be an oval shape, while the pond's edging will be made from a parallel offset.

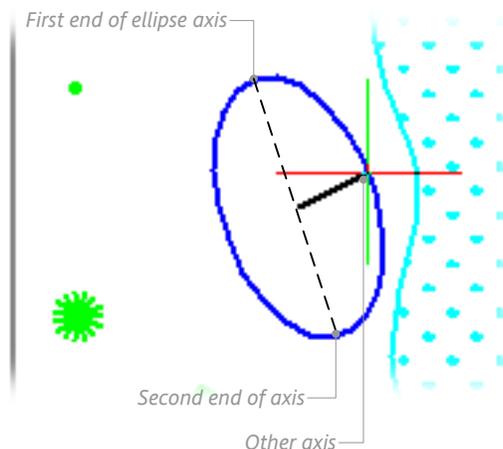
1. Switch to the **Pond** layer via the layer droplists.
2. The pond is to be an oval 4.5m tall by 1.5m wide (15 by 5 feet in imperial units). To draw the pond with the Ellipse command, follow these steps:

: **ellipse**

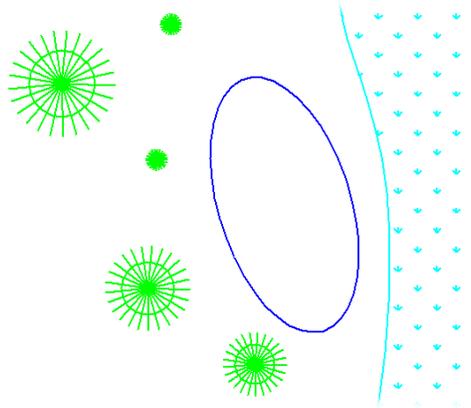
Pick the starting point of the ellipse anywhere in the garden area.

First end of ellipse axis or [Arc/Center]: c

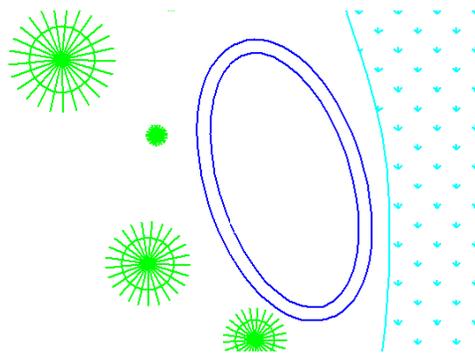
Center of ellipse: (Pick a point)



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**

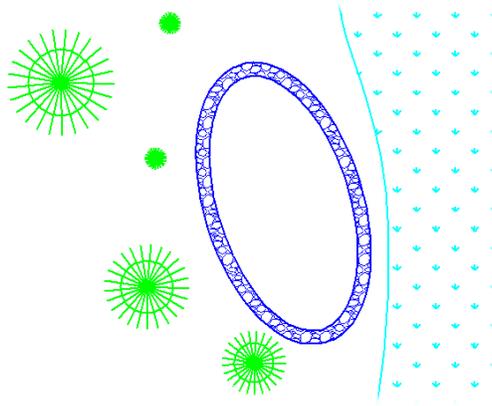


5. The pond is edged with rock. You could draw the edging of by repeating this command to draw a second, larger ellipse. The easier alternative is to use the **Offset** command, which creates precise parallel copies. Here, you use the command to create a concentric ellipse:
 - a. The offset distance is 0.3m (or one foot in imperial units):
 : **offset**
Enter offset distance or [Through point/Erase/Layer] **0.3**
 - b. Choose the ellipse; notice that only one entity can be offset at a time:
Select entity/subentity or [Exit] <Exit>: (Pick the ellipse)
 - c. Place the offset copy on the outside of the ellipse:
Select side for parallel copy or [Both sides/Multiple]: (Pick outside ellipse)
 - d. The command repeats itself to offset other entities. But to exit the command, press **Enter**.
Select entity/subentity or [Undo/Exit] <Exit>: (Press **Enter** to exit)



The Offset command also creates parallel lines, parallel polylines, and concentric circles and arcs.

If you wish, add the Gravel hatch pattern to the pond edging.

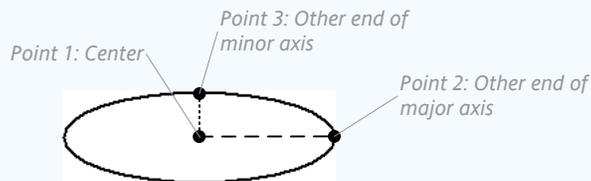


QUICK SUMMARY OF DRAWING ELLIPSES

BricsCAD provides three methods for drawing ellipses (ovals):

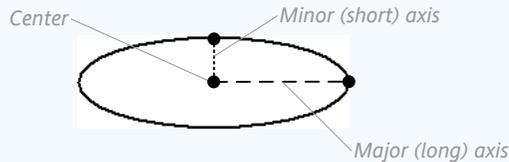
: **ellipse**

First end of ellipse axis or [Arc/Center]:



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.

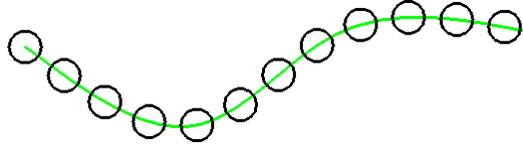


Arc — draws elliptical arcs.

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.

ARRAY PATHS

Another type of array can repeatedly place an object along a path. This is called a “path array,” appropriately enough. In the figure below, the path is in green while the arrayed entities are the black circles.



You use this construction technique to lay a path of paving stones from the house, across the lawn, and to near the pond.

As the 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). The paving stones are made of polygons (**Polygon** command), and then arrayed (**ArrayPath** command).

Drawing Splines

Command	Spline
Alias	spl
Menu Bar	Draw Spline
Ribbon	Draw Polylines Spline
Toolbar	Draw Spline

Splines were first used in ship building to make hulls from smooth flowing strips of wood. Hundreds of years later, we still use spline to draw smooth curves, with the **Spline** command. Follow along with these steps:

1. First, change the layer to “Road” using one of the methods you learned earlier. (My preference is to use the “clayer road” shortcut shown below.) This means the paving stones are placed on the same layer as the driveway.

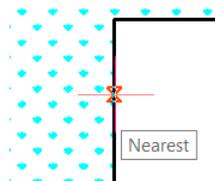
: clayer

New current value for CLAYER: **road**

2. Draw the guide path using a spline. If you ignore the command’s many options, then drawing a spline is as just about as easy as drawing a line:

: spline

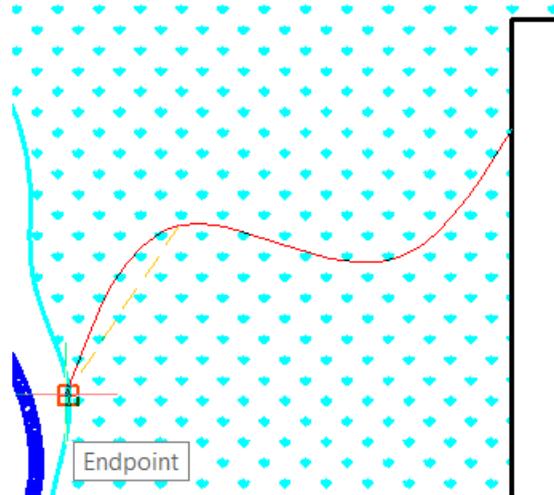
First point for spline: (Pick a point on the house outline, as show below)



Second point: (Pick a point along the Lawn)

Set next point or [Close/Fit tolerance]: (Pick another point along the Lawn)

Set next point or [Close/Fit tolerance]: (Pick a point at the edge of the Lawn near the pond, as shown below)



3. To end the command, press **Enter** three times, once for each prompt:

Set next point or [Close/Fit tolerance]: (Press **Enter**)

Starting point tangent point: (Press **Enter**)

Enter tangent for ending point: (Press **Enter**)

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 Spline 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:

4. To zoom into the spline, use the **Object** option of the **Zoom** command, as follows:

```
: zoom
```

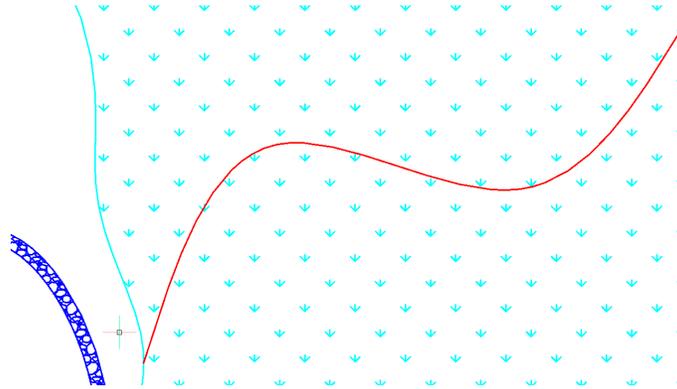
```
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/  
Window/Object] <Scale (nX/nXP)>: ob
```

```
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.

5. 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



QUICK SUMMARY OF SPLINE

: spline

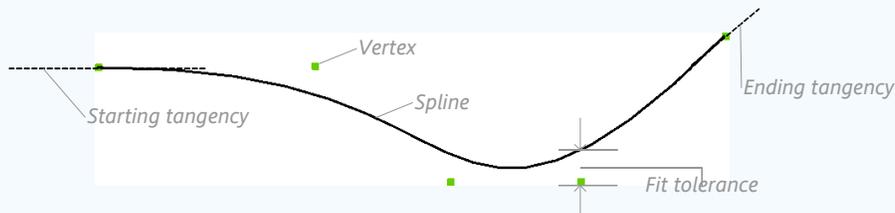
First point for spline:

Second point:

Set next point or [Close/Fit tolerance]:

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 — 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

Paving Stones from Polygons

Command	Polygon
Alias	pol
Menu Bar	Draw Polygon
Ribbon	Draw Polygons
Toolbar	Draw Polygon

6. 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; so follow along:
- The first thing you do is specify the number of sides. For a hexagon, you specify six sides:

: polygon

Polygon: Set number of sides or [Width of line/Multiple polygons] <4>: 6

QUICK SUMMARY OF POLYGON

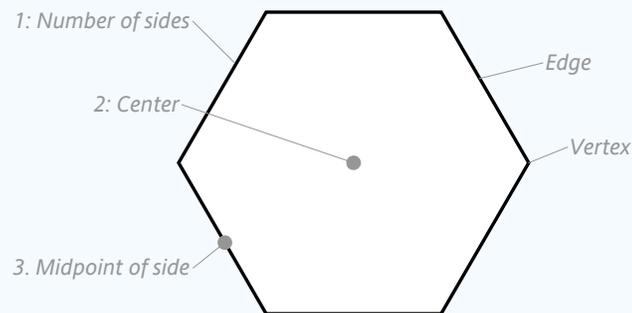
This command draws regular polygons, which means that every side has the same length. (To draw irregular ones, use another command such as Rectang or PLine.) The minimum and maximum sides is 3 and 1024.

: polygon

Polygon: Set number of sides or [Width of line/Multiple polygons] <4>:

Specify by: Set center of polygon or [specify by Edge]:

Specify by: Select midpoint of side or [specify by Vertex]:



Set number of sides — specifies the number of sides between 3 and 1024

Multiple polygons — repeats the command to draw additional polygons

Width of line — specifies the width of the lines, because polygons are made of polylines

Specify by edge — specifies the length of one side

Set center of polygon — specifies the center point of the polygon

Specify by vertex — specifies the location of a vertex (corner of the polygon)

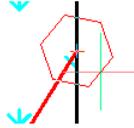
Select midpoint of side — specifies the midpoint of one of the sides

- b. Then you pick a center point for the polygon. For this tutorial, pick one end of the spline using END-point entity snap:

Specify by: Set center of polygon or [specify by Edge]: **end**
Snap to endpoint of: (*Pick end of the spline near the house*)

- c. 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: Select midpoint of side or [specify by Vertex]: (*Move cursor away from center of polygon*)



Arraying Along a Path

Command	ArrayPath
Menu Bar	Modify Array Path Array
Ribbon	Change Array Path Array
Toolbar	Modify Arrays

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.

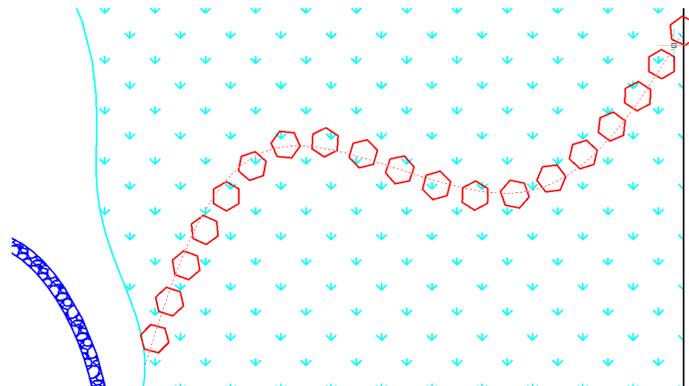
1. Start the **ArrayPath** command, and then select the polygon, as follows:

: **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*)

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.

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.

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]<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

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

Editing Associative Arrays

Command ArrayEdit

Menu Bar Modify | Array | Edit Array

One of the stepping stones sits on the edge of the house. When you select it to erase it, notice that the entire array of polygons is selected. (See figure below.) This shows you that associative

QUICK SUMMARY OF ARRAYEDIT

: arrayedit

Select associative array:

Select associative array — select any entity in the associative array; after you do so, the next prompt depends on the command used to make the array.

- ArrayPath command:

Source/REPlace/Method/Base point/Items/Rows/Levels/Align items/Z direction/RESet/<eXit>:

- ArrayPolar command:

Source/REPlace/Base point/Items/Angle between/Fill angle/Rows/Levels/ROtate items/RESet/<eXit>: (Select an option.)

- ArrayRect command:

Source/REPlace/Base point/Rows/Columns/Levels/RESet/<eXit>: (Select an option.)

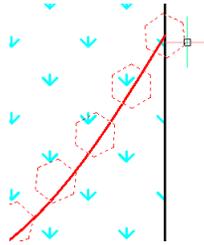
Many of the options are the same as with each array-creation command; those that are unique are described below.

Source — edits the source entity, following which all other entities in the array are modified; use the **ArrayClose** command when finished editing.

Replace — replaces some or all entities in the array with another entity.

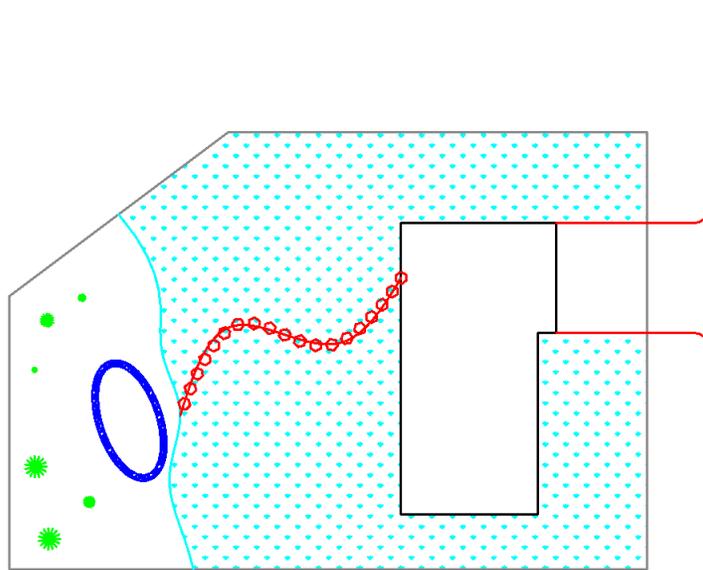
Reset — restores erased entities and item overrides.

arrays act like a single entity, kind of like trying to edit part of a block. (The entities in arrays made with the regular Array command are individual.)



Just as you can edit blocks with the RefEdit command, you edit associative arrays with the ArrayEdit command.

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.

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. But CAD software shows itself to be most powerful at effortlessly making changes, and flushing out the data held by drawings.

In this chapter, you learn how to change parts of the drawing and how to extract information.

IN THIS CHAPTER

- Modifying properties of entities
- Applying linetypes
- Changing the length of open and closed entities
- Understanding selection sets
- Finding information about entities in drawings
- Nudging entities

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

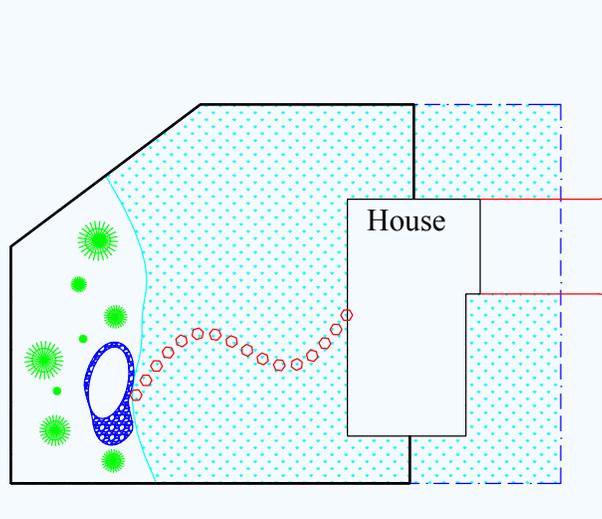
.lin Linetype definition file

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:

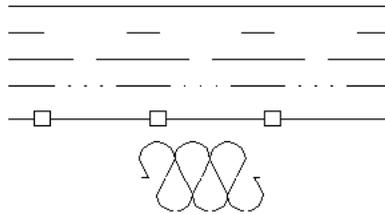


Tutorial-06 drawing file

Changing the Look of Lines

When you drew the lot lines in an earlier chapter, they were solid lines. Lot lines are, however, usually shown by a dashed pattern. Drafters use patterned lines to identify the meaning of lines, such as for hot water piping, insulation, or property lines, as in this case.

Just as BricsCAD comes with several hatch patterns, it also includes a number of line patterns made of dashes, dots, and more complex patterns. They are called “linetypes,” and some of them look like this:



To change a line from solid (called “Continuous” by BricsCAD) to dashed, you load a linetype definition into the drawing, and then select the line you want to have the new look.

LOADING LINETYPES

Command	Linetype
Alias	lt
Menu Bar	Tools Linetypes
Status Bar	Linetype

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. Remember that you can access them from my Dropbox folder: <https://www.dropbox.com/s/ft6gzwr408ndgd/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 [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>: **all**

Follow this by an 80% zoom that adds breathing space around the drawing (80% is the same as 0.8x):

: **zoom**

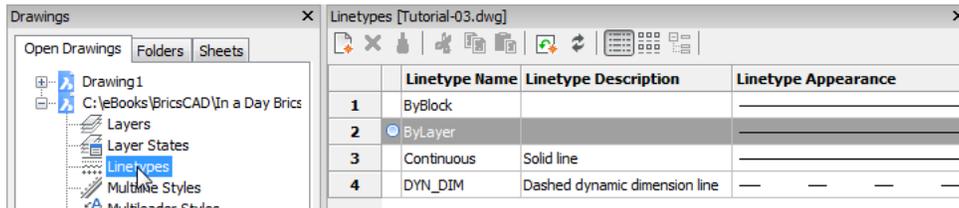
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>: **0.8x**

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.

Enter the linetype command:

: **linetype**

Notice that BricsCAD displays the Drawing Explorer:



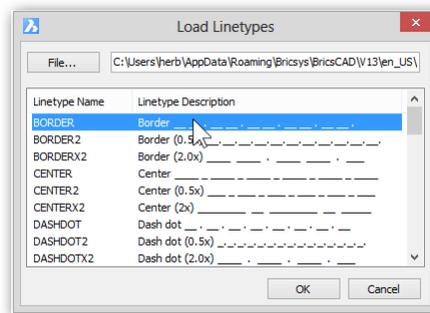
Notice that the drawing already contain some linetypes; all new drawings do. Here is what their names mean:

- ▶ **ByLayer** — entities take the linetype assigned by the layer. If a layer is assigned the Dashed linetype, then all entities on that layer are displayed with dashed linetypes.
- ▶ **ByBlock** — entities take the linetype defined by the block of which they are part
- ▶ **Continuous** — lines are solid, with no gaps
- ▶ **DYN_DIM** — lines with dashes; meant for use by dynamic dimensions

Although BricsCAD uses terms like *linetypes* and *line weights*, these apply to nearly any kind of entity, such as circles and arcs — not just lines. Linetypes and line weights are not, however, applied to text or 3D entities, because linetypes assign meaning to lines, which are not needed by text and 3D entities.

3. 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 you to a dialog box from which you load pre-defined linetypes.

Notice the Load Linetypes dialog box.



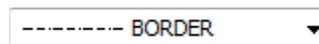
The dialog box lists in alphabetical order the names of all linetype definitions available to you. (They are stored in the *default.lin* file.) Scroll through the list to see what is available. Some linetypes have names that include a code:

- ▶ 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

4. Load the Border linetype like this:
 - a. Select **BORDER** from the Linetype Name list.
 - b. Click **OK** to close the dialog box.
5. You are back in the Drawing Explorer dialog box. The next step is to set the Border linetype as *current*. “Current” means that it is the working linetype: if you were to draw a line, it would now take on the Border linetype. Lines already in the drawing are not affected.
 - a. Under Linetype Name, click **BORDER** to select it.
 - b. Under the **Current** column, click the blank square so that the  blue dot appears. This means the linetype is now current.
 - c. Click **X** (in the upper right corner) to dismiss the Drawing Explorer dialog box.

	Current	Linetype Name
1	<input checked="" type="checkbox"/>	BORDER
2	<input type="checkbox"/>	ByBlock

If the Properties panel or the Entity Properties toolbar are open, notice that the default linetype changes to Border.

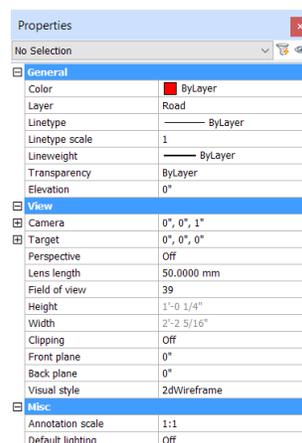


CHANGING PROPERTIES

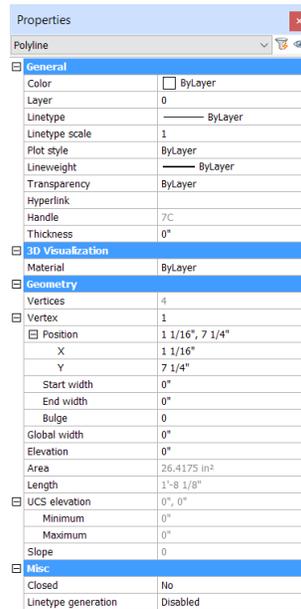
Command	Properties
Alias	pr
Menu Bar	Modify Properties
Right-click	Properties
Toolbar	Standard Properties

The Properties panel (or bar or palette) is excellent for changing several properties of one or more entities. It is probably one of the commands I used the most, because it provides three kinds of information:

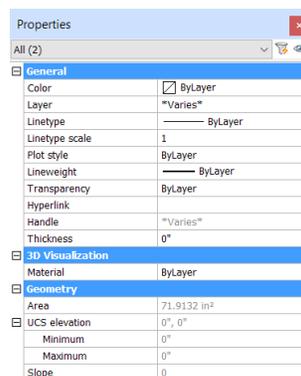
- ▶ **When No Entities are Selected** — The Properties panel reports the current default color, layer, linetype, and so on. The next entity you draw will take on these properties. For instance, the figure below shows that entities will be drawn with color ByLayer (red), on layer “Road,” and so on.



- ▶ **One Entity is Selected** — The Properties panel describes the properties of the entity that you selected. The General section is common to all entities; the Geometry section changes depending on the type of entity selected. In the figure below (at left), you see that a line is selected and that the Properties panel shows the properties of the line.



- ▶ **Two or More Entities Selected** — When two or more entities are selected, the Properties panel reports only those properties that the entities have in common. When properties are different between entities, the panel reports “*Varies*”, as shown below.



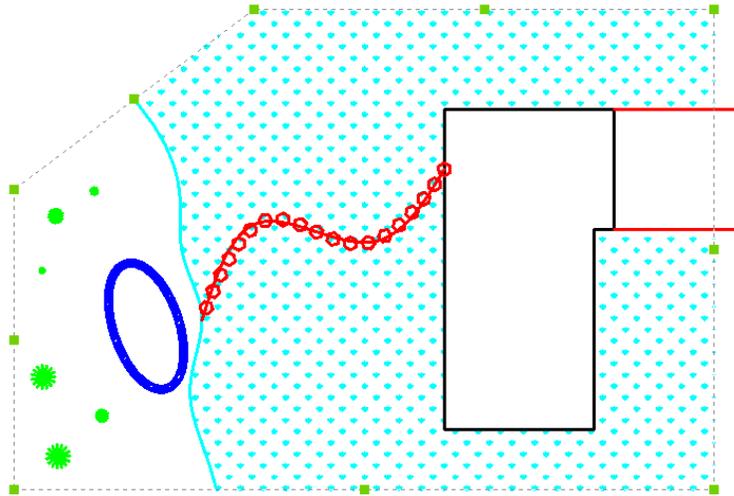
Use the Properties panel to now change the lot lines from ‘Continuous’ to ‘Border’, like this:

1. If the Properties panel is not displayed, enter the **Properties** command to open it now:

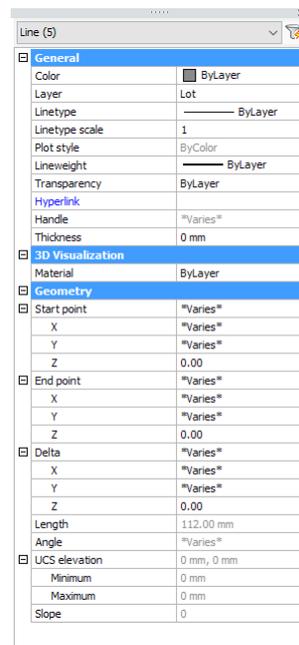
: **properties**

Notice that the Properties panel appears.

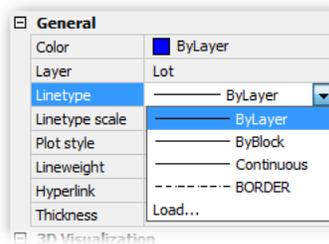
2. In the drawing, pick the five blue lot lines directly. As you pick them, notice that BricsCAD highlights them: they are shown as dashed lines, with green grip squares appearing on each line.



At the top of the Properties panel, notice that the list box reports **Line (5)**. This confirms that you selected five lines. The properties that you see listed belong to the five, such as layer “Lot” and linetype “Border.” Because more than one line is selected, many properties are reported as *Varies*, because these properties, such as the x and y coordinates, differ for each line.



- Next to **Linetype** in the Properties panel, click **Bylayer**. Notice that BricsCAD lists the names of the line-types in the drawing, ByLayer; ByBlock, Continuous, Dyn_Dim, and Border.



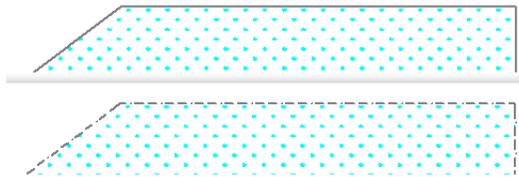
4. Select the **Border** linetype by clicking its name.
5. Press **Esc** to clear the grips.

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
Alias lts

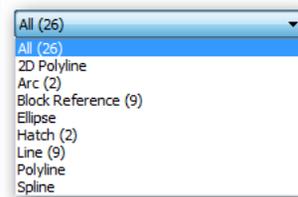
The reason linetypes sometimes look continuous is that they are sensitive to scale, just like hatch patterns. Sometimes the lines should show gaps, but don't, as shown below. The solution is to change the scale (size) of the linetype.



Top: Incorrect linetype scale; above: Correct linetype scale.

QUICK SUMMARY OF PROPERTIES PANEL

Properties describe the look and position of entities, such as 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 “Arc (2).”



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:

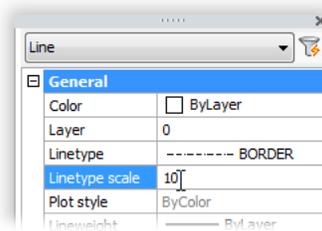
- ▶ Enter new values by typing them in
- ▶ Select preset values from droplists (when available)
- ▶ Click buttons (when available) to select values from drawings or dialog boxes

All linetypes start with a scale factor of 1.0, which usually is too small to display the pattern of dots and dashes correctly.

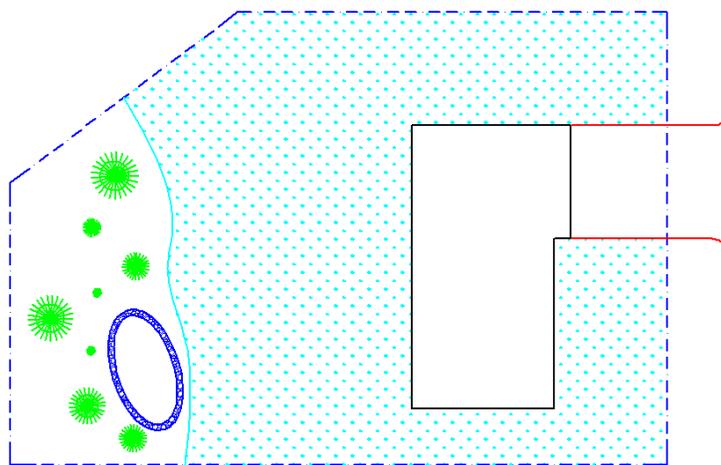
Warning! Setting linetype scale is one of the trickiest aspects of BricsCAD. Lintypes look continuous when the scale is too large — and when too small!

One method to change the scale of linetypes is to use **LtScale**; another is to use the Properties panel and its **Linetype Scale** field. I find using the Properties panel is the easier of the two methods:

1. Continuing in the Properties panel, click the field next to **Linetype Scale**.
2. Change 1.0 to something like **10**.



3. Check whether the linetype pattern becomes visible. If not, try another scale value, such as **0.01**.



4. It may be necessary for you to use the **Regen** command to fix the display, following a change in linetype scale.

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.

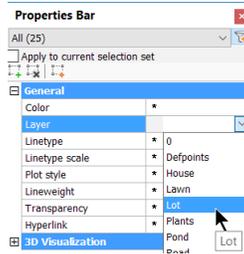
Selecting Entities by Their Properties

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 by their properties. For this, BricsCAD offers

the Quick Select mode, which isn't so much about selecting things quickly as selecting them by their properties.

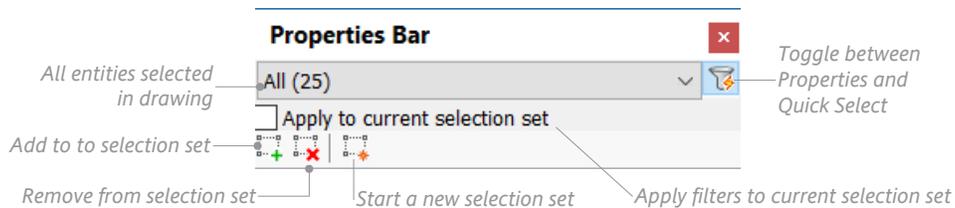
When using Quick Select mode, you modify filters that apply to the current selection set. The current selection set starts off as all entities in the drawing. A *filter* reduces the selection set. For instance, when you select “Red” as the **Color**, you filter out all entities that are not red. When you select “Lot” as the Layer, you remove all entities that are not on layer Lot.

1. Press **Esc** to ensure no entities are selected.
2. In the Properties pane, click the  **Quick Select** button. Notice that the panel changes its look to Quick Select mode:

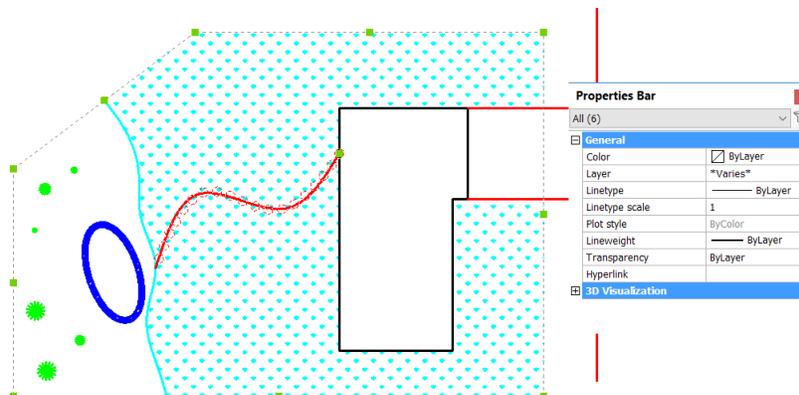


Notice the following about the new look:

- ▶ All entities are selected, as shown by “All (25)”
- ▶ Thee buttons populate the toolbar
- ▶ Basic properties show ‘*’ (asterisk), which means that all properties are selected.



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 the properties of just the lot lines, such as the linetype or color.



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	[pick]	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 or +	A or +	Enters add-entities mode
Remove or -	R or -	Enters remove-entities mode
Multiple	M	Select entities without highlighting; faster in complex drawings
Undo	U	Removes most recent selection group
Dialog	D	Displays the Entity Selection section of the Settings dialog box
End	press [Enter]	Ends entity selection
Cancel	press [Esc]	Cancels entity selection
Location	LO	Switches back to original select prompt
\		

...continued

...continued

SELECTION BY PROPERTIES

The **PRO**PERTIES 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

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.) These let 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
Aliases	len, editlen
Menu Bar	Modify Lengthen

Once you’ve drawn stuff in a drawing, the entities are not 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 entities, some only with closed ones, 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.)

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)

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.

Stretching the Pond

Command	Stretch
Alias	s
Menu Bar	Modify Stretch
Ribbon	Change Modify Stretch
Toolbar	Modify Stretch

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**

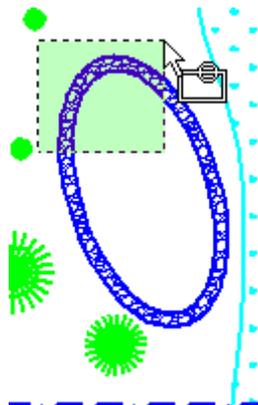
The Stretch command has an odd limitation, in that entities are initially selected only by two methods, both of them *crossing* selection modes. You can specify either Crossing (which forms a rectangular selection window) or CPolygon, which forms a polygonal selection window. The reason for the limitation 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.

Some entities cannot be stretched, like text and 3D solids. When they cross the selection window, the Stretch command moves them.

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)*

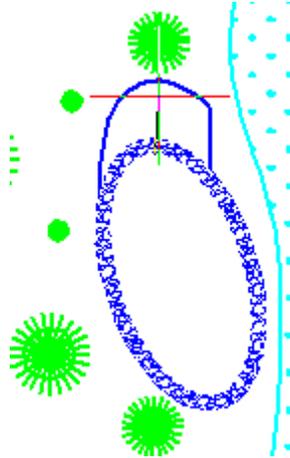


C is short for “crossing,” an entity selection mode similar to the window mode you used earlier with the Zoom command. In this case, BricsCAD selects all entities within the selection rectangle and all entities crossing or touching the rectangle.

- 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)

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.

- 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.

- 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
Alias	m
Menu Bar	Modify Move
Ribbon	Change Modify Move
Toolbar	Modify Move

If the pond isn't exactly where you want it, you can relocate it with the **Move** command.

- 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.

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.

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*.

NUDGING ENTITIES

(new to V19) Some parts of CAD call for precision, such as the legal definition of the property line and the location of the house. Other parts, however, call for artistry, such as the border between the lawn and garden areas. The only thing that mattered about the polyline curve was that it look attractive.

Similarly, the trees placed in the garden were placed arbitrarily. Now, a landscape designer may want to move them around a bit to create attractive groupings. For this, BricsCAD has the nudge function.

Nudge is a keyboard function that works like this:

1. Select one or more entities.
2. Hold down the Ctrl key.
3. Use the cursor keys on the keyboard to nudge the entity(ies) around — up, down, left, and right.

Each time you tap the cursor key, BricsCAD reports the distance moved:

```

NUDGE: offset along the Y axis of the UCS: 0.20000000
NUDGE: offset along the X axis of the UCS: 0.20000000
NUDGE: offset along the X axis of the UCS: -0.20000000
NUDGE: offset along the Y axis of the UCS: -0.20000000

```

The distance is a function of the zoom level. The closer in you are to the drawing, the shorter the zoom distance.

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.

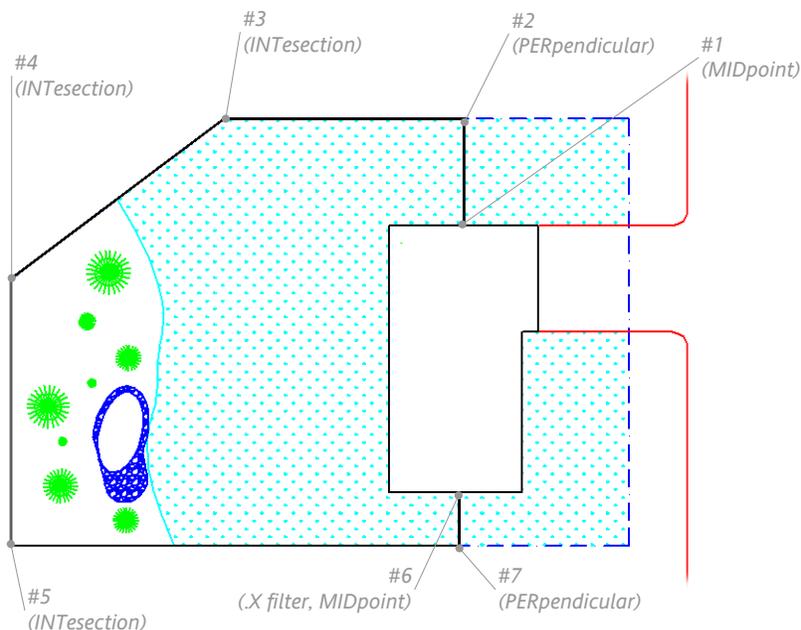
The fence is drawn as a polyline, 0.1m-wide (4"-wide). Along the way, you'll use a variety of entity snap modes and tracking mode.

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: mid  
Snap to midpoint of: (Pick upper house line)  
Current line-width is 0
```
5. To change the width of the polyline from zero to 10cm (4"), use the PLine command's **w** (Width) option, as follows:
Set next point or [draw Arcs/Distance/Follow/Halfwidth/Width]: **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:

```
Set next point or [draw Arcs/Distance/Follow/Halfwidth/Width]: .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:

```
Set next point or [draw Arcs/Distance/Follow/Halfwidth/Width]: (Press ENTER to exit the command)
```

LISTING DATA

Command	List
Alias	li
Menu Bar	Tools Inquiry List Entity Info
Ribbon	Tools Inquiry List
Toolbar	Inquiry List

Command	Dist
Alias	di
Menu Bar	Tools Inquiry Distance
Ribbon	Tools Inquiry Distance
Toolbar	Inquiry Distance

Now that you've drawn the fence, you can use the **List** command to tell you its length.

1. Enter the List command:

: list

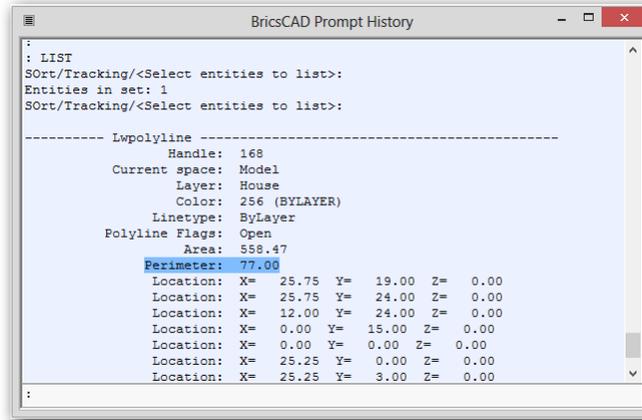
Sort/Tracking/<Select entities to list>: (Pick the fence's polyline)

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").

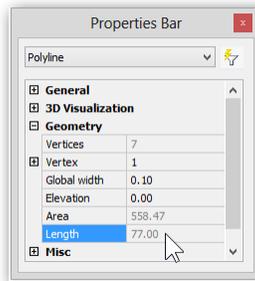
The total length of the polyline is shown in the middle of the listing:

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:**

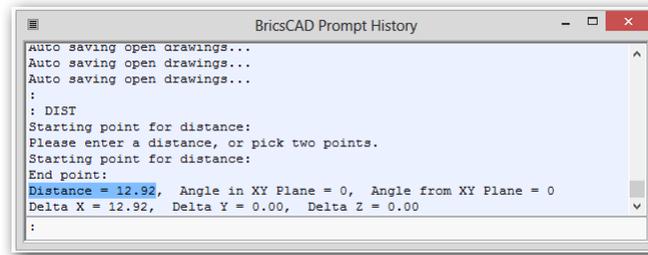
: _dist

Starting point for distance: **nea**

Snap to nearest of: (Pick inside edge of pond.)

End point: **per**

Snap to perpendicular of: (*Pick house wall.*)



```
BricsCAD Prompt History
Auto saving open drawings...
Auto saving open drawings...
Auto saving open drawings...
:
: DIST
Starting point for distance:
Please enter a distance, or pick two points.
Starting point for distance:
End point:
Distance = 12.92, Angle in XY Plane = 0, Angle from XY Plane = 0
Delta X = 12.92, Delta Y = 0.00, Delta Z = 0.00
:
```

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.

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. In this chapter, you learn how to add these finishing touches to the drawing.

IN THIS CHAPTER

- Learning about drawing scales
- Placing text in drawings
- Defining styles and justification modes for text
- Understanding annotative scaling
- Editing text
- Attaching dimensions to entities
- Placing leaders and multileaders

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

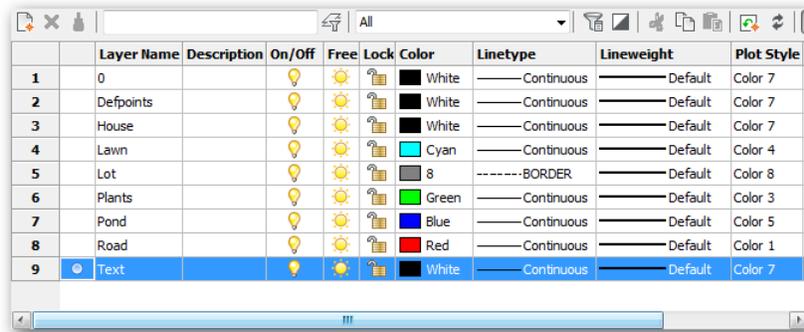
dims	Dimensions
dimvars	Dimension variables
dimstyle	Dimension style

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 this exercise, take the following 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 the files from my Dropbox folder: <https://www.dropbox.com/s/ft6gzw3r408ndgd/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 “Text” name.

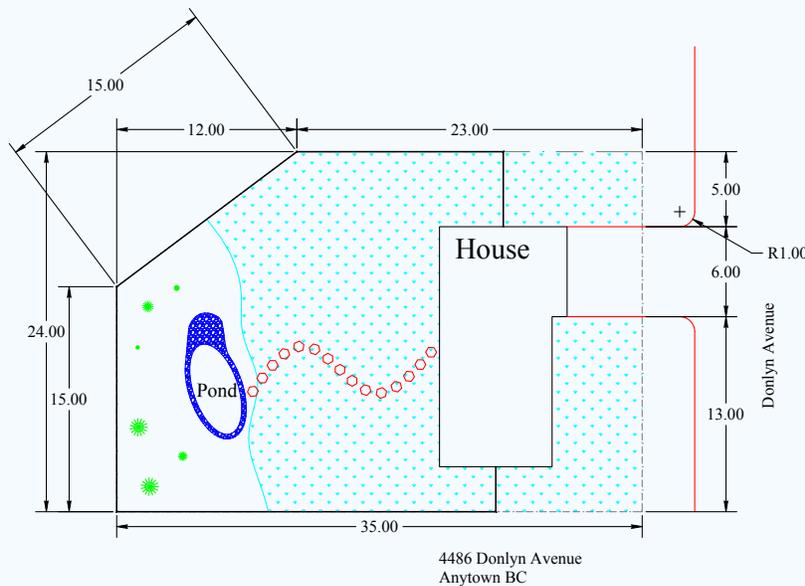


	Layer Name	Description	On/Off	Free	Lock	Color	Linetype	Lineweight	Plot Style
1	0					White	Continuous	Default	Color 7
2	Defpoints					White	Continuous	Default	Color 7
3	House					White	Continuous	Default	Color 7
4	Lawn					Cyan	Continuous	Default	Color 4
5	Lot					8	-----BORDER	Default	Color 8
6	Plants					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:

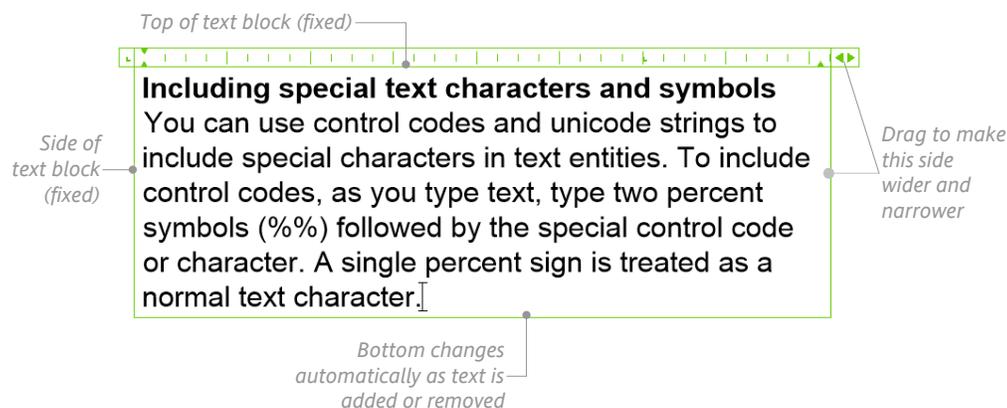


Adding Notes to Drawings

Command	MText
Aliases	MT, T
Menu Bar	Draw Multiline Text
Ribbon	Draw Text Multiline Text
Toolbar	Tools Multiline Text

You add callouts with the **MText** command, short for “multiple-line text” and known also as *paragraph text*.

BricsCAD places multiline text in a rectangle, much like page design software does. The rectangle is shown in green.



Text is constrained by the rectangle’s top and sides so that you can force the text to fit specific areas in drawings, such as title blocks and note blocks. As you add text, it flows down; there is no bottom to the rectangle.

You define the size of the rectangle with just two picks at opposite corners, and later you can resize the rectangle at any time — narrower or wider. You can move the block of mtext to any location in the drawing.

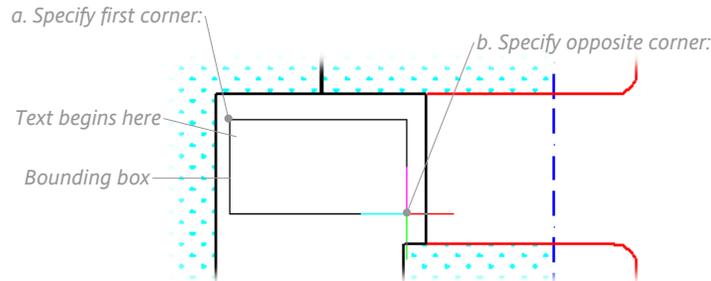
As well, you can format individual words and letters of the text with a variety of fonts and colors, and apply effects such as **boldfacing** and *italics*, just as in a word processor. Special symbols can be included, such as the center line symbol and the degree symbols. You can define margin indent (such as for the starts of paragraphs) and where tabs are set.

All this makes the MText command flexible, but more complex than the simpler and less capable Text command you learn about later.

Place some mtext in the drawing now:

1. Start the **MTtext** command:
: **mtext**
2. Notice that BricsCAD wants from you a starting point. This is one of the two picks that define the initial size of the text rectangle. Pick the first point 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. As you do, notice that a rectangle is formed. Pick a second point to define the width of the text:
Select Opposite corner for block of text or [Justification/Rotation angle/text Style/text Height/Direction/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; it is otherwise neither displayed nor printed.

When you start typing the text you want in the boundary box, BricsCAD will start at the top of the box, and the right edge of the box will determine where the text *wraps* — where the text flows onto the next line. (For right-to-left text like Hebrew, the *left* edge determines the word wrap location.) This is like setting margins in 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 you want the text to flow upwards, such as with revision notes; you change the direction of the flow with the command's Direction option.)

HOW TO DETERMINE 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 on the next page, *Quick Summary on Calculating Text Heights*. Based on the size of this drawing, the text height needs to be 400mm tall (or 18" in imperial units).

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:

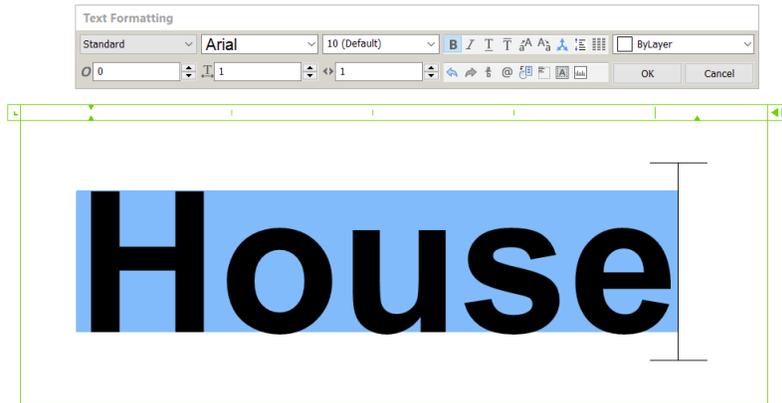
$$0.125" \times 142 = 18"$$

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>.

Continue with the MText command to specify the text height and other font parameters:

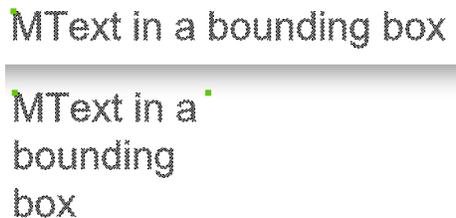
4. After picking the second point, BricsCAD displays the Text Formatting dialog box, in which you specify font settings, and then enter the text. First, change the height, as follows:
 - a. In the Font droplist, ensure “Arial” is specified.
 - b. In the **Height** droplist, highlight 0.2, and then enter **0.4** (a.k.a. 400mm).
 - c. Press **Enter**.



5. Type **House**, and then press **Enter**. Notice that the text appears in the drawing.
6. Click **OK** to exit the MText command.

You 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

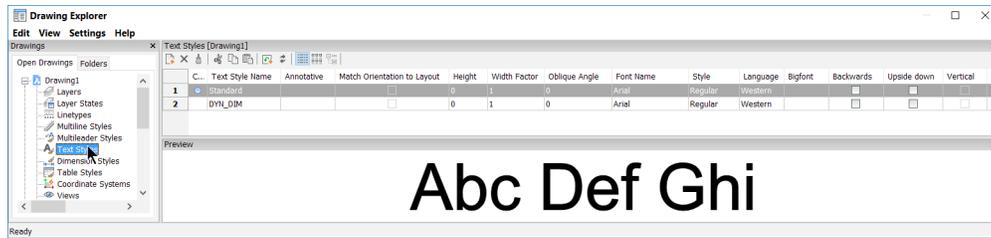
Command	Style
Aliases	st, ddstyle, expstyle, expstyles, expfonts
Menu Bar	Tools Drawing Explorer Text Styles
Ribbon	Home Settings Drawing Explorer Text Styles

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, and this is done with the **Style** command.

1. Enter the **Style** command. Notice that the Drawing Explorer dialog box appears on the screen.

: style



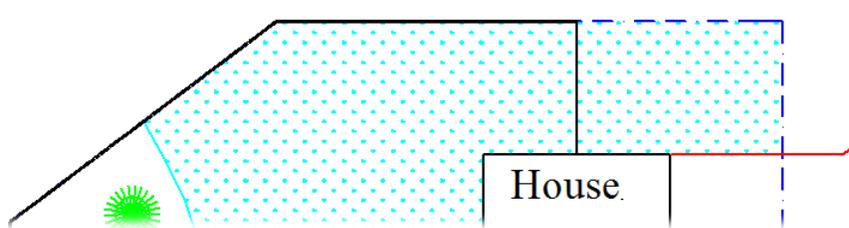
2. Make the following changes to the Standard style:

Property	Old Value	New Value
Font Name	Arial	Times New Roman
Height	0	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.

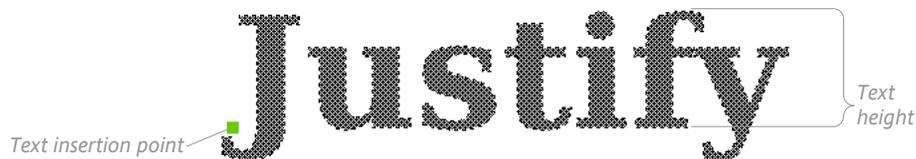
The height of **0** has a special meaning: it means that the text height is not predefined and so must be specified during the Text command.

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.

Here is how BricsCAD measures the height of text: it starts at the baseline and then goes to the top of the ascenders.



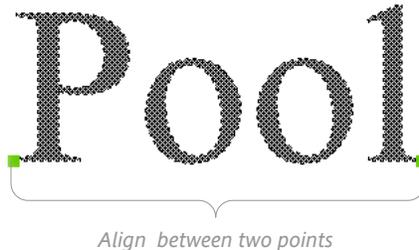
SIMPLE TEXT

Command	Text
Alias	TX
Menu Bar	Draw Text
Ribbon	Draw Text Text
Toolbar	Draw Text

BricsCAD has a second command for placing text: **Text** places a single line of text each time you use it. I find it is easier to use than MText for small amount of text, and is ideal for placing lots of bits of text all over drawings, as needed.

To try out the Text command, first zoom in on the pond with **Zoom Window**.

1. Now that you see the pond area more clearly, start the **Text** command:
: text
2. Select the **Align** justification mode, which places the text fitted between two points that you pick:

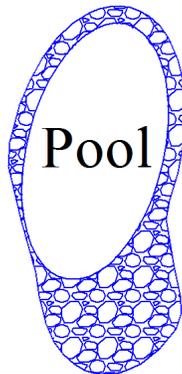


Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: a
Text start point: (Pick one end of the pond)
Text end point: (Pick other end of the pond)

Because the Align option's pick points define the width and the angle of the text, Text doesn't need to ask you for the rotation angle, as the command normally would with other justification options.

3. The Text command goes straight to the prompt:
Text: **Pool**

Notice that the word "Pool" is drawn with the Times New Roman text font, as specified by the Standard style.



4. Press **Enter** at the "Text:" prompt to exit the command.
Text: (Press ENTER)

Changing Text

Command DdEdit
Alias ED
Shortcut *double-click text*

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 a better way/

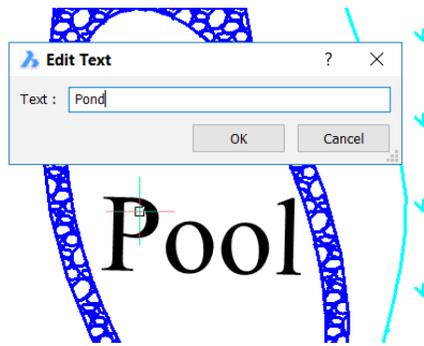
QUICK SUMMARY OF TEXT JUSTIFICATION MODES

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)

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.

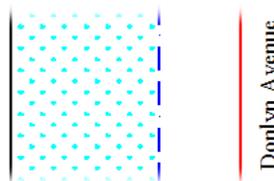


PLACING ROTATED TEXT

Now we’ll add some more text. This time you place the text sideways to see that BricsCAD can do that. 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*)



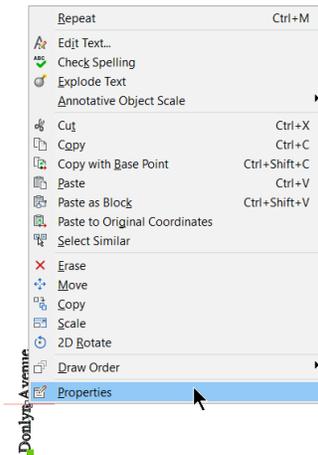
By specifying a rotation angle of 90 degrees, BricsCAD draws the text sideways. You can, of course, place text at any angle — from 0 degrees to 359 degrees.

3. If you specified the wrong rotation angle, you change that after the fact. Use the **Zoom Object** command to take a closer look at the text.

- Now that you are zoomed in, pick the text...

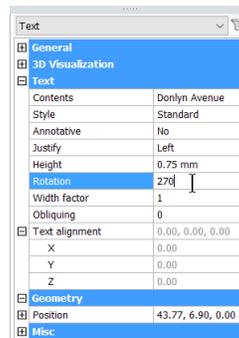
: (Pick text)

..and then right-click the text. From the shortcut menu, choose **Properties**.



- In the Properties palette:

- Look for **Rotation** in the Text section.
- Change 90 to **270**, and then press **Tab**.



- Notice that BricsCAD rotates the text by 180 degrees. Press **Esc** to exit grips editing.

PLACING MULTIPLE LINES OF TEXT

Now you use the Text command to place more callouts throughout the drawing. First, though, use **Zoom All** to see the entire drawing, and then follow these steps:

- 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)

- 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** to end the command)

Notice how BricsCAD places the second line of text precisely below the first.

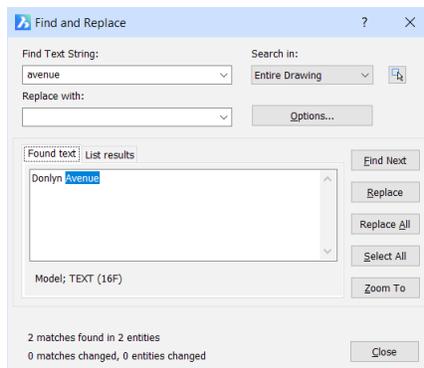
4. Repeat the command to add labels to trees, such as Birch, Aspen, Yellow Pine, or Western Red Cedar.

At this point, it is a good idea to save your work using the **Save** command.

FINDING AND REPLACING TEXT

Command	Find
Keyboard Shortcut	Ctrl+F
Menu Bar	Edit Find
Ribbon	Draw Text Find

To find text in drawings and/or replace it with different text, use the **Find** command. This displays the following dialog box.

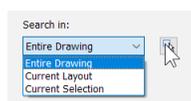


In the **Find Text String** field, enter the text for which you are looking, and then click **Find Next**. The Found Text tab displays the current instance of found text, while the List Results tab shows all instances:

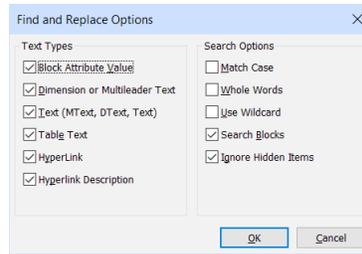
Location	Object Type	Text
Model	TEXT (16F)	Donlyn Avenue
Model	TEXT (170)	4486 Donlyn Avenue

When there is too much text in the drawing, you can narrow the search a couple of ways:

- ▶ To search only a specific area of the drawing: click the **Search In** droplist and then choose Current Selection, Current Layout (model or other layouts), or the Entire Drawing. To create a current selection, click the **Select Entities** button.



- ▶ To search only a specific kind of text, click the **Options** button, and then choose the kind of text.



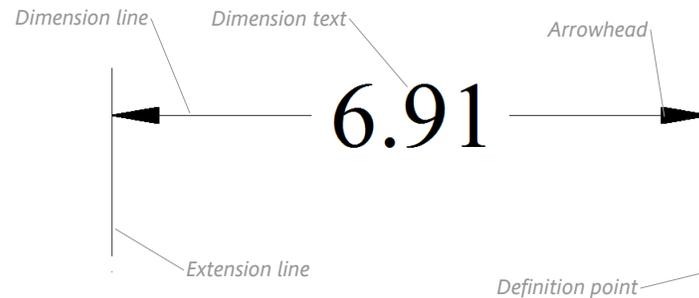
Text Types	Description
Block Attribute Value	Attribute text stored in blocks
Dimension Annotation Text	Text in dimensions, including leaders
Text (MText, DText, Text)	All other kinds of text, including field text
Table Text	Text found in the cells of tables
Hyperlink	Text in URLs (Web links)
Hyperlink Description	Text found in the descriptions of hyperlinks

And then there are these additional options for narrowing down the search further:

Match Case	Text must match the pattern of upper and lower case
Whole Words	Text has spaces or punctuation on either side, not part of larger words
Use Wildcard	Any text matching wildcards, such as * = any number of characters ? = any single character
Search Blocks	Searches for text inside of blocks
Ignore Hidden Items	Does not search for text on hidden and frozen layers

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"). The names of elements employed by all 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 set to reasonable values, for both metric and imperial units.

The settings are controlled by *dimensioning variables*, or "dimvars" for short. You can change dimvars in two ways: with the Dimension Styles section in the Drawing Explorer dialog box, or by directly entering their names at the ':' prompt. In the following tutorials, you use both methods.

BricsCAD draws all of the components of a dimension automatically: both extension lines, the dimension line, both arrowheads, and the dimension text.

PREPARING DRAWINGS 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**

The *height* of dimension text is controlled by **Style**, and is unaffected by the **DimStyle** scale factor.

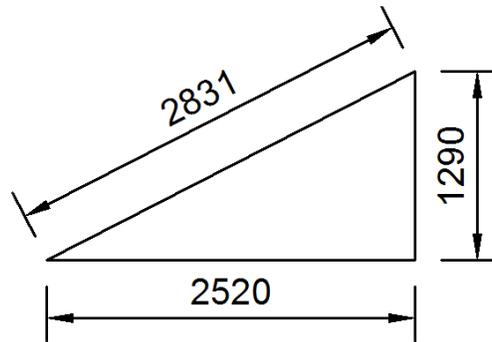
3. Create a new layer called **Dims** (black color, continuous linetype), and then make it current.
4. If you wish, freeze unnecessary layers, such as Text, Lawn, Plants, and Pond. This keeps them from getting in the way. (I left some of these layers on for the illustrations below so that it is easier 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
Aliases	dli, dimlin, dimhorizontal, dimrotated, dimvertical
Menu Bar	Dimension Linear Dimension
Ribbon	Draw Dimensions Linear
Toolbar	Draw Leaders Linear
Command	DimContinue
Aliases	dco, dimcont
Menu Bar	Dimension Continuous
Ribbon	Draw Dimensions Continuous
Toolbar	Dimension Continuous

In this tutorial, you 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 how you pick points in the drawing. Pick the first point, then move the cursor, and you see the dimension jumping between horizontal and vertical:

- ▶ **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.

Start dimensioning, like this:

1. Begin the DimLinear commands.

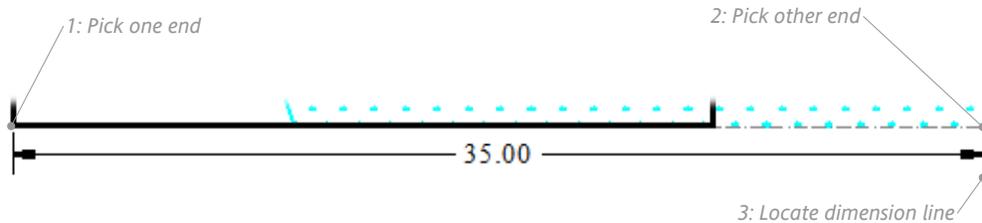
```
: dimlinear
```

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 two pick points, and then places all the elements of the dimension for you automatically, including determining the length between the two extension lines.

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.

Try another horizontal dimension of the top lot line. This time, use BricsCAD's *entity dimensioning* mode, in which you pick the entity and BricsCAD dimensions it. The method uses just two picks, instead of three.

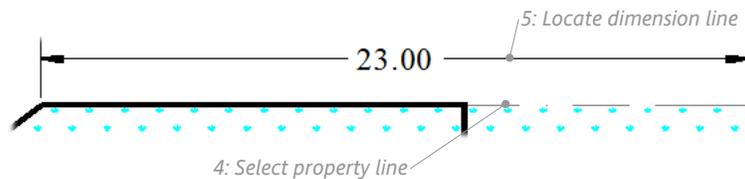
3. Press the **Spacebar** to repeat the **DimLinear** command:

