Copyright Information

Copyright © 2019 by upFront.eZine Publishing, Ltd.
All rights reserved worldwide.
This 10th edition is based on BricsCAD V20
6 December 2019

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.

Brief Contents

Tutorials
1. A Quick Tour Through BricsCAD ................................................................. 1
2. Navigating the BricsCAD Interface ............................................................ 23
3. Setting Up A New Drawing ........................................................................ 49
4. Creating Your First Drawing .................................................................... 77
5. Adding Details to Drawings ..................................................................... 105
6. Making Changes to Drawings .................................................................. 139
7. Adding Notes and Dimensions ................................................................ 159
8. Bills of Material ......................................................................................... 195
9. Working with 2D Regions & Booleans ...................................................... 229
10. 3D Modeling & Direct Editing ................................................................. 249
11. Dimensional & Geometric Constraints .................................................. 291

Appendices
A. Concise Summary of Command Names .................................................. 307
B. Concise Summary of Variables & Settings .............................................. 333
C. What's New in BricsCAD V20 ................................................................. 359
# Table of Contents

1. **A Quick Tour Through BricsCAD** ..................................................... 1

## How to Start BricsCAD ............................................................................... 3
- Starting BricsCAD on Windows .......................................................... 3
  - Windows Vista and 7 ........................................................................ 3
  - Windows 8 .................................................................................... 3
  - Windows 10 .................................................................................. 3
- Starting BricsCAD on Linux ..................................................................... 4
- Starting BricsCAD on Mac MacOS .......................................................... 5

## BricsCAD V20 User Interface ..................................................................... 6

## The Launcher and the Start Tab ................................................................. 8
- The Start Tab .................................................................................... 9
- The BricsCAD Window ....................................................................... 10
- A Basic Tour of the User Interface ...................................................... 10
  - Crosshair and Arrow Cursors ...................................................... 11
- Command Bar .................................................................................. 12
  - All About Command Prompts ..................................................... 13
  - Handling Options Efficiently ...................................................... 15
  - Undoing What You’ve Done: U .................................................. 15
  - Seeing What You Did Before: Command History ................... 16
  - Typing Less: Aliases and AutoComplete ................................ 17

## SUPPORTED GRAPHICS BOARDS .......................................................... 20

## UCS Icon ............................................................................................. 20

## Online Help .......................................................................................... 21

## Exiting BricsCAD .................................................................................. 22
2. Navigating the BricsCAD Interface

IN THIS CHAPTER

Above the Drawing Area
Title Bar
Quick Access Toolbar
Ribbon Tabs and Panels
Drawing (Document) Tabs

In the Drawing Area
Look-From Control
The UCS Icon
Hotkey Assistant
Dynamic Input
Shortcut Menus
quad cursor
Quick Properties
Scroll Bars
Prompt Menu
Manipulator Widget
Clean Screen
Panels

Below the Drawing Area
Layout Tabs
Status Bar

3. Setting Up A New Drawing

Before You Begin
Starting A New Drawing
HOW TO START NEW DRAWINGS
Default Settings
Selecting the Linear Units
Selecting the Angle Style
Choosing the Plot Style
Setting Entity Properties
Finishing the Wizard
Additional Important Settings ................................................................................................. 61
  Setting Drawing Limits ........................................................................................................ 61
  Accessing and Changing Settings ....................................................................................... 63
  Changing the Snap and Grid Spacing ................................................................................ 65

Creating Layers ..................................................................................................................... 67
  QUICK SUMMARY OF LAYERS .......................................................................................... 68
  Naming Layers .................................................................................................................... 69
  QUICK SUMMARY OF THE SAVEAS COMMAND ............................................................. 72

Saving Drawings .................................................................................................................... 73
  QUICK SUMMARY OF ALTERNATIVE SAVE FORMATS .................................................. 73
  Making Backups Automatic ............................................................................................... 75

4. Creating Your First Drawing .............................................................................................. 77

  Reopening Drawings .......................................................................................................... 79
  Drawing the Lot’s Boundary ............................................................................................... 80
  Planning the Next Steps ..................................................................................................... 83
    Changing Layers ................................................................................................................ 83
  Drawing the Outline of the House ...................................................................................... 85
    Direct Distance Entry ....................................................................................................... 87
      QUICK SUMMARY OF THE PLINE COMMAND ............................................................ 87
      QUICK SUMMARY OF ENTITY SNAP MODES ............................................................... 88
    Dynamic Input ................................................................................................................... 89
  Moving the House into Position ......................................................................................... 91
  Starting on the Driveway ...................................................................................................... 93
    Finishing the Driveway ..................................................................................................... 94
    Mirroring Entities ............................................................................................................ 95

Putting Drawings to Paper .................................................................................................... 96
  QUICK SUMMARY OF PRINT PREVIEW ......................................................................... 98
  Solving Printing Problems ................................................................................................. 100
  Smudgy Prints with Bad Colors with Inkjets .................................................................... 101
  Generating PDFs of Drawings ............................................................................................ 102
  Specifying PDF Output Options ....................................................................................... 102
  What PDF Options Mean ................................................................................................... 103
5. Adding Details to Drawings ........................................ 105

Dividing the Lot ................................................................. 107
  Smoothing Polylines .......................................................... 110
  QUICK SUMMARY OF THE PEDIT COMMAND ..................... 111

Grips Editing ...................................................................... 112
  QUICK SUMMARY OF EDITING WITH GRIPS ....................... 114
  Editing by Double-clicking .................................................. 116
  Hatching the Lawn ............................................................ 117
  Modifying Hatching ............................................................ 119
  QUICK SUMMARY OF DRAWING CIRCLES ............................ 120

Creating Symbols (Blocks) .............................................. 120
  Drawing Circles ............................................................... 121
  Zooming in Real Time ......................................................... 121
  Creating Arrays ................................................................. 122

Making Blocks .................................................................... 125
  Adding Many More Trees (Insert) ........................................ 126

Drawing the Pond ............................................................. 128
  QUICK SUMMARY OF DRAWING ELLIPSES ......................... 130
  Array Paths ....................................................................... 130
  Drawing Splines ............................................................... 131
  Zooming to Objects .......................................................... 132
  Paving Stones from Polygons .............................................. 132
  Arraying Along a Path ....................................................... 133
  QUICK SUMMARY OF SPLINE ............................................ 133
  QUICK SUMMARY OF POLYGON ....................................... 134
  QUICK SUMMARY OF ARRAYPATH ..................................... 135
  QUICK SUMMARY OF ARRAYEDIT .................................. 136
  Editing Associative Arrays .................................................. 137

6. Making Changes to Drawings ....................................... 139

Changing the Look of Lines ............................................. 141
  Loading Linetypes ........................................................... 141
  Changing Properties .......................................................... 143
  QUICK SUMMARY OF PROPERTIES PANEL ....................... 144
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing the Linetype Scale</td>
<td>147</td>
</tr>
<tr>
<td>QUICK SUMMARY OF SELECTION SET OPTIONS</td>
<td>148</td>
</tr>
<tr>
<td>Selecting Entities by Other Methods</td>
<td>149</td>
</tr>
<tr>
<td>Selecting Entities by Their Properties</td>
<td>149</td>
</tr>
<tr>
<td>Selecting an Entity by Tabbing</td>
<td>150</td>
</tr>
<tr>
<td>Changing Line Lengths</td>
<td>150</td>
</tr>
<tr>
<td>QUICK SUMMARY OF THE LENGTHEN COMMAND</td>
<td>151</td>
</tr>
<tr>
<td>Stretching the Pond</td>
<td>152</td>
</tr>
<tr>
<td>Moving Entities</td>
<td>153</td>
</tr>
<tr>
<td>Nudging Entities</td>
<td>154</td>
</tr>
<tr>
<td>Adding the Fence</td>
<td>155</td>
</tr>
<tr>
<td>Listing Data</td>
<td>156</td>
</tr>
<tr>
<td>7. Adding Notes and Dimensions</td>
<td>159</td>
</tr>
<tr>
<td>Adding Notes to Drawings</td>
<td>162</td>
</tr>
<tr>
<td>QUICK SUMMARY ON CALCULATING TEXT HEIGHTS</td>
<td>164</td>
</tr>
<tr>
<td>How to Determine the Size of Text</td>
<td>165</td>
</tr>
<tr>
<td>Creating Text Styles</td>
<td>165</td>
</tr>
<tr>
<td>Simple Text</td>
<td>167</td>
</tr>
<tr>
<td>QUICK SUMMARY OF TEXT JUSTIFICATION MODES</td>
<td>168</td>
</tr>
<tr>
<td>Changing Text</td>
<td>168</td>
</tr>
<tr>
<td>Placing Rotated Text</td>
<td>169</td>
</tr>
<tr>
<td>Changing the Angle of Rotated Text</td>
<td>170</td>
</tr>
<tr>
<td>Placing Multiple Lines of Text</td>
<td>170</td>
</tr>
<tr>
<td>Finding and Replacing Text</td>
<td>171</td>
</tr>
<tr>
<td>Dimensions in Drawings</td>
<td>173</td>
</tr>
<tr>
<td>Preparing Drawings for Dimensioning</td>
<td>174</td>
</tr>
<tr>
<td>Dimensioning the Yard</td>
<td>174</td>
</tr>
<tr>
<td>Vertical and Baseline Dimensions</td>
<td>176</td>
</tr>
<tr>
<td>Baseline Dimensioning</td>
<td>177</td>
</tr>
<tr>
<td>Aligned and Radial Dimensions</td>
<td>178</td>
</tr>
<tr>
<td>Radial Dimensions</td>
<td>178</td>
</tr>
<tr>
<td>Dim, the Super Dimension Command</td>
<td>179</td>
</tr>
</tbody>
</table>
### Annotatively-Scaled Text and Dimensions

- **A FEW EXTRA THINGS ABOUT ANNOTATIVE SCALING**
- Using Annotative Scaling

### Leaders and Multileaders

- **Single-line Leaders**
- **QUICK SUMMARY OF QLEADER**
- **Multi-line Leaders**
- **QUICK SUMMARY OF MLEADER**

### 8. Bills of Material

- **About Attribute Data**
- **Creating Blocks with Attributes**
  - **Drawing Blocks**
  - **Defining Attributes**
  - **Adding More Attributes**
  - **Combining Entities and Attributes into Blocks**
- **Inserting Blocks with Attributes**
  - Alternatives to the Insert Command
  - Inserting Blocks with Drag’n Drop
- **Exporting Data from Drawings**
  - Data Extraction
  - Importing Data Files into Spreadsheets
  - **QUICK SUMMARY OF ADDITIONAL FORMATS**
- **Placing Data in Drawings as Tables**
  - Linking Data in Drawings with Tables
  - Editing Datalinks
  - Updating the Table
- **Automatic BOMs from 3D Components**
  - Exploding Assemblies
  - **QUICK SUMMARY OF BMEXPLODE**
  - Generating BOMs
  - **QUICK SUMMARY OF BMBOM**
  - Attaching Balloons
  - **QUICK SUMMARY OF BMBALLOON**
9. Working with 2D Regions & Booleans................................. 229

About Regions ...................................................................................................................... 231
   How to create Regions........................................................................................................ 231
   Differences Between Boundary and Region Commands.................................................... 232
   QUICK SUMMARY OF BOUNDARY OPTIONS.................................................................... 233
   What To Do With the Leftovers......................................................................................... 236
   Tutorial: Creating Boundaries............................................................................................ 236
   QUICK SUMMARY OF BOOLEAN OPERATIONS................................................................. 238

Boolean Operations .............................................................................................................. 240
   Tutorial: Creating a Waffle Shape....................................................................................... 240

Measuring Regions ................................................................................................................ 246
   QUICK SUMMARY OF MASS PROPERTIES......................................................................... 247

10. 3D Modeling & Direct Editing...................................................... 249

About 3D Solid Models ......................................................................................................... 251
   Planning Ahead.................................................................................................................... 251
   ELEMENTS OF 3D MODELS............................................................................................ 252
   Planning Ahead by Deconstructing 3D............................................................................... 253
   Traditional Extrusion Method............................................................................................ 253
   Preparing the Drawing........................................................................................................ 254
   Modeling a Box................................................................................................................... 257
   QUICK SUMMARY OF EXTRUDE COMMAND................................................................. 258
   QUICK SUMMARY OF LOOKFROM COMMAND............................................................... 260
   3D View Rotation................................................................................................................. 260
   Extrusion by Direct Modeling............................................................................................... 261
   Applying Dynamic UCS...................................................................................................... 261
   QUICK SUMMARY OF DUCS ENTITIES.............................................................................. 263
   Extruding Directly with the Quad Cursor............................................................................ 264
   Subtracting Solids To Make Holes..................................................................................... 266
   PushPull Modeling................................................................................................................. 267
   Aligning the UCS................................................................................................................ 270
   Pulling, Instead of Pushing.................................................................................................. 271
   Filleting 3D Solids............................................................................................................... 275
   Joining Parts with Union...................................................................................................... 276
Generating 2D Drawings .............................................................. 278
QUICK SUMMARY OF VIEWBASE OPTIONS................................................. 280
QUICK SUMMARY OF VIEWSECTION OPTIONS.............................................. 282
Adding Section Views And Detailed Views.............................................. 283
Resizing Views ...................................................................................... 283
QUICK SUMMARY OF VIEWDETAIL OPTIONS............................................. 284
Making Section Views ............................................................................. 285
 Creating Detail Views .............................................................................. 287

11. Dimensional & Geometric Constraints  ......................... 291

Working with Constraints ............................................................... 293
About Dimensional Constraints............................................................ 293
QUICK SUMMARY OF DIMENSIONAL CONSTRAINTS...................................... 294
Using Dimensional Constraints............................................................. 296
QUICK SUMMARY OF 2D GEOMETRIC CONSTRAINTS.................................. 298
About Geometric Constraints............................................................... 299
Identifying Geometric Constraints......................................................... 299
Using Geometric Constraints ............................................................... 300
Appendices

A. Concise Summary of Command Names ......................... 307

B. Concise Summary of Variables & Settings ................... 333

C. What's New in BricsCAD V20 ................................. 359

   New Entities ................................................................ 360
   2D Drawing and Editing ........................................... 363
   Dimensioning .......................................................... 366
   General 3D .............................................................. 366
   Customization and Security ...................................... 372
Welcome to BricsCAD V20!

You want to make drawings with BricsCAD, and this book shows you how — in as little as a day! Before doing any kind of drafting, you may want to take a tour of the BricsCAD user interface so that you know your way around the software.

Even when you know other CAD programs, you may find it useful to skim this chapter and take note of where 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 V20
• 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 — when clicked, executes the associated command
Cursor — provides you with feedback from the operating system and from BricsCAD
Flip screen — switches between the drawing window and a text window
Flyout — when clicked, reveals a secondary toolbar
Icon — represents commands pictorially
Layout — determines how drawings are plotted
Pickbox — specifies the area in which points being picked, entities are selected
Quick Access toolbar — miniature toolbar with often-used commands
Ribbon — collects buttons into tabs and panels
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut*</th>
<th>Menu Selection**</th>
<th>Ribbon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>? or F1</td>
<td>Help</td>
<td>Help</td>
</tr>
<tr>
<td>Line</td>
<td>L</td>
<td>Draw</td>
<td>Line</td>
</tr>
<tr>
<td>Quit</td>
<td>Alt+F4</td>
<td>File</td>
<td>Exit</td>
</tr>
<tr>
<td>TextScr</td>
<td>F2</td>
<td>View</td>
<td>Prompt History Window</td>
</tr>
<tr>
<td>Undo</td>
<td>Ctrl+Z</td>
<td>Edit</td>
<td>Undo</td>
</tr>
<tr>
<td>UcsIcon</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
How to Start BricsCAD

If BricsCAD is not yet set up on your computer, do so. Your computer must be operating any release of Windows since Vista, or just about any recent dialect of Linux or MacOS.

The easiest way to start the program is to double-click the BricsCAD icon found on the computer desktop.

If you do not see the icon on the desktop, then start the program using traditional methods.

STARTING BRICSCAD ON WINDOWS

BricsCAD V20 works with Windows Vista and newer.

Windows Vista and 7
In Windows Vista or 7, you start the program from the taskbar:

1. Click the taskbar’s Start icon.
2. Choose All Programs.
3. Select Bricsys, followed by the BricsCAD V20 folder, and then click on BricsCAD V20.

The exact name you see depends on the language version you downloaded. For example, “BricsCAD V20 (x64) en_US” is the name of the 64-bit program for English speakers in the US dialect.

Windows 8
In Windows 8, you make these moves:

1. If necessary, switch to the Start screen.
2. In the Start screen, tap on the BricsCAD V20 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 V20 item.

**TIP** Your current licence runs all versions of BricsCAD going back to V14.

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 V20, 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 V20 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 V20 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 V20 USER INTERFACE

Illustrated is BricsCAD running on Windows 10.

Title bar
Quick Access toolbar
Crosshair cursor
Pickbox
Drawing area
Drawing origin (0,0)
UCS icon
Grid
Layout tabs
Command panel
X, y coordinate and
No matter the operating system, BricsCAD looks pretty much the same in each one. Instructions in this book specific to Linux and Mac are shown in gray text.
Depending on the speed of your computer, it can take as little as a couple of seconds to load the program. First, the BricsCAD Launcher appears.

The Launcher and the Start Tab

(NEW IN V20) The first thing you see is the BricsCAD Launcher dialog box. (It replaces the Getting Started dialog box from earlier releases.)

This dialog box has options:

- **Shape** — install and launch Shape, the free 3D modeler
- **Drafting through BIM** — start BricsCAD with one of these workspaces
- **Last used** — starts BricsCAD with the workspace you last used
- **Manage license** — opens the Licensing Information dialog box to tell you about your licensee, such as expiry date
- **Do not show this window again** — disables this dialog box the next time you start BricsCAD

TIP  The workspaces available in the Launcher depend on the license level of BricsCAD you purchased. When you are using the 30-day free version, you have access to all workspaces.

For now, enter the program by clicking **Drafting**. You see the splash screen, whose job is to entertain you while the program continues to load. It clears itself from the screen automatically.
THE START TAB

(NEW IN V20) Notice that BricsCAD opens up, displaying the Start tab. (The Start tab replaces the Welcome dialog box of V19.)

This tab has three ways of seeing it:

- **Home** — accesses drawing files
- **Learn** — accesses online resources for using BricsCAD
- **Applications** — access third-party applications that work with BricsCAD

The Home view lets you start a new drawing based on a specific template, chose a drawing you opened earlier (under Recent Drawings, empty on a new installation), or open any other DWG file through the operating system’s file manager.

**TIP** Most of the ribbon does not work when no drawing is open. Clicking the blue Bricsys icon does, however, access file and drawing management functions.

For now, click **New Drawing**.
THE BRICSCAD WINDOW

Take a look at the overview of the BricsCAD window by checking out the figure spread across the previous 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 UCS icon with its red-green x,y axes. The drawing region is surrounded by several panels of information — toolbars or the ribbon, a status bar, and so on.

Along the very top of the BricsCAD window, you see the title bar with the name of the current drawing. Below it is the Quick Access toolbar with oft-used commands, below them the fat ribbon, and then the thin row of drawing tabs. (Depending on how BricsCAD is configured, you might see toolbars and the traditional dropdown menu bar.)

The way BricsCAD initially looks

Near 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 of BricsCAD’s 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.

The elements of the cursor in the drawing area

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-ended 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](image)

**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.

![The elements of the command bar](image)

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.

   ![The BricsCAD drawing area](image)

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)*

   ![Following the instructions in the command bar](image)
4. As you keep moving the mouse, notice the line that stretches like a “rubber band” from the point you picked. This rubber band is also known as the “drag line.” It shows where the line would be located if you were to click now.

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

5. Move the mouse some more, and then press the pick button again. There: you’ve drawn your first line with BricsCAD!

6. Continue drawing some more lines by repeating the same steps: (a) move the mouse and then (b) press the pick button. Draw as many lines as you like.
7. 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)

Handling Options Efficiently

I want to point out that the prompt text — ‘Set end point or [Angle/Length/Follow/Close/Undo]’ — has subtle aspects that are not immediately obvious.

You can specify an option by typing just the letter(s) displayed in uppercase, such typing ‘a’ to start the Angle option. Options are shown in the square brackets.

(New in V20) Alternatively, you can use the mouse cursor to pick the name of the option.

Sometimes, there is a default option (or value) that is shown angle brackets, such as <Last point>. “Default” means that this is what BricsCAD will do when you just press Enter, without picking an option. This becomes a pretty fast way of working.

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

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Effect of pressing Enter at the prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of line:</td>
<td>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.</td>
</tr>
<tr>
<td>&lt;End point&gt;:</td>
<td>Pressing Enter terminates the Line command, just as Esc does.</td>
</tr>
<tr>
<td>:</td>
<td>Pressing Enter repeats the last command, which in this case is the Line command.</td>
</tr>
</tbody>
</table>

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 Quick Access toolbar
- Select Undo from the Edit menu
  ![Undo icon](image.png)
- 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.

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

### Seeing What You Did Before: Command History

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

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

![Changing the size of the command bar](image)

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

![Floating command bar](image)
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\App\Data\Local\Bricsys\BricsCAD\V20x64\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."

   ![Auto-complete showing commands beginning with L](image)

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.'

   ![Auto-complete showing commands beginning with LI](image)

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

   ![Auto-complete showing commands beginning with LIN](image)

   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**
- **Alias with command name in brackets**
- **System variable name**
- **Command name**
- **Auto-Append**

*The auto-complete display*
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.

Changing auto-complete options

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.

Accessing more options for the look of the command bar

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:

![UCS Icon Diagram]

- **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.

---

**SUPPORTED GRAPHICS BOARDS**

BricsCAD works with whatever graphics board is built into your computer. When it comes to non-wireframe renderings, however, BricsCAD employs RedSDK GPU-acceleration technology provided by Redway3D. (GPUs are the processing chips on graphics boards.)

**WINDOWS**


**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.
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

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.
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.

![Saving changes to the drawing before exiting BricsCAD](image)

Alternatively, press Ctrl+Q, or else select Exit from the File menu.
Navigating the BricsCAD Interface

In this chapter, we continue the tour of BricsCAD’s user interface (UI) in great 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 some of the nuances to the UI, such as entering aliases and working with relative coordinates.

IN THIS CHAPTER

- Working with the upper half of BricsCAD: title bar, Quick Access toolbar, ribbon, and drawing tabs
- Trying out some user interface elements of the drawing area: Quad, LookFrom widget, and more
- Finding out about the lower half of BricsCAD: layout tabs, scroll bar, command bar, and status bar
- Understanding panels (palettes)
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]”

**Quick Access toolbar** — contains commands that are often used in BricsCAD

**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)]
TIP   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.
QUICK ACCESS TOOLBAR

**(NEW IN V20)** The Quick Access toolbar lists commonly used commands in a format that is always visible — unlike the ribbon, where tabs and drop panels hide commands from view.

The Quick Access toolbar for 2D drawings

The commands are grouped as follows, with a vertical bar separating groups:

- **Filing commands** — start a new drawing, open a drawing, save the current drawing, and save the current drawing by another name
- **Printing commands** — preview, print the drawing, and publish a group of drawings
- **Undo and redo** — undo the last command, and then redo it
- **Layer droplist** — control the display of layers
- **Selection commands** — match properties from one entity to others, and isolate-unisolate entities
- **Panel toggles** — turn the display of Drawing Explorer, Sheet Sets, Settings, and Properties panels on and off, as well as maximize the drawing area (clean screen)
- **Help** — access online help

A slightly different toolbar is displayed in 3D workspaces; it offers 3D selection modes.

**TIP** The Quick Access toolbar can contain any commands that you like. To change the toolbar, enter the **Customize** command, choose the **Toolbars** tab, and then select the **Access** node.

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.

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

:ribbon

Shown below is the ribbon’s standard Home tab for 2D drawings. The Home tab changes somewhat when you work with 3D workspaces.
Other tab names are Insert, View, and so on. 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.

The View tab of the ribbon

(NEW IN V20) The blue Bricsys logo is a button that accesses a menu of file-related commands. This replaces the Files tab from earlier releases of the program.

Menu of file-related commands

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

DRAWING (DOCUMENT) 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 this interface element “Document Tabs.”

Drawing tabs show the drawing names that are currently open

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

Context menu of document tab actions
Most of the commands are probably already 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.

*(NEW IN V20)* The first tab displays the Start window, in which you can also access recently-opened drawings, tutorials, and third-party add-ons. The tab is called “Start.” See Chapter 1.

There are two variables that control the visibility of these 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 here.

*The drawing area in BricsCAD*
Here I will tell you about many of the user interface elements you find in the drawing area:

- **Look From widget** — interactively changes the 3D viewpoint of drawings; turn on-off and adjust settings with the LookFrom command
- **UCS icon** — reports the orientation of x,y- and x,y,z-axes in 2D and 3D space; turn on-off and adjust settings with the UCSicommand
- **Hot Key assistant** — indicates options available with the Ctrl key; toggle on-off by clicking HKA on the status bar
- **Dynamic Input** — reports distances and angles of entities as they are being drawn; toggle on-off by clicking DYN on the status bar
- **Shortcut menus** — right-click menus that provide access to useful commands and options
- **Quad cursor** — at-cursor access to context-sensitive drawing and editing commands; toggle on-off by clicking QUAD on the status bar
- **Scroll bars** — pan the drawing horizontally and vertically; turn on and off with the ScrollBars command
- **Prompt menu** — displays command options in a small menu
- **Manipulator widget** — directly edits entities with actions like move, copy, and rotate; toggle on-off with the Manipulator system variable
- **Clean screen** — minimizes the user interface to maximize the drawing area
- **Panels** — sources of information and control over drawings and entities

## LOOK-FROM CONTROL

**Command**  LookFrom

The LookFrom widget lets you quickly change the viewpoint of 3D drawings. The widget is found in the upper right corner of the drawing area.

*LookFrom control with no cursor interaction*

When you pass the cursor over the widget, the preview image of a chair appears. Pausing the cursor over one of the triangles previews what the 3D view will look like as the chair changes its position. The small green dot indicates the cursor position, kind of like a laser pointer:

*Cursor pausing over one of the direction triangles*
Clicking the triangle changes the 3D viewpoint.

![3D Viewpoint Image]

**TIPS**  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, or press the **Home** button on the keyboard. This is particularly helpful when running the widdget in **Twist** mode.

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

![Shortcut Menu Image]

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

![Isometric and Twist Modes]

*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:

<table>
<thead>
<tr>
<th>LookFromDirectionMode</th>
<th>Number of Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6 orthogonal views</td>
</tr>
<tr>
<td>1</td>
<td>14 views; no flat views of corners</td>
</tr>
<tr>
<td>2</td>
<td>18 views; top down corners</td>
</tr>
<tr>
<td>3</td>
<td>26 views; eight top down corners</td>
</tr>
</tbody>
</table>

**TIP** To quickly change the visual style of 3D models, such as wireframe or rendered, use the **Visual Style** setting in the Properties panel.

---

**THE UCS ICON**

**Status bar** DUCS  
**Command** UCS

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 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.** When we draw things at an angle in 2D, such as the angled wing of a building, then we would use a UCS. Changing the x,y-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 matches the new angle.
To change the angle of the UCS icon, you need to 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 parameters, BricsCAD figures out the direction of the z axes using the right-hand rule.

