Inside ARES

for Windows, Mac OS X, Linux, Android, IOS and Web

An introduction to drawing in 2D and 3D with ARES Commander 2019

Eighth Edition

by Ralph Grabowski

Eighth Edition, July 2019

written by Ralph Grabowski

upFront.eZine Publishing, Ltd.
WorldCAD Access

Technical writer	Ralph Grabowski
Technical editor	Volker Prasuhn
Copy editors	Stephen Dunning Herbert Grabowski

© Copyright **2019** by Graebert GmbH
http://www.graebert.com

All rights reserved worldwide.

Distribution Notice

This PDF-file is not for resale.
Distribution is not permitted.

If you intend to buy this book, please visit our online shop at **http://store.graebert.com**

This book is covered by copyright. As the owner of the copyright, Graebert GmbH 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.

Copyright Information

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.

The ARES Logo, ARES and ARES Commander are registered trademarks of Graebert GmbH, Berlin / Germany

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.

Quick Contents

Part V — Appendices

Full Table of Contents

Part I — Introduction to ARES Commander 2019

2. A Quick Tour Through ARES Commander . . .21

Part II — 2D Architectural Drafting

Part III – ARES on Mobile Devices & Web Browsers

Part IV – 2D & 3D Mechanical Design

Part V — Appendices

DOWNLOAD ARES FREE

If you do not have a copy of the software to work through the tutorials in this book, you can download a free 30-day demo version of the ARES Command software for desktop and laptop computers from https://www.graebert.com (after registration).

ARES Touch runs on tablets and smartphone, and you can download the free 30-day Android version from the Google Play store or iOS version from the Apple app store.

ARES Kudo runs inside of Web browsers, and so you access it free through kudo.grabaert.com.

PART I

Introduction to ARES Commander 2019

Notes

CHAPTER 01

Introducing ARES Commander 2019

In This Chapter

- Welcome to ARES Commander 2019!
- Learning about ARES Commander and related software
- Understanding the system requirements
- Seeing the new features of the latest release

Welcome to ARES Commander 2019!

In this book, you learn about one of the most modern CAD software packages available today. The program is the result of a multi-year effort by Graebert Gmbh to write a computer-aided design program based on the latest innovations in software.

The line of CAD software from Graebert works on desktop computers, laptops, tablets, phones, and in Web browsers. It consists of the following editions.

For desktop and laptop computers running Windows, Linux, and MacOS

- **ARES Commander** for day-to-day 2D design work and 3D modeling, with a broad range of programming interfaces
- **ARES Standard** for 2D drafting, with the full set of programming interfaces
- **ARES Mechanical** for the design of 2D mechanical systems
- **ARES Map** for interfacing with GIS databases from ESRI
- **ARES Sketch** for laying out room dimensions and uses, for property tax assessments
- **SiteMaster** for measuring the interiors of buildings
- **SiteMaster Kitchen** for measuring and planning kitchens

For mobile devices running Android and iOS:

- **ARES Touch** for viewing and designing on tablets and smartphones running Android and iOS

For any modern Web browser:

- **ARES Kudo** for running CAD in Web browsers

In addition, ARES OEM is a CAD engine designed for third-party developers to build stand-alone applications on desktop and laptop computers, on mobile devices, and for Web browsers.

The name is pronounced “airies,” and is based on the name of the Greek god of unpredictable warfare. In Greek mythology, Ares is the son of Zeus and Hera, but he is better known through the Roman myths as Mars. The chariot of Ares is drawn by fire-breathing stallions, and this image inspired the flame logo used for the ARES software; the flame can be seen at the top of each page in the book.

Functions new in ARES Commander 2019 are highlighted throughout the book by the **NEW IN 2019** marker.

About ARES Commander

ARES Commander is designed to be compatible with all of existing DWG and DXF files, right up to R2018. The program has its own command structure, yet uses aliases to be compatible with many of the commands and variables found in other CAD programs like AutoCAD and IntelliCAD.

For more about the place of ARES Commander in the world of CAD, see https://www.graebert.com/blog/press-articles/graebert-is-the-second-leading-source-of-dwg-based-drafting-tech-in-the-world-after-autodesk/.