```
: (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)



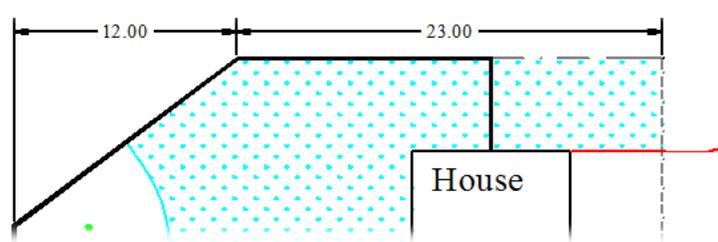
4. You can continue drawing horizontal dimensions with the **DimContinue** command. This is another way to dimension with fewer picks. When there is a lot of dimensioning to do, you want to minimize the number of picks! From the **Dimension** menu, select **Continue**:

```
: _dimcontinue
```

Continue: ENTER to select starting dimension/<Origin of next extension line>: (Pick #6)

- 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 I noted earlier, the DimLinear command also draws vertical dimensions. When you move the cursor in a vertical direction, BricsCAD knows to draw the dimension vertically.

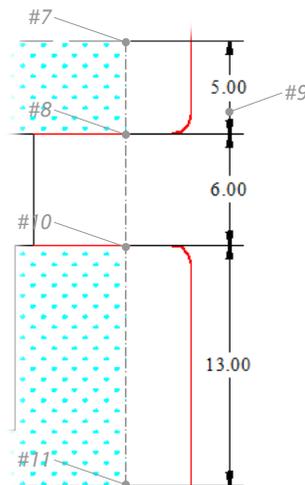
- To draw vertical dimensions, select **Dimension | Linear**, and then follow the numbered pick point from the illustration below.

: **_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)



- 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
Aliases	dba, dimbase
Menu Bar	Dimensions Baseline
Ribbon	Draw Dimensions Continue Baseline
Toolbar	Dimensions Baseline

A variation on the DimContinue command is the **DimBaseline** command. Rather than continuing a dimension from the *previous* extension line, DimBaseline continues dimensions from the *original* extension line (a.k.a. “the base line”). To see how it works, first place a vertical dimension:

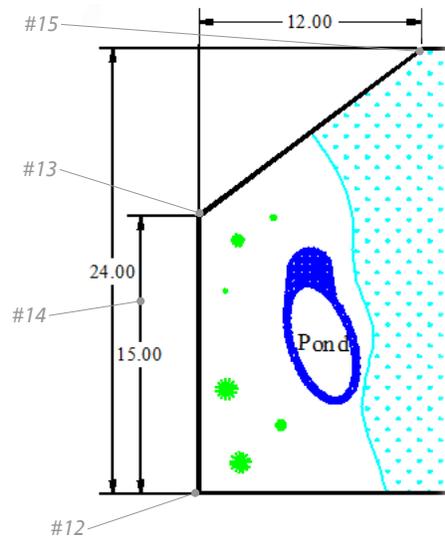
3. With the **DimLinear** command, place the first extension line at #12. Place the second extension line at #13, and then the dimension line at #14.
4. Now try out the DimBaseline command.

: 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
Aliases	dal, dimali
Menu Bar	Dimension Aligned Dimension
Ribbon	Draw Dimensions Aligned
Toolbar	Dimension Aligned Dimension

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.

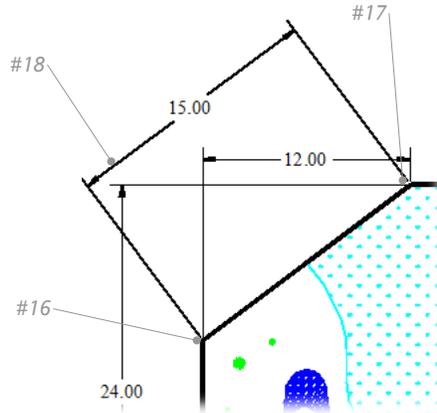
1. Enter the **DimAligned** command:

: **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
Aliases	dra, dimrad
Menu Bar	Dimensions Radius
Ribbon	Draw Dimensions Circle Radius
Toolbar	Dimensions Radius

So far, all dimensioning commands have presented pretty much the same series of prompts. Now here is one that's different. The **DimRadius** command dimensions arcs and circles. The related DimDiameter command places dimensions across diameters of circular entities.

2. (Zoom in for a closer look, if necessary.) Enter the **DimRadius** command, and then follow the prompts:

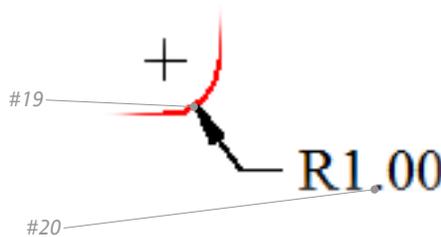
: **dimradius**

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.



3. Save your work, and then print out a copy.

DIM, THE SUPER DIMENSION COMMAND

Command	Dim
Alias	dimension

(NEW TO V19) I've had you using individual dimensioning commands, such as for linear and radial dimensioning, to understand how they work. Once you are expert at how they work, I suggest that you switch to Dim, the super dimensioning command.

It is “super” because

- ▶ It handles nearly all forms of dimensioning in a single command (just MLeader is missing)
- ▶ It knows the entity under the cursor and so draws the correct dimension for it
- ▶ It operates in entity mode, so you make only two picks: (1) the entity and (2) the dimension line
- ▶ It repeats itself automatically, so you don't need to pause to reload the appropriate command

When you enter the command, try not to be overwhelmed by the many options it presents, the longest of any command in BricsCAD:

```
: dim  
  
Dimensioning command [HORizontal/VERtical/ALigned/ANGular/Leader/OBLique/ROtated/CEnter/  
Diameter/RADius/Baseline/COntinue/ORdinate/Position/DIStribute/UPdate dimensions/varia-  
ble SStatus/OVerride/SEttings.../LAYer]:
```

Yip, that's 20 options. I counted them for you. The options are there in case you want to override its automatic entity-detection mode (the first 13 options), or else to edit an existing dimension. You've already used most of the dimension drawing commands, and so here is a summary of the editing options:

Dim Command Option	Description
Position	Repositions dimension text anywhere along the dimension lines, as well as by: specify Angle relative to dimension line Left justify along dimension line Center justify Right justify REstore text to original position
DIStribute	Spaces parallel or concentric dimensions by a specified distance
UPdate dimensions	Applies the current dimstyle to selected dimensions
variable SStatus	Lists the names of all dimvars and their current values
OVerride	Changes the value of a dimvar
SEttings...	Displays the Dimension Styles section of the Drawing Explorer
LAYer	Changes the name of DimLayer, the layer on which dimensions are placed

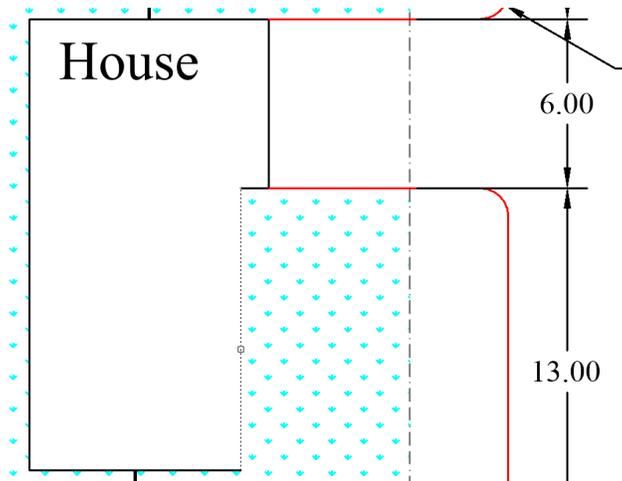
Let's use Dim to dimension a line and an arc.

1. Start the **Dim** command:

```
: dim
```

Dimensioning command: *(Move cursor over an entity)*

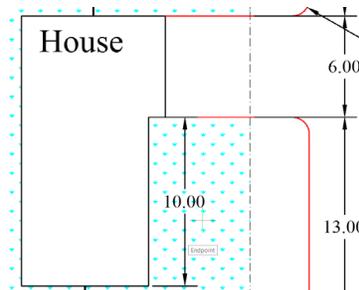
2. Move the cursor over one of the lines making up the outline of the house. Notice that the line becomes dashed, as BricsCAD tells you it has located it.



In the command line, BricsCAD prompts you to click:

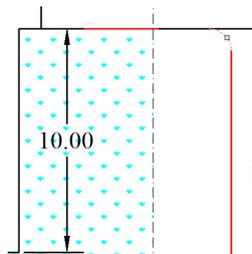
Select entity to dimension: *(Click on the vertical house line)*

3. Notice that BricsCAD draws the dimension. Now you need to position it:
Location of dimension line [Angle/Text]: *(Move the cursor to position the dimension line)*
(The Angle and Text options let you set an angle for the text, and to edit the text.)
4. Click to locate the dimension line.

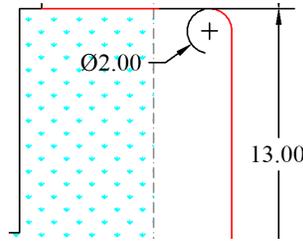


5. Notice that the Dim command is ready for another dimension. Now move the cursor over to one of the arcs of the driveway's curb return:

Dimensioning command: *(Move cursor over the arc)*



6. As before, click on the arc, and then position the text at the end of the leader line:
Select arc or circle to specify diameter or [Radial/Angular]: (*Click on arc, and then position the leader*)
Dimension text: = 2.00



(The **Radial** option switches between radius and diameter dimensions, while the **Angular** option switches to 3-point angular dimensioning.)

9. Press **Enter** to end the Dim command:
Dimensioning command: (*Press Enter to end the command*)

Annotatively-Scaled Text and Dimensions

In this and the previous chapter, you needed to work out the scale factor for *annotative* entities, specifically 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.

Sometimes, however, the paper size does change, such as sometimes from A- (A4) to D-size (A1) paper. You need a handy way of changing the size of annotative entities, preferably automatically. BricsCAD provides it 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 then only in three areas of drawings:

- ▶ **Annotative entities** — whose size needs to change according to the output size
- ▶ **Viewports** — whose scale factor is related to the output size
- ▶ **Plots** — whose scale factor affects the output size

To understand this, I have put together a rule:

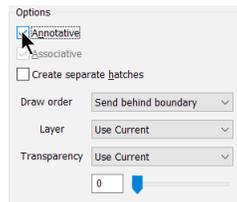
$$\text{Annotative scale} = \text{Viewport scale} = \text{Plot scale}$$

The scale at which you plot the drawing determines the scale factor of annotative entities. (If your drawing uses viewports, the same scale factor applies.)

To use annotative scaling, you have to turn it on in styles by clicking the Annotative item in styles for text, multilines, dimensions, multiline leaders, and blocks. Below, I clicked the Annotative property in the Drawing Explorer for text styles; I happened to name the text style “Annotative.”

	Current	Text Style Name	Annotative	Match Orientation to Layout	Height	Width Factor	Oblique Angle	Saved Font	Four
1		DYN_DIM	<input type="checkbox"/>	<input type="checkbox"/>	0	1	0	Arial	bt.sh
2		Standard	<input type="checkbox"/>	<input type="checkbox"/>	0.75	1	0	Times New Roman	times
3		Annotative	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.75	1	0	Times New Roman	times

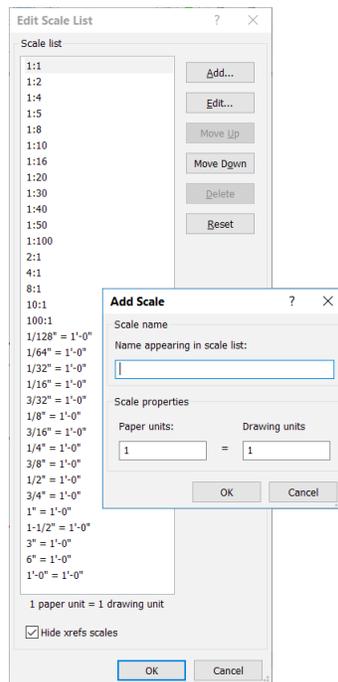
The same works for blocks, dimensions, multilines, and multiline leaders. For entities that don't have styles, such as hatches and attributes, the dialog box with which you create them has the Annotative check box, as shown by the fragment of the Hatch and Gradient dialog box illustrated below.



Linetypes are a bit different, in that they use the **MsLtScale** variable to match the current annotative scale factor. Some other considerations:

Annotative scales apply to entities, not to layers.

You can't just type one in any annotative scale factor; you are limited to the ones provided by BricsCAD. You can, however, edit them with the **ScaleListEdit** command to add and remove scale factors that your office uses or doesn't use.



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.

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

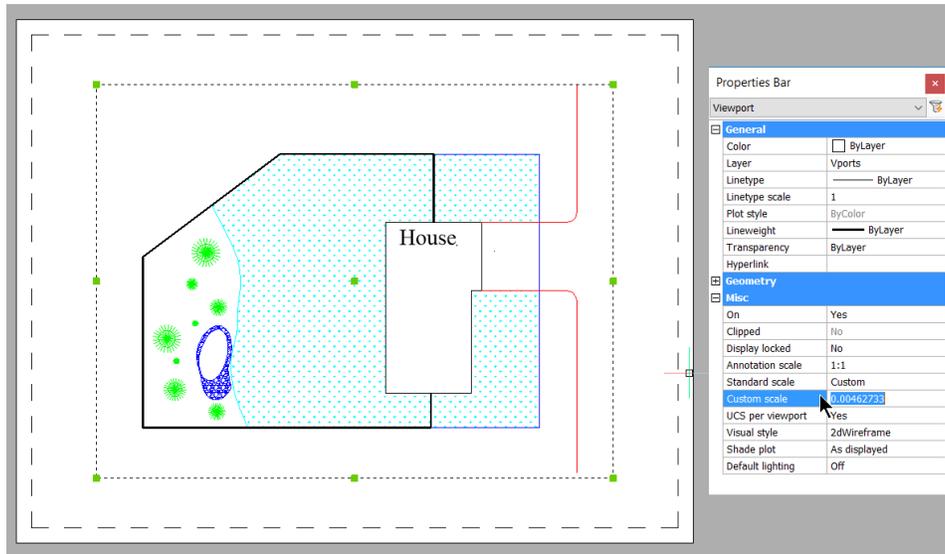
An entity affected by annotative scaling shows the  icon when you pass the cursor over it. When the entity has been assigned more than one annotative scale, then the double version  of the icon is displayed. The icon represents the end view of an engineer's scale ruler, which allows engineers to work with multiple scales.

- 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



- 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

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**.)



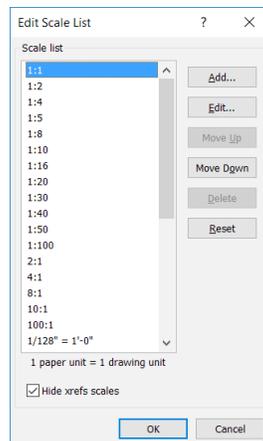
The annotation scale button does double-duty of setting the scale for the viewport and for annotative objects.

5. Enter the model area by double-clicking inside the viewport border (the black rectangle), because annotative scaling works only in model space.
6. Use the **Zoom Extents** command to ensure that the drawing fits the viewport snugly. This gives you the largest scale permissible.
7. 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.)

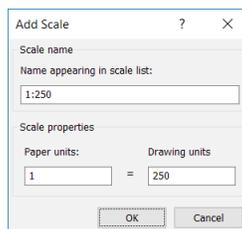
There is, however, no scale factor listed at 1:250, so you need to add it with the ScaleListEdit command, as follows:

- a. Enter the **ScaleListEdit** command. Notice the Edit Scale List dialog box.

: **scalelistedit**

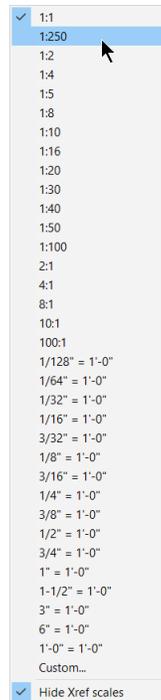


- b. In the Edit Scale List dialog box, click **Add**.
- c. For the 'Name appearing in scale list' you can type anything. For this tutorial, enter **1:250**.



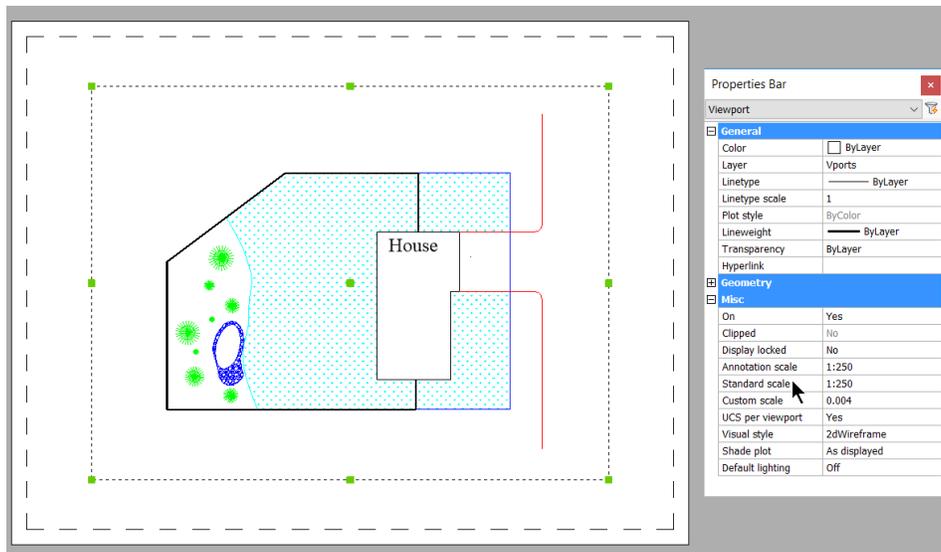
- d. For 'Drawing Units' enter **250**. Leave the Paper Units set to 1.
- e. Click **OK**.

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.

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:
 - a. Click the **Model** tab.

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)

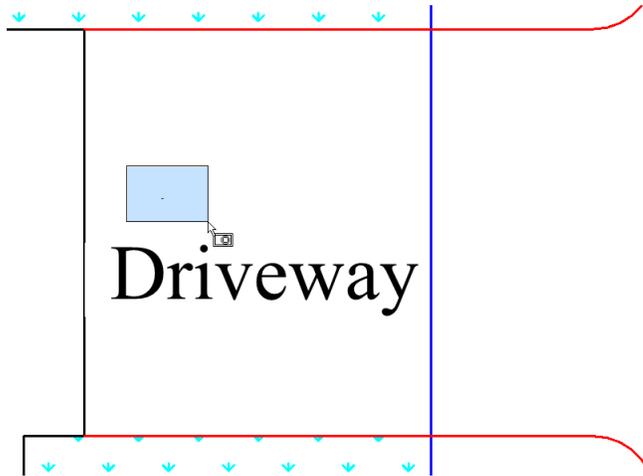
Height of text <0.2000>: **0.125**

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)

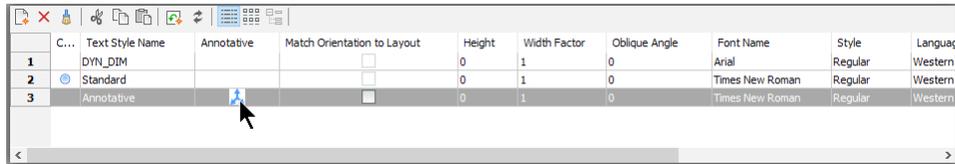
- b. Again place text with a height of 1/8".
- Height of text <0.2000>: **0.125**
- Rotation angle of text <0>: (Press **Enter**)
- Text: **Driveway**
- Text: (Press **Enter**)

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.

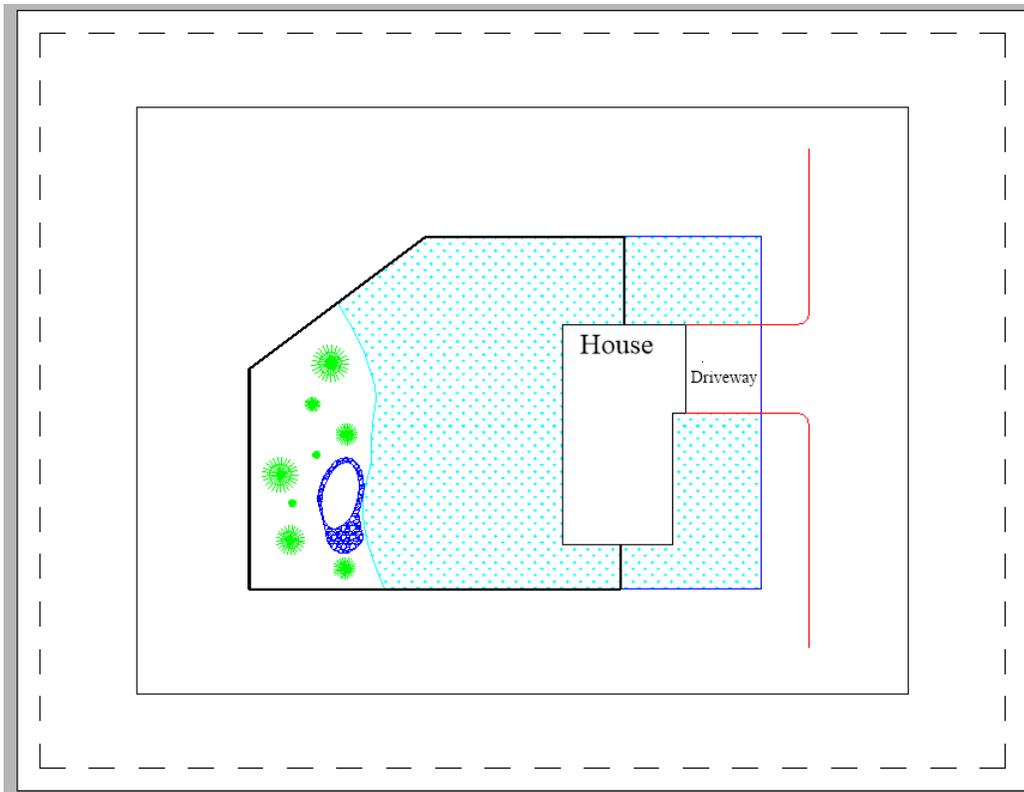
11. Any text you place now in Model tab will be scaled appropriately for the viewport and the scale factor. To see that the text appears at the correct size in the viewport, switch to **Layout1**. The text placed with the Standard style is effectively invisible, but the annotative text "Driveway" is correctly visible.

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!

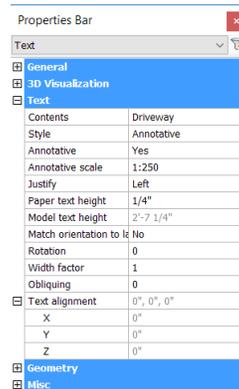


4. Give the style a clever name, such as "Annotative," and then exit the Drawing Explorer.



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")**



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.

Leaders and Multileaders

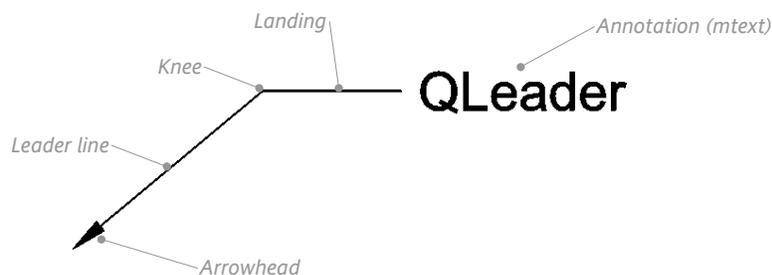
Sometimes you want to point out things in the drawing, and so this is where leaders come in. They point an arrow at the item, and have explanatory text at the other end of a connecting line.

BricsCAD offers commands for two types of leaders:

- ▶ **QLeader** command draw a single-line leader, with settings that allows lots of variations, such as curved lines and different shapes of arrowhead.
- ▶ **MLeader** command draws single- and multi-lined leaders, which are useful for pointing a single line of text at more than one item.

In both cases, the text at the end of the leader line is `mtext`. (There is also a `Leader` command, but it has fewer options than `QLeader`.)

Here are the parts of a typical leader:



In this tutorial, you place both kinds of leader in the drawing.

QUICK SUMMARY OF QLEADER

: qleader
Start of leader <Settings>:
Next point:
To point: <Undo>:
First line of annotation text:
Next line of annotation text:

Start of leader — starts drawing the leader line at the arrowhead end

Settings — displays the Settings dialog box

Next point — draws knee of the leader line, where it bends

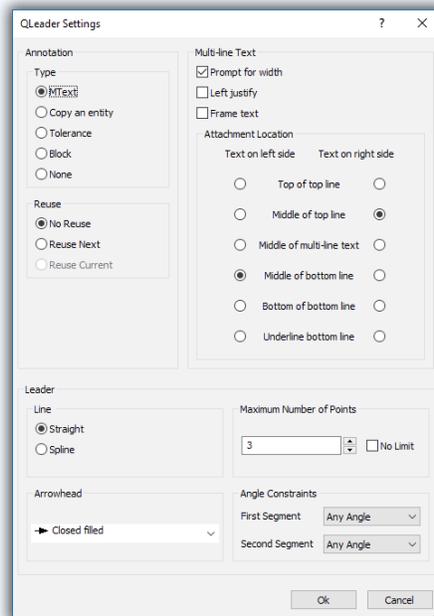
To point — continues drawing the leader line with a knee at each pick point, until you press **Enter**

Undo — reverses the drawing of the last leader line

First line of annotation text — enter text for the leader

Next line of annotation text — enters additional lines of text until you press **Enter**

SETTINGS DIALOG BOX



Annotation — specifies the type of annotation to add: mtext (default), an entity from the drawing, a tolerance specification, a block, or no annotation at all.

Reuse — determines if the annotation used for the previous leader should be reused.

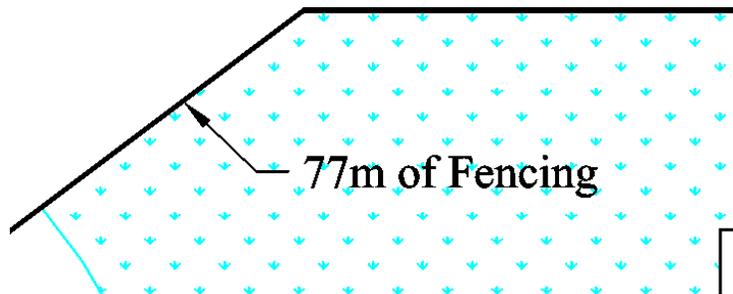
Leader — draws the leader as straight segments or as a spline

SINGLE-LINE LEADERS

Command	QLeader
Ribbon	Draw Leaders Leaders QLeader

In this tutorial, you place a leader in the drawing to point out the fencing and specify its perimeter length, 77m (260 feet).

1. Start the **QLeader** command using one of the methods listed above. (The 'Q' in QLeader is short for "quick," a bit ironic in that this command has many more options than Leader.)
: **qleader**
2. Pick a convenient point along the fencing. This is where BricsCAD will place the arrowhead, which is one end of the leader.
Start of leader <Settings>: *(Pick a point at the fence)*
3. Pick another point. This is the "knee," where the leader line bends to become horizontal. This location is not crucial.
Next point: *(Pick a point where you want the leader line to bend)*
4. Press **Enter** to stop drawing the leader line. (If you keep clicking points, the leader line continues to be drawn with more knees.) This horizontal line is known as the "landing line."
To point: <Undo>: *(Press Enter)*
5. With the leader line drawing business done, now it is time to enter text.
Specify text width <0.00>: *(Press Enter to accept)*
First line of annotation text: **77m of fencing**
6. Press **Enter** after entering '77m of fencing.' BricsCAD prompts you to enter another line of text:
Next line of annotation text: *(Press Enter to end the command)*
You can enter more text, or else just press **Enter** to end the command.



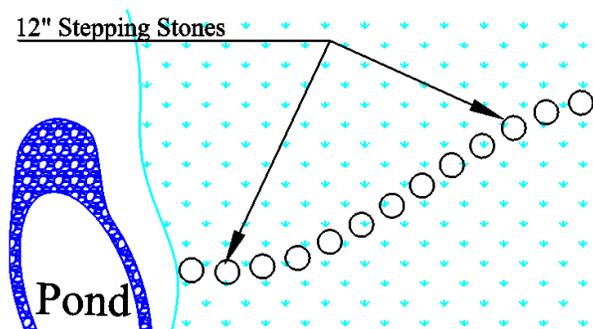
You can add more leaders to point out other parts of the landscape plan. To change the look of the leader, enter the **Format** option.

MULTI-LINE LEADERS

Command	MLeader
Menu Bar	Dimension Multileader
Ribbon	Draw Leaders Leaders Multileader
Toolbar	Dimensions Multileader
Command	MLeaderEdit
Menu Bar	Dimension Edit Multileader
Ribbon	Draw Leaders Leaders Edit Multileader

In this tutorial, you place multiple leaders in the drawing to point out the stepping stones. First, you place one leader, and then you add a second one to the same annotation. This is known as a multi-leader. It is useful for pointing out multiple items in drawings.

1. Start the **MLeader** command:
: **mleader**
2. Pick a point at one of the stepping stones, roughly in the middle of them. BricsCAD will place the arrowhead here, just like the QLeader command.
Specify leader arrowhead location or [leader Landing first/Content first/Options] <Options>: (Pick a point to start the mLeader)
3. Pick a point for the knee:
Specify the leader landing location: (Pick a point)
4. Notice that BricsCAD goes immediately into annotation mode — without a prompt. The MText toolbar appears and you can start entering text. Enter text, such as...
12" Stepping Stones
...and then click **OK** to end the command.
5. To add another leader line, start the **MLeaderEdit** command:
: **mleaderedit**
6. Select the mleader you just drew:
Select a multileader: (Select the mLeader)
7. As you drag the mouse, notice that the second leader line is drawn automatically, following the cursor like a puppy dog. Pick a point near another stepping stone to locate the arrow:
Specify leader arrowhead location or [Remove leaders]: (Pick a point at another stepping stone)



8. Right-click to end the command. (You cannot press Enter to end the command, unlike most other repeating commands.)

Specify leader arrowhead location or [Remove leaders]: *(Press the right mouse button)*

Alternatively, you can use the undocumented **aiMLeaderEditAdd** command to add leader lines. This command differs from MLeaderEdit in that it adds multi-legged leaders.

The related **aiMLeaderEditRemove** command removes leader lines; be aware that the lines are removed *after* you press **Enter** to end the command — and not as you select them. This command also is undocumented.

QUICK SUMMARY OF MLEADER

: mleader

Specify leader arrowhead location or [leader Landing first/Content first/Options]
<Options>:

Specify leader arrowhead location — starts drawing the leader line at the arrowhead end

leader Landing first — starts drawing the leader line in reverse, from the landing

Content first — start the leader by placing first the annotation

Options — continues drawing the leader line with a knee at each pick point, until you press **Enter**

OPTIONS OPTIONS

Enter an option [Leader type/leader lAnding/Content type/Maxpoints/First angle/
Second angle/eXit options] <eXit options>:

Leader type — specifies the leader type, straight, spline or none

Leader landing — toggles the leader landing

Content type — select the annotation types, multiline text, block or none

Maxpoints — specifies the maximum leader line nodes; default = 2

First and second angle — specifies the multiple* angle for the first leader segment; default = 0 degrees

Exit options — exits the options part of this command

* The angle specified here is used as a multiple for the actual angle. For instance, enter 15 to place the line at increments of 15 degrees, such as 15, 30 ,and 45.

ADDITIONAL MLEADER COMMANDS

MLeaderEdit adds and removes leader lines from multileaders.

MLeaderEditExt adds and removes leader lines and vertices from multileaders.

MLeaderStyle creates, edits, and deletes multileader styles (operates in Drawing Explorer).

aiMLeaderEditAdd adds multi-legged leaders.

aiMLeaderEditRemove removes leader lines.

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 when in an earlier lesson you used the Dist and List commands to find distances, lengths, areas, and totals.

In this chapter, you learn how to extract information stored in the drawing, and then how to create a bill of materials (BOM) in BricsCAD drawings and externally in spreadsheets.

IN THIS CHAPTER

- Defining attributes (custom data)
- Attaching attributes to blocks
- Exporting attributes and block data from drawings, then importing data into spreadsheets for analysis
- Importing spreadsheets into drawings as tables
- Automatic bills of materials from 3D component drawings

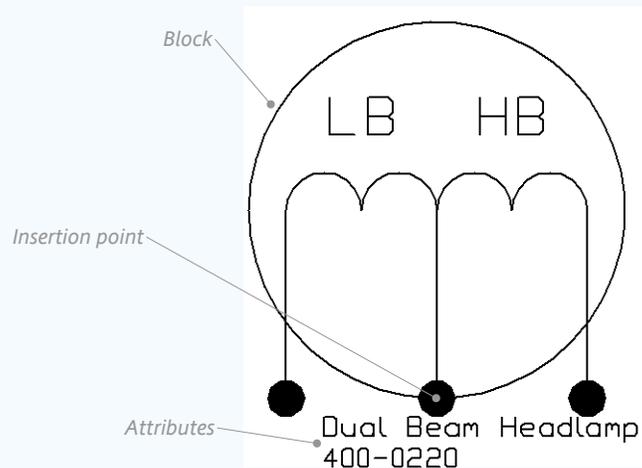
KEY TERMS IN THIS CHAPTER

Attribute — consists of text data attached to blocks

Attribute extraction — describes a process that exports attribute and block data to files

Balloon — leader that identifies parts in BOM tables

Block — describes BricsCAD’s term for a symbol. (Other CAD software packages call blocks “components,” “cells,” “shapes,” “symbols,” or “parts.”)



BOM — bill of materials

Data link — link established between an external data file and a table in a drawing

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

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
bmBalloon	...	Assembly Balloon	Assembly Inquire Balloon
bmBom	...	Assembly Bill of Materials	Assembly Inquire Bill of Materials
bmExplode	...	Assembly Assembly Explode Explode	Assembly Tools Explode
DataLink
DataLinkUpdate
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 from standard parts, such as desks, windows, bolts, and switches. When the parts contain attribute data, then the drawings become truly useful.

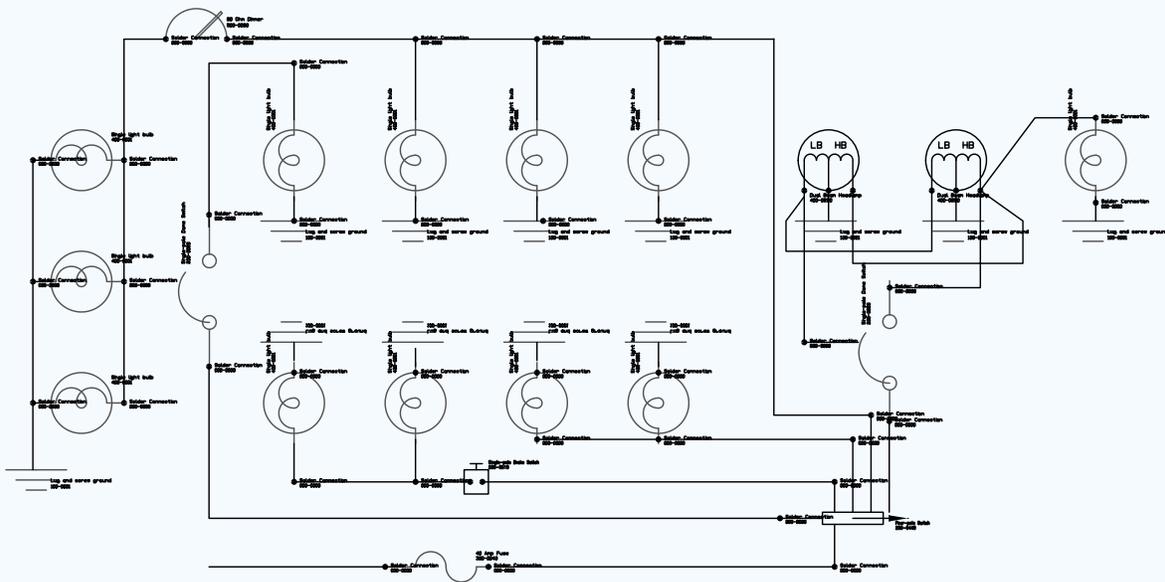
In this chapter, you create the schematic drawing for an automobile electrical system. The drawing contains numerous components, such as a battery, fuse, several kinds of light bulbs, and quite a few ground and solder connections. You could count these components by hand, but I'd wager that you would 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 software 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*.



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.

Download the files for this chapter from

<https://www.dropbox.com/s/ft6gzw3r408ndgd/Inside-BricsCAD-Tutorial-Files.zip?dl=0>.

Creating Blocks with Attributes

The example drawing for this lesson is part of the electrical schematic of an antique automobile. See the previous page. (Wondering which auto? It’s adapted from the electrical system of my first car, a 1965 Volkswagen Beetle.) If you have repair manuals or other schematic drawings laying around, you can use them to produce your own schematic with BricsCAD. If not, then follow along with the schematic sketch I provide with this book.

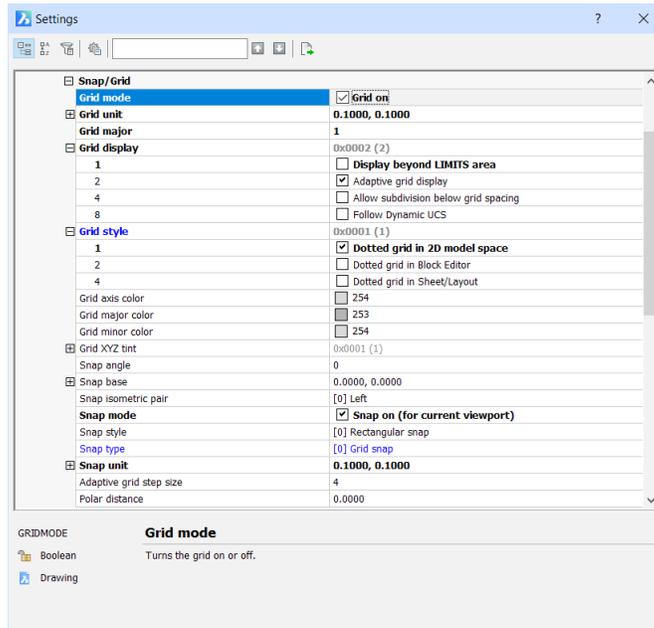
To begin, you create one block and then attach attributes to it. Working with attributes is kind of tedious, which is why I won’t get you create *all* of the blocks. I’ve got them all in a drawing file that you can download.

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, as well as 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
Grid Style	✓ Dotted Grid in 2D model space

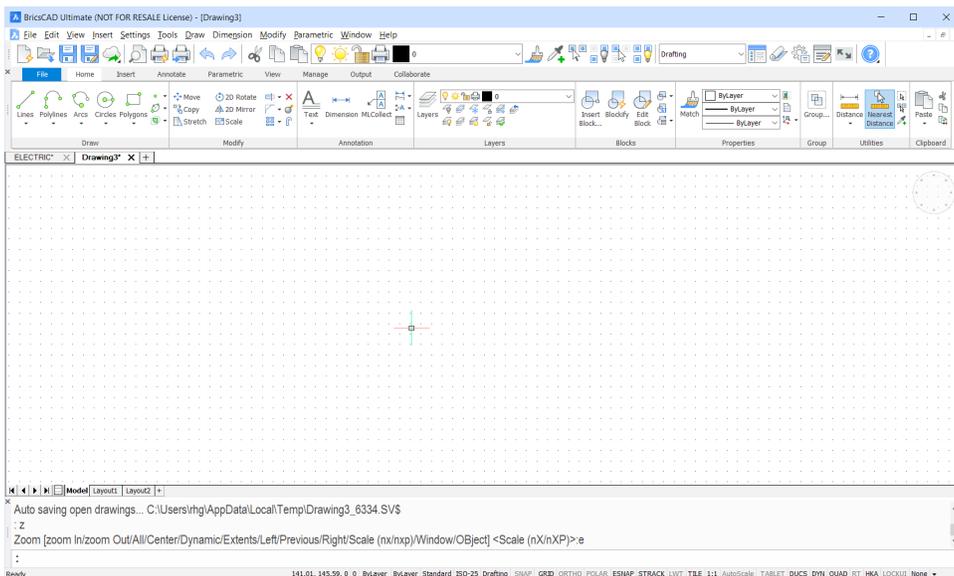
Snap Mode ✓ Snap on
 Snap Unit 0.1,0.1

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 will be ready to draw:
 - a. There is no need for the UCS icon, so turn it off with the **UcsIcon** command:


```
: ucsicon
[ucs icon ON/ucs icon OFF/display in All views/display at ORigin/display in Corner]
<ON>: off
```
 - b. Perform a **Zoom All** to center the drawing on the screen. All you see is the grid, for now.



DRAWING BLOCKS

Command	Donut
Alias	doughnut
Menu Bar	Draw Donut
Ribbon	Draw Draw Polygons Donut
Toolbar	Draw Polygons Donut

The symbol that is used for solder connections is one that's really easy to draw. It is just a fat dot, and so is best drawn with the **Donut** command. This command creates solid looking circles out of wide polyline arcs. To draw one, you specify an outer and an inner diameter.

1. Start the **Donut** command.
: donut
2. The solder connection is 0.1 units in diameter. To make a solid-filled donut, you specify an inside diameter of zero. Enter the following sizes at the prompts:
Inside diameter of donut [2 Point/3 Point/Tangent Tangent Radius] <0.5>: 0
Outside diameter of donut <1>: 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 [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/Object] <Scale (nX/nXP)>: e



(If the donut looks like it has straight sides, then use the **Regen** command to smooth the edges.)

: regen

DEFINING ATTRIBUTES

Command	AttDef
Menu Bar	Tools Attributes Define Attributes
Ribbon	Insert Blocks Define Attributes
Toolbar	Tools Attributes Define Attributes

With one solder connection drawn as a donut, you now create the attribute data for it. The process takes these steps:

Step 1: **AttDef** command defines attribute definitions

Step 2: **Block** command attaches attribute definitions to blocks

Step 3: **Insert** command defines attribute data during block insertion.

Optional step: **AttEdit** command edits attribute data after insertion (not in this tutorial)

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.

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 attributes and properties — from all entities for further processing by data management software.

A single block can contain one or more attributes. Just as blocks are graphical descriptions, attributes are textual descriptions. Think of attributes as custom data, or user-defined properties. And only text or numbers can be used; graphical data, such as images and other entities, cannot be included in attributes.

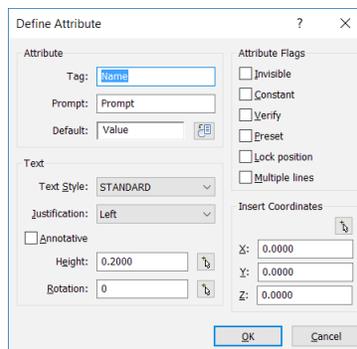
Attributes can be used to describe the block’s part number, its manufacturer, the price, and any other text-based information you want to include. These attributes can have descriptive labels, such as “Product name,” “Manufacturer,” “Model number,” “Stock number,” “Serial number,” and “Material.”

You could include “Price” as an attribute field; but since prices tend to change, it is better to deal them in a spreadsheet program. Instead, I suggest that if you need prices in attributes that you use a price code instead, such as AA, which is substituted later 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**



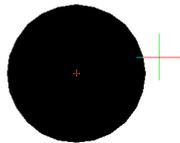
- In the **Attribute** section, enter the following text:

Attribute Section	Value	Meaning
Tag	Product	Identifies the attribute to the program
Prompt	Product	Prompts the user about the data to enter
Default	Solder Connection	Default value of the data; can be overridden by the user

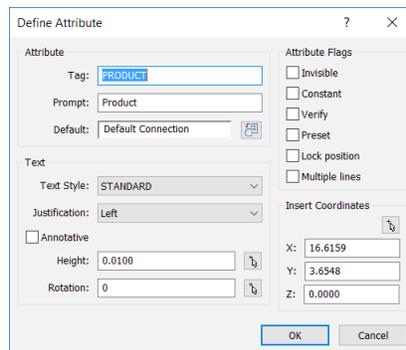
- In the Text section of the dialog box, set the **Height** (of the text) to a tiny, unobtrusive value, such as **0.01**. You don't actually want to see attributes in drawings, and so you work with them only through dialog boxes. (Attributes can be hidden entirely from view, but by being tiny you can tell where they are, but they don't get in the way.)

Text Section	Value	Meaning
Height	.01	Specifies the height of the attribute text

- The **Insert Coordinates** section determines the point in the drawing at which the attribute text starts. A logical location would be on or near the block. To obtain the x,y coordinates, follow these steps:
 - In the Insert Coordinates section, click the  button. The dialog box disappears temporarily.
 - 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)*
 - Pick a point to the right of the connection block:



As soon as you do, the dialog box returns. Its X, Y, and Z fields should now be filled out with values that look similar to the following:



- Click **OK**. Notice that BricsCAD shows the tag next to the donut.



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, you can add it as a second attribute directly below the first, like this:

1. Press the spacebar to repeat the **AttDef** command:

: (Press spacebar)

_attdef

And 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 that 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: click SNAP on the status bar.)
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.” This 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

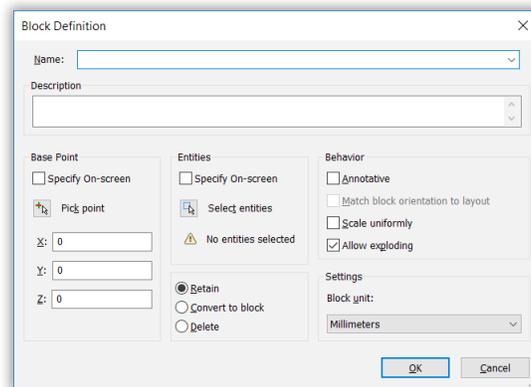
Command	Block
Alias	b
Menu Bar	Tools Create Block
Ribbon	Insert Blocks Create Block
Toolbar	Tools Create Block

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:

Block Section	Value	Meaning
Name	Connect	Identifies the block to you and the program

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.

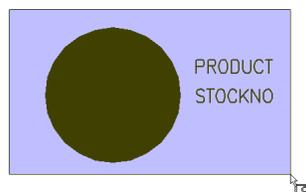
Insertion point for new block: *(Pick the center of the donut)*



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**.



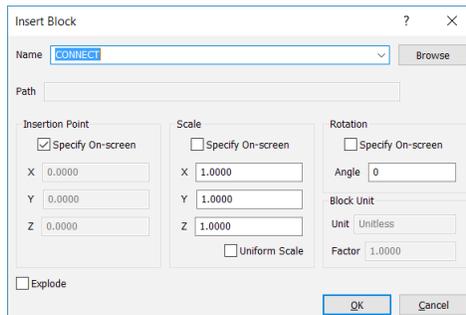
Option	Meaning
Retain	Retains the entities after defining them as a block
Convert to Block	Creates a block from the entities, and then inserts them in the drawing; this option combines two steps: erases the entities and inserts the block
Delete	Erases the entities after defining them as a block; this 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

In an earlier lesson, you placed blocks with the Insert command. To see how attributes work, you will use this command with the Connect block.