Here is how to use the **UCS** command to do this:

```
: 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 can give it a name to let you switch 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, does this for you automatically through dynamic UCS. Here is a brief example. In the figure below, the UCS icon is initially in the default “world” position:

![UCS in the world (default) position, aka world coordinate system](image)

To write some text on the bottom of the piston, the UCS needs to be relocated to the flat face. To do so, first turn on 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 the two areas highlighted in blue in the figures below.

![Left: UCS dynamically placed on one face...; right: ...and on another face](image)
**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.

![UCS icon in paper space](image)

In the early days of CAD, back when layout mode was first introduced, the software interface was vague enough that it could be difficult for users to determine if they were in model or paper space. In some CAD programs, the UCS icon would switch to a triangle shape; in BricsCAD, it looks like a piece of paper.

In CAD today, however, the user interface makes layout mode quite clear, and so the paper icon remains for sentimental reasons.

**HOTKEY ASSISTANT**

**Status bar** **HKA**

The hotkey assistant is an interactive widgets that reports options accessible by pressing the Ctrl key that might otherwise be unknown to users. (In earlier releases of the software, it was known as the “Tips widget.”)

For example, the following widget appears during the Polysolid command.

![Hotkey Assistance appears at the bottom of the drawing area](image)

Notice that the widget shows several icons in a row:

- 1 and Ctrl “buttons” — remind you to tap the Ctrl key to make the changes; the 1 and Ctrl icons do nothing
- Two or more icons — display the change that would occur each time you tap the Ctrl key
- x — closes the widget
In the figure above, tapping the **Ctrl** key changes 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.

![Tool tips reporting the purpose of the icon](image)

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.

![Shortcut menu for the status bar’s HKA button](image)

The **Configure** button leads you to a dialog box that lets you determine when the widget appears.

![Determining when the Hotkey assistant appears](image)
DYNAMIC INPUT

Status bar DYN

While traditionally we controlled drawing and editing commands by entering options, distances, and angles in the Command bar, another way of using dynamic input, because it is interactive within the drawing area. It is known as “dynamic input,” because it displays distances and angles dynamically — right at the cursor in the drawing.

(On the status bar, ensure DYN is turned on. When it is black (instead of gray), then dynamic input is on: )

![Dynamic input displaying the length and angle of the line being drawn](image)

These are known as “dynamic dimensions,” because the length and angle values change as you move the cursor:

![Dynamic dimension values changing as the cursor moves](image)

Notice in the figures above that the length field is highlighted in blue. You can change the value at the keyboard, which is better for precision. As you enter a number at the keyboard, it replaces the value in the blue length field.

Then, to change the angle, press the Tab key. The angle field turns blue, and the distance field turns red (to indicate it is locked temporarily). Now you can enter an angle value at the keyboard. If necessary, press Tab to return to the length field, or else press Enter to finish data input.

![Changing the value of the angle field](image)
SHORTCUT MENUS

Scattered all about BricsCAD are accesses to shortcut menus. Nearly any place you right-click, a context 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.

Holding down the Shift key while clicking the right mouse button accesses entity snap modes

I don’t detail them all here, because there are so many, but in later lessons, I’ll point out shortcut menus that are useful to the work at hand. Short cut menus can be customized with the Customize command.

QUAD CURSOR

Status bar QUAD
Shortcut F12

The Quad is unique to BricsCAD in the way that it incorporates drawing and editing commands into a single interface that operates right at the cursor. 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.

Initial display of the Quad

(If the Quad 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.

*Editing commands*

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.

*More editing commands*

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:

*Changing the settings of the Quad*

**Quick Properties**

*Status bar RT*

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

Hover the cursor over an entity, and the Quad displays some of its properties:

*Properties displayed by the Quad*

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

Command ScrollBars

A pair of scroll bars rest at the right and bottom edges of the drawing area. They let you pan the drawing up and down, and left and right. Normally, they are turned off, but I find them handy, so I turn them on. To turn on scroll bars, enter the ScrollBar command.

Scroll bars are limited to panning left-right and up-down. As a faster and more flexible alternative to scroll bars, 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 options that are available for the current command. This menu appears in the upper right corner (usually) of the drawing area. 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.

To change how the Prompt menu works, search for ‘prompt menu’ in the Settings dialog box.

MANIPULATOR WIDGET

Variable Manipulator

To manipulate entities interactively, BricsCAD provides the manipulator widget. When the widget is turned on, it attaches to the 2D or 3D entity that you select.
You drag arrowheads and 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:

<table>
<thead>
<tr>
<th>Manipulator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not displayed (default)</td>
</tr>
<tr>
<td>1</td>
<td>Display manipulator when entities are selected</td>
</tr>
<tr>
<td>2</td>
<td>Display manipulator when left mouse button is pressed longer than 250msec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manipulator Action</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drag a blue arrowhead</td>
<td>Scale (resize) or Mirror (default)</td>
</tr>
<tr>
<td>Drag a gold bar</td>
<td>Move</td>
</tr>
<tr>
<td>Hold Ctrl while dragging bar</td>
<td>Copy</td>
</tr>
<tr>
<td>Drag a yellow arcs</td>
<td>Rotate</td>
</tr>
</tbody>
</table>

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).
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

CLEAN SCREEN

Shortcut Ctrl+0

Sometimes the user interface can get pretty cluttered, and so it’s a good idea to clear most of it away quickly.

Press Ctrl+0 to toggle clean screen on and off.
The Settings dialog box lets you choose which user interface elements you still want to see during a clean screen session: enter “cleanscreen” in the search field.

---

As an alternative to clean screening your display, consider adding a second monitor and placing panels, toolbars, and other UI item there.

---

PANELS

Panels are so important to BricsCAD that I am going to give you this quick tour of their user interface features. (These were known as “bars” or “palettes” in earlier releases.) Once you learn the UI elements for the Properties panel, you can use them with many 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
The easiest way to access panels is to right-click the ribbon or a toolbar, and then choose **Panels**:

![Panels](image)

**Turning panels on and off**

From the list, select the panels you wish to open or close. The blue checkmark indicates the associated panel is open. Several panels and bars can be opened and closed using shortcut keystrokes. Here are some of them:

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
<th>Shortcut Keystroke</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open or close Properties panel</td>
<td>Properties</td>
<td>Ctrl+1</td>
</tr>
<tr>
<td>Open the Drawing Explorer</td>
<td>Explorer</td>
<td>Ctrl+2</td>
</tr>
<tr>
<td>Open or close Command bar</td>
<td>CommandLine</td>
<td>Ctrl+9</td>
</tr>
</tbody>
</table>

It can become cumbersome when many panels are open, and so BricsCAD offers two ways to stack them:

- **Tabs** — overlapping panels are indicated by tabs along the top of the panels
- **(new in V20) Icons** — overlapping panels are indicated by an icon along the edge of the panels

![Tabs vs Icons](image)

**Left**: Tabbed panels; **right**: Iconized panels

To choose the method, right-click the ribbon or a toolbar, and then choose **Panel stacking**.

![Panel stacking](image)

Choose the method of panel stacking
Below the Drawing Area

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

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

![Layouts bar screenshot](image)

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

```
ready: -92.25, 122.26, 0
```

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.
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.

![Layout Manager dialog box](image)

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.

![Right-clicking a layout tab](image)

Layouts are discussed in greater detail later in this book.

So, there are drawing tabs to switch between drawings, and then 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 by default:

```
X, Y coordinates, and elevation (Z)
Help text
Current settings for styles and workspaces
Mode toggles
Status bar options
```

“By default,” because you can select which items are displayed on the status bar: click the Status Bar Options button to see the full list.

- **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.
- **Linestyle** — reports the current line type (ByLayer)
- **Style** — reports the current text style (Standard)
- **Dimension style** — reports the current dimension style (Standard)
- **Workspace** — reports the current workspace (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
- **DUCS** — 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
- **LOCKUI** — prevents the toolbars from being moved accidently
- **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,

![Typical context menu for status bar buttons](image)

**Settings** is a shortcut to the Settings 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 other status bar buttons, they may present longer lists of options. For example, right-click the x,y,z coordinates to change how they are displayed:

![Changing the coordinate display style](image)

Clicking the tiny black triangle near the right end of the status bar displays a shortcut menu that lists all possible toggles.

![Shortcut menu for status bar buttons](image)

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.
Before creating new drawings with BricsCAD, you must prepare them. Here you learn how to set up new drawings with the following tutorials:

- **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.
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 of 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 (1 000mm per meter)
m Meters

NEW COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Alias</th>
<th>Menu Bar</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer</td>
<td>la</td>
<td>Settings</td>
<td>Home</td>
</tr>
<tr>
<td>Limits</td>
<td></td>
<td>Settings</td>
<td>Drawing Limits</td>
</tr>
<tr>
<td>NewWiz</td>
<td>ddnew</td>
<td>File</td>
<td>New Wizard</td>
</tr>
<tr>
<td>Settings</td>
<td>options</td>
<td>Settings</td>
<td>Settings</td>
</tr>
<tr>
<td>Save</td>
<td>Ctrl+S</td>
<td>File</td>
<td>Save</td>
</tr>
<tr>
<td>SaveA$</td>
<td></td>
<td>File</td>
<td>Save A$</td>
</tr>
</tbody>
</table>
Before You Begin

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

Before beginning this series of tutorials, however, you may want to measure your own yard to locate its 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 used for the 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 complete set of tutorial files for this book can be downloaded from https://my.pcloud.com/publink/show?code=XZnq8mkZ7T7f1pnxJYb7vDyVI6DwyqK1Ysk. 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**

<table>
<thead>
<tr>
<th>Command</th>
<th>NewWiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>File</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Home</td>
</tr>
</tbody>
</table>

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 start the first tutorial

In this chapter, you create the new drawing with the assistance of a **wizard**. 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.

The “wizard” consists of 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.

1. If BricsCAD is not running, start it now by double-clicking its icon on the desktop
2. Notice the BricsCAD Launcher dialog box. Choose “Drafting” as the workspace, as it is the one best suited for 2D drawings.

![Choosing the Drafting workspace](image)
3. When the Start page appears, click **New Drawing**.

4. Start the wizard by taking one of the following steps:
   - In the ribbon, click the blue **B** and then choose **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
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 drop list.

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.
5. Select the **Use a Wizard** button by clicking the circle (called a “radio button”) next to its label.

6. 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)**

7. If you are following along with my tutorial drawing, then choose **Metric**, as shown below.

8. Click **Next**.

### 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.

![Angular Measurement](image)

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.

<table>
<thead>
<tr>
<th>Unit Name</th>
<th>Measures In</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural</td>
<td>Feet, inches, fractional inches</td>
<td>4’-6 1/16”</td>
</tr>
<tr>
<td>Engineering</td>
<td>Feet, inches, decimal inches</td>
<td>4’-6.0625”</td>
</tr>
<tr>
<td>Fractional</td>
<td>Inches, fractional inches</td>
<td>54 1/6”</td>
</tr>
<tr>
<td>Decimal (default)</td>
<td>Units and decimal units</td>
<td>2128.4449</td>
</tr>
<tr>
<td>Scientific</td>
<td>Units, decimal units, and exponents</td>
<td>2.1284E+03</td>
</tr>
</tbody>
</table>
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; 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.)

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

10. 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".)

11. 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:

<table>
<thead>
<tr>
<th>Angle Name</th>
<th>Measures In</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal degrees</td>
<td>Degrees and decimals</td>
<td>12.3456</td>
</tr>
<tr>
<td>Degrees/Minutes/Seconds (d '&quot;)</td>
<td>Degrees, minutes, seconds, and decimal seconds</td>
<td>12d34'56&quot;</td>
</tr>
<tr>
<td>Grads (g)</td>
<td>Grads and decimals</td>
<td>123.45g</td>
</tr>
<tr>
<td>Radians (r)</td>
<td>Radians and decimals</td>
<td>1.23r</td>
</tr>
<tr>
<td>Surveyor’s Units (N d ’” E)</td>
<td>North/South degrees towards East/West</td>
<td>N12d34'56&quot;E</td>
</tr>
</tbody>
</table>

There are 400 grads (used in Germany) and 2*pi 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 anytime with the Units command without affecting the drawing.)
12. Ensure that the default, **Decimal Degrees**, is selected for the angle of measurement.
13. Change the default **Precision** to 0 degrees, as we are measuring angles to the nearest degree.

![Image of Precision settings]

14. 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.

15. 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.

![Image of plot style options]

16. 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.

17. 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.

18. 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 an 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Bar</td>
<td>Settings</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Settings</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Command</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>SE, options, ddmodes, rm, dsettings, ddsetvar</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Settings</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Settings</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Settings</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>GridDisplay</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Turns off the display of the grid</td>
</tr>
<tr>
<td>1 (default)</td>
<td>Display grid beyond the area of the limits</td>
</tr>
<tr>
<td>2 (default)</td>
<td>Display adaptive grid</td>
</tr>
<tr>
<td>4</td>
<td>Allow sub-divisions below grid spacing</td>
</tr>
<tr>
<td>8</td>
<td>Follow dynamic UCS</td>
</tr>
</tbody>
</table>

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**.

![Grid Dialog Box]

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.

![Grid Style Options]

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**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Layer, layerpanelon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>LA</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Settings</td>
</tr>
<tr>
<td></td>
<td>Tools</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Home</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Settings</td>
</tr>
</tbody>
</table>

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.
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.

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

<table>
<thead>
<tr>
<th>Current</th>
<th>indicates the current layer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Name</td>
<td>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.</td>
</tr>
<tr>
<td>Description</td>
<td>describes the purpose or content of the layer; optional.</td>
</tr>
<tr>
<td>On/Off (obsolete)</td>
<td>when on, displays entities; when off, entities cannot be seen nor are they plotted.</td>
</tr>
<tr>
<td>Freeze</td>
<td>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.</td>
</tr>
<tr>
<td>Locked</td>
<td>locks layers, which means entities are seen but cannot be edited. Unlock layers to make entities available for editing.</td>
</tr>
<tr>
<td>Color</td>
<td>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.</td>
</tr>
<tr>
<td>Linetype</td>
<td>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.</td>
</tr>
<tr>
<td>Lineweight</td>
<td>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.</td>
</tr>
<tr>
<td>Plot Style</td>
<td>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.</td>
</tr>
<tr>
<td>Plot</td>
<td>specifies whether the layer is plotted; when off, the layer is not plotted.</td>
</tr>
</tbody>
</table>

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. |
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.

   Step 1: Click to create a new layer
   Step 2: Rename the generic name
   Step 3: Set the properties
   Step 4: Set the layer as current

   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.

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

<table>
<thead>
<tr>
<th>New Layer</th>
<th>Layer Color</th>
<th>Color Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lot</td>
<td>Blue</td>
<td>Color #5</td>
</tr>
<tr>
<td>House</td>
<td>White (black)</td>
<td>...</td>
</tr>
<tr>
<td>Road</td>
<td>Red</td>
<td>Color #1</td>
</tr>
<tr>
<td>Lawn</td>
<td>Cyan (light blue)</td>
<td>Color #4</td>
</tr>
<tr>
<td>Plants</td>
<td>Green</td>
<td>Color #3</td>
</tr>
<tr>
<td>Pond</td>
<td>Blue</td>
<td>Color #5</td>
</tr>
</tbody>
</table>

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.
**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 V20. 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:

<table>
<thead>
<tr>
<th>Save As File Format</th>
<th>Saves Drawings in File Formats Used By AutoCAD Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCAD 2018</td>
<td>2018 2019 2020</td>
</tr>
<tr>
<td>AutoCAD 2013</td>
<td>2013 2014 2015 2016 2017</td>
</tr>
<tr>
<td>AutoCAD 2010</td>
<td>2010 2011 2012</td>
</tr>
<tr>
<td>AutoCAD 2007</td>
<td>2007 2008 2009</td>
</tr>
<tr>
<td>AutoCAD 2004</td>
<td>2004 2005 2006</td>
</tr>
<tr>
<td>AutoCAD 2000</td>
<td>2000 2001 2002</td>
</tr>
<tr>
<td>AutoCAD Release 14</td>
<td>Release 14 (released in 1997)</td>
</tr>
<tr>
<td>AutoCAD Release 13</td>
<td>Release 13 (released in 1994)</td>
</tr>
<tr>
<td>AutoCAD Release 11/12</td>
<td>Releases 11 and 12 (released in 1990 and 1992, respectively)</td>
</tr>
</tbody>
</table>

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:

- 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.
Saving Drawings

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

1. In the Quick Access toolbar, click the **Save** button that looks like a diskette (tooltip: Save).
2. Because new drawings have a generic name, such as “Drawing1.Dwg,” BricsCAD displays the Save Drawing As dialog box so that you can give it a better name. Type the name “Yard” in the **File name** text entry box.

**QUICK SUMMARY OF ALTERNATIVE SAVE FORMATS**

In addition to saving drawings in AutoCAD's .dwg file format, BricsCAD exports drawings with the **Export** command in these 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
When you need to translate drawings to and from other formats, then you may want to enlist Communicator. This is an optional, extra-cost add-on for BricsCAD Pro and Platinum that translates files in additional formats.

### Import

<table>
<thead>
<tr>
<th>File format</th>
<th>Direct translator</th>
<th>Parasolid-based translator</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACIS</td>
<td>V17 - 2019.1.3</td>
<td></td>
</tr>
<tr>
<td>CATIA V4</td>
<td>4.1.9 - 4.2.4</td>
<td></td>
</tr>
<tr>
<td>CATIA V5</td>
<td>V5RE - V5-6R2017</td>
<td></td>
</tr>
<tr>
<td>3DExperience (CATIA V6)</td>
<td>up to V6 R2017x</td>
<td></td>
</tr>
<tr>
<td>IGES</td>
<td>up to 5.3</td>
<td></td>
</tr>
<tr>
<td>Inventor</td>
<td>V6 - V2017</td>
<td></td>
</tr>
<tr>
<td>JT</td>
<td>8.x, 9.x and 10.x</td>
<td>8.x, 9.x and 10.x</td>
</tr>
<tr>
<td>NS</td>
<td>NX 1 - NX 11</td>
<td>11 – NX 11</td>
</tr>
<tr>
<td>Parasolid</td>
<td>9.0.x – 29.0.x</td>
<td>9.0 - 29.0.137</td>
</tr>
<tr>
<td>Pro/E / Creo</td>
<td>16 – Creo 4.0</td>
<td></td>
</tr>
<tr>
<td>Solid Edge</td>
<td>V18 – ST9</td>
<td>V18 – ST9</td>
</tr>
<tr>
<td>STEP</td>
<td>AP203, AP214, AP242 (*)</td>
<td></td>
</tr>
<tr>
<td>STL</td>
<td>All (Graphical data only)</td>
<td></td>
</tr>
<tr>
<td>VDA – FS</td>
<td>1.0 - 2.0</td>
<td></td>
</tr>
<tr>
<td>XCGM</td>
<td>R2012 – 2018.1.0</td>
<td></td>
</tr>
</tbody>
</table>

### Export

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

You access the additional formats with the **Import** and **Export** commands, after the software is installed on your computer.
3. Click the Save button. BricsCAD saves the drawing with the name yard.dwg. From now on, you and BricsCAD refer to this drawing as “Yard.”

MAKING BACKUPS AUTOMATIC

For speed, BricsCAD keeps parts of the drawing in the computer’s memory (RAM). The drawback is that when the operating system crashes or when the power is cut to the computer, you may lose some or all of your work. Because your work is valuable, 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.

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. Press Ctrl+S often!
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**

<table>
<thead>
<tr>
<th>Command</th>
<th>Aliases</th>
<th>Menu Selection</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel</td>
<td>Esc</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Fillet</td>
<td>f or fi</td>
<td>Modify</td>
<td>Fillet</td>
</tr>
<tr>
<td>Mirror</td>
<td>mi</td>
<td>Modify</td>
<td>2D Mirror</td>
</tr>
<tr>
<td>Move</td>
<td>m</td>
<td>Modify</td>
<td>Move</td>
</tr>
<tr>
<td>Open</td>
<td>op or Ctrl+O</td>
<td>File</td>
<td>Open</td>
</tr>
<tr>
<td>OSnap</td>
<td>os or F3</td>
<td>Settings</td>
<td>Settings</td>
</tr>
<tr>
<td>PLine</td>
<td>pl</td>
<td>Draw</td>
<td>Polyline</td>
</tr>
<tr>
<td>Plot</td>
<td>Ctrl+P</td>
<td>File</td>
<td>Print</td>
</tr>
<tr>
<td>QSave</td>
<td>Ctrl+S</td>
<td>File</td>
<td>Save</td>
</tr>
<tr>
<td>Zoom</td>
<td>z</td>
<td>View</td>
<td>Zoom</td>
</tr>
</tbody>
</table>

* The ellipsis (...) indicates no menu selection is available.
Reopening Drawings

<table>
<thead>
<tr>
<th>Command</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>op</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>File</td>
</tr>
<tr>
<td>Ribbon</td>
<td>B</td>
</tr>
<tr>
<td>Shortcut</td>
<td>Ctrl+O</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Standard</td>
</tr>
</tbody>
</table>

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

**TIP** The Recent Drawings item always holds the names of the last ten drawings you opened with BricsCAD. Here’s how to access them:

1. Click **B** on the ribbon.
2. Slide cursor down to **Recent Drawings**, and then click on **yard.dwg**. See figure below:

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

**THIS CHAPTER’S DRAWING**

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

![Tutorial-04 drawing file](image-url)
Drawing the Lot’s Boundary

<table>
<thead>
<tr>
<th>Command</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>L</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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) of the drawing, and then working your way counterclockwise 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

3. Respond to the 'Specify first point:' prompt by typing the coordinates of the origin:
   
   **Start of line**: 0,0

4. To draw the lower boundary line 35m long, you need to tell BricsCAD that the far end of the line is located at the x,y coordinates of **35,0**. (In imperial units, this is **116',0"**)
   
   **Set end point** or [Angle/Length/Undo]: 35,0
See the figure below for what the first line should look like.

5. The next line is 24 m 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!)

   \textit{Set end point} or [Angle/Length/Follow/Undo]: \texttt{35,24}

\textbf{TIP} Notice that the prompt line \textit{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”...

   \textit{Set end point} or [Angle/Length/Follow/Undo]: \texttt{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 encounter another option, “Close.”

6. You drew the first two lines with \textit{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 \textit{polar coordinates}. In these, you specify the distance and angle. Continue drawing the lot boundary by combining relative and polar coordinates like this:

   \textit{Set end point} or [Angle/Length/Follow/Close/Undo]: \texttt{@23<180}

   (In imperial units: End point: \texttt{@76'<180})

   With all that punctuation, entering \texttt{@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!

\textbf{TIPS} Using relative polar coordinates makes sense when you have many angled lines to draw.

   When you tell BricsCAD to draw a line with the relative polar coordinates, you use a special notation with the following meaning:

   \begin{tabular}{ll}
   \textbf{Notation} & \textbf{Meaning} \\
   \texttt{@} & Use relative coordinates \\
   \texttt{23} & Distance is 23 m from the current point \\
   \texttt{<} & Draw the line at the angle... \\
   \texttt{180} & ...of 180 degrees \\
   \end{tabular}

   Lines are drawn relative to the current point.

   Angles are measured in \textit{absolute} degrees, using the East-is-0-degrees convention.

   If you make a mistake entering any coordinate notation, simply use the \textbf{Undo} option (type \texttt{U} at the prompt) and reenter the coordinates.
7. Enter the coordinates for the next endpoint using the **Angle** option, which then prompts for the length of the line. Notice that the **Angle** option removes the need to use the `@` and `<` characters:

```
Set end point or [Angle/Length/Follow/Close/Undo]: a
Angle of line: 216.88
Length of line: 15
```

(In imperial units, the angle is the same, but the length of the line is 50'.) 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). BricsCAD automatically draws a line from the current endpoint to the beginning of the first line, without you having to work it out.

```
Set end point or [Angle/Length/Follow/Close/Undo]: c
```

As I mentioned earlier, slashes separate option 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** — asks 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 some working space around what you’ve drawn, 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 an 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 
   ':

   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)

   TIP  You can cancel commands at any time by pressing Esc. Sometimes, however, you may need to press Esc 
   two or even three times in commands that have large numbers of sub-options, such as the PEdit command.

5. Look at the ribbon, status bar, or toolbar to confirm that BricsCAD has changed the working layer from layer 
   Lot to layer House.
Drawing the Outline of the House

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>
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 0.1 m 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).

DDE combines cursor movement with keyboard entry — you show BricsCAD the angle directly, and then type in the distance. It’s a lot quicker than typing values of angles — and more intuitive, too. It works best with large angles, such as 90 degrees.

Carrying on with the tutorial, here is how to use direct distance entry:

4. The second line of the prompt displays many options. Don’t let them intimidate you! For now, you can ignore all of them, except for the default, ‘Set 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 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
QUICK SUMMARY OF ENTITY SNAP MODES

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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Command</th>
<th>Snaps to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>APParent</td>
<td>Apparent intersections of two entities</td>
</tr>
<tr>
<td>cen</td>
<td>CENTER</td>
<td>Center of arcs, circles, and polyarcs</td>
</tr>
<tr>
<td>end</td>
<td>ENDpoint</td>
<td>Either end of lines, arcs, and other open entities</td>
</tr>
<tr>
<td>ext</td>
<td>EXTension</td>
<td>Extension of two entities</td>
</tr>
<tr>
<td>from</td>
<td>FROM</td>
<td>Offset distance from a specified point</td>
</tr>
<tr>
<td>gcen</td>
<td>GCENTER</td>
<td>Geometric center of objects</td>
</tr>
<tr>
<td>ins</td>
<td>INSertion</td>
<td>Insertion point of blocks and text</td>
</tr>
<tr>
<td>int</td>
<td>INTERsection</td>
<td>Intersection of lines, arcs, circles, and other entities</td>
</tr>
<tr>
<td>mid</td>
<td>MIDpoint</td>
<td>Middle of lines, arcs, and other open entities</td>
</tr>
<tr>
<td>nea</td>
<td>NEAREST</td>
<td>Nearest point on the nearest entity</td>
</tr>
<tr>
<td>nod</td>
<td>NODE</td>
<td>Point entities</td>
</tr>
<tr>
<td>par</td>
<td>PARallel</td>
<td>Parallel to lines and other entities</td>
</tr>
<tr>
<td>per</td>
<td>PERpendicular</td>
<td>Perpendicular to lines, arcs, and other entities</td>
</tr>
<tr>
<td>qua</td>
<td>QUADRant</td>
<td>0-, 90-, 180-, and 270-degree points on arcs, circles, and polyarcs</td>
</tr>
<tr>
<td>qui</td>
<td>QUICK</td>
<td>First geometric feature found by BricsCAD</td>
</tr>
<tr>
<td>tan</td>
<td>TANGent</td>
<td>The tangents of arcs and circles</td>
</tr>
</tbody>
</table>

The M2p command is kind of like an entity snap, in that it finds the middle between two points.

Because more than one entity snap can be active at a time, BricsCAD provides a number of visual cues to identify the current one. You can toggle the visual cues through the Program Options | Display section of the Settings dialog box.

The table below illustrates on-screen markers associated with each entity snap mode.
DYNAMIC INPUT

Let’s now move to yet another way of using direct distance entry, one that is even more interactive in 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-long 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:

   ![Dynamic dimensions]

   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.

   ![Dynamic dimension values changing as the cursor moves]

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

   **TIP** When you enter numbers at the keyboard, they are entered into the blue field. Notice in the figures above that the length field is highlighted in blue.

   To move between the length and angle fields, press the **Tab** key. In the figure below, the **angle** field (135 degrees) turns blue, as shown below.

   ![Angle field highlighted]

   ![Angle field highlighted]
c. Enter the length of the wall, 1 and then press Tab. (In imperial units: 3’.)

![Diagram showing the length of a wall]

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)...

![Diagram showing the angle of the wall]

...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.

![Diagram showing the complete wall and angles]
6. Draw the remainder of the house outline by moving the cursor in the appropriate direction, and then entering the distances:

<table>
<thead>
<tr>
<th>Cursor Direction</th>
<th>Metric Distance</th>
<th>Imperial Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>1</td>
<td>3'</td>
</tr>
<tr>
<td>North</td>
<td>6</td>
<td>20'</td>
</tr>
<tr>
<td>West</td>
<td>8.5</td>
<td>28'</td>
</tr>
<tr>
<td>South</td>
<td>16</td>
<td>50'</td>
</tr>
</tbody>
</table>

7. Complete the polyline with the C option, as you did with the Line command.

When Lot was the working layer, the lines you drew showed up in blue. You changed the layer to House, and 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>m</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Change</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Modify</td>
</tr>
</tbody>
</table>

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

1. Enter Move the ‘Command:’ prompt:
   
   : move

2. Notice that BricsCAD asks what you want to move:
   
   Select entities to move:
   
   ("Entities" is an older word that means the same as objects.)

   At the same time, BricsCAD changes the crosshair cursor into a small square cursor, called the pick cursor.
3. Move the cursor to any part of the house’s polyline, and then press the pick button.
   
   Select entities to move: *(Pick the outline of the house)*
   
   The entire house outline is highlighted. The highlighting shows as a dotted line, which is how BricsCAD lets you know it found the entity you picked.

4. BricsCAD reports the number of entity selected, and then lets you make additional selections, if need be. The prompt repeats:
   
   Select entities to move: *(Press Enter to exit entity selection)*
   
   Because you are moving just one polyline, press Enter to end the entity selection process.

5. Just as when drawing lines, the Move command needs to know a *from*-point and a *to*-point. But here the *from*-point is named the “base point,” as follows:
   
   Enter base point [Displacement] <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.

![Toolbar 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/Undo]: (Press Enter to end the command)
```

![Line Command Example]

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

```
Set end point or [Angle/Length/Undo]: u
Set end point or [Angle/Length/Undo]: 12
```
FINISHING THE DRIVEWAY

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.

```
Fillet radius <radius=0.5>: 1
```

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=0.5>: 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Mirror</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>mi</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Change</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Modify</td>
</tr>
</tbody>
</table>

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](image)
   
   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](image)
   
   Start of mirror line: mid
   
   Snap to midpoint of: (Pick center of garage entrance)
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Print</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Bar</td>
<td>File</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Home</td>
</tr>
<tr>
<td>Shortcut</td>
<td>Ctrl+P</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Standard</td>
</tr>
</tbody>
</table>

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:

   ![Printer/Plotter Configuration](image)

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

   **Plot Area** determines which part of the drawing is printed:

   ![Plot Area](image)

   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**.

---

**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 drawing.

To make preview available, select a printer — any printer! — from the Printer/Plotter Configuration droplist:
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.
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.

![Plot offset](image1)

![Plot area](image2)

*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?

![Warning lights](image3)

*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.

![Printer configuration](image4)

› 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 to share drawings electronically. You can send them by email or post them to Web sites, without anyone needing CAD software to view them.

Another benefit: when drawings are saved in PDF format, they can be viewed but not altered. (Well, they can, but only with PDF editing software, which is not that common.)

To save drawings in PDF format in BricsCAD, follow these steps:

1. Enter the `ExportPDF` command.

   ```iscript
   : exportpdf
   ```

2. Notice the Export Drawing As dialog box. You can change the folder in which to save the file through the “Save In” field, as well as to change the file name ("File name"), such as `Yard-revision-1`.

   ![Export Drawing As dialog box](image)

   Click Save.

You can open the PDF file in Acrobat Reader or another program that displays `.pdf` files.

Specifying PDF Output Options

You control the way PDF files are produced by the Settings dialog box. The quick way to access PDF-specific options is with the `PdfOptions` command: it opens the 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:
What PDF Options Mean
Here is what the PDF output options mean (with default values shown in brackets):

- **PDF Embed Fonts** (on) embeds TTF font files in the PDF file:
  - **PDF TTF Text as Geometry** (off) converts text made with TTF fonts into geometry (lines and arcs)
  - **PDF SHX Text as Geometry** (off) converts text made with SHX fonts into geometry

- **PDF Simple Geometry Optimization** (on) reduces the file size by combining individual line segments into polylines, and uses Bezier control points to define splines. This option does not affect the drawing, only the resulting PDF file.

- **PDF Zoom to Extents Mode** (on) exports the drawing unscaled, so that its extents fit the page

- **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 and frozen

- **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.

(new in V20) Options related to PRC export mode were removed from BricsCAD V20. PRC is short for “Product Representation Compact.”

To export a 3D model as a 3D PDF file, you need to purchase and install the optional Communicator add-on, which reads and writes many 3D file formats. Use the Export command’s “3D PDF File” option.”

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).
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, such as a lawn, some trees, and a pond.

In this chapter, you learn to use BricsCAD’s intermediate commands like those that create ellipses, place hatch patterns, make arrays, and draw offset copies.
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 as 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Aliases</th>
<th>Menu Bar</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayClassic</td>
<td>ar</td>
<td>Modify</td>
<td>2D Array</td>
</tr>
<tr>
<td>Block</td>
<td>b</td>
<td>Tools</td>
<td>Create Block</td>
</tr>
<tr>
<td>Circle</td>
<td>c</td>
<td>Draw</td>
<td>Circle</td>
</tr>
<tr>
<td>Ellipse</td>
<td>el</td>
<td>Draw</td>
<td>Ellipse</td>
</tr>
<tr>
<td>Hatch</td>
<td>h</td>
<td>Draw</td>
<td>Hatch</td>
</tr>
<tr>
<td>Insert</td>
<td>i</td>
<td>Insert</td>
<td>Block</td>
</tr>
<tr>
<td>Offset</td>
<td>o</td>
<td>Modify</td>
<td>Offset</td>
</tr>
<tr>
<td>Pan</td>
<td>p, -p</td>
<td>View</td>
<td>Pan</td>
</tr>
<tr>
<td>PEdit</td>
<td>pe</td>
<td>Modify</td>
<td>Edit Polyline</td>
</tr>
<tr>
<td>RtZoom</td>
<td>...</td>
<td>View</td>
<td>Realtime</td>
</tr>
<tr>
<td>Zoom Window</td>
<td>z w</td>
<td>View</td>
<td>Zoom</td>
</tr>
</tbody>
</table>
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** (short for “polyline”) 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 cloud folder at [https://my.pcloud.com/publink/show?code=XZn98mkZ7T7f1pynxJYb7vDyVl6DwyqK1Ysk](https://my.pcloud.com/publink/show?code=XZn98mkZ7T7f1pynxJYb7vDyVl6DwyqK1Ysk).