The user interface with its ribbon, palettes, and command bar will probably look immediately familiar to you. The program has the standard 2D drawing and editing commands, like Line and Trim, as well as advanced functions like associative dimensioning and annotative scaling. The program creates and modifies 3D ACIS-based solid drawing objects, and can apply geometric and dimensional constraints on and between entities.

(NEW IN 2019) ARES creates and edit custom blocks using actions and elements inside the Block Editor environment. See chapter 8.

BENEFITS OF USING ARES

Graebert used its three decades of experience in the CAD business to build the program from scratch, employing the most current computer programming technology. The program offers you a number of significant benefits.

Impressive Compatibility

The program uses DWG as its primary file format for storing drawings. It incorporates the Teigha DWG engine from the Open Design Alliance, and so it reads and writes DWG and DXF files from as far back as R12 to as recently as R2018.

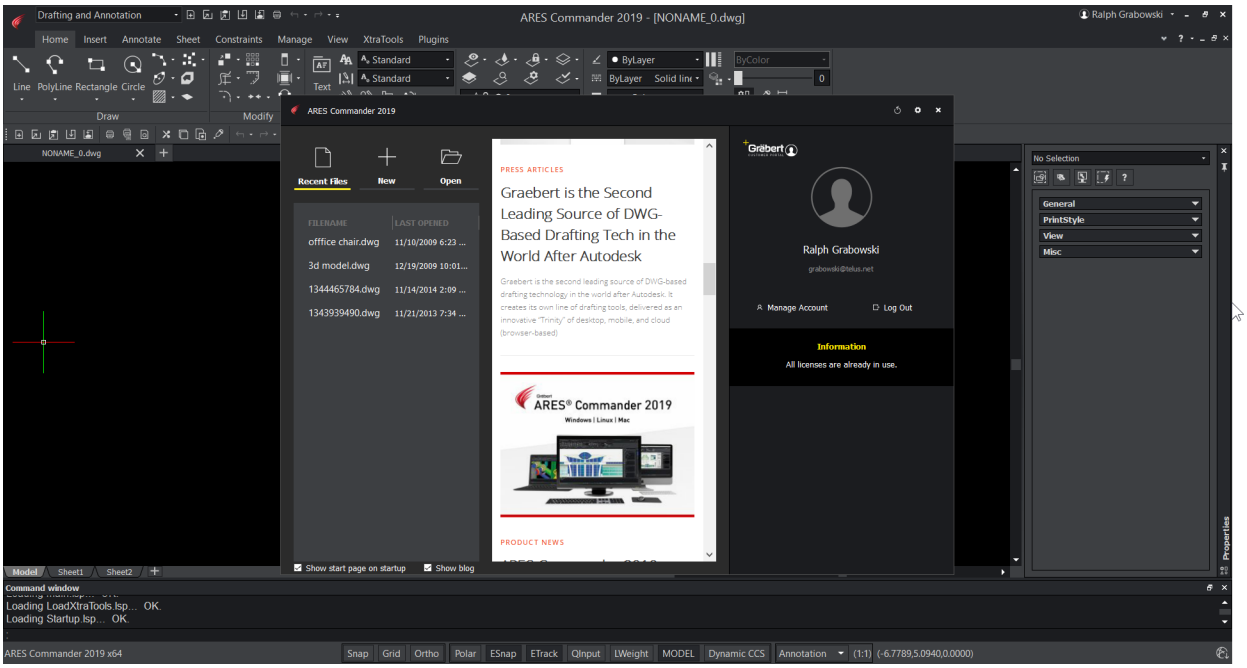
The program exchanges data in a wide range of CAD and graphics file formats, such as ACIS SAT, PDF, DGN, and SVG. When needed, it protects your data investment by recovering DWG and DXF files that may have become damaged.

Familiar User Interface

The user interface is built on Qt technology (the same used in AutoCAD), complete with ribbon, dockable toolbars and palettes, menu bar, right-click menus, aliases, and shortcut keystrokes.

You can change the user interface to suit your needs without any programming; yet programmers can define and manage user interfaces elements through XML resources. You can create and store multiple configurations to adapt the user interface to the needs of specific projects.

The program offers mouse gestures to access commonly-used commands directly on the screen: just hold down the right mouse key while you move the mouse in one of eight directions. In addition, the 3D



ARES Commander 2019 for Windows

mouse from 3Dconnexion