1. Start 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 in the dialog box:

Attribute Field	Value	Meaning
Insertion Point	✓ Specify On-screen	Point at which block is inserted in drawing
Scale X	(off; scale = 1)	Size of the block
Rotation	(off; angle = 0)	Angle at which the block is placed

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:

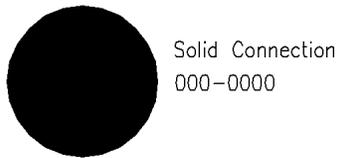
Insertion point for block: *(Pick a point)*

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 default values:

Product <Solder Connection>: *(Press Enter)*

Stock No. <000-0000>: *(Press Enter)*

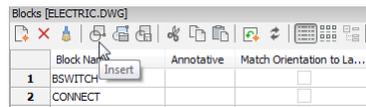


Notice that the attribute values appear next to the connection block in exactly the same location where you placed the tags.

ALTERNATIVES TO THE INSERT COMMAND

In addition to the Insert command, BricsCAD has several other methods for placing blocks in drawings:

- ▶ **Drawing Explorer** (ExpBlocks command) accesses blocks stored in other drawings and inserts them into drawings with the Insert button

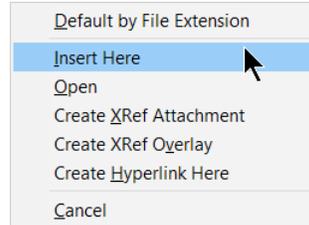


- ▶ **-Insert** command displays insertion options in the command bar, and is meant for use in scripts and macros
- ▶ **Drag'n drop** technique drags *.dwg* files from the operating system's file manager right into the drawing

Inserting Blocks with Drag'n Drop

When you use drag'n to drag .dwg files from the file manager and drag them into the drawing, BricsCAD reacts differently depending on which mouse button you hold down during the action:

- ▶ Hold down the **left** mouse button: the 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 the drawing files according to their file types, with the appropriate commands:

File Extension	Command Activated	Result
Drawing .dwg	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

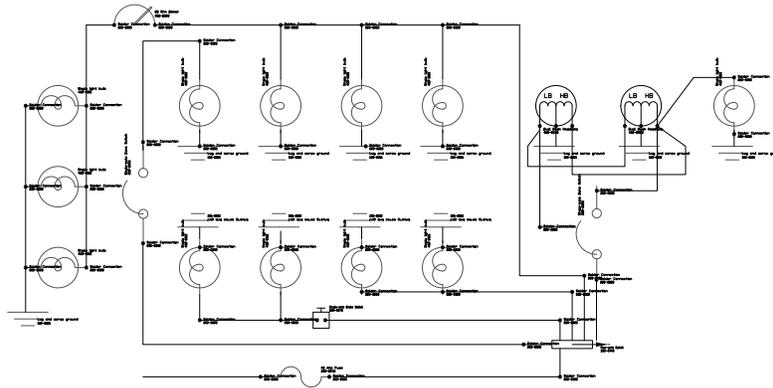
Create Hyperlink Here — prompts you to select one or more entities, and then attaches a URL to them that points to the file through the -HyperLink command

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 Dropbox folder:

<https://www.dropbox.com/s/ft6gzw3r4o8ndgd/Inside-BricsCAD-Tutorial-Files.zip?dl=0>.



DATA EXTRACTION

Command AttDisp
Alias ad
Menu bar Tools | Attributes | Attribute Display
Ribbon Insert | Blocks | Attribute Display
Toolbar Tools | Attributes | Attribute Display

Command DataExtraction
Menu Bar Tools | Data Extraction

To export data from drawings, you use the **DataExtraction** command. It saves the data in a CSV file, which is short for “comma delimited values.” The CSV format separates each piece with a character, such as a comma. Software like spreadsheets and databases recognize the comma as the separator, and place the next piece of data into the next cell or field. This makes CSV a kind of universal exchange format for data.

Once the CSV file is in the spreadsheet or database programs, it can be processed further, such as adding up columns and creating charts. The processed data can then be imported back into BricsCAD. When the data comes from the Excel brand of spreadsheet, it is a linked table. This means that when you make a change in the Excel spreadsheet, the table in the drawing is updated. (Sorry, but this function is not available with LibreCalc and other competitors to Excel.)

(This command supersedes the aging AttExt command, because it does not need template files to be prepared first.)

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**

The **AttDisp** command changes the way BricsCAD displays attribute text:

Normal — displays attributes, except for those set to Invisible mode.

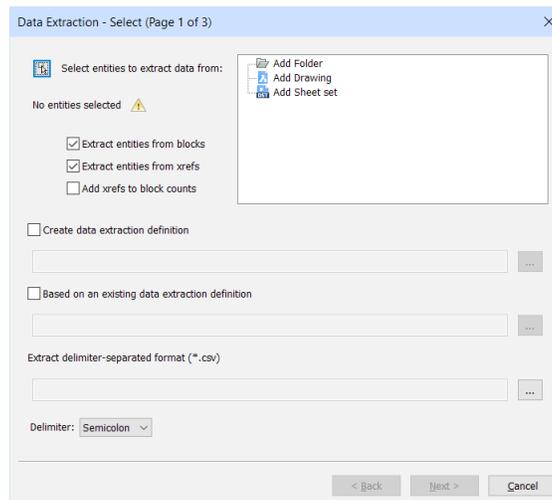
On — displays all attributes, including invisible ones.

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 to 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.

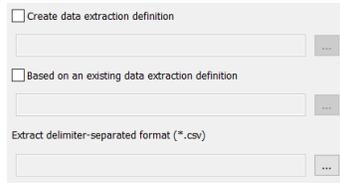
5. Back in the dialog box, notice the first set of options:



You are only interested in extracting attribute *data* from blocks, so turn off all options. The section now looks like this:



- The second set of options specify files that are needed. Let's go through them one by one:



Create Data Extraction Definition — I recommend that you on this option.

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 don’t have to set things up all over again. You just use the “Based on an existing data extraction definition” option. You’ll 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 on a drawing.

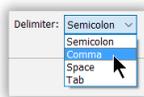
Extract Delimiter-separated Format — Specify the file name and folder location for the extracted data. Data is saved in CSV format, which separates data values with commas: one record per row, with fields separated by commas. You have to fill out this option, as BricsCAD prevents you from moving further without it by keeping the Next button deactivated.

For ease of reference, specify these names for each file:

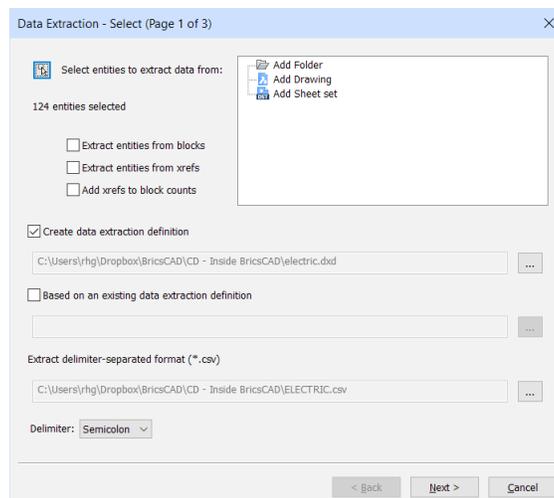
Create Data Extraction Definition	electric.dxd
Based on an Existing Data Extraction Definition	<i>(ignore this time around)</i>
Extract Delimiter-separated Format	electric.csv

For each of the options that you are using, click the **...** **Browse** button, and then choose a convenient folder location. Specify the file names as ‘electric’ with appropriate extensions (.dxd and .csv).

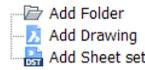
- Click the **Delimited** droplist, and change it to ‘Comma.’ A *delimiter* is punctuation that separates data values in the CSV file. It can be a semi-colon, a comma, a space, or a tab. I recommend using the comma.



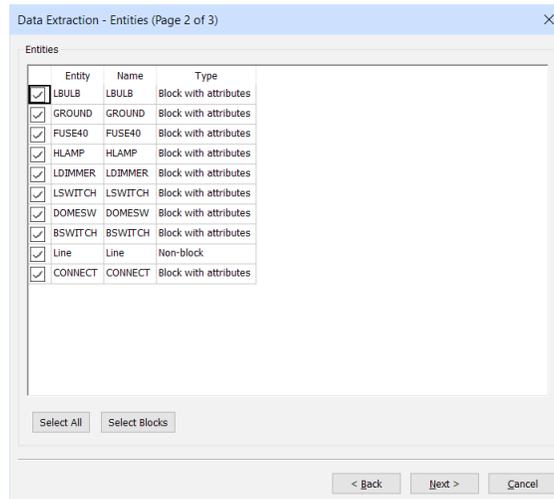
When you are done, the dialog box should look something like this.



(The area in the upper right is useful for making a massive extraction of data by adding entire folders of drawings, individual drawings, and sheet set collections of drawings.)



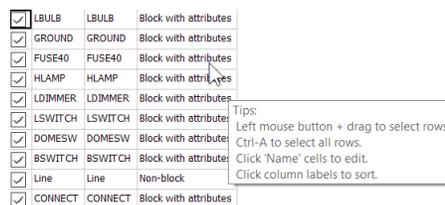
- Click **Next**. Notice that the Page 2 dialog box lists the name of blocks and line entities. You want data about all of them extracted, so leave everything as it is. Even the line entities are useful, because you can get the length of each one; the total length tells you the amount of wiring needed.



Here is the meaning of the options:

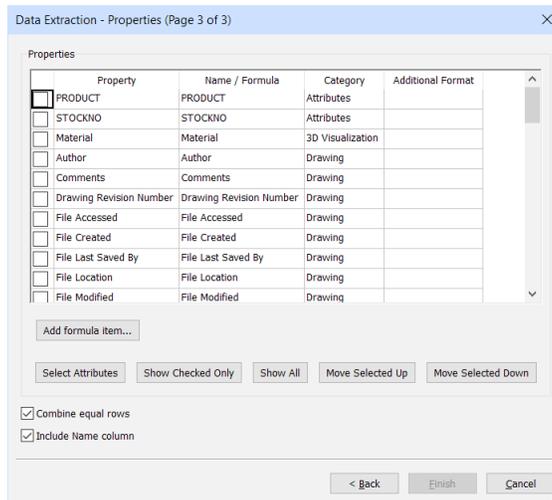
- ▶ **Check boxes** — to exclude an item, click the check box
- ▶ **Select all** — selects all items in the list (turns on all check boxes)
- ▶ **Select blocks** — selects just blocks, and avoids non-block entities (like the lines)

When you pause the cursor over the data fields, a tooltip advises you on additional operations that are possible in this dialog box:



- 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
- ▶ **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



Your big job now is to turn off all properties, except for the following ones:

Length

Product

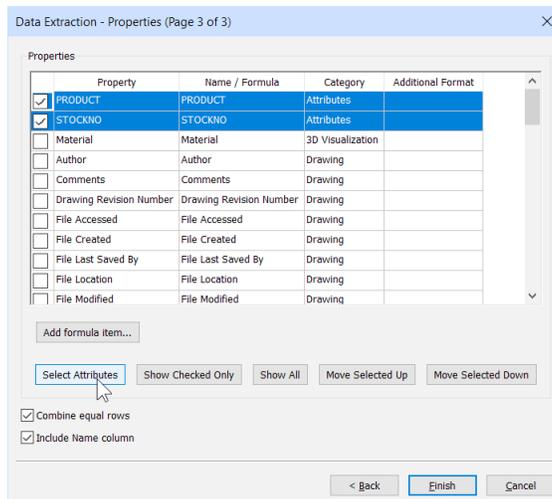
StockNo

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.

To do this, follow these steps:

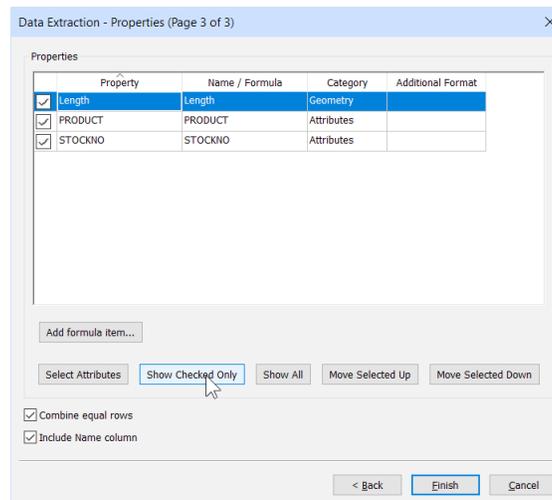
- Click the **Select Attributes** button. This highlights the attributes in the list.



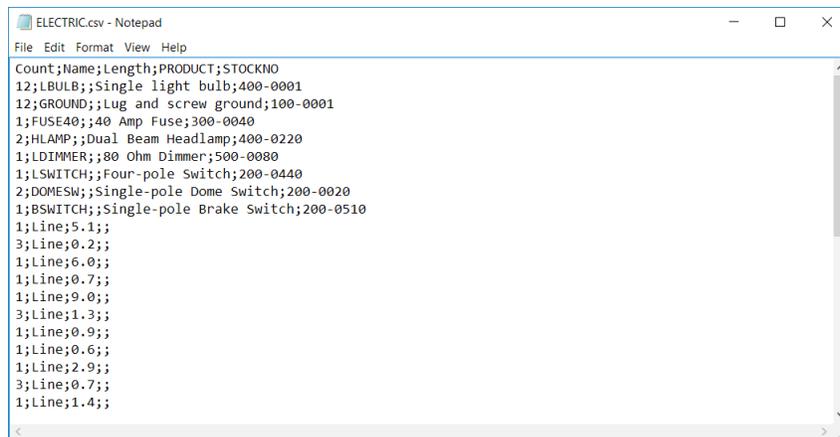
- Click a checkbox in front of an attribute name. This action selects both of them.
- Click the **Property** header. This sorts the items alphabetically by property name.
- Go through the list until you find **Length**, and then click its checkbox.

- e. Click **Show Checked Only** to ensure that you have the correct ones.

Here is the result of your actions:



10. Click **Finish**. After a second or two, the results are deposited in the *electric.csv* file.
11. 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:

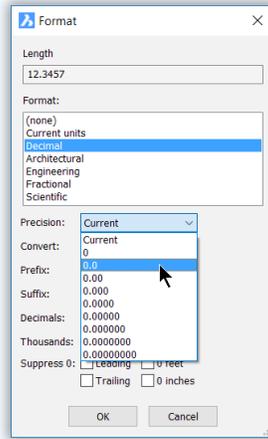
Field	Meaning	Example Value
Count	Number of occurrences in the drawing	12 (light bulbs)
Name	Name of the block or entity	LBULB
Length	Length of line (does not apply to attributes)	5.1
Product	Block's attribute value (does not apply to lines)	Single light bulb
StockNo	Block's attribute value (does not 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.

QUICK SUMMARY OF ADDITIONAL FORMATS

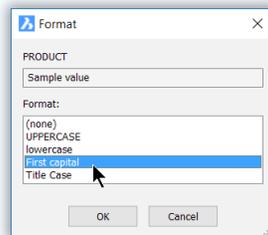
The Data Extraction Step 3 dialog box has a column labeled **Additional Format** column. Its purpose is to specify different formats for values. You don't want too much formatting, because that might confuse the spreadsheet program. Nevertheless, here is how to use it:

1. 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 had 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.

2. The Product and StockNo attributes are text, and I changed the "Format" to **First capital**.



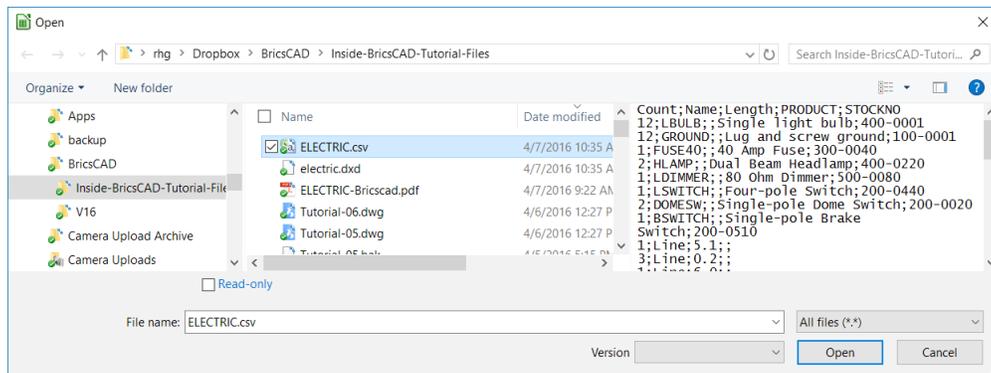
3. Click **OK** to exit the Format dialog box.

The formatting codes used by the DataExtraction command are the same ones used for field text and the Field command.

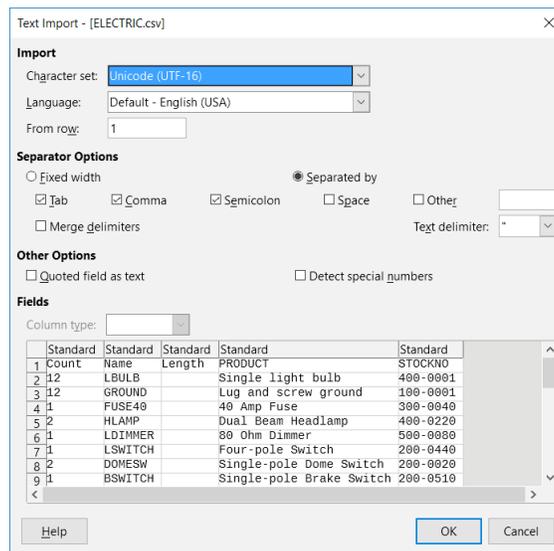
IMPORTING DATA FILES INTO SPREADSHEETS

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:

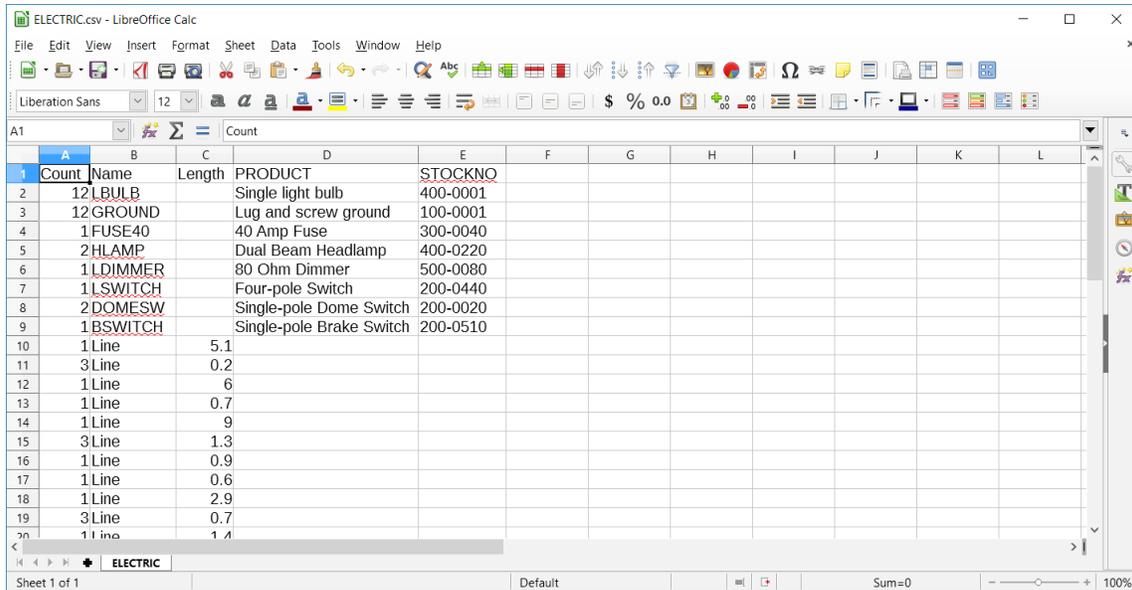
Import

From row: **1**

Separator Options

Separated by: **Tab Comma Semicolon**

- Click **OK**. Notice that LibreCalc 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.

Placing Data in Drawings as Tables

Command	Table
Menu Bar	Draw Table
Ribbon	Draw Tables Table
Ribbon	Draw Table

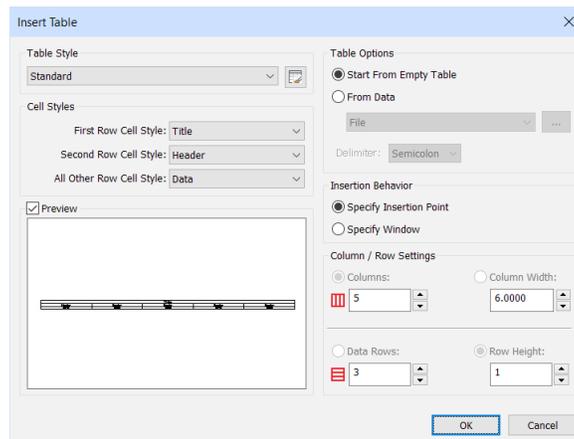
Data that you export as CSV files can be placed in drawings as tables. You can use this to place bills of materials right in the drawing. BricsCAD's Table command imports data in several formats.

- ▶ **CSV** — output from BricsCAD's DataExtraction command, as well as other programs
- ▶ **XLS** — spreadsheet files saved by Excel 2003 and earlier (**NEW IN V19**)
- ▶ **XLSX** — encapsulated spreadsheet files saved by Excel 2007 and newer (**NEW IN V19**)
- ▶ **XML** — non-product specific spreadsheet format using XML codes (extended markup language), and can be exported by LibreOffice Calc and Excel. (Don't confuse XML with XLS, the latter being Excel's native format.)

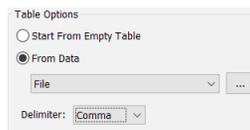
\$

You place these data files in drawings as tables with the Table command. In the following tutorial, you import the *electric.csv* file you created earlier.

1. Start the **Table** command. Notice the Insert Table dialog box.

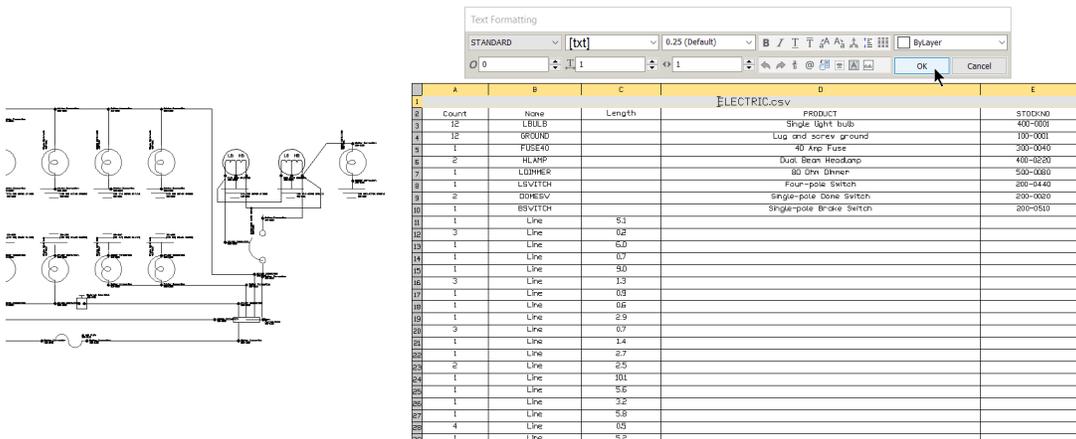


2. In the Table Options section, choose **From Data**.



3. Open the *electric.csv* file you created earlier:
 - a. Click the **...** **Browse** button.
 - b. In the Open dialog box, navigate to the folder in which the *electric.csv* file is stored.
 - c. Choose it, and then click **Open**.
4. If necessary, change the **Separator** field to "Comma" or which ever character is used to separate fields in the file. If you specify the wrong separator character, BricsCAD will later complain "Invalid format."

- Click **OK**. Notice that BricsCAD prompts you to pick an insertion point:
Specify insertion point or [Style/Width/Height]: (Pick a point)



The diagram shows an electrical circuit with various components like bulbs, fuses, switches, and ground connections. To the right, the 'Text Formatting' dialog box is shown with 'STANDARD' style, font size '0.25 (Default)', and 'ByLayer' layer. Below the dialog is a table with columns A, B, C, D, and E. The table contains data for an 'ELECTRIC.csv' file, listing components like bulbs, ground, fuses, switches, and lines with their respective counts and lengths.

	A	B	C	D	E
1				ELECTRIC.csv	
2	Count	Name	Length	PRODUCT	STOCKNO
3	12	LBULB		Single light bulb	400-0001
4	12	GROUND		Lug and screw ground	100-0001
5	1	FUSE40		40 Amp Fuse	300-0040
6	2	NLDRP		Dist. Screw Headlamp	400-0200
7	1	LDHHEK		80 Ohm Diener	500-0000
8	1	LBVETR		Four-pole Switch	200-0440
9	2	DDHESV		Single-pole Dome Switch	200-0000
10	1	SBVTRH		Single-pole Brake Switch	200-0000
11	1	Line	5.1		
12	3	Line	0.2		
13	1	Line	6.0		
14	1	Line	0.7		
15	1	Line	9.0		
16	3	Line	1.3		
17	1	Line	0.9		
18	1	Line	0.6		
19	1	Line	2.9		
20	3	Line	0.7		
21	1	Line	1.4		
22	1	Line	2.7		
23	2	Line	3.5		
24	1	Line	10.1		
25	1	Line	5.6		
26	1	Line	3.2		
27	1	Line	5.9		
28	4	Line	0.5		
29	1	Line	9.2		

- The table is placed in the drawing, and BricsCAD immediately goes into formatting mode, should you wish to further change the look of the table.

The data you inserted in the drawing is static. It will change only if you edit the table.

LINKING DATA IN DRAWINGS WITH TABLES

Command DataLink

(NEW IN V19) Sometimes you want the spreadsheet data in drawings to remain static — it shouldn't change, and so you would use the Table command's From Data option.

Other times, you want the data to update every time the spreadsheet changes. For instance, you might want to make prices dynamic: change the price in the spreadsheet and the price shown in the drawing's table updates.

Or the parts in the drawing change, and so you want the spreadsheet updated.

To integrate this kind of automation, you need to create a link between the spreadsheet and the table. The link is established with the DataLink command, and then when the data needs to be updated, you use the DataLinkUpdate command.

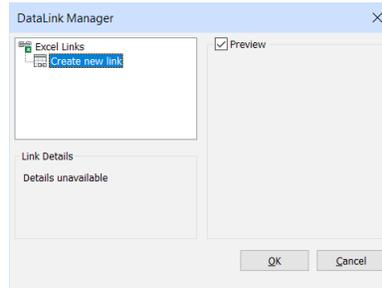
The command links to the following formats of files:

- ▶ **CSV** — output from the BricsCAD DataExtraction command and other programs
- ▶ **XLS** — spreadsheet files saved by Excel 2003 or earlier
- ▶ **XLSX** — encapsulated spreadsheet files saved by Excel 2007 or newer

There is one limitation to be aware of: DataLink works fully only with the full Excel program, and so you cannot use it with Excel Viewer, LibreCalc or any other spreadsheet program, unfortunately.

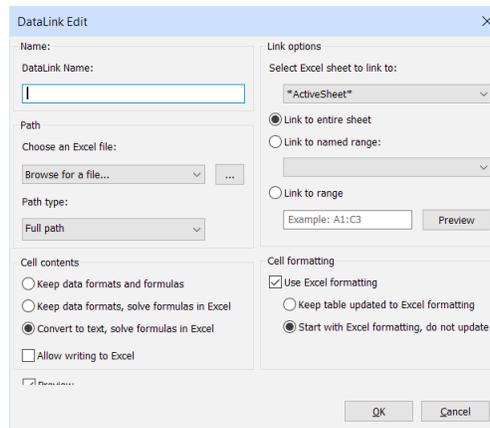
To create a linked table, follow these steps:

0. If necessary, open the *electric.csv* file in Excel, and then save it as an XLS or XLSX file.
1. Open BricsCAD with the *electric.dwg* file.
2. Start the **DataLink** command:
: **dataLink**
3. Notice the Data Link Manager dialog box. It creates new data links with spreadsheets and manages existing ones.



Choose **Create New Link**.

4. Notice the Data Link dialog box:

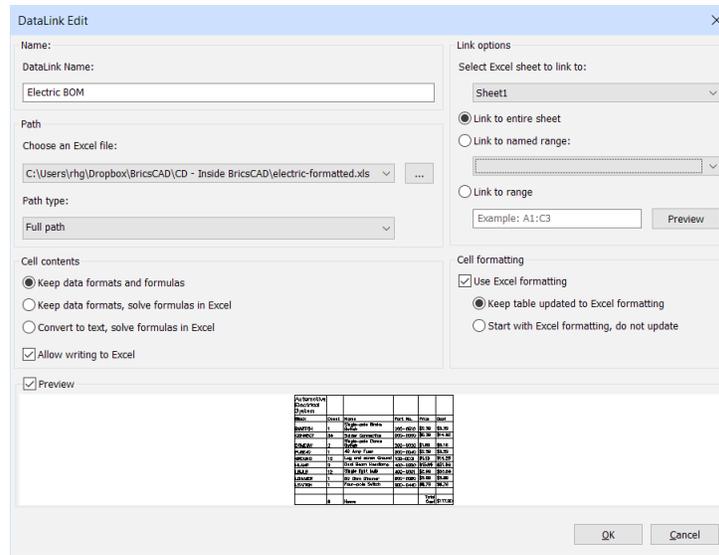


Enter the following information into the dialog box.

Field	Data	Meaning
DataLink Name	Electric BOM	This name identifies the data link; it can be any useful name
Choose an Excel File	electric.xls	Source of data can be a CSV, XLS, or XLSX file
Allow writing to Excel	Yes	Changes made in the BricsCAD table can be written back to Excel
Select Excel sheet	ActiveSheet	This is the sheet that is currently displayed by Excel
Link to entire sheet	Yes	BricsCAD imports only the cell range that is in use
Use Excel formatting	Yes	BricsCAD uses the Excel formatting, instead of a table style

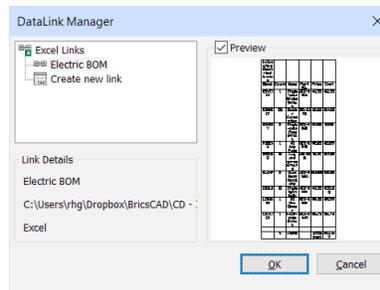
When you choose a CSV file, many of the Excel-related options become unavailable (are grayed out).

- When you are finished, the dialog box should look like this:



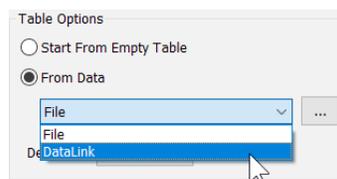
Click **OK**.

- Notice that you are back in the original DataLink Manager dialog box, and that “Electric BOM” is added to the list of Excel Links.

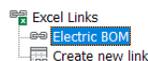


Click **OK**.

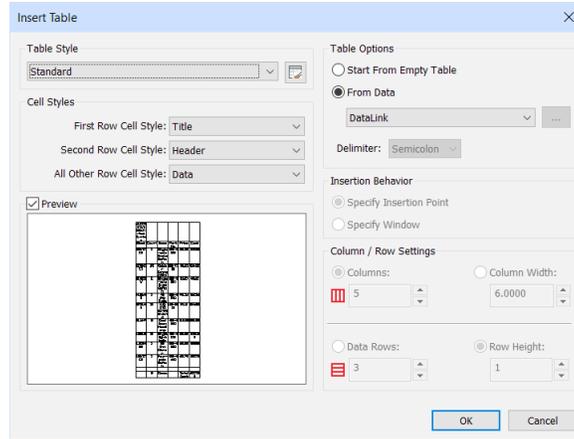
- To place the linked BOM, you use the Table command again, with a small twist:
 - Start the **Table** command:
 - : **table**
 - In the Table Options section, choose **DataLink**:



- The Data Manager dialog box pops up immediately. Choose “Electric BOM” and then click **OK**.



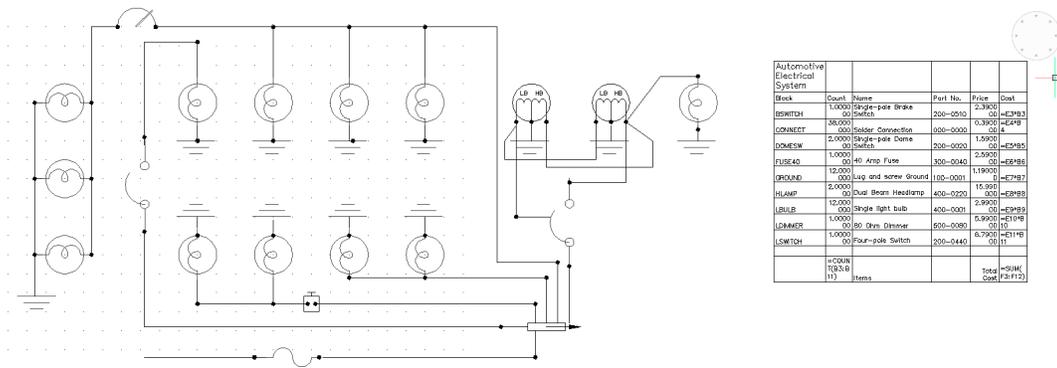
d. BricsCAD brings back in the Table dialog box. It should look like this:



Click **OK**.

- In the command bar, BricsCAD prompts you to place the table:
Specify insertion point or [Style/Width/Height]: (Pick a point)

Pick a point to place the table, which might look something like this:



Warning. The table is *locked*. This means you cannot edit its content, such as text and number. It is locked, because the content is controlled by Excel.

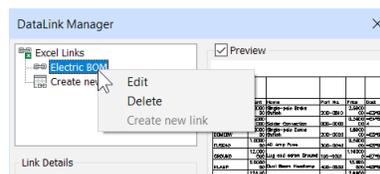
Cell contents are locked

To change the content, do so in Excel, and then use the `DataLinkUpdate` command, as described next.

You still can format the table, such as change the look of text and lines.

Editing a Datalink

To edit a datalink, right-click its name and then choose **Edit**.



The Data Link dialog box appears; make changes and then click **OK**.

Updating the Table

Command DataLinkUpdate

(NEW IN V19) With the table in BricsCAD linked to the spreadsheet in Excel, you can make changes in the spreadsheet, and then force the table to update itself. To reflect the changes, the update is made with the DataLinkUpdate command, like this:

1. To start this tutorial, open Excel with the *electric.xls* spreadsheet, and then make an obvious change. For example, change the number of LBULB (light bulbs) from 12 to 120. This forces the cost to update, as well.)

	A	B	C	D	E	F	G
1	Automotive Electrical System						
2	Block	Count	Name	Part No.	Price	Cost	
3	BSWITCH	1	Single-pole Brake Switch	200-0510	\$2.39	\$2.39	
4	CONNECT	38	Solder Connection	000-0000	\$0.39	\$14.82	
5	DOMESW	2	Single-pole Dome Switch	200-0020	\$1.59	\$3.18	
6	FUSE40	1	40 Amp Fuse	300-0040	\$2.59	\$2.59	
7	GROUND	12	Lug and screw Ground	100-0001	\$1.19	\$14.28	
8	HLAMP	2	Dual Beam Headlamp	400-0220	\$15.99	\$31.98	
9	LBULB	120	Single light bulb	400-0001	\$2.99	\$358.80	
10	LDIMMER	1	80 Ohm Dimmer	500-0080	\$5.99	\$5.99	
11	LSWITCH	1	Four-pole Switch	200-0440	\$6.79	\$6.79	
12							
13		9 Items			Total Cost	\$440.82	
14							
15							

2. Save the spreadsheet, and then exit Excel.
3. In BricsCAD, start the DataLinkUpdate command:
: **data linkupdate**
4. Enter **u** to update the links, and then press Enter.
Select an option [Update data link/Write data link]: **u**
5. You are asked to select the table:
Select objects or [Update all]: (Press Enter to select the table)
6. Wait for a few moments as BricsCAD processes the request. When the command prompt returns, the table shows the updated numbers for LBULB and Cost.

Automatic BOMs from 3D Components

(PLATINUM EDITION ONLY) BricsCAD Platinum can generate BOMs with a single click, when drawings are made with its components. A *component* is a special kind of block, about which BricsCAD knows. In this tutorial, you take a 3D sample drawing, “explode” it, generate the BOM, and then tag parts with balloons.

The commands in this section work only with BricsCAD Platinum and with drawings that have inserts of mechanical components. In the following tutorial, you work through the following:

Stage 1 — Explode an assembly into parts

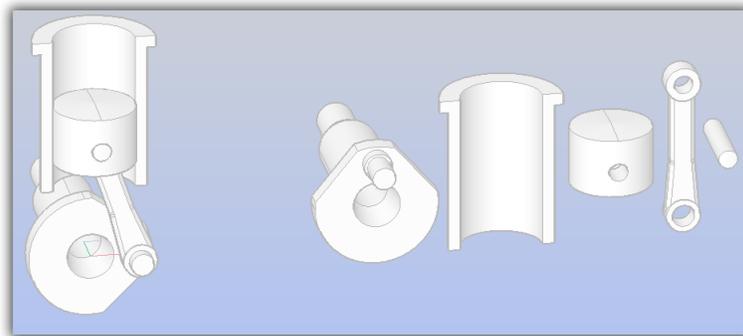
Stage 2 — Generate a BOM automatically

Stage 3 — Label parts with balloons

EXPLODING ASSEMBLIES

Command	<code>bmexplode</code>
Transparent	<code>'bmexplode</code>
Menu bar	Assembly Assembly Explode Explode
Ribbon	Assembly Tools Explode
Toolbar	Assembly Assembly Explode Explode

Complex 3D models are usually made from many parts. Each part is designed separately, as described later in this book, and then assembled to make the entire design — whether a remote control or an entire automobile. A part can be as simple as a pin or as complex as a crankshaft.



Left: An assembly of parts; right: parts exploded

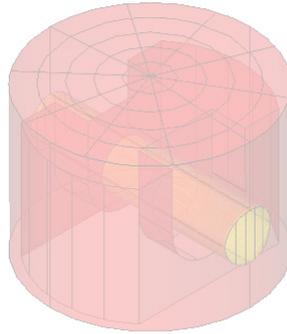
To document the design, it is common to explode the assembly. Exploding separates each part from the other, so that each one can be seen clearly; no parts are hiding any other parts. This kind of exploding has nothing to do with the Explode command used in 2D drafting.

In BricsCAD Platinum, assemblies are exploded with the **bmExplode** command. The “bm” prefix to this command name (and others in this section) indicates “BricsCAD Modeling,” and that these commands are specific to 3D modeling.

In part 1 of this tutorial, you open a 3D assembly model supplied as a sample file with BricsCAD

Platinum, and then separate the parts.

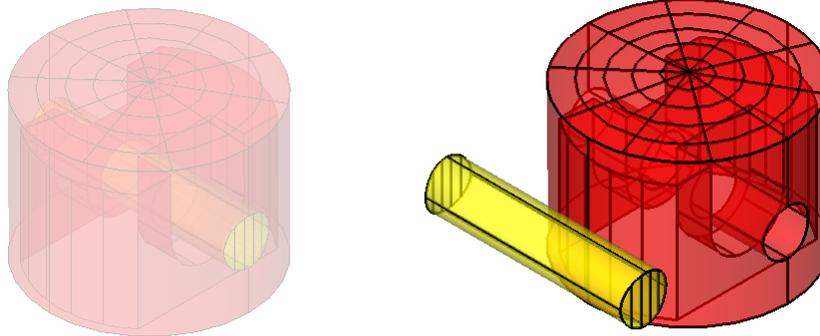
1. Start BricsCAD Platinum, and then open the 3D model *piston-pin.dwg*. This drawing contains two components in an assembly.



2. Enter the **bmExplode** command.

: **bmexplode**

Select explosion algorithm [Table by Level/Table by Types/Manual/Settings] <Manual>: TL



Left: An assembly of two parts; right: parts exploded

QUICK SUMMARY OF bmEXPLODE

: **bmexplode**

Select explosion algorithm [Table by Level/Table by Types/Manual/Settings]

<Manual>:

Select a position of the exploded representation:

Table by Level (TL) — places components according to their level in the assembly tree in the same row

Table by Types (TT) — places components of the same type together in the same row

Manual — places components to be moved apart manually

Settings — determines how assembly is exploded:

- ▶ **Top** — explodes only top level components to suppress minor components
- ▶ **Bottom** — explodes components
- ▶ **Name** — assigns a name to the settings so that they can be used for future explosions

- Notice the BricsCAD turns the exploded parts into a single block. This makes it easier to place many parts at once. Pick a point in the drawing to place the exploded components.
Select a position of the exploded representation: (Pick a point in the drawing away from the source components)

GENERATING BOMs

Command bmbom
Transparent 'bmbom
Menu bar Assembly | Bill of Materials
Ribbon Assembly | Inquire | Bill of Materials
Toolbar Assembly | Bill of Materials

BricsCAD automatically generates a bill of material from any drawing made with components with the bmBom command. Bills of material are listed in table form, like this. The “Quantity” column in the table tells the manufacturer how many parts to manufacture of each item (named in the “Component” column. The “No.” column identifies the part in the drawing, which has a balloon next to it (see the tutorial following).

Bill of Materials lift		
No.	Component	Quantity
1	axis_b	4
2	axis_m	3
3	axis_s	4
4	bar_D	6
5	bar_U	6
6	base	1
7	bracket	1
8	connector	8
9	platform_U	1

QUICK SUMMARY OF bmBOM

: bmbom

Insertion point [Name/Top level/Bottom level/Configure]:

Insertion point — places the table in the drawing by its upper-left corner

Name — adds a name to the title, such as “Lift Parts.”

Top level — makes the BOM table from the top level components only; this is useful for suppressing many minor components.

Bottom level — makes the BOM table from all components.

Configure — specifies which columns are included in the table: Description, Density, Volume, Mass, Material, Thickness (only for sheet metal parts), and Parameters.

The `bmBom` command does nearly all the work for you. All you have to do is specify the BOM table's location in the drawing. The command creates a table that is a regular BricsCAD table that you can edit anyway you see fit. It is subject to the properties of the `TableStyle` command.

For the `bmBom` command to work, it needs at least one component in a drawing. The components do not need to be assembled, nor does the assembly need to be exploded.

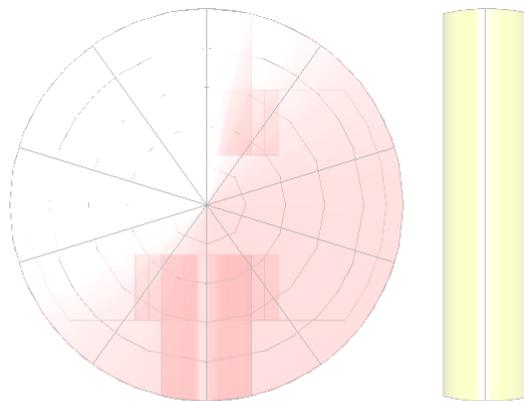
Here is how you generate a bill of materials from the parts:

1. Continue with the model from the previous tutorial.

2. Enter the `bmBom` command:

: `bmBom`

Insertion point [Name/Top level/Bottom level/Configure]: *(Pick a point in the drawing to place the table)*



Bill of Materials Component_4		
No.	Component	Quantity
1	pin	1
2	piston	1

That's it, you're done! BricsCAD does all the work creating the table for you. If you want to change the look, then use the `TableStyle` command.

When you want to change the content of the table, then use the `bmBom` command's `Configure` option, which lets you choose among these columns:

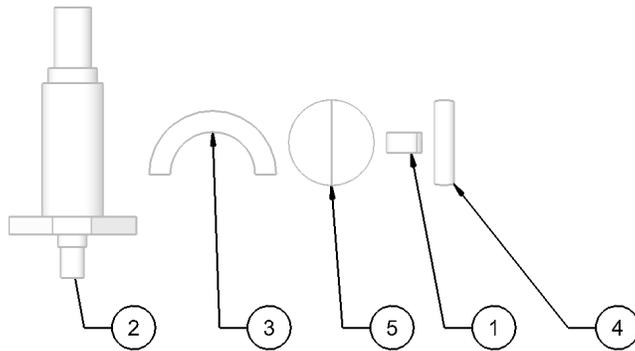
- ▶ Description
- ▶ Density
- ▶ Volume
- ▶ Mass
- ▶ Material
- ▶ Thickness
- ▶ Parameters

The command gets this information from the parts themselves, because they are 3D solid models.

ATTACHING BALLOONS

Command	bmballoon
Transparent	'bmballoon
Menu bar	Assembly Balloon
Ribbon	Assembly Inquire Balloon
Toolbar	Assembly Balloon

BricsCAD semi-automatically attaches balloons to components using the **bmBalloon** command. We use balloons to identify components, making a visual link between the geometry and the text in the BOM table.



During the **bmBalloon** command, you pick the component and then specify the location of the balloon. The command identifies the component, and then assigns the ID number, which it gets from the BOM table. *Balloons can be attached only after the BOM is generated.* This command can be used only in model space.

BricsCAD constructs balloons from multileaders, and so they are subject to the properties of the **MLeaderStyle** command. You can edit the balloons like multileaders, such as pointing multiple leaders to multiple, identical components.

QUICK SUMMARY OF **bmBALLOON**

: bmballoon

Select a component insert [select other Table/choose balloon Frame]:

Pick point to place balloon:

Insert — places the arrowhead end of the balloon, preferably on the component

select other Table — chooses the BOM table to use, should more than one exist in the drawing

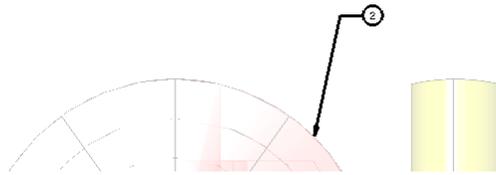
choose balloon Frame — choose the balloon style: circle, rectangle, triangle, or hexagon.

Pick point — places the balloon end; hold down the Shift key to align balloons

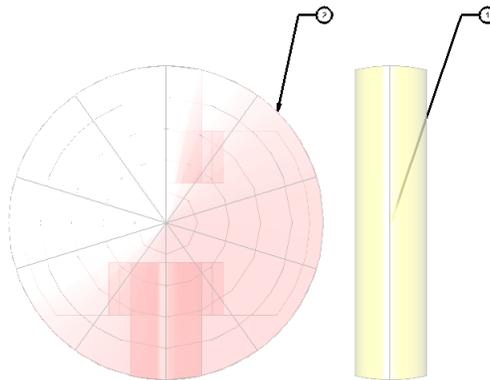
Traditionally, balloons are lined up to make it easier to read and find them, and so BricsCAD accommodates this: hold down the **Shift** key while placing them. This lines up the balloons, vertically or horizontally.

Here is how to attach reference balloons to the parts:

1. Start the `bmBalloon` command:
: `bmBalloon`
2. Pick one of the components. Notice that the arrowhead end of the balloon attaches itself to the component.
Select a component insert [select other Table/choose balloon Frame]: (*Pick a component*)
3. Place the rest of the balloons at a convenient location in the drawing, away from the other components:
Pick a point to place balloon: (*Pick a point*)



4. Repeat the `bmBalloon` command to place a balloon for the other component. At the 'Pick point to place balloon' prompt, hold down the **Shift** key so that the second balloon lines up with the first one.



Bill of Materials Component_4		
No.	Component	Quantity
1	pin	1
2	piston	1

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. At the end of the chapter, you learned how to explode a 3D assembly, have BricsCAD generate a BOM automatically from components, and then place identifying balloons. You learn more about 3D in later chapters.

Next, you learn about a more sophisticated 2D entity known as the region.

Working with 2D Regions & Booleans

Here you learn how to increase your CAD efficiency by working with regions and applying Boolean operations. These are two BricsCAD features that are often used to construct complex 2D and 3D shapes, and then to analyze them for information useful to designers, such as center of gravity. BricsCAD does this far faster than if you were to perform the calculations by hand.

In this chapter you learn to use the similar Boundary and Region commands, and commands related to boolean operations. (The Region command is available only in the Pro and Platinum editions of BricsCAD.)

IN THIS CHAPTER

- Converting entities into regions
- 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 its area, centroid, and radius of gyration

Point filter — returns a single coordinate

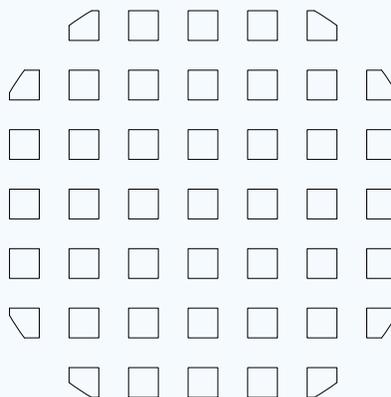
Region — consists of a closed 2D area

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:

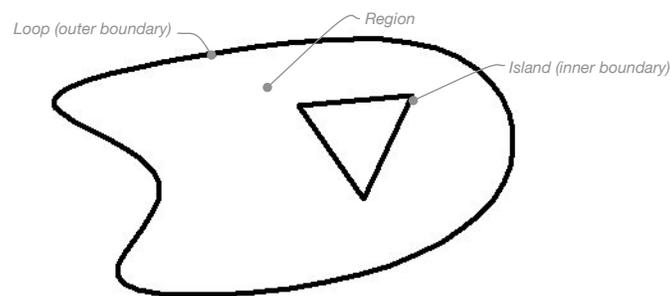


RegionTutorial drawing

About Regions

(**PRO AND PLATINUM EDITIONS ONLY**) 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. Boolean operations join, intersect, and subtract one region from the other. This lets you create a single, complex entity that has holes of any shape in it, and then lets you analyze its physical properties.

Technically, regions are ACIS entities, the same BricsCAD technology uses for its 3D solid models. Regions have their own jargon. The outer boundary of a region goes by the name of the “loop.” The holes inside of regions are called “islands.” The loops and islands can be of any shape, from a simple triangle to flowing curves.



HOW TO CREATE REGIONS

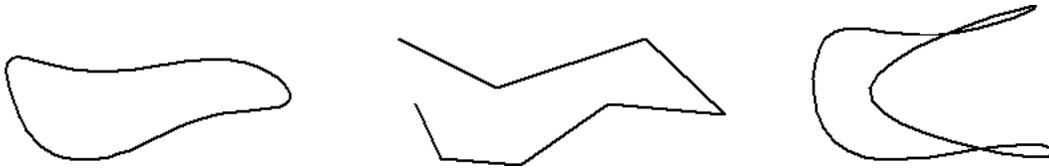
In a curious twist of fate, BricsCAD cannot create regions from scratch. There is no region command that can draw it, as we have with the PLine command for drawing polylines. Rather, the Region command converts *existing* closed entities into regions. To do this, BricsCAD provides you with two commands, **Boundary** and **Region**. They are subtly different:

- ▶ **Boundary** command displays a dialog box, then prompts you to pick a point inside a closed area. It finds the boundary, and makes a region or a polyline along it. It is, in fact, a subset of the Hatch command, and so you get the island options and gap tolerance as a bonus.
- ▶ **Region** command prompts you to select entities that make up a closed area, and then generates a region entity; it operates at the command line.

Either way, 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 when used with the Region command; the Boundary command allows gaps of up to 5000 drawing units in size. 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 results of the commands differs for the shapes shown in the figure 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 is smaller than the tolerance.

The two commands aren't able to combine non-overlapping entities into one. 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.)

Differences Between Boundary and Region

The Boundary and Region commands work differently from each other, and so one might be more useful to you in certain situations than the other. Because of the differences in how the two commands select detect boundaries, the shape of the resulting region can differ greatly.

Boundary. First of all, the Boundary command's dialog box does not let you switch between creating polylines and boundaries. The droplist is grayed out and does not work, for some reason.



Initially, the command is set up to create polylines; to change it to regions, you need to change the HpBound system variable to off, either at the command line or in the Settings dialog box:

```
: hpbound
New current value for HPBOUND [1 for ON/0 for OFF] <1 for ON>: off
```

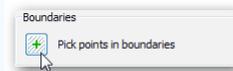
HpBound	Meaning
0	Boundary and Hatch commands make boundaries out of regions
1	Boundaries are made of polylines

QUICK SUMMARY OF BOUNDARY OPTIONS

The Boundary dialog box contains options that control how boundaries are created.

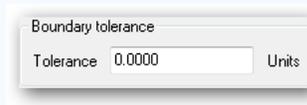
BOUNDARIES

The Pick Points button dismisses the dialog box temporarily, and the prompts you to pick a point in the drawing or select entities. You can pick inside more than one boundary.



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.



BOUNDARY SET

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 does not work in this dialog box. Use the **HpBound** variable to set the boundary as 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.



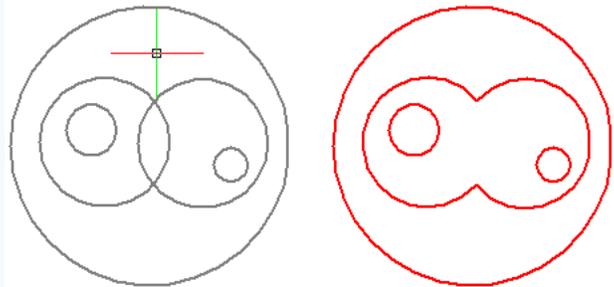
The circles shown below illustrate the difference between the Nested, Outer, and Ignore options. Notice the location of the cursor, and that the pick point is the same in each case.

Continued...

...continued.

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.

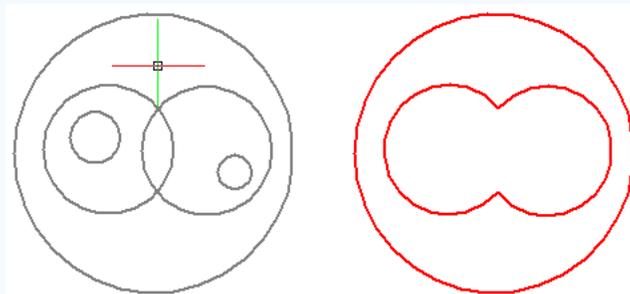
Nested — all entities (the circles) form boundaries, as shown in red at the right.



Left: Original circles

Right: All circles turned into boundaries

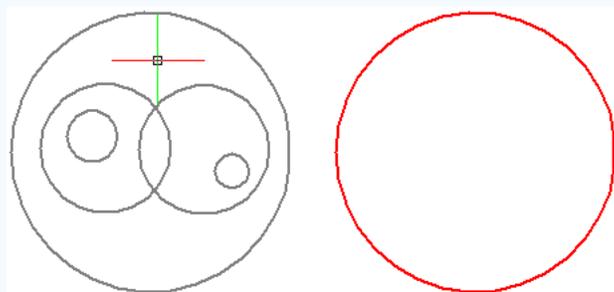
Outer — the outermost circles form boundaries (shown in red below); the innermost circles are ignored.



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.