Before starting to draw, first 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.

**THIS CHAPTER’S DRAWING**

By the end of this lesson, your drawing should look similar to this one:
2. Change the working layer to **Lawn**; its color is cyan (light blue). There are several ways to do this:
   - In the toolbar or ribbon interfaces, choose "Lawn" from the Layers droplist (shown below).
   - Or, from the Properties panel, choose "Lawn" from the Layers droplist
   - Or, open the Layers panel, and then click the **Current** field next to "Lawn"
   - Or in the status bar, right-click the current layer field...

   ![Layers Panel]

   ... and then choose "Lawn" from the list

   ![Layers Dropdown]

   - Or at the command prompt, use my favorite: enter the **CLayer** system variable (short for "current layer").
     This is a quick way to change to another layer when you know the name. In fact, this is the method used by the status bar. Enter **clayer**, and then the name of the layer, **lawn**:

   ```
   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.

3. To draw the polyline, start the **PLine** command.

   ```
   pline
   ```

4. 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:
   a. Hold down the **Shift** key on the keyboard.
   b. Press the right mouse button, and then let go of the **Shift** key. Notice the new menu that pops up on the screen. This is called a “shortcut menu.” The menu lists all of BricsCAD’s entity snap modes.
c. Move the cursor 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 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-foot intervals.)

   **TIP** 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.
8. Pick anywhere the line crosses the cursor’s aperture box. BricsCAD snaps the polyline precisely to the lower lot line.

9. Press Esc to end the PLine command.

Here is the reason you created the boundary as a polyline: you now use the PEdit command to easily modify the crooked segments into a smooth flowing curve.

**SMOOTHING POLYLINES**

<table>
<thead>
<tr>
<th>Command</th>
<th>PEdit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>PE</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
<tr>
<td>Mouse</td>
<td>Double-click a polyline</td>
</tr>
</tbody>
</table>

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.

1. To edit the polyline, begin the PEdit command (short for “polyline edit”):
   
   :edit

2. 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 **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 redrew the line segments as a **cubic Bezier curve** based on the polyline as the 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 a 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, 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.

---

**TIP**  If you cannot see the entity you want to edit, hold down the roller or middle button on the mouse and then move it. This pans 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 are at the 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:

   Drag grip to relocate end point
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
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.)

   New location for control point: `nea`

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 simply 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 name of the command! The table below lists the entities that react to double-clicking.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Command Executed by Double-clicking</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>ArrayEdit</td>
<td>Edits only associative arrays</td>
</tr>
<tr>
<td>Attribute Definition</td>
<td>DdEdit</td>
<td></td>
</tr>
<tr>
<td>Attribute Block</td>
<td>EAttEdit</td>
<td></td>
</tr>
<tr>
<td>Block</td>
<td>BEdit</td>
<td>Opens block in Block Editor environment</td>
</tr>
<tr>
<td>Camera</td>
<td>-View</td>
<td><em>NEW IN V20</em></td>
</tr>
<tr>
<td>Dimension</td>
<td>DdEdit</td>
<td>Edits only text of dimensions</td>
</tr>
<tr>
<td>Hatch</td>
<td>HatchEdit</td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>ImageAdjust</td>
<td>Adjusts brightness, contrast, fade of images</td>
</tr>
<tr>
<td>Polyline and Lwpolyline</td>
<td>PEdit</td>
<td></td>
</tr>
<tr>
<td>Multileader</td>
<td>DdEdit</td>
<td>Edits only text of mleaders</td>
</tr>
<tr>
<td>Section</td>
<td>ClipDisplay</td>
<td>Adjusts clipping planes of sections</td>
</tr>
<tr>
<td>Spline</td>
<td>SplinEdit</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>DdEdit</td>
<td></td>
</tr>
<tr>
<td>Tolerance</td>
<td>DdEdit</td>
<td>Edits only text tolerances</td>
</tr>
<tr>
<td>Viewport</td>
<td>VpMax</td>
<td><em>NEW IN V20</em></td>
</tr>
<tr>
<td>XRef</td>
<td>RefEdit</td>
<td>Opens xref in RefEditor environment</td>
</tr>
</tbody>
</table>

For all entities not listed in the table, BricsCAD opens the Properties panel when you double-click them. You use the Double Click Actions section of the **Customize** command’s Mouse tab to change which command is activated when specific entities are double-clicked; see our book *Customizing BricsCAD*. 

116  
Inside BricsCAD V20
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, shown below.

1. First, turn off the grid, since its lines can interfere with seeing the hatch pattern: on the status bar, click GRID so that the word turns gray.
2. Enter the Hatch command to start the Hatch command:
   : hatch
   Notice the Hatch and Gradient dialog box:
3. To find the grass hatch pattern, click the button to the right of “Name.”

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

5. Select Grass1, and then click OK to return to the Hatch and Gradient dialog box. Notice that Name changes to “Grass1.”

6. Ensure the Scale is set to 1 for metric drawings. (For imperial drawings, change the Scale to 25.) If the scale factor were too small, BricsCAD would refuse to draw the hatch pattern and complain:
   Hatch spacing too dense, or dash size too small.
   Please increase pattern scale or MAXHATCH value.

7. BricsCAD has a useful feature that searches a contiguous area, no matter how many different borders the area has. To employ this feature, click the Pick points in Boundaries button.

8. The dialog box disappears and BricsCAD prompts 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:

> The pattern floods incorrect areas — look out for patterns that leak into unwanted areas, or don’t appear at all

> The pattern is 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 \texttt{Hatch} command that operates at the command line. It is meant for use by macros and programming routines.

9. Press \texttt{Enter} to return to the Hatch and Gradient dialog box:

\begin{verbatim}
Select a point to define a boundary or hatch area: (Press Enter)
\end{verbatim}

After you press \texttt{Enter}, the Hatch and Gradient dialog box reappears.

10. Click \texttt{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 \texttt{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:

> Move \texttt{entities} that make up the boundary of a pattern, and the hatching automatically updates itself.

> Move, copy, and erase \texttt{hatch patterns}, because it acts like a block; it does not need to stay in one place.

> Click the hatch entity to bring up the \texttt{Properties} panel, and then change the color, pattern, scale factor, rotation angle, and so on.

> Double-click the hatch pattern to bring up the \texttt{Hatch Edit} dialog box, which looks identical to the Hatch and Gradient dialog box, but displays 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. This is known as a symbol, called a “block” by BricsCAD.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alias</strong></td>
<td>C</td>
</tr>
<tr>
<td><strong>Menu Bar</strong></td>
<td>Draw</td>
</tr>
<tr>
<td><strong>Ribbon</strong></td>
<td>Draw</td>
</tr>
<tr>
<td><strong>Toolbar</strong></td>
<td>Draw</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command</th>
<th>RtZoom</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Menu Bar</strong></td>
<td>View</td>
</tr>
<tr>
<td><strong>Ribbon</strong></td>
<td>View</td>
</tr>
<tr>
<td><strong>Shortcut</strong></td>
<td>roll the mouse wheel</td>
</tr>
<tr>
<td><strong>Toolbar</strong></td>
<td>View</td>
</tr>
</tbody>
</table>

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.)

**TIP** 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)
```

**TIP** To draw a perfectly horizontal or vertical line, hold down the **Shift** key. This action turns on ortho mode temporarily, for as long as you hold down the Shift key.

The **CENter** entity snap begins the line precisely at the center of the circle. The other end of the line extends beyond the edge of the circle.

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

3. Enter the **ArrayClassic** command:

```
: arrayclassic
```

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

<table>
<thead>
<tr>
<th>Array Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array Type</td>
<td>Polar</td>
</tr>
<tr>
<td>Select Entities</td>
<td>Click</td>
</tr>
<tr>
<td>Center</td>
<td>Click</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of items</td>
<td>15</td>
</tr>
<tr>
<td>Rotate items as copied</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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

1. To turn the tree symbol into a block, use the **Block** command:

```
: block
```

Notice that BricsCAD displays the Block Definition dialog box.

2. Enter the following parameters:

<table>
<thead>
<tr>
<th>Block Option</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Tree</td>
<td>You can give the block any name you like, up to 255 characters long</td>
</tr>
<tr>
<td>Base point</td>
<td></td>
<td>Click <em>Pick Point</em> button, and then choose the center of the circle</td>
</tr>
<tr>
<td>Entities</td>
<td>16</td>
<td>Click <em>Select Entities</em>, and then choose all 16</td>
</tr>
<tr>
<td>Convert to block</td>
<td>On</td>
<td>Converts the entities to a block when you click <em>OK</em></td>
</tr>
<tr>
<td>Scale uniformly</td>
<td>On</td>
<td>Forces the circle to remain round</td>
</tr>
</tbody>
</table>

a. 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 CENter 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)

<table>
<thead>
<tr>
<th>Command</th>
<th>Insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>I</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Insert</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Insert</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Insert</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Insert Options</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insertion Point</td>
<td>Specify On-screen (on)</td>
</tr>
<tr>
<td>Scale</td>
<td>Specify On-screen (on)</td>
</tr>
<tr>
<td>Rotation</td>
<td>Specify On-screen (off)</td>
</tr>
<tr>
<td>Angle: 0</td>
<td></td>
</tr>
</tbody>
</table>

   ![Insert Block Dialog Box](image-url)
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 <1.000000>: 2
```

The x scale factor of 2 draws the block twice as large as the original. Because you drew the original symbol one meter in diameter, the newly inserted tree has a diameter of two meters. If you 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](image1.png)

![Right: Many more tree blocks inserted into the drawing](image2.png)

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:
   
   ```plaintext
   : 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:
   
   ```plaintext
   Endpoint of axis: (Move the cursor) 4.5
   ```

4. And the pond is five feet wide:
   
   ```plaintext
   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 0.3 m (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.
ARRAY PATHS

Another type of array repeatedly places objects 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 circles.

You use this construction technique to lay a path of paving stones from the house to the pond. The path array needs a guide path, an entity along which to place the arrayed entities, and so in the first part of the tutorial, you draw the guide path as a spline (Spline command). The paving stones will be made of polygons (Polygon command), and then arrayed along the spline (Array-Path command).

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.
Drawing Splines

<table>
<thead>
<tr>
<th>Command</th>
<th>Spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>spl</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Draw</td>
</tr>
</tbody>
</table>

Splines were first used in ship building to make hulls from smooth flowing strips of wood. Hundreds of years later, we still use splines to draw smooth curves, but now 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 preferences is to use the “clayer road” shortcut shown below.) This means the paving stones are placed on the same layer as the driveway.

```
: clayer
New current value for CLAYER: road
```

2. Draw the guide path using a spline. 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 in some cases be an easier approach than using the Spline command directly.

**Zooming to Objects**

The Zoom command has a handy option that lets you zoom to a single entity. Here is how it works:

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

```plaintext
: 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: l
```

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):

```plaintext
: zoom .9x
```

**Paving Stones from Polygons**

<table>
<thead>
<tr>
<th>Command</th>
<th>Polygon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>pol</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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:

a. The first thing you do is specify the number of sides. For a hexagon, you specify six sides:

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

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

```plaintext
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.
Adding Details to Drawings

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.

---

**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
ArrayPath is one of a trio of commands that create associative array. (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). Here’s the benefit: when you change the path, the array changes to suit.

1. Start the **ArrayPath** command, and then select the polygon, as follows:
   
   ```
   : arraypath
   Select entities to array: (Pick the polygon)
   ```

2. Press **Enter** to exit the selection process, and then pick the spline:

   ```
   Entities in set: 1
   Select entities to array: (Press Enter)
   Type = Path, Associative = Yes
   Select the path curve: (Pick the spline)
   ```

---

### QUICK SUMMARY OF POLYGON

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

```
: polygon
Polygon: 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
3. Now you see many paving stones along the path.

---

**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
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.

![Image](image-url)

---

### 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.
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 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.
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

<table>
<thead>
<tr>
<th>Command Aliases</th>
<th>Menu Selection</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dist</td>
<td>di Tools</td>
<td>Inquiry</td>
</tr>
<tr>
<td>Lengthen</td>
<td>len Modify</td>
<td>Lengthen</td>
</tr>
<tr>
<td>Linetype</td>
<td>lt Settings</td>
<td>Linetypes</td>
</tr>
<tr>
<td>List</td>
<td>li Tools</td>
<td>Inquiry</td>
</tr>
<tr>
<td>Properties</td>
<td>pr Modify</td>
<td>Properties</td>
</tr>
<tr>
<td>Stretch</td>
<td>s Modify</td>
<td>Stretch</td>
</tr>
</tbody>
</table>

THIS CHAPTER’S DRAWING

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

![House MText in a bounding box](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,” even though they apply to nearly all 2D entities, such as also to circles and polylines. 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 entity(ies) you want to have the new look.

LOADING LINETYPES

<table>
<thead>
<tr>
<th>Command</th>
<th>Linetype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>lt</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Tools</td>
</tr>
<tr>
<td>Status Bar</td>
<td>Linetype</td>
</tr>
</tbody>
</table>

Start BricsCAD, and then open your copy of yard.dwg. (Or open the tutorial-05.dwg file or Imp-tutorial-05.dwg in imperial units.)

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)>:

   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)>:

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. Drawing Explorer is a master dialog box that references many aspects of drawings, such as the way text and dimensions look.

3. To add a linetype to this drawing, click the **New** button. BricsCAD doesn’t actually create new linetypes, as suggested by the name of this button. Instead, the button leads you to a dialog box from which you load pre-defined linetypes.

Notice the Load Linetypes dialog box.

The drawing already contain some linetypes; all new drawings do. Here is what the 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

TIP: Although BricsCAD uses terms like *linetypes* and *line weights*, these apply to nearly any kind of entity, such as circles and arcs — not just 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.

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 unaffected.
   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.

When the Properties panel or the Entity Properties toolbar is open, notice that the default linetype changes to Border:

![Properties panel](image)

**CHANGING PROPERTIES**

<table>
<thead>
<tr>
<th>Command</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>pr</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
<tr>
<td>Right-click</td>
<td>Properties</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Standard</td>
</tr>
</tbody>
</table>

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

- **When No Entities are Selected** — 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 — 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.

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
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:
   ```plaintext
   : 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.

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

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.

![Incorrect linetype scale; above: Correct linetype scale.](image)

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

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

**Warning!** Setting linetype scale is one of the trickiest aspects of BricsCAD. Linetypes 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**.
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:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Abbreviation</th>
<th>Selects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>[pick]</td>
<td>A single entity</td>
</tr>
<tr>
<td>All</td>
<td>ALL</td>
<td>All non-frozen entities; can also press Ctrl+A</td>
</tr>
<tr>
<td>Previous</td>
<td>P</td>
<td>Most recently selected entity</td>
</tr>
<tr>
<td>Last</td>
<td>L</td>
<td>Most-recently created entity visible on the screen</td>
</tr>
<tr>
<td>Window</td>
<td>W</td>
<td>All entities inside a rectangular window</td>
</tr>
<tr>
<td>Crossing</td>
<td>C</td>
<td>All entities crossing and within a rectangular window</td>
</tr>
<tr>
<td>Outside</td>
<td>O</td>
<td>All entities outside the rectangular window</td>
</tr>
<tr>
<td>Window Polygon</td>
<td>WP</td>
<td>All entities within a polygon window</td>
</tr>
<tr>
<td>Crossing Polygon</td>
<td>CP</td>
<td>All entities crossing and within a polygon</td>
</tr>
<tr>
<td>Outside Polygon</td>
<td>OP</td>
<td>All entities outside a polygon window</td>
</tr>
<tr>
<td>Window Circle</td>
<td>WC</td>
<td>All entities inside a circular window</td>
</tr>
<tr>
<td>Crossing Circle</td>
<td>CC</td>
<td>All entities crossing and inside a circular window</td>
</tr>
<tr>
<td>Outside Circle</td>
<td>OC</td>
<td>All entities outside a circular window</td>
</tr>
</tbody>
</table>
| Box              | B            | All entities depending on how direction selection window is drawn: 
|                  |              | • 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
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:

   ![Quick Select Mode](image)

   ...continued

**SELECTION BY PROPERTIES**

The **PROperties** option selects entities by common properties:

<table>
<thead>
<tr>
<th>Properties</th>
<th>Abbreviation</th>
<th>Selects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>C</td>
<td>All entities of the same color name or number such as “red” or “122”</td>
</tr>
<tr>
<td>Layer</td>
<td>LA</td>
<td>All entities with the same layer name</td>
</tr>
<tr>
<td>Ltype</td>
<td>LT</td>
<td>All entities of the same linetype</td>
</tr>
<tr>
<td>Name</td>
<td>N</td>
<td>All entities of the same name</td>
</tr>
<tr>
<td>Thickness</td>
<td>TH</td>
<td>All entities of the same thickness</td>
</tr>
<tr>
<td>Type</td>
<td>TY</td>
<td>All entities of the same entity type, such as “circle”</td>
</tr>
<tr>
<td>Value</td>
<td>V</td>
<td>All entities of the specified handle</td>
</tr>
<tr>
<td>Width</td>
<td>W</td>
<td>All entities of the specified width</td>
</tr>
</tbody>
</table>
> 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.

![Properties Bar](image)

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.

![Properties Bar](image)

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Lengthen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>len, editlen</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
</tbody>
</table>

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

---

**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.
4. Curiously, BricsCAD asks you a second time to select the entity; for some reason, it does not automatically change the length of the line you choose first:

   Mode/<Select entity to change>: (Pick the same line)

5. Notice that the line lengthens. Press **Enter** to end the command.

   Mode/Undo/<Select entity to change>: (Press **ENTER**)

---

**Stretching the Pond**

<table>
<thead>
<tr>
<th>Command</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>s</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Change</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Modify</td>
</tr>
</tbody>
</table>

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.

3. After picking two corners of a rectangle that covers part of the pond, press **Enter**:

   Select entities to stretch by crossing-window or crossing-polygon: *(Press ENTER to end entity selection)*

   If the pond were entirely inside the entity selection rectangle, the Stretch command would move the pond, not stretch it. If you accidentally select entities other than the pond, type the R option (short for “remove from selection set”), and then select the entities to remove.

4. To tell BricsCAD how much you want the pond stretched, pick two points that indicate the distance:

   Base point of displacement: *(Pick a point near the pond)*
   Second point of displacement: *(Pick a point away from the pond)*

   You have now created a whole new look to your pond! (Your pond will probably look different from the one illustrated in this book.) Note that you cannot create this effect using grips editing.

5. If you don’t like it, you can undo the stretch with the **U** command, as follows:

   U: STRETCH
   ... and try stretching the pond again.

---

### MOVING ENTITIES

<table>
<thead>
<tr>
<th>Command</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>m</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Change</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Modify</td>
</tr>
</tbody>
</table>

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

1. Start the **Move** command by entering the letter ‘m’, as follows:

   : m

   Here you used another of BricsCAD’s shortcuts. **M** is the alias (abbreviation) for the Move command.

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

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.
6. 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.
7. 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.

a. Here is how to get just the x coordinate:

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

b. 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!

c. 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.

8. Exit the **PLine** command:

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

### LISTING DATA

<table>
<thead>
<tr>
<th>Command</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>li</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Tools</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Tools</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Inquiry</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>di</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Tools</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Tools</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Inquiry</td>
</tr>
</tbody>
</table>

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:

\textbf{Perimeter} \ 77.0

So, 77 meters (roughly 260 feet). Now you know how much fencing you would need.

2. Press function key \textbf{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 \textbf{Dist} command (short for “distance”). To find the shortest distance from the house to the pond with the Dist command, from the \textbf{Tools} menu bar, select \textbf{Inquiry | Distance}:

\begin{verbatim}
: _dist
Starting point for distance: \texttt{nea}
Snap to nearest of: (Pick inside edge of pond.)
End point: \texttt{per}
Snap to perpendicular of: (Pick house wall.)
\end{verbatim}

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.
With the yard plan nearly finished, now is the time to add callouts (or text) to describe the parts of the yard and dimensions to show its size. In this chapter, you learn how to add the 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

Annotative — 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcuts</th>
<th>Menu Bar</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>DdEdit</td>
<td>ed</td>
<td>...</td>
<td>Draw</td>
</tr>
<tr>
<td>DimAligned</td>
<td>dal, dimali</td>
<td>Dimension</td>
<td>Aligned</td>
</tr>
<tr>
<td>DimBaseline</td>
<td>dba, dimbase</td>
<td>Dimension</td>
<td>Baseline</td>
</tr>
<tr>
<td>DimContinue</td>
<td>dco, dimcont</td>
<td>Dimension</td>
<td>Continue</td>
</tr>
<tr>
<td>DimEdit</td>
<td>ded, dimed</td>
<td>Modify</td>
<td>Edit Dimension Text</td>
</tr>
<tr>
<td>DimLinear</td>
<td>dli, dimlin</td>
<td>Dimension</td>
<td>Linear</td>
</tr>
<tr>
<td>DimRadius</td>
<td>dra, dimrad</td>
<td>Dimension</td>
<td>Radius</td>
</tr>
<tr>
<td>MText</td>
<td>t</td>
<td>Draw</td>
<td>Multiline Text</td>
</tr>
<tr>
<td>QText</td>
<td>qt</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Rotate</td>
<td>ro</td>
<td>Modify</td>
<td>Rotate</td>
</tr>
<tr>
<td>Style</td>
<td>st</td>
<td>Settings</td>
<td>Text Style</td>
</tr>
<tr>
<td>Text</td>
<td>tx</td>
<td>Draw</td>
<td>Text</td>
</tr>
</tbody>
</table>
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.
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.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>MText</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>MT, T</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Tools</td>
</tr>
</tbody>
</table>

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

Just as in a word processor, BricsCAD lets you format individual words and letters with **boldfacing** and **italics**, apply different fonts and colors, and include special symbols, such as the center line and the degree symbols. You can define indents for the margin (such as for the starts of paragraphs) and where tabs should be.

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

BricsCAD places multiline text in a rectangle, much like page design software does. In the figure below, the rectangle is shown in green.

---

**TIP** 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.

---
The paragraphs of text are constrained by the rectangle’s top and two sides. This allows you to make the text fit specific areas in drawings, such as in title blocks and note blocks. As you add more text to the block, it flows down.

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, taller or shallower. You can move that entire block of mtext to any where in the drawing.

Place some mtext in the drawing now:

1. Start the MText 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.
4. 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 starts at the top of the bounding box. The right edge of the box will determine where the lines of text wrap — where the text flows to the next line. (For right-to-left text like Hebraic, it is the left edge that determines the line wrap location.) Adjusting the bounding box is like setting the margins in word processing software.

The bottom of the boundary is ignored, usually. When there is too much text for the box, BricsCAD automatically extends the bounding box downwards.

The bottom is used in only one case: when you want text to flow upwards, such as with revision notes. You change the direction of the flow with the command’s Direction option.
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](http://www.papersizes.org/a-paper-sizes.htm).
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 standard for the height of text is 10mm (or 3/8" in imperial units) tall in drawings. But if text were placed at that size, it would be nearly invisible when printed; 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 in this particular drawing it has to be larger, much larger.

To figure out how tall to make the text, read the boxed text on the next page, Quick Summary on Calculating Text Heights. Based on the size of this drawing, the text height needs to of 400mm tall (or 18" in imperial units).

4. After you pick the second point, BricsCAD displays the Text Formatting dialog box, in which you specify font settings. 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 (meters or 400mm).
   c. Press Enter. The height is set.

5. Type House, and then press Enter. Notice that the text appears in the drawing.

6. Click OK to exit the MText command.

CREATING TEXT STYLES

Command  Style
Aliases    st, dstyle, 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

   ![Screen capture of the Drawing Explorer dialog box with the Standard style selected.](image)

   Abc Def Ghi

2. Make the following changes to the Standard style:

<table>
<thead>
<tr>
<th>Property</th>
<th>Old Value</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Font Name</td>
<td>Arial</td>
<td>Times New Roman</td>
</tr>
<tr>
<td>Height</td>
<td>0</td>
<td>0.75</td>
</tr>
</tbody>
</table>

   (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.

   **TIP** A height of 0 has a special meaning in BricsCAD: 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:

   ![House text in Times New Roman font.](image)

   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.

   ![Diagram showing the baseline and ascenders.](image)
BricsCAD has another command for placing text that I find more efficient than MText. The **Text** command places a single line of text at a time. This makes it ideal for placing lots of bits of text all over the drawing, 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:

   ```plaintext
   : text
   ```

2. Select the **Align** justification mode, which places the text fitted between two points that you pick:

   ```plaintext
   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 ask you for a rotation angle, as the command normally would with most other justification options.

3. The Text command goes straight to the prompt:

   ```plaintext
   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.

   ```plaintext
   Text: (Press ENTER)
   ```
Changing 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:

<table>
<thead>
<tr>
<th>Justification</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start point</td>
<td>Baseline left</td>
</tr>
<tr>
<td>Align</td>
<td>Fitted between two points</td>
</tr>
<tr>
<td>Fit</td>
<td>Fitted with constant text height</td>
</tr>
<tr>
<td>Center</td>
<td>Baseline center</td>
</tr>
<tr>
<td>Middle</td>
<td>Exact center of text</td>
</tr>
<tr>
<td>Right</td>
<td>Baseline right</td>
</tr>
<tr>
<td>TL</td>
<td>Top left</td>
</tr>
<tr>
<td>TC</td>
<td>Top center</td>
</tr>
<tr>
<td>TR</td>
<td>Top right</td>
</tr>
<tr>
<td>ML</td>
<td>Middle left</td>
</tr>
<tr>
<td>MC</td>
<td>Middle center (equivalent to Middle)</td>
</tr>
<tr>
<td>MR</td>
<td>Middle right</td>
</tr>
<tr>
<td>BL</td>
<td>Baseline left (equivalent to Start point)</td>
</tr>
<tr>
<td>BC</td>
<td>Baseline center (equivalent to Center)</td>
</tr>
<tr>
<td>BR</td>
<td>Baseline right (equivalent to Right)</td>
</tr>
</tbody>
</table>
1. Double-click the word **Pool**. This action activates the **DdEdit** command (short for "dynamic dialog editor"). Notice the Edit Text dialog box.

![Edit Text dialog box](image)

2. Replace "Pool" with **Pond**.

3. Click **OK**. Notice that the word is corrected in the drawing.

![Corrected word](image)

---

**PLACING ROTATED TEXT**

Now we'll add more text, this time placing it sideways. 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)

![Rotated text](image)

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.
Changing the Angle of Rotated Text

If you placed the text at the wrong rotation angle, you change it after the fact.

1. Use the Zoom Object command to take a closer look at the text.
2. Now pick the text...
   : (Pick text)
3. ...and then right-click the text. From the shortcut menu, choose Properties.

![Properties palette](image1.png)

4. In the Properties palette:
   a. Look for the Rotation field under the Text section.
   b. Change 90 to 270, and then press Tab.

![Rotation setting](image2.png)

5. 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:

1. Use the Text command to add multiple callouts to the drawing.
   
   : text
   
   Text: Style/Align/Fit/Center/Middle/Right/Justify/<Start point>: (Pick near the bottom of the drawing)

2. Remember to change the rotation angle back to 0 degrees.
   
   Rotation angle of text <90>: 0

3. Enter two lines of text, as follows:
   
   Text: 4486 Donlyn Avenue (Press Enter)
   Text: Anytown BC (Press Enter)
   Text: (Press Enter 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**

<table>
<thead>
<tr>
<th>Command</th>
<th>Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard Shortcut</td>
<td>Ctrl+F</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Edit</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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 dialog box offers two tabs with results:

- **Found Text** tab — displays the current instance of found text
- **List Results** tab — shows all instances

When there is too much text in the drawing, you can narrow the search by searching only a specific area of the drawing. Click the **Search In** droplist and then choose from...

- **Current Selection** — searches for the text among currently selected entities
- **Current Layout** — searches for text only in the current layer, either Model space or a specific layout
- **Entire Drawing** — searches the entire drawing
Or, you can limit the search to a specific type of text. Click the **Options** button, and then choose the type of text from the list presented in the dialog box:

![Find and Replace Options dialog box](image)

<table>
<thead>
<tr>
<th>Text Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Attribute Value</td>
<td>Attribute text stored in blocks</td>
</tr>
<tr>
<td>Dimension Annotation Text</td>
<td>Text in dimensions, including leaders</td>
</tr>
<tr>
<td>Text (MText, DText, Text)</td>
<td>All regular kinds of text, including field text</td>
</tr>
<tr>
<td>Table Text</td>
<td>Text found in the cells of tables</td>
</tr>
<tr>
<td>Hyperlink</td>
<td>Text in URLs (Web links)</td>
</tr>
<tr>
<td>Hyperlink Description</td>
<td>Text in the descriptions of hyperlinks</td>
</tr>
</tbody>
</table>

And then there are even more options for narrowing the searches further:

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

With callouts firmly placed in the drawing, let's turn to dimensioning. This is done with commands that start with “Dim” (short for “dimensioning”).

Now, a drawing properly created would not need dimensions. You can use the CAD program’s Dist command to determine distances and angles, but this gets tedious. It really is much easier to see all important distances at a glance by dimensioning drawing. And so today, in this computer age, we still place dimensions in drawings.

Dimensions are made up of several elements, and the CAD program takes care of drawing all of them for us. The names of elements employed by all linear dimensions are shown below.

![Dimensions in Drawings Diagram]

The bad news is that dimensions require dozens and dozens and dozens of settings to specify the look. Is the dimension text above, below, or in-line with the dimension line, centered or not centered; if above, how far above. And in what color? The list of options runs to over a hundred.

The settings are needed so that drawings to adhere to drafting standards set by standards organizations. The good news is that BricsCAD comes pre-configured with most settings set to reasonable values, in both metric and imperial units.

Settings are controlled by dimensioning variables, or “dimvars” for short. You can change dimvars in several ways:

- Adjusting values in the Dimension Styles section in the Drawing Explorer dialog box
- Directly entering their names at the ‘:’ prompt
- After the dimension is placed, aspects of it can be changed with dimension editing commands

In the following tutorials, you use both methods.

**Tip** BricsCAD draws all components of dimensions automatically, according to the dimvar settings.
PREPARING DRAWINGS FOR DIMENSIONING

**System Variable**  
**DimScale**

Like text, hatches, and linetypes, the scale of a dimension is relative to the printed size. Otherwise, arrowheads and text will be too small or too large to read. Before any drawing dimension, you should set the scale, like this:

1. To set the dimension scale, enter the **DimScale** dimvar like a command:
   
   \[ \text{dmscale} \]

2. The same scale factor that you calculated for text works here: 4 for metric units (or 142, for imperial units).
   
   Enter new value for DIMSCALE <1>: 4

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

3. With the **Layer** command, create a new layer named “Dims” with black color and continuous linetype.
4. Make the layer current.
5. If you wish, freeze unnecessary layers, such as Text, Lawn, Plants, and Pond. This keeps them from getting in the way of your dimensioning work. (For the illustrations below, I left some of these layers on so to make it easier to see the locations in the drawing where the dimensions are being applied.)
6. Because dimensioning often takes place at intersections, turn on **INTERsection** entity snap mode, as follows:
   
   \[ \text{intersection} \]

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

By tradition, most dimensions are either horizontal or vertical. In rare cases, they are placed at an angle (“rotated”).