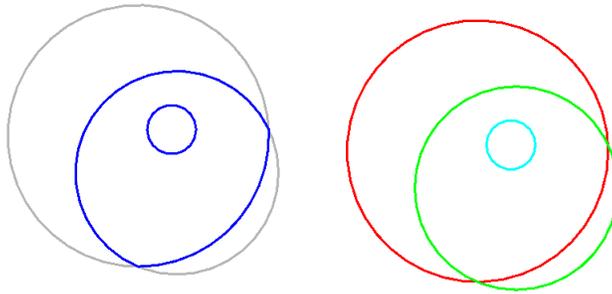
Left: Original circles

Right: Only outermost circle turned into a boundary; other circles ignored

I'd draw three circles to show you how the resulting regions differ according to how you select them. When you click the **Pick Points in Boundaries** button, you are prompted at the command line:

Pick a point to define a boundary or hatch area or [Select entities/Undo]: *(Enter an option)*

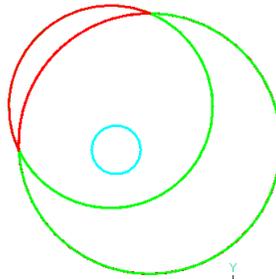
Option	Meaning
Pick a point	Uses boundary detection to find regions around the pick point
Select entities	Picks individual entities, and then forms regions out of them
Undo	Unselects the last pick area or entity



Left: Blue region entities created by picking a point
Right: Red, green, and cyan region entities created by selecting each circle entity

Region. The Region command creates only regions. At the command prompt, it asks you to pick entities:

Select entities or boundaries: *(Select one or more entities)*



In the figure above, the command created the two red and green crescent moon-like regions, plus the circular cyan-colored region.

Summary. I've put together this table to highlight the differences between the two commands:

Activity	Boundary	Region
Entities created (Hpbound)	Polylines Regions	Regions only
Islands detected	Yes, multi-level	Yes, only one level
Allow gaps (HpGapTol)	Yes, up to 5000 units	No
Selection method	Pick enclosed areas Select entities	Select entities only
Retain source entities	Optional	Optional

In summary, the Boundary command is more flexible with more options, but the Region command is simpler with no options.

What To Do with Leftovers

Both commands leave the source entities in place. Change the **DelObj** (short for “delete objects”) variable to change this behavior. Set it to -2 or 2 to have Region erase the source entities automatically:

DelObj	Meaning
-2	Prompts if source entities should be erased with Region
-1	Prompts if source entities should be erased during commands like Extrude, Revolve, and Sweep
0	Leaves source entities in place
1 (default)	Erases source entities only during commands like Extrude, Revolve, and Sweep
2	Erases source entities with Region

The **Explode** 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
Alias	bo
Menu Bar	Draw Boundaries Boundary Polyline
Ribbon	Draw Draw Polygons Boundary
Toolbar	Draw Boundaries

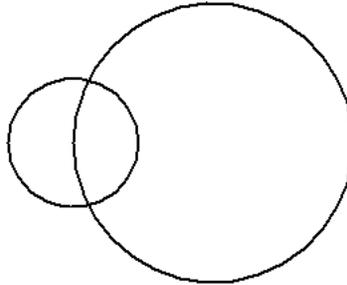
The Boundary command draws a boundary around the inside of selected areas and/or entities. It draws them as polylines or as regions — your choice, although in this tutorial you want regions.

(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 outline, but keeps it.)

The Boundary command displays a dialog box that looks like a simplified version of the Hatch and Gradient dialog box, but the hatch 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.

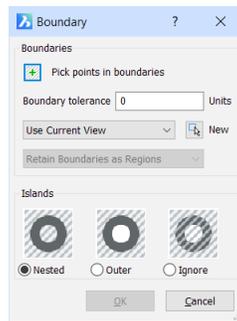


3. Turn off the HpBound system variable to ensure the resulting entities will be regions:

: hpbound

New current value for HPBOUND [1 for ON/0 for OFF] <1 for ON>: **off**

4. Start the **Boundary** command. Notice the Boundary dialog box; the boxed text on a nearby page explains its functions.

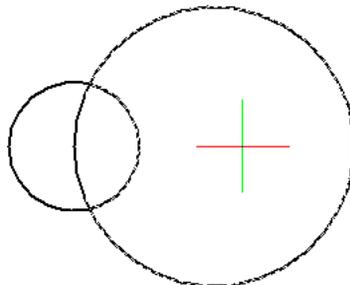


5. Click the **Pick Points in Boundaries**  button. It lets you pick a point inside the circles or the circles themselves. Based on your pick point, BricsCAD determines the boundary(ies) automatically.

6. 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)*

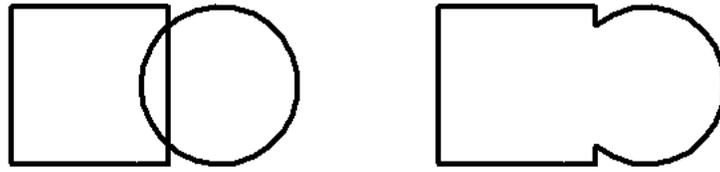
Pick a point *inside* one of the circles, as illustrated below. It doesn't matter which one, just don't select a circle itself.



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.

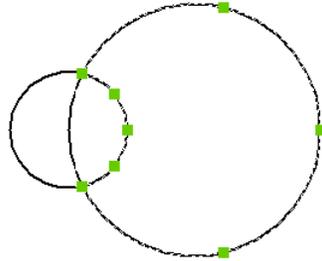
There are two outcomes possible from subtraction. The outcome depends on the order in which you select the regions, as shown by the figure below. In the center result, the circle was removed from the square; in the right result, the opposite occurred: the square was removed from the circle. (Selection order does not matter for the union and intersect operations.)



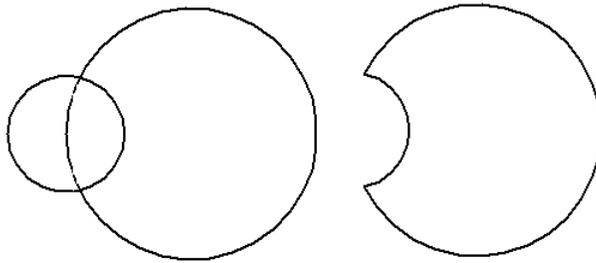
Left: Original objects (two regions); center: circle subtracted from square; right: square subtracted from circle

In boolean terms, the subtraction operation returns everything that's in region #1 but NOT in region #2.

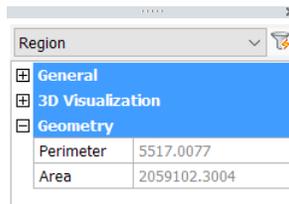
7. Press **Enter** to return to the dialog box:
8. And then click **OK** to exit the dialog box.
9. The circles look no different, yet BricsCAD has drawn an entity inside one of them. To see it, drag it away from the circles, like this:
 - a. Click the circle that surrounds your pick point made during the Boundary command. Notice that when BricsCAD highlights it, the highlight looks like a crescent moon. The 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 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.

After the boundary (region or polyline) is created, the Boundary command does not 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 named after Mr Boole, whose mathematics describe 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 names of the commands, and what they do to regions:

- ▶ **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 unavailable in BricsCAD Classic.

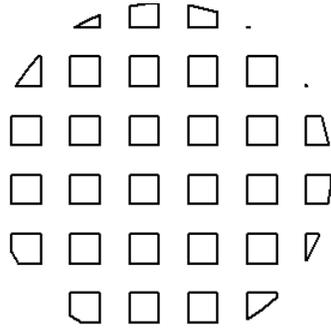
Together with Boundary and Regions commands, boolean commands are employed to construct complex shapes with BricsCAD. Indeed, boolean operations work only with region entities and 3D solids. They do not work with regular 2D and 3D entities like lines and 3D polymeshes; you need to first convert regular entities into regions or 3D solids.

The trickiest of the boolean operations is subtraction, which is why you get to employ it in the following tutorial.

TUTORIAL: CREATING A WAFFLE SHAPE

Commands	ArrayClassic and -Array
Aliases	ar, -ar
Menu Bar	Modify Array Array Dialog Box
Command	Region
Alias	reg
Menu Bar	Draw Region
Ribbon	Surface Create Region
Toolbar	Draw Region
Command	Subtract
Alias	su
Menu Bar	Model 3D Solids Editing Subtract
Ribbon	Solid Edit Boolean Subtract
Toolbar	3D Solid Editing Subtract

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.



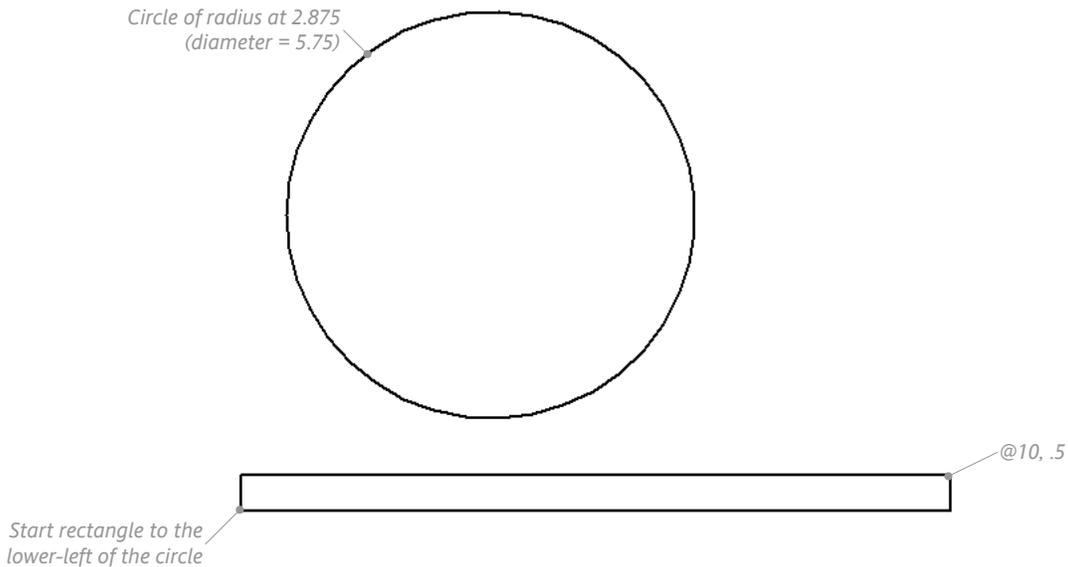
1. Start BricsCAD with a new drawing.
2. Using the **Circle** command, draw a circle with a radius of **2.875** units at the origin (0,0), as follows:

: circle

Select center of circle or [2Point/3Point/TTR/Arc/Multiple]: **0,0**

Set Radius or [Diameter]: **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.

Select first corner of rectangle or [Chamfer/Fillet/Rotated/Square/Elevation/Thickness/Width of line/Area/Dimensions]: **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**

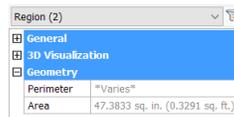
Select entities or boundaries: **all**

Select entities or boundaries: (Press **ENTER** to end entity selection)

2 regions(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] <Rectangular>: (Press **Enter** to accept 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!

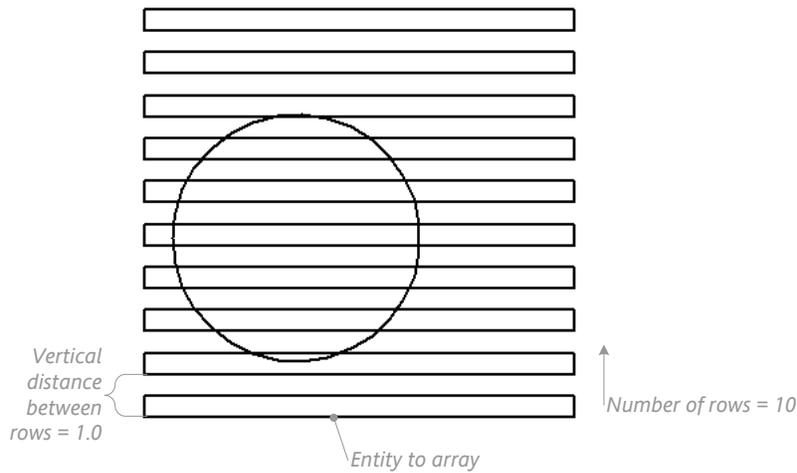
Number of rows in the array <1>: **10**

Number of columns <1>: (Press **Enter** to accept default, 1)

The distance between rows is 1 unit:

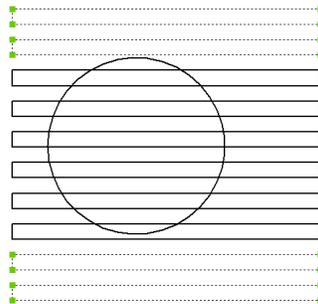
Vertical distance between rows, or spacing rectangle: **1**

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)

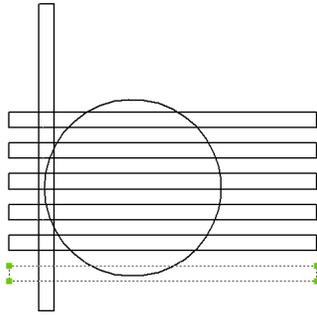
Select entities to mirror: (Press **ENTER** to end entity selection)

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:

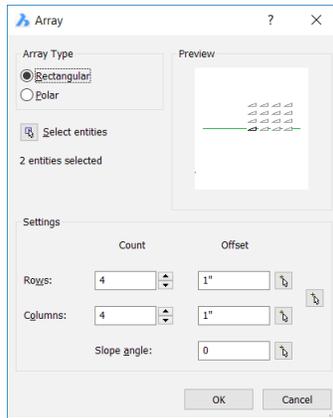
Start of mirror line: `0,0`

End of mirror line: `@1<45`

Delete the original entities? [Yes-delete entities/No-keep entities] <No-Keep entities>:
(Press **Enter** to accept default, *N*)



9. With the vertical rectangle in place, start the **ArrayClassic** 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

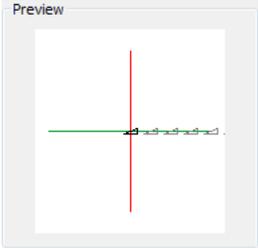
Select entities to array: (Press **Enter** to return to the dialog box)

- c. Specify the number of copies to make:

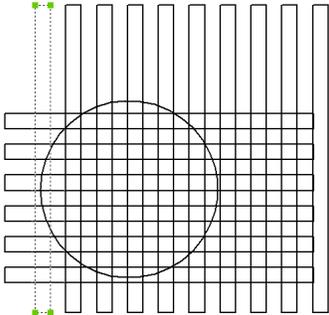


Rows Count: **1**
 Columns Count: **10**
 Column Offset: **1**

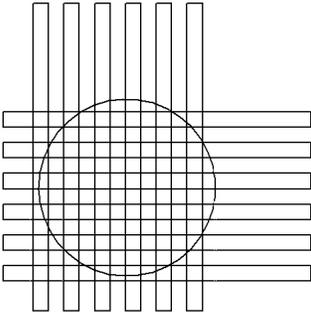
Notice that the preview window shows several columns, one row high.



d. Click **OK**, and BricsCAD instantly creates ten copies of the vertical rectangle.



10. Erase the rectangles that don't lie on the circle.

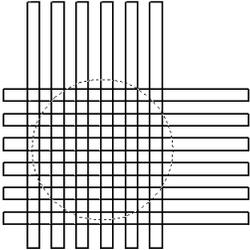


11. Now you finally get to use a Boolean. With the **Subtract** command, remove the rectangular regions from the circular region, as follows:

: **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.

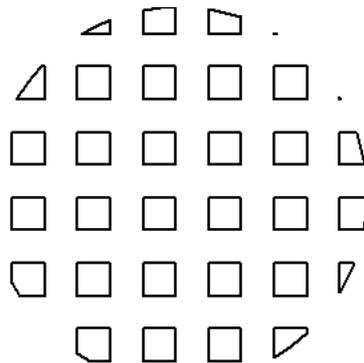
Select ACIS entities to subtract: **r**

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.



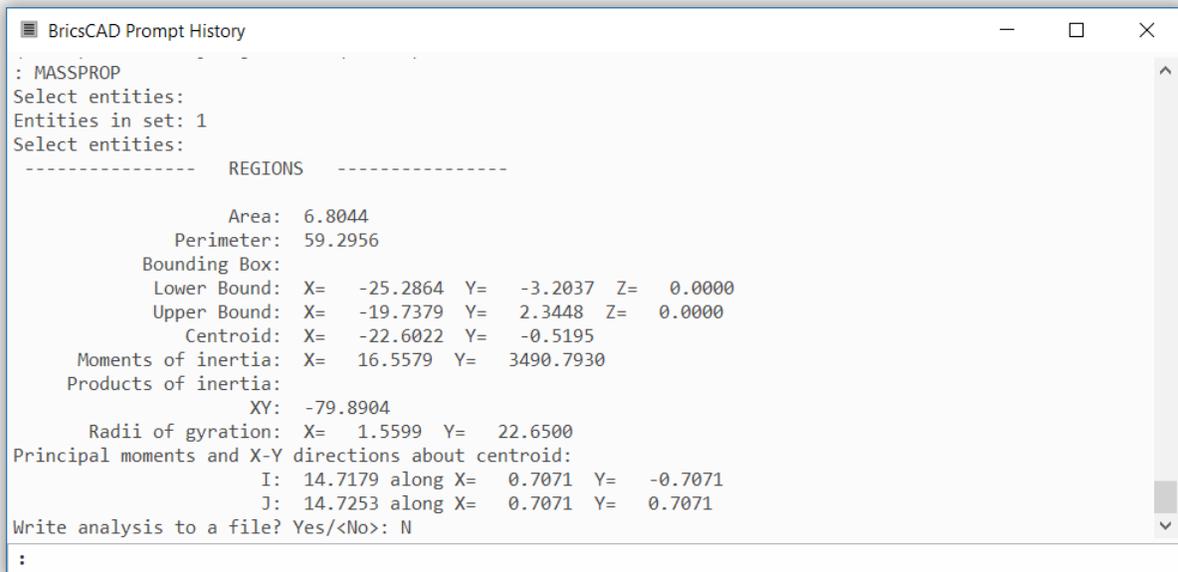
Measuring Regions

Command	MassProp
Menu Bar	Tools Inquiry Mass Properties
Ribbon	Tools Inquiry Mass Properties
Toolbar	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 the waffle shape)
Select entities: (Press **Enter**)
2. BricsCAD displays the results of the analysis in the text window. If necessary, press **F2**:



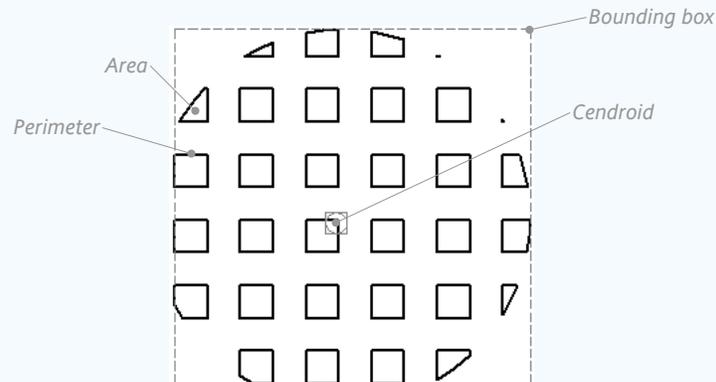
```

BricsCAD Prompt History
: MASSPROP
Select entities:
Entities in set: 1
Select entities:
----- REGIONS -----
                Area: 6.8044
                Perimeter: 59.2956
        Bounding Box:
        Lower Bound: X= -25.2864 Y= -3.2037 Z= 0.0000
        Upper Bound: X= -19.7379 Y= 2.3448 Z= 0.0000
                Centroid: X= -22.6022 Y= -0.5195
        Moments of inertia: X= 16.5579 Y= 3490.7930
        Products of inertia:
                XY: -79.8904
        Radii of gyration: X= 1.5599 Y= 22.6500
Principal moments and X-Y directions about centroid:
                I: 14.7179 along X= 0.7071 Y= -0.7071
                J: 14.7253 along X= 0.7071 Y= 0.7071
Write analysis to a file? Yes/<No>: N
:
```

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.

QUICK SUMMARY OF 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.

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; the modern technique is to manipulate 2D and 3D entities *directly*.

In this chapter, you learn how to create 3D models by directly manipulating elements, without necessarily using a command. (Direct modeling and editing are available only 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

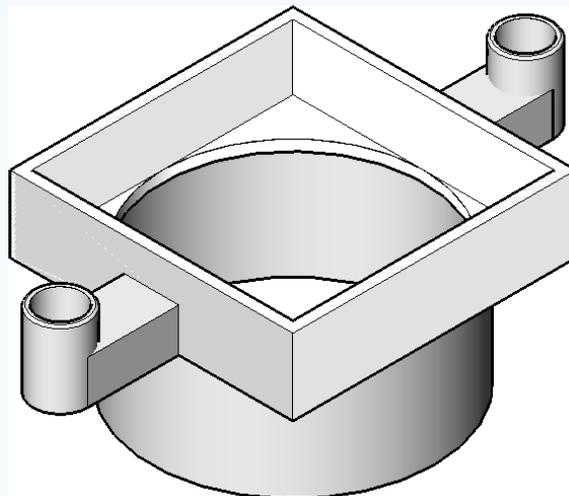
QUAD	Quad cursor
SUB	Sub-entity selection

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:

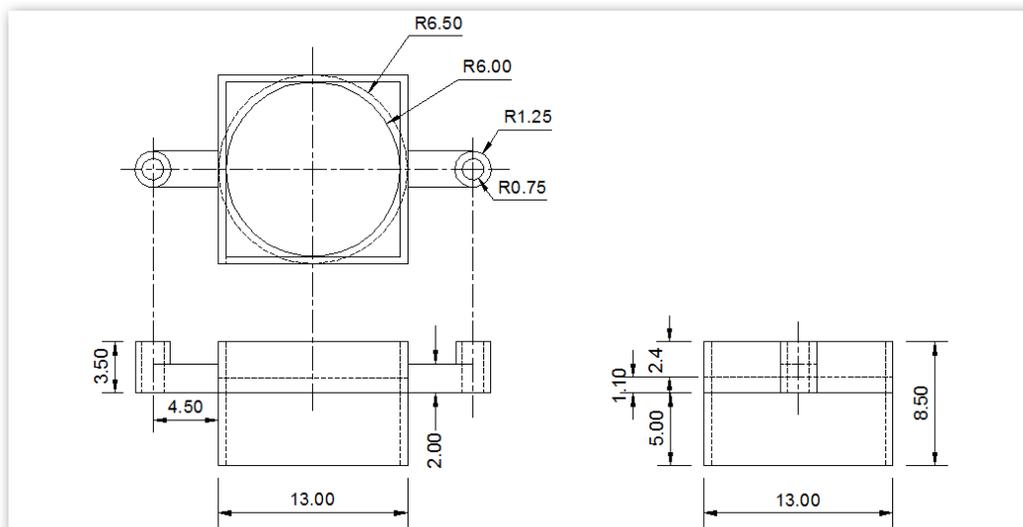


Focus-bracket drawing 3d-part.dwg

About 3D Solid Models

(**PRO AND PLATINUM EDITIONS ONLY**) 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. Being solid through and through is important, because they 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, both traditional commands and direct modeling. In these tutorials, you model the focus ring mount for a webcam. The 2D plans are illustrated below and finished 3D model is shown on the facing page. The dimensioned 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 Dropbox folder:

<https://www.dropbox.com/s/ft6gzw3r408ndgd/Inside-BricsCAD-Tutorial-Files.zip?dl=0>

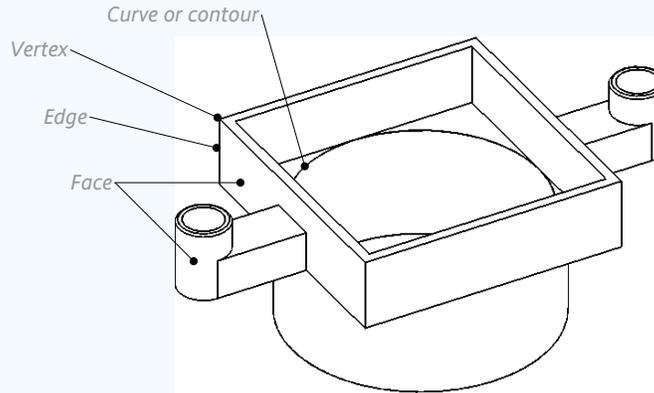
PLANNING AHEAD

There are differences between 2D drafting and 3D modeling. One is that 3D has a limited number of ways to create parts that make up models, whereas there are many way in 2D. For instance, to create a box shape in 3D, you use the **Box** command or extrude a rectangle; that's pretty much it; in 2D, by contrast, you can draw a rectangle with lines, polylines, traces, with **Rectangle**, **Polygon** commands, and so on.

ELEMENTS OF 3D MODELS

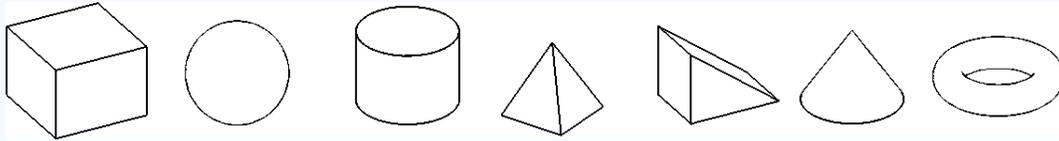
Three-dimensional modeling employs a jargon of its own. Here is a visual reference to some common terms.

PARTS OF A 3D MODEL



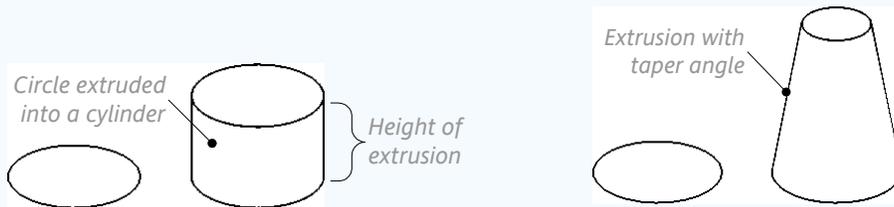
CREATE 3D MODELS FROM 3D PRIMITIVES

Primitives are basic 3D parts are made with the **Box**, **Sphere**, **Cylinder**, **Pyramid**, **Wedge**, and **Torus** commands.

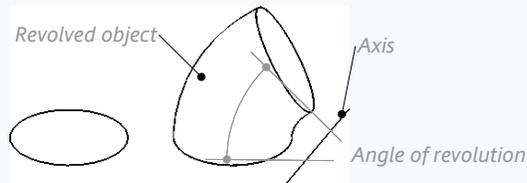


CREATE 3D MODELS FROM 2D ENTITIES

Apply the **Extrude** and **Revolve** commands to turn 2D entities into 3D ones:



Left: Applying the Extrude command to a 2D circle; **right:** Extruding the circle with a 5-degree taper



Extruding a circle to 45 degrees with the Revolve command, about an axis

Another difference is that you plan ahead to construct 3D models; this differs from 2D, where you simply start drawing. There is a Catch-22 here: you only become effective in planning ahead after you gain experience with 3D modeling, so this chapter shows you tricks to putting together 3D models.

Planning Ahead by Deconstructing 3D

The way to plan ahead is to deconstruct the model. Here are some tips:

- ▶ Look 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 made uniquely (made once).
- ▶ Rounded edges are added with the **Fillet** command.
- ▶ 2D drawings determine the dimensions of the part.

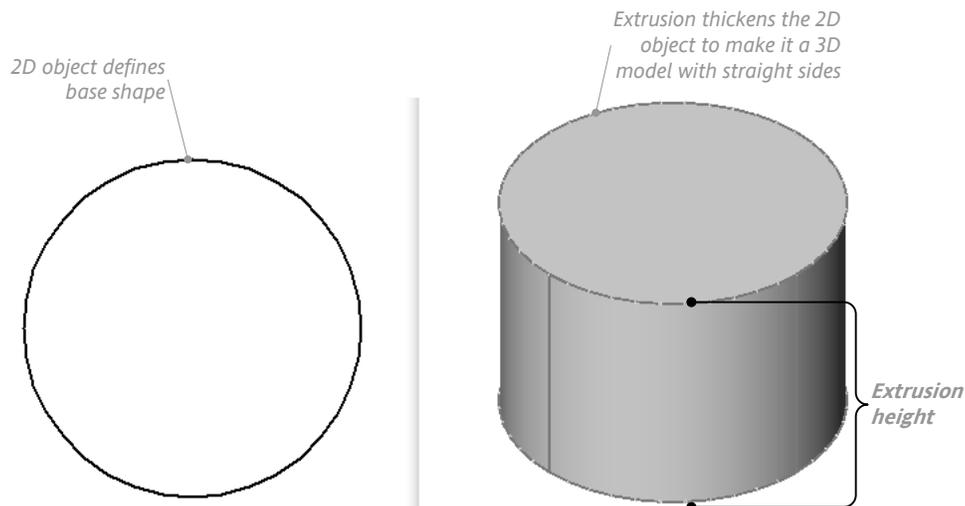


Examine the photograph (right) and identify sub-parts (a.k.a “primitives”) that could be modeled first. Notice that there are some square ones and some round ones. How would these be modeled? (Hint: with the **Box** and **Cylinder** commands.)

TRADITIONAL EXTRUSION METHOD

Because the sides are parallel, much of this part can be constructed from *extrusions*. An “extrusion” takes a 2D outline and then makes it taller to make it 3D.

To create the extrusion, you 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 have either *straight* or *slanted* sides. (If you want a part to have curved sides, then revolve the 2D entity with the Revolve command.)



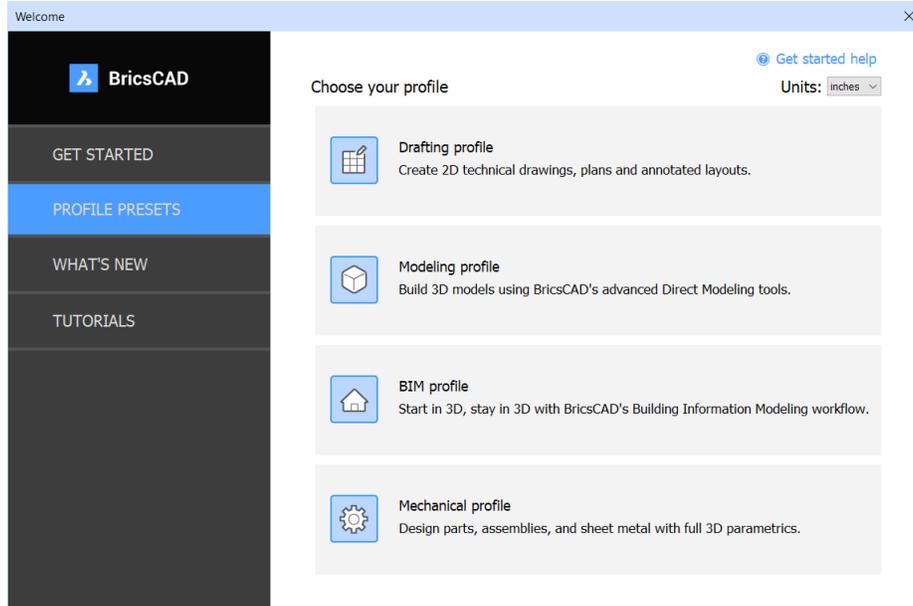
Left: 2D circle defining the diameter of the cylinder; **right:** Circle extruded to become a 3D cylinder

In this set of tutorials, you use variations on commands to 7model this webcam focus ring holder. In the first tutorial you use the traditional **Extrude** command, which in CAD dates back to the late 1980s; in a later tutorial, you get to use a newer approach to do the same thing known as “direct modeling.”

Preparing the Drawing

Here are the steps you need to take to prepare the drawing for 3D modeling.

1. Start BricsCAD.
2. Notice the Welcome dialog box. Click **Profile Presets**.



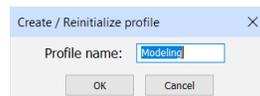
3. Change the **Units** to “Metric.”



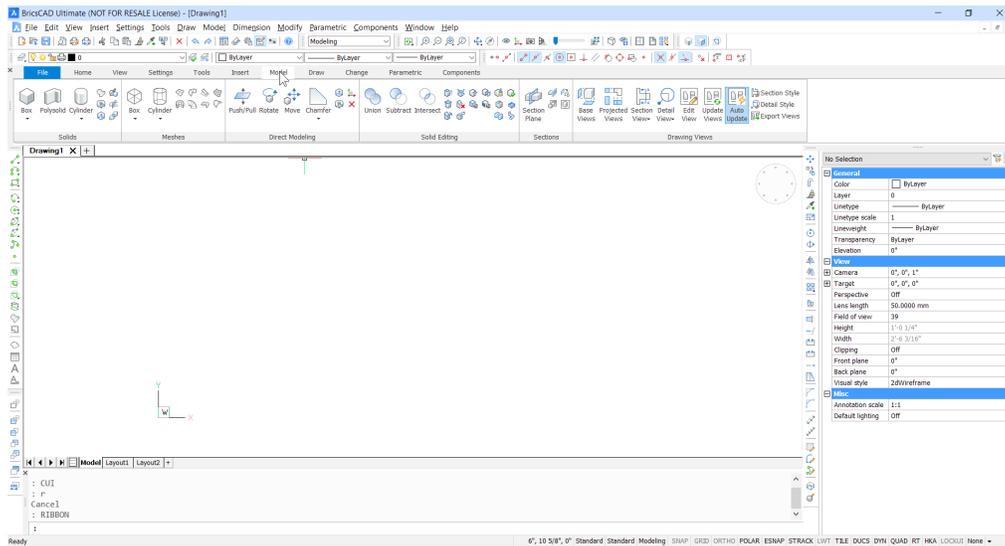
4. Click **Modeling Profile**.



5. When the Create/Reinitialize Profile dialog box appears, click **OK**.



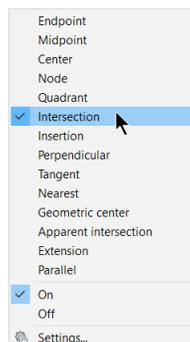
- Notice that BricsCAD opens in the Modeling workspace, but with toolbars. We'll use the ribbon in this tutorial, so enter the Ribbon command to turn it on.
- On the ribbon, click the **Model** tab.



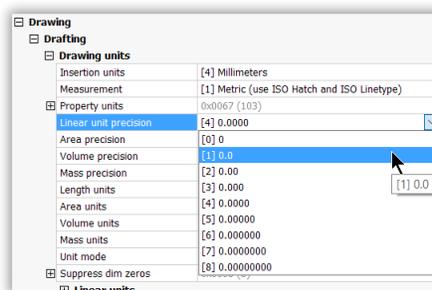
- Prepare this drawing by changing the following settings in the status bar:

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 INTERsection snap; turn off all others

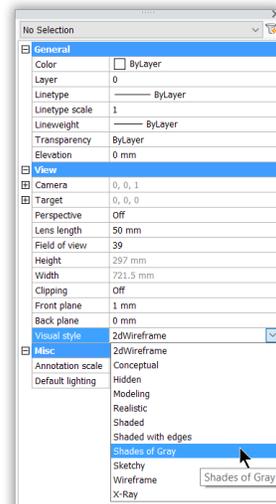
To turn on Intersection entity snap, right-click **ESNAP** on the status bar, as shown below. Ensure all other ensaps are turned off.



- Use the **Units** command to change the display precision of decimal places (linear units) to 1mm. You will be working with dimensions to the nearest 0.5mm.



10. Close the Settings dialog box by clicking the **X**.
11. Change the visual style to “Shades of Gray,” which I like the best for 3D modeling. You do this through the Properties panel.
 - a. If the panel is not open, then enter the **Properties** command.



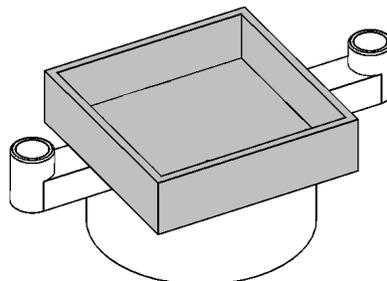
- b. In the **View** section, click the **Visual Style** droplist
 - c. Choose “Shades of Gray.” (By the way, there are 256 shades of gray, not just 50.)
12. Finally, change the value of **DelObj** to **0**. Recall from a previous lesson that this system variable determines what happens to 2D entities 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

Command Extrude
Alias ext
Menu Models | 3D Solids | Extrude
Ribbon Solid | Create | Extrude
Toolbar 3D Solids | Extrude

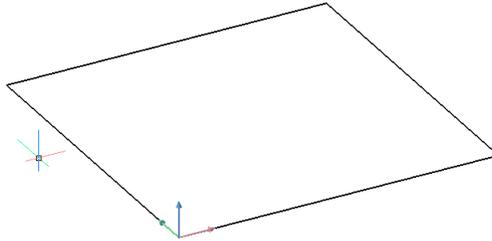
Command SolidEdit
Menu Model | 3D Solid Editing | Shell
Toolbar 3D Solid Editing | Solid Edit
Ribbon Solid | Edit | Shell

The top of the part is a square open box. It is shown in gray by the figure below.



To make it, you'll work through three commands, in this order:

- a. **Rectangle** command defines the size and base of the square
 - b. **Extrude** command changes the 2D square into a solid 3D box
 - c. **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

Choose first corner of rectangle or [Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/Area/Dimensions]: **d**

Length to use for rectangles <13.0>: **13**

Width to use for rectangles <13.0>: **13**

Select first corner of rectangle or [Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/Area/Dimensions]: **0,0**

Other corner of rectangle: *(Pick a point; it's not crucial where you pick it)*

QUICK SUMMARY OF EXTRUDE COMMAND

Extrude commands thicken 2D entities into 3D solids or surfaces (included in BricsCAD Pro and Platinum).

: extrude

Select entities/subentities to extrude or set [M0de]:

Specify height of extrusion or [Direction/Path/Taper angle]:

Entities/subentities — selects 2D closed entities or faces of 3D solids

M0de — specifies the result of the extrusion, solid or surface

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

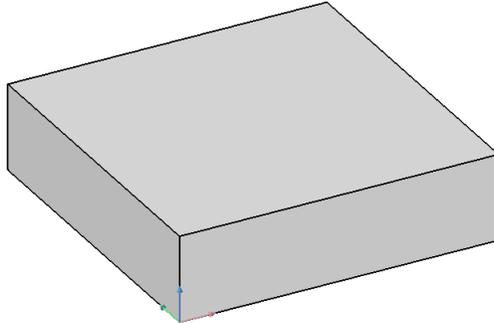
2. If necessary, use the **Zoom Extents** to see it better.
3. Now use the **Extrude** command to convert the square into a short 3D box. Specify a height of 3.5mm, as follows:

: **extrude**

Select entities/subentities to extrude or set [MOde]: *(Pick the square)*

Select entities/subentities to extrude or set [MOde]: *(Press ENTER to end selection)*

Specify height of extrusion or [Direction/Path/Taper angle]: 3.5



4. “Shell” is an option of the SolidEdit command that turns solid entities into “walls” — it uniformly removes the insides. Follow these steps to make the box hollow:

- a. Because the SolidEdit command has many options, it is faster to get to the Shell option via the ribbon. 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.)

: **_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**

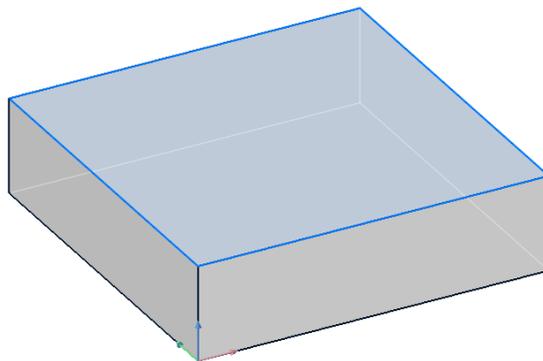
- b. As prompted, select the extruded box:

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:

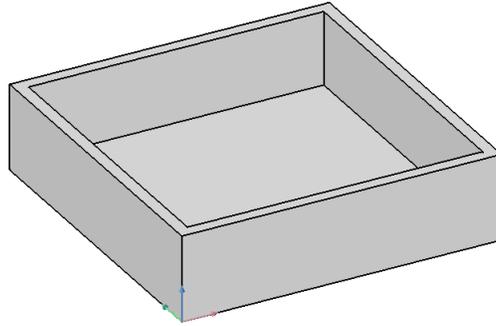
Enter the shell offset distance: .5

e. And finally press **Enter** twice to exit the command:

Enter a body editing option Imprint/seParate/Shell/cLean/Check/Undo/eXit: (Press **Enter**)

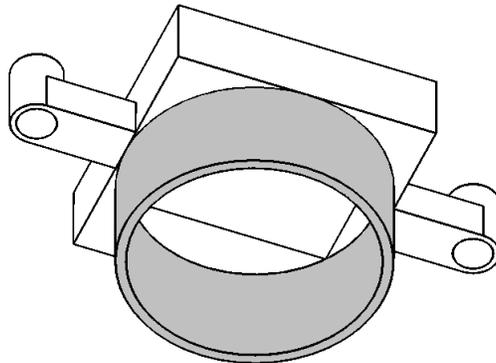
Enter a solids editing option Face/Edge/Body/Undo/eXit: (Press **Enter**)

Notice that the result is a hollow box with no top.



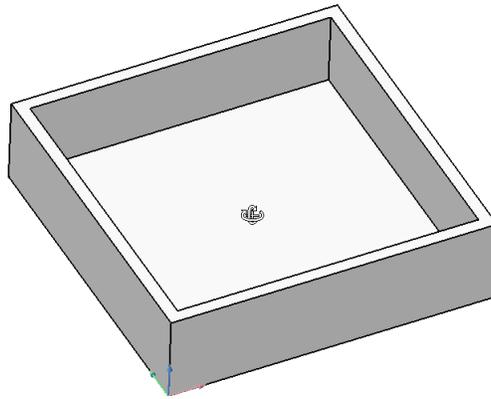
3D View Rotation

The next step is to add the round part underneath the box, shown in gray below.

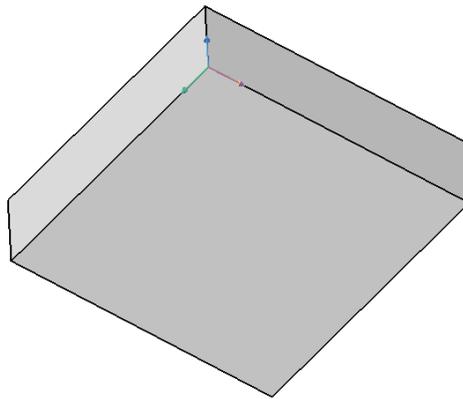


But to do this, you need to spin the model so that you can see the bottom of it. There are several methods you could employ, but the easiest is the most direct way: real-time rotation. Here's how:

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

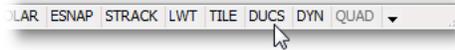
The bottom of the part has a hollow cylinder. In this part of the tutorial, you draw two circles and then extrude them using *direct modeling* — this means you employ no 3D commands to do the work. But first, this word of explanation of how to draw in 3D space...

Applying Dynamic UCS

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. And so a crucial aspect to direct modeling is working with the correctly-positioned plane. No matter which 3D CAD package you use, they all fall back to drawing on the x,y -plane.

Fortunately, CAD systems today make it easy to place 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.

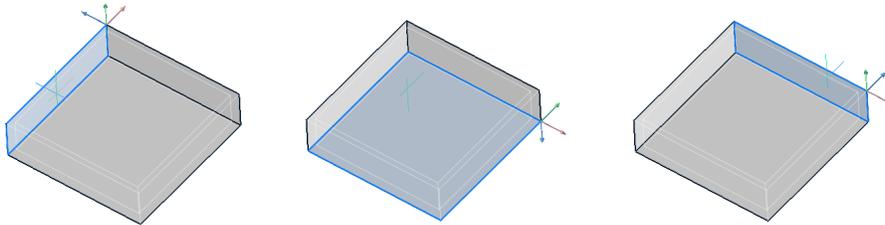


2. Start to draw a circle on the bottom of the box with the **Circle** command and its **2P** option, as follows:

: **circle**

Select center of circle or [2Point/3Point/TanTanRad/Arc/Multiple]: **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 cursor passes over face, they turn blue and the UCS icon is relocated (tri-color icon)

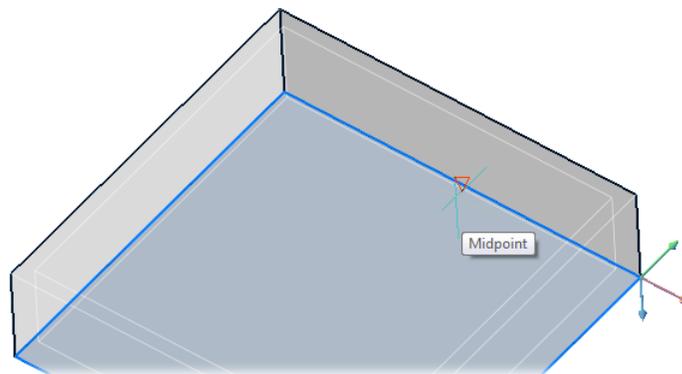
In summary: As the cursor passes over a face during DUCS, the face turns blue to tell you that it is the current, temporary x,y drawing plane. The UCS icon is positioned at the current, temporary origin (0,0,0). DUCS is available only during drawing and editing commands, because most of them operate only on an x,y plane.

DUCS works only during drawing and editing commands.

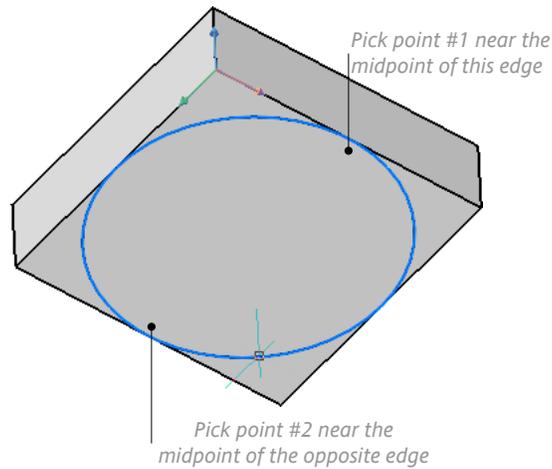
If you want to see a more dramatic effect of dynamic UCS at work, turn on the grid display.

(Click **GRID** on the status bar.) As you move the cursor from face to face, the grid jumps to match the face.

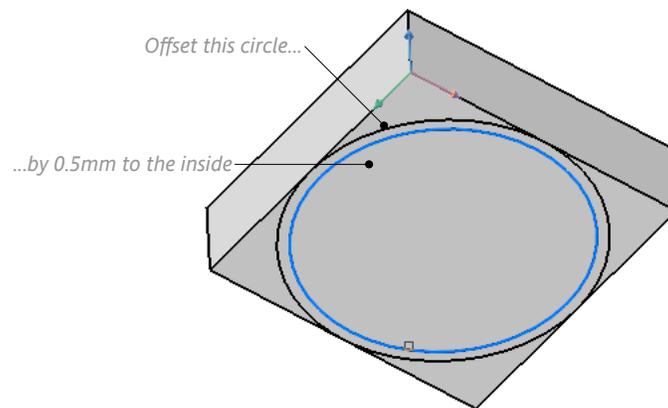
4. To locate the circle, pick two points on the edges of the box, right at the middle of two facing edges. Use the **MIDpoint** ensap to assist you.
 - a. Position the cursor over the bottom face of the box. Notice that it turns blue.
 - b. Move the cursor close to the edge illustrated below, but do not go beyond the edge! (Were you to go beyond the edge, then the DUCS would jump to the adjacent face, telling BricsCAD that you want to draw the circle on a different 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

Enter offset distance or [Through point/Erase/Layer] <Through point>: .5

Select entity/subentity or [Exit] <Exit>: (Select the black circle)

Select side for parallel copy or [Both sides/Multiple]: (Pick a point inside the circle)

Select entity/subentity or [Exit] <Exit>: (Press Enter to end the command)

Extruding Directly with the Quad Cursor

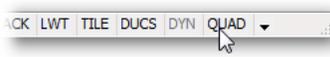
With the pair of 2D circles in place, you extrude them to create a 5mm-tall hollow cylinder. This time, you use the Quad to extrude. The Quad is unique to BricsCAD, and gives you fast access to commands right at the cursor. In many cases, commands suitable to the highlighted element are presented to you. (See the boxed text for more on using the Quad.)

Cylinders are extruded from circles. Making a hollow cylinder takes these steps:

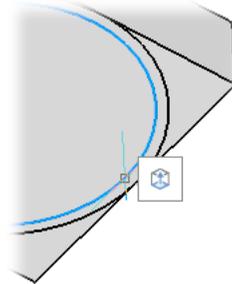
- Extrude the inside circle into a cylinder that is more than 5mm in height
- Extrude the outside circle by exactly 5mm
- Subtract the taller cylinder from the shorter to make the hole.

You extrude the inside circle by a distance longer than 5mm, because its height does not matter; indeed, in a later step, you “erase” it to create a hole. Follow these steps:

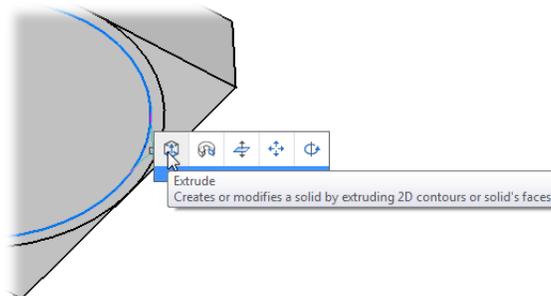
- On the status bar, ensure that **QUAD** is turned on.



- 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).

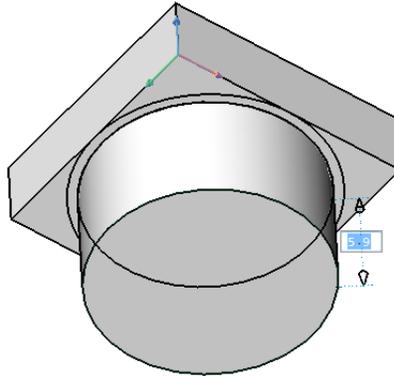


- The icon  on the cursor indicates the Extrude command. To confirm, pass the cursor over the icon. Notice that the Quad cursor expands to show more commands; after a moment the tooltip appears explaining the name and purpose of the button.



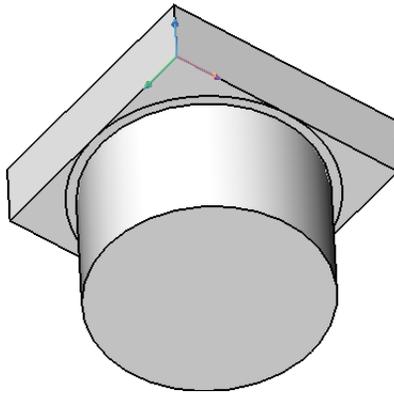
- Click the **Extrude**  button, and then move the cursor. As you do, notice that the circle extrudes into a cylinder (i.e., thickens in 3D). The direction you move the cursor determines the direction the circle extrudes, upwards or down.

5. 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.

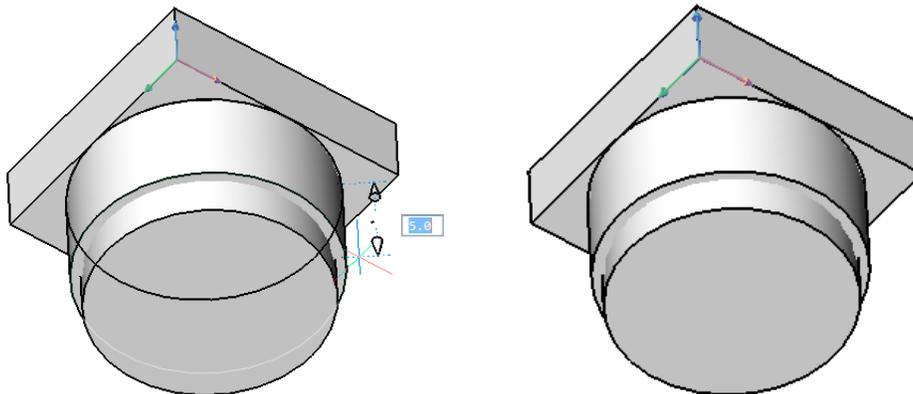


Watching the dynamic dimension, drag the extrusion so that it is longer than 5mm — like 7 or 8mm.

6. Then click. The cylinder is formed — without entering commands.



7. 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

8. Save your work with **Ctrl+S**, naming it "Focus-Bracket.dwg."

Subtracting Solids To Make Holes

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.

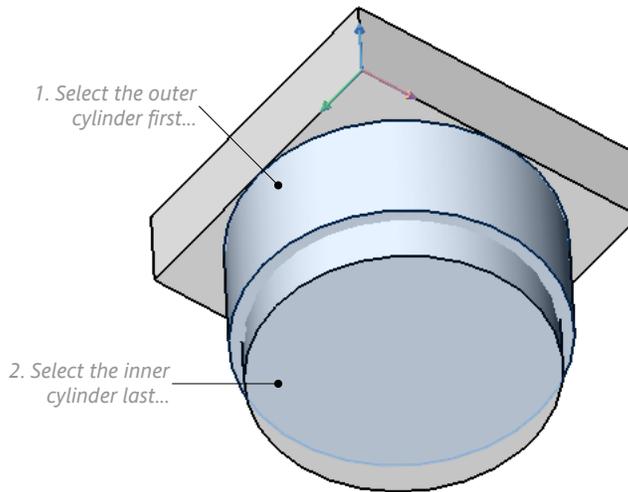
The **Subtract** command is sensitive to the order in which entities are selected. Make sure you pick the correct ones in the correct order: first the one that stays, and then the one that will be removed (subtracted).

1. 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)

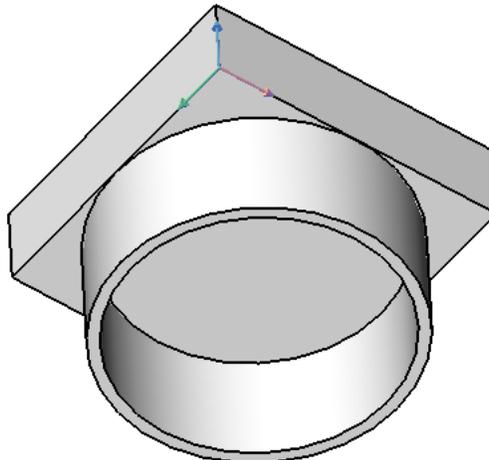


2. Now pick the cylinder to be removed (erased):

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.

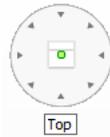
1. The easiest way to draw a circle (and other 2D entities) is in plan view. The quick way to switch between static viewpoints — such as the plan and isometric viewpoints — is to use the Look From widget. To use it, follow these steps:
 - a. Locate the Look From widget. Usually, it is in the upper right corner of the drawing area.

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.