---

**Example Diagram**

- Dimensions:
  - 2831
  - 1290
  - 2520
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.

BricsCAD determines how to construct the dimension from how you pick points in the drawing. Start DimLinear, pick a first point, then move the cursor, and you see the dimension jumping between horizontal and vertical. Depending on how you pick the second point, a horizontal or vertical dimension is drawn by BricsCAD:

- **Horizontal dimensioning** — pick two points that are roughly horizontal, BricsCAD draws the dimension perfectly horizontally
- **Vertical dimensioning** — pick two points roughly vertically, BricsCAD draws the dimension exactly vertical
- **Rotated dimensioning** — enter R to tell BricsCAD that you want to place the dimension at an angle, and BricsCAD prompts you to specify the angle. (A more useful command for drawing dimensions at an angle, 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)

   ![Diagram](image)

   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.

   **TIP** The DimLinear command has several options
   - **Angle** — prompts you for a rotation angle for the text of the dimension
   - **Text** — lets you modify or replace the dimension text.
   - **Horizontal** — forces the dimension line to be horizontal
   - **Vertical** — forces the dimension line to be vertical
   - **Rotated** — prompts you for a rotation angle to the dimension line

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)
```

5. To end the command, it is important that you press **Esc**, because pressing Enter makes BricsCAD prompt you to choose the starting dimension.

```
  Continue: ENTER to select starting dimension/<Origin of next extension line>: (Press ESC to end the command)
```

Because BricsCAD knows where the last extension line was, all it now needs to know the location of the next extension line to draw in the second dimension.

**VERTICAL AND BASELINE DIMENSIONS**

As 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.

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

```
  : _dimlinear
  ENTER to select entity/<Origin of first extension line>: (Pick #7)
  Origin of second extension line: (Pick #8)
  Angle/Text/Orientation of dimension line: Horizontal/Vertical/Rotated: (Pick #9)
```

2. Use **DimContinue** to continue the vertical dimensions along the right side of the lot at points 10 and 11.

```
  : dimcont
  Continue: ENTER to select starting dimension/<Origin of next extension line>: (Pick #10)
  Continue: ENTER to select starting dimension/<Origin of next extension line>: (Pick #11)
  Continue: ENTER to select starting dimension/<Origin of next extension line>: (Press Esc)
```
Baseline Dimensioning

- **Command**: DimBaseline
- **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**

<table>
<thead>
<tr>
<th>Command</th>
<th>DimAligned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>dal, dimali</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Dimension</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Dimension</td>
</tr>
</tbody>
</table>

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)
```

![Diagram of aligned dimension](image)

The dimension you draw may look different, depending on how the angled line was drawn.

**Radial Dimensions**

<table>
<thead>
<tr>
<th>Command</th>
<th>DimRadius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>dra, dimrad</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Dimensions</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Dimensions</td>
</tr>
</tbody>
</table>

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.

   ![Image of a circle with a radius dimension](image)

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

### DIM, THE SUPER DIMENSION COMMAND

<table>
<thead>
<tr>
<th>Command</th>
<th>Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>dimension</td>
</tr>
</tbody>
</table>

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/R0tated/CEnter/
   Diameter/RAdius/Baseline/COntinue/ORdinate/Position/Distribute/UPdate dimensions/variable
   STatus/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:
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.

   ![House diagram with dimension lines]

   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.)

---

### 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.
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.”

![Drawing Explorer](image1)

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.

![Hatch and Gradient dialog box](image2)

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. In the following tutorial, I have you switching back and forth between model and layout modes. Layout mode shows you what the drawing looks like on the paper you’ll be printing it on; it is like an interactive print preview mode. (I’ve frozen the Dims layer so that the dimensions do not clutter the drawing.)

You find tabs for switching between the model and layout modes at the bottom of the drawing area.

1. Open the Imp-Tutorial-06.dwg file.
2. Switch to layout mode by clicking the Layout1 tab.

Notice that a layout looks different from the default drawing area (a.k.a. model space):

- White rectangular area represents the paper size
- Dashed rectangle is the margin, the area at the edge of the paper on which the printer cannot print
- Black rectangle is the viewport, in which you see the model. In the figure below, the viewport has green grips and dashed lines, because I have selected it

3. When the drawing is first opened in a viewport, it most likely is 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
4. 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 **Annotation 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.

   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. 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.

 **TIP** Why place the annotative text in model tab? (Why not in the layout tab?) My CAD-expert friend Bill Fane explains, “If you zoom and pan while entering annotative text in a layout tab, you mess up the viewport’s scale; when you go back into model space, the annotations are missing, because their scale no longer matches the scale of the viewport. For this reason, it is best to enter annotative text in Model tab.”

b. Enter the **Text** command to place regular text with the “Standard” style (which is non-annotative) at a height of 1/8” (0.125”):

```text
: text
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
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”.

```text
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.
**TIP** If the drawing does not contain an annotative text style, then you can create one like this:

1. Enter the **Style** command.
2. Make a copy of an existing style by clicking the **New** button.
3. Click the **Annotative** option. It’s that simple!

   ![Annotative Text Style](image)

   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”)**

   ![Properties Panel](image)
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:

![Diagram of a leader showing Knee, Landing, Arrowhead, Leader line, and Annotation (mtext)]

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

**SINGLE-LINE LEADERS**

<table>
<thead>
<tr>
<th>Command</th>
<th>QLeader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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)
   ```
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
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.

---

**MULTI-LINE LEADERS**

<table>
<thead>
<tr>
<th>Command</th>
<th>MLeader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Bar</td>
<td>Dimension</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Dimensions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>MLeaderEdit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Bar</td>
<td>Dimension</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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)

---

**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.
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)
```

![Diagram of stepping stones and pond with leader lines]

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.

---

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.
One of the most powerful aspects of CAD is its ability to generate information from drawings. You saw a hint of this in an earlier lesson when you used the Dist and List commands to find distances, lengths, areas, and totals of them.

In this chapter, you learn how to add information to drawings, and then extract it to create a bill of materials (BOM) in the drawing and in a spreadsheet.

**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

<table>
<thead>
<tr>
<th>Command</th>
<th>Aliases</th>
<th>Menu Bar</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttDef</td>
<td>at</td>
<td>Tools</td>
<td>Attributes</td>
</tr>
<tr>
<td>AttDisp</td>
<td>ad</td>
<td>Tools</td>
<td>Attributes</td>
</tr>
<tr>
<td>bmBalloon</td>
<td>...</td>
<td>Assembly</td>
<td>Balloon</td>
</tr>
<tr>
<td>bmBom</td>
<td>...</td>
<td>Assembly</td>
<td>Bill of Materials</td>
</tr>
<tr>
<td>bmExplode</td>
<td>...</td>
<td>Assembly</td>
<td>Assembly Explode</td>
</tr>
<tr>
<td>DataExtraction</td>
<td>...</td>
<td>Tools</td>
<td>Data Extraction</td>
</tr>
<tr>
<td>DataLink</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>DataLinkUpdate</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Donut</td>
<td>do, doughnut</td>
<td>Draw</td>
<td>Donut</td>
</tr>
<tr>
<td>Table</td>
<td>...</td>
<td>Draw</td>
<td>Table</td>
</tr>
</tbody>
</table>
About Attribute Data

CAD can be used for many different kinds of drawings, floor plans, building construction, mechanical design, electrical layouts, and so on. Drawings usually are made from standard parts, such as desks, windows, bolts, and switches. When the parts contain attribute data, then drawings become truly useful.

In this chapter, you create a schematic drawing of an automobile's electrical system. The drawing contains numerous components, such as 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’d miss a few and end up with the wrong total. (Well, you might not miscount, but I know I would.)

It’s faster and 100% accurate to let the 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 then be imported into a spreadsheet to count the components, determine costs, and so on. As well, the parts count can be placed in drawings as tables.

THIS CHAPTER'S DRAWING

During this lesson, you work with the following drawing file, electric.dwg.
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”).

The attribute extraction process progresses through these steps:

- **Step 1: Define Attributes** — define attributes with the AttDef command
- **Step 2: Add Attributes** — use the Block command to merge attributes into symbols, the the Insert command to add the attributed blocks to the drawing
- **Step 3: Export Attributes** — export the attributes with the DataExtraction command to a file or to a table in the drawing

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://my.pcloud.com/publink/show?code=XZn98mkZ7T7fypynxJYb7vDyVl6DwyqK1Ysk](https://my.pcloud.com/publink/show?code=XZn98mkZ7T7fypynxJYb7vDyVl6DwyqK1Ysk).

## Creating Blocks with Attributes

The example drawing for this lesson is part of the electrical schematic of an antique automobile. See the drawing shown in the previous page. (Wondering which auto? It’s adapted from the electrical system of my first car, the 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 a single block and then attach attributes to it. Working with attributes is pretty tedious, which is why I won’t get you to create all of the blocks. I’ve put them into a drawing file that you can download.

1. Start BricsCAD with a new drawing using the 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
      ```
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Donut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>doughnut</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command</th>
<th>AttDef</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Bar</td>
<td>Tools</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Insert</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Tools</td>
</tr>
</tbody>
</table>

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.

**TIP** Attributes attach only to blocks. It is possible to add data to other kinds of entities, but the process is more technical, because it involves xdata (short for “extended entity data”). Nevertheless, BricsCAD can export all data — such as 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.

2. In the **Attribute** section, enter the following text:

<table>
<thead>
<tr>
<th>Attribute Section</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>Product</td>
<td>Identifies the attribute to the program</td>
</tr>
<tr>
<td>Prompt</td>
<td>Product</td>
<td>Prompts the user about the data to enter</td>
</tr>
<tr>
<td>Default</td>
<td>Solder Connection</td>
<td>Default value of the data; can be overridden by the user</td>
</tr>
</tbody>
</table>

3. In the Text section of the dialog box, set the **Height** (of the text) to a tiny, unobtrusive value, such as 0.01. You don’t actually want to see 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.)

<table>
<thead>
<tr>
<th>Text Section</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>.01</td>
<td>Specifies the height of the attribute text</td>
</tr>
</tbody>
</table>

4. 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:

   a. In the Insert Coordinates section, click the button. The dialog box disappears temporarily.

   b. In the command bar, notice that BricsCAD prompts you for the location of the insertion point.
      
      Specify insertion point: *(Pick a point to the right of the donut)*

   c. Pick a point to the right of the connection block:

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

![Define Attribute dialog box](image)

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

![PRODUCT tag](image)

**TIP** Although blocks can contain multiple attributes, the **AttDef** command creates just one attribute at a time. You have to repeat the command to define additional attributes, up to a maximum of 245.

**ADDING MORE ATTRIBUTES**

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

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

   _attdef

   And the Define Attribute dialog box reappears.
2. Fill in the following data in the Attribute section:

<table>
<thead>
<tr>
<th>Attribute Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>Stockno</td>
</tr>
<tr>
<td>Prompt</td>
<td>Stock No.</td>
</tr>
<tr>
<td>Value</td>
<td>000-0000</td>
</tr>
</tbody>
</table>

Ensure that the tag name “Stockno” is a single word, with no spaces.

3. Click the ![button](image.png) 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](image.png). 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>b</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Tools</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Insert</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Tools</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Block Section</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Connect</td>
<td>Identifies the block to you and the program</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retain</td>
<td>Retains the entities after defining them as a block</td>
</tr>
<tr>
<td>Convert to Block</td>
<td>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</td>
</tr>
<tr>
<td>Delete</td>
<td>Erases the entities after defining them as a block; this option saves using the Erase command to later remove the entities</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Attribute Field</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insertion Point</td>
<td>✓ Specify On-screen</td>
<td>Point at which block is inserted in drawing</td>
</tr>
<tr>
<td>Scale X</td>
<td>(off; scale = 1)</td>
<td>Size of the block</td>
</tr>
<tr>
<td>Rotation</td>
<td>(off; angle = 0)</td>
<td>Angle at which the block is placed</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>File Extension</th>
<th>Command Activated</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drawing.dwg</td>
<td>Open command</td>
<td>Inserted as a block</td>
</tr>
<tr>
<td>Raster images</td>
<td>ImageAttach command</td>
<td>Attached as a raster image</td>
</tr>
<tr>
<td>Unknown file types</td>
<td>InsertObj command</td>
<td>Inserted as an OLE entity (Windows only)</td>
</tr>
</tbody>
</table>

**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 prepared a drawing to be used by the next set of tutorials. It contains all needed blocks and attributes. Get your copy of the `electric.dwg` file from my cloud folder: [https://my.pcloud.com/publink/show?code=XZn98mkZ7f1pynxJYb7vDyVl6DwyqK1Ysk](https://my.pcloud.com/publink/show?code=XZn98mkZ7f1pynxJYb7vDyVl6DwyqK1Ysk).
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 of data with a character, such as a comma. Software like spreadsheets and databases recognize the comma as the separator, and place the pieces of data into their own cells or fields. This make CSV a kind of universal exchange format for data.

Once the data from the CSV file is in a spreadsheet or database program, 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
   ```

   **TIP**  The **AttDisp** command changes the way BricsCAD displays attribute text:
   - Normal — displays attributes, except for those set to Invisible mode.
   - On — displays all attributes, including invisible ones.
   - Off — hides all attributes.

3. Start the attribute extraction entering the **DataExtraction** command.
   ```
   : dataextraction
   ```
Notice the Data Extraction dialog box.

4. Click the **Select** button 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:

   ```
   [Check boxes for **Extract entities from blocks**, **Extract entities from parts**, and **Add parts to block counts**]
   ```

   You are only interested in extracting attribute data from blocks, so turn off all options. The section now looks like this:

6. The second set of options specify files that are needed. Let's go through them one by one:

   **Create Data Extraction Definition** — I recommend that you turn on this option.

   When 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: `electric.csv`
- 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).

7. Click the **Delimited** drop list, 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.)

8. Click **Next**.

Notice that the Page 2 dialog box lists the name of blocks and of line entities. You want data about all of them extracted, so leave everything as it is. Even the line entities are useful, because from them you get the length wiring.

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)
TIP When you pause the cursor over the data fields, a tooltip advises you on additional operations that are possible in this dialog box:

9. Click **Next** to carry on.

When the Page 3 dialog box appears, it looks scary, as it lists every property of every entity, as well as attributes. The options in this dialog box have the following meaning:

- **Checkbox** — selects an item to be exported to the CSV file
- **Select Attributes** — selects only attributes in the list, and excludes other items
- **Show Checked Only** — shows only the items that are selected, and hides the rest from the list
- **Show All** — shows all items, checked and unchecked
- **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
To turn off (or on) all properties at once, select the first property in the list, hold down the Shift key, select the last property, and then click the checkbox.

To sort the columns alphabetically, click the header of each one. When you click the invisible header of the checkboxes column, it sorts by which boxes are and are not checked.

Your big job now is to turn off all properties, except for the following ones:

- Length
- Product
- StockNo

To do this, follow these steps:

a. Click the Select Attributes button. This highlights the attributes in the list.

b. Click a checkbox in front of an attribute name. This action selects both of them.

c. Click the Property header. This sorts the items alphabetically by property name.

d. Go through the list until you find Length, and then click its checkbox.

e. Click Show Checked Only to ensure that you have the correct ones.

Here is the result of your actions:

10. Click Finish. After a second or two, the results are deposited in the electric.csv file.
To see the result, open the *electric.csv* file with a text editor. The content should look something similar to that illustrated below.

![electric.csv content]

For each block and line in the Electric drawing, BricsCAD lists the following pieces of data:

<table>
<thead>
<tr>
<th>Field</th>
<th>Meaning</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Number of occurrences in the drawing</td>
<td>12 (light bulbs)</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the block or entity</td>
<td>LBULB</td>
</tr>
<tr>
<td>Length</td>
<td>Length of line (does not apply to attributes)</td>
<td>5.1</td>
</tr>
<tr>
<td>Product</td>
<td>Block’s attribute value (does not apply to lines)</td>
<td>Single light bulb</td>
</tr>
<tr>
<td>StockNo</td>
<td>Block’s attribute value (does not apply to lines)</td>
<td>400-0001</td>
</tr>
</tbody>
</table>

You’ve created a rudimentary bill of material, which can be printed out on your printer or imported to a spreadsheet — as described next. Later you learn how to bring the spreadsheet into the drawing as a bill of materials.

**IMPORTING DATA FILES INTO SPREADSHEETS**

To format the raw data and perform calculations, you can import the exported data into a spreadsheet. Shown in this tutorial is LibreOffice Calc, part of a free software package available from [https://www.libreoffice.org/discover/calc/](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:
**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.

![Additional Format Dialog Box](image)

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**.

![First capital Dialog Box](image)

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.
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.

![Text Import dialog box](image)

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

4. 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**.
5. 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).

<table>
<thead>
<tr>
<th>Count</th>
<th>Name</th>
<th>Length</th>
<th>PRODUCT</th>
<th>STOCKNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1Line</td>
<td>3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>1Line</td>
<td>5.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>1Line</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1Line</td>
<td>5.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>2Line</td>
<td>1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>3Line</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>3Line</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>3Line</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>3Line</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>1Line</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>1Line</td>
<td>2.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>2Line</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>1Line</td>
<td>2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>1Line</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Total Length</td>
<td>83.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Remember to save the spreadsheet file.

Placing Data in Drawings as Tables

<table>
<thead>
<tr>
<th>Command</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
</tbody>
</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
- **XLSX** — encapsulated spreadsheet files saved by Excel 2007 and newer
- **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 whatever character is used to separate fields in the file. If you specify the wrong separator character, BricsCAD will later complain “Invalid format.”
5. Click OK. Notice that BricsCAD prompts you to pick an insertion point:
   Specify insertion point or [Style/Width/Height]: (Pick a point)

6. 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

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

**TIP**  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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataLink Name</td>
<td>Electric BOM</td>
<td>This name identifies the data link; it can be any useful name</td>
</tr>
<tr>
<td>Choose an Excel File</td>
<td>electric.xls</td>
<td>Source of data can be a CSV, XLS, or XLSX file</td>
</tr>
<tr>
<td>Allow writing to Excel</td>
<td>Yes</td>
<td>Changes made in the BricsCAD table can be written back to Excel</td>
</tr>
<tr>
<td>Select Excel sheet</td>
<td>ActiveSheet</td>
<td>This is the sheet that is currently displayed by Excel</td>
</tr>
<tr>
<td>Link to entire sheet</td>
<td>Yes</td>
<td>BricsCAD imports only the cell range that is in use</td>
</tr>
<tr>
<td>Use Excel formatting</td>
<td>Yes</td>
<td>BricsCAD uses the Excel formatting, instead of a table style</td>
</tr>
</tbody>
</table>

TIP When you choose a CSV file, many of the Excel-related options become unavailable (grayed out).

5. When you are finished, the dialog box should look like this:

Click OK.
6. 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**.

7. To place the linked BOM, you use the Table command again, with a small twist:
   a. Start the **Table** command:
      : `table`
   b. In the Table Options section, choose **DataLink**:
   c. 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**
8. 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:

![Diagram](image)

**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 Datalinks**

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

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.)
2. Save the spreadsheet, and then exit Excel.
3. In BricsCAD, start the DataLinkUpdate command:
   : datalinkupdate
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 AND ULTIMATE EDITIONS ONLY) BricsCAD Platinum can generate BOMs with a single click, when drawings are made with 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 or Ultimate, and with drawings that have inserts of mechanical components. In the following tutorial, you work through the following stages:

Stage 1 — Explode an assembly into parts
Stage 2 — Generate a BOM automatically
Stage 3 — Label parts with balloons

EXPLODING ASSEMBLIES

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 a as complex as a crankshaft.

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 and Ultimate, assemblies are exploded with the \texttt{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.

\textit{Left: An assembly of parts; right: parts exploded}

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 \textit{piston-pin.dwg}. This drawing contains two components in an assembly.

\begin{itemize}
  \item \texttt{bmexplode}
  \item \textit{Select explosion algorithm [Table by Level/Table by Types/Manual/Settings]} \texttt{<Manual>}
  \item \textit{Select a position of the exploded representation:}
  \begin{itemize}
    \item \textit{Table by Level (TL)} — places components according to their level in the assembly tree in the same row
    \item \textit{Table by Types (TT)} — places components of the same type together in the same row
    \item \textit{Manual} — places components to be moved apart manually
    \item \textit{Settings} — determines how assembly is exploded:
      \begin{itemize}
        \item \textit{Top} — explodes only top level components to suppress minor components
        \item \textit{Bottom} — explodes components
        \item \textit{Name} — assigns a name to the settings so that they can be used for future explosions
  \end{itemize}
\end{itemize}
\end{itemize}
2. Enter the `bmExplode` command.

```
: bmexplode
Select explosion algorithm [Table by Level/Table by Types/Manual/Settings] <Manual>: TL
```

3. 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**

<table>
<thead>
<tr>
<th>Command</th>
<th><code>bmbom</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transparent</strong></td>
<td><code>bmbom</code></td>
</tr>
<tr>
<td><strong>Menu bar</strong></td>
<td>Assembly</td>
</tr>
<tr>
<td><strong>Ribbon</strong></td>
<td>Assembly</td>
</tr>
<tr>
<td><strong>Toolbar</strong></td>
<td>Assembly</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>No.</th>
<th>Component</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>axis_b</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>axis_m</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>axis_s</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>bar_D</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>bar_U</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>base</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>bracket</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>connector</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>platform_U</td>
<td>1</td>
</tr>
</tbody>
</table>

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)
   ```

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.

### QUICK SUMMARY OF bmBOM

: `bmBom`

**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.
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**

<table>
<thead>
<tr>
<th>Command</th>
<th><code>bmballoon</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transparent</td>
<td><code>bmballoon</code></td>
</tr>
</tbody>
</table>

**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.
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.

---

**QUICK SUMMARY OF bmBALLOON**

: `bmballoon`

Select a component insert [select other Table/choose balloon Frame]:

*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
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 with Boolean operations. These two are 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 centers of gravity. BricsCAD does the calculations far quicker than if you were to perform them by hand.

In this chapter you learn to use the Boundary and Region commands, and commands related to boolean operations. (The Region command is available in the Pro, Platinum, and Ultimate 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut</th>
<th>Menu Bar</th>
<th>Ribbon</th>
</tr>
</thead>
<tbody>
<tr>
<td>DelObj</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intersect</td>
<td>in</td>
<td>Modify</td>
<td>Solid Editing</td>
</tr>
<tr>
<td>MassProp</td>
<td></td>
<td>Tools</td>
<td>Inquiry</td>
</tr>
<tr>
<td>Region</td>
<td>reg</td>
<td>Draw</td>
<td>Region</td>
</tr>
<tr>
<td>Subtract</td>
<td>su</td>
<td>Modify</td>
<td>Solid Editing</td>
</tr>
<tr>
<td>Union</td>
<td>uni</td>
<td>Modify</td>
<td>Solid Editing</td>
</tr>
</tbody>
</table>

THIS CHAPTER’S DRAWING

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

RegionTutorial drawing
About Regions

(Pro, Platinum, Ultimate 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.

Regions have their own jargon. The outer boundary of a region goes by the name of “loop.” The holes inside of regions are “islands.” The loops and islands can be of any shape, from a simple triangle to flowing curves.

Technically, regions are ACIS entities, the same technology BricsCAD uses for 3D solid models.

HOW TO CREATE REGIONS

In a curious twist of fate, BricsCAD cannot create regions from scratch. There is no version of the Region command that draws them, as there is 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. BricsCAD finds the boundary, and then makes a region or a polyline along its edges. (This command is, in fact, a subset of the Hatch command, and so you get the island options and gap tolerance as a bonus.)
- **Region** command — operates at the command line and prompts you to select entities that make up a closed area, and then generates a region entity.

Either way, both commands create a region entity from a closed area, which can be 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.
You can end up with one region, no region, or three regions:

Left: Closed non-self-intersecting shape become a single region;

Center: Unclosed shapes cannot become regions;

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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Closed Entity</th>
<th>Open Entity</th>
<th>Self-intersecting Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary</td>
<td>1 region</td>
<td>0 or 1 regions *</td>
<td>1 region per pick</td>
</tr>
<tr>
<td>Region</td>
<td>1 region</td>
<td>0 regions</td>
<td>3 regions with one pick</td>
</tr>
</tbody>
</table>

* 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 Commands

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.

Boundary. First of all, the Boundary command’s dialog box does not let you switch between creating polylines and boundaries. Initially, the dialog box is set up to create polylines; the droplist is grayed out and does not work, for some reason.

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:

```bash
: hpbound
New current value for HPBOUND [1 for ON/0 for OFF] <1 for ON>: off
```

<table>
<thead>
<tr>
<th>HpBound</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Boundary and Hatch commands make boundaries out of regions</td>
</tr>
<tr>
<td>1</td>
<td>Boundaries are made of polylines</td>
</tr>
</tbody>
</table>
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...
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.

**Outer** — the outermost circles form boundaries (shown in red below); the innermost circles are ignored.

**Ignore** — only the outermost circle is used to form the boundary; the others are ignored.
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)
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick a point</td>
<td>Uses boundary detection to find regions around the pick point</td>
</tr>
<tr>
<td>Select entities</td>
<td>Picks individual entities, and then forms regions out of them</td>
</tr>
<tr>
<td>Undo</td>
<td>Unselects the last pick area or entity</td>
</tr>
</tbody>
</table>

I’ve drawn three circles to show you how the resulting regions differ according to how you select them.

**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:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Boundary</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities created</td>
<td>Polylines</td>
<td>Regions only</td>
</tr>
<tr>
<td>(Hpbound)</td>
<td>Regions</td>
<td></td>
</tr>
<tr>
<td>Islands detected</td>
<td>Yes, multi-level</td>
<td>Yes, only one level</td>
</tr>
<tr>
<td>Allow gaps (HpGapTol)</td>
<td>Yes, up to 5000 units</td>
<td>No</td>
</tr>
<tr>
<td>Selection method</td>
<td>Pick enclosed areas</td>
<td>Select entities only</td>
</tr>
<tr>
<td></td>
<td>Select entities</td>
<td></td>
</tr>
<tr>
<td>Retain source entities</td>
<td>Optional</td>
<td>Optional</td>
</tr>
</tbody>
</table>

In summary, the Boundary command is more flexible with more options, but the Region command is simpler with no options.
What To Do With the Leftovers

Both commands leave the source entities in place. To change this behavior, change the DelObj (short for “delete objects”) variable. To have Region erase the source entities automatically, set it to -2 or 2:

<table>
<thead>
<tr>
<th>DelObj</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>Prompts if source entities should be erased with Region</td>
</tr>
<tr>
<td>-1</td>
<td>Prompts if source entities should be erased during commands like Extrude, Revolve, and Sweep</td>
</tr>
<tr>
<td>0</td>
<td>Leaves source entities in place</td>
</tr>
<tr>
<td>1 (default)</td>
<td>Erases source entities only during commands like Extrude, Revolve, and Sweep</td>
</tr>
<tr>
<td>2</td>
<td>Erases source entities with Region</td>
</tr>
</tbody>
</table>

**TIPS**  
Regions are always closed; there are no open regions.

The **Explode** command change region entities back to their original components.

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 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*

<table>
<thead>
<tr>
<th>Command</th>
<th>Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>bo</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Draw</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Draw</td>
</tr>
</tbody>
</table>

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.

![Overlapping circles](image)

3. Turn off the HpBound system variable to ensure the resulting entities will be regions:
   
   
   ![HpBound command](image)

   : 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.

   ![Boundary dialog box](image)

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.

   ![Pick point](image)

7. Press Enter to return to the dialog box:

8. And then click OK to exit the dialog box.
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.
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.

---

**TIP** 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, Platinum, Ultimate versions of BricsCAD

When drawings contain two or more regions or 3D solids, you can perform boolean operations on them. These operations allow you to combine separate regions and 3D solids and make them one — or to separate one into many. “Boolean” is a named after Mr Boole, whose mathematics describe the logical operations that you may have learned in school, such as AND, OR, and NOT.

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.

BricsCAD provides three boolean commands to combine and separate regions and solids. (See the details in the boxed text 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 not available in BricsCAD Classic, unfortunately.

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

<table>
<thead>
<tr>
<th>Commands</th>
<th>ArrayClassic and -Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>ar, -ar</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Modify</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>reg</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Draw</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Surface</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Draw</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Subtract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>su</td>
</tr>
<tr>
<td>Menu Bar</td>
<td>Model</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Solid</td>
</tr>
<tr>
<td>Toolbar</td>
<td>3D Solid Editing</td>
</tr>
</tbody>
</table>

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:
   ```plaintext
   : 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:
   ```plaintext
   : 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:

```plaintext
: 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:

```plaintext
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.

![Image of superfluous rows removed from a rectangular grid]

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)
   ```

   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)
   ```

   ![Image of mirror command used to create a second set of rectangles at 90 degrees]

9. With the vertical rectangle in place, start the **ArrayClassic** command. This is the dialog box version of the command.

   ![Image of ArrayClassic command dialog box]
a. For Array Type, ensure **Rectangular** is chosen:

![Array Type](image)

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:

![Settings](image)

- 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

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](image)

   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 XY 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.
Here you learn how to create 3D models using traditional and modern editing techniques. Traditional techniques use commands to create and manipulate 3D models; modern techniques 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, Platinum, and Ultimate editions of BricsCAD.)
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut</th>
<th>Menu Bar</th>
<th>Ribbon Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>DelObj</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extrude</td>
<td>Ext</td>
<td>Model</td>
<td>3D Solids</td>
</tr>
<tr>
<td>QuadDisplay</td>
<td>F12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SolidEdit</td>
<td></td>
<td>Model</td>
<td>3D Solid Editing</td>
</tr>
<tr>
<td>UCS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THIS CHAPTER’S MODEL

By the end of this lesson, your 3D model will look like this:

![Focus-bracket drawing 3d-part.dwg](image)
About 3D Solid Models

(pro, platinum, ultimate editions only) BricsCAD Pro, Platinum, and Ultimate 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), designs can be tested on the computer to ensure they work properly before they are manufactured.

BricsCAD offers several methods for creating 3D models from solids, and here you learn about two of them, traditional commands and direct modeling.

In this chapter’s 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 this cloud folder:
https://my.pcloud.com/publink/show?code=XZnq8mkZ7T7f1ynxJYb7vDyVl6DwyqK1Ysk

PLANNING AHEAD

There are specific 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 ways 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 made with Box, Sphere, Cylinder, Pyramid, Wedge, and Torus commands.

CREATE 3D MODELS FROM 2D ENTITIES

Extrude and Revolve commands 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 need to plan ahead to construct 3D models; this differs from 2D, where you can simply start drawing. There is, however, a Catch-22 here: you become effective in planning ahead only after you gain experience with 3D modeling, and so this chapter shows you some of the tricks in putting together 3D models.

Planning Ahead by Deconstructing 3D
One 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 are 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 an extrusion, 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 a few commands to model the 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 Launcher dialog box. Click Modeling.

![Launcher dialog box](image)

3. In the Getting Started tab, change the template file to “Default-mm.” This ensures the drawing will be made with metric units.

![Getting Started tab](image)

4. Click New Drawing.

![New Drawing dialog box](image)

Notice that BricsCAD opens in the Modeling workspace.

5. Prepare this drawing by changing the following settings in the status bar:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Toggle</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNAP</td>
<td>Off</td>
<td>You use entity snaps and dynamic dimensions, in place of snap spacing</td>
</tr>
<tr>
<td>GRID</td>
<td>Off</td>
<td>The grid is not useful for this project</td>
</tr>
<tr>
<td>ESNAP</td>
<td>On</td>
<td>Turn on INTERsection snap; turn off all others</td>
</tr>
</tbody>
</table>
To turn on INTersection entity snap, right-click **ESNAP** on the status bar, as shown below. Ensure all other esnaps are turned off.

6. Use the **Units** command to change the display precision of decimal places (linear units) to 0.0mm. You will be working with dimensions to the nearest 0.5mm.
7. Close the Settings dialog box by clicking the X.

8. 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.)

9. Change the value of variable DelObj to 0. This ensures 2D entities are retained after 3D objects are made from them.

   Recall from a previous lesson that this 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.)

10. Finally, save the drawing by any name that you like, such as “3d-part.dwg”.

With the drawing prepped, you are ready to start modeling!
### Modeling a Box

<table>
<thead>
<tr>
<th>Command</th>
<th>Extrude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>ext</td>
</tr>
<tr>
<td>Menu</td>
<td>Solids</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Solid</td>
</tr>
<tr>
<td>Toolbar</td>
<td>3D Solids</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>SolidEdit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>Solids</td>
</tr>
<tr>
<td>Toolbar</td>
<td>3D Solid Editing</td>
</tr>
<tr>
<td>Ribbon</td>
<td>Solid</td>
</tr>
</tbody>
</table>

The top of the part is a square open box. It is shown in gray by the figure below.

![Image of a square open box](image.png)

To make it, you’ll work through three commands, in this order:

1. **Rectangle** command — defines the overall size and the base of the square
2. **Extrude** command — changes the 2D square into a solid 3D box
3. **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
```

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 [M0de]: (Pick the square)
Select entities/subentities to extrude or set [M0de]: (Press ENTER to end entity 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 ones with “walls” — it uniformly removes the insides. Follow these steps to make a hollow box:
   a. Because the SolidEdit command has many options, it is faster to get to the Shell option via the ribbon. (Don’t enter the “shell” command, because it doesn’t do what you need!)

   From the ribbon’s **Solids** tab, look in the **Solid Editing** panel, and then click **Shell**. (It’s a bit hard to find it, but Shell is among the ones that drop down from the last icon, Separate.)

---

**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
b. As prompted, select the extruded box:

```
Select 3d solid: (Pick anywhere on the box)
```

Notice that the box turns blue, as BricsCAD confirms your selection.

c. Next, pick the top face, which is shown in blue in the figure below.

```
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; it no longer has a 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. Hold down the Ctrl key.
2. On the Lookfrom widget, click the triangle at the 7pm position.

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
Notice that BricsCAD swirls the box around so that we now see the bottom of it.

3. If necessary, use the **Zoom Extents** command to see the entire model.

**EXTRUSION BY DIRECT MODELING**

Now that you have a taste of modeling using commands, we'll switch to direct modeling — modeling without commands.

The bottom of the part has a hollow cylinder. In this part of the tutorial, you draw two circles and then extrude them without using commands to do the work. But first, this word of explanation of how to draw in 3D space...

**Applying Dynamic UCS**

A box has six sides in 3D. CAD, however, can only draw on one side at a time: the side that is in the current x,y-plane. No matter which 3D CAD package you use, they all fall back to drawing on the x,y-plane. And so a crucial aspect to direct modeling is working on a plane that is correctly-positioned.

Fortunately, CAD systems make it easy today to place the x,y-plane so that it matches the area on which you want to work, pretty much automatically. In BricsCAD, this feature is called *dynamic UCS*. Here is how it works for drawing a 2D circle on a 3D box:

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. Use 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 happen: the face turns blue, and the UCS icon jumps to a corner of the face. Move the cursor and another face turns blue.
This is dynamic UCS at work: BricsCAD automatically relocates the x,y drawing plane to the face that you pick.

Left to right: As cursor passes over a face, it turns blue and the UCS icon (tri-color icon) is relocated

**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 works during drawing and editing commands, because most of them operate only on an x,y plane.

---

**TIP** 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 from face to face.

---

4. To locate the circle, pick two points on the edges of the box, right on the middle of two facing edges; see the following figures.
   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) near where the word "Midpoint" appears, 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 a temporary MIDpoint entity snap mode:
      
      ![Midpoint](image)

      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)

QUICK SUMMARY OF DUCS ENTITIES

Dynamic UCS works with many kinds of entities during drawing, editing, and dimensioning:

- All 3D solid faces
- Most 2D entities

DUCS does not work with curved 3D entities, such as spheres, as they have no flat faces, and so the UCS cannot be determined.

(NEW IN V20) DUCS works with the following 2D entities: points, lines, 2D and 3D polylines, rays and xlines, arcs, circles, ellipses, splines, text and mtext, 2D solids, 3D faces, traces, blocks, viewports (in paper space), mlines, leaders and mleaders, hatches, helices, cameras and lights, sections, PDF and image underlays, and shapes.

BricsCAD lets you determine whether DUCS supports 3D, 2D, or both through the UcsDetect variable:

<table>
<thead>
<tr>
<th>UcsDetect</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Detect 2D entities and 3D faces</td>
</tr>
<tr>
<td>2</td>
<td>Detect only 2D entities</td>
</tr>
<tr>
<td>1</td>
<td>Detect only 3D faces</td>
</tr>
<tr>
<td>0</td>
<td>DUCS is off</td>
</tr>
</tbody>
</table>

Alternatively, right-click DUCS on the status bar, and then choose the entity types DUCS should work with:
5. A second circle is needed to define the thickness of the 0.5mm wall. But instead of drawing it, you make the copy using the **Offset** command.

![Offset circle](image)

: **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)*

Notice that the second circle appears.

### Extruding Directly with the Quad Cursor

With the pair of 2D circles in place, you extrude them to create a 5mm-tall hollow cylinder. One way to make cylinders is to extrude circles.

This time, you use the Quad cursor to start the extrusion. The Quad is unique to BricsCAD, giving you fast access to commands right at the cursor, as described earlier in this book. Making a hollow cylinder takes these steps:

i. Extrude the inside circle into a cylinder that is more than 5mm in height.

ii. Extrude the outside circle to exactly 5mm.

iii. To make the hole, subtract the taller cylinder from the shorter one.

You extrude the inside circle by a distance longer than 5mm, because its height does not matter; making it taller makes it easier to erase it to create a hole.

Follow these steps:

1. On the status bar, ensure that **QUAD** is turned on.

![Quad status bar](image)
2. Move the cursor over the smaller circle. Notice that it turns blue, and that a second cursor appears. This is the Quad cursor, sporting a single icon (for now).

3. 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, a tooltip appears, explaining the name and purpose of the button.

4. 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. It consists of 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.

**TIP** If you have difficulty selecting the circle, hover the cursor over a part of the circle, and press Tab until the Quad reports “Circle.”

*Left:* Extruding the outer circle into a 5mm cylinder; *right:* Two cylinders in place

**TIP** There is also a dynamic angle, shown as 0 in the figures above. Its purpose is to extrude entities at an angle; press Tab to edit the angle value.

**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.

1. Select the outer cylinder first...

2. Select the inner cylinder last...
TIP  The **Subtract** command is sensitive to the order in which entities are selected. Make sure you pick the correct ones in the correct order: first the one that stays, and then the one that will be removed (subtracted).

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 by hollowing out the cylinder.

---

**PushPull Modeling**

A hole needs to be punched through the thin wall that’s 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 then use the PushPull function 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. Click in the center of the widget. Notice that the viewpoint changes.

   ![Look From Widget](image)

   b. You may need to do a **Zoom Extents** to see the entire model.

   TIP  If the Look From widget is turned off, you can turn it on with the **LookFrom** command:

   ```
   : lookfrom
   LookFrom [ON/OFF/Settings] <ON>: on
   ```
2. To draw the circle, use the same technique as before: employ the **Circle** command with the **2P** option and **MIDpoint** snaps:

```
: 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)
```

3. 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.

   a. Pass the cursor over the newly added circle. Notice that it turns blue.
b. Move the crosshair cursor into the Quad cursor. Notice that it expands to show more commands. Choose the **PushPull** button.

![PushPull button](image)


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.

![Cursor dragging](image)

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 into the Quad. The crosshair turns into an arrow cursor.
3. Move the arrow cursor into the blue Direct Modeling title bar. Notice that the Quad cursor expands to show more icons.
4. Choose the **Align UCS** button, and then press **Enter**. The UCS plane is now fixed on the face.

**TIP**  If BricsCAD does not highlight the feature in which you are interested, then press the **Tab** key. Each time you press Tab, a different feature laying under the cursor is highlighted. Press Tab enough times, and the selection cycles around to the first one highlighted. See figure below.

---

**Pulling, Instead of Pushing**

**Command**  dmPushPull  
**Menu**  Model | Direct Modeling | Push/Pull  
**Ribbon**  Model | Edit | Push/Pull  
**Toolbar**  Direct Modeling | Push/Pull  

Here you will draw a rectangle on a face, then pull it out with the direct PushPull operation. Later, when it comes to the screw holes I want you to make a mistake so that I can show you that BricsCAD also performs direct **editing** — in this case, changing the diameter of a 3D hole without using a command.

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
```
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"); start the `dmExtrude` command and specify extrusion distance of 4.5mm; done!

3. The screw hole hangs on the wing. It is made of a pair of cylinders, which (again) are made from a pair of circles. First, though, align the UCS to the bottom face of the wing, as illustrated below:

Remember to press Tab, if you find you have difficulty selecting the bottom face.

4. Drawing 2D entities in 3D can be tricky when the viewpoint is the plan view. By default, Bricsys will snap to the nearest geometric feature; in 3D, this can be the one you don't want too easily. 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.

<table>
<thead>
<tr>
<th>OsnapZ</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Snaps to the nearest z coordinate</td>
</tr>
<tr>
<td>On</td>
<td>Snaps to the elevation setting (usually 0; z = elevation)</td>
</tr>
</tbody>
</table>

5. Using the LookFrom widget, change the viewpoint to the top (or plan) view.

6. 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, so that you can correct it later using 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 Extrude again to turn the inner one to any height taller than 3.5mm; notice that it creates the hole automatically — no need to use the Subtract command!
9. The diameter of the hole is too large. It should be 1mm, but is currently 0.5mm. BricsCAD can edit solid models using direct editing. Here is how to correct the diameter of the hole:

   a. Move the cursor over the edge of the hole. If the hole does not highlight in blue, then press the Tab key until it does.

   b. In the Quad cursor, choose the \texttt{dmExtrude} button.

   c. The hole must be changed to a diameter of 1.0mm. Drag the hole dynamically, or enter 1.0 in the dynamic dimension.
The hole is the correct size. Press Ctrl+S to save your work.

TIP To change the UCS back to “normal,” enter the UCS command, and then choose the World option:

:ucs Specify origin of UCS or [Face/Named/Entity/Previous/View/X/Y/Z/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)*
TIP You can continue to directly edit the model, even after all its parts are joined by the Union command.

2. Now that the model is one unified part, you can apply commands like MassProp and Properties to find out the properties of this focusing bracket.

: massprop
  Select entities: all
  Select entities: (Press Enter to continue)
  Write analysis to a file? Yes/No: y

The report you see in the text window can be saved to a .mpr (mass properties report) file.

3. Save the drawing.
Generating 2D Drawings

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 manufacturers. BricsCAD generates these drawings semi-automatically: all you need to do is pick the spots where you want the views placed. BricsCAD figures out the orientations of the views, and then generates the drawings.

Let’s see how it works.

1. The Drawings Views panel needed for this tutorial is not found in the Modeling workspace, so you need to switch to the Mechanical workspace:
   a. On the status bar, right-click **Modeling**
   b. From the shortcut menu, choose **Mechanical**
   c. Wait a moment as BricsCAD changes the user interface. Then, on the ribbon, click the **Annotate** tab.

Focus on the **Drawing Views** panel, as it contains the commands you need for this tutorial.

2. 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, etc — in a new layout.

   : _viewbase
   Preset: “None", View scale: “Adapt to paper size”

3. Press **Enter** to make drawings from the entire model:

   Select objects or [Entire model/preseTs] <Entire model>: (Press **Enter**)
4. At the next prompt, press **Enter** to use Layout1 for the drawings:

   Enter new or existing layout name to make current `<Layout1>`: (Press **Enter**)

5. 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 tutorial, position the cursor in the upper left quadrant, and then click. See figure above.

```
Select position for base view [Scale/Hidden lines/Tangent edges/Orientation/Projection type/Isometric style/sElect/Cancel] <Cancel>: (Move the cursor to upper left, and then click)
```

The quadrant in which you click determines the viewpoint generated by BricsCAD automatically:

<table>
<thead>
<tr>
<th>Quadrant</th>
<th>2D View Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper left</td>
<td>Top view</td>
</tr>
<tr>
<td>Upper right</td>
<td>Side view</td>
</tr>
<tr>
<td>Lower left</td>
<td>Front view</td>
</tr>
<tr>
<td>Lower right</td>
<td>Isometric view</td>
</tr>
</tbody>
</table>

5. 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)
```
QUICK SUMMARY OF VIEWBASE OPTIONS

: viewbase
Select objects or [Entire model/preseTs/Special view] <Entire model>: 
Enter new or existing layout name to make current <LayoutName>: 
Select position for base view [Scale/Hidden lines/Tangent edges/Orientation/Projection type/view Quality/Isometric style/select]<Cancel>: 
Select position for projected view [Done] <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.

SPECIAL VIEW OPTION
Offers to make the view exploded or not exploded (default).

SCALE OPTION
Sets the scale of all views to fit the following arrangements:

- fit 4 views — adjusts scale to fit four orthographic views: Front, Top, Left, and Right. The Front view (base view) is defined by the Orientation option.
- 9 views — adjusts scale to fit five orthographic views and four isometric views.
- 5 views — adjusts scale to fit five orthographic views: Front (*), Top, Left, Right, and Back.
- 10 views — adjusts scale to fit six orthographic views and four isometric views.
- Standard scales — displays the scales lorded over by the ScaleListEdit command.
- Custom — prompts for any scale factor.

HIDDEN LINES OPTION
Toggles the representation of hidden lines.

TANGENT EDGES OPTION
Determines whether tangent edges between tangent faces are displayed or not; tangent edges are always displayed in isometric views.

Continued...
HIDDEN LINES OPTION
Determines whether hidden lines are shown.

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).

VIEW QUALITY
Toggles the quality of the projected view between low (faster to display) and high (default)

ISOMETRIC STYLE OPTION
View style for isometric views: rendered using Conceptual visual style, wireframe, or any other visual style.

SELECT OPTION
Adds and removes model space elements to and from the projected view.
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/Rotate view] <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 2D Wireframe 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 another 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.

ROTATE VIEW OPTION
  Rotates section viewports to be horizontal, vertical, or at a custom angle.
6. 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**)

**ADDING SECTION VIEWS AND DETAILED VIEWS**

<table>
<thead>
<tr>
<th>Command</th>
<th>ViewSection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>Model</td>
</tr>
<tr>
<td>Ribbon</td>
<td>(Mechanical workspace)</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Drawing Views</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>ViewDetail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>Model</td>
</tr>
<tr>
<td>Ribbon</td>
<td>(Mechanical workspace)</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Drawing Views</td>
</tr>
</tbody>
</table>

With the 2D drawings in place, it is common to add section views that show the insides of models, which use hatch patterns to indicate the materials with which they should be manufactured. When necessary, greatly enlarged detail views are also added to show complex areas.

**Resizing Views**

Before you can add a cross-section view, you have to adjust the existing views, because the page is full — no room for another view! One solution is to make the existing views smaller. Here is how to do this:

1. Select a viewport border.
2. In the Properties panel, click **Custom Scale**.
3. Enter a new value, such as `.1`.
4. Notice that the viewport is smaller. Move the viewport by dragging it (with its center grip) or through the **Move** command.

Another option is to erase unwanted views. To erase a view, select the viewport border, and then press the **Del** key on the keyboard.
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 inherent 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.
Making Section Views

Okay, with space freed up, you can go ahead to make the section view:

1. From the ribbon’s Annotate > Drawing Views panel, click the Section View button (or else enter the ViewSection command).

2. BricsCAD prompts you to choose a drawing view. The section view will be generated from the view you pick, so choose carefully!
   a. For this tutorial, move the cursor to the side view, which is in the upper left quadrant (see figure below).
   b. Click inside the viewport.

   TIP  When selecting the view, don’t select the viewport’s rectangle, as BricsCAD will just repeat the prompt. Click inside the viewport, as shown by the crosshair cursor below.

   Notice that BricsCAD highlights the selected viewport with dashed lines.

3. With the drawing view selected, it’s time to show BricsCAD the drawing that you want to section. At the next prompt, pick two points outside the viewport, shown by the arrowheads in the figure below:

   TIP  I recommend holding down the Shift key to enforce ortho mode. The result is a section line that’s precisely horizontal or vertical.
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 notation of “A-A.” Engineers use the A-A to see what is referenced elsewhere in the drawing. The label also indicates the scale factor of the section, 1:5 — also automatically determined by BricsCAD.

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.

Here is the meaning of the ANSI hatch patterns included with BricsCAD:

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Example</th>
<th>Pattern Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI31</td>
<td><img src="image.png" alt="Image" /></td>
<td>Iron, general purpose hatch</td>
</tr>
<tr>
<td>ANSI32</td>
<td><img src="image.png" alt="Image" /></td>
<td>Steel</td>
</tr>
<tr>
<td>ANSI33</td>
<td><img src="image.png" alt="Image" /></td>
<td>Bronze, brass, copper, composites</td>
</tr>
<tr>
<td>ANSI34</td>
<td><img src="image.png" alt="Image" /></td>
<td>Rubber, plastic, electrical insulation</td>
</tr>
<tr>
<td>ANSI35</td>
<td><img src="image.png" alt="Image" /></td>
<td>Defined by the legend</td>
</tr>
<tr>
<td>ANSI36</td>
<td><img src="image.png" alt="Image" /></td>
<td>Defined by the legend</td>
</tr>
<tr>
<td>ANSI37</td>
<td><img src="image.png" alt="Image" /></td>
<td>White metal, zinc, lead, babbit, and alloys of them</td>
</tr>
<tr>
<td>ANSI38</td>
<td><img src="image.png" alt="Image" /></td>
<td>Magnesium, aluminum, and alloys of them</td>
</tr>
</tbody>
</table>
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 at a larger scale factor by the ViewDetail command.

Here you make a detail of one of the small round parts:

1. Enter the ViewDetail command:
   
   : viewdetail

2. BricsCAD prompts you to choose a drawing view:
   
   Select drawing view: (Pick inside a viewport)
   
   This is the drawing view from which the detail will be taken. For this tutorial, click inside the lower left quadrant’s viewport, as shown below.

3. Pick the center of the detail, as BricsCAD by default uses 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)

4. Now size the circle to indicate the extent of the detail. You can see the circle in the figure below, it already has the “B” reference attached.

   Select radius of detail view: (Drag the circle to size the detail area)

   **TIPS** You can make sections of details, and details of sections, and details of details.
   
   To hide the rectangles that make up the viewports, freeze the “Drafting Viewports” layer. You can use the Layer command (shown below) or the Layers droplist in the Properties panel.
5. 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 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 to create drawings that are *constrained*. In these drawings, dimensions determine the sizes of entities — not the other way around, as you learned in an earlier chapter. In addition, constraints can be used to lock geometry 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Menu Bar</th>
<th>Ribbon</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaterialBrowserOpen</td>
<td>Parametric</td>
<td>Mechanical Browser</td>
</tr>
<tr>
<td>ConstraintBar</td>
<td>Parametric</td>
<td>2D Constraints Bar</td>
</tr>
<tr>
<td>DimConstraint</td>
<td>Parametric</td>
<td>2D Constraints</td>
</tr>
<tr>
<td>GeomConstraint</td>
<td>Parametric</td>
<td>2D Constraints</td>
</tr>
</tbody>
</table>

THIS CHAPTER’S MODEL

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

![Diagram](image)

*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 and Ultimate add 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. To see this toolbar, right-click any toolbar or the ribbon, and then choose **BricsCAD | Toolbars | 2D Constraints**:

![Buttons geometric constraints (left half) and for dimensional constraints (right half)](image)

**TIP** If you find the buttons on the toolbar too small, then right-click the toolbar, and then choose **Toolbar Size | Large Icons**. This trick does not work for the ribbon.

If you prefer the ribbon, then switch to the “Modeling” workspace, and then choose the ribbon’s **Parametric** tab:

![Parametric tab](image)

**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 dimensional constraints on right half of the toolbar; below: 2D dimensional 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.

<table>
<thead>
<tr>
<th>Type of Constraint</th>
<th>Command</th>
<th>Constraining Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>dcLinear</td>
<td>Horizontal or vertical distance</td>
</tr>
<tr>
<td>Horizontal</td>
<td>dcHorizontal</td>
<td>Horizontal (X) distance between two points</td>
</tr>
<tr>
<td>Vertical</td>
<td>dcVertical</td>
<td>Vertical (Y) distance between two points</td>
</tr>
<tr>
<td>Aligned</td>
<td>dcAligned</td>
<td>Distance between two points at any angle</td>
</tr>
<tr>
<td>Angular</td>
<td>dcAngular</td>
<td>Angle between two lines or linear polyline segments; or angle of an arc or polyline arc; or angle between three points on an entity</td>
</tr>
<tr>
<td>Radial</td>
<td>dcRadius</td>
<td>Radius of a circle, arc, or polyline arc</td>
</tr>
<tr>
<td>Diameter</td>
<td>dcDiameter</td>
<td>Diameter of a circle, arc, or polyline arc</td>
</tr>
<tr>
<td>DimConvert</td>
<td>DimConvert</td>
<td>Converts associative dimensions to dim constraints</td>
</tr>
</tbody>
</table>

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 $\text{dia}_1$ (diameter of one circle) equal to $d_1$ (distance between circles) using a simple formula, $\text{dia}_1 = d_1$.

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 $d_1$ (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.

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
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.

   ![Edit Text Dialog Box](image)

   b. Enter a new value, such as **2500**, and then twice press **Enter** to exit the dialog box.
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/COLinear/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'.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Type of Constraint</th>
<th>Command</th>
<th>Constraining Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>Horizontal</td>
<td>gcHorizontal</td>
<td>Keeps entities horizontal (parallel to the x-axis)</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Vertical</td>
<td>gcVertical</td>
<td>Keeps entities vertical (parallel to the y-axis)</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Perpendicular</td>
<td>gcPerpendicular</td>
<td>Keeps entities perpendicular to one other</td>
</tr>
<tr>
<td>![Icon]</td>
<td>PArallel</td>
<td>gcParallel</td>
<td>Keeps entities parallel to one another</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Tangent</td>
<td>gcTangent</td>
<td>Keeps circular and straight entities tangent</td>
</tr>
<tr>
<td>![Icon]</td>
<td>SMooth</td>
<td>gcSmooth</td>
<td>Keep splines smooth with splines, lines, arcs, polylines</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Coincident</td>
<td>gcCoincident</td>
<td>Keeps end points attached, such as of two lines</td>
</tr>
<tr>
<td>![Icon]</td>
<td>CONcentric</td>
<td>gcConcentric</td>
<td>Keeps circles, arcs, ellipses, and elliptical arcs centered</td>
</tr>
<tr>
<td>![Icon]</td>
<td>COLinear</td>
<td>gcCollinear</td>
<td>Makes linear entities to lie in the same line</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Symmetric</td>
<td>gcSymmetric</td>
<td>Keeps entities or points symmetric about mirror lines</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Equal</td>
<td>gcEqual</td>
<td>Gives curved entities the same radius; open entities the same length</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Fix</td>
<td>gcFix</td>
<td>Keeps entities fixed in-place in the drawing</td>
</tr>
</tbody>
</table>

ACCEPTABLE GEOMETRY

Constraints apply to lines, segments of polylines, circles, arcs, ellipses, elliptical arcs, and splines.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Constraint Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arcs</td>
<td>Endpoints, center points, and midpoints</td>
</tr>
<tr>
<td>Arrays</td>
<td>Parameters</td>
</tr>
<tr>
<td>Block insertions</td>
<td>Insertion point</td>
</tr>
<tr>
<td>Circles</td>
<td>Center points</td>
</tr>
<tr>
<td>Ellipses</td>
<td>Center points</td>
</tr>
<tr>
<td>Elliptical arcs</td>
<td>Endpoints, center points, and midpoints</td>
</tr>
<tr>
<td>Lines</td>
<td>Endpoints and midpoints</td>
</tr>
<tr>
<td>Polyline line and arc segments</td>
<td>Endpoints, midpoints, center points of arc segments</td>
</tr>
<tr>
<td>Splines</td>
<td>Endpoints</td>
</tr>
<tr>
<td>Tables</td>
<td>Insertion point</td>
</tr>
<tr>
<td>Text, mtext, and attributes</td>
<td>Insertion point</td>
</tr>
<tr>
<td>Xref attachments</td>
<td>Insertion point</td>
</tr>
</tbody>
</table>
Notice that the line changes its length to match the new value of the constraint, as does the distance measured by the associative dimension.

![Dimensional Constraint Example](image)

**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 constraints 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.

![Horizontal Constraint Example](image)

*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 cloud folder: [https://my.pcloud.com/publink/show?code=XZnq8mkZ7f1pynxYb7vDyVL6DwyqK1Ysk](https://my.pcloud.com/publink/show?code=XZnq8mkZ7f1pynxYb7vDyVL6DwyqK1Ysk)

1. Start BricsCAD with a new drawing.
2. Draw a shape similar to the one illustrated above with the **Line**, **Arc**, and **Circle** commands. The exact size and exact placement of entities is not critical.
3. The 2D Constraints toolbar assists you in assigning geometric constraints to entities. Open it, as follows:
   a. Right-click any toolbar.
   b. From the shortcut menu, choose BRICSCAD, choose Toolbars, and then 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 x 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.
      :
   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 \texttt{gcConcentric} to match the center point of the circle to that of the arc, as follows:

\begin{verbatim}
: gcConcentric
Select first entity: (Pick the circle)
Select second entity: (Pick the arc)
\end{verbatim}

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 \texttt{gcTangent} constraint, as follows:

\begin{verbatim}
: gcTangent
Select first entity: (Pick the arc)
Select second entity: (Pick a line)
\end{verbatim}

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:

\begin{verbatim}
: constraintbar
Select option to [Show/Hide/Reset] constraints: S
Select entities or ENTER to select all: (Press Enter)
\end{verbatim}
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]: (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.
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
- Civil commands *(new to V20)*
- Cloud- commands (ex-Chapoo commands)
- 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

Commands with a hyphen prefix, such as -Color, are ones that run at the command prompt, and have a complimentary command, such as Color, that displays a dialog box.

Command names new in V20 are shown in **blue**.
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.

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.

*AiMleaderEditAdd* adds leader lines to multi-leaders.

*AiMleaderEditRemove* removes leader lines from multi-leaders.

*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_TileMode1* 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.

**B Commands**

*Background* displays the Background dialog box for creating background colors and images in shaded modes and renderings.

*Base* changes the drawing's insertion point when 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.

*B Hatch* 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.
- **bimClassify** classifies an entity as a building element with a name and an internal 'guid' (globally unique identifier).
- **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.
- **bimFlip** flips the starting face from which the layers of a composition are set out.
- **bimFlowConnect** connects linear solids.
- **bimGrid** creates rectangular and radial grids with automatically-applied labels.
- **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.
- **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.

bimStair creates stairs between two floors (as a rectangular parametric array).

bimStretch stretches BIM entities.

bimStructuralConnect connects linear solids.

bimTag tags BIM sections.

bimUpdateRoom updates data about the selected room.

bimUpdateThickness re-applies the overall thickness of a composition to the solid.

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.

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.

-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.

**CenterDisassociate** disassociates center lines and marks from circles and arcs.

**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.

**CenterReset** resets 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.
CopyGuided copies entities along guidelines.
CopyHist copies the command history to the Clipboard.
CopyToLayer copies selected entities to another layer.
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.

Civil Commands

(Civil engineering commands are part of Platinum)
Alignment creates horizontal and vertical alignments typically used to design roads.
AlignmentEdit edits horizontal and vertical alignments.
AlignmentView views alignment along TIN surfaces.
AlignmentVInitial creates vertical alignments.
Grading interactively modifies TIN surfaces to create grading effects, such as for roads and foundations.
LandXmlEexport exports the drawing in LandXML format
LandXmlImport imports LandXML files into the current drawing
Tin (triangulated irregular network) imports data from TIN files to create land surfaces, and converts Civil 3D surfaces to the BricsCAD format
TinEdit adds and removes points, break lines, and boundaries in TIN surfaces.
TinExtract creates a mesh or 3D solid between TIN surfaces or between a TIN surface and elevation.
**Cloud 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.

**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.
- **DdPTyoe** 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.
- **DelEData** deletes extended entity data from the selected entity (short for “delete entity data”).
- **DgnImport** imports Microstation design files and converts them to entities.
- **DgnImportOptions** opens the Settings dialog box at the DgnImport section.
Dish draws dishes (bottom half-spheres) 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.
Drag moves faces.
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”).
DstConvert converts sheetset DST files to XML format.
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.

**Dimensioning 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.
AiDimStyle creates dimension styles from a selected dimension.
Dim places and edits dimensions at the ‘Dimensioning command:’ prompt.
DimI executes a single dimension command at the ‘Dimensioning command:’ prompt.
DimAligned draws dimensions parallel to (aligned with) selected entities; works with lines, polylines, arcs, and circles.
DimAngular dimensions angles.
DimArc places arc length dimensions.
DimBaseline places multiple linear or angular dimensions starting at the same base point; command can only be used when at least one other dimension is already in the drawing.
DimCenter places center marks at the center points of circles and arcs.
DimContinue continues linear and angular dimensions from the endpoint of the previous dimension.
DimDiameter dimensions the diameter of circles and arcs, and places a center mark.
DimDisassociate removes associativity from selected dimension entities.
**DimEdit** changes wording and angle of dimension text; changes the angle of extension lines.