QUICK SUMMARY OF LOOKFROM COMMAND

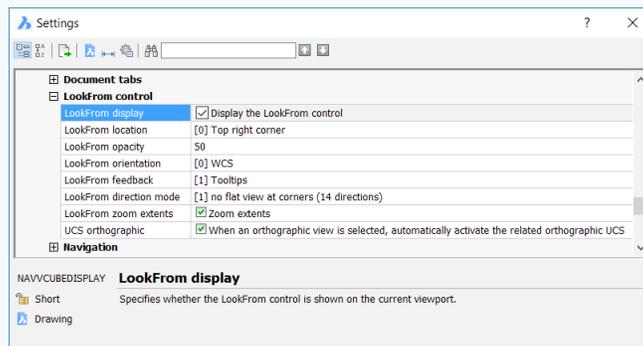
```
: lookfrom
```

```
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



- To draw the circle, use the same technique as before: employ the **Circle** command with the **2P** option and **MIDpoint** esnaps:

: c

Select center of circle or [2 Point(2P)/3 Point(3P)/Tangent-Tangent-Radius(TTR)/

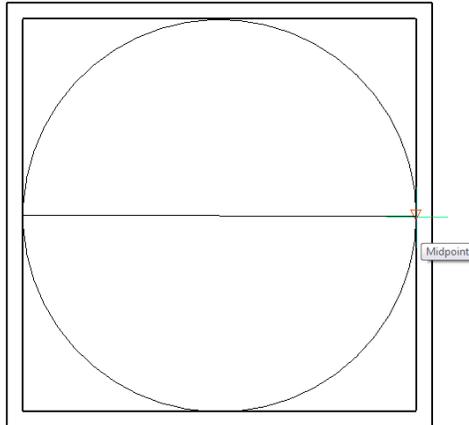
Turn arc into circle(A)/Multiple circles]: **2p**

First point on diameter: mid

Snap to midpoint of: (*Pick one edge*)

Second point on diameter: mid

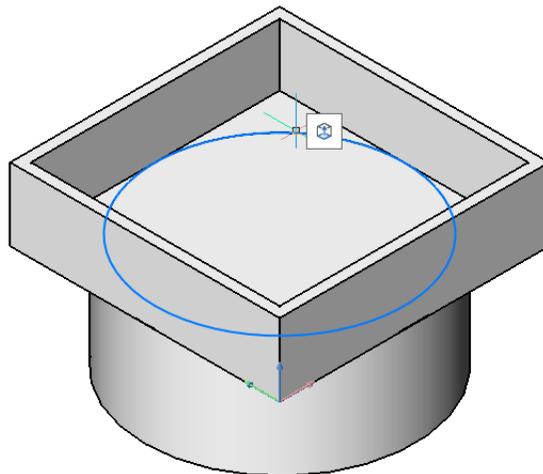
Snap to midpoint of: (*Pick the opposite edge*)



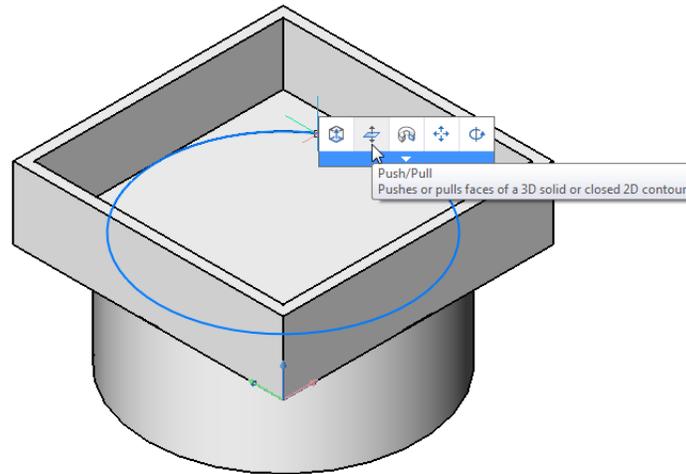
- Using the Quad cursor, punch out the circle using direct modeling. You may find it easier to control the extrusion by clicking the **Top Front Left** position on the Look From widget.



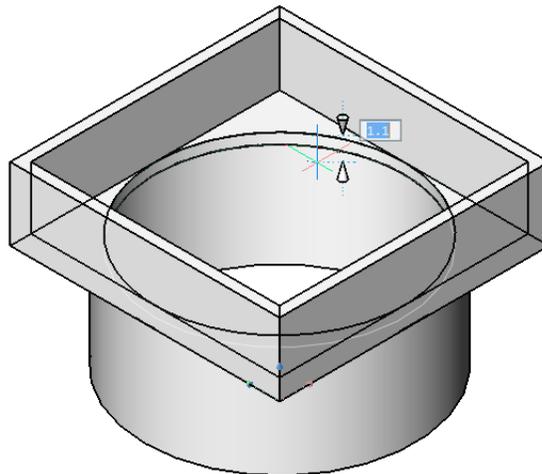
- Pass the cursor over the newly added circle. Notice that it turns blue.



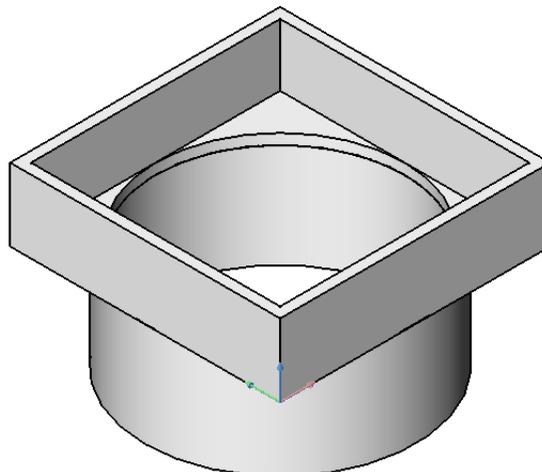
- b. Move the crosshair cursor into the Quad cursor. Notice that it expands to show more commands. Choose the **PushPull**  button.



- 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.



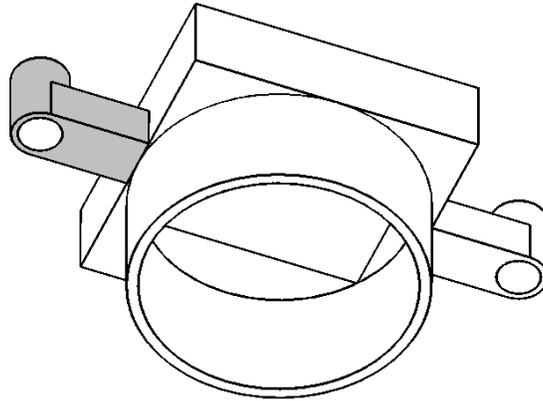
- d. Click to finish the hole. This was quicker and easier than using the Extrude command!



Aligning the UCS

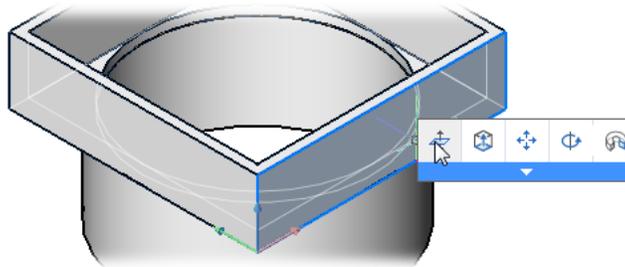
Command UCS, Face option

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.

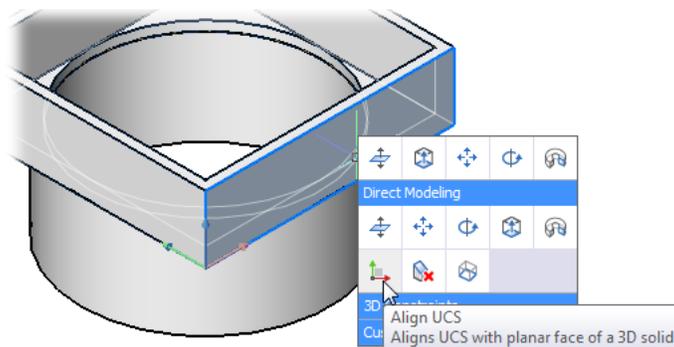


In this case, the 2D entity is a rectangle. Before drawing the rectangle, however, you align the UCS on the face by another method: align UCS. This permanently reorients the UCS plane to a face — permanent until you change it again. This is different from the temporary, dynamic UCS alignment you used 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.

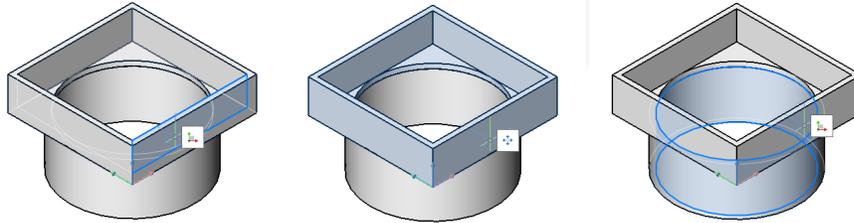


3. Move the arrow cursor into the blue Direct Modeling title bar. Notice that the Quad cursor expands to show more icons.



4. Choose the **Align UCS**  button, and then press **Enter**. The UCS plane is now fixed on the face.

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.

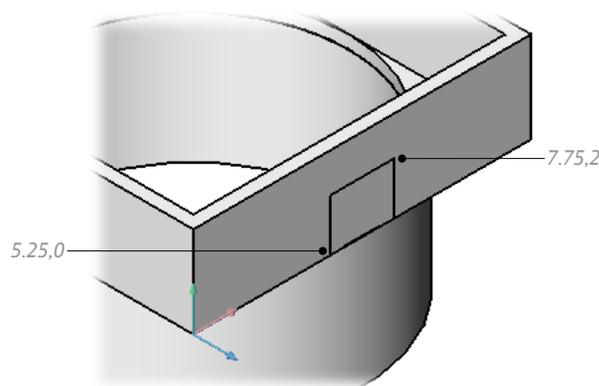


Pulling, Instead of Pushing

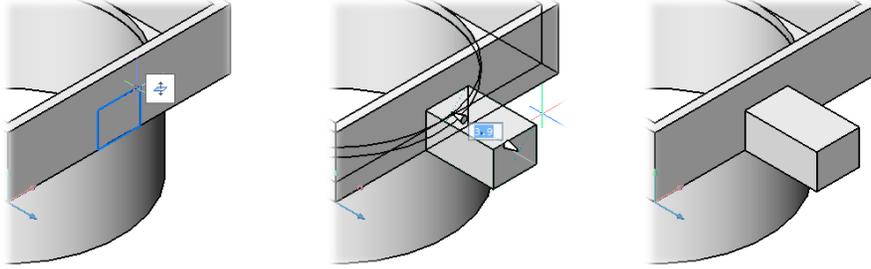
Command	dmPushPull
Menu	Model Direct Modeling Push/Pull
Ribbon	Solid Edit Push/Pull
Toolbar	Direct Modeling Push/Pull

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**
Choose first corner of rectangle or [Chamfer/Elevation/Fillet/Rotated/Square/Thickness/Width/Area/Dimensions] **5.25,0**
Other corner of rectangle: **7.75,2**

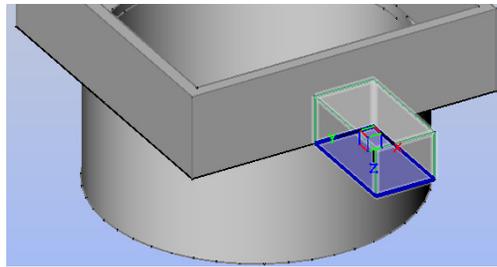


- Use the **dmExtrude** command to pull the profile to a length of **4.5mm**.



Left to right: Choose the rectangle (a.k.a "profile"); select dmExtrude command and specify extrusion distance of 4.5mm; done!

- 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.

- 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 = elevation)

- Using the Isometric Views toolbar, change the viewpoint to the top (or plan) view.

- Use the Circle command to draw the round profile:

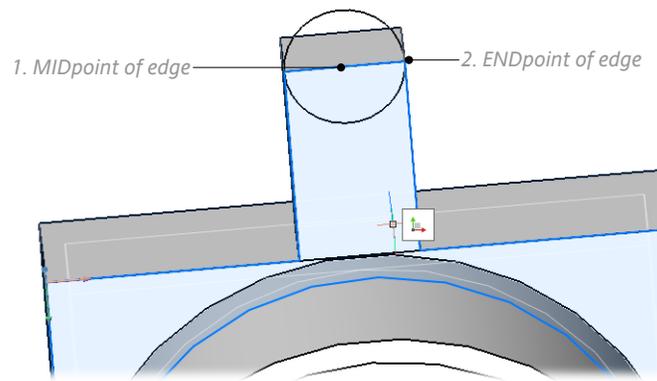
: circle

Select center of circle or [2 Point(2P)/3 Point(3P)/Tangent-Tangent-Radius(TTR)/Turn arc into circle(A)/Multiple circles]: **mid**

Snap to midpoint of: (*Pick the midpoint of the edge.*)

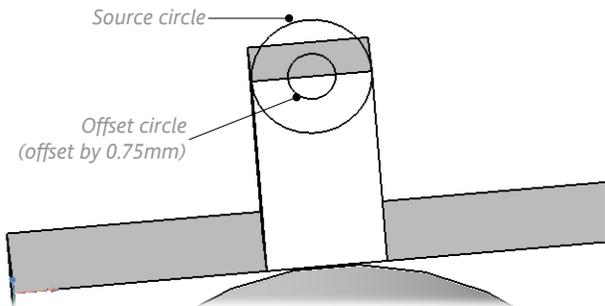
Set Diameter or [Radius] <1.3>: end

Snap to endpoint of: (Pick the end of the edge)

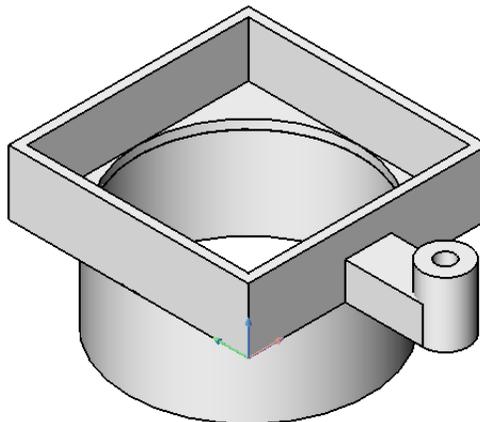


When you rotate the view back to isometric, you see the circle placed on the bottom face, thanks to OsnapZ.

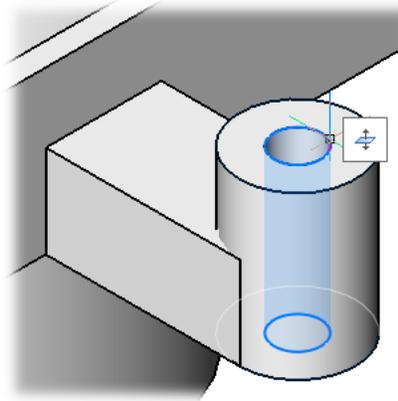
7. Use the **Offset** command to place a second circle, offset by **0.75mm**. (This value is deliberately incorrect, and you correct it later with direct editing.)



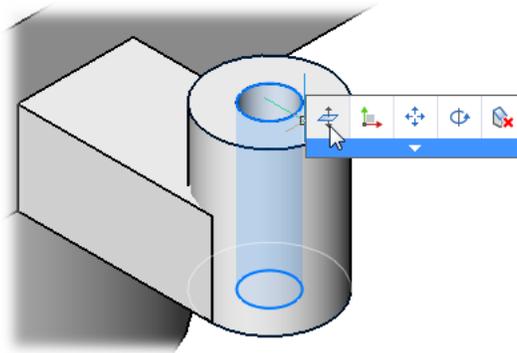
8. As in the earlier tutorials, use direct modeling to extrude the two circles into cylinders:
 - a. Use Extrude to turn the outer circle in a cylinder with a height of precisely 3.5mm.
 - b. Use the **dmExtrude** command to turn the inner one to any height taller than 3.5mm; notice that it creates the hole automatically — no need to use the Subtract command!



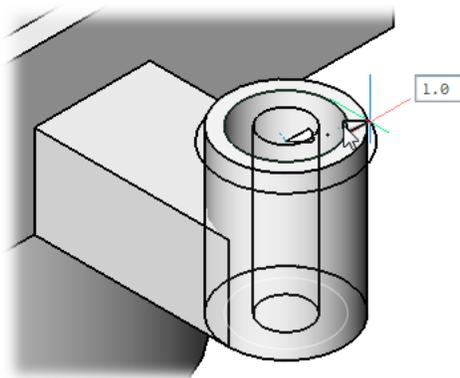
9. The diameter of the hole is too large. It should be 1mm, but is currently 0.5mm. BricsCAD can edit solid models using direct editing. Here is how to correct the diameter of the hole:
- Move the cursor over the edge of the hole. If the hole does not highlight in blue, then press the **Tab** key until it does.



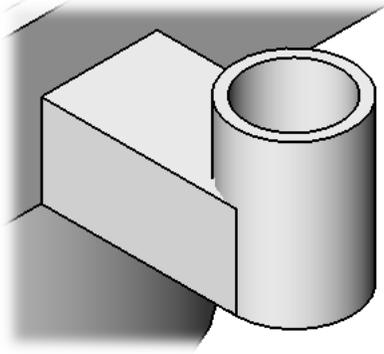
- In the Quad cursor, choose the **dmExtrude**  button.



- 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.



To change the UCS back to “normal,” enter the **UCS** command, and then choose the **World** option:

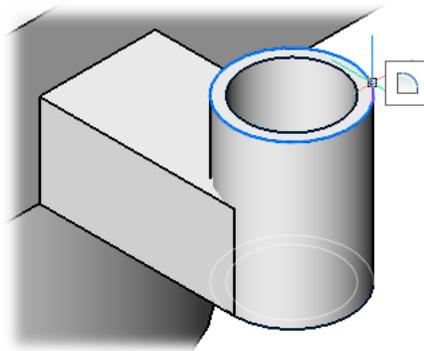
: **ucs**

Specify origin of UCS or [Face/NAMED/Entity/Previous/View/X/Y/Z/ZAxis/Move/
World]<World>: **w**

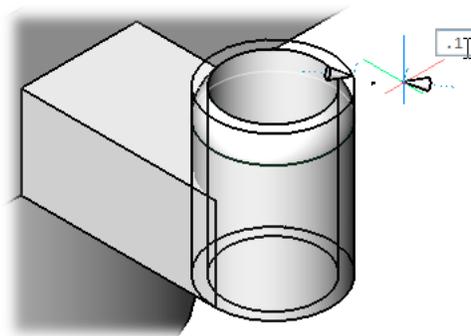
Filleting 3D Solids

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:

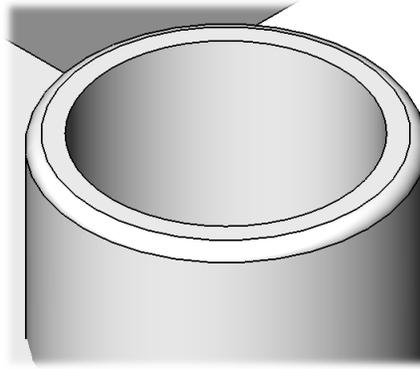
1. Move the cursor over the edge of the cylinder. Remember: if you cannot select the outer edge, then press **Tab** until BricsCAD highlights it. See the figure below.



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)

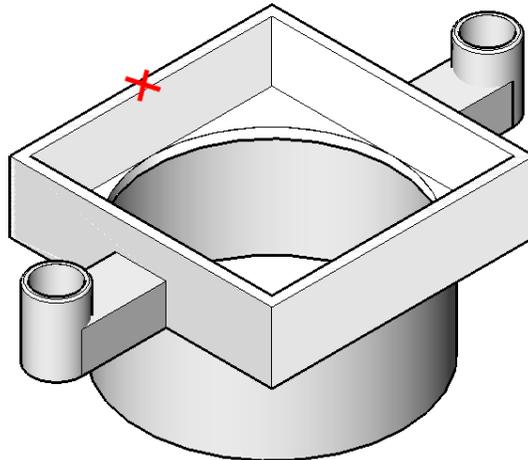
Start of mirror line: mid

Snap to midpoint of: (Pick middle of rectangular part, as illustrated below)

End of mirror line: mid

Snap to midpoint of: (Pick a point on the other side)

Delete the original entities? [Yes-delete entities/No-keep entities] <No-Keep entities>: 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)

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.

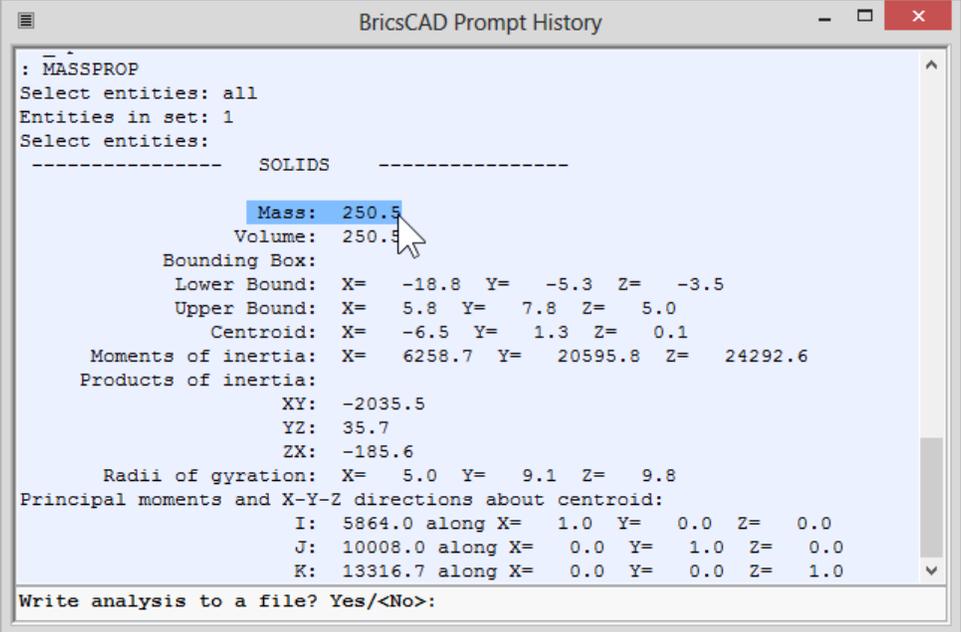
: **massprop**

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.



```
: MASSPROP
Select entities: all
Entities in set: 1
Select entities:
----- SOLIDS -----
      Mass: 250.5
      Volume: 250.5
      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>:
```

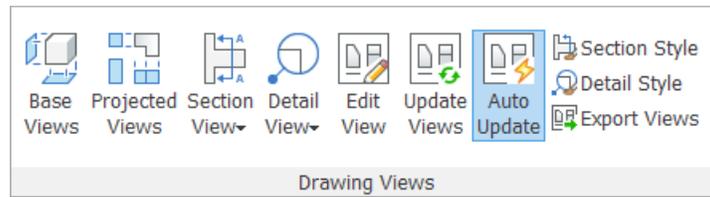
3. Save the drawing.

Generating 2D Drawings

Command	ViewBase
Menu	Model Direct Modeling Drawing Views Generate Drawing Views
Ribbon	Annotation Views Base Views
Toolbar	Drawings Views Base View

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 **Drawing Views** panel of the ribbon's **Model** tab. It contains the commands you need for this tutorial.



Click the  **Base Views** button. (It runs the ViewBase command). This command semi-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:

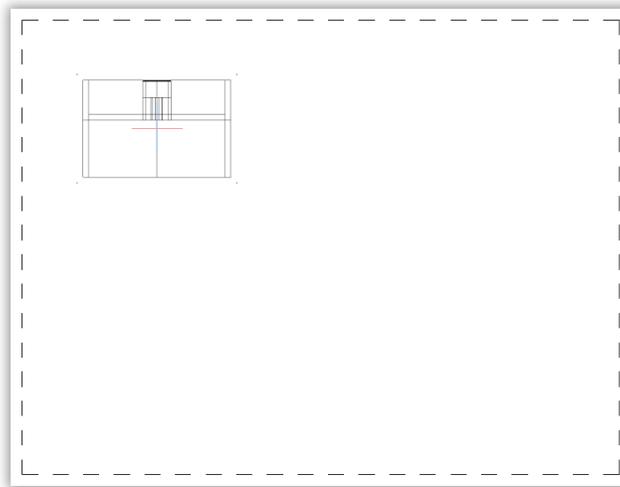
Enter new or existing layout name to make current <Layout1>: (Press **Enter**)

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.

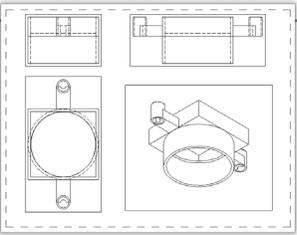


In this step of the the tutorial, position the cursor in the upper left quadrant, and then click.

Select position for base view [Scale/Hidden lines/Tangent edges/Orientation/Projection type/Isometric style/sElect] <Cancel>: (Move the cursor to upper left, and then click)

The quadrant of the layout in which you click determines the viewpoint generated by BricsCAD:

Quadrant	2D View Generated
Upper left	Top view
Upper right	Side view
Lower left	Front view
Lower right	Isometric view



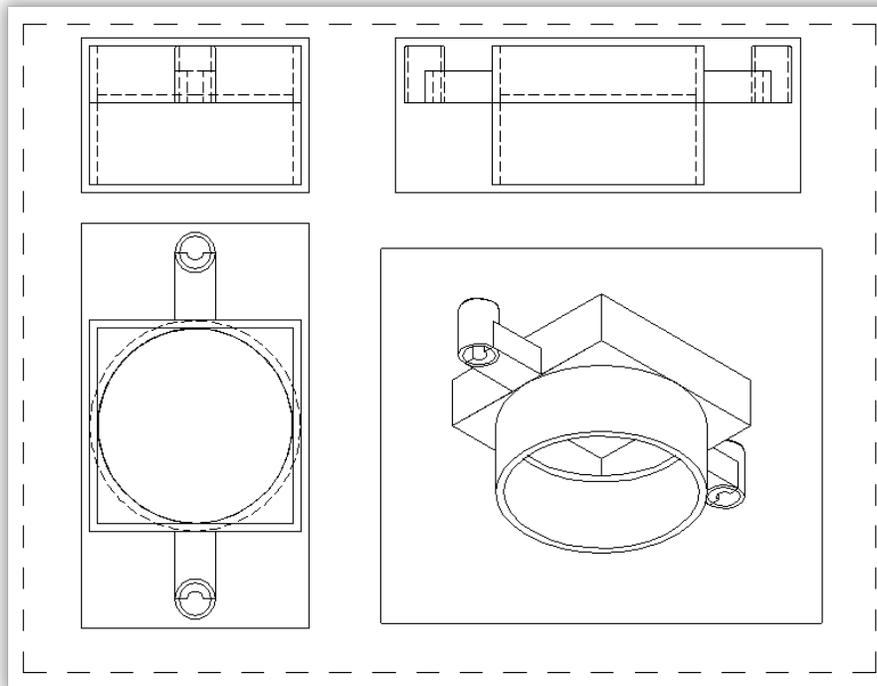
- The command repeats its prompt. Repeat the process: move the cursor to a quadrant of the layout, and then click to position the view.

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**)

QUICK SUMMARY OF VIEWBASE OPTIONS

: **viewbase**

Select objects or [Entire model/presets] <Entire model>:

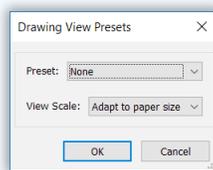
Enter new or existing layout name to make current <LayoutName>:

Select position for base view [Scale/Tangent edges/Orientation/Projection type/Isometric style]<Cancel>:

Select position for projected view <Done>:

PRESETS OPTION

Displays the Drawing View 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 loded 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:

HIDDEN LINES OPTION

Determines whether hidden lines are shown.

Continued...

Continued...

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.

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.

ADDING SECTION VIEWS AND DETAILED VIEWS

Command	ViewSection
Menu	Model Drawing Views Section Types
Ribbon	Annotation Drawing Views Section View
Toolbar	Drawing Views Section Types
Command	ViewDetail
Menu	Model Drawing Views Detail View Types
Ribbon	Annotation Drawing Views Detail View
Toolbar	Drawing 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.

QUICK SUMMARY OF VIEWSECTION OPTIONS

: viewsection

Select drawing view:

Specify start point of section line or select type [Full/Half/Offset/Aligned] <Full>:

Specify next point of section line [Done] <Done>:

Select position for section view:

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

TYPE OPTION

- ▶ **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 per 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:

GEOMETRY OPTION

Defines the visual style for the section view: conceptual 3D view or 2dWireframe visual style; alternately, select a visual style from the Properties panel. (Hidden Line option is moved to the ViewBase command.)

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 style for isometric views: rendered using Conceptual visual style, wireframe, or other one.

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.

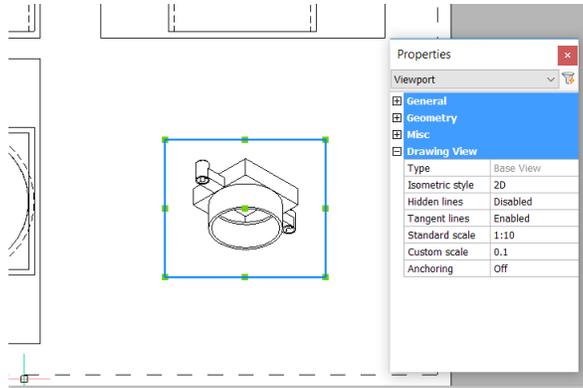
DEPTH OPTION

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.

- In the Properties panel, click **Custom Scale**.



- Enter a new value, such as **.1**.
- 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.

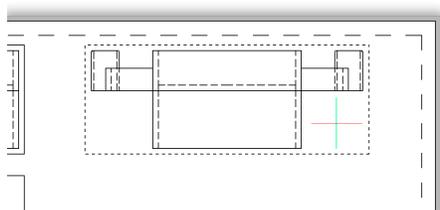
Making Section Views

Okay, with space freed up, you can go ahead to make the section view:

- From the ribbon's **Model > Drawing Views** panel, click the  **Section View** button (or else enter the **ViewSection** command).
- 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)

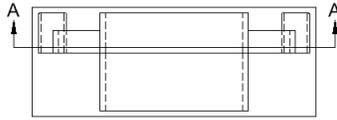


Notice that BricsCAD highlights the selected viewport with dashed lines.

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.

- 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:

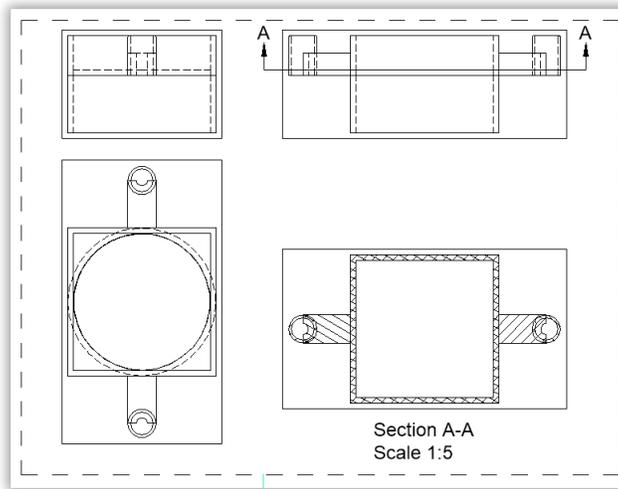
Specify start point of section line or select type [Full/Half/Offset/Aligned] <Full>:
(Pick a point outside the viewport)
Specify next point of section line [Done] <Done>: (Pick a second point on the other
side of the viewport)



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)



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 is used by drafters to describe two types of information:

- ▶ The **presence** of hatching shows that areas of the are solid; areas without hatching are empty air.
- ▶ The **style** of hatching indicates the type of material; “ANSI31” hatching is the default pattern, and represents iron.

QUICK SUMMARY OF VIEWDETAIL OPTIONS

: **viewdetail**

Select drawing view:

Specify detail center or select boundary type [Circular/Rectangular] <Circular>:

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>:

CIRCULAR OPTION

Creates a circular boundary, prompting you for the radius.

RECTANGULAR OPTION

Creates a rectangular boundary. You can still change the boundary type with the **Boundary** option in the last prompt of this command.

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.

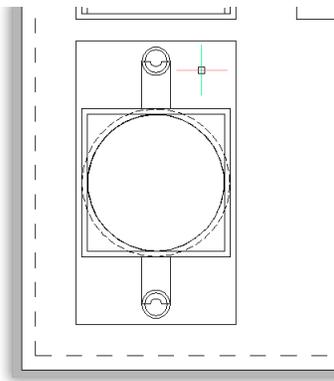
Here is the meaning of the ANSI hatch patterns included with BricsCAD:

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, babbitt, and alloys of them
ANSI38		Magnesium, aluminum, and alloys of them

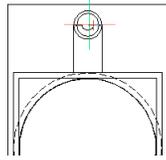
Creating Detail Views

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*)

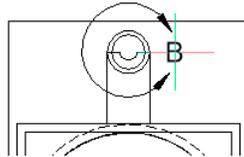


- Pick the center of the detail, as BricsCAD will use a circle as the view's boundary:
Specify detail center or select boundary type [Circular/Rectangular] <Circular>:(Pick the center of what will be detailed)

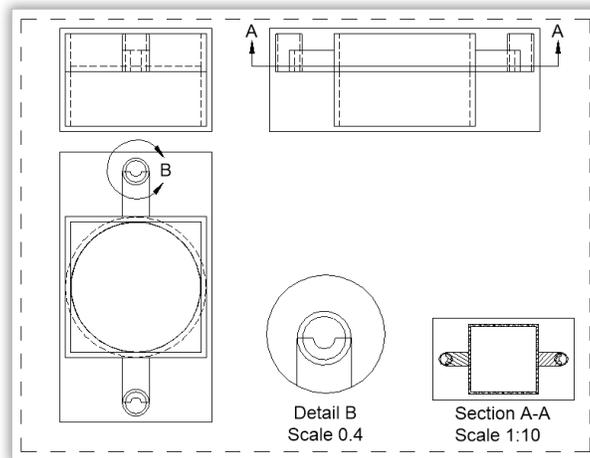


- Now size the circle to indicate the extent of the detail. You can see the circle in the figure above, it has the "B" reference attached.

Select radius of detail view: (Drag the circle to size the detail area)

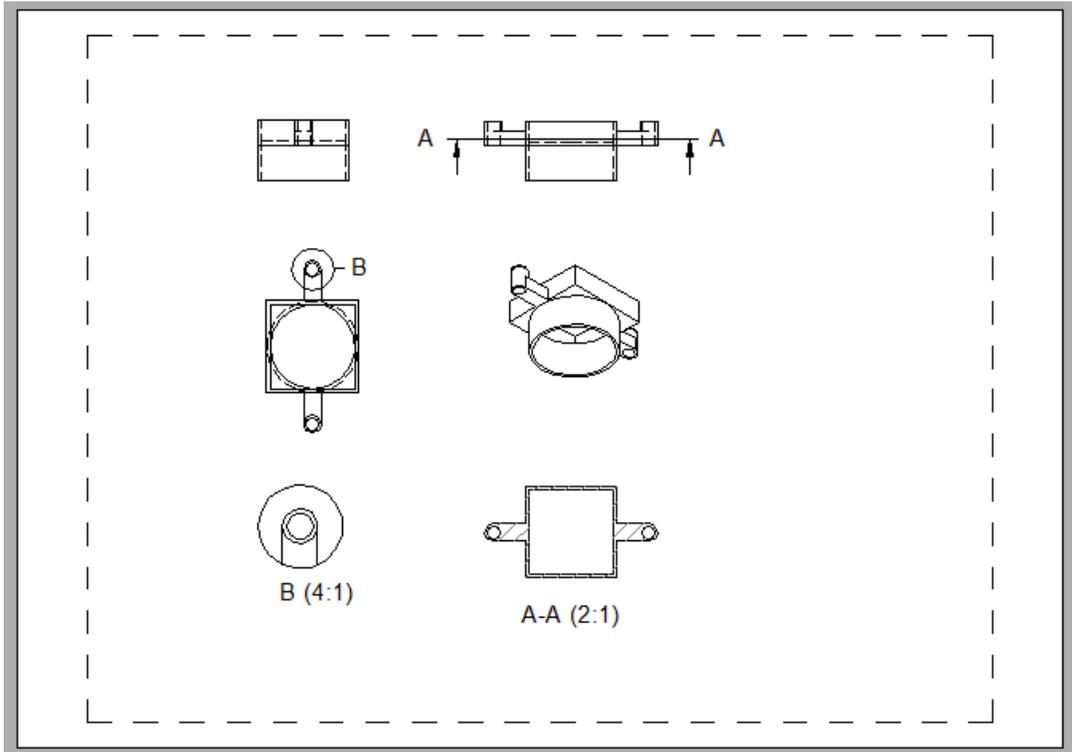


- Position the detail view somewhere in the drawing:
Select position for detail view [Scale] <Cancel>: (Pick a point)



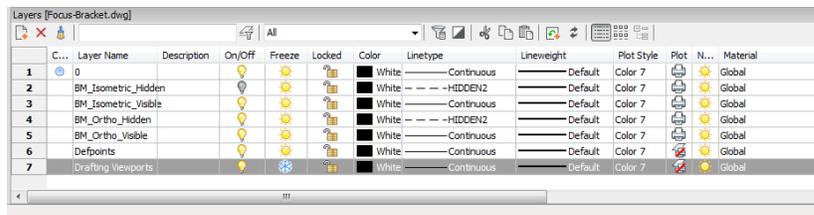
BricsCAD automatically labels the detail using the standard method of "B," which is used to referenced the source of the detail drawing. It also indicates the scale factor of the section, 4:1. You can use the Scale option to make the detail larger or smaller.

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.



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.



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*. This is where dimensions determine the sizes of entities, and not the other way around. It is where geometry is locked into place to locate entities relative to one other. Together with constraints, parameters determine the positions of entities through formulae.

In this chapter, you learn how to apply dimensional and geometric constraints to 2D drawings.

IN THIS CHAPTER

- Applying dimensional constraints
- Using 2D geometric constraints
- Controlling constraints through parameters

KEY TERMS IN THIS CHAPTER

Constraint bar — describes the small tooltips that report the constraints applied to entities

Dimensional constraint — specifies the size of entities in a dimension-like manner

Geometric constraint — determines the geometric relationship between entities, much like a semi-permanent entity snap

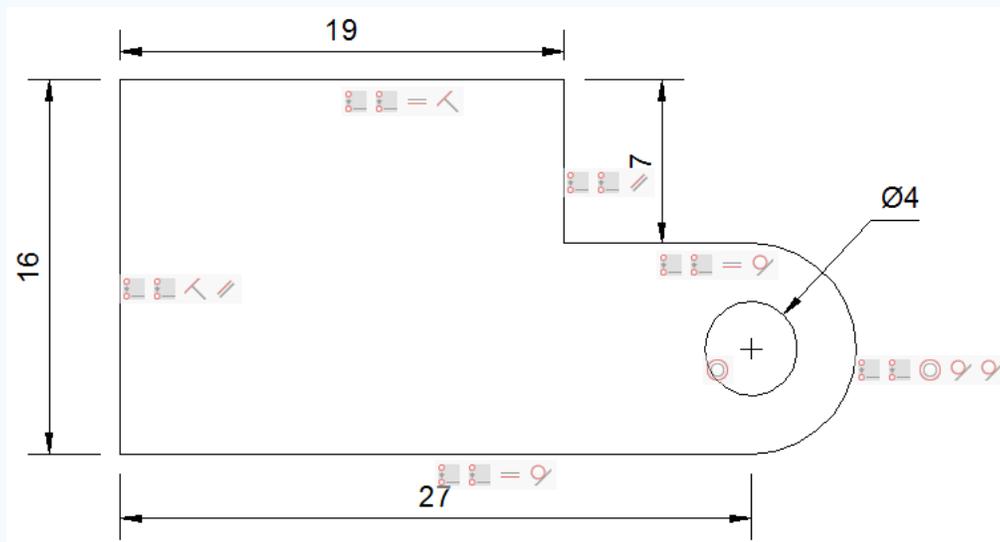
Parameter — specifies a formula that determines relationships between constraints

NEW COMMANDS

Command	Menu Bar	Ribbon
MaterialBrowserOpen	Parametric Mechanical Browser	...
ConstraintBar	Parametric 2D Constraints Bar	2D Constraints Bar
DimConstraint	Parametric 2D Constraints	2D Constraints panel
GeomConstraint	Parametric 2D Constraints	2D Constraints panel

THIS CHAPTER'S MODEL

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



Drawing file begun as bracket-ragged.dwg and ended as bracket-constrained.dwg

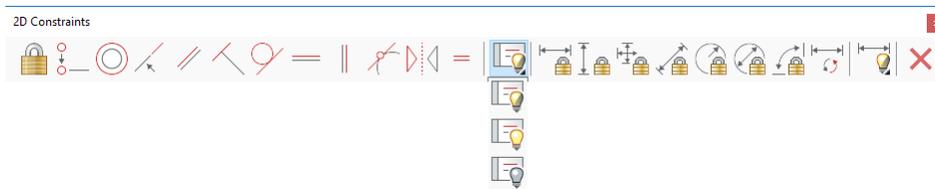
Working with Constraints

BricsCAD allows you to control the size of entities in drawings through constraints. There are two types of constraints: one specifies the size of entities, the other locates their positions.

- ▶ **Dimensional constraints** regulate the sizes of entities, and the distances between them
- ▶ **Geometric constraints** determine the position of entities relative to others

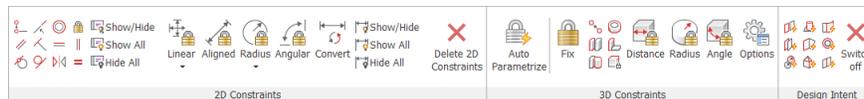
BricsCAD Classic and Pro provide 2D constraints. BricsCAD Platinum adds 3D constraints, which are applied to faces and edges of 3D bodies; they are not meant for 2D entities. (Three-dimensional constraints are not covered by this book.)

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**:



Left: Buttons geometric constraints; right: buttons for dimensional constraints

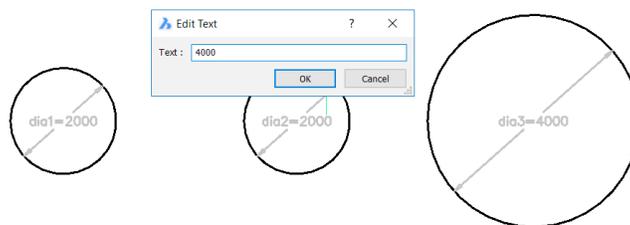
Switch to the “Modeling” workspace, and then choose the ribbon’s **Parametric** tab:



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 when 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

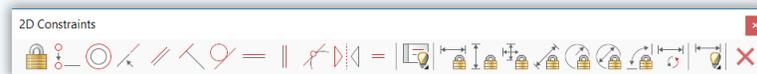
QUICK SUMMARY OF DIMENSIONAL CONSTRAINTS

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:

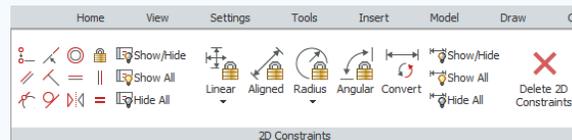
DIMCONSTRAINT COMMAND

: **dimconstraint**

Select associative dimension to convert or [LInear/Horizontal/Vertical/Aligned/ANgular/Radial/Diameter] <LInear>: (Enter an option.)



Above: 2D constraints on right half of the toolbar; **below:** 2D constraints on ribbon



DIMENSIONAL CONSTRAINT COMMANDS

The other set of commands consists of ones specific to each constraint mode, as listed in the table below. Notice that the command names all begin with 'dc', short for dimensional constraint.

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; or angle of an arc or polyline arc; or angle between three points on an entity
	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 dim constraints

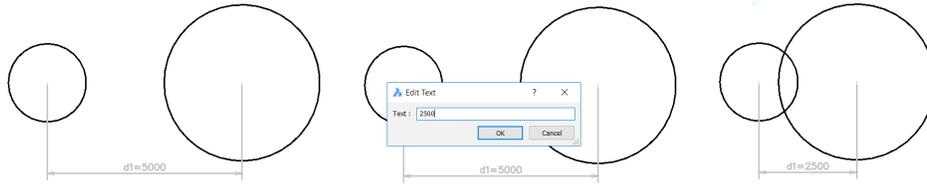
Dimensional constraints are applied in the current UCS. Icons for dimensional constraints are found at the right end of the 2D Constraints toolbar or ribbon tab:

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.

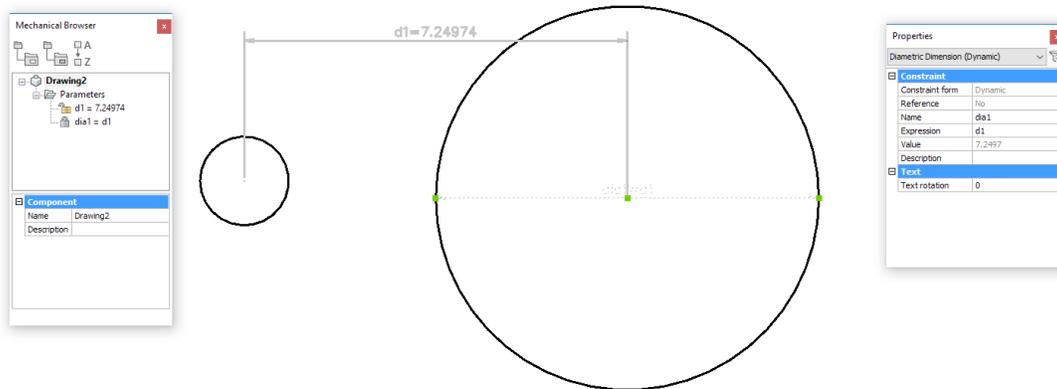
To control how far apart entities are, you place linear dimensional constraints between them. Decrease the value of the constraint, and BricsCAD moves the entities closer together; increase the value, and they are moved further apart. To edit the value, just double-click the dimension text.



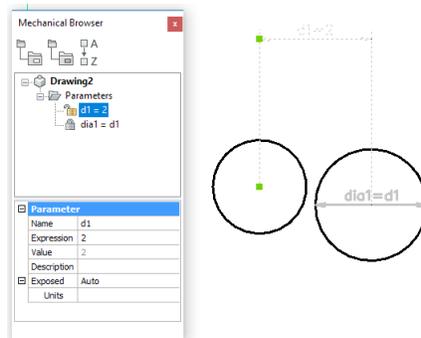
Left: Two circles separated by 5000 units; **center:** Distance being edited; **right:** Circles now separated by 2500 units

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 it and another entity. 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, $dia1=d1$.



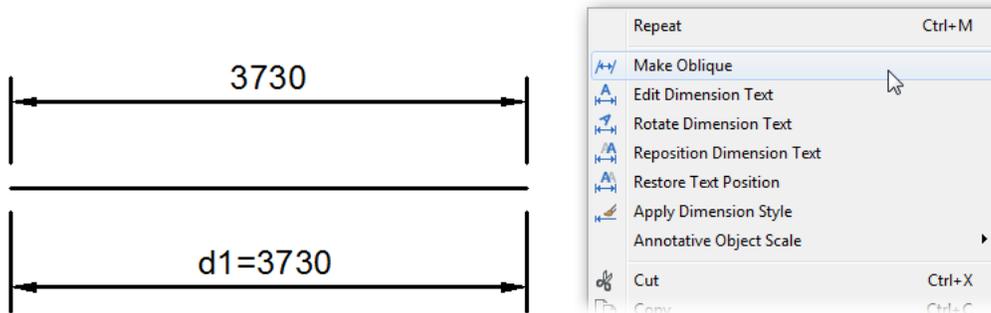
To change the values of dimensional constraints, open the Mechanical Browser with the **MechanicalBrowserOpen** command. (You can also edit values with the Properties panel.) As I change the value of *d1* (the distance between the two circles), the diameter of the big circle changes.



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 one another; you even 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.)

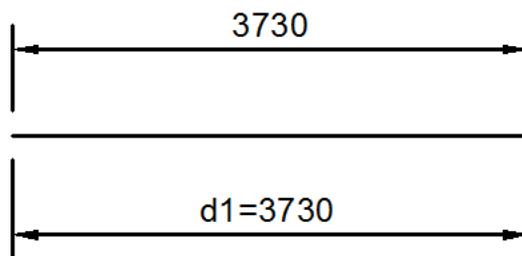
: **dclinear**

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)

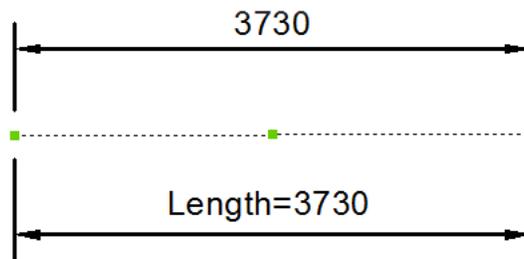
Dimension text <3730>: (Press **Enter**)



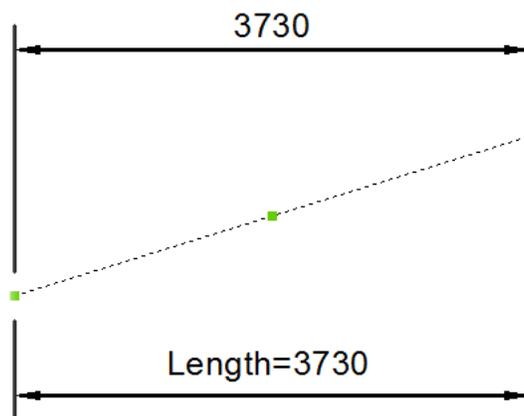
In the figure above, the two linear dimensions look identical — other than the ‘d1=’ name. The ‘d’ is short for “distance,” and the ‘1’ indicates this is the first distance constraint placed in the drawing. You can change this name through the Properties panel.

Continuing with the tutorial, you now edit the line and the dimensions to see how one affects the other.

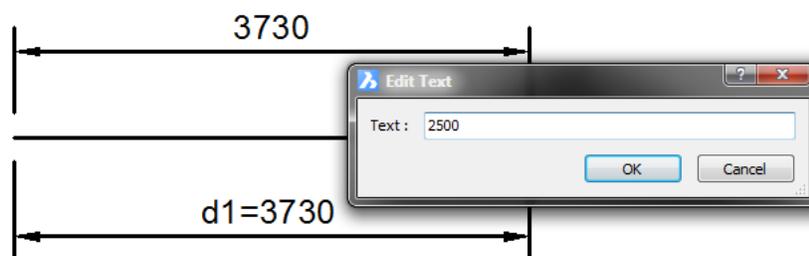
5. Edit the line by stretching one of its end grips; see figure below. Notice that you cannot. This is because the length of the line is “locked” by the dimensional constraint.



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 dcLinear.



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.



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/Coincident/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. The names of geometric constraint commands all begin with 'gc'.

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

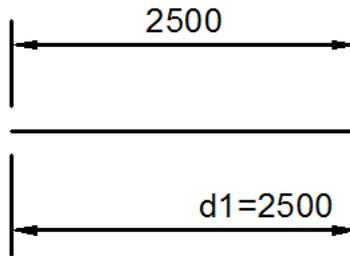
ACCEPTABLE GEOMETRY

Constraints apply to lines, segments of polylines, circles, arcs, ellipses, elliptical arcs, and splines.

Entity	Constraint Points
Arcs	Endpoints, center points, and midpoints
Arrays	Parameters
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

- 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.

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.

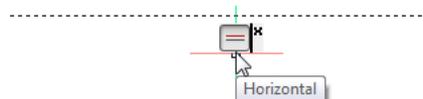


Left: Line drawn at an arbitrary angle; **right:** Line snapped flat by horizontal constraint

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).

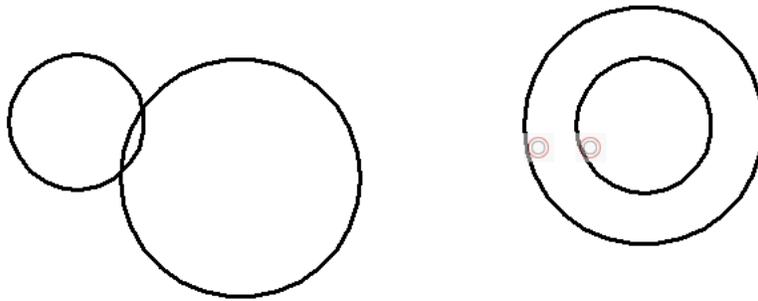
Identifying Geometric Constraints

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.)

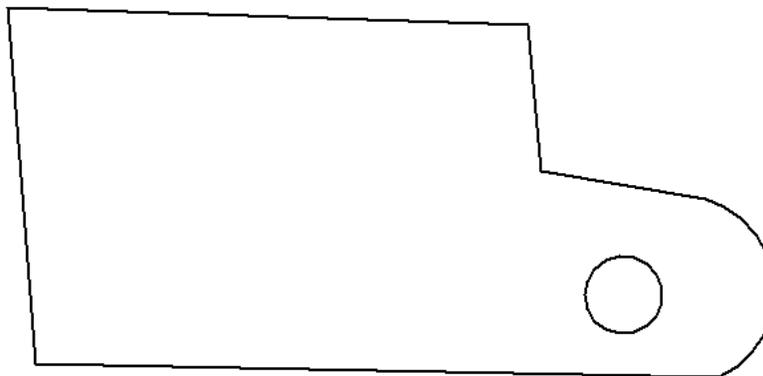
To remove a geometric constraint, you have to use the **DelConstraint** command; you cannot simply “erase” geometric constraints.

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 **BRISCAD**, 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:

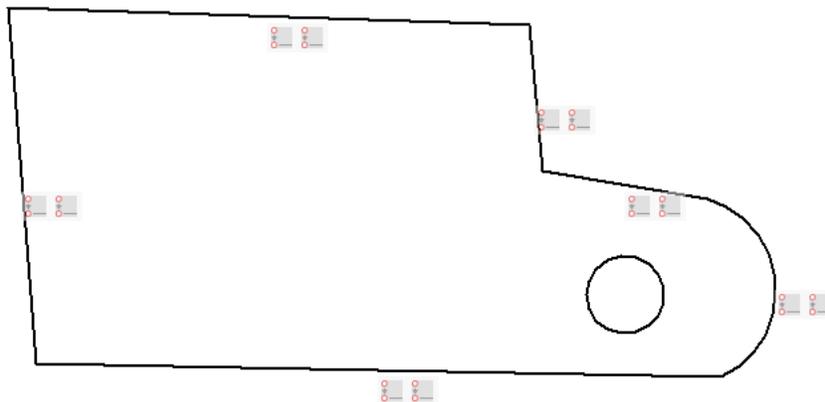
Select first point or [Entity/Autoconstrain] <Entity>: **a**

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:

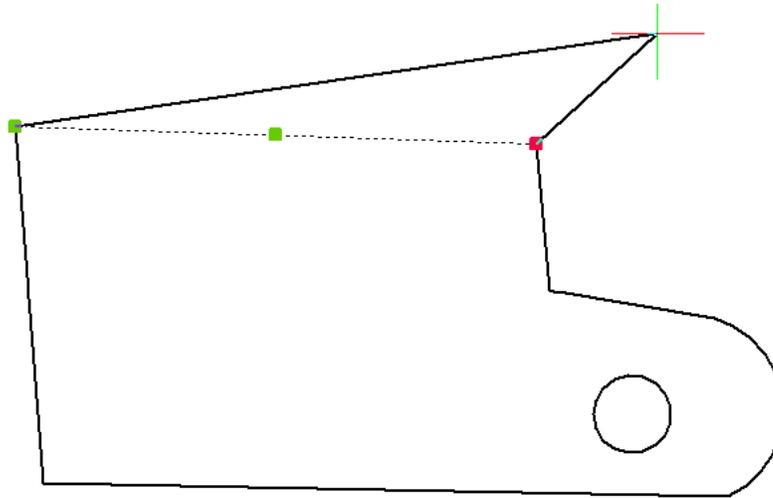
`: constraintbar`

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:

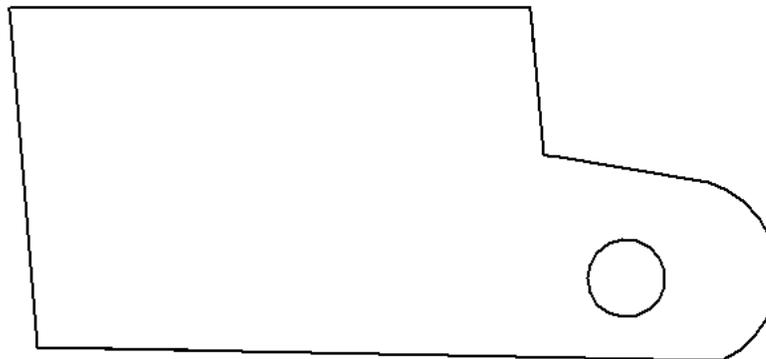
- a. In the 2D Constraints toolbar, click the **Horizontal** button, and then follow the prompts in the command bar:

```
: _gchorizontal
```

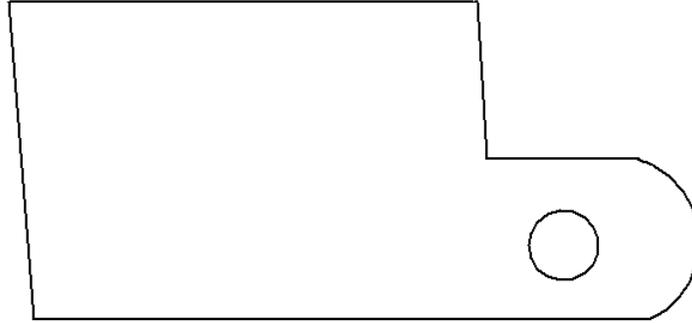
- b. Select a nearly-horizontal line:

```
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:

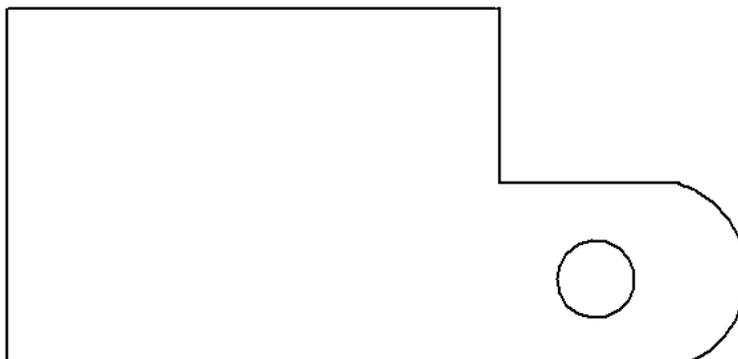
Select first entity: (Choose a horizontal line)

- 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.