**DimLeader** draws leaders.

**DimLinear** places linear dimensions horizontally, vertically, or rotated.

**DimOrdinate** measures x and y ordinate distances from a common origin, specified by the current UCS origin.

**DimOverride** overrides the values of the current dimension style.

**DimRadius** dimensions the radii of arcs and circles.

**DimReassociate** reassociates or associates dimensions to entities or points on entities.

**DimRegen** updates associative dimensions (short for “dimension regeneration”).

**DimStyle** and **-DimStyle** creates and modifies dimension styles through the Drawing Explorer.

**DimStyleSet** reports the current dimension style in the command bar.

**DimTEdit** changes the position of dimension text.

**Dimensional Constraint Commands**

*(dc = dimensional constraint)*

**CleanUnusedVariables** purges variables not used by constraint expressions and not linked to dimensions.

**dcAligned** constrains the distance between two defining points on entities.

**dcAngular** constrains the angle between three constraint points on entities; or between two lines; or between two polyline segments; or constrains the angles of arcs or polyline arcs.

**dcConvert** converts an associative dimension to a dimensional constraint.

**dcDiameter** constrains the diameters of circles, arc, or polyline arcs.

**dcDisplay** shows and hides dimensional constraints.

**dcHorizontal** constrains the horizontal distance between two defining points on entities.

**dcLinear** constrains horizontal or vertical distance between two defining points on entities.

**dcRadial** constrains the radius of circles, arcs, or polyline arcs.

**dcVertical** constrains the vertical distance between two defining points on entities.

**DelConstraint** removes all dimensional (and geometrical) constraints from an entity.

**DimConstraint** applies a dimensional constraint to an entity or between constraint points on entities; converts associative dimensions to dynamic dimensions.

**Direct Modeling Commands**

*(Available for Pro or Platinum editions only; dm = direct modeling)*

**dmAngle3D** applies angle constraints between the faces of a solid or of different solids.

**dmAudit** checks and fixes 3D models.

**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.

**ExpImages** opens the Drawing Explorer at the Images section.

**ExpLayers** opens the Drawing Explorer at the Layers section.

**Explode** breaks complex objects into their component entities.


**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.

**ExpPdfs** opens the Drawing Explorer at the PDF section.

**ExpUcs** creates, modifies, deletes named UCSes through the Drawing Explorer (short for “explore user-defined coordinate systems”).

**ExpXrefs** opens the Drawing Explorer at the XRefs section.

**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
Concise Summary of Command Names

Inside BricsCAD V20

Boundary Detection of SelectionModes is activated.

**GeographicLocation** sets the geographic location of the drawing.

**GoToStart** displays the Start tab.

**Gradient** and **-Gradient** fill closed areas with gradient fills of one or two colors.

**GradientBkgOff** and **GradientBkgOn** turn off and on the gradient displayed in the working area.

**GraphScr** switches from the text windows to the graphics windows (short for “graphics screen”).

**Grid** turns the grid display on or off and sets other grid options.

**Group** and **-Group** creates and modifies named groups of entities.

---

**Geometric Constraint Commands**

*For 3D constraints, see Direct Modeling Commands section; gc = geometric constraints*

**ConstraintBar** shows, hides, and resets the display of geometric constraint icons.

**DelConstraint** removes all geometrical (and dimensional) constraints from an entity.

**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.

**HatchGripEdit** adds and removes grips from hatches and gradients.

**HatchToBack** sets the draw order of all hatch entities in the drawing to display behind all other entities.

**Helix** draws 2D spirals or 3D helixes.
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 attach 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 are imported with the optional Communicator module.

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.

InsertGuided and -InsertGuided inserts blocks along guide curves.

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.

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").

**OleOpen** opens OLE objects for modification.

**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.

**ParametricBlock** creates a parametric block from entities in the drawing; useful for BEdit.

**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.

**PdfImport** and **-PdfImport** imports PDF files and converts them to drawing entities.

**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** and **-PointCloudAttach** attach Bricsys-format point cloud files to the current drawing

**PointCloudColorMap** changes the colors of point based on their elevation.

**PointCloudCrop** and **PointCloudUncrop** crop the extents of the current point cloud, and undo the cropping.

**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** and **-PointCloudPreprocess** convert ASCII PTS, PTX, LAS, and other cloud files into the compressed binary Bricsys file format.

**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.
PSsetupIn and -PSsetupIn 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.

Rlinit 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.


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 an 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.

RScrip 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 walks left, right, forward or backward through a 3D scene 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 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.

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.

Screenshot takes a screen grab of the current space, excluding all UI elements.

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\V20x64\en_US\Support` folder.

**SvgOptions** controls the output as SVG files.

**Sweep** creates solid primitives or surfaces by sweeping two dimensional entities along a path.

**SysWindows** arranges windows.

---

**Sheet Metal Commands**

*(Available for 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.

**smFlangeContour** creates flange from a closed contour.

**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.

**smFlip** switches flange sides to reverse reference faces.
Concise Summary of Command Names

Inside BricsCAD V20

- **smForm** adds forms to sheet metal.
- **smHemCreate** creates a variety of hems on sheet metal models.
- **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.
- **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.
- **smSplit** splits flanges and lofted bend; replaces the old smFlangeSplit command.
- **smTabCreate** creates a tab between two flanges.
- **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\V20x64\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.

**UndoEnt** undoes property changes to selected 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.

**ViewHorizontal** rotates the viewpoint to make z=0 (horizontal)

**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”).

**VpMax** and **VpMin** maximize and minimize the current viewport in paper space.

**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)

VbaIde opens the BLADE editing window; short for “integrated development environment”.

VbaLoad and -VbaLoad loads VBA projects.

VbaMan manages VBA projects; short for “manager”.

VbaRun and -VbaRun runs, creates, edits, and deletes VBA macros.

VbaSecurity sets the security level for running VBA macros.

VbaUnload unloads VBA projects.

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 paper space layouts.

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, in an overlapping manner, or 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.

WsSaveAs 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 V20
- **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.
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACADLSPASDOC</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ACADPREFIX</td>
<td>r/o</td>
<td>“C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V20x64\en_US\Support;&quot;; C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Support; C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Fonts; C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Help\en_US”</td>
</tr>
<tr>
<td>ACADVER</td>
<td>r/o</td>
<td>“20.0 BricsCAD”</td>
</tr>
<tr>
<td>AcisHlrResolution</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>ACISOUTVER</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>AcisSaveAsMode</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AdaptiveGridStepSize</td>
<td>4.0000</td>
<td></td>
</tr>
<tr>
<td>AFLAGS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ALLOWBREAKLINECROSSINGS</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ALLOWEDBENDANGLES</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>AllowTabExternalMove</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AllowTabMove</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AllowTabSplit</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ANNGBASE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ANGDIR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Anglesamplinginterval</td>
<td>“5”</td>
<td></td>
</tr>
<tr>
<td>ANNOALLVISIBLE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ANNOAUTOSCALE</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>AnnoSelected</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>ANNOTATIVEWDG</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AntiAliasRender</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AntiAliasScreen</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>APBOX</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ArcTessellation</td>
<td>“0.01”</td>
<td></td>
</tr>
<tr>
<td>APERTURE</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>AREA</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>AREAPREC</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>AREAUNITS</td>
<td>“in ft mi µm mm cm m km”</td>
<td></td>
</tr>
<tr>
<td>ARRAYASSOCIATIVITY</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ARRAYEDITSTATE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>ARRAYTYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Associativity</td>
<td>“3”</td>
<td></td>
</tr>
<tr>
<td>ATTDIA</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ATTMODE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AttractionDistance</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>ATTREQ</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AUDITCTL</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AuditErrorCount</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>AUNITS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AUPREC</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AutoAdoptSizes</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>AUTOCOMPLETETIME</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>AUTOCOMPLETETYPE</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>AutoFlipQuarterTurn</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>AutomaticConnection</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>AutomaticTees</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>AUTOMENULOAD</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AutoResetScales</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AutosaveChecksOnlyFirstBitDBMOD</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
### Variable Name | Read-only | Default Value
--- | --- | ---
AUTOSNAP | | 119
AutoTrackingVecColor | | 171
AutoUpdateRooms | | “1”
AutoVpFitting | | 1
AXISMODE | | 0
AXISUNIT | | X= 0 Y= 0 Z= 0

#### B Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
</table>
BACKGROUNDPLOT | | 2
BAActionColor | | “7”
BACKZ | r/o | 0
BASEFILE | | “Default-mm.dwt”
BDependencyHighlight | | 1
bimConnectCutType | | “0”
bimMatchProp | | “1”
BIMOSMODE | | 0
BINDTYPE | | 0
BGUNIT | | X= 0 Y= 0 Z= 0
BVMODE | | 0
BBlocks | | 0
BlocksPath | | “C:\Users\userid\Documents”
bmAutoUpdate | | 1
bmForceUpdateMode | | 0
bmFormTemplatePath | | “”
bmReportPanel | | 0
BMUPDATEMODE | | 0
BndLimit | | 1000
BoundaryColor | | 95
BParameterSize | | 12
BpTextHorizontal | | 1
BSysLibCopyOverwrite | | 0
BtMarkDisplay | | 1

#### C Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
</table>
CACHELAYOUT | | 1
CAMERADISPLAY | | 0
CAMERAHEIGHT | | 0
CANNOSCALE | | “1:1”
CANNOSCALEVALUE | r/o | 1
CDATE | r/o | 20160211.15522
CECOLOR | | “ByLayer”
CELSCALE | | 1
CELTYPE | | “ByLayer”
CELWEIGHT | | -1
CenterCrossGap | | “0.05x”
CenterCrossSize | | “0.1x”
CenterExe | | 0.1200
CenterLayer | | “”
CenterLtscale | | 1.0000
CenterLtype | | “Center2”
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTERLTYPEFILE</td>
<td></td>
<td>&quot;Default.Lin&quot;</td>
</tr>
<tr>
<td>CenterMarkExe</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CETRANSPARENCY</td>
<td></td>
<td>&quot;ByLayer&quot;</td>
</tr>
<tr>
<td>CGEOCS</td>
<td>r/o</td>
<td>&quot;-&quot;</td>
</tr>
<tr>
<td>CHAMFERA</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CHAMFERB</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CHAMFERC</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CHAMFERD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CHAMMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ChapooLog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChapooLogVerbose</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChapooOnModified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChapooServer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChapooTempFolder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChapooUploadDependencies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChapooWebsite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CheckDwlPresence</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CIRCLERAD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CLAYER</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>CLEANSCREENOPTIONS</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>CLEANSCREENSTATE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>ClipBoardFormat</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CLIPBOARDFORMATS</td>
<td></td>
<td>127</td>
</tr>
<tr>
<td>ClIPromptLines</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>CLISTATE</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>CloseChecksOnlyFirstBitDBMOD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CloudDownloadPath</td>
<td></td>
<td>&quot;c:\users\userid\documents\bricsys247&quot;</td>
</tr>
<tr>
<td>CloudLog</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CloudLogVerbose</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CloudOnModified</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CloudServer</td>
<td></td>
<td>&quot;<a href="https://my.bricsys247.com/">https://my.bricsys247.com/</a>&quot;</td>
</tr>
<tr>
<td>CloudSessionPath</td>
<td></td>
<td>&quot;c:\users\userid\appdata\local\bricsys\bricscad&quot;</td>
</tr>
<tr>
<td>CloudTempFolder</td>
<td></td>
<td>&quot;C:\Users\userid\AppData\Local\Temp\Bricsys_24_7&quot;</td>
</tr>
<tr>
<td>CloudUploadDependencies</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CMATERIAL</td>
<td></td>
<td>&quot;ByLayer&quot;</td>
</tr>
<tr>
<td>CMDACTIVE</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>CMDDDIA</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CMDDECHO</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CmdLineEditBgColor</td>
<td></td>
<td>&quot;#fefe&quot;</td>
</tr>
<tr>
<td>CmdLineEditFgColor</td>
<td></td>
<td>&quot;#202020&quot;</td>
</tr>
<tr>
<td>CmdLineFontName</td>
<td></td>
<td>&quot;Consolas&quot;</td>
</tr>
<tr>
<td>CmdLineFontSize</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>CmdLineListBgColor</td>
<td></td>
<td>&quot;#ecf1ff&quot;</td>
</tr>
<tr>
<td>CmdLineListFgColor</td>
<td></td>
<td>&quot;#000000&quot;</td>
</tr>
<tr>
<td>CMDLNTEXT</td>
<td></td>
<td>&quot;-&quot;</td>
</tr>
<tr>
<td>CMDNAMES</td>
<td>r/o</td>
<td>&quot;SETTINGS&quot;</td>
</tr>
<tr>
<td>CMLHEADERSTYLE</td>
<td></td>
<td>&quot;Standard&quot;</td>
</tr>
<tr>
<td>CMLJUST</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CMLSCALE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CMLSTYLE</td>
<td></td>
<td>&quot;Standard&quot;</td>
</tr>
<tr>
<td>CMPCLRMISS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CMPCLRMOD1</td>
<td></td>
<td>253</td>
</tr>
<tr>
<td>CMPCLRMOD2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>CMPCLRNEW</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
### Concise Summary of System Variables and Settings

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPDIFFLIMIT</td>
<td></td>
<td>1000</td>
</tr>
<tr>
<td>CmpFadeCtl</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>CmpLog</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ColorTheme</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>COLORX</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>COLORY</td>
<td></td>
<td>112</td>
</tr>
<tr>
<td>COLORZ</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>COMAcadCompatibility</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>COMPASS</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ComponentsPath</td>
<td></td>
<td>“C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V20x64\en_US\Support\Bim\Components\”</td>
</tr>
<tr>
<td>CONSTRAINTBARDISPLAY</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>ContinuousMotion</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ConvertToArrays</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>COORDS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>COPYMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>CPLOTSTYLE</td>
<td></td>
<td>“ByColor”</td>
</tr>
<tr>
<td>CPROFILE</td>
<td>r/o</td>
<td>“Default”</td>
</tr>
<tr>
<td>CreateThumbnailOnTheFly</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CREATEVIEWPORTS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CrosshairDrawMode</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>CROSSINGAREACOLOR</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>CTAB</td>
<td></td>
<td>“Model”</td>
</tr>
<tr>
<td>CTABLESTYLE</td>
<td></td>
<td>“Standard”</td>
</tr>
<tr>
<td>Ctrl3DMouse</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CtrlMBUTTON</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CTRLMOUSE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CURSORSIZE</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>CVPORT</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

#### D Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataLinkNotify</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>DATE</td>
<td>r/o</td>
<td>2456335.6613464</td>
</tr>
<tr>
<td>DBCSTATE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>DBLCLKEDIT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DBMOD</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>DCTCUST</td>
<td></td>
<td>“en_US.dic”</td>
</tr>
<tr>
<td>DCTMAIN</td>
<td></td>
<td>“en_US.dic”</td>
</tr>
<tr>
<td>ddBetweenKnots</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>ddFastMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ddGridAspectRatio</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ddMaxFacetEdgeLength</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ddMaxNumGridLines</td>
<td></td>
<td>10000</td>
</tr>
<tr>
<td>ddNormalTol</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>ddPointsPerEdge</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ddSurfaceTol</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ddUseFacetRES</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DefaultBlockName</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DefaultInsPoint</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DEFAULTLIGHTING</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DefaultLightShadowBlur</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>DefaultNewSheetTemplate</td>
<td></td>
<td>“en_US.dic”</td>
</tr>
<tr>
<td>DefaultRoomHeight</td>
<td></td>
<td>“en_US.dic”</td>
</tr>
<tr>
<td>DEFLPLSTYLE</td>
<td></td>
<td>“Normal”</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>DEFPSTYLE</td>
<td></td>
<td>&quot;ByColor&quot;</td>
</tr>
<tr>
<td>DeleteInterference</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DeleteTool</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DELOBJ</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DEMANDLOAD</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>DGNFRAME</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>dgnimp2dClosedBSplineCurveImportMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimp2dEllipseImportMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimp2dShapeImportMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimp3dClosedBSplineCurveImportMode</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dgnimp3dEllipseImportMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimp3dObjectImportMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimp3dShapeImportMode</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dgnImpBreakDimensionAssociation</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnImpConvertDgnColorIndicesToTrueColors</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnImpConvertEmptyDataFieldsToSpaces</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dgnImpEraseUnusedResources</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimpExplodeTextNode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnimpImportActiveModelToModelSpace</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dgnimpImportInvisibleElements</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dgnimpImportPaperSpaceModels</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dgnimpImportViewIndex</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>dgnImpRecomputeDimensionsAfterImport</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dgnImpSymbolResourceFiles</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>dgnImpXRefImportMode</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>DGNOSNAP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIASTAT</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>DisplayAxes</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>DisplayAxesForMep</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>DisplayScaling</td>
<td>r/o</td>
<td>125</td>
</tr>
<tr>
<td>DisplaySidesAnd Ends</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>DisplaySnapMarkerInAllViews</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DisplayTooltips</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DISPPAPERBKG</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DISPPAPERMARGINS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DISPSILH</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>dmAuditLevel</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DMAUTOUPDATE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dmConnectionCutType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dmExtrudeMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>dmPushPullSubtract</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DMRECOGNIZE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DockPriority</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DocTabPosition</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DONUTID</td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>DONUTOD</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DRAGMODE</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>DragModeHide</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DRAGMODEINTERRUPT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DRAGOPEN</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DRAGP1</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>DRAGP2</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>DRAGSNAP</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
## Variable Name

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrawingPath</td>
<td></td>
<td>&quot;C:\Users\userid\Documents&quot;</td>
</tr>
<tr>
<td>DrawingViewPreset</td>
<td></td>
<td>&quot;none&quot;</td>
</tr>
<tr>
<td>DrawingViewPresetHidden</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DrawingViewPresetScale</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>DrawingViewpPresetTangent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DrawingViewPresetScale</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>DrawingViewPresetTangent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DRAWORDERCTL</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>DxfTextAdjustAlignment</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DWFFRAME</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>DWFOSNAP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DwfVersion</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>DWGCHECK</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DWGCODEPAGE</td>
<td>r/o</td>
<td>&quot;ANSI_1252&quot;</td>
</tr>
<tr>
<td>DWGNAME</td>
<td>r/o</td>
<td>&quot;Drawing1.dwg&quot;</td>
</tr>
<tr>
<td>DWGNAME</td>
<td>r/o</td>
<td>&quot;C:\Program Files (x86)\Bricsys\BricsCAD V20x64&quot;</td>
</tr>
<tr>
<td>DWGTITLED</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>DXEVAL</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>DxfTextAdjustAlignment</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DYNCONSTRAINTMODE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DYNDIGRIP</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>DynDimAperture</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>DynDimColorHot</td>
<td></td>
<td>142</td>
</tr>
<tr>
<td>DynDimColorHover</td>
<td></td>
<td>142</td>
</tr>
<tr>
<td>DynDimDistance</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DynDimLineType</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DYNDIVIS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DynInputTransparency</td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>DYNMODE</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>DynPiCoords</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

### Dimensions Variables

<p>| 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                                       |</p>
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMCLRD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMCLRE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMCLRT</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMDEC</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>DIMDLE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMDLI</td>
<td></td>
<td>0.38</td>
</tr>
<tr>
<td>DIMDSEP</td>
<td></td>
<td>“0”</td>
</tr>
<tr>
<td>DIMEXE</td>
<td></td>
<td>0.18</td>
</tr>
<tr>
<td>DIMEXO</td>
<td></td>
<td>0.0625</td>
</tr>
<tr>
<td>DIMFIT</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>DIMFRAC</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMFXL</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMFXLON</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMGAP</td>
<td></td>
<td>0.09</td>
</tr>
<tr>
<td>DIMJOGANG</td>
<td></td>
<td>0.7853981634</td>
</tr>
<tr>
<td>DIMJUST</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Dimlayer</td>
<td></td>
<td>“-“</td>
</tr>
<tr>
<td>DIMLDRBLK</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>DIMLFAC</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMLIM</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMLTEX1</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>DIMLTEX2</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>DIMLTYPE</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>DIMLUNIT</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>DIMLWD</td>
<td></td>
<td>-2</td>
</tr>
<tr>
<td>DIMLWE</td>
<td></td>
<td>-2</td>
</tr>
<tr>
<td>DIMPOST</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>DIMRND</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSAH</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSCALE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMSD1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSD2</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSE1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSE2</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSHO</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMSOXD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMSTYLE</td>
<td>r/o</td>
<td>“Standard”</td>
</tr>
<tr>
<td>DIMTAD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTDEC</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>DIMTFAC</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMTFILL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTFILLCLR</td>
<td></td>
<td>“BYBLOCK”</td>
</tr>
<tr>
<td>DIMTIH</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMTIX</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTM</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTMOVE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTOFL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTOH</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMTOL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTOLJ</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIMTP</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTSZ</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTVP</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>DIMTXSTY</td>
<td></td>
<td>“Standard”</td>
</tr>
</tbody>
</table>
### Variable Name

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMTXT</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>DIMTXTDIRECTION</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DIMTZIN</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DIMUNIT</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>DIMUPT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DIMZIN</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>E Variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDGEMODE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ELEVATION</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ElevationAtBreaklineCrossings</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>EnableAttraction</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EnableBimBkUpdate</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>EnableHyperlinkMenu</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EnableHyperlinkTooltip</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ERRNO</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>EXPERT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ExpInsAlign</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ExpInsAngle</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ExpInsFixAngle</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ExpInsFixScale</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ExpInsScale</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EXPLMODE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ExportAcisFormatVersion</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ExportCatiav4FormatVersion</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ExportCatiav5FormatVersion</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ExportHiddenParts</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>EXPORTMODELSPACE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>EXPORTPAGESETUP</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>EXPORTPAPERSPACE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ExportParasolidFormatVersion</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ExportProductStructure</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ExportStepFormatVersion</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>EXTMAX</td>
<td>r/o -1.0000E+20,-1.0000E+20,-1.0000E+20</td>
<td></td>
</tr>
<tr>
<td>EXTMIN</td>
<td>r/o 1.0000E+20,1.0000E+20,1.0000E+20</td>
<td></td>
</tr>
<tr>
<td>EXTNAMES</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F Variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FACETRATIO</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>FACETRES</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>fbxExportCameras</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>fbxExportEntities</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>fbxExportEntitiesSelType</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>fbxExportLights</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>fbxExportMaterials</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>fbxExportTextures</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>fbxExportTexturesPath</td>
<td>“c:\program files\bricsys\bricscad v20 en_us”</td>
<td></td>
</tr>
<tr>
<td>FEATURECOLORS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FIELDDISPLAY</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FIELDEVAL</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>FILEDIA</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FILLETRAD</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>FILLMODE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>FittingRadiusType</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>FittingRadiusValue</td>
<td>“1.5”</td>
<td></td>
</tr>
<tr>
<td>FLATLAND</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>FONALT</td>
<td>“simplex.shx”</td>
<td></td>
</tr>
<tr>
<td>FONMAP</td>
<td>“default.fmp”</td>
<td></td>
</tr>
<tr>
<td>FRAME</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>FRAMESELECTION</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>FRONTZ</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>FULLOPEN</td>
<td>r/o</td>
<td>1</td>
</tr>
</tbody>
</table>

**G Variables**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDIOBJECTS</td>
<td>r/o</td>
<td>3768</td>
</tr>
<tr>
<td>GearteethNumber</td>
<td></td>
<td>“1”</td>
</tr>
<tr>
<td>GENERATEASSOCVIEW</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GEOLATLONGFORMAT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>GEOMARKERVISIBILITY</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>GeomRelations</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GetStarted</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>GfAng</td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td>GfClr1</td>
<td></td>
<td>“5”</td>
</tr>
<tr>
<td>GfClr2</td>
<td></td>
<td>“7”</td>
</tr>
<tr>
<td>GfClrLum</td>
<td></td>
<td>1.0000</td>
</tr>
<tr>
<td>GfClrState</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GfName</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>GfShift</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GLSWAPMODE</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>GradientColorBottom</td>
<td></td>
<td>“#d2d2d2”</td>
</tr>
<tr>
<td>GradientColorMiddle</td>
<td></td>
<td>“#fafafa”</td>
</tr>
<tr>
<td>GradientColorTop</td>
<td></td>
<td>“#ffffff”</td>
</tr>
<tr>
<td>GradientMode</td>
<td></td>
<td>“0”</td>
</tr>
<tr>
<td>GRIDAXISCOLOR</td>
<td></td>
<td>252</td>
</tr>
<tr>
<td>GRADDISPLAY</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>GRIDMAJOR</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>GRIDMAJORORCOLOR</td>
<td></td>
<td>253</td>
</tr>
<tr>
<td>GRIDMINORCOLOR</td>
<td></td>
<td>254</td>
</tr>
<tr>
<td>GRIDMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GRIDSTYLE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GRIDUNIT</td>
<td></td>
<td>1/2&quot;,1/2“</td>
</tr>
<tr>
<td>GRIDXYZT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>GRIPBLOCK</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GRIPCOLOR</td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>GRIPDYNCOLOR</td>
<td></td>
<td>140</td>
</tr>
<tr>
<td>GRIPHOT</td>
<td></td>
<td>240</td>
</tr>
<tr>
<td>GRIPHOVER</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>GRIPOBJLIMIT</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>GRIPS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>GRIPSIZE</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>GRIPTIPS</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**GsDeviceType**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GsDeviceType2D</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>GsDeviceType3D</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td><strong>H Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HALOGAP</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HANDLES</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>HANDSEED</td>
<td></td>
<td>“64”</td>
</tr>
<tr>
<td>Headroom</td>
<td></td>
<td>“2000”</td>
</tr>
<tr>
<td>HIDEPRECISION</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HideSystemPrinters</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HIDETEXT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HидEXREFScales</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HIGHLIGHT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HIGHLIGHTCOLOR</td>
<td></td>
<td>142</td>
</tr>
<tr>
<td>HIGHLIGHTEFFECT</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HomeGradientColorBottom</td>
<td></td>
<td>replaced by HorizonBkg_SkyLow</td>
</tr>
<tr>
<td>HomeGradientColorMiddle</td>
<td></td>
<td>replaced by HorizonBkg_SkyHorizon</td>
</tr>
<tr>
<td>HomeGradientColorTop</td>
<td></td>
<td>replaced by HorizonBkg_SkyHigh</td>
</tr>
<tr>
<td>HomeGradientMode</td>
<td></td>
<td>replaced by HorizonBkg_Enable</td>
</tr>
<tr>
<td>HorizonBkg_Enable</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HorizonBkg_GroundHorizon</td>
<td></td>
<td>“#878787”</td>
</tr>
<tr>
<td>HorizonBkg_GroundOrigin</td>
<td></td>
<td>“#5F5F5F”</td>
</tr>
<tr>
<td>HorizonBkg_SkyHigh</td>
<td></td>
<td>“#239BFF”</td>
</tr>
<tr>
<td>HorizonBkg_SkyHorizon</td>
<td></td>
<td>“#FFFFFF”</td>
</tr>
<tr>
<td>HorizonBkg_SkyLow</td>
<td></td>
<td>“#FAFAFF”</td>
</tr>
<tr>
<td>HotkeyAssistant</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HPANG</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPLANOTATIVE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPASSOC</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HpBackgroundColor</td>
<td></td>
<td>“-“</td>
</tr>
<tr>
<td>HpColor</td>
<td></td>
<td>“-“</td>
</tr>
<tr>
<td>HPBOUND</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HPBOUNDRETAIN</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPDOUBLE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPDRAWORDER</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>HPGAPTOL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPLAYER</td>
<td></td>
<td>“-“</td>
</tr>
<tr>
<td>HLINE TYPE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HpdIslandDetection</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPMAXAREAS</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPNAME</td>
<td></td>
<td>“--”</td>
</tr>
<tr>
<td>HPOBJECTWARNING</td>
<td></td>
<td>10000</td>
</tr>
<tr>
<td>HPORIGIN</td>
<td></td>
<td>“.0,.0”</td>
</tr>
<tr>
<td>HPSCALE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HPSEPARATE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>HPSPACE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>HPSTYLE</td>
<td></td>
<td>replaced by HpdIslandDetection</td>
</tr>
<tr>
<td>HPTransparency</td>
<td></td>
<td>“-“</td>
</tr>
<tr>
<td>HYPERLINKBASE</td>
<td></td>
<td>“--”</td>
</tr>
<tr>
<td><strong>I Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IfcExplodeExternalReferences</td>
<td></td>
<td>“0”</td>
</tr>
<tr>
<td>IfcExportBaseQuantities</td>
<td></td>
<td>“0”</td>
</tr>
<tr>
<td>IfcExportElementsOnOffAndFrozenLayer</td>
<td></td>
<td>“1”</td>
</tr>
<tr>
<td>IfcExportMultiplyElementsAsAggregated</td>
<td></td>
<td>“0”</td>
</tr>
<tr>
<td>IfcExportTessellation</td>
<td></td>
<td>“0”</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>IfcImportBimData</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>IfcImportBrepGeometryAsMeshes</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>IfcImportModelOrigin</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>IfcImportParametricComponents</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImportIfcProjectStructureAsXrefs</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>IfcImportSpaces</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImageCacheFolder</td>
<td>“C:\Users\userid\AppData\Local\Temp\ImageCache\”</td>
<td></td>
</tr>
<tr>
<td>ImageCacheMaxMemory</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>ImageDiskCache</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IMAGEFRAME</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IMAGEHLT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ImageNotify</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ImportColors</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ImportCreoAlternateSearchPaths</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>ImportCuiFileExists</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ImportHiddenparts</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImportIfcProjectStructureAsXrefs</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImportigesSimplify</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ImportigesStitch</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ImportInventorAlternateSearchPaths</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>ImportNxAbsoluteSearchPaths</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>ImportPmi</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ImportProductStructure</td>
<td>“2”</td>
<td></td>
</tr>
<tr>
<td>ImportRepair</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImportSimplify</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImportSolidedgeAlternateSearchPaths</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>ImportSolidworksAlternateSearchPaths</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>ImportSolidworksRotateYz</td>
<td>“1”</td>
<td></td>
</tr>
<tr>
<td>ImportStepRotateYz</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>ImportStitch</td>
<td>“0”</td>
<td></td>
</tr>
<tr>
<td>IncludePlotStamp</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>INEXCCTL</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>INETLOCATION</td>
<td>“<a href="http://www.bricsys.com%E2%80%9D">http://www.bricsys.com”</a></td>
<td></td>
</tr>
<tr>
<td>INSBASE</td>
<td>“0&quot;,0&quot;,0”</td>
<td></td>
</tr>
<tr>
<td>INSNAME</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>INSUNITS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>INSUNITSDFSOURCE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>INSUNITSDFTARGET</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>InsUnitsScaling</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>INTERFERENCECOLOR</td>
<td>“BYLAYER”</td>
<td></td>
</tr>
<tr>
<td>InterfereLayer</td>
<td>“Interference”</td>
<td></td>
</tr>
<tr>
<td>INTERFERENCEOBJVS</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>INTERFERENCEPVS</td>
<td>“--”</td>
<td></td>
</tr>
<tr>
<td>InteriorElevationMinLength</td>
<td>“20”</td>
<td></td>
</tr>
<tr>
<td>InteriorElevationOffset</td>
<td>“2”</td>
<td></td>
</tr>
<tr>
<td>INTERSECTIONCOLOR</td>
<td>257</td>
<td></td>
</tr>
<tr>
<td>INTERSECTIONDISPLAY</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ISAVEBAK</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ISAVEPERCENT</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>ISOLINES</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**L Variables**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASTANGLE</td>
<td>0</td>
</tr>
<tr>
<td>LASTPO</td>
<td>“0&quot;,0&quot;,0”</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>LASTPROMPT</td>
<td>r/o</td>
</tr>
<tr>
<td>LATITUDE</td>
<td></td>
</tr>
<tr>
<td>LayerFilterExcess</td>
<td></td>
</tr>
<tr>
<td>LAYERMODE</td>
<td></td>
</tr>
<tr>
<td>LAYLOCKFADECTL</td>
<td></td>
</tr>
<tr>
<td>LAYOUTREGENCTL</td>
<td></td>
</tr>
<tr>
<td>LayoutTab</td>
<td></td>
</tr>
<tr>
<td>LegacyCodeSearch</td>
<td>r/o</td>
</tr>
<tr>
<td>LengthSamplingInterval</td>
<td></td>
</tr>
<tr>
<td>LENGTHUNITS</td>
<td></td>
</tr>
<tr>
<td>LENSLength</td>
<td></td>
</tr>
<tr>
<td>LevelOfDetail</td>
<td></td>
</tr>
<tr>
<td>LicExpDays</td>
<td></td>
</tr>
<tr>
<td>LICFLAGS</td>
<td></td>
</tr>
<tr>
<td>LICKEY</td>
<td>r/o</td>
</tr>
<tr>
<td>LightGlyphColor</td>
<td></td>
</tr>
<tr>
<td>LIGHTGLYPHDISPLAY</td>
<td></td>
</tr>
<tr>
<td>LIGHTINGUNITS</td>
<td></td>
</tr>
<tr>
<td>LightWebGlyphColor</td>
<td></td>
</tr>
<tr>
<td>LIMCHECK</td>
<td></td>
</tr>
<tr>
<td>LIMMAX</td>
<td></td>
</tr>
<tr>
<td>LIMMIN</td>
<td></td>
</tr>
<tr>
<td>LINEARBRIGHTNESS</td>
<td></td>
</tr>
<tr>
<td>LINEARCONTRAST</td>
<td></td>
</tr>
<tr>
<td>LISPINIT</td>
<td></td>
</tr>
<tr>
<td>LoadMechanical2d</td>
<td></td>
</tr>
<tr>
<td>LOCALE</td>
<td></td>
</tr>
<tr>
<td>LocalRootFolder</td>
<td></td>
</tr>
<tr>
<td>LOCALROOTPREFIX</td>
<td>r/o</td>
</tr>
<tr>
<td>LOCKUI</td>
<td></td>
</tr>
<tr>
<td>LOFTANG1</td>
<td></td>
</tr>
<tr>
<td>LOFTANG2</td>
<td></td>
</tr>
<tr>
<td>LOFTMAG1</td>
<td></td>
</tr>
<tr>
<td>LOFTMAG2</td>
<td></td>
</tr>
<tr>
<td>LOFTNORMALS</td>
<td></td>
</tr>
<tr>
<td>LOFTPARAM</td>
<td></td>
</tr>
<tr>
<td>LOGFILEMODE</td>
<td></td>
</tr>
<tr>
<td>LOGFILENAME</td>
<td>r/o</td>
</tr>
<tr>
<td>LOGFILEPATH</td>
<td>r/o</td>
</tr>
<tr>
<td>LOGINNAME</td>
<td>r/o</td>
</tr>
<tr>
<td>LONGITUDE</td>
<td></td>
</tr>
<tr>
<td>LookFromDirectionMode</td>
<td></td>
</tr>
<tr>
<td>LookFromFeedback</td>
<td></td>
</tr>
<tr>
<td>LookFromZoomExtents</td>
<td></td>
</tr>
<tr>
<td>LTScale</td>
<td></td>
</tr>
<tr>
<td>LUNITS</td>
<td></td>
</tr>
<tr>
<td>LUPREC</td>
<td></td>
</tr>
<tr>
<td>LWDEFAULT</td>
<td></td>
</tr>
<tr>
<td>LWDISPLAY</td>
<td></td>
</tr>
<tr>
<td>LWDISPSSCALE</td>
<td></td>
</tr>
<tr>
<td>LWUNITS</td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>M Variables</strong></td>
<td></td>
</tr>
<tr>
<td>M_axisDiameter</td>
<td></td>
</tr>
<tr>
<td>M_totalAxisLength</td>
<td></td>
</tr>
<tr>
<td>MACROREC</td>
<td></td>
</tr>
<tr>
<td>MACROTRACE</td>
<td></td>
</tr>
<tr>
<td>MANIPULATOR</td>
<td></td>
</tr>
<tr>
<td>ManipulatorColorTheme</td>
<td></td>
</tr>
<tr>
<td>MANIPULATORDURATION</td>
<td></td>
</tr>
<tr>
<td>ManipulatorSize</td>
<td></td>
</tr>
<tr>
<td>MassPrec</td>
<td></td>
</tr>
<tr>
<td>MassPropAccuracy</td>
<td></td>
</tr>
<tr>
<td>MASSUNITS</td>
<td></td>
</tr>
<tr>
<td>MaterialsPath</td>
<td></td>
</tr>
<tr>
<td>MAXACTVP</td>
<td></td>
</tr>
<tr>
<td>MAXHATCH</td>
<td></td>
</tr>
<tr>
<td>MAXSORT</td>
<td></td>
</tr>
<tr>
<td>MAXTHREADS</td>
<td></td>
</tr>
<tr>
<td>MBSTATE</td>
<td>r/o</td>
</tr>
<tr>
<td>MBUTTONPAN</td>
<td></td>
</tr>
<tr>
<td>MEASUREINIT</td>
<td></td>
</tr>
<tr>
<td>MEASUREMENT</td>
<td></td>
</tr>
<tr>
<td>Mech2dSaveFormat</td>
<td></td>
</tr>
<tr>
<td>MENUBAR</td>
<td></td>
</tr>
<tr>
<td>MENUCTL</td>
<td></td>
</tr>
<tr>
<td>MENUECHO</td>
<td></td>
</tr>
<tr>
<td>MENUNAME</td>
<td>r/o</td>
</tr>
<tr>
<td>MESHTYPE</td>
<td></td>
</tr>
<tr>
<td>MiddleClickClose</td>
<td></td>
</tr>
<tr>
<td>MILLISECS</td>
<td>r/o</td>
</tr>
<tr>
<td>MIRRTEXT</td>
<td></td>
</tr>
<tr>
<td>MLEADERSCALE</td>
<td></td>
</tr>
<tr>
<td>MODEMACRO</td>
<td></td>
</tr>
<tr>
<td>MSLTSSCALE</td>
<td></td>
</tr>
<tr>
<td>MSOLESCALE</td>
<td></td>
</tr>
<tr>
<td>MTEXTCOLUMN</td>
<td></td>
</tr>
<tr>
<td>MTEXTDETECTSPACE</td>
<td></td>
</tr>
<tr>
<td>MTEXTED</td>
<td></td>
</tr>
<tr>
<td>MTEXTED</td>
<td></td>
</tr>
<tr>
<td>MTEXTED</td>
<td></td>
</tr>
<tr>
<td>MTEXTFIXED</td>
<td></td>
</tr>
<tr>
<td>MTFLAGS</td>
<td></td>
</tr>
<tr>
<td>MultiSelectAngularTolerance</td>
<td></td>
</tr>
<tr>
<td>MyDocumentsFolder</td>
<td></td>
</tr>
<tr>
<td>MYDOCUMENTSPREFIX</td>
<td>r/o</td>
</tr>
<tr>
<td><strong>N Variables</strong></td>
<td></td>
</tr>
<tr>
<td>NEARESTDISTANCE</td>
<td></td>
</tr>
<tr>
<td>NAVVCUBEDISPLAY</td>
<td></td>
</tr>
<tr>
<td>NAVVCUBELOCATION</td>
<td></td>
</tr>
<tr>
<td>NAVVCUBE POLYTYCEITY</td>
<td></td>
</tr>
<tr>
<td>NAVVCUBEORIENT</td>
<td></td>
</tr>
<tr>
<td>NavVCubeSize</td>
<td></td>
</tr>
<tr>
<td>NFILELIST</td>
<td></td>
</tr>
<tr>
<td>NOMUTT</td>
<td></td>
</tr>
</tbody>
</table>
### O Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTISOLATIONMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OBSCURED COLOR</td>
<td></td>
<td>257</td>
</tr>
<tr>
<td>OBSCURED LTYPE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OFFSET DIST</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>OFFSET ERASE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OFFSET GAP TYPE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OLE FRAME</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>OLE HIDE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OLE QUALITY</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OLE STARTUP</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OPM STATE</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>Orbit Auto Target</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ORTHOMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OSMODE</td>
<td></td>
<td>4133</td>
</tr>
<tr>
<td>OSNAP COORD</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>OSNAP Z</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>OS OPTIONS</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

### P Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan Buffer</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Panel Button Size</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>PAPER UPDATE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PARAMETER COPY MODE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Cache</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Pdf Embedded TTF</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Export Solid Hatch Type</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>PDF FRAME</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Hatch to Bmp Dpi</td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>Pdf Image Anti Alias</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Image Compression</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Image DPI</td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>Pdf Import Apply Lineweight</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Import As Block</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Import Convert Solids To Hatches</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Import Image Path</td>
<td></td>
<td>&quot;pdf images&quot;</td>
</tr>
<tr>
<td>Pdf Import Join Line And Arc Segments</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Import Layers Use Type</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Import Raster Images</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Import Solid Fills</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Import True Type Text</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Import True Type Text As Geometry</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Import Use Geometry Optimization</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Import Vector Geometry</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Layers Setting</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Layouts To Export</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Merge Control</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pdf Notify</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PDF OS SNAP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Pdf Paper Height</td>
<td></td>
<td>297</td>
</tr>
<tr>
<td>Pdf Paper Size Override</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>PdfPaperWidth</td>
<td></td>
<td>210</td>
</tr>
<tr>
<td>PdfPRCCompression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PdfPRCExport</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PdfPRCSingleViewMode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PdfRenderDPI</td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>PdfShxTextAsGeometry</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PdfSimpleGeomOptimization</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PdfTextSearchable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PdfTtfTextAsGeometry</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PdfUsePlotStyles</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PdfVectorResolutionDpi</td>
<td></td>
<td>2400</td>
</tr>
<tr>
<td>PdfZoomToExtentsMode</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PDMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PDSIZE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PEDITACCEPT</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PELLIPSE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PERIMETER</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PERSPECTIVE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PFACEVMAX</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PICKADD</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PICKAUTO</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PICKBOX</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PICKDRAG</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PICKFIRST</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PICKSTYLE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PictureExportScale</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>_PKSER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PlacesBarFolder1</td>
<td>r/o</td>
<td></td>
</tr>
<tr>
<td>PlacesBarFolder2</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PlacesBarFolder3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PlacesBarFolder4</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>PLATFORM</td>
<td>r/o</td>
<td>Microsoft Windows NT Version 6.2</td>
</tr>
<tr>
<td>PLINECACHE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PLINECONVERTMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PLINETYPE</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>PLINETYPE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PLINETYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PlotCfgPath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PlotID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PlotOutputPath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOTROTMODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PlotStylePath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOTTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOTTRANSPARENCYOVERRIDE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PLQUIET</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Pointcloud2dVsDisplay</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PointcloudBoundary</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PointcloudCacheFolder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PointcloudPointMax</td>
<td></td>
<td>40000000</td>
</tr>
<tr>
<td>PointcloudPointSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POLARADDANG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POLARANG</td>
<td></td>
<td>90</td>
</tr>
<tr>
<td>POLARDIST</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>POLARMODE</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
### Variable Name

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLYSIDES</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>POPUPS</td>
<td>r/o 1</td>
<td></td>
</tr>
<tr>
<td>PpState</td>
<td>r/o 0</td>
<td></td>
</tr>
<tr>
<td>PreviewDelay</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>PREVIEWEFFECT</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>PREVIEWFILTER</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>PreviewTopdown</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREVIEWTYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>PreviewWndInOpenDlg</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PrintFile</td>
<td>&quot; &quot;</td>
<td></td>
</tr>
<tr>
<td>PrintPdfPreview</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PRODUCT</td>
<td>r/o “Bricscad”</td>
<td></td>
</tr>
<tr>
<td>PROGBAR</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROGRAM</td>
<td>r/o “BRICSCAD”</td>
<td></td>
</tr>
<tr>
<td>PROJECTIONTYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>PROJECTNAME</td>
<td>&quot; &quot;</td>
<td></td>
</tr>
<tr>
<td>ProjectSearchPaths</td>
<td>&quot; &quot;</td>
<td></td>
</tr>
<tr>
<td>PROJMODE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROMPTMENU</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>PromptMenuFlags</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PromptOptionFormat</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>PromptOptionTranslateKeywords</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PropertyPreview</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PropertyPreviewDelay</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>PropertyPreviewObjLimit</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>PropPrevTimeout</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROPUNITS</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>PropUnitsVersion</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROXYGRAPHICS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROXYNOTICE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROXYSHOW</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PROXYWEBSEARCH</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PSLTSCALE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PSOLHEIGHT</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>PSOLWIDTH</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>PSTYLEMODE</td>
<td>r/o 1</td>
<td></td>
</tr>
<tr>
<td>PSTYLEPOLICY</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PSVPSCALE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>PUBLISHALLSHEETS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PUCSBASE</td>
<td>&quot; &quot;</td>
<td></td>
</tr>
</tbody>
</table>

### Q Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QAFLAGS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QTEXTMODE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QuadAperture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QuadCommandLaunch</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>QuadCommandSort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QuadDisplay</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QuadExpandDelay</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>QuadExpandGroup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QuadExpandTabDelay</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>QuadGoTransparent</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QuadHideDelay</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>QuadHideMargin</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>QuadIconSize</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>QuadIconSpace</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>QuadMostRecentItems</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>QuadPopupCorner</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>QuadShowDelay</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>QuadTabFlags</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>QuadTooltipDelay</td>
<td></td>
<td>1200</td>
</tr>
<tr>
<td>QuadWarpPointer</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>QuadWidth</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**R Variables**

R12SaveAccuracy
R12SaveDeviation
RASTERPREVIEW
RE_INIT
RealTimeSpeedUp
REALWORLDSCALE
RecentPath
"C:\Users\userid\Documents\"
Redhilite_ducslocked_face_alpha
25
Redhilite_ducslocked_face_color
"#007aff"
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"
RedHiliteFull_Face_Alpha
10
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_SelectedFace_ShowHidden
1
RedHilitePartial_SelectedFace_Smoothing
1
RedHilitePartial_SelectedFace_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
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFEDITNAME</td>
<td>r/o</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>REGENMODE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RegExpand</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>REMEMBERFOLDERS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RenderMaterialDownload</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RenderMaterialPath</td>
<td></td>
<td>&quot;C:\ProgramData...&quot;</td>
</tr>
<tr>
<td>RenderMaterialStaticPath</td>
<td></td>
<td>&quot;C:\Program Files...&quot;</td>
</tr>
<tr>
<td>RenderUsingHardware</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ReportPanel</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ReportPanelMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RestoreConnections</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>RevCloudArcStyle</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>REV CLOU D CREAT EMODE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>REV CLOUDGRIPS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RevCloudMaxArcLength</td>
<td></td>
<td>0.375</td>
</tr>
<tr>
<td>RevCloudMinArcLength</td>
<td></td>
<td>0.375</td>
</tr>
<tr>
<td>RhinoVersion</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>RIBBON DOCKEDHEIGHT</td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>RIBBON STATE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>Riserheight</td>
<td></td>
<td>&quot;170&quot;</td>
</tr>
<tr>
<td>RoamableRootFolder</td>
<td>r/o</td>
<td>&quot;c:\users...&quot;</td>
</tr>
<tr>
<td>ROAMABLEROOTPREFIX</td>
<td>r/o</td>
<td>&quot;C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V20x64\en_US&quot;</td>
</tr>
<tr>
<td>ROLLOVEROPA C I TY</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>ROLLOVERTIPS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RolloverSelectionSet</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Roomheight</td>
<td></td>
<td>&quot;3000&quot;</td>
</tr>
<tr>
<td>RT DISPLAY</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RTRotationSpeedFactor</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RT Walk Speed Factor</td>
<td></td>
<td>replaced by the StepSize variable</td>
</tr>
<tr>
<td>RubberbandColor</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>RubberbandStyle</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RunAsLevel</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

**S Variables**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safemode</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>SaveChangeToLayout</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SAVEFIDELITY</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SAVEFILE</td>
<td>r/o</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SAVEFILEPATH</td>
<td></td>
<td>&quot;C:\Users\userid\AppData\Local\Temp&quot;</td>
</tr>
<tr>
<td>SaveFormat</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SaveLayerSnapshot</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SAVENAME</td>
<td>r/o</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SaveOnDocSwitch</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SAVEROUNDTRIP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SAVETIME</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>SCREENBOXES</td>
<td>r/o</td>
<td>26</td>
</tr>
<tr>
<td>SCREENMODE</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>SCREENSIZE</td>
<td>r/o</td>
<td>145'-8&quot;,73'-3&quot;</td>
</tr>
<tr>
<td>SCRLHIST</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>SDI</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SearchAll</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SectionScale</td>
<td></td>
<td>&quot;0.02&quot;</td>
</tr>
<tr>
<td>SectionSettingsSearchPath</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SectionSheetsetTemplateImperial</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>SectionSheetsetTemplateMetric</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>SecureLoad</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>SELECTIONANNODISPLAY</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SELECTIONAREA</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SELECTIONAREAOPACITY</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>SelectionModes</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SELECTIONPREVIEW</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>SELECTSIMILARMODE</td>
<td></td>
<td>130</td>
</tr>
<tr>
<td>SHAEDGE</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>SHADEDIF</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>SheetNumberLeadingZeroes</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SheetSetAutoBackup</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SheetSetTemplatePath</td>
<td></td>
<td>&quot;C:\Users\userid\AppData\Local\Bricsys\BricsCAD\V20x64\en_US\Templates\Sheet Sets&quot;</td>
</tr>
<tr>
<td>SHORTCUTMENU</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>SHORTCUTMENUDURATION</td>
<td></td>
<td>250</td>
</tr>
<tr>
<td>ShowDocTabs</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ShowFullPathInTitle</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SHOWLAYERUSAGE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ShowScrollButtons</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ShowTabCloseButton</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ShowTabCloseButtonActive</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ShowTabCloseButtonAll</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ShowTabControls</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ShowWindowListButton</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SHPNAME</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>SingletonMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SKETCHINC</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>SKPOLY</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SkpStitch</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SKYSTATUS</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SlabThickness</td>
<td></td>
<td>&quot;250&quot;</td>
</tr>
<tr>
<td>smAssemblyExportReportPathType</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smAssemblyExportSolidTypesInReports</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>smAttributesLayerColor</td>
<td></td>
<td>&quot;7&quot;</td>
</tr>
<tr>
<td>smAttributesLayerTextheight</td>
<td></td>
<td>&quot;0.01&quot;</td>
</tr>
<tr>
<td>smAttributesLayerTextHeightType</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smBendAnnotationsLayerColor</td>
<td></td>
<td>&quot;5&quot;</td>
</tr>
<tr>
<td>smBendAnnotationsLayerTextHeight</td>
<td></td>
<td>&quot;0.01&quot;</td>
</tr>
<tr>
<td>smBendAnnotationsLayerTextHeightType</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smBendlinesDownlayerColor</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>smBendlinesDownlayerLinetype</td>
<td></td>
<td>&quot;Continuous&quot;</td>
</tr>
<tr>
<td>smBendlinesDownlayerLinewidth</td>
<td></td>
<td>&quot;-3&quot;</td>
</tr>
<tr>
<td>smBendlinesUplayerColor</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>smBendlinesUplayerLinetype</td>
<td></td>
<td>&quot;Continuous&quot;</td>
</tr>
<tr>
<td>smBendlinesUplayerLineweight</td>
<td></td>
<td>&quot;-3&quot;</td>
</tr>
<tr>
<td>SMCOLORBEND</td>
<td></td>
<td>&quot;#FFDC50&quot;</td>
</tr>
<tr>
<td>SMCOLORBENDRELIEF</td>
<td></td>
<td>&quot;#64D296&quot;</td>
</tr>
<tr>
<td>SMCOLORCORNERRELIEF</td>
<td></td>
<td>&quot;#64D296&quot;</td>
</tr>
<tr>
<td>SMCOLORFLANGE</td>
<td></td>
<td>&quot;#90A4AE&quot;</td>
</tr>
<tr>
<td>smcolorflangereferenceside</td>
<td></td>
<td>&quot;#68a4ae&quot;</td>
</tr>
<tr>
<td>Smcolorform</td>
<td></td>
<td>&quot;#8791e1&quot;</td>
</tr>
<tr>
<td>SMCOLORJUNCTION</td>
<td></td>
<td>&quot;#F6E40&quot;</td>
</tr>
<tr>
<td>SMCOLORLOFTEDBEND</td>
<td></td>
<td>&quot;#A0DCFA&quot;</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>smColorMiter</td>
<td></td>
<td>#af46d8</td>
</tr>
<tr>
<td>smColorTab</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>smColorWrongBend</td>
<td></td>
<td>#ff3300</td>
</tr>
<tr>
<td>smContoursLayerColor</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>smContoursLayerLinetype</td>
<td></td>
<td>&quot;continuous&quot;</td>
</tr>
<tr>
<td>smContoursLayerLineweight</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>smConvertPreferFormFeatures</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smConvertRecognizeHoles</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smConvertRecognizeRibControlCurves</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smConvertWrongFeatureThicknessDeviationType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smConvertWrongFeatureThicknessDeviationValue</td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td>smDefaultbendlineextenttypetype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultbendlineextentvalue</td>
<td></td>
<td>0.25</td>
</tr>
<tr>
<td>smDefaultbendradiustype</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>smDefaultbendradiusvalue</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>smDefaultbendreliefwidthtypetype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultbendreliefwidthvalue</td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>smDefaultcornerreliefdiametervalue</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>smDefaultflangesplitextensiontypetype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultflangesplitextensionvalue</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>smDefaultflangesplitgaptype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultflangesplitgapvalue</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>smDefaultformfeatureunfoldmode</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>smDefaultHemGapType</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>smDefaultHemGapValue</td>
<td></td>
<td>0.005</td>
</tr>
<tr>
<td>smDefaultjunctionalignmenttorelief</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultjunctiongaptypetype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultjunctiongapvalue</td>
<td></td>
<td>0.001</td>
</tr>
<tr>
<td>smDefaultkfactor</td>
<td></td>
<td>0.27324</td>
</tr>
<tr>
<td>smDefaultLoftedBendNumberSamples</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>smDefaultreliefextensiontypetype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultreliefextensionvalue</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>smDefaultribfilletradiustype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultribfilletradiusvalue</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>smDefaultribprofileradiustype</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultribprofileradiusvalue</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultribroundradiustype</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>smDefaultribroundradiusvalue</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>smDefaultsharpbendradiuslimitratio</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>smDefaultTabChamferDistanceType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabChamferDistanceValue</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>smDefaultTabClearanceType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabClearanceValue</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>smDefaultTabDistanceType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabDistanceValue</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>smDefaultTabEdgeType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabFilletRadiusType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabFilletRadiusValue</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabHeightType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabHeightValue</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>smDefaultTabLengthType</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>smDefaultTabLengthValue</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>smDefaultTabSlotNumber</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>smDefaultthickness</td>
<td></td>
<td>0.078740157480315</td>
</tr>
</tbody>
</table>

APPENDIX A  Concise Summary of System Variables and Settings
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>smexportosmapproximationaccuracy</td>
<td></td>
<td>&quot;0.000393701&quot;</td>
</tr>
<tr>
<td>smexportosminimaledgelength</td>
<td></td>
<td>&quot;0.001968505&quot;</td>
</tr>
<tr>
<td>smformfeaturesdowncolor</td>
<td></td>
<td>&quot;6&quot;</td>
</tr>
<tr>
<td>smformfeaturesdownlayerlinetype</td>
<td></td>
<td>&quot;continuous&quot;</td>
</tr>
<tr>
<td>smformfeaturesdownlayerlinewidth</td>
<td></td>
<td>&quot;-3&quot;</td>
</tr>
<tr>
<td>smformfeaturesupcolor</td>
<td></td>
<td>&quot;6&quot;</td>
</tr>
<tr>
<td>smformfeaturesuplayerlinetype</td>
<td></td>
<td>&quot;continuous&quot;</td>
</tr>
<tr>
<td>smformfeaturesuplayerlinewidth</td>
<td></td>
<td>&quot;-3&quot;</td>
</tr>
<tr>
<td>smjunctioncreatehealcoincident</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smoverallannotationslayercolor</td>
<td></td>
<td>&quot;3&quot;</td>
</tr>
<tr>
<td>smoverallannotationslayerlinetype</td>
<td></td>
<td>&quot;continuous&quot;</td>
</tr>
<tr>
<td>smoverallannotationslayerlinewidth</td>
<td></td>
<td>&quot;-3&quot;</td>
</tr>
<tr>
<td>smparametrizeholesparametrization</td>
<td></td>
<td>&quot;3&quot;</td>
</tr>
<tr>
<td>smrepairloftedbendmerge</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smsmartfeatures</td>
<td></td>
<td>&quot;3&quot;</td>
</tr>
<tr>
<td>smSplitAmbiguousInput</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smsgreenpointbendtojunction</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>smsgreenpointhealcoincident</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>smsplitorthogonalbendsplit</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>SMTARGETCAM</td>
<td></td>
<td>&quot;==&quot;</td>
</tr>
<tr>
<td>SNAPANG</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SNAPBASE</td>
<td></td>
<td>0&quot;,0&quot;</td>
</tr>
<tr>
<td>SNAPISOPAIR</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SnapMarkerColor</td>
<td></td>
<td>&quot;20&quot;</td>
</tr>
<tr>
<td>SnapMarkerSize</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SnapMarkerThickness</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>SNAPMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SNAPSTYL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SNAPTYPE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SNAPUNIT</td>
<td></td>
<td>1/2&quot;,1/2&quot;</td>
</tr>
<tr>
<td>SOLIDCHECK</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SORTENTS</td>
<td></td>
<td>127</td>
</tr>
<tr>
<td>spaAdjustMode</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>spaCheckLevel</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>spaGridAspectRatio</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>spaGridMode</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>spaMaxFacetEdgeLength</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>spaMaxNumGridLines</td>
<td></td>
<td>512</td>
</tr>
<tr>
<td>spaMinUGridLines</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>spaMinVGridLines</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>spaNormalTol</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>spaSurfaceTol</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>spaTriangMode</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>spaUseFacetRES</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SPLFRAME</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SPLINESEGS</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>SPLINETYPE</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SRCHPATH</td>
<td></td>
<td>&quot;C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD\V20x64\en_US\Support; C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Support; C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Fonts; C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Help\en_US&quot;</td>
</tr>
<tr>
<td>SSFOUND</td>
<td></td>
<td>&quot;==&quot;</td>
</tr>
<tr>
<td>SSLOCATE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SSMAUTOOPEN</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-----------------</td>
</tr>
<tr>
<td>SSMPOLLTIME</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>SSMSHEETSTATUS</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>SSMSTATE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>StackPanelType</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>Stairwidth</td>
<td></td>
<td>&quot;1000&quot;</td>
</tr>
<tr>
<td>StampFontSize</td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td>StampFontStyle</td>
<td></td>
<td>&quot;Arial&quot;</td>
</tr>
<tr>
<td>StampFooter</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>StampFooterOffsetX</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>StampFooterOffsetY</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>StampHeader</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>StampHeaderOffsetX</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>StampHeaderOffsetY</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>StampUnits</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Statusbar</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>STARTUP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>STEPSIZE</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>StlPositiveQuadrant</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>STEPSPERSEC</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>StructureTreeConfig</td>
<td></td>
<td>&quot;mechanical.cst&quot;</td>
</tr>
<tr>
<td>SURFTAB1</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SURFTAB2</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SURFTYPE</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SURFU</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SURFV</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SvgBlendedGradients</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SvgDefaultImageExtension</td>
<td></td>
<td>&quot;.png&quot;</td>
</tr>
<tr>
<td>SvgGenericFontFamily</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SvgHiddenLineRemoving</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SvgImageBase</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SvgImageUrl</td>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SvgLineWidthScale</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SvgOutputHeight</td>
<td></td>
<td>768</td>
</tr>
<tr>
<td>SvgOutputWidth</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>SvgPrecision</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>SvgScaleFactor</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SYSCODEPAGE</td>
<td>r/o</td>
<td>&quot;ANSI_1252&quot;</td>
</tr>
</tbody>
</table>