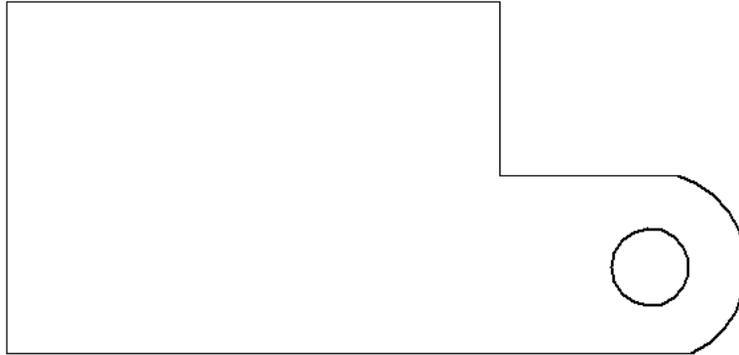


12. The circle needs a geometric constraint to fix its location. Use  **gcConcentric** to match the center point of the circle to that of the arc, as follows:

```
: _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:

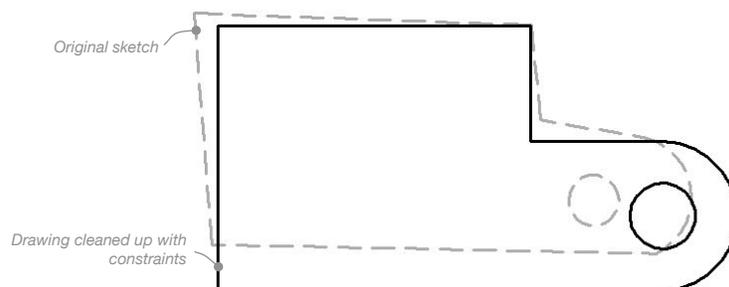
```
: _gctangent
```

```
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**)

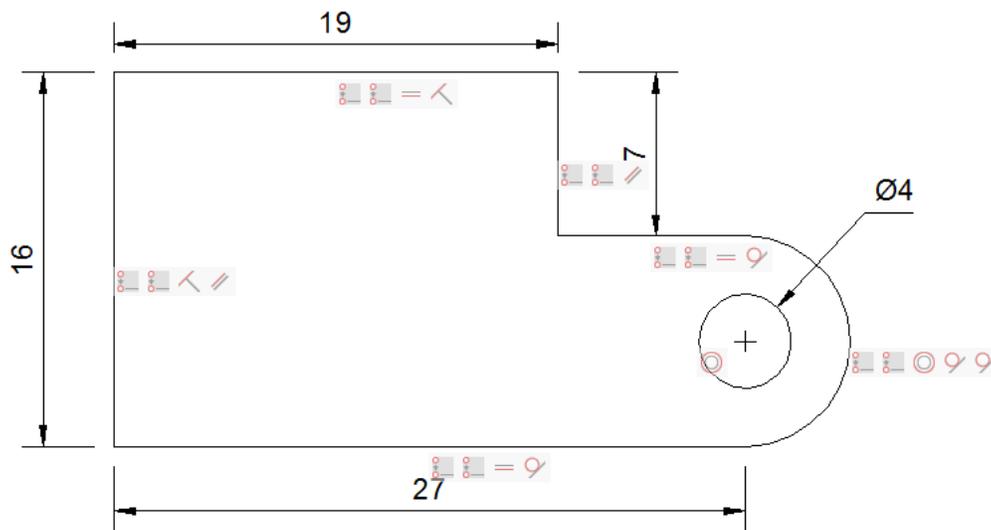
Select an entity: (Pick a line)

Specify dimension line location: (Pick a point)

Dimension text <17.993713>: **19**

- ▶ Use  **dcVertical** for vertical dimensional constraints

- ▶ Use  **dcDiameter** for diameter dimensional constraints



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 Names

IN THIS APPENDIX, YOU CAN REFERENCE THE NAMES OF OVER 900 COMMANDS IN BRICSCAD.

They are listed alphabetically by name, as well as in groupings of common commands, as follows:

ai- commands

bim- (building information modeling) commands

bm- (BricsCAD mechanical) Commands

Cloud- commands (ex-Chapoo)

Dim- (dimension) commands

dc- (dimensional constraint) commands

dm- (direct modeling) commands

gc- (geometric constraint) commands

Layer Commands

sm- (sheet metal) Commands

VBA (Visual Basic for Applications) commands

ViewBase commands

“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 new in V19 are shown in **blue**. Command names specific to the Pro and Platinum versions are shown in **boldface**; these commands are not available in the Classic version.

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.

AttachmentsPanelOpen opens the Attachments panel for managing Xref, Raster Image, PDF, and Pointcloud attachments.

AttachmentsPanelClose closes the Attachments panel.

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).

BClose closes the Block Editor

BEdit and **-BEdit** open the Block Editor environment

BHatch and **-BHatch** fills closed areas with repeating patterns, solid colors, or gradients.

Blade opens the LISP editing environment

BlipMode enables and disables display of marker blips.

Block and **-Block** groups entities into blocks (symbols).

Blockify converts entities to blocks to save space and increase speed

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 as an add-on to the Platinum edition only; *bim* = building information modeling)

bimAddEccentricity controls relative positions of the axes in linear solids.

bimApplyProfile applies profiles to linear entities and linear solids.

bimAttachComposition attaches BIM compositions to solids.

bimAttachSpatialLocation locates the drawing in mapping references.

bimCheck removed from BricsCAD V19.

bimClassify classifies an entity as a building element with a name and an internal 'guid' (globally unique identifier).

bimConnect removed from BricsCAD V19.

bimCopy copies entities *normal* (at 90 degrees) to the selected face.

bimCurtainWall creates curtain walls made of planar quadrilateral panels from free-form surfaces.

bimDisplayComposition toggles the display of compositions on and off.

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 removed from BricsCAD V19, replaced by Export command's IFC option

bimFlip flips the starting face from which the layers of a composition are set out.

bimFlowConnect connects linear solids.

bimGetStatisticalData removed from BricsCAD V19.

bimGrid creates rectangular and radial grids with automatically-applied labels.

bimIfcImport removed from BricsCAD V19, replaced by Import command's IFC option

bimIfy automatically classifies and spatial locates the entire bim model.

bimInsert and **-bimInsert** insert windows and doors in solids.

bimLinearSolid creates chains of linear solids.

bimList list names and properties of BIM entities in the current drawing.

bimMultiSelect selects one or more linear solids with coplanar and/or parallel axes based on the initial solid or face selected.

bimPatch reserves an of a BIM model for editing with the RefEdit command.

bimProfiles displays the Profiles dialog box for creating and editing profiles.

bimProjectInfo displays the BIM Project Info dialog box for specifying project library databases.

bimPropagate (replaces **bimSuggest**) maps details from selected solids to all similar solids, as well as on grids.

bimPropagateEdges propagates along the edges of planar solids, such as railings.

bimPropagateLinear propagates connections to linear elements, such as connections to walls and slabs.

bimPropagatePattern propagates a single element (such as a switch) to multiple locations and grids.

bimPropagatePlanar propagates connections to planar elements, such as walls, slabs, and roofs.

bimProperties displays the BIM Properties dialog box for specifying and editing properties of bim projects.

bimQuickDraw draws rooms and stories from rectangles and L-shapes.

bimRecalculateAxis recalculates the axes of structural elements back to their centroids.

bimReposition removed from BricsCAD V19, replaced by editable dimensions to neighboring edges.

bimRoom defines room areas with markers.

bimRoomBoundingElements determines which elements (walls, floors, etc) determine bounds of rooms.

bimSchedule generates linked schedule tables after analyze building elements in BIM models.

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.

bimSetReferenceFace controls the layout of plies through reference and opposing faces.

bimSplit splits segmented solids into separated solids automatically; splits solids using cutting faces.

bimStretch stretches BIM entities.

bimStructuralConnect connects linear solids.

bimSuggest removed from BricsCAD V19, replaced by the **bimPropagate** command.

bimTag tags BIM sections.

bimUpdateRoom updates data about the selected room.

bimUpdateThickness re-applies the overall thickness of a composition to the solid.

bimWindowArray removed from V19.

bimWindowCreate replaces closed entities with parametric window entities; displays the choose window style dialog box.

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.

BM (BricsCAD Mechanical) Commands

(Available in Platinum edition only; bm = BricsCAD mechanical)

bmBalloon associates balloon with assembly components in Model space and in generated views in layouts.

bmBom inserts bill of material (BOM) tables in the current drawing.

bmBrowser removed from V19, replaced by the **ComponentsPanelOpen** command

-bmCreateComponent creates a component from a selection set; add it to the library.

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.

bmExplode creates a block of an exploded representation of an assembly.

bmExplodeMove allows users to created exploded representations of assemblies.

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.

bmLispGet retrieve variables for blocks and parameters of components.

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.

bmUnlink breaks links between 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.

Centerline places associative center lines on circles and arcs

Centermark places associative center marks on circles and arcs

CenterReassociate reassociates centerlines/marks with circles and arcs

CenterResets centerline and mark entities, if moved

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.

CleanScreenOn hides most user interface elements to maximize the drawing area.

CleanScreenOff restores the user interface to its default configuration.

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.

CommunicatorInfo reports the status of the Communicator add-on

ComponentsPanelOpen opens the Components panel for accessing symbols

ComponentsPanelClose closes the Components panel

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).

ConvToMesh converts 3D solids and surfaces to mesh objects

ConvToSolid converts watertight meshes, circles, and closed polylines to 3D solids

ConvToSurface converts 3D solids, open polylines and other entities to 3D surfaces

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.

Cloud (24/7, ex-Chapoo) Commands

All Chapoo- commands were renamed Cloud- in V18

CloudAccount reports the status of the 24/7 account at the command bar.

CloudDownload downloads drawings from the 24/7 project to a local folder.

CloudLogoff logs off from the 24/7 project.

CloudLogon logs on to 24/7.

CloudOpen opens a drawing after downloading it from 24/7.

CloudProject opens the 24/7 project in the default browser.

CloudUpload uploads the current drawing to 24/7.

CloudWeb connects to the 24/7 website at <https://www.bricsys.com/en-intl/247/>.

D Commands

DataExtraction exports entity properties, block attributes and drawing information to CSV (comma separated values) file.

DataLink imports Excel spreadsheets and CSV files as linked table entities

DataLinkUpdate updates the data linked between a table and an external 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.

DefaultScaleList displays the Scale List Edit dialog box to edit the default scale factors

DesignTable creates new design tables for the Mechanical Browser.

-DesignTableEdit configures, replaces, exports, and deletes design tables at the command line.

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”).

DgnImport imports Microstation design files and converts them to entities

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.

DwgCompare compares differences between two drawings, and visually merges drawings.

DwgProps opens the Drawing Properties dialog box, showing the general information and user defined properties stored with a drawing.

DxfIn and **DxfOut** imports DXF files (short for “drawing exchange format”) and exports drawings in ASCII or binary DXF format.

DIM (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.

DC (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.

DM (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.

dmAuditAll also checks and fixes 3D ACIS models in externally-referenced drawings

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.

dmConcentric3D applies concentric constraints between two cylindrical, spherical, or conical surfaces.

dmConstraint3D applies geometric relationships and dimensional constraints between sub-entities (such as faces, surfaces, and edges) of 3D entities.

dmCopyFaces copies features like holes and ribs to the same or other 3D solids

dmDeformCurve deforms one or more connected faces of a 3D solid/surface by replacing their edges with given curves.

dmDeformMove deforms one or more connected faces of a 3D solid/surface by moving and rotating their edges.

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.

dmDelete deletes faces and solids.

dmDistance3D applies a distance constraint between two sub-entities of a solid or of different solids.

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.

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.

dmPerpendicular3D applies a perpendicular constraint between two faces of a solid or of different solids.

dmPushPull adds or removes volume from a solid by moving a face.

dmRadius3D applies a radius constraint to cylindrical surfaces or circular edges.

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.

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.

dmSimplifyAll also unnecessary elements in externally referenced drawings

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.

dmThicken creates 3D solids by thickening (i.e. adding thickness to) surfaces, their faces, and faces of 3D solids.

dmTwist twists 3D solids, surfaces, and regions by an angle.

dmUpdate forces 3D constraints to update.

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.

EndCompare ends the drawing compare session

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 Commands

FbxExport and **-FbxExport** export 3D models in FBX format for rendering programs

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 the 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.

Grading interactively shapes terrains

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.

GC (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.

gcCenter snaps to the centroid of closed entities.

gcCoincident constrains points on entities coincidently; 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 helixes at common endpoints.

K Command

KeepMe visually merges drawings during the DrawingCompare command

L Commands

Layer: see *Layer Commands* below.

Layout creates, copies, renames, and deletes layouts.

LayoutManager displays the Layout Manager dialog box for creating, naming, and reordering sets of layouts

LConnect creates connections between faces of two solids

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 linestyles.

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.

Manipulate launches the widget for rotating, copying, moving, mirroring, and scaling entities.

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”).

MaterialAssign assigns materials from the Material Browser onto 3D objects

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.

MechanicalBrowserClose closes the Mechanical Browser panel.

MechanicalBrowserOpen displays the Mechanical Browser panel.

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.

MLeaderAlign aligns multiple leaders

MLeaderCollect collects multiple leader blocks

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

Navigate walks and flies through 3D models.

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.

Number adds incremented number tags for BIM entities

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.

Panelize command draws freeform surfaces as subdivision meshes, optionally planarizing the panels.

Parallel turns on parallel entity snap.

-Parameters create and edit constraint expressions and values.

ParametersPanelOpen opens the Parameters panel

ParametersPanelClose closes the Parameters panel

Parameterize adds constraints and parameters to models automatically

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.

PlotStamp specifies a header and footer for plotted output.

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.

PointCloud displays the Point Cloud section of the Drawing Explorer

PointCloudAttach attaches BPT point cloud files to the current drawing

PointCloudColorMap changes the colors of point based on their elevation

PointCloudPointSize specifies the size of points in a point cloud

PointCloudPointSize_Minus decreases the size of points in a point cloud

PointCloudPointSize_Plus increases the size of points in a point cloud

PointCloudPreprocess converts ASCII PTS, PTX, and LAS files into compressed binary BPT files

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.

ReInit 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.

ReportPanelClose and **ReportPanelOpen** close and open the Report panel.

ResetAssocViews removes associative data from blocks

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.

SaveFileFolder opens the File Explorer to the folder in which the current drawing is being saved

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.

Site imports terrain models from points and Civil 3D surfaces, or creates them from entities

SiteEdit edits terrain sites

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.

StlOut 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.

StandardPartsPanelClose and **StandardPartsPanelOpen** close and open the Standard Parts panel.

StructurePanel and **StructurePanelClose** open and close the Structure panel displaying tree structure of the drawing content

+StructurePanel opens a CST structure tree configuration file

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 9x64\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.

SM (Sheet Metal) Commands

(Available for Mechanical edition; requires an additional license; sm = sheet metal)

LicPropertiesSheetmetal reports the license state of the sheet metal module.

smAssemblyExport converts 3D solid sheet metal parts to DXF files with unfolding information

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.

smExtrude extrudes polylines to sheet metal parts

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 removed from V19; replaced by **smSplit**

smFlip switches flange sides to reverse reference faces

smForm adds forms to sheet metal.

smImprint uses imprinted edges to split thickness faces of sheet metal parts

smJunctionCreate converts hard edges (sharp edges between flange faces) and bends into junctions.

smJunctionSwitch changes symmetrical junction features to overlapping faces.

smLispGet returns values related to sheet metal variables.

smLispSet changes values related to sheet metal variables.

smLoft creates sheet metal part with lofted bends and flanges from two non-coplanar curves.

smParametrize generates consistent sets of 3D constraints for sheet metal parts

smReliefCreate creates proper corner (three or more adjacent flanges) and bend reliefs (at the start and end of a flange edge).

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 removed from V19.

smRibCreate adds associative rib (form) features on sheet metal parts based on 2D profiles

smSelect selects hard edges and form features of sheet metal parts

smSelectHardEdges selects all hard edges on sheet metal parts.

smSplit splits flanges and lofted bend; replaces the old **smFlangeSplit** command

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.

TConnect connects solids by their faces

TemplateFolder opens the C:\Users\<login>\AppData\Local\Bricsys\BricsCAD\W19x64\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.

TfLoad and **TfSave** open and save handle, xsd, and strip data from DWT template files

Time reports on the time spent in the drawing.

TInsert inserts blocks in the cells of tables.

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; vba = Visual Basic for Applications)

Vbalde opens the **BLADE** 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.

XmlSave prompts for handles to save in an XML file.

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.

Concise Summary of Variables & 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 over 1,000 variable names in alphabetical order.

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 in V19

ikeThrough text indicates the variable was removed from BricsCAD

userid or *login* refers to your computer login name

When you see **Read-only** (r/o), it means that you cannot change the variable's value; the value has been set by BricsCAD or by the operating system.

Variable Name	Read-only	Default Value
---------------	-----------	---------------

A Variables

ACADLSPASDOC		0
ACADPREFIX	r/o	"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V19x64\en_US\Support\"; C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Support\"; C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Fonts\"; C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Help\en_US\"
ACADVER	r/o	"20.0 BricsCAD"
AcisHlrResolution		-1
ACISOUTVER		70
AcisSaveAsMode		0
AdaptiveGridStepSize		4.0000
AFLAGS		0
ALLOWBREAKLINECROSSINGS		"1"
ALLOWEDBENDANGLES		"1"
AllowTabExternalMove		1
AllowTabMove		1
AllowTabSplit		1
ANGBASE		0
ANGDIR		0
Anglesamplinginterval		"5"
ANNOALLVISIBLE		1
ANNOAUTOSCALE		-4
AnnoSelected	r/o	0
ANNOTATEDWG		0
AntiAliasRender		2
AntiAliasScreen		1
APBOX		0
APERTURE		10
AREA	r/o	0
AREAPREC		-1
AREAUNITS		"in ft mi µm mm cm m km"
ARRAYASSOCIATIVITY		1
ARRAYEDITSTATE	r/o	0
ARRAYTYPE		0
Associativity		"3"
ATTDIA		0
ATTMODE		1
AttractionDistance		3
ATTREQ		1
AUDITCTL		0
AuditErrorCount	r/o	0
AUNITS		0
AUPREC		0
AUTOCOMPLETEDELAY		0.3
AUTOCOMPLETEMODE		47
AutomaticConnection		"1"
AUTOMENULOAD		1
AutoResetScales		0
AutosaveChecksOnlyFirstBitDBMOD		1
AUTOSNAP		119
AutoTrackingVecColor		171
AutoUpdateRooms		"1"

Variable Name	Read-only	Default Value
AutoVpFitting		1
AXISMODE		0
AXISUNIT		X= 0 Y= 0 Z= 0

B Variables

BACKGROUNDPLOT		2
BACKZ	r/o	0
BASEFILE		"Default-mm.dwt"
BimConnectCutType		"0"
BIMOSMODE		0
BINDTYPE		0
BKGCOLOR		7
BKGCOLORPS		7
BLIPMODE		0
BLOCKEDITLOCK		0
BLOCKEDITOR		0
BlocksPath		"C:\Users\userid\Documents\"
bmAutoUpdate		1
bmForceUpdateMode		0
bmReportPanel		0
BMUPDATEMODE		0
BndLimit		1000
BoundaryColor		95
BVMODE		0

C Variables

CACHELAYOUT		1
CAMERADISPLAY		0
CAMERAHEIGHT		0
CANNOSCALE		"1:1"
CANNOSCALEVALUE	r/o	1
CDATE	r/o	20160211.15522
CECOLOR		"ByLayer"
CELTSCALE		1
CELTYPE		"ByLayer"
CELWEIGHT		-1
CenterCrossGap		"0.05x"
CenterCrossSize		"0.1x"
CenterExe		0.1200
CenterLayer		."
CenterLtscale		1.0000
CenterLtype		"Center2"
CENTERLTPEFILE		"Default.Lin"
CenterMarkExe		1
CETRANSPARECNY		"ByLayer"
CGEOCS	r/o	""
CHAMFERA		0
CHAMFERB		0
CHAMFERC		0
CHAMFERD		0
CHAMMODE		0
ChapooLog		0
ChapooLogVerbose		0

Variable Name	Read-only	Default Value
ChapooOnModified		1
ChapooServer		"https://my.chapoo.com/"
ChapooTempFolder		"C:\Users\userid\AppData\Local\Temp\Chapoo\"
ChapooUploadDependencies		1
ChapooWebsite		"http://www.chapoo.com/"
CheckDwlPresence		0
CIRCLERAD		0
CLAYER		"0"
CLEANSCREENOPTIONS		15
CLEANSCREENSTATE	r/o	0
ClipboardFormat		1
CLIPBOARDFORMATS		127
CliPromptLines		4
CLISTATE	r/o	1
CloseChecksOnlyFirstBitDBMOD		0
CloudLog		0
CloudLogVerbose		0
CloudOnModified		1
CloudServer		"https://my.bricsys247.com/"
CloudTempFolder		"C:\Users\login\AppData\Local\Temp\Bricsys_24_7\"
CloudUploadDependencies		1
CMATERIAL		"ByLayer"
CMDACTIVE	r/o	1
CMDDIA		1
CMDECHO		1
CmdLineEditBgColor		"#fefefe"
CmdLineEditFgColor		"#202020"
CmdLineFontName		"Consolas"
CmdLineFontSize		10
CmdLineListBgColor		"#ecf1ff"
CmdLineListFgColor		"#000000"
CMDLNTEXT		":"
CMDNAMES	r/o	"SETTINGS"
CMLEADERSTYLE		"Standard"
CMLJUST		0
CMLSCALE		1
CMLSTYLE		"Standard"
CMPCLRMISS		1
CMPCLRMOD1		253
CMPCLRMOD2		2
CMPCLRNEW		3
CMPIFFLIMIT		1000
COLORX		11
COLORY		112
COLORZ		150
COMAcadCompatibility		0
COMPASS		0
Componentspath		"C:\Users\login\AppData\Roaming\Bricsys\BricsCAD\V19x64\en_US\Support\Bim\Components\"
CONSTRAINTBARDISPLAY		3
ContinuousMotion		0
COORDS		1
COPYMODE		0
CPLOTSTYLE		"ByColor"

Variable Name	Read-only	Default Value
CPROFILE	r/o	"Default"
CREATEVIEWPORTS		1
CROSSINGAREACOLOR		91
CTAB		"Model"
CTABLESTYLE		"Standard"
Ctrl3DMouse		1
CTRLMOUSE		1
CURSORSIZE		3
CVPORT		2

D Variables

DATE	r/o	2456335.6613464
DBCSTATE	r/o	0
DBLCLKEDIT		1
DBMOD	r/o	0
DCTCUST		""
DCTMAIN		"en_US.dic"
ddBetweenKnots		2
ddFastMode		0
ddGridAspectRatio		0
ddMaxFacetEdgeLength		0
ddMaxNumGridLines		10000
ddNormalTol		15
ddPointsPerEdge		0
ddSurfaceTol		0
ddUseFacetRES		1
DEFAULTLIGHTING		0
DefaultLightShadowBlur		8
DefaultNewSheetTemplate		""
DefaultRoomHeight		"120"
DEFLPLSTYLE		"Normal"
DEFPLSTYLE		"ByColor"
DeleteTool		1
DELOBJ		1
DEMANDLOAD		3
DGNFRAME		2
DgnImp2dClosedBSplineCurveImportMode		0
DgnImp2dEllipseImportMode		0
DgnImp2dShapeImportMode		0
DgnImp3dClosedBSplineCurveImportMode		1
DgnImp3dEllipseImportMode		0
DgnImp3dObjectImportMode		0
DgnImp3dShapeImportMode		1
DgnImpBreakDimensionAssociation		0
DgnImpConvertDgnColorIndicesToTrueColors		0
DgnImpConvertEmptyDataFieldsToSpaces		1
DgnImpEraseUnusedResources		0
DgnImpExplodeTextNodes		0
DgnImpImportActiveModelToModelSpace		1
DgnImpImportInvisibleElements		1
DgnImpImportPaperSpaceModels		1
DgnImpImportViewIndex		-1
DgnImpRecomputeDimensionsAfterImport		0
DgnImpSymbolResourceFiles		""

Variable Name	Read-only	Default Value
DgnImpXRefImportMode		2
DGNOSNAP		1
DIASTAT	r/o	0
Displayaxes		"0"
Displayaxesformep		"0"
Displaysidesandends		"1"
DisplaySnapMarkerInAllViews		0
DisplayTooltips		1
DISPPAPERBKG		1
DISPPAPERMARGINS		1
DISPSILH		0
DISTANCE	r/o	0
DMAUTOUPDATE		1
DmExtrudeMode		0
DmPushPullSubtract		1
DMRECOGNIZE		0
DockPriority		1
DocTabPosition		0
DONUTID		0.5
DONUTOD		1
DRAGMODE		2
DragModeHide		0
DRAGMODEINTERRUPT		1
DRAGOPEN		1
DRAGP1		10
DRAGP2		25
DRAGSNAP		0
DrawingPath		"C:\Users\userid\Documents\"
DrawingViewPreset		"none"
DrawingViewPresetHidden		0
DrawingViewPresetScale		""
DrawingViewPresetTangent		0
DRAWORDERCTL		3
DWFFRAME		2
DWFOSNAP		1
DwfVersion		2
DWGCHECK		0
DWGCODEPAGE	r/o	"ANSI_1252"
DWGNAME	r/o	"Drawing1.dwg"
DWGPREFIX	r/o	"C:\Program Files (x86)\Bricsys\BricsCAD V19x64\"
DWGTITLED	r/o	0
DXEVAL		12
DxfTextAdjustAlignment		0
DYNCONSTRAINTMODE		1
DYNDIGRIP		31
DynDimColorHot		142
DynDimColorHover		142
DynDimDistance		1
DynDimLineType		1
DYNDIVIS		1
DynInputTransparency		65
DYNMODE		3

Dimensions Variables

DIMADEC		0
DIMALT		0
DIMALTD		2
DIMALTF		25.4
DIMALTRND		0
DIMALTTD		2
DIMALTTZ		0
DIMALTU		2
DIMALTZ		0
DIMANNO	r/o	0
DIMAPOST		""
DIMARCSYM		0
DIMASO		1
DIMASSOC		2
DIMASZ		0.18
DIMATFIT		3
DIMAUNIT		0
DIMAZIN		0
DIMBLK		""
DIMBLK1		""
DIMBLK2		""
DIMCEN		0.09
DIMCLRD		0
DIMCLRE		0
DIMCLRT		0
DIMDEC		4
DIMDLE		0
DIMDLI		0.38
DIMDSEP		"0"
DIMEXE		0.18
DIMEXO		0.0625
DIMFIT		3
DIMFRAC		0
DIMFXL		1
DIMFXLON		0
DIMGAP		0.09
DIMJOGANG		0.7853981634
DIMJUST		0
Dimlayer		":"
DIMLDRBLK		""
DIMLFAC		1
DIMLIM		0
DIMLTEX1		""
DIMLTEX2		""
DIMLTYPE		""
DIMLUNIT		2
DIMLWD		-2
DIMLWE		-2
DIMPOST		""
DIMRND		0
DIMSAH		0
DIMSCALE		1

Variable Name	Read-only	Default Value
DIMSD1		0
DIMSD2		0
DIMSE1		0
DIMSE2		0
DIMSHO		1
DIMSOXD		0
DIMSTYLE	r/o	"Standard"
DIMTAD		0
DIMTDEC		4
DIMTFAC		1
DIMTFILL		0
DIMTFILLCLR		"BYBLOCK"
DIMTIH		1
DIMTIX		0
DIMTM		0
DIMTMOVE		0
DIMTOFL		0
DIMTOH		1
DIMTOL		0
DIMTOLJ		1
DIMTP		0
DIMTSZ		0
DIMTVP		0
DIMTXSTY		"Standard"
DIMTXT		0.18
DIMTXTDIRECTION		0
DIMTZIN		0
DIMUNIT		2
DIMUPT		0
DIMZIN		0

E Variables

EDGEMODE		0
ELEVATION		0
ElevationAtBreaklineCrossings		"0"
EnableAttraction		1
EnableHyperlinkMenu		1
EnableHyperlinkTooltip		0
ERRNO		0
EXPERT		0
ExplnsAlign		0
ExplnsAngle		0
ExplnsFixAngle		1
ExplnsFixScale		1
ExplnsScale		1
EXPLMODE		1
ExportHiddenParts		"0"
EXPORTMODELSPACE		0
EXPORTPAGESETUP		0
EXPORTPAPERSPACE		0
ExportProductStructure		"1"
ExportStepFormatVersion		"1"
EXTMAX	r/o	-1.0000E+20,-1.0000E+20,-1.0000E+20
EXTMIN	r/o	1.0000E+20,1.0000E+20,1.0000E+20

Variable Name	Read-only	Default Value
EXTNAMES		1
F Variables		
FACETRATIO		0
FACETRES		0.5
FEATURECOLORS		1
FIELDDISPLAY		1
FIELDDEVAL		31
FILEDIA		1
FILLETRAD		0.5
FILLMODE		1
FittingRadiusType		"0"
FittingRadiusValue		"1.5"
FLATLAND		Off
FONTALT		"simplex.shx"
FONTMAP		"default.fmp"
FRAME		3
FRAMESELECTION		0
FRONTZ	r/o	0
FULLOPEN	r/o	1
G Variables		
GDIOBJECTS	r/o	3768
GearteethNumber		"1"
GENERATEASSOCVIEWS		0
GEOLATLONGFORMAT		1
GEOARKERVISIBILITY		1
GetStarted		1
GfAng		0.0000
GfClr1		"5"
GfClr2		"7"
GfClrLum		1.0000
GfClrState		0
GfName		1
GfShift		0
GLSWAPMODE		2
GradientColorBottom		"#d2d2d2"
GradientColorMiddle		"#fafafa"
GradientColorTop		"#ffffff"
GradientMode		"0"
GRIDAXISCOLOR		252
GRIDDISPLAY		3
GRIDMAJOR		5
GRIDMAJORCOLOR		253
GRIDMINORCOLOR		254
GRIDMODE		0
GRIDSTYLE		0
GRIDUNIT		1/2", 1/2"
GRIDXYZT		1
GRIPBLOCK		0
GRIPCOLOR		72
GRIPDYNCOLOR		140
GRIPHOT		240

Variable Name	Read-only	Default Value
GRIPHOVER		150
GRIPOBJLIMIT		100
GRIPS		1
GRIPSIZE		4
GRIPTIPS		1
GsDeviceType		0
GsDeviceType2D		0
GsDeviceType3D		1

H Variables

HALOGAP		0
HANDLES	r/o	1
HANDSEED		"64"
HIDEPRECISION		0
HIDETEXT		1
HIDEXREFSCALES		1
HIGHLIGHT		1
HIGHLIGHTCOLOR		142
HIGHLIGHTEFFECT		0
HomeGradientColorBottom		210,210,210
HomeGradientColorMiddle		250,250,250
HomeGradientColorTop		White
HomeGradientMode		"0"
HorizonBkg_Enable		1
HorizonBkg_GroundHorizon		"#878787"
HorizonBkg_GroundOrigin		"#5F5F5F"
HorizonBkg_SkyHigh		"#239BFF"
HorizonBkg_SkyHorizon		"#FFFFFF"
HorizonBkg_SkyLow		"#FAFAFF"
HotkeyAssistant		1
HPANG		0
HPANNOTATIVE		0
HPASSOC		1
HpBackgroundColor		": "
HpColor		": "
HPBOUND		1
HPBOUNDRETAIN		0
HPDOUBLE		0
HPDRAWORDER		3
HPGAPTOL		0
HPLAYER		": "
HPLINETYPE		0
HplslandDetection		0
HPMAXAREAS		0
HPNAME		""
HPOBJWARNING		10000
HPORIGIN		0",0"
HPSCALE		1
HPSEPARATE		0
HPSPACE		1
HPSTYLE	replaced by HplslandDetection	
HPTRANSPARENCY		": "
HYPERLINKBASE		""

Variable Name	Read-only	Default Value
I Variables		
IfcExplodeExternalReferences		"0"
IfcExportBaseQuantities		"0"
IfcExportElementsOnOffAndFrozenLayer		"1"
IfcExportMultiplyElementsAsAggregated		"0"
IfcImportBimData		"1"
IfcImportParametricComponents		"0"
IfcImportSpaces		"0"
ImageCacheFolder		"C:\Users\userid\AppData\Local\Temp\ImageCache\"
ImageCacheMaxMemory		160
ImageDiskCache		1
IMAGEFRAME		1
IMAGEHLT		0
ImageNotify		0
ImportColors		"1"
ImportCreoAlternateSearchPaths		""
ImportCuiFileExists		0
ImportHiddenparts		"0"
ImportIfcProjectStructureAsXrefs		"0"
ImportIgesSimplify		"1"
ImportIgesStitch		"1"
ImportInventorAlternateSearchPaths		""
ImportNxAlternateSearchPaths		""
ImportPmi		"1"
ImportProductStructure		"2"
ImportRepair		"0"
ImportSimplify		"0"
ImportSolidedgeAlternateSearchPaths		""
ImportSolidworksAlternateSearchPaths		""
ImportSolidworksRotateYz		"1"
ImportStepRotateYz		"0"
ImportStitch		"0"
IncludePlotStamp		1
INDEXCTL		0
INETLOCATION		"http://www.bricsys.com"
INSBASE		0",0",0"
INSNAME		""
INSUNITS		1
INSUNITSDEFSOURCE		0
INSUNITSDEFTARGET		0
InsUnitsScaling		1
INTERFERECOLOR		"BYLAYER"
InterfereLayer		"Interference"
INTERFEREOBJS		""
INTERFEREVPVS		""
InteriorElevationMinLength		"20"
InteriorElevationOffset		"2"
INTERSECTIONCOLOR		257
INTERSECTIONDISPLAY		0
ISAVEBAK		1
ISAVEPERCENT		50
ISOLINES		4

Variable Name	Read-only	Default Value
L Variables		
LASTANGLE		0
LASTPO		0",0",0"
LASTPROMPT	r/o	": SETTINGS"
LATITUDE		37.795
LayerFilterExcess		250
LAYERPMODE		1
LAYLOCKFADECTL		50
LAYOUTREGENCTL		2
LengthSamplingInterval		"40"
LENGTHUNITS		""
LENSLENGTH		50
LicExpDays		31
LICFLAGS		7
LICKEY	r/o	"7897-9999-0000-99999-0000"
LightGlyphColor		30
LIGHTGLYPHDISPLAY		1
LIGHTINGUNITS		0
LightWebGlyphColor		1
LIMCHECK		0
LIMMAX		1',9"
LIMMIN		0",0"
LINEARBRIGHTNESS		0
LINEARCONTRAST		0
LISPINIT		1
LOCALE		"en_US"
LocalRootFolder		"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V19x64\en_US\"
LOCALROOTPREFIX	r/o	"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V19x64\en_US\"
LOCKUI		0
LOFTANG1		1.5707963268
LOFTANG2		1.5707963268
LOFTMAG1		0
LOFTMAG2		0
LOFTNORMALS		1
LOFTPARAM		7
LOGFILEMODE		0
LOGFILENAME	r/o	""
LOGFILEPATH	r/o	"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V19x64\en_US\"
LOGINNAME	r/o	"userid"
LONGITUDE		-122.394
LookFromDirectionMode		1
LookFromFeedback		1
LookFromZoomExtents		1
LTSCALE		1
LUNITS		4
LUPREC		4
LWDEFAULT		25
LWDISPLAY		0
LWDISPSCALE		0.55
LWUNITS		1

Variable Name	Read-only	Default Value
---------------	-----------	---------------

M Variables

MACROREC		0
MACROTRACE		0
MANIPULATOR		0
MANIPULATORDURATION		250
ManipulatorSize		1
MassPropAccuracy		0.01
MASSUNITS		"oz lbs stone mg g kg tonne"
MAXACTVP		64
MAXHATCH		100000
MAXSORT		200
MAXTHREADS		0
MBSTATE	r/o	0
MBUTTONPAN		1
MEASUREINIT		0
MEASUREMENT		0
MENUBAR		1
MENUCTL		1
MENUCHO		0
MENUNAME	r/o	"C:\Users\userid\AppData\Roaming\Bricsys\Bricscad\V19x64\en_US\Support\default.cui"
MESHTYPE		1
MiddleClickClose		1
MILLISECS	r/o	436750804
MIRRTEXT		1
MLEADERSCALE		1
MODEMACRO		""
MSLTSCALE		1
MSOLESCALE		1
MTEXTCOLUMN		0
MTEXTDETECTSPACE		1
MTEXTED		""
MTEXTFIXED		2
MTEXTFLAGS		0
MultiSelectAngularTolerance		"3"
MyDocumentsFolder		"C:\Users\login\Documents\"
MYDOCUMENTSPREFIX	r/o	"C:\Users\login\Documents\"

N Variables

NAVVCUBEDISPLAY		3
NAVVCUBELOCATION		0
NAVVCUBEOPACITY		50
NAVVCUBEORIENT		1
NavVCubeSize		4
NFILELIST		10
NOMUTT		0
NORTHDIRECTION		0

O Variables

OBJECTISOLATIONMODE		0
OBSCUREDOLOR		257
OBSCUREDTYPE		0

Variable Name	Read-only	Default Value
OFFSETDIST		-1
OFFSETERASE		0
OFFSETGAPTYPE		0
OLEFRAME		2
OLEHIDE		0
OLEQUALITY		0
OLESTARTUP		0
OPMSTATE	r/o	1
OrbitAutoTarget		1
ORTHOMODE		0
OSMODE		4133
OSNAPCOORD		2
OSNAPZ		0
OSOPTIONS		1

P Variables

PanBuffer		1
PanelButtonSize	r/o	1
PAPERUPDATE		0
PARAMETERCOPYMODE		1
PdfCache		2
PdfEmbeddedTtf		1
PdfExportSolidHatchType		2
PDFFRAME		1
PdfHatchToBmpDpi		300
PdfImageAntiAlias		1
PdfImageCompression		1
PdfImageDPI		300
PdfImportApplyLineweight		1
PdfImportAsBlock		0
PdfImportConvertSolidsToHatches		0
PdfImportJoinLineAndArcSegments		1
PdfImportLayersUseType		0
PdfImportRasterImages		0
PdfImportSolidFills		1
PdfImportTrueTypeText		1
PdfImportTrueTypeTextAsGeometry		0
PdfImportVectorGeometry		1
PdfLayersSetting		1
PdfLayoutsToExport		0
PdfMergeControl		0
PdfNotify		0
PDFOSNAP		1
PdfPaperHeight		297
PdfPaperSizeOverride		0
PdfPaperWidth		210
PdfPRCCompression		0
PdfPRCExport		0
PdfPRCSingleViewMode		1
PdfRenderDPI		300
PdfShxTextAsGeometry		0
PdfSimpleGeomOptimization		1
PdfTextIsSearchable		0
PdfTtfTextAsGeometry		0

Variable Name	Read-only	Default Value
PdfUsePlotStyles		1
PdfZoomToExtentsMode		1
PDMODE		0
PDSIZE		0
PEDITACCEPT		0
PELLIPSE		0
PERIMETER		0
PERSPECTIVE		0
PFACEVMAX		4
PICKADD		1
PICKAUTO		3
PICKBOX		4
PICKDRAG		0
PICKFIRST		1
PICKSTYLE		1
PictureExportScale		1
_PKSER	r/o	""
PlacesBarFolder1		0
PlacesBarFolder2		1
PlacesBarFolder3		3
PlacesBarFolder4		5
PLATFORM	r/o	"Microsoft Windows NT Version 6.2"
PLINECACHE		0
PLINECONVERTMODE		0
PLINEGEN		0
PLINETYPE		2
PLINEWID		0
PlotCfgPath		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V19x64\en_US\PlotConfig\"
PLOTID		""
PlotOutputPath		""
PLOTROTMODE		2
PlotStylePath		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V19x64\en_US\PlotStyles\"
PLOTTER		0
PLOTTRANSPARENCYOVERRIDE		1
PLQUIET		0
POLARADDANG		""
POLARANG		90
POLARDIST		0
POLARMODE		0
POLYSIDES		4
POPUPS	r/o	1
PreviewDelay		30
PREVIEWEFFECT		2
PREVIEWFILTER		5
PreviewTopdown		0
PREVIEWTYPE		0
PreviewWndInOpenDlg		1
PrintFile		""
PRODUCT	r/o	"Bricscad"
PROGBAR		1
PROGRAM	r/o	"BRICSCAD"
PROJECTIONTYPE		0

Variable Name	Read-only	Default Value
PROJECTNAME		""
ProjectSearchPaths		""
PROJMODE		1
PROMPTMENU		3
PromptMenuFlags		1
PromptOptionFormat		2
PromptOptionTranslateKeywords		1
PropertyPreview		1
PropertyPreviewDelay		500
PropertyPreviewObjLimit		500
PropPrevTimeout		1
PROPUNITS		103
PropUnitsVersion		1
PROXYGRAPHICS		1
PROXYNOTICE		1
PROXYSHOW		1
PROXYWEBSEARCH		1
PSLTSCALE		1
PSOLHEIGHT		4
PSOLWIDTH		0.25
PSTYLEMODE	r/o	1
PSTYLEPOLICY		1
PSVPSCALE		0
PUBLISHALLSHEETS		1
PUCSBASE		""

Q Variables

QAFLAGS		0
QTEXTMODE		0
QuadAperture		20
QuadCommandLaunch		1
QuadCommandSort		0
QuadDisplay		0
QuadExpandDelay		110
QuadExpandTabDelay		50
QuadExpandGroup		0
QuadGoTransparent		0
QuadHideDelay		1000
QuadHideMargin		40
QuadIconSize		32
QuadIconSpace		1
QuadMostRecentItems		2
QuadPopupCorner		1
QuadShowDelay		150
_QuadTabFlags		12
QuadToolipDelay		1200
QuadWarpPointer		4
QuadWidth		5

R Variables

R1ZSaveAccuracy		8
R1ZSaveDeviation		0

Variable Name	Read-only	Default Value
RASTERPREVIEW		1
RE_INIT	r/o	0
RealTimeSpeedUp		5
REALWORLDSCALE		1
RecentPath		"C:\Users\userid\Documents\"
RedHiliteFull_Edge_Alpha		100
RedHiliteFull_Edge_Color		"#007AFF"
RedHiliteFull_Edge_ShowHidden		0
RedHiliteFull_Edge_Smoothing		1
RedHiliteFull_Edge_Thickness		2
RedHiliteFull_Face_Alpha		10
RedHiliteFull_Face_Color		"#007AFF"
RedHilitePartial_SelectedEdgeGlow_Alpha		75
RedHilitePartial_SelectedEdgeGlow_Color		"#FFFFFF"
RedHilitePartial_SelectedEdgeGlow_Smoothing		1
RedHilitePartial_SelectedEdgeGlow_Thickness		3
RedHilitePartial_SelectedEdge_Alpha		100
RedHilitePartial_SelectedEdge_Color		"#007AFF"
RedHilitePartial_SelectedEdge_ShowGlow		1
RedHilitePartial_SelectedEdge_Smoothing		1
RedHilitePartial_SelectedEdge_Thickness		2
RedHilitePartial_SelectedFace_Alpha		10
RedHilitePartial_SelectedFace_Color		"#007AFF"
RedHilitePartial_UnselectedEdge_Alpha		20
RedHilitePartial_UnselectedEdge_Color		"#007AFF"
RedHilitePartial_UnselectedEdge_ShowHidden		1
RedHilitePartial_UnselectedEdge_Smoothing		1
RedHilitePartial_UnselectedEdge_Thickness		1
RedHilite_HiddenEdge_Alpha		50
RedHilite_HiddenEdge_Color		"#FFFFFF"
RedHilite_HiddenEdge_Smoothing		1
RedHilite_HiddenEdge_Thickness		1
RedSdkLineSmoothing		0
ReduceLengthType		"0"
ReduceLengthValue		"0.5"
RefeditLockNotInWorkset		0
REFEDITNAME	r/o	""
REGENMODE		1
REMEMBERFOLDERS		1
RenderMaterialPath		"C:\ProgramData\..."
RenderMaterialStaticPath		"C:\Program Files\..."
RenderUsingHardware		1
ReportPanel		0
ReportPanelMode		0
RevCloudArcStyle		0
REVCLLOUDCREATEMODE		1
REVCLLOUDGRIPS		1
RevCloudMaxArcLength		0.375
RevCloudMinArcLength		0.375
RIBBONDOCKEDHEIGHT		120
RIBBONSTATE	r/o	0
RoamableRootFolder	r/o	"c:\users..."
ROAMABLEROOTPREFIX	r/o	"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\19x64\en_US\"
ROLLOVEROPACITY		100

Variable Name	Read-only	Default Value
ROLLOVERTIPS		1
RolloverSelectionSet		1
RTDISPLAY		1
RTRotationSpeedFactor		1
RTWalkSpeedFactor		<i>replaced by the StepSize variable</i>
RunAsLevel		2

S Variables

SaveChangeToLayout		1
SAVEFIDELITY		1
SAVEFILE	r/o	""
SAVEFILEPATH		"C:\Users\userid\AppData\Local\Temp\"
SaveFormat		1
SaveLayerSnapshot		1
SAVENAME	r/o	""
SaveOnDocSwitch		0
SAVEROUNDTRIP		1
SAVETIME		60
SCREENBOXES	r/o	26
SCREENMODE	r/o	1
SCREENSIZE	r/o	145'-8",73'-3"
SCR LHIST		256
SDI		0
SectionScale		"0.02"
SectionSheetsetTemplateImperial		""
SectionSheetsetTemplateMetric		""
SELECTIONANNODISPLAY		1
SELECTIONAREA		1
SELECTIONAREAOPACITY		25
SelectionMode		0
SELECTIONPREVIEW		3
SELECTSIMILARMODE		130
SHADEDGE		3
SHADEDIF		70
SheetNumberLeadingZeroes		1
SheetSetAutoBackup		1
SheetSetTemplatePath		"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V19x64\en_US\Templates\Sheet Sets\"
SHORTCUTMENU		18
SHORTCUTMENUDURATION		250
ShowDocTabs		1
ShowFullPathInTitle		0
SHOWLAYERUSAGE		0
ShowScrollButtons		1
ShowTabCloseButton		0
ShowTabCloseButtonActive		0
ShowTabCloseButtonAll		1
ShowTabControls		1
ShowWindowListButton		1
SHPNAME		""
SingletonMode		0
SKETCHINC		0.1
SKPOLY		0
SkpStitch		1

Variable Name	Read-only	Default Value
SKYSTATUS		0
SmAttributesLayerColor		"7"
SmAttributesLayerTextHeight		"0.01"
SmAttributesLayerTextHeightType		"0"
SmBendAnnotationsLayerColor		"5"
SmBendAnnotationsLayerTextHeight		"0.01"
SmBendAnnotationsLayerTextHeightType		"0"
SmBendlinesDownlayerColor		"1"
SmBendlinesDownlayerLinetype		"Continuous"
SmBendlinesDownlayerLineweight		"-3"
SmBendlinesUplayerColor		"1"
SmBendlinesUplayerLinetype		"Continuous"
SmBendlinesUplayerLineweight		"-3"
SMCOLORBEND		"#FFDC50"
SMCOLORBENDRELIEF		"#64D296"
SMCOLORCORNERRELIEF		"#64D296"
SMCOLORFLANGE		"#90A4AE"
Smcolorflangereferenceside		"#68a4ae"
Smcolorform		"#8791e1"
SMCOLORJUNCTION		"#FF6E40"
SMCOLORLOFTEDBEND		"#A0DCFA"
smcolormiter		"#af46d8"
smcolorwrongbend		"#ff3300"
smcontourslayercolor		"7"
smcontourslayerlinetype		"continuous"
smcontourslayerlineweight		"30"
smconvertpreferformfeatures		"0"
smconvertrecognizeholes		"0"
smconvertrecognizeribcontrolcurves		"0"
smdefaultbendlineextenttype		"0"
smdefaultbendlineextentvalue		"0.25"
smdefaultbendradiustype		"2"
smdefaultbendradiusvalue		"1"
smdefaultbendreliefwidthtype		"0"
smdefaultbendreliefwidthvalue		"0.5"
smdefaultcornerreliefdiametervalue		"-1"
smdefaultflangesplitextensiontype		"0"
smdefaultflangesplitextensionvalue		"0.1"
smdefaultflangesplitgaptype		"0"
smdefaultflangesplitgapvalue		"0.1"
smdefaultformfeatureunfoldmode		"4"
smdefaultjunctionalignmentto relief		"0"
smdefaultjunctiongaptype		"0"
smdefaultjunctiongapvalue		"0.001"
smdefaultkfactor		"0.27324"
smdefaultreliefextensiontype		"0"
smdefaultreliefextensionvalue		"0.1"
smdefaulttribfilletradiustype		"0"
smdefaulttribfilletradiusvalue		"5"
smdefaulttribprofileradiustype		"0"
smdefaulttribprofileradiusvalue		"2"
smdefaulttribroundradiustype		"0"
smdefaulttribroundradiusvalue		"1"
smdefaultsharpbendradiuslimitratio		"5"

Variable Name	Read-only	Default Value
smdefaultthickness		"0.078740157480315"
smexportosmapproximationaccuracy		"0.000393701"
smexportosmminimaledglength		"0.001968505"
smformfeaturesdowncolor		"6"
smformfeaturesdownlayerlinetype		"continuous"
smformfeaturesdownlayerlineweight		"-3"
smformfeaturesupcolor		"6"
smformfeaturesuplayerlinetype		"continuous"
smformfeaturesuplayerlineweight		"-3"
smjunctioncreatehealcoincident		"0"
smoverallannotationslayercolor		"3"
smoverallannotationslayerlinetype		"continuous"
smoverallannotationslayerlineweight		"-3"
smparametrizeholesparametrization		"3"
smrepairloftedblendmerge		"0"
smsmartfeatures		"3"
smsplitconvertbendtojunction		"1"
smsplithealcoincident		"0"
smsplitorthogonalbendsplit		"0"
SMTARGETCAM		""
SNAPANG		0
SNAPBASE		0",0"
SNAPISOPAIR		0
SnapMarkerColor		20
SnapMarkerSize		6
SnapMarkerThickness		2
SNAPMODE		0
SNAPSTYL		0
SNAPTYPE		0
SNAPUNIT		1/2",1/2"
SOLIDCHECK		1
SORTENTS		127
spaAdjustMode		0
spaGridAspectRatio		0
spaGridMode		1
spaMaxFacetEdgeLength		0
spaMaxNumGridLines		512
spaMinUGridLines		0
spaMinVGridLines		0
spaNormalTol		15
spaSurfaceTol		-1
spaTriangMode		1
spaUseFacetRES		1
SPLFRAME		0
SPLINESEGS		8
SPLINETYPE		6
SRCHPATH		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V19x64\en_US\Support\; C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Support\; C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Fonts\; C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Help\en_US\ "
SSFOUND		""
SSLOCATE		1
SSMAUTOOPEN		1

Variable Name	Read-only	Default Value
SSMPOLLTIME		15
SSMSHEETSTATUS		2
SSMSTATE		0
StackPanelType		r/o 0
StampFontSize		0.2
StampFontStyle		"Arial"
StampFooter		""
StampHeader		""
StampUnits		0
STARTUP		1
STEPSIZE		6
StlPositiveQuadrant		1
STEPSPERSEC		2
StructureTreeConfig		"mechanical.cst"
SURFTAB1		6
SURFTAB2		6
SURFTYPE		6
SURFU		6
SURFV		6
SvgBlendedGradients		0
SvgDefaultImageExtension		".png"
SvgGenericFontFamily		0
SvgHiddenLineRemoving		0
SvgImageBase		""
SvgImageUrl		""
SvgLineWeightScale		1
SvgOutputHeight		768
SvgOutputWidth		1024
SvgPrecision		6
SYSCODEPAGE	r/o	"ANSI_1252"

T Variables

TabControlHeight		25
TABMODE		0
TabsFixedWidth		0
TangentLengthType		"0"
TangentLengthValue		"0"
TARGET		0",0",0"
TDCREATE	r/o	2456335.5399919
TDINDWG	r/o	0.121354456
TDUCREATE	r/o	2456335.8733252
TDUUPDATE	r/o	2456335.5399919
TDUSRTIMER	r/o	0.121354456
TDUUPDATE	r/o	2456335.8733252
TemplatePath		"C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V19x64\en_US\Templates\"
TEMPPREFIX		""
TestFlags		0
TEXTANGLE		0
TEXTED		2
TEXTEDITMODE		0
TEXTEVAL		0
TEXTFILL		1
TEXTQLTY		50
TEXTSIZE		0.2

Variable Name	Read-only	Default Value
TEXTSTYLE		"Standard"
TextureMapPath		"C:\Program Files (x86)\Bricsys\BricsCAD V19x64\Textures\1\"
THICKNESS		0
ThreadDisplay		"0"
THUMBSIZE		1
TILEMODE		1
TILEMODELIGHTSYNCH		1
TIMEZONE		-8000
ToolbarMargin	r/o	0
ToolbuttonSize	r/o	0
TooliconPadding	r/o	0
Tips		1
ToolbarIconSize		16
TOOLPALETTEPATH		"C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\19x64\en_US\Support\ToolPalettes\"
TOOLTIPS		1
TPSTATE	r/o	0
TRACEWID		0.05
TRACKPATH		0
TRANSPARENCYDISPLAY		1
TREEDEPTH		3020
TREEMAX		10000000
TRIMMODE		1
TSPACEFAC		1
TSPACETYPE		1
TSTACKALIGN		1
TSTACKSIZE		70
TTFTEXT		3

U Variables

UCSAXISANG		90
UCSBASE		""
UCSDETECT		0
UCSFOLLOW		0
UCSICON		3
UCSICONPOS		0
UCSNAME	r/o	""
UCSORG	r/o	0",0",0"
UCSORTHO		1
UCSVIEW		1
UCSVP		1
UCSXDIR	r/o	1",0",0"
UCSYDIR	r/o	0",1",0"
UNDOCTL	r/o	5
UNDOMARKS	r/o	0
UNITMODE		0
UseBIM		2
UseCommunicator		1
UseMechanical		1
USER11 thru USER15		0
USERR1 thru USERR5		0
USERS1 thru USERS5		""
UseSheetMetal		2
UseStandardOpenFileDialog		0

Variable Name	Read-only	Default Value
---------------	-----------	---------------

V Variables

VbaMacros		1
VENDORNAME	r/o	"Bricsys"
VerboseBimSectionUpdate		"1"
_VERNUM	r/o	"19.1.06 (UNICODE)"
VersionCustomizableFiles	r/o	"344"
VIEWCTR	r/o	10 7/16",4 1/2",0"
VIEWDIR	r/o	0",0",1"
VIEWMODE	r/o	0
VIEWSIZE	r/o	297
VIEWTWIST	r/o	0
VIEWUPDATEAUTO		1
VISRETAIN		1
VOLUMEPREC		-1
VOLUMEUNITS		"in ft mi µm mm cm m km"
VPROTATEASSOC		1
VSMAX	r/o	-1.0000E+20,-1.0000E+20,-1.0000E+20
VSMIN	r/o	1.0000E+20,1.0000E+20,1.0000E+20
VTDURATION		750
VTENABLE		3
VTFPS		7

W Variables

WarningMessages		65535
WHIPARC		1
WHIPTHREAD		0
WINDOWAREACOLOR		150
WIPEOUTFRAME		1
WMFBKGND		0
WMFFOREGND		0
WNDLMAIN		2
WNDLSCRL		0
WNDLSTAT		1
WNDLTABS		1
WNDLTEXT		1
WNDPMAIN	pt2d	0",0"
WNDPTEXT	pt2d	3'-4",3'-4"
WNDMAIN	pt2d	101'-2",66'-11"
WNDSTEXT	pt2d	118'-4",86'
WorkspaceSecurity		1
WORLDUCS		1
WORLDVIEW		1
WRITESTAT	r/o	1
WSAUTOSAVE		1
WSCURRENT		"2D Drafting"

X Variables

XCLIPFRAME		2
XDwgFadeCtl		70
XEDIT		1
XFADECTL		50
XLOADCTL		1
XLOADPATH		"C:\Users\userid\Documents\"
XNotifyTime		5

Variable Name	Read-only	Default Value
XREFCTL		0
XRefNotify		1
XREFOVERRIDE		0
Z Variables		
ZOOMFACTOR		60
ZOOMWHEEL		0
# Variables		
3DCOMPAREMODE		3
3DOSMODE		11
3dSnapMarkerColor		5

What's New in BricsCAD V19

BRICSYS UPDATES BRICSCAD SEVERAL TIMES A YEAR, WITH A MAJOR UPDATE EACH FALL.