**T Variables**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TabControlHeight</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>TABMODE</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>TabsFixedWidth</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>TangentLengthType</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>TangentLengthValue</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>TARGET</td>
<td></td>
<td>&quot;0&quot;, &quot;0&quot;, &quot;0&quot;</td>
</tr>
<tr>
<td>TDCREATE</td>
<td>r/o</td>
<td>2456335.5399919</td>
</tr>
<tr>
<td>TDINDWG</td>
<td>r/o</td>
<td>0.121354456</td>
</tr>
<tr>
<td>TDUCREATE</td>
<td>r/o</td>
<td>2456335.8733252</td>
</tr>
<tr>
<td>TDUPDATE</td>
<td>r/o</td>
<td>2456335.5399919</td>
</tr>
<tr>
<td>TDUSRTIMER</td>
<td>r/o</td>
<td>0.121354456</td>
</tr>
<tr>
<td>TDUUPDATE</td>
<td>r/o</td>
<td>2456335.8733252</td>
</tr>
<tr>
<td>TeeTangentLengthType</td>
<td></td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>TeeTangentLengthValue</td>
<td></td>
<td>&quot;0.5&quot;</td>
</tr>
<tr>
<td>TemplatePath</td>
<td></td>
<td>&quot;C:\Users\userid\AppData\Local\Bricsys\BricsCAD\20x64\en_US\Templates&quot;</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>TEMPPREFIX</td>
<td>--</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>TestFlags</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TestGsFlags</td>
<td>0</td>
<td>TestFlags</td>
</tr>
<tr>
<td>TEXTANGLE</td>
<td>0</td>
<td>TEXTANGLE</td>
</tr>
<tr>
<td>TEXTED</td>
<td>2</td>
<td>TEXTEDITMODE</td>
</tr>
<tr>
<td>TEXTEDITMODE</td>
<td>0</td>
<td>TEXTED</td>
</tr>
<tr>
<td>TEXTED</td>
<td>2</td>
<td>TEXTEDITMODE</td>
</tr>
<tr>
<td>TEXTFILL</td>
<td>1</td>
<td>TEXTED</td>
</tr>
<tr>
<td>TEXTQLTY</td>
<td>50</td>
<td>TEXTED</td>
</tr>
<tr>
<td>TEXTSIZE</td>
<td>0.2</td>
<td>TEXTED</td>
</tr>
<tr>
<td>TEXTSTYLE</td>
<td>&quot;Standard&quot;</td>
<td>TEXTED</td>
</tr>
<tr>
<td>TextureMapPath</td>
<td>&quot;C:\Program Files (x86)\Bricsys\BricsCAD V20x64\Textures\1&quot;</td>
<td>TEXTED</td>
</tr>
<tr>
<td>THICKNESS</td>
<td>0</td>
<td>THICKNESS</td>
</tr>
<tr>
<td>ThreadDisplay</td>
<td>&quot;0&quot;</td>
<td>THICKNESS</td>
</tr>
<tr>
<td>THUMBSIZE</td>
<td>1</td>
<td>THREADMODE</td>
</tr>
<tr>
<td>TILEMODE</td>
<td>1</td>
<td>THUMBSIZE</td>
</tr>
<tr>
<td>TILEDROPSYNCH</td>
<td>1</td>
<td>THREADMODE</td>
</tr>
<tr>
<td>TIMEZONE</td>
<td>-8000</td>
<td>ToolbarMargin</td>
</tr>
<tr>
<td>ToolbarMargin</td>
<td>r/o 0</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>ToolbuttonSize</td>
<td>r/o 0</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TooliconPadding</td>
<td>r/o 0</td>
<td>Tips</td>
</tr>
<tr>
<td>Tips</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>ToolPalettes</td>
<td>&quot;C:\Users\userid\AppData\Roaming\Bricsys\BricsCAD V20x64\en_US\Support\ToolPalettes&quot;</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>ToolTips</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TPSTATE</td>
<td>r/o 0</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TRACEWID</td>
<td>0.05</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TRACKPATH</td>
<td>0</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TRANSPARENCYDISPLAY</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>Trayicons</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TrayNotify</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TrayTimeout</td>
<td>0</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TreadLength</td>
<td>&quot;290&quot;</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TREEDEPTH</td>
<td>3020</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TREEMAX</td>
<td>10000000</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TRIMMODE</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TrustedPaths</td>
<td>r/o &quot;&quot;</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TSPACEFAC</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TSPACETYPE</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TSTACKALIGN</td>
<td>1</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TSTACKSIZE</td>
<td>70</td>
<td>ToolbuttonSize</td>
</tr>
<tr>
<td>TTFASTEXT</td>
<td>3</td>
<td>ToolbuttonSize</td>
</tr>
</tbody>
</table>

**U Variables**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSAXISANG</td>
<td>90</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSBASE</td>
<td>&quot;&quot;</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSDETECT</td>
<td>0</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSFOLLOW</td>
<td>0</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSICON</td>
<td>3</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSICONPOS</td>
<td>0</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSNAME</td>
<td>r/o &quot;&quot;</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSORG</td>
<td>r/o 0&quot;&quot;,0&quot;,0&quot;</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>UCSORTHO</td>
<td>1</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>UCSVIEW</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>UCSVP</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>UCSXDIR</td>
<td>r/o</td>
<td>1&quot;,0&quot;,0&quot;</td>
</tr>
<tr>
<td>UCSYDIR</td>
<td>r/o</td>
<td>0&quot;,1&quot;,0&quot;</td>
</tr>
<tr>
<td>UNDOCTL</td>
<td>r/o</td>
<td>5</td>
</tr>
<tr>
<td>UNDOMARKS</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>UNITMODE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>UseBIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UseCommunicator</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>UseMechanical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERI1 thru USERI5</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>USERR1 thru USERR5</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>USERS1 thru USERS5</td>
<td>r/o</td>
<td>--</td>
</tr>
<tr>
<td>UseSheetMetal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UseStandardOpenFileDialog</td>
<td>r/o</td>
<td>0</td>
</tr>
</tbody>
</table>

### V Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VbaMacros</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>VENDORNAME</td>
<td>r/o</td>
<td>&quot;Bricsys&quot;</td>
</tr>
<tr>
<td>VerboseBimSectionUpdate</td>
<td>r/o</td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>_VERNUM</td>
<td>r/o</td>
<td>&quot;19.1.06 (UNICODE)&quot;</td>
</tr>
<tr>
<td>VersionCustomizableFiles</td>
<td>r/o</td>
<td>&quot;344&quot;</td>
</tr>
<tr>
<td>VIEWCTR</td>
<td>r/o</td>
<td>10 7/16&quot;,4 1/2&quot;,0&quot;</td>
</tr>
<tr>
<td>VIEWDIR</td>
<td>r/o</td>
<td>0&quot;,0&quot;,1&quot;</td>
</tr>
<tr>
<td>VIEWMODE</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>VIEWSIZE</td>
<td>r/o</td>
<td>297</td>
</tr>
<tr>
<td>VIEWTWIST</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>VIEWUPDATEAUTO</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>VISRETAIN</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>VOLUMEPREC</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>VOLUMEUNITS</td>
<td></td>
<td>&quot;in ft mi µm mm cm m km&quot;</td>
</tr>
<tr>
<td>VPROTATEASSOC</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>VSMAX</td>
<td>r/o</td>
<td>-1.0000E+20,-1.0000E+20,-1.0000E+20</td>
</tr>
<tr>
<td>VpMaximizedState</td>
<td>r/o</td>
<td>0</td>
</tr>
<tr>
<td>VSMIN</td>
<td>r/o</td>
<td>1.0000E+20,1.0000E+20,1.0000E+20</td>
</tr>
<tr>
<td>VTDURATION</td>
<td></td>
<td>750</td>
</tr>
<tr>
<td>VTENABLE</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>VTFPS</td>
<td></td>
<td>7</td>
</tr>
</tbody>
</table>

### W Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Read-only</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wallwidth</td>
<td></td>
<td>&quot;250&quot;</td>
</tr>
<tr>
<td>WarningMessages</td>
<td></td>
<td>65535</td>
</tr>
<tr>
<td>WHIPARC</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>WHIPTHREAD</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>WINDOWWAREACOLOR</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>WIPEOUTFRAME</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>WMFBKGDND</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>WMFFOREGND</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>WNDLMAIN</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>WNDLSCRL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>WNDLSTAT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>WNDLtabs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WNDLTEXT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Read-only</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>WNDPMAIN</td>
<td></td>
<td>0&quot;,0&quot;</td>
</tr>
<tr>
<td>WNDPTEXT</td>
<td></td>
<td>3'-4&quot;,3'-4&quot;</td>
</tr>
<tr>
<td>WNDSMAIN</td>
<td></td>
<td>101'-2&quot;,66'-11&quot;</td>
</tr>
<tr>
<td>WNDSTEXT</td>
<td></td>
<td>118'-4&quot;,86&quot;</td>
</tr>
</tbody>
</table>

**WorkspaceSecurity**

<table>
<thead>
<tr>
<th>Variable</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WORLDUCS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>WORLDVIEW</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>WRITESTAT</td>
<td>r/o</td>
<td>1</td>
</tr>
<tr>
<td>WSAUTOSAVE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>WSCURRENT</td>
<td></td>
<td>“2D Drafting”</td>
</tr>
</tbody>
</table>

**X Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XCLIPFRAME</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>XDwgFadeCtl</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>XEDIT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>XFADECTL</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>XLOADCTL</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>XLOADPATH</td>
<td></td>
<td>“C:\Users\userid\Documents\”</td>
</tr>
<tr>
<td>XNotifyTime</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>XREFCTL</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>XRefNotify</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>XREFOVERRIDE</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

**Z Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOOMFACTOR</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>ZOOMWHEEL</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

**# Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3DCOMPAREMODE</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3DOSMODE</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>3dSnapMarkerColor</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>
BRICYS UPDATES BRICSCAD SEVERAL TIMES A YEAR, WITH A MAJOR UPDATE EACH FALL.

This appendix lists BricsCAD’s new and changed functions in V20, and is compiled from version 20.1.05. 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:

- New Entities
- Command Name Changes
- User Interface, File Operations, and Printing
- 2D Drawing and Editing
- Blocks, Tables, and Data Extraction
- Dimensioning
- 3D
- Point Clouds
- BIM
- Civil
- Mechanical
- Sheet Metal
- Communicator
- Customization and Security
BricsCAD V20 installs and runs independently from previous BricsCAD versions.

Shape is no longer supplied in a separate program, and is installed with BricsCAD V20. It can be started from the new Launcher dialog box.

**New Entities**

BricsCAD V20 displays surfaces containing t-spline sub-surfaces, NURBS surfaces where control points termination in the shape of a T.

New entities and commands for doing civil engineering work in BricsCAD V20 Platinum. The new entities are:

- TIN surfaces
- Grading entities
- Alignment entities

The commands associated with civil engineering are Alignment, AlignmentEdit, AlignmentView, Grading, Tin, TinEdit, TinModify, TinMerge, TinVolume, and TinExtract.

**COMMAND NAME CHANGES**

**Drag** is the new name for the **BimDrag** command.

**bmInsert** and **DesignTable** commands are now available at the BricsCAD Pro license level.

**Parameters** panel is renamed **Parameters and Constraints** panel as it supports non-parametric 3D constraints.

**Plot** command now displays the Print command’s dialog box, but reverts to command line prompts in unattended mode.

**WNDLSTAT** variable is renamed **STATUSBAR**, and the **WNDLTABS** variable is renamed **LAYOUTTAB**.

**USER INTERFACE, FILE OPERATIONS, AND PRINTING**

BricsCAD V20 starts with a new Launcher dialog and then displays a new Start page; these replace the GetStarted dialog box.

Launcher displays workspaces to choose from.

Start page lists recently opened drawings, and starts new ones from scratch or from templates.

BricsCAD **GotoStart** command opens the new start tab.

**ColorTheme** variables toggles between the traditional light theme (1) and the new dark theme (0; default).
**CommandLine** now allows a foreground color of solid black and a custom background color. When AutoComplete is in use, pressing the backspace key removes one typed character each time it is pressed.

**DgnImport** command now imports 3D polylines from .DGN files.

**DgnImportOptions** opens the Settings dialog box at the DgnImport section.

**DstConvert** converts sheetset DST files to XML format.

**DynPiCoords** variable displays dynamic coordinate input with relative (0) or absolute (1) distances and angles.

**UcsDetect** variable is expanded so that dynamic UCS supports entities other than faces of 3D solids:

- 3 — enable for 3D faces and 2D entities
- 2 — enable for 2D entities only
- 1 — enable for 3D faces only
- 0 — disable
- negative — disable, but remember the previous setting

**DUCS status bar control** toggles what dynamic UCS detects:

- Faces of 3D solids (default)
- Other entity types: point, line, polyline, 2D polyline, 3D polyline, ray, xline, arc, circle, ellipse, spline, text, mtext, solid, 3Dface, trace, block insert, viewport, mline, leader, mleader, hatch, helix, camera, light, section, shape, PDF underlay, and image.

**Explorer** command alerts you when a referenced text style or dimension style is removed

Grip selection process is more efficient. Grips, pickbox, and other glyphs are now scaled automatically for high resolution (4K) displays.

**HideSystemPrinters** variable allows users to hide Windows system printers in the Print dialog, keeping just the .pc3 printer configurations.

**ImageAttach** command now loads compressed TIFF files with more than half a billion pixels; this limitation had not affected images in Big-TIFF format which use 64 bit offsets.

IME (input method editor- composition window no longer closes unexpectedly when the mouse is moved over the drawing view.

**Layer, PDF, Image, and XRef** commands now bring up modeless, dockable panels instead of the Drawing Explorer. Use the ExpLayers, ExpPdf, ExpImage And ExpXrefs commands bring up the Drawing Explorer for these items.

**LookFrom** widget now remains active during commands, and no longer disappears when entering clean screen mode. But it is no longer active while other application windows are in the foreground.
Manipulator has an improved initial location of the Manipulator for associative arrays. The MIRROR and SCALE options support the C key (hidden Copy option) to make multiple copies with the Manipulator.

NearestDistance variable now controls the types of dimensions displayed: current values, or individual X, Y and Z components of nearest distances. Scaling of arrowheads displayed during zooming is added.

New and Open commands that launch another instance of BricsCAD with the \t (template) option create new drawings based on appropriate template, instead of opening the .dwt template file itself. Drawings with errors in AcDbAssocAction records can now be opened as corrupted records are ignored.

Ortho is properly reported in the status bar.

PdfImport command displays an alert message when no objects are imported, or when errors occur during the import process. The amount of “grainy dust” in the display of PDF underlays of scanned image is reduced, when the PDF display color is set to match the background color.

PeditExt command (for editing polyline vertices and segments) now works with 2D classic and 3D polylines, in addition to lightweight polylines.

Print command is improved when printing layouts with many viewports. Plot command now displays the print dialog box in the interactive mode, and behaves as -Plot in the unattended mode. When running in unattended or scripted mode, -Plot overwrites files silently when Expert variable is > 1. Users can choose to use multiple folder paths for the PlotCfgPath and PlotStylePath variables.

Properties panel Coordinate pickbutton behavior now makes: dynamic dimensions available.

Polar snap (Snaptype=1) is now supported with the PolarDist, PolarAng, SnapMode, and AutoSnap variables.

Quad now opens correctly on a hot-plugged secondary display. When the quad is turned off, a short-right-click suffices to display the context menu for selected entities. Polyline segments can now be selected as entity filters for quad buttons. The LwPolyline_Edge entity is added to the list displayed in Add Entity Alias dialog box.

Redraw command is improved for drawings with many inserts with attributes located at the drawing origin and when silhouette display is on (DispSilh = 1).

RrRot command improves the calculation of the center of rotation when OribtAutoTarget variable is 0 and the selection point is not on any object or surface.

SaveAs command keeps the old name of a not-yet-saved drawing when the SaveAs operation fails.

Screenshot takes a screen grab of the current space, excluding all UI elements

Shell allows non-ASCII characters in paths.

ShowFulPathInTitle variable displays the full document path in the title of the main BricsCAD
window, when set to \textbf{1}. The title bar displays the name of the lower license level when \textbf{RunAsLevel} is applied to run BricsCAD V20 at a reduced level.

\textbf{Snap} previously gave precedence to the point with higher priority when snap points with different snap priorities (e.g. end point and intersection) were inside the snap aperture box; now the winning snap point is the one closest to the cursor. Previously, the distance to the cursor was computed only in the case where the snap points had equal priority.

\textbf{StatusBar} changes the appearance of the Cursor Coordinate values, depending on whether its enabled or disabled.

\textbf{Structure} command gains the option to deselect entities when hiding them via the panel; . “on” by default.

\textbf{Toolbar} command now shows previously hidden toolbars when it is repositioned. \textbf{Tooltips} variable is now functional and shown in Settings dialog. Toolbars are shown in sub-menues by menu group, and similar named toolbars are further grouped into nested sub-menues.

\textbf{Toolpalettes} command gain support for palette separators.

\textbf{ViewEdit} gains the option to change the selection set of several views, as long as they are the same for all views.

\textbf{ViewHorizontal} rotates the viewpoint to make z=0 (horizontal)

\textbf{VpMax} and \textbf{VpMin} commands maximize and restore current viewport. Paper space viewport(s) in model space can now be edited, copied, moved, and erased. VPLOCK status bar button allows users to see and set the display lock status of the active layout viewport.

\textbf{WndlStat} variable is renamed \textbf{StatusBar}, and the \textbf{WndlTabs} variable is renamed \textbf{LayoutTab}.

\section*{2D Drawing and Editing}

\textbf{Arc} and \textbf{Pline} commands change the direction arcs are drawn to clockwise when you hold down the \textbf{Ctrl} key.

\textbf{CopyGuided} copies entities along guidelines.

\textbf{CopyToLayer} command copies entities to layers.

\textbf{-Gradient} command is the command-line version of the \textbf{Gradient} command’s dialog box.

\textbf{Hatch} command no longer creates undesired associations between hatches and externally adjoining boundaries (e.g. for externally adjoining rectangles). The \textbf{-Hatch} command now fills gradient patterns. \textbf{HatchGripEdit} adds and removes grips from hatches and gradients.

\textbf{-ImageAttach} command has improved text prompts.

\textbf{InsertGuided} and \textbf{-InsertGuided} inserts blocks along guide curves
**Layer** command that works on the command-line now allows the locking and unlocking of xref-dependent layers. **LayerP** command now properly reverts changes made by sequential calls to LayIso.

The Layer Filter Panel now supports Layer Group filters. Entity selection from locked layers is now available for the LayOff and LayFrz commands for any block selection setting.

MPolygon objects now support grips.

**MText** command gains a Paragraph Spacing dialog box. The on-screen mtext editor now offers better support for high resolution displays.

**MLleader** command no longer allows annotative blocks to be used with block content. MLeader command Creating an annotative mleader in a custom UCS adds an MText copy when mtext content is used.

**Number** command gains a new parameter, Tolerance: it considers two solids equal when their centroids are within a given tolerance.

**ParametricBlock** creates a parametric block from entities in the drawing; useful for BEdit.

Project Browser gains the Add Sheet, Add Subset, and Remove Subset options, and the Properties option to Sheet/View/Subset nodes.

Sections can now be updated in the background, enabled by checking the Enable Background Update item in the Project Browser menu. Tree controls are now user-resizable and will always fit in the browser window, and configurable margins are added for viewport placement.

You can now display Sheet/View labels as “number-name”. Also implemented is drag and drop of schedules onto a sheet/layout; you can also drag & drop a Sheet/Subset to another subset. You can extend an existing sheetset from the Project Browser, and there’s a new project setup dialog, too.

**Propagate** command gains support for features other than windows and doors. This command is faster at the propagation of window features, and section movement.

Its Box select behavior now aligns with BricsCAD’s regular box select behavior.

Two types of propagated details can be parameterized: connections between linear solids and connections between planar solids. This parameterization allows users to apply a detail to target configurations with different geometric parameters, such as layer thicknesses, profiles and angles.

**Properties** panel can now use Start offset and End offset properties to edit path arrays.

**QuickDraw** command now draws the projected grid on top of a slab when the user drags the blue quickdraw cursor from the outside (of a building) inwards. It also now copies stairs across multiple floors automatically.
Style command no longer applies the vertical flag for fonts that do not support vertical orientation (all TTF fonts and some SHX fonts).

Text command now displays text using its own color, instead of the color of the current layer, during editing.

UndoEnt undoes property changes to selected entities.

**BLOCKS, TABLES, AND DATA EXTRACTION**

2D constraints now can be applied to block insertion points, and to entities inside blocks. It is now possible to add flip states to a parametric block by defining flip lines.

AttDef command adds the Select Property context menu item to select properties from the list. The positioning of multi-line attribute definitions is improved. The fields specified in the Default value of single line attribute definitions are preserved after closing the Define Attribute dialog box.

AttSync command now allows wild-card patterns for block names.

BEdit (block editor) and RefEdit (reference editor) commands now edit blocks with proxy entities, if they are cloneable, transformable, eraseable, and able to change visibility.

Blockify command now gives users the option to specify the names of the blocks to be created. The input set to the command now supports wipeout entities. Block transformation matrices are now simplified after Blockify executes.

DataExtraction command adds the 'table column total' option. Tables generated from data extractions have improved formatting. Data extraction now offers block views, which are graphical thumbnails that are inserted into tables containing the results from data extraction operations. Changed dragging initiation keypress on the Properties wizard page to **ctrl+right-click drag**, to support context menus.

DataLink command makes tables with attached data links that are compatible with AutoCAD.

Dynamic blocks now sport grips for lookup parameters.

Table command adds the datalink update tool to the quad. Tables with breaks have improved selection and editing. Improved table highlighting when changing "Enable breaks" from true to false. Now you can edit repeating label rows in a sub-table. The command’s DataLink update now removes or clears rows when a datalink range shrinks.
Dimensioning

Improved the application of dimension text height settings, and text style changes made in the MText editor.

**CenterDisassociate** disassociates center lines and marks from circles and arcs.

**DimLayer, HpLayer, and CenterLayer** variables do not accept xref layers as targets.

**DimTEdit** now behaves properly when dragging an annotative dimension text entity.

**DimGap** variable now allows only positive values now; a zero value is considered as positive so that there will be no frame around dimension text.

General 3D

While hovering the vertex of a 3D solid, holding down the **Ctrl**-key displays the coordinates in the roll-over properties panel.

**QuickDraw** command assigns roof slabs to new floors in buildings.

**Sweep** command gains an interactive version to allows users to sweep profiles along a sequence of curves (not just a single curve), or just along a portion of the curve(s).

Radial grids are now supported in section views.

**POINT CLOUDS**

**PointCloudAttach** command attach point cloud files at the command line.

**PointCloudPreprocess** command pre-processes additional file formats, including ReCAP project and source files to produce highly efficient .vrm (Virtual Reality Model) files that attach directly to drawings. You specify the units for the point cloud and a progress indicator shows the status as the pre-processor runs. **PointCloudPreprocess** is the command-line version.

**PointCloudCrop** crops the extents of the current point cloud, while **PointCloudUncrop** removes cropping from point clouds.
BIM

(BUILDING INFORMATION MODELING)

AutoRoom and Bimify commands find more rooms with walls of a non-standard shape.

BimDrag command has been renamed to Drag, and is available in BricsCAD Pro, Platinum, BIM and Mechanical.

BCF (OpenBIM Collaboration Format) panel now allows users to add comments and change properties in the BCF panel. It allows users to view and manage model-based issues based on the BCF standard.

BimApplyProfile command applies profiles to flow fittings and flow fitting faces. When "AutomaticConnection" is on, applying a profile to a flow element causes adjacent elements to adapt. Setting “RestoreConnection” now controls the restoration of structural connections after BimApplyProfile command runs.

BimAttachComposition command now supports alignment options, and extends ply slicing to accommodate building elements containing a cylindrical reference face.

BimClassify command gains a new BIM element type called Annotation; it is accompanied by a default template (used in BIMTAG).

BimCurtainWall command now provides connection-type nodes.

BimDecompose command now decomposes composition-based solids into their separate plies.

BimFlip command now works with asymmetric profile-elements, both structural and MEP.

BimFlowConnect command creates tee flow fittings in all T-shaped connections when the 'AutomaticTees' option is switched on. Connecting solids with invalid axes now behaves as expected. The command allows generation of a flow layout from an array of flow terminals and a designated main terminal.

BimGrid command now supports anonymous grid blocks in section views, and offer improved snaps. Previews of a BIM grid, and the resulting grid, now use the current "No Selection" properties.

Bimify command now works on selection sets, and you can now specify a project type (Architectural/Structural/MEP) during Bimify.

BimRoom command now labels invalid rooms with an ‘invalid room’ stamp.

BimSection command now changes the clipping state when you press the Ctrl key. The default of “on”, which corresponds to dynamic clipping; when off, you can position the section by snapping to entity snap points, when entity snap mode is on.

The command now displays all hatch patterns of the same material at the same scale in the same 2D drawing when working with different scales. It now applies hatch patterns to generated elevations, and section indicators for detail sections and interior elevations on plans.
**BimSectionUpdate** command now creates room lines, and its performance was improved for non-associative views.

**BimSetLoadBearingDirection** sets load bearing directions for slabs.

**BimSpatialAllocations** command adds building and story definitions at the command line.

**BimStair** command creates straight stairs as a rectangular parametric array, optionally fixed to a wall. You can attach selectable dimensions at the extents of the stair.

**BimTag** command now allows you to place tags on nested entities manually, and offers a new Composition option. The command now highlights related entities during manual tagging mode.

Mleader styles for BIM objects are now supported.

Pressing the **Ctrl** key during manual tag placement switches between different tags.

The format of the _TagTypeToStyle.csv file has been changed to .xml. New styles can be added, based on the styles used for automatically placed tags.

**BmInsert** command now detects if a selected .rfa file has been previously imported, and creates an instance of the existing component to reduce file size.

BIM components have been updated with correct naming and classifications. Windows and Door components use index colors instead of true colors.

**DisplayAxesForMEP** variable now displays the axes of flow bends, reducers and tees, when switched on.

Profiles of flow fittings are now shown in, and can be changed from the property panel.

**IfcExport** command now supports the IFC4 Standard, and you can select the IFC format (2X3 or 4). It exports block attributes and component parameters, has improved performance when exporting polyface meshes, and exports profiles as parameterized IFC profiles. It also now exports analysis models.

**IfcImport** command supports the IFC4 standard and offers a progress meter shows a more accurate indication of the IFC import process.

Invalid symbols in the names of spatial structure elements are replaced to get valid file names for external references.

Windows and doors on a different storey than their containing walls are now correctly positioned when the project structure is imported as a set of xrefs.

Manipulator preserves connections between flow elements and structural elements during rotation.

**MatchProp** command now offers the option to match BIM properties between BIM objects. The new **BimMatchProp** variable controls its behavior.
Properties of a linear element’s axis (start/end points, delta, length) are shown in the property panel and can be edited.

Ribbon in the BIM workspace was re-designed to contain all of BricsCAD V20’s new BIM features.

Schedules now generate elevation and plan symbols for windows and doors. A dialog box to create Schedule definitions is now available in the Project Browser.

**SectionSettingsSearchPath** variable can be configured to specify alternative locations for files containing section plane settings, section, room and grid callout symbols, tag mappings, story indicators and 2D section result layers.

**Separate** command now retains BIM data as expected.

**SetLevelOfDetail** command displays render materials and composition plies when set to high.

**CIVIL**

**Alignment** creates horizontal, vertical, or 3D alignments on TIN surfaces

**AlignmentEdit** edits horizontal and vertical alignments

**AlignmentView** views alignment along TIN surface

**AlignmentVInitial** creates vertical alignments

**LandXmlExport** exports the drawing in LandXML format

**Grading** interactively modifies TIN surfaces to create grading effects, such as for roads and foundations.

**LandXmlImport** imports LandXML files into the current drawing

**Tin** (short for triangulated irregular network) imports data from TIN files to create land surfaces, and converts Civil 3D surfaces to BricsCAD format.

**TinEdit** adds and removes points, break lines, and boundaries in TIN surfaces.

**TinExtract** creates a mesh or 3D solid between TIN surfaces or between a TIN surface and elevation or vertical offset.

**TinMerge** merges two or more TIN surfaces into a single one.

**TinModify** deforms or smooths TIN surfaces.

**TinVolume** creates a TIN volume surface between a base and TIN surfaces or an elevation.
The 3D constraints gain a new **Path** constraint moves a point along a curve specified by an expression and the curve’s parameters.

Constraints with numeric parameters can now be geometry-driven, which means their values aren’t imposed upon the model; instead, they are derived from the model as other constraints are solved. Geometry-driven constraints can be used as inputs for expressions, including variables, other constraints, associative arrays and component parameters.

The vertices of a 3D solid can now be selected for 3D constraints with the **SelectionModes** variable set to 15.

A 3D constraint icon is now shown when a constrained line, circle or xline is selected.

**bmBom** command now creates BOM tables with new capabilities, including extraction of built-in properties of components; adjusting units for properties; configuring the sorting rules; adding formula columns and footers; and saving and loading of BOM templates.

**bmExplode** command adds the option to automatically create an exploded view for a given assembly.

**bmExternalize** command now forces saving of the current drawing prior to calling this command in batch mode, without asking for file names.

**bmMech** command now allows selection of blocks and external references to be converted into components.

**DesignTable** command improves the diagnostics for design table import from a CSV file. Several diagnostic messages were added, including ‘invalid column names’ and ‘unknown error’.

**dmFrozenGroup** command defines groups of entities that can be frozen by setting the group parameter to a non-zero value. This is equivalent to putting the group on a frozen layer. The group parameter can be also controlled by an expression.

Mechanical Browser now allows group features to be edited via a right-click on the feature’s icon. It also allows description editing for nested local definitions, and the animation range works with smaller values.

**-Parameters** command now supports string values.

**Parameters** panel is renamed ‘Parameters and Constraints’ panel as it supports non-parametric 3D constraints. Unsatisfied constraints are displayed in the color purple. “Show driven”, “Show driving” and “Clear unused variables” options were added to the context menu.
SHEET METAL

(SM = SHEET METAL)

Settings for Sheet Metal are now categorized and grouped by feature type.

**smAssemblyExport** command adds the sheet metal type “Standard part” classification HTML and JSON reports. The “Solid types in reports” setting determines what types of solids will be included in reports, ‘Sheet Metal’ and/or ‘Poor Sheet Metal’.

The new “Report path type” setting determines which file path types will be used in reports: relative (default) or absolute.

The JSON file is generated with a list of reachable components, their solids with unfolding information, and the assembly tree structure (starting at the root).

The HTML report uses relative paths to embedded images.

**smExplode** explodes bends, forms, hem, junctions, and tabs

**smFlangeContour** command automatically creates a flange, linked with an existing SM model via a bend, from a provided, closed contour.

**smHemCreate** command creates different types of hem features.

**smJunctionCreate** command now creates junction features on curved hard edges between two bends (or lofts).

**smSelect** command gains a new option, Flat edge, that suggests the selection of co-planar free edges, similar to the smFlangeEdge and smHemCreate commands.

**smSplit** command automatically finds the face to split, based on underlying geometry for input line or point. In case of ambiguity, the UCS associated with the required face is used. The Propagate option automatically suggests multiple splits of a similar type.

**smTabCreate** command creates a tab connection between two flanges.

**smUnfold** command has a new option to place the unfolded view (with all necessary annotations) into a Paperspace view. Bend table information is added to the attributes layer of the un-folded part’s .dxf file. The unfolding of models with coincident geometry cases is substantially improved.

**ViewBase** and **smUnfold** commands place unfolded views of sheet metal parts in paper space using new options.
COMMUNICATOR

Communicator now supports the import of McNeel’s Rhinoceros (.3DM) files. Importable data types are: B-Reps (Solids, Surfaces, Regions), Curves, Meshes, Text annotations, Blocks and Block references, Materials and Layers.

Customization and Security

BricsCAD V20 is compiled with Visual Studio 2017. C++ extension DLLs (Dynamic Link Libraries) need to be compiled with the same platform toolset in order to be compatible with BricsCAD V20.

BOM (bills of material) status of a component can now be queried via the components API.

RibbonPanelMargin workspace property specifies the margin, in pixels, along the sides of ribbon panels.

Select a workspace on the Workspace tab, and that workspace’s property pane can be expanded by dragging the splitter bar.

LegacyCodeSearch, TrustedPaths, and SecureLoad variables are read-only but whose values are changed by users with administrator privileges.