This appendix lists BricsCAD's new and changed functions in V19, and is compiled from version 19.1.06-2. Changes are highlighted throughout this book, but be aware that information on these pages is not comprehensive. For information on functions added since this book was published, please see <https://www.bricsys.com/common/releasenotes.jsp>.

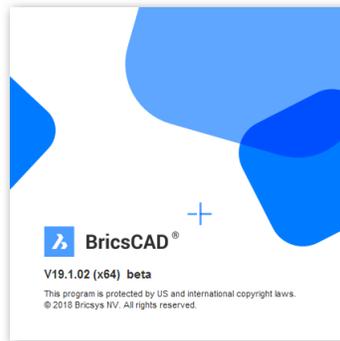
New command and variable names are shown here in boldface **blue**, and updated ones in boldface **black**. They are listed in alphabetical order in the following sections:

- User interface
- General commands & variables
- Assemblies
- BIM module
- Communicator module
- Import & export commands
- Generated views
- Sheet metal module
- APIs
- Licenses

What's New in the User Interface

BricsCAD V19 installs and runs independently from previous BricsCAD versions.

BricsCAD V19 displays a new splash screen when it starts up:

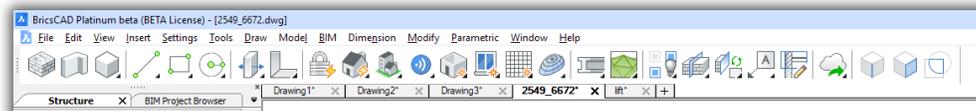


BricsCAD showing new splash screen for V19

Workspace command add a new workspace, "Drafting (Toolbars)," which replaces the ribbon with toolbars and the menu bar.

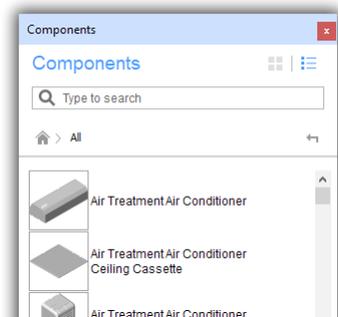
ProfileManager command now can change profiles without requiring Bricscad to restart.

The BIM workspace is changed to look like that of Shape, with a toolbar-like ribbon.



BIM workspace sporting a toolbar-like ribbon

Some panels (palettes) have taken on the look from Shape, such as the panel name in a large blue font.



Panel showing Shape-like look

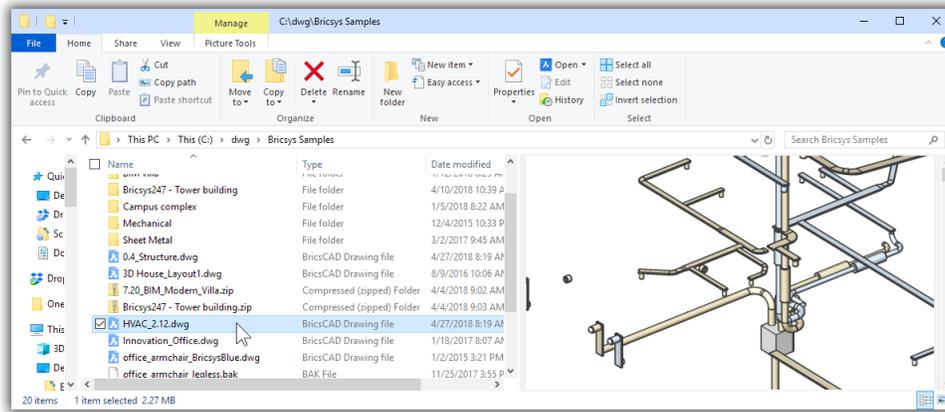
The ribbon now supports row breaks.

V19 improves display performance of BricsCAD:

- Zooms and pans are 2x faster for drawings containing a lot of tiny geometry; when using anti-aliased mode (AntiAliasScreen > 1) the performance improvement is 5x.
- Hatching is 100x faster for hatches with boundaries containing thousands of segments.

Dragged entities now remain visible during view manipulations like zoom, pan and view rotation.

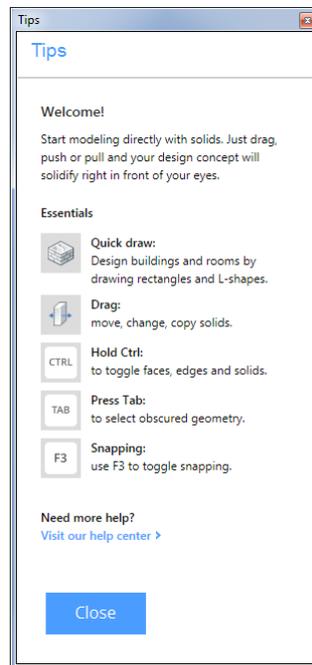
V19 registers with Windows to provide preview images for File Explorer. This lets you quickly view see the contents of files in folders of DWG files before opening them in BricsCAD.



Previewing a DWG file in Windows Explorer

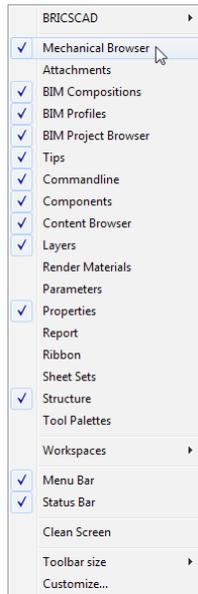
Help command provides help and tutorials online at <https://help.bricsys.com/hc/en-us>.

The new **Tips** panel animates command tutorials. To access it, right-click the ribbon or a toolbar, and then choose **Tips**.



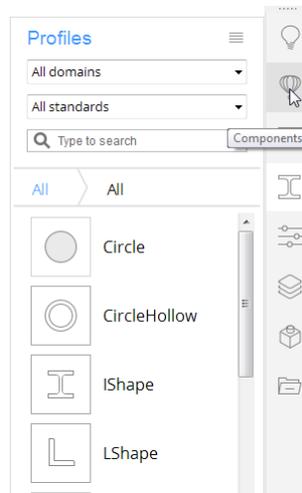
Animated help in the Tips panel

BricsCAD now has 19 panels; right-click a user interface element, such as the ribbon or a toolbar to see the complete list.



Panels included in V19

Flyouts minimize the space panels take up, replacing the tabs from earlier releases of BricsCAD. Flyouts can appear along the left or right edge of the drawing screen, and can be dragged from one edge to the other.



Flyouts appearing from the right edge of the V19 screen.

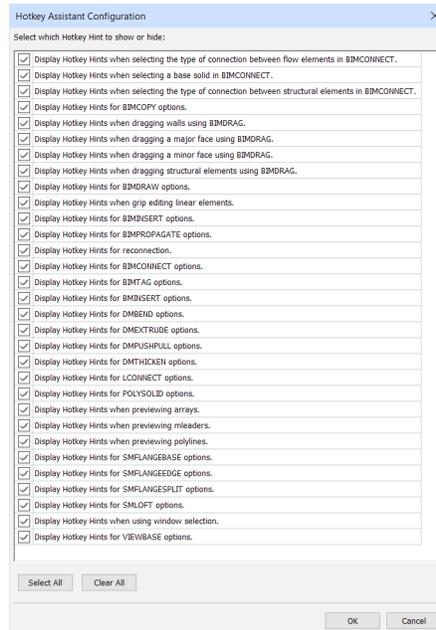
“BricsCAD V19 Mechanical” is the name of a new vertical package. It replaces the previous Sheet Metal add-on, and combines mechanical design and sheet metal design with BricsCAD Platinum.

Hot Key Assistant replaces the Tips facility. On the status bar, HKA replaces TIPS.



The Hot Key Assistant for the PolySolid command

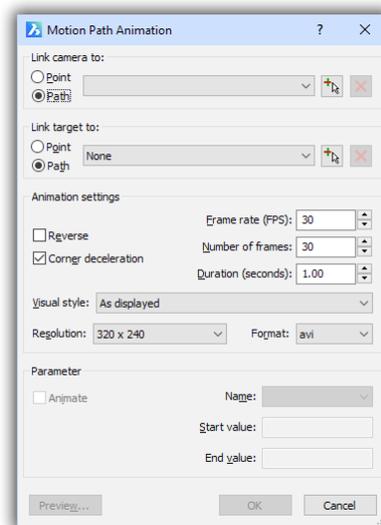
Right-click **HKA** and then choose **Configure** from the shortcut menu to access the Hotkey Assistant Configuration dialog box. It determines when hints are displayed by BricsCAD.



Determining which actions display the Hot Key Assistant

WHAT'S NEW IN GENERAL COMMANDS & VARIABLES

Anipath command's dialog box now supports variable parameter values during movies.



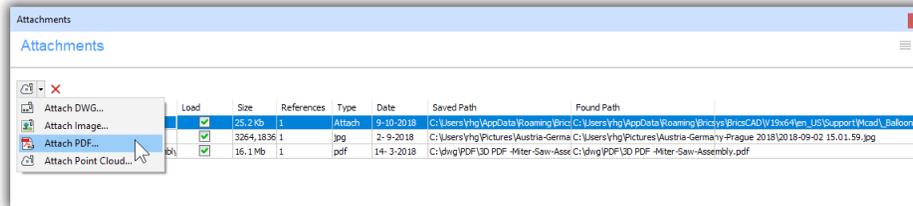
Parameter settings are found in the lower portion of the Motion Path Animation dialog box

Array command now works with parametric components.

TIP Component-based features (such as windows and doors) created with the new BC_UNITE and BC_SUBTRACT layers can be multiplied using associative arrays. This makes it possible to create solids, such as walls, with repetitive openings like windows.

AttDef command accepts **Ctrl+Enter** as the shortcut for clicking the **OK** button.

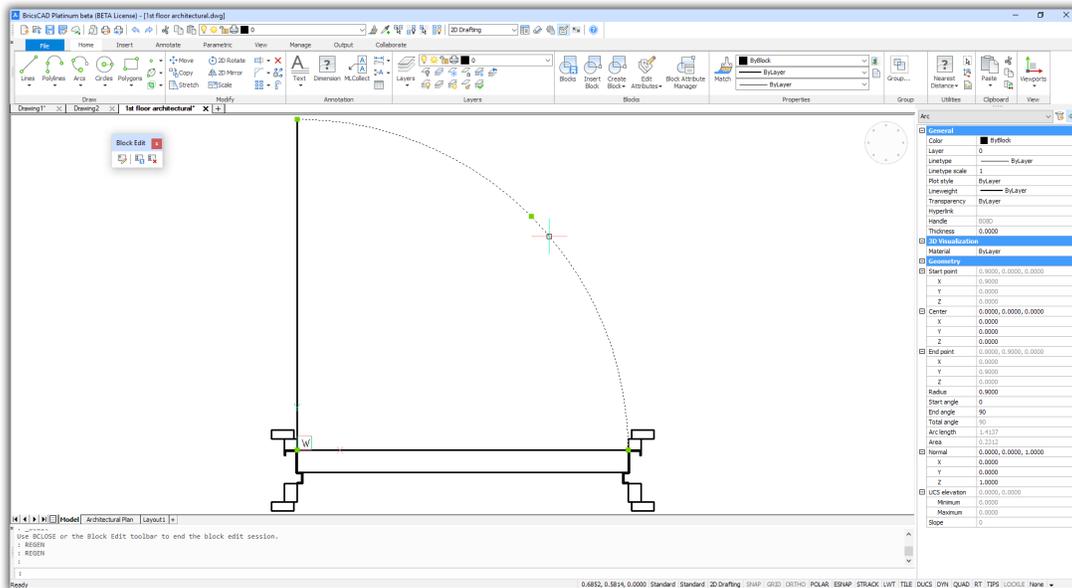
AttachmentsPanelOpen opens the Attachments panel for loading and managing xrefs, raster images, PDF files, and pointcloud attachments.



Attachments panel controlling DWG, image, PDF, and point cloud attachments

AttachmentsPanelClose closes the Attachments panel.

BEdit displays block entities in a temporary drawing session to create and edit block definitions; the block is opened in using the its local coordinate system. (It does not support AutoCAD's dynamic blocks.)



Block editing environment with Block tab in the ribbon

TIP Double-click a block to open it in the new block editor environment.

BClose exits the block editor, saving or not saving changes to the block.

Blockify command detects equally-shaped entities, and then replaces them with an equivalent block. The entities can be manually selected 2D entities (lines, polylines, arcs, circles, ellipses, splines, points) or 3D solids, or 3D solids automatically selected by BricsCAD. In this release, a mixed selection of 2D and 3D entities is not supported.

For example, select a line segment. BricsCAD searches the drawing for all other lines of the same length, creates a block that mimics them, and then replaces the lines with the block:

```
: BLOCKIFY
Select input entities or [Find all groups] <Find all groups>: (Select the Line segment)
Entities in set: 1
Select input entities or [Find all groups] <Find all groups>: (Press Enter to continue)
Select search space or [use entire Model space] <use entire Model space>: (Press Enter to search
the entire drawing)
Duration of finding similar groups: 0.000164 sec
Number of block inserts: 7
Select block insertion point or [use Default point] <use Default point>: (Press Enter)
```

The drawing looks no different, as the created block looks identical to the replaced entities. BricsCAD gives the block a generic name, such as 'block1'.

Here is the meaning of the Blockify command's options:

Select input entities	Select one or more representative entities to be replaced by block
Find all groups	Have BricsCAD find multiple instances of entities automatically
Select search space	Select the area of the drawing to search
use entire Model space	Have BricsCAD search the entire drawing
Select block insertion point	Pick a point, should you wish the block offset from the entities
use Default point	Have BricsCAD use the natural insertion point

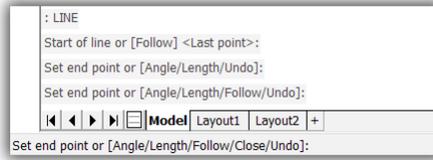
TIPS Replacing common elements with block references reduces the drawing size in memory and when saved to disk, as well as improving opening, drawing, zooming, and saving performance.

Use Blockify to convert general polylines in imported PDF files to blocks.

Centerline and **CenterMark** commands can now be applied to geometry in blocks and in drawing view viewports. The centerline entities can be copied and pasted, and exploded. **CL** and **CM** aliases are added.

TIP V19 no longer allows the creation of Center entities referring to geometry contained in non-uniformly scaled blocks

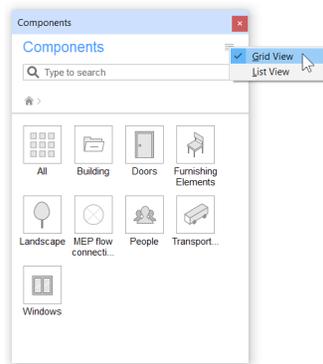
CliPromptLines variable specifies the number of lines of command history to appear in the drawing area. (CLI is short for “command line interface.”) When the command bar is closed (with Ctrl+9), the text of the command history is displayed in the drawing area. The text fades away after the number specified by this variable.



Prompt lines in the drawing area

- 0 Turns off the display of the command history in the drawing area
- 4 Default
- 64 Maximum number of lines of command history displayed

ComponentsPanelOpen command displays the Components panel for accessing architectural and mechanical parts; some are parametric. It accepts user-defined parametric components through the new -bmCreateComponent command.



Components panel handles regular and parametric blocks for BIM and mechanical

ComponentsPanelClose closes the Components panel.

ConvToMesh command converts the following entities mesh objects (mesh Smoothing is not yet supported):

- › 3D faces
- › 3D solids
- › 3D surfaces
- › Closed polylines
- › Polyface meshes
- › Polygon meshes
- › Regions

ConvToSolid command converts the following entities to 3D solids:

- › Watertight 3D meshes
- › Watertight surfaces

- Polyface meshes
- Closed polylines with thickness and uniform width
- Circles with thickness

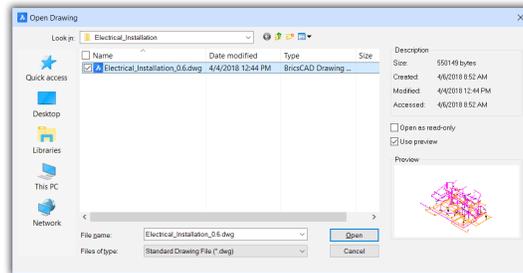
TIP A watertight mesh is one that completely encloses a volume with no gaps or openings – so that no “water” can leak out.

ConvToSurface command converts the following entities 3D surfaces:

- 2D solids
- 3D solids
- Arcs with thickness
- Lines with thickness
- Meshes
- Open polylines with thickness but of zero width
- Planar 3D faces
- Regions

TIP The **DelObj** variable controls whether entities converted to 3D surfaces are deleted.

CreateThumbnailOnTheFly variable toggles whether thumbnail preview images are generated for previews, such as in the Open dialog box (see figure below) and File Manager, when drawings lacks them.



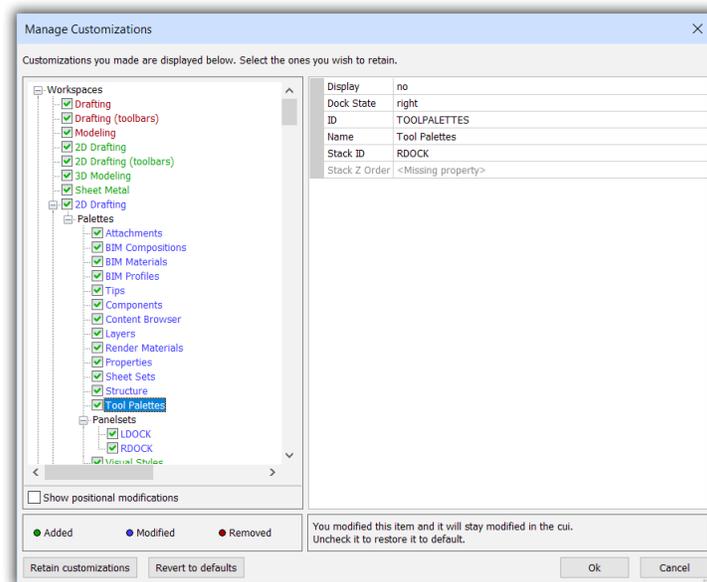
Preview image showing the content of the selected drawing

Customize command now reports on the status of user interface customizations. To access this function, in the Customize dialog box, click the new **Manage your customizations** button. The Manage Customizations dialog box lists the contents sorted by workspace.



New button in Customization dialog box

None of the fields in the right pane can be modified. In the left pane, the colors of the items have the following meaning:



Managing customizations

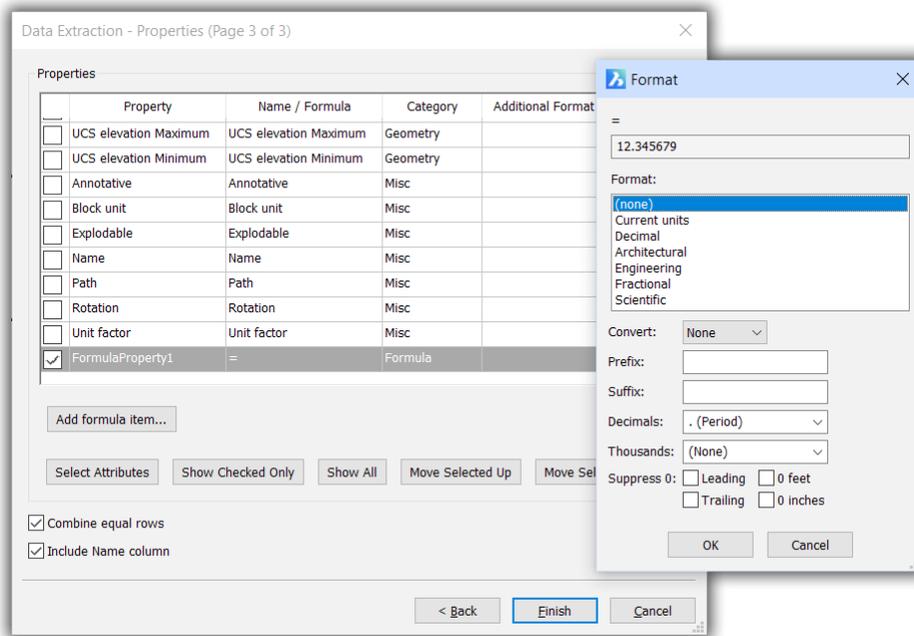
-
- Check mark** Keep modification
-
- No check mark** Revert modifications to the default
-
- Green text** Content added to the customization
-
- Blue text** Content that was changed
-
- Red text** Content that was removed
-

TIP This command also reverts changes made to IU customizations. This is handy when merging your personal CUI settings with an updated CUI file installed with a BricsCAD update.

VersionCustomizableFiles variable reports the current version of the CUI and PGP files, such as 317 (read-only).

DataExtraction command has a new Formula column to show results from combining values from regular columns, and defines filters in the DXD (data extraction definition) file:

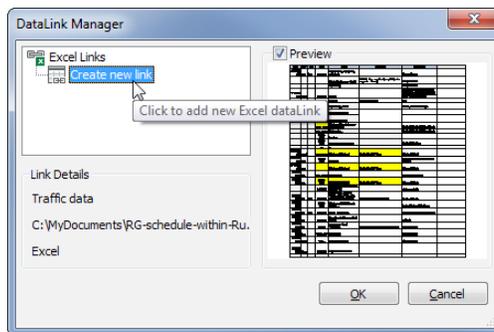
- ▶ Drawing properties
- ▶ Handle and entity type properties
- ▶ Coordinate properties
- ▶ Fixed symbol table record properties
- ▶ Vertex properties
- ▶ BIM ply properties
- ▶ Dynamic block properties



Adding formulae to the data extraction template

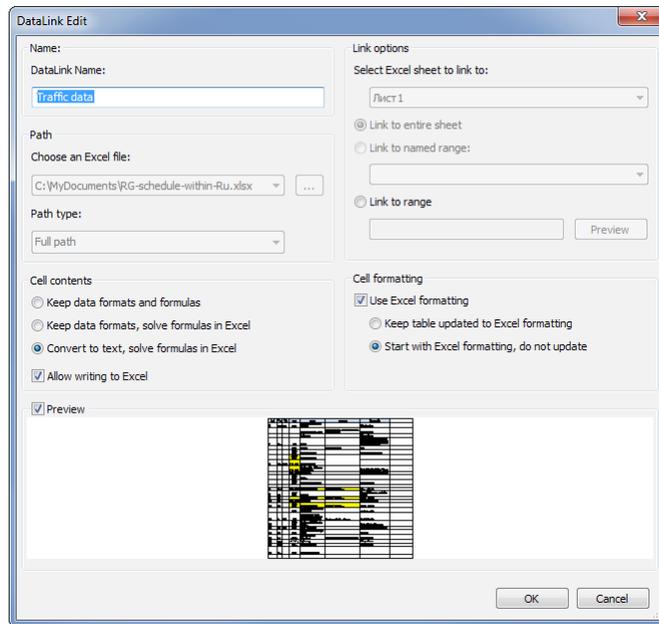
Datalink command imports Excel spreadsheet ranges into tables; preserves the link to update the table when the spreadsheet changes.

Warning This command works only when Excel is installed on the computer; it does not work with the view-only version of Excel, nor do Excel substitutes, such as Libre Office.



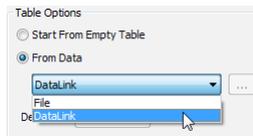
Above: Creating new data links; below: Managing links





Editing the data link

TIPS To link a table with a spreadsheet, start the **Table** command and then choose “Datalink” from the **From Data droplist** in the Table Options section of the Insert Table dialog box.



You can copy'n paste a range of cells from a table to Excel, and vice versa.

DataLinkNotify variable controls data link notifications:

- 0 Disabled; changes to the data link are not reported
- 1 Enabled; changes to the data link are reported
- 2 Enabled; changes to the data link are reported with a balloon (default)

DatalinkUpdate command updates links in drawings with Excel spreadsheets:

: **DATALINKUPDATE**

Select an option[Update data link / Write data link] <Update data link>: (Press Enter)

Select objects or [Update all]: (Press Enter)

Dim command is rewritten to apply dimensions based on the entity selected, such as radial dimensioning of circles. Select a dimension to continue dimensioning in continuous or baseline mode;. The updated command also works with sub-entities and dimensions through layout viewports.

Here is the full prompt line, which includes the new **DIStribute** and **LAYER** options:

```
: DIM
Dimensioning command [HORizontal/VERTical/ALigned/ANGular/Leader/OBlique/ROtated/CenTer/Diam-
eter/RADius/BASeline/COntinue/ORdinate/POsition/DIStribute/UPdate dimensions/variable SStatus/
OVerride/SEttings.../LAYER]:
```

When the cursor passes over a circle, for instance, the prompt changes to the following:

```
Select arc or circle to specify diameter or [Radial/Angular]: (Pick a circle)
Location of dimension line [Angle/Text]: (Pick a point)
Dimension text: = 5.0474
```

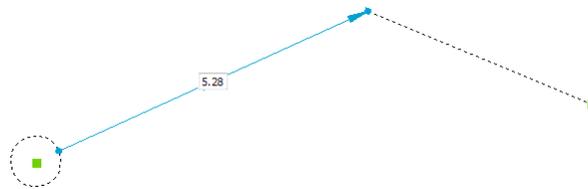
Most of the terse option names are self-explanatory; here is the meaning of the option names I find vague:

- Position** repositions the dimension text (and associated leader, if any)
- DIStribute** spaces dimensions evenly or by an offset distance
- SEttings** displays the Drawing Explorer window with dimension styles
- LAYER** specifies the name of the layer on which to place the dimension

DimLayer variable specifies the existing layer on which to place dimensions; creates the 'dimlayer' (for dimensions), 'centerlayer' (for center marks), and 'hplayer' (for hatch patterns) layers when you type in the names, but they do not already exist in the drawing. Default is the current layer (.).

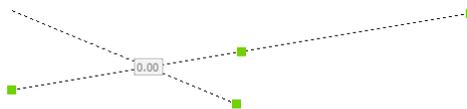
Dimension style and text style controls now change the style of dimension and text entities in the current selection set.

NearestDistance variable toggles a dynamic dimension that reports the nearest distance between two selected entities. The distance reported is the shortest distance. (This function does not work when three or more entities are selected.)



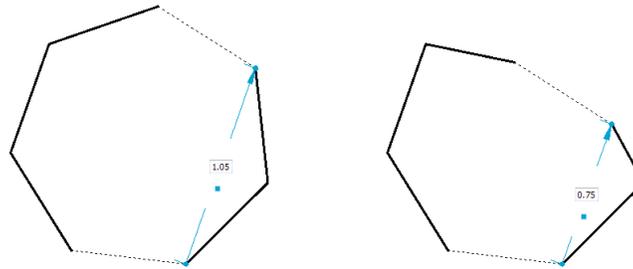
A dynamic dimension showing the distance between two selected entities

When two entities intersect or connect, the distance is zero; the distance cannot be changed.



*Zero distance between intersecting lines;
gray field cannot be edited*

Hold down the **Ctrl** to select sub-entities, such as two of the polyline segments that make up a rectangle.



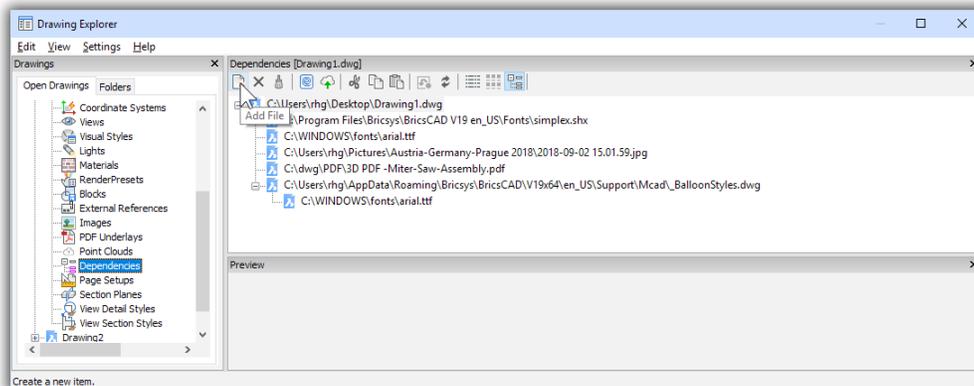
*Left: Shortest distance between two segments of a polygon;
right: editing the distance between them*

TIP You can edit the distance to move the last selected entity; the arrowhead points at it. When an entity cannot be moved, the dimension field is read-only (not editable).

Entity snaps now snap to the geometry of AcDbSubDMesh mesh entities and the content of mleader blocks. **Tangent** and **Perpendicular** entity snaps work with grips to edit coincident entities.

TIP With **Tangent** and **Perpendicular** entity snaps, you can select the opposite end of a line that's coincident with an arc. As you move the entity close to being perpendicular or tangent, it snaps into place.

eTransmit command now adds files to transmittal packages through the **Add File** button. (In previous versions, the Add File button was grayed out and unavailable.)



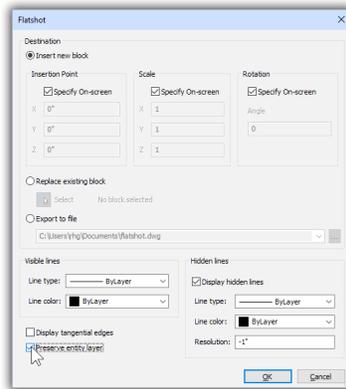
The Add File button works for transmittals in V19

ExportLayout command no longer explodes hatches laying on the view boundary.

TIP Unlike other dwg editors, V19 preserves the hatch and trims its boundary to match the viewport boundary. Preserving the hatch avoids creation of potentially thousands of lines for the dashes.

Field command supports multi-part formulas, such as (Table(261).Sum(A2:A3)+1+Table(261).A2).

Flatshot command adds a **Preserve entity layer** check box in its dialog box to preserve the layer settings of entities in hidden-line perspective drawings. The command now accepts PolyfaceMeshes and 3dFaces as 3D entities that can be flattened.



Preserve entity layer option added to Flatshot dialog box

Gradient command gets variables to control the look of gradient hatches; see list below.

GfAng variable specifies the angle of a gradient; range is 0 to 360 degrees.

GfClr1 variable specifies the first color gradients; any color designation can be entered.

GfClr2 variable specifies the second color gradients; any color designation can be entered.

GfClrLum variable specifies the level of shade level in one-color gradients; range is between 0 (black) to 1 (white).

GfClrState variable determines whether the gradient uses one or two colors:

- 0 Two-color gradient (default)
- 1 One-color gradient

GfName variable determines the look of the gradient:

- 1 Linear (default)
- 2 Cylinder
- 3 Inverted cylinder
- 4 Sphere
- 5 Hemisphere
- 6 Curve
- 7 Reverse sphere
- 8 Reverse hemisphere
- 9 Reverse curve

GfShift variable toggles the center of the gradient:

- 0 Center
- 1 Upper left corner

Grading command shapes terrains for building sites, such as building pads, retaining walls, parking lots, and streets.

Grid command now create rectangular and radial grids with automatic labels.

Hatch command operates as much as 100x faster for hatches with boundaries containing thousands of segments. Also, it gets new color and background colors in the Pattern section of the dialog box, along with matching **HpColor** and **HpBackgroundColor** variables.

TIP **HpStyle** variable is renamed **HpIslandDetection**.

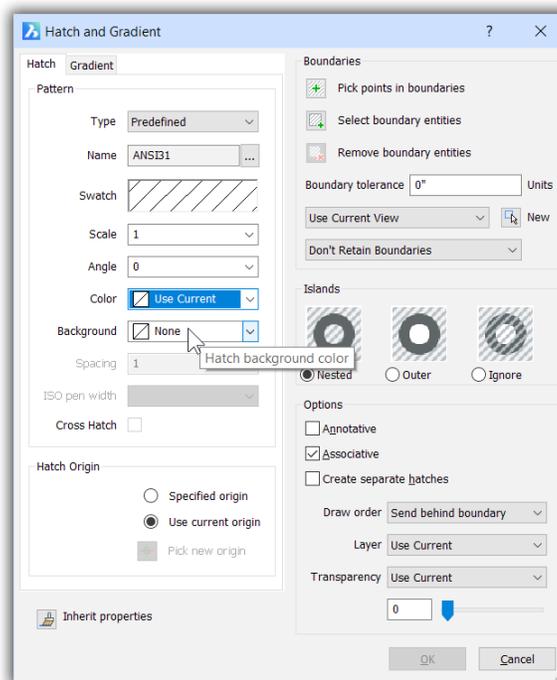
HpBackgroundColor variable specifies the background color for new hatch patterns; any color designation can be entered.

HpColor variable specifies the color of new hatch patterns; any color designation can be entered.

HpIslandDetection variable determines how islands are handled as hatch boundaries:

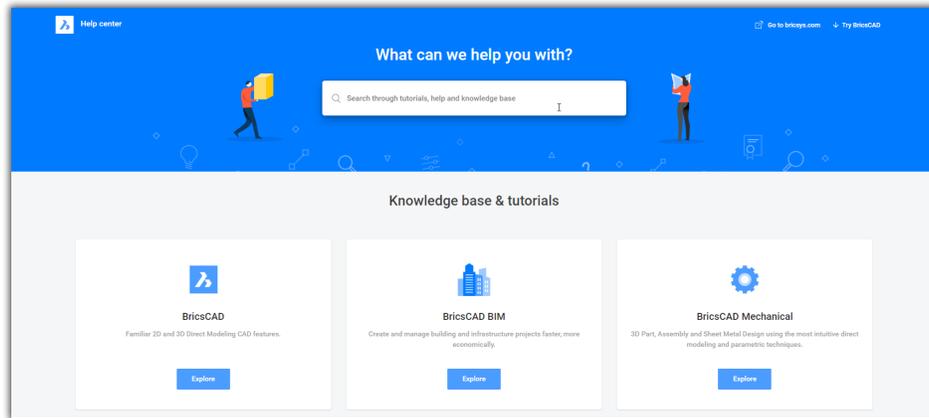
- 0 Normal -- island areas are hatched and not-hatched in alternating order (default)
- 1 Outer -- only the outermost area is hatched
- 2 Ignore -- everything is hatched within boundaries

HpStyle variable is removed from V19 and replaced by **HpIslandDetection**.



New options in the Hatch and Gradient dialog box

Help command offers BricsCAD documentation online at <https://help.bricsys.com/hc/en-us>.



New look of the online help for Bricsys products

InsUnitsScaling variable controls the INSUNITS variable:

- 1 Enable InsUnits-based scaling flag
- 2 Disable InsUnits, and instead use paper size unit for paper space insertions

LConnect command creates a connection between two faces of 3D solids:

: **LCONNECT**

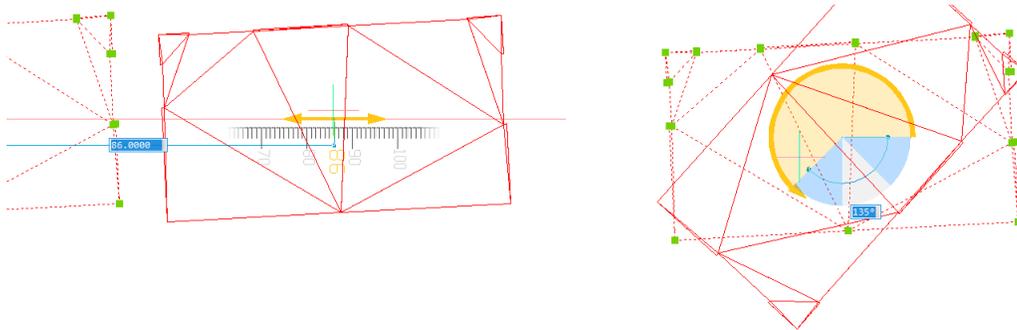
Select solids or faces to connect:

Select solids or faces to connect:

MaterialAssign command applies the material specified by the **CMaterial** variable to the selected 3D entities. Alternatively, you can drag a material from the Material Browser panel onto the entities.

TIP To change the material during the **MaterialAssign** command, hold down the **Alt** key while selecting the entity.

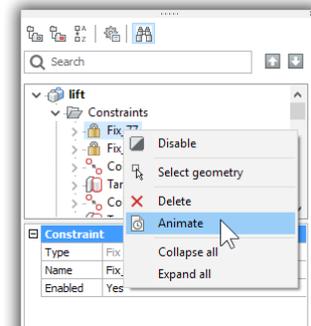
Manipulator command adds a ruler for distances, and a protractor for angles. The size of the markings change dynamically according to zoom level. After making a copy, the copied entities now become the selected ones.



Left: Manipulator with ruler for distances; right: with protractor for angles

Also, the manipulator is added to many more commands such as Drag and PushPull. The manipulator also appears when you press **Ctrl+A** to select all objects in the drawing.

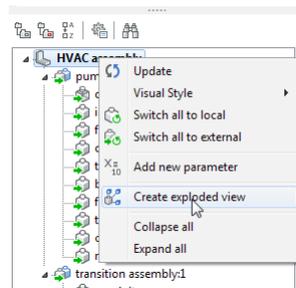
MechanicalBrowserOpen command can search for nodes by name through the new **Show Search** button.



New Animate option in the context menu, and Search button on the toolbar

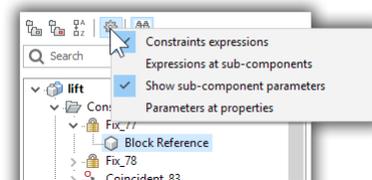
When right-clicking parameters and 3D constraints, the context menu lists the new **Animate** option (see above). to run a short animation that shows the geometry controlled by the selected item,

The topmost node gets a new **Create Exploded View** option (see below); it runs the bmExplode command.



Newly added Create Exploded View option

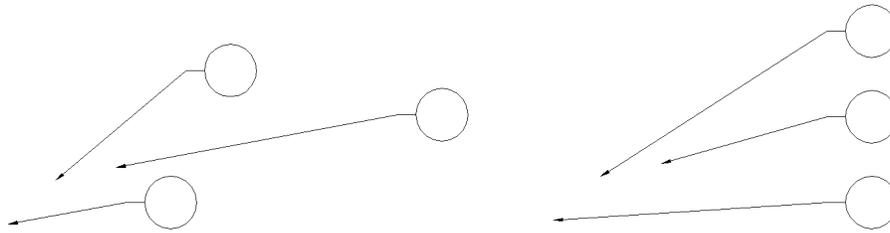
The new **Settings** menu (see figure below) controls the visibility of parameters of sub-components.



Settings button displaying options in Mechanical browser

Searches in the Mechanical Browser are activated by pressing the **Show Search** button on toolbar panel, or by pressing **Ctrl+F**, or by just typing some text while the focus is inside the browser.

MLeaderAlign command aligns mleaders:



Left: Mleaders before... ; right: ...and after being aligned

: MLEADERALIGN

Select multileaders: **all**

Entities in set: 3

Select multileaders: *(Press Enter to continue)*

Specify first point or [Options]: *(Pick a point)*

Specify second point: *(Pick a second point)*

Here is the meaning of the MleaderAlign command's options:

byMleader	Aligns other mleaders to the selected mleader
onpolyLine	Arranges mleaders' contents along a path whose points you pick
Parallel	Arranges mleader lines parallel to the selected one
Spacing	Spaces mleaders the specified distance from the selected one
Circle	Arranges mleaders in a circle around a specified center point

TIP This command can be used repeatedly to keep changing the alignment of the group of mleaders.

MLeaderCollect command collects two or more multi-leaders *with blocks* and then combines all leader lines into a single one, with the blocks aligned vertically or horizontally, or wrapped to a specified width:

: MLEADERCOLLECT

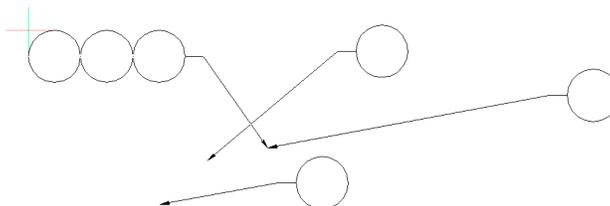
Select entities: **all**

Entities in set: 3

Select entities: *(Press Enter to continue)*

Specify collected multileader location or [Vertical/Horizontal/Wrap/Sorting/Collapsing]: *(Pick a point)*

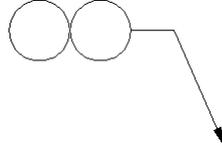
TIP This command works only with mleaders that have blocks as their annotation. Blocks can be sorted in ascending and descending order. The command cannot be reused once applied to a group of mleaders.



Three mleaders (at right) being collected into a single one (at left)

Here is the meaning of the MLeaderCollect command's options:

Vertical	Stacks blocks vertically
Horizontal	Arranges blocks horizontally
Wrap	Limits the maximum number of blocks in a user-specified width
Sorting	Sorts the blocks in ascending or descending order
Collapsing	Joins selected leaders into a single leader, preserving blocks



Two mleaders collapsed into one

MText command supports the NODe entity snap.

NODE command now supports mtext entities

Nudge moves selected entities by small increments:

X direction	Hold down Ctrl key and press Left and Right arrow keys
Y direction	Hold down Ctrl key and press Up and Down arrow keys
Z direction	Hold down Shift and press Up and Down arrow keys

NUDGE: offset along the X axis of the UCS: 5/128"

NUDGE: offset along the Y axis of the UCS: 5/128"

NUDGE: offset along the Z axis of the UCS: 5/128"

TIP The step size is specified by the adaptive grid-snap resolution: the further out the drawing is zoomed, the bigger the nudge distance.

Number command draws incremented number tags.

: NUMBER

Select entities to number and press Enter **all**

Entities in set: 6

Select entities to number and press Enter (*Press Enter to continue*)

Current settings: First index = 1, Increment = 1, Prefix = "", Suffix = "", Number Style = Arabic, Entities Sorting = None, Existing Numbers are <overwritten>

Provide First index or [Increment/Prefix/Suffix/Number style/Entities sorting/Overwrite Numbers]
<1>:

Here is the meaning of the Number command's options:

First index	Value with which numbering should start; default = 1
Increment	Value by which numbers increment; default = 1
Prefix	Prefix for numbers; default = "" (none)
Suffix	Suffix for numbers; default = "" (none)

Number style	Specifies the type of numbering:
0	Arabic, such as 1, 2, 3
1	Roman uppercase, such as I, II, III
2	Roman lowercase, such as i, ii, iii
3	Letters uppercase, such as A, B, C
4	Letters lowercase, such as a, b, c
Entities sorting	Sorts the entities by z, y, z, coordinates
Overwrite numbers	Determines if numbers are kept or overwritten

OrbitAutoTarget variable controls the target point for real-time view rotations:

0	Target point is where you click to start orbiting
1	Target point is at the center of all entities seen on the screen, or of selected entities (default)

Pan command is 2x faster for drawings containing a lot of tiny geometry; when using anti-aliased mode (**AntiAliasScreen** > 1) the performance improvement is 5x.

Panelize command draws free form surfaces as subdivision meshes, collected into a block.

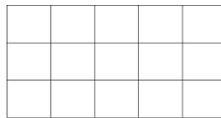
: **PANELIZE**

Select face or set [Result]:

Create grid or [Length U panels/Length V panels/Number U panels/Number V panels]:

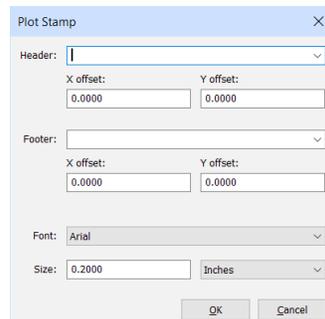
Max deviation from planarity: 0.000000

The **Result** option determines if the grid is made of polylines or meshes.



Face panelized by polylines

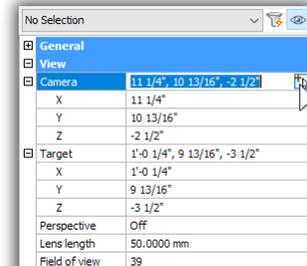
PlotStamp command displays the Plot Stamp dialog box for specifying header and footer text added to plot output. This dialog box is not new, as it was accessible through the Plot Stamp button in the Print dialog box; now it is accessible directly with a command.



You can enter your own text, or else select from a list of \$-prefixed metadata that gets info from the system and from the data stored by the DwgProps command.

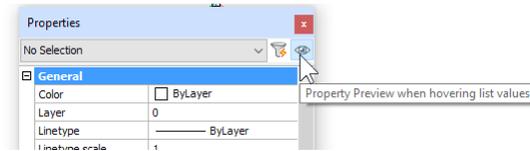


Properties command's panel changes the **Camera** and **Target** properties from read-only to editable.



Editable Camera and Target fields in the Properties panel

The new Eye icon toggles whether previews of selected entities are displayed.



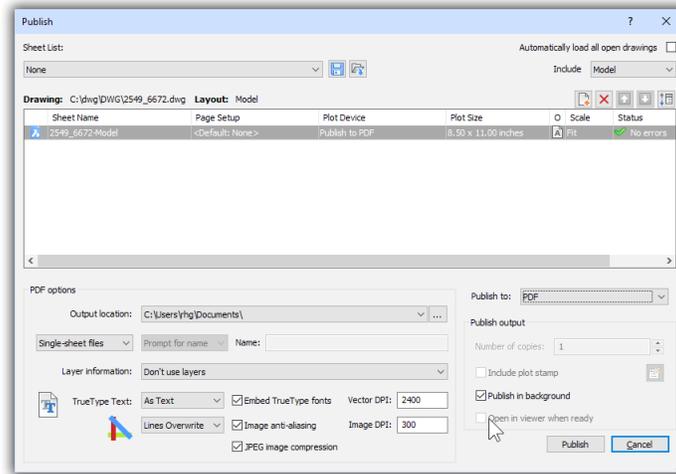
The Eye icon

PropertyPreview variable toggles the display of previews of selected entities. This variable can be toggled by the new eye button in the Properties panel, as shown above.

PropertyPreviewDelay variable specifies a delay before PropertyPreview starts; range is 100 to 10000 milliseconds; default is 500..

PropertyPreviewObjLimit variable specifies the maximum number of entities used by Property-Preview; range is 1 to 30,000; default is 500.

Publish command gains a **Open in viewer when ready** check box for PDF files; PDF files are opened after publishing when '**Publish to**' is set to PFD, and **Publish in background** is off.



New 'Open in viewer when ready' option in Publish dialog box

Purge command now purges the following unused tables from drawings, but still no dialog box!

- Detail view styles
- Groups
- Multileader styles
- Section view styles
- Shapes
- Zero-length geometry

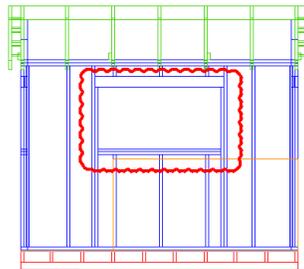
: **PURGE**

Purge [BATch all/purge All/Blocks/**DE**tai**l** view styles/Dimension styles/**G**roups/**L**Ayers/**L**ine**T**ypes/**M**Aterials/**M**Line styles/**MU**ltileader styles/Plot styles/Regapps/**S**ection view styles/**S**Hapes/Table styles/text **S**Tyles/**V**isual styles/**Z**ero-length geometry/Empty text entities/Orphaned data]: (Enter an option)

RevCloud command adds **Rectangular** and **Polygonal** options for shapes of revision clouds:

: **REVCLLOUD**

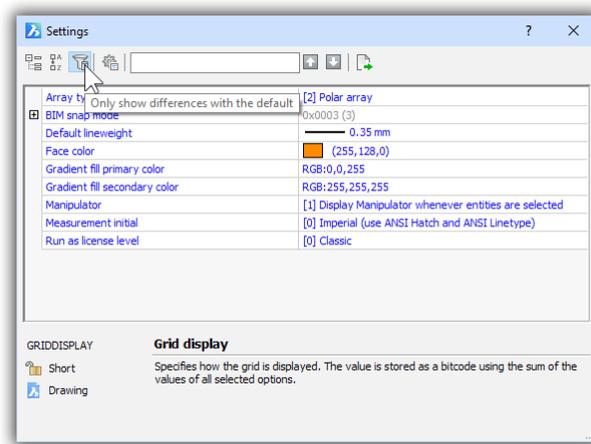
Specify first corner point or [Arc length/Entity/**Rectangular**/**Polygonal**/Freehand/Style] <Entity>:



Rectangular revision cloud

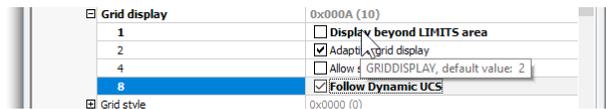
SectionPlaneToBlock command now accepts PolyfaceMeshes and 3dFaces as valid types.

Settings command's dialog box now shows non-default values in a user-definable color; click the Filter button:



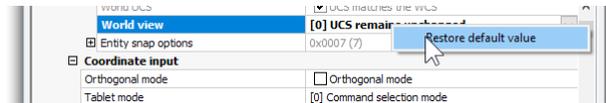
Settings dialog box showing only those values that differ from the defaults

A tooltip reports the default value when hovering over a setting:



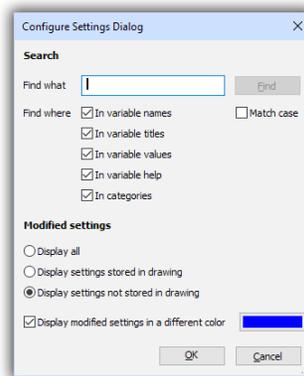
Tooltip reporting the default value of a setting

An option in the right-click menu resets the value to its default value:



Right-clicking a changed value to reach the restore option

The new **Configuration** button determines how settings are displayed:



Configuring the Settings dialog box

The buttons for jumping directly to Drawings, Dimensions, and Program sections were removed from V19

Site command imports points files in CSV (comma-separated values) format, drawing entities or Civil 3D surfaces to create terrain models.

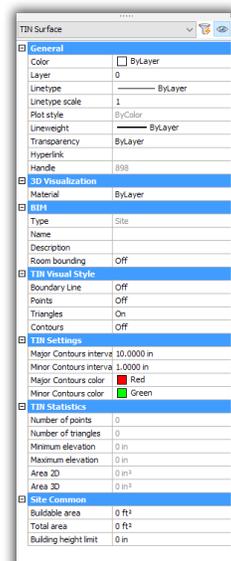
: **SITE**

Select entities to create site or [Import from file/Place points/create from civil 3d surface]:
(Enter an option)

Here is the meaning of the Site command's options:

Entities	Use entities already in the drawing
Import from file	Import data from a points file, saved in CSV format
Place points	Pick points in the drawing
create from civil 3d surface	Use an existing Civil 3D surface already in the drawing

V19 works with a new entity called "TIN Surface"; TIN is short for triangular irregular network.



Properties for the new TIN entity

SiteEdit command modifies terrains by adding/removing boundaries, breaklines, and points.

SheetSet command switched creating sheet list tables from plain text to fields and hyperlinks, which refer to sheet properties..

TIP Because it uses fields and hyperlinks, the sheet list now automatically updates when the properties of sheets are changed.

SnapType variable adds adaptive snap step size option (2) to grids.

TIP The grid display and the snap distance change according to the current zoom factor. This adaptive snap step is also used by the new Manipulator ruler and new **Nudge** command

Slice command gains the **Multislice** option.

: **SLICE**

Select entities: *(Select one or more 3D solid entities)*

Entities in set: 1

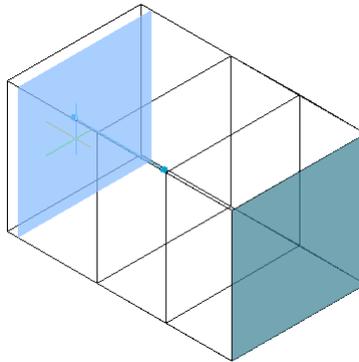
Select entities: *(Press Enter to continue)*

Specify first point on slicing plane or [Object/Surface/Zaxis/View/XY/YZ/ZX/Line-point/3points/**Multislice**] <3points>: m

Select a planar Face or a planer Surface or <XY>: *(Pick a face)*

Specify distance to create slice: *(Pick one or more slice Locations)*

Specify distance to create slice or Repeat [**Repeat**]: *(Press Enter to exit)*



Making multiple slices

Spell command now works with fields in texts, mtexts, leaders, mleaders, tables, and block attributes.

DctCust variable now accepts just a name, or a name and path for the custom dictionary; if the dictionary can't be found, then a new one is created.

Spline command gains the **Undo** option to back up through pick points:

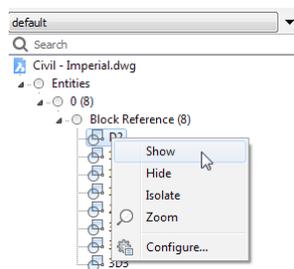
: **SPLINE**

First point for spline: *(Pick a point)*

Second point: *(Pick another point)*

Set next point or [Close/Fit tolerance/**Undo**]: *(Pick another point)*

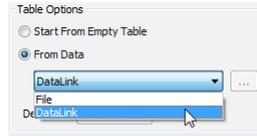
StructurePanel command gets **Show**, **Hide**, and **Isolate** options in its right-click-menu; also get a new text search field.



New options in the context menu

TIP The Structure panel allows you to select specific entities to change visibility, unlike the **Unhide** and **Unisolate** commands, which make all hidden entities visible. Hidden entities are shown in gray text.

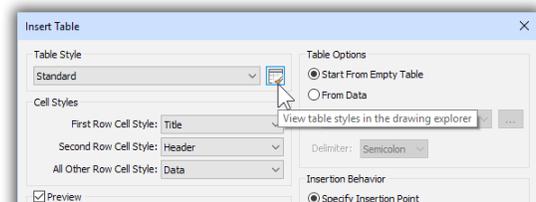
Table command now links with Excel spreadsheet files; see the DataLink command.



Starting the link between table and spreadsheet

TIP Changing the spreadsheet changes the content of the table, and vice versa.

The Insert Table dialog box's user interface gains tooltips.



Tooltips explaining the meaning of elements in the Insert Table dialog box

TConnect command connects planar faces and linear solids to other solids and faces. (See also LConnect command.)

: TCONNECT

Select face to connect:

Select face to connect:

Select entities to connect to or [Connect to nearest] <Connect to nearest>:

TIP Press the **Tab** key to select the face you want.

TextEd variable's default value is changed to 2 so that the in-place editor is used for single line text.

TK (tracking) now allows any number of tracking segments, and is no longer limited to 7.

Tolerance command-created entities now launch the **DdEdit** command — instead of the **Properties** command — when a tolerance is double-clicked.

ToolPalettes command can now configure dynamic block properties inserted from the Tool palette panel.

VisualStyles command now displays a warning when a visual style is edited in a way that is incompatible with AutoCAD.

Zoom command is 2x faster for drawings containing a lot of tiny geometry; when using anti-aliased mode (**AntiAliasScreen** > 1)the performance improvement is 5x.

WHAT'S NEW IN ASSEMBLIES

(dm- commands; requires a Platinum license)

dmAngle3d command now creates Planar Angle constraint by default, using coordinate planes of the WCS as the third reference entity when possible.

dmAudit command now checks and heals blocks, sliver faces, and coincident faces.

dmAuditAll command extends the **dmAudit** command to check for, and heal flaws in drawings inserted as external references.

```
: DMAUDITALL
Select entities to audit [Entire model] <Entire model>: (Press Enter)
Choose action [Check/Fix/Settings] <Fix>: (Enter an option)
Selected count: 4
===== Block "Model space" =====
----- Solid -----
      Handle: 79
      Name in Mechanical Browser: <unknown>
      Flaws: None
----- Skipped: -----
      1 Spline
No flaws were found.
===== Before fix =====
Flaws found in this drawing: 0
===== After fix =====
Flaws found in this drawing: 0
```

The meaning of the dmAuditAll command's options are the same as for the dmAudit command:

dmConstraint3d command now manages 3D constraints and edits their properties, as shown by the new prompt line below.

```
: DMCONSTRAINT3D
Enter option [New/Edit/Rename/Delete/?] <?>: new
Specify 3D constraint type [Fix/Coincident/CONcentric/Parallel/PErpendicular/Tangent/RIGidset/
Distance/Radius/Angle]:
```

Here is the meaning of the dmConstraint3d command's new options:

New	Displays the list of 3D constraints that can be applied
Edit	Prompts to enter the constraint's name to be edited
Rename	Prompts to enter the constraint's name to be renamed
Delete	Prompts to enter the constraint's name to be deleted
?	Displays help for this command

The **3Ddistance** and **3Dcoincident** constraints now support point-cylinder and point-sphere combinations.

3D constraints now display widgets when a constrained 3D solid face is selected, and receive “Directions” and “Placement” properties. The widget for the **3Ddistance** constraint displays a dimension.

3Dnearest now snaps to edges of ACIS entities.

TIP Mechanical Browser can be used to edit the new properties of 3D constraints.

dmExtrudeMode variable gets new flag, 4, to prevent intersecting 3D solids from being modified (as created by the dmExtrude command):

4 Set on to not modify solids which intersect with created volume (off, by default)

dmPushPull command makes it easier to choose a reference face: hover the cursor over a reference face, and then press **Tab** — or **Shift+Tab** — to select the opposite parallel face.

dmSimplifyAll command extends the **dmSimplify** command to simplify geometry in drawings inserted as external references.

WHAT'S NEW IN BUILDING INFORMATION MODELING

(bim- commands require an extra-cost module)

The BIM workspace has a new user interface that mimics the Shape program, with a toolbar made of large icons and tabbed dock panels. BIM-related panels are re-designed to match the look of Shape.

V19 now classifies elements according to any national or company standard classification system. Linear building elements now support the same grip-stretch operations as with lines. Reflected ceiling plans are 2D sections showing elements on the ceilings of rooms and other spaces. When a connected structural element is rotated by 90°, the connection is (optionally) restored.

bimApplyProfile gains the [convert solids to Line](#) option to convert linear solids (straight solids) to lines (along their axes:

```
: BIMAPPLYPROFILE
Select path(s): (Choose a Linear solid)
Entities in set: 1
Select path(s):
Select profile [in Dialog/convert solids to Line] <in Dialog>: 1
```

bimAttachComposition command now allows you to choose reference and opposite surfaces manually with the new [Entity](#) option.

```
: BIMATTACHCOMPOSITION
Select entities to attach composition: (Pick an element)
Entities in set: 1
Select entities to attach composition: (Press Enter to continue)
Enter composition name or [Dialog/Entity] <Dialog>: e
Select other entity to get composition: (Select an entity)
```

bimCheck removed from V19.

bimConnect removed from V19, replaced by LConnect.

bimCopy command works like the **bimDrag** command for moving entities normal (at 90 degrees) to the selected face, but this command makes a copy.



White beam being copied by the bimCopy command

: BIMCOPY

Select several entities/subentities: *(Select one or more entities)*

Entities in set: 1

Select several entities/subentities: *(Press Enter to continue)*

Specify distance to create copy: *(Move the cursor or enter a value)*

Specify distance to create copy or switch to [Copy/Repeat/Accept] <Accept>: *(Press Enter)*

bimCurtainWall command creates curtain walls made of planar quadrilateral panels from free form surfaces.

: BIMCURTAINWALL

Select a face: *(Select a face)*

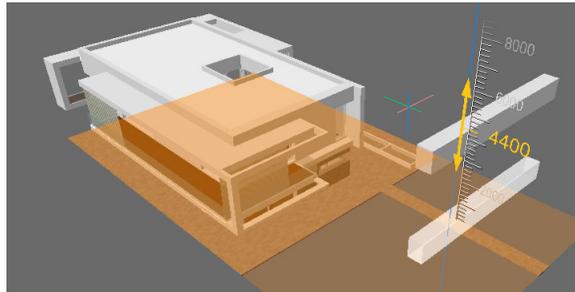
bimDisplayComposition variable toggles the display of compositions on and off.

: BIMDISPLAYCOMPOSITION

Display composition: [Toggle/On/off]: *(Enter an option)*

Select entities to display the composition <Drawing>: *(Select one or more entities, or press Enter)*

bimDrag command now moves the entire 3D solid when dragging a face; it maintains connectivity between non-orthogonal walls; and works with the Manipulator.



bimDrag displaying the ruler from the Manipulator

TIP Hold down the **Ctrl** key to turn off connectivity. Use **dmPushPull** to move the face of a 3D solid.

bimExport removed from V19, replaced by Export command's IFC option.

bimFlowConnect connects linear solids.

bimGetStatisticalData removed from V19;

bimGrid command creates rectangular and radial grids with automatic labels.

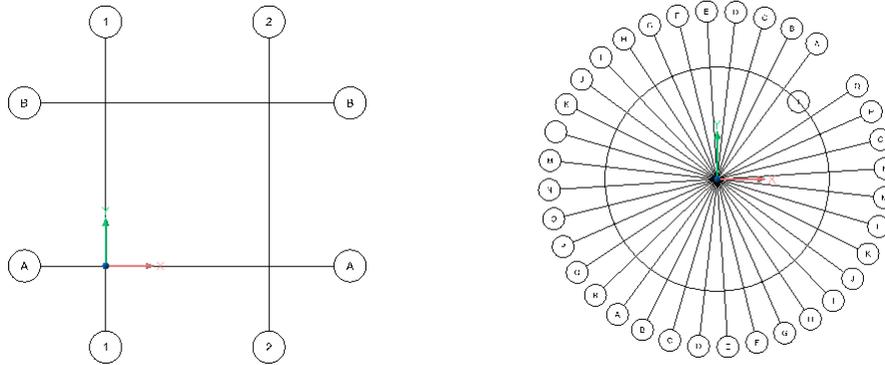
: BIMGRID

Enter first point [offsetU/offsetV/offset Labels/Radial]: **0,0**

Enter second point [offsetU/offsetV/offset Labels]: *(Pick a point)*

Here is the meaning of the bimGrid command's options:

First point	Specifies the starting point of the grid
Second point	Specifies the opposite corner for a rectangular grid
offsetU, offsetV	Specifies the distance between u and v grid axes
offset Labels	Specifies the offset distance for labels
Radial	Switches to the radial grid

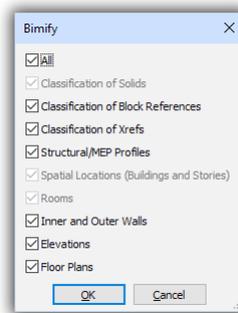


Left: Rectangular grid... ; right: ...and radial grid

bimIfcImport is removed from V19, replaced by Import command's IFC option

bimIfy command adds the following functions:

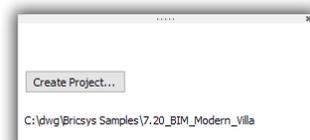
- ▶ Detects flow segments and fittings, depending on the drawing type (such as architectural, structural, or MEP)
- ▶ Classifies block references automatically; detects rooms and outer walls automatically
- ▶ Adds a dialog box to select entities for a partial bim-ification.



New dialog box added to the bimIfy command

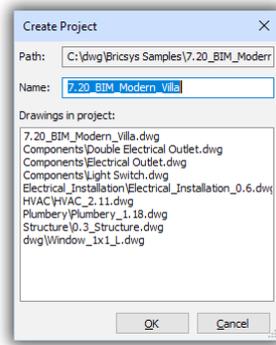
BIM Project Browser panel generates and navigates between models, sheets, and schedules in the BIM project. To access this panel, right-click the ribbon or a toolbar, and then select BIM Project Browser. When no project exists for the current BIM drawing, follow these steps:

1. Click the **Create Project** button.



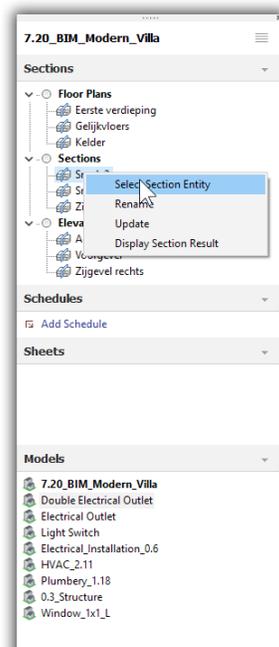
Creating a new BIM project

2. Fill in information about the project. If you are in a hurry, just click **OK**.



Creating a new project;

TIP There is no command to open this panel. Instead, right-click the ribbon, and then choose **BIM Project Browser** from the context menu.



BIM Project Browser panel

bimProperties command now imports custom properties from XML files.

bimPropagate command (replaces **bimSuggest**) maps any detail (such as solids, holes, and finishing geometry) from certain base solids to all similar base solids, as well as on a grid.

: **BIMPROPAGATE**

Select base solids. (Select one or more solids)

Entities in set: 1

Select base solids. (Press Enter)

Select detail objects or detail sub entities (optional).

Four geometry-specific versions of this command are available:

bimPropagateEdges command propagates along the edges of planar solids, such as with railings, gutters, borders, and wall caps.

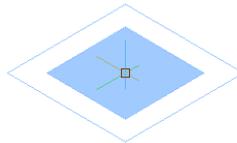
bimPropagateLinear command propagates connections to linear elements, such as beams, columns, pipes, ducts, and connections to walls and slabs.

bimPropagatePattern command propagates a single element on flat surfaces to multiple locations and grids, such as with lights, light switches, windows, and air diffusers.

bimPropagatePlanar command propagates connections to planar elements, such as walls, slabs, and roofs.

bimQuickDraw command draws rectangles and L-shapes with height for conceptually designing buildings and room layouts.

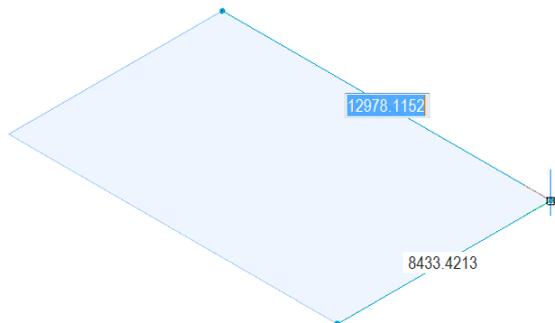
1. When you start the command, you see this initial square:



Starting the bimQuickDraw command

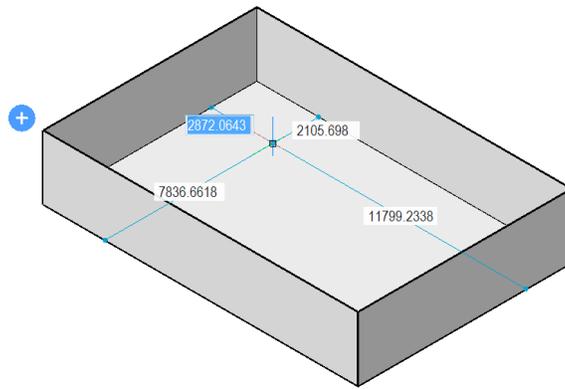
The blue square represents the floor area, the white outline are the walls.

2. As you move the cursor, the square elongates:



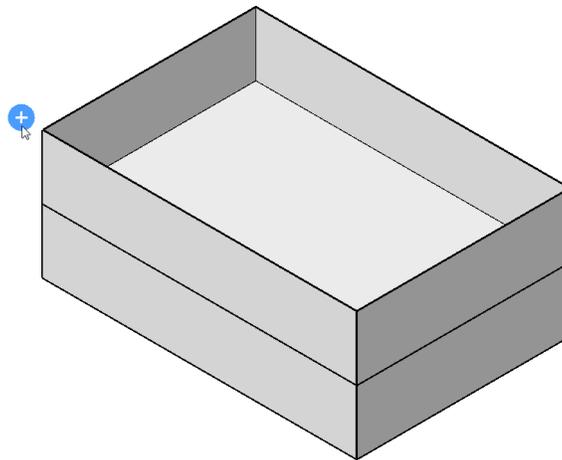
Moving the cursor to set the size of the room

3. When you click a point to indicate the opposite corner (and the size of the floor), walls appear. The thickness and height of the walls are fixed at 1/4" (5mm) wide and 10' (3m) tall.



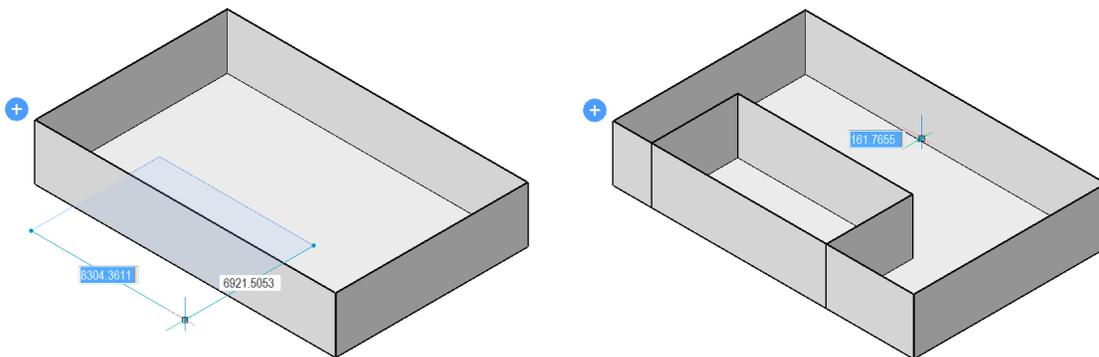
Defining a room with two points

4. Click the blue + to add stories to the floor. You cannot subtract stories while this command is active.



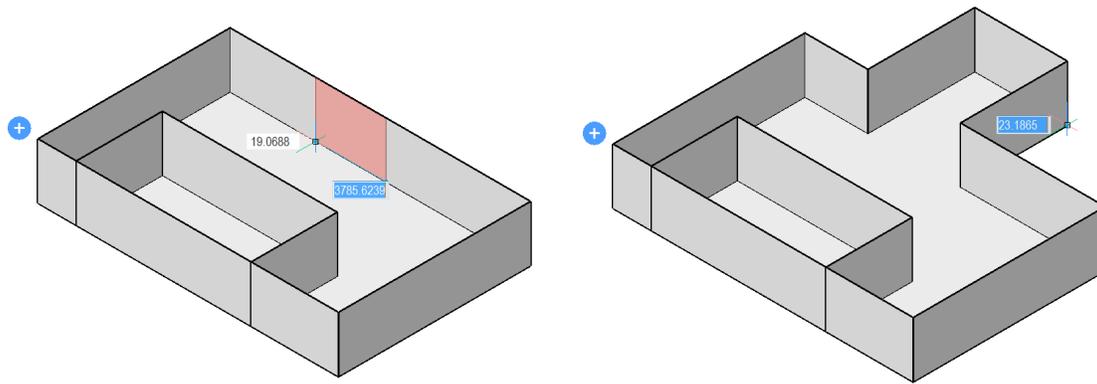
Adding stories by clicking the + button

5. Draw additional attached rooms by starting next to the existing ones.



Left: Drawing more rooms; **right:** ...attached to the first one

6. Cut out a wall by clicking at the base of a wall. The red section hints at the area that will be cut out.



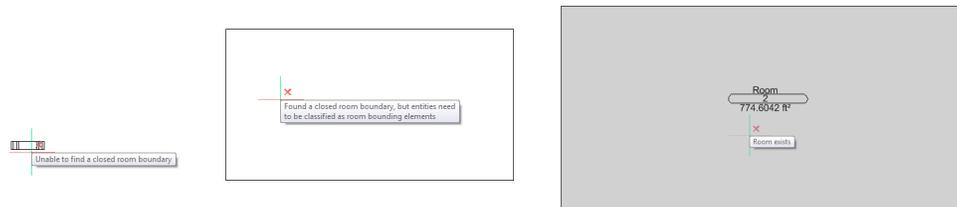
Left: Starting to cut a wall; right:...and adding an L

TIP During the **bimQuickDraw** command, hold down the **Alt** key to move around the model.

bimRecalculateAxis recalculates the axes of structural elements back to their centroids.

bimRepositionWindow command removed from V19 and replaced by editable dimensions to neighbouring edges.

bimRoom command now displays an icon at the cursor to indicate whether a room can be created at the cursor position; if not, the tooltip explains why.



Left to right: No room detected; rectangle not yet a room element; room detected

Rooms can now be calculated on the center lines of walls, curtain walls and columns, and configured for each of bounding element individually.

bimRoomBoundingElements determines which elements (walls, floors, etc) determine bounds of rooms.

bimSchedule command generates schedule tables after analyzing elements in BIM models; schedules update when the model changes.

bimSection command now creates reflected ceilings with the new **Reflected ceiling** option:

```
: BIMSECTION
Select a point to place section or [Detail/Interior/Scale/Reflected ceiling]: r
Select a point to place section: (Pick a point)
Specify distance: (Pick a point)
```

bimSectionUpdate command now generates grid curves and labels on 2D drawings; boundary lines of envelope solids; and story indicator lines and symbols showing story names and elevations.

bimSetReferenceFace command controls the layout of plies by selecting a reference and an opposing face to control ply layout; the reference and opposing face can be non-parallel when the composition has variable thickness.

: **BIMSETREFERENCEFACE**

Select reference face: *(Pick a face)*

Opposing face(s) are detected [Select manually] <Accept>: *(Press Enter)*

TIP For non-parallel faces, first the fixed-thickness plies are set out starting from each reference face, and the remainder of the solid, which does not have a variable thickness, is filled by the variable ply.

bimStructuralConnect connects linear solids.

bimSuggest is removed from V19.

bimTag command reads mappings between BIM types and tag styles from the *_TagTypeToStyle.csv* file; improves automatic placement of tags; supports more properties, including native properties and quantities; and uses a new syntax for tag attribute names 'property category'/'property name'

bimWindowArray removed from V19; replace by new capabilities in the Array command.

bimWindowCreate command now works with a grid that defines subdivisions of the window.

WHAT'S NEW IN COMMUNICATOR

(Requires a separate license and is updated independently of BricsCAD.)

BricsCAD V19 is not compatible with Communicator V18, an upgrade to Communicator V19 is required. Communicator launches as a separate process to increase stability of it and BricsCAD.

InsUnitsScaling variable is supported on import and export. It changes the names of all unacceptable incoming symbols to those that can be stored in the DWG file format; similarly, changes the names of hidden blocks and complex mechanical structures upon exporting.

ImportProductStructure variable's default value is now set to

```
2 Mechanical components for Platinum licenses
.....
1 Blocks for non-Platinum licenses
.....
```

ExportStructure variable's default value is set to 1 (mechanical structure) by default, but is considered as blocks for non-Platinum licenses.

WHAT'S NEW IN IMPORT & EXPORT COMMANDS

DgnImport command imports DGN (design) files created by Microstation from Bentley Systems into the current drawing.

TIP Microstation uses the word “elements” for entities or objects.

The following system variables determine how the design files are imported:

DgnImp2dClosedBSplineCurveImportMode variable determines how to convert closed 2D b-splines:

- 0 Convert to spline (default)
- 1 Convert to region

DgnImp2dEllipseImportMode2D variable determines how to convert ellipses:

- 0 Convert to ellipse (default)
- 1 Convert to region

DgnImp2dShapeImportMode variable determines how to convert 2D shapes and 2D complex shapes:

- 0 Convert to polyline (default)
- 1 Convert to region
- 2 Convert to polyface mesh

TIP Microstation uses the words “shape” and “complex shape” for polyline elements. If an element (entity) is filled, then a hatch is created.

DgnImp3dClosedBSplineCurveImportMode variable determines how to convert closed 3D b-spline curves.

- 0 Convert to spline
- 1 Convert to region (default)

DgnImp3dEllipseImportMode variable determines how to convert 3D ellipses:

- 0 Convert to ellipse (default)
- 1 Convert to region

DgnImp3dObjectImportMode variable determines how to convert 3D elements:

- 0 Convert to polyface mesh (default)
- 1 Convert to a 3D solid or body

DgnImp3dShapeImportMode variable determines how to convert 3D shapes and 3D complex shapes:

- 0 Convert to polyline
- 1 Convert to region (default)
- 2 Convert to polyface mesh

DgnImpBreakDimensionAssociation variable determines if dimensions lose their associativity upon import:

- 0 Associativity is maintained (default)
- 1 Dimension associations are broken

DgnImpConvertDgnColorIndicesToTrueColors variable determines how Microstation colors are mapped to BricsCAD colors:

- 0 Converts DGN color indices to DWG color indices
- 1 Converts DGN color indices to RGB true colors (default)

TIP Microstation assigns color numbers to different colors from BricsCAD, and so BricsCAD gets the color from the DGN color table and then attempts to match it with a color in the DWG color table. If no match is found, the DGN color is saved as an RGB (true color index) value.

DgnImpConvertEmptyDataFieldsToSpaces variable determines how empty field values are handled:

- 0 Replaced by underscore symbols (_)
- 1 Replaced by space symbols () (default)

DgnImpEraseUnusedResources variable determines how to import unreferenced elements, such as text styles and linetypes:

- 0 Import unreferenced elements (default)
- 1 Erase unreferenced imported items

DgnImpExplodeTextNodes variable determines how to handle text nodes (empty text fields):

- 0 Convert them to multiline text (default)
- 1 Convert them to simple entities, such as text and lines

TIP Microstation uses “text nodes” as empty fields used to reserve space for text that will be added later, and so the length of the text is not yet known.

DgnImpImportActiveModelToModelSpace variable determines how active spaces are handled:

- 0 Import active models to model space first, and then attach the design model from model table
- 1 Import the active model to model space (default)

TIP Microstation uses the phrase “design model” for model space, and “active model” for the current view of a model.

DgnImpImportInvisibleElements variable determines how invisible elements (entities) are handled:

- 0 Skip invisible DGN elements
- 1 Import invisible them as invisible entities (default)

DgnImpImportPaperSpaceModels variable determines how to import sheet models (paper space):

- 0 Import one DGN model to model space only
- 1 Import all DGN sheet models to paper space layouts (default)

TIP Microstation uses the phrase “sheet model” for paper space.

DgnImpImportViewIndex variable determines DGN view settings:

- 0 - 7 Specifies level mask
- 1 View is not defined

TIP Microstation uses the word “level” for layers; a mask hides content in areas or levels.

DgnImpRecomputeDimensionsAfterImport variable determines how to handle dimensions:

- 0 Create DGN-style dimension geometry blocks (default)
- 1 Re-compute all dimensions to create DWG dimension geometry blocks

DgnImpSymbolResourceFiles variable specifies the paths to folders holding DGN and RSC files.

TIP Microstation uses RSC resource files to store fonts, line styles, and so on.

DgnImpXRefImportMode variable determines how to import reference attachments:

- 0 Don't import DGN reference attachments
- 1 Convert attached DGN files to DWG-style xref files
- 2 Create a block definition of the attached DGN file, then create a block reference (default)
- 3 Attach the DGN references as an underlay

TIP Microstation uses the word “cell” for blocks.

Export command now creates 24-bit-per-pixel BMP files instead of 8-bit-per-pixel ones; as well, it records log files in the folder of exported IFC files; beams and columns with library profiles are exported with the profile name.

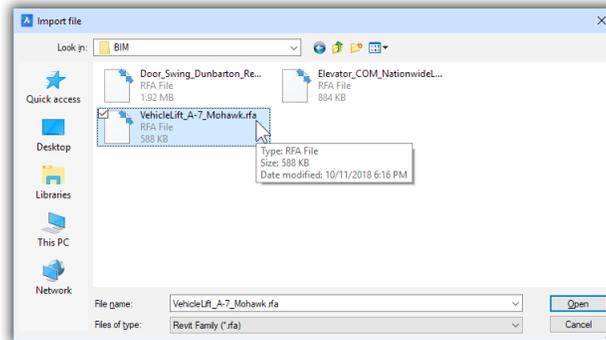
FbxExport command exports drawings as FBX files (short for “filmbox”):

```
: FBXEXPORT
What entities to export? [Selected/Visible] <Visible>:
Export options. Entities: yes, lights: yes, cameras: yes, materials: yes
What types to export? [Select/All] <Select>:
Export entities? [Yes/No] <Yes>:
Export lights? [Yes/No] <Yes>:
Export cameras? [Yes/No] <Yes>:
Export materials? [Yes/No] <Yes>:
Export options. Entities: yes, lights: yes, cameras: yes, materials: yes
How to export textures? [Embedded/Reference to file/Copy of file] <Embedded>:
Enter path to export fbx file: [C:\Program Files\Bricsys\BricsCAD V19 en_US\Drawing2.fbx]:
Export of 'C:\Program Files\Bricsys\BricsCAD V19 en_US\Drawing2.fbx' succeeded: 10 entities, 1
lights, 2 cameras, 6 materials
```

TIP The FBX format was invented by Kaydara and is now maintained by Autodesk. It is used to transport 3D models with light, camera, and material data to rendering software like 3ds Max and Blender.

-FbxExport command does the same thing.

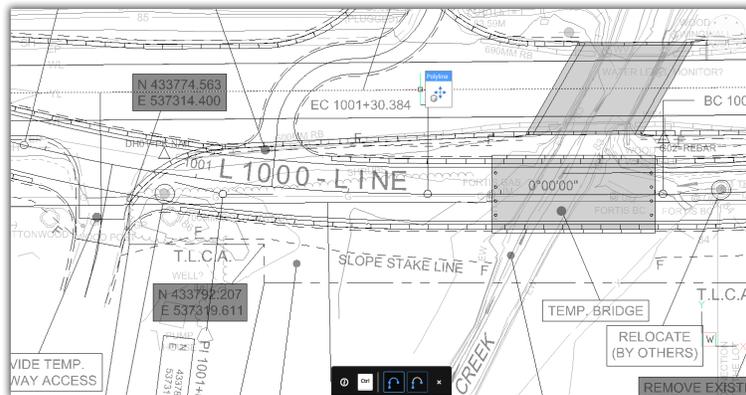
Import command now imports Microstation DGN and Revit Family RFA files into new drawings; IFC profile definitions are now imported to the profile library. (See DgnImport command.)



Import dialog box selecting an RFA file to import

TIP An RFA file holds Revit families, which are like parametric parts but in a Revit format. This command opens RFA files in a new drawing. To insert an RFA file as a component (block) into an existing drawing, use the **bmlnsert** command.

PDFimport command converts vector data in PDF files to DWG entities, primarily polylines.



PDF imported as polyline entities

: PDFIMPORT

Select pdf underlay or [File] <File>: (Press Enter to access the Select PDF File dialog box)

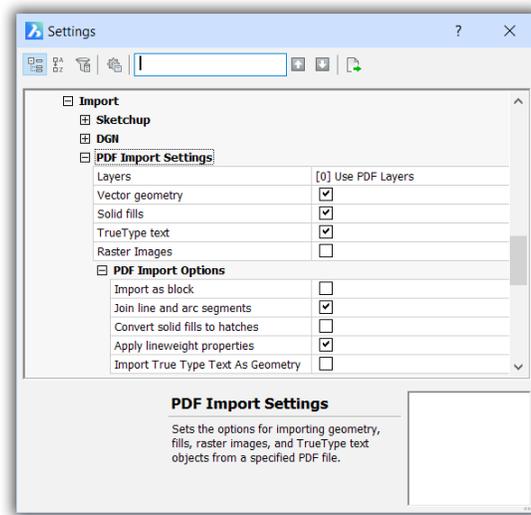
Specify page number to import or list pages [? to list the pages / Settings] <1>: (Press S for settings, or else enter a page number)

Specify insertion point <0, 0>: (Press Enter)

Specify scale <1.>: (Press Enter)

Specify rotation <0.>: (Press Enter)

TIP You may need to use the **Zoom Extents** command to see the entire imported PDF file.



Settings dialog box for changing PDF import options

PdfImportApplyLineweight variable toggles whether BricsCAD honors the lineweight values in the PDF file:

- 0 Ignore lineweights in the PDF file
- 1 Honor the lineweight values (default)

PdfImportAsBlock variable toggles whether the import PDF is stored as a block:

- 0 Insert PDF file as individual entities (default)
- 1 Insert PDF file as a block

PdfImportConvertSolidsToHatches variable toggles whether to convert areas filled with solid colors in the PDF file to solid-filled hatches:

- 0 Leave solid-filled areas as-is (default)
- 1 Convert solid-filled areas into solid-color hatches

PdfImportJoinLineAndArcSegments variable toggles whether to join individual lines and arcs into polylines:

- 0 Leave lines and arcs as individual segments; entities in PDFs file already joined imported as polylines
- 1 Join lines and arcs into polylines (default)

PdfImportLayersUseType variable determines how layers in PDF files are handled:

- 0 Import layer names from PDF file and create matching layer names in the drawing
- 1 Create layers in the drawing for each entity type, such as (default)
- 2 Place all PDF entities on the current layer in the drawing

TIP BricsCAD creates the following layers to segregate PDF content:

- PDF_Geometry for vector content
- PDF_Images for raster content
- PDF_SolidFills for solid-filled areas
- PDF_Text for TrueType text

PdfImportRasterImages variable toggles whether to import raster images from the PDF file:

- 0 Don't import raster content
- 1 Import raster content and place as a raster image in the drawing

PdfImportSolidFills variable toggles how to handle solid filled areas:

- 0 Ignore solid-filled areas in the PDF file
- 1 Import solid-filled areas and convert them to solid-color hatches (default)

TIP Solid filled areas are automatically given a 50% transparency level by BricsCAD so that underlying content can be seen.

If the PDF file was generated by an AutoCAD workalike, then solid filled areas include the following:

- 2D solids
- Arrowheads with width (such as from dimensions and leaders)
- Hatched areas filled with solid colors
- Polylines with width
- Wipeouts

PdfImportTrueTypeText variable toggles whether text made from TrueType fonts are imported:

- 0 Does not import text
- 1 Import TrueType text as mtext, and create a text style named after the font (default)

TIP The text style created for imported TrueType fonts has a "PDF" prefix to the name, such as "PDF Arial."

PdfImportTrueTypeTextAsGeometry variable determines how text made of TrueType fonts is imported:

- 0 Convert text to TrueType text in the drawing (default)
- 1 Convert text to equivalent entities in the drawing

PdfImportVectorGeometry variable toggles whether vector geometry is imported:

- 0 Don't import vector geometry
- 1 Import vector geometry from the PDF file and convert them to the nearest DWG entities (default)

TIP Most vector geometry is converted to polylines; geometry that is similar enough to arcs, circles, and ellipses are converted to such. Solid filled areas are converted to solid-filled hatches.

WHAT'S NEW IN GENERATED VIEWS

GenerateAssocViews variable (when on) no longer generates views from entities on layers that are off, hidden, or frozen to improve the performance of **ViewBase** and **bimSectionUpdate** commands; associative data are not set on switched off, hidden or frozen layers.

ResetAssocViews command now removes associative data from nested blocks.

ViewBase command now processes 3dFace and PolyFaceMesh entities, and creates exploded 2D representations of 3D assemblies through the new **Special views** option:

: **VIEWBASE**

Preset: "None", View scale: "Adapt to paper size"

Select objects or [Entire model/presentation/**Special views**] <Entire model>: **s**

Select view [Exploded view/Back] <Back>: **e**

A warning is displayed to emphasize that this command meant for mechanical drawings, not BIM ones; the warning can be disabled.



Warning dialog box from ViewBase

ViewExport command now respects hidden and tangent lines.

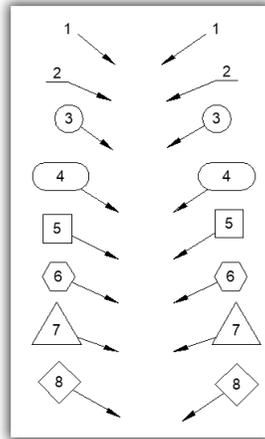
WHAT'S NEW IN MODELING

(bm- commands are available in the Platinum edition of BricsCAD only)

bmBalloon command adds the **Auto mode** option to place balloons on all components automatically in a specific drawing view; uses user-defined styles and predefined balloon styles from *_BalloonStyles.dwg*.

: **BMBALLOON**

Select a component insert [select Table/select Style/**Auto mode**]:



Default balloon styles included with BricsCAD

bmBOM command adds a BOM (bills of material) status parameter to determine whether components and their subcomponents are included in BOM tables.

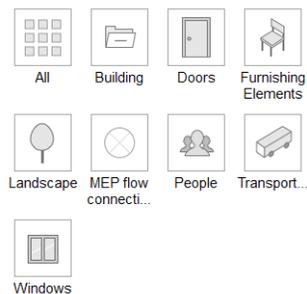
bmBrowser removed from V19, replaced by the **ComponentsPanelOpen** command.

-bmCreateComponent command adds entities in the drawing to the Components panel/library:

: **-BMCREATECOMPONENT**

Select entities to create component from or use <Drawing>: *(Select one or more entities, or press Enter to select the entire drawing)*

Specify component category: [1 - All/ 2 - Building/ 3 - Doors/ 4 - Furnishing elements/ 5 - Landscape/ 6 - Mep flow connection points/ 7 - People/ 8 - Transportation/ 9 - Windows/ 0 - Add new category/] <Current>: *(Choose a category to which to add the component)*



The default component categories

bmExplode command now edits exploded representation and generates 2D drafting views; exploded representations can be edited in the Mechanical Browser by adding, deleting, and reordering components. It gains a [Linear](#) option to explode assemblies linearly in a given direction, taking into account collisions between components. It animates either selected steps or the entire sequence.

: BMEXPLODE

Select explosion algorithm [Table by types/[Linear](#)/Manual/Settings] <Manual>: *(Enter an option)*

bmExplodeMove command creates user-defined exploded representations.

: BMEXPLODEMOVE

Select entities to explode [Entire model] <Entire model>: *(Press Enter)*

Select base part <None>: *(Select an entity)*

Entities in set: 1

Select base part <None>: *(Press Enter to continue)*

Select axial entity or define axis by [2Points/Xaxis/Yaxis/Zaxis] <2Points>: **2p**

Specify start point of axis <0,0,0>: *(Press Enter)*

Specify end point of axis or <use axis parallel to view direction>: *(Pick a point)*

bmInsert command now inserts Revit Family RFA files as components; it also creates parametric components, and supports window insertions similar to the **bimInsert** command.

IP To open RFA files in a new drawing, use the **Import** command.

bmLispGet now retrieves variables for blocks and parameters of components.

-bmParameters command now edits associative arrays of parametric components and can be assigned to an expression using this command, or Mechanical Browser or Properties panel.

bmReplace command now prompts with [from File](#) to select a replacement component, instead of opening the File dialog box; adds the [Parameters changes](#) option to control which parameter values are to be used after replacement — reapply changes of parameters for the component being replaced or use replacement as-is.

: BMREPLACE

Select component inserts to be replaced: *(Select one or more components)*

Select component inserts to be replaced: *(Press Enter to continue)*

Select component insert to use as a replacement [SIMilar inserts/component Type/ [Parameter changes/from File](#)] <from File>: *(Enter an option)*

bmUnlink breaks links between components.

WHAT'S NEW IN POINT CLOUDS

Point clouds are a new entity type in BricsCAD V19: `AcDbPointCloudEx`.

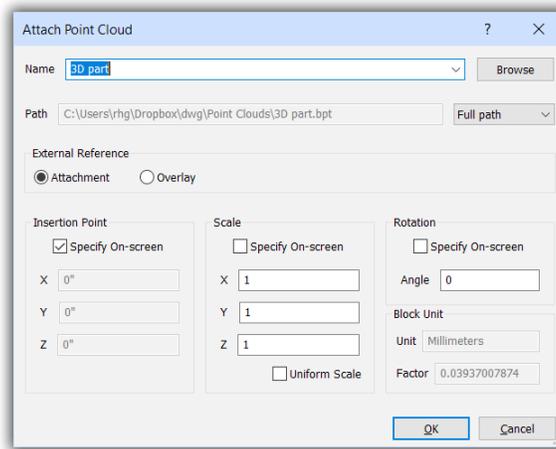
Attaching a point cloud file takes two steps:

1. Run the **PointCloudPreprocess** command to convert PTS, PTX, and LAS files to Bricsys' own BPT format.
2. Run the **PointCloudAttach** command to attach the BPC file to the current drawing.

PointCloud displays the Point Cloud section of the Drawing Explorer:

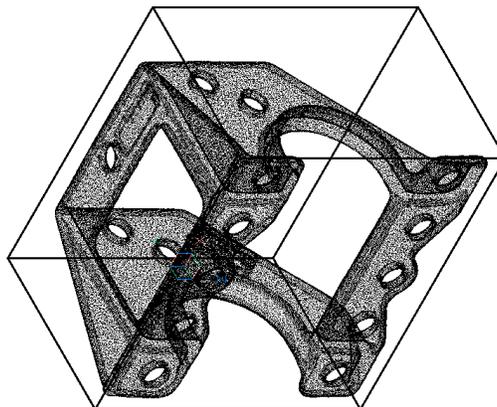
PointCloudAttach command opens a BPT point cloud file, and then attaches it to the current drawing.

: POINTCLOUDATTACH



Specifying the attachment settings

Set point cloud UNITS (0..20), default is to use current INSUNITS value <1>: (Press Enter)
SUCCEEDED to attach Bricsys point tree C:\Users\rhg\Dropbox\dwg\Point Clouds\3D part.bpt!!

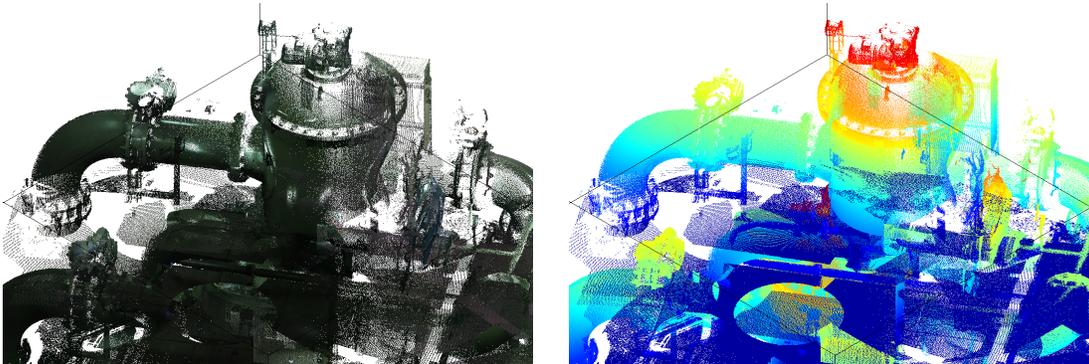


Point cloud displayed inside its bounding box

PointcloudColorMap command displays the Point Cloud Color Map dialog box to set colors based on Intensity, Elevation, and Classification of the point cloud.

: POINTCLOUDCOLORMAP

Color map [Scan/Jet/Earth/Hydro/Gray] <Scan>: (Enter an option)



Left: Color of points set to "scan"; right: color set to color "jet"

The options of the PointcloudColorMap command have the following meaning:

Scan	Display points by the colors specifies in the source laser capture file
Jet	Display points in a range of colors from red through green to blue
Earth	Display points in colors ranging from orange to brown
Hydro	Display points in colors ranging from light to dark blue
Gray	Display points in colors ranging from black to light gray

PointcloudPointSize variable sets the size of new point cloud objects; range is from 1 to 10 pixels; default is 1 pixel.

PointcloudPointSize_Minus command decreases the size of points.

: POINTCLOUDPOINTSIZEMINUS

Decreased point size to 0.500000

PointcloudPointSize_Plus command increases the size of points.

: POINTCLOUDPOINTSIZEPPLUS

Increased point size to 1.000000

PointcloudPreprocess command converts raw point cloud data from PTS, PTX, and LAS files to the optimized BPT format, which can be attached to a drawing.

: POINTCLOUDPREPROCESS

The command opens the Point Cloud Data File dialog box; choose one of the recognized formats, and then save it in BPT format in the Give Name for Preprocessing Output File dialog box.

BPT	BricsCAD point cloud format (compressed)
LAS	LASer (lidar) file format
PTS	Leica PoinTS cloud format (x,y,z points only)
PTX	Enhanced Leica point cloud format (with luminosity)

TIP Point cloud data always consists of an x,y,z-coordinates of each scanned point, and sometimes the color and light intensity — usually in simple ASCII format.

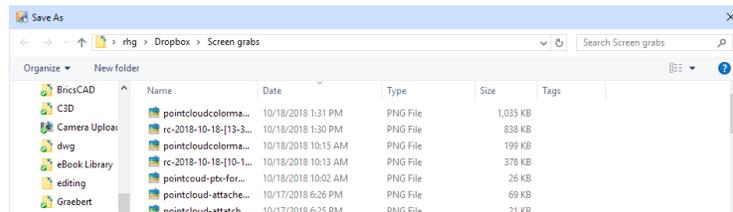
Next, the command prompts you as follows:

Start preprocessing of raw point cloud data?

This is a necessary step, but will take ~1 minute per million points. [Yes/No] <Yes>: (Press Enter)

Preprocessing data: C:\Users\rhg\Dropbox\dwg\Point Clouds\3D part.bpt -> This will take approx 1 minute per million points source data.

The processing is done in the background, and may take several minutes or hours, depending on the number of points in the file. In the meantime, you can do other things in BricsCAD. When the processing done, BricsCAD displays the following dialog box:



Being notified that the processing is complete

WHAT'S NEW IN SHEET METAL

(sm- commands require a separate license)

Smart feature technology rebuilds relief features and geometry upon local changes, when cut is no longer needed for the new design. Also it makes edges, imprinted on thickness faces, resistant to local simplifying operations.

The **Settings** option is removed from sm- commands and relocated to the Sheet Metal group in the Settings dialog box.

Quad menu updates the “Sheet Metal” section to provide the complete set of sheet metal tools.

smAssemblyExport command now recognizes single parts, and skips entities on layers ‘bim_subtract’, ‘bim_unite’, ‘bc_subtract’, and ‘bc_unite’.

smConvert command now recognizes straight holes as a special type of form feature, as well as form features not smoothly connected with flanges. It uses the Sharp Bend Radius setting to distinguish regular bends from lofted bends, and defillets recognized control curves for rib features.

smExportOsm command now exports straight holes and non-sheet metal solids as inserts within selected sheet metal parts.

smFlangeConnect command in some cases no longer needs to create corner reliefs

smFlangeEdge command now creates miters and junctions automatically for a number of configurations, and eliminates self-intersections.

smFlangeSplit command is removed from V19; replaced by the **smSplit** command.

smFlip command removes some options from the prompt and adds the [Flip reference side only](#) option:

: SMFLIP

Old prompt:

```
Select solids, flange faces and allow rebuild geometry [Disallow rebuild geometry/Entire model]
<Entire model>:
```

New prompt:

```
Select solids or flange faces to flip flange(s) [Flip reference side only/Entire model] <Entire
model>:
```

smJunctionCreate command now produces junction cuts on curved hard edges, including elliptical and spline curves, and assigns junction features to cutouts solely created on linear hard edges.

smLoft command now converts lofted bends to ordinary bends, if possible.

smReliefCreate command is more accurate near non-sharp, large-radius bends by minimizing the cut to link better with junctions on curved edges.

: SMRELIEFCREATE

Select a hard edge, bend face, flange face or 3D solid [Entire model]:

Enter relief size by bend radius ratio or [force Bend reliefs/Auto] <Auto>:

smReliefSwitch command now manually selects the faces to be converted to smooth bend reliefs.

: SMRELIEFSWITCH

Select faces, 3d solids to switch reliefs to V-type [RECTangular/CIRcular/V-type/SMooth/RIP/ROund/relief EXTension/Entire model] <Entire model>:

smRibCreate command now uses a fillet radius as a parameter; ribs can be exploded into flanges, bends, and other features.

smRepair command now repairs flange faces locally.

smParametrize command now recognizes straight hole rectangular arrays.

smSelect command get a new option **Select non-orthogonal geometry** option select faces of the same thickness but are not orthogonal.

: SMSELECT

Select the option [Hard edges/Same form features/Similar form features/Non-orthogonal thickness faces]:

smSplit command combines the smFlangeSplit command and the new Lofted Bend Face functions to split lofted bends.

: SMSPLIT

Select flange or lofted bend face: (Pick a flange or Lofted blend feature)

Select lines or edges to split the flange or [use SMART split/draw a New line] <draw a New line>: (Enter an option)

smTargetCAM variable specifies the target cams to be used by smUnfold.

smUnfold command now has associativity between the 3D model and its 2D unfolded model; also, it does not change the properties of layers that exist.

TIP When the layer has to be created, the settings of the various unfolding variables are used.

WHAT'S NEW IN APIS

BricsCAD V19 is compiled with Visual Studio 2017. To be compatible, C++ extension DLLs must be compiled with the same platform toolset.

PDF-related commands now use the Google's free PDFium engine.

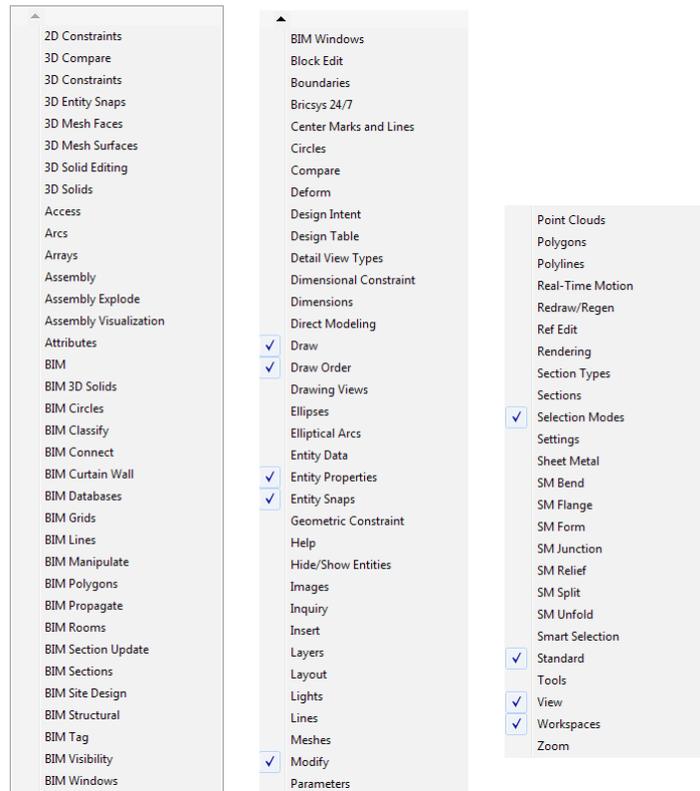
BLADE for interactive LISP development has been updated.

New entity types in V19:

- ▶ Point clouds are entity type 'AcDbPointCloudEx'
- ▶ Terrain surfaces are entity type 'TIN Surface' (TIN is short for triangular irregular network)

Many, Many More Toolbars

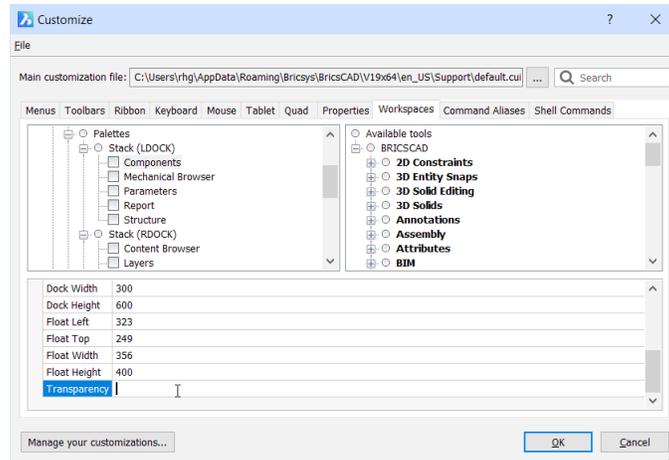
Even though the emphasis by BricsCAD is on the ribbon and command bar, V19 adds a host of new toolbars, especially in the areas of BIM and sheetmetal. Here is what the list of toolbar names looks like:



As well, the ribbon is updated in some areas with new commands.

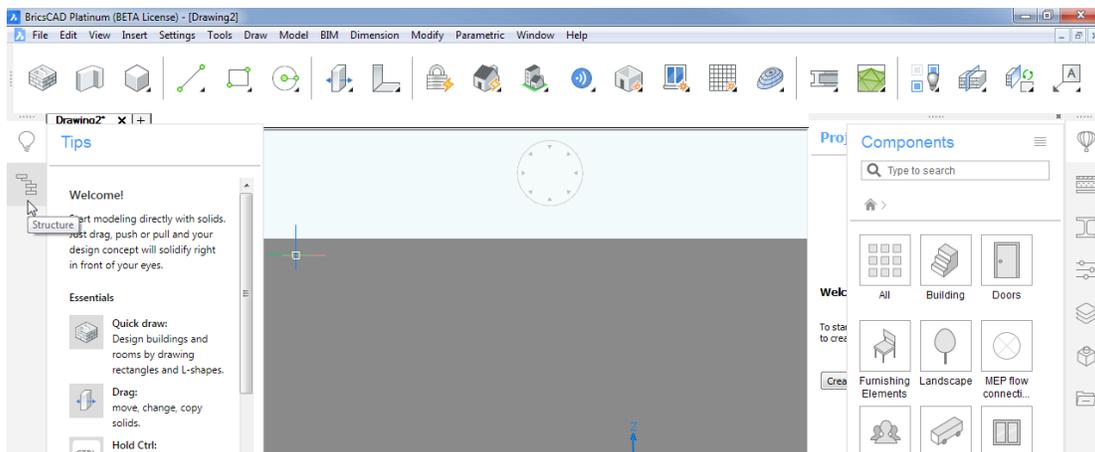
Customize Dialog box

Floating panels can now be transparent. This is controlled by the new Transparency property. To access it, open the Customize dialog box, choose Workspace tab, choose the name of a panel, and then in the Properties pane change the value of Transparency.



Setting the transparency level of the Components panel

Panels can be collected into “stacks” on the left or right sides of the drawing area.



Panel stacks along the left and right edges of BricsCAD

The location is controlled by new properties in the Customize dialog box:

- Stack ID** Determines the panel icon location:
 - LDOCK** Appears in the left stack
 - RDOCK** Appears in the right stack
- Stack Z Order** Determines the order in which the icon appears in the stack

ID	COMPONENTS
Title	Components
Display	Hide
State	Floating
Stack ID	Ldock
Stack Z Order	1
Dock Column	0
Dock Row	0

Changing properties for panel stacks

TIP If the stack docks do not appear, you need to erase settings from an earlier release of BricsCAD V19. Exit BricsCAD, delete the %AppData%\Roaming\Bricsys\V19x64 folder, and then restart V19, which will rebuild the folder correctly.

WHAT'S NEW IN LICENSES

The following commands are removed from one edition and moved to another:

Command	From Edition	To Edition
bmBalloon	Platinum	Mechanical
bmBom	Platinum	Mechanical
bmExplode	Platinum	Mechanical
bmMassProp	Platinum	Mechanical

The following editions of BricsCAD V19 are available for licensing:

BricsCAD V19 Standard

BricsCAD V19 Pro license is required for

- › Rendering
- › ACIS-based 3D modeling
- › Drawing view generation

BricsCAD V19 Platinum license is required for

- › 3D constraints
- › Automatic 3D solids parametrization
- › Design intent
- › Parametric components and arrays
- › Deformable 3D modeling
- › Search for differences between 3D models

BricsCAD V19 BIM license is required for

- › BIM design
- › Includes BricsCAD Platinum

BricsCAD V19 Mechanical (formerly known as Sheet Metal) license is required for

- › Configurable bill of materials
- › Automatic balloon placement
- › Mass properties of assemblies
- › Automatic exploded views of assemblies and animation
- › Sheet metal design from scratch and reworking sheet metal parts created in other CAD systems
- › Simultaneous editing of unfolded and folded sheet metal views
- › Includes BricsCAD Platinum

BricsCAD V19 Ultimate Suite

- › Contains all modules, except for Communicator

Communicator V19 license is required for

- › MCAD file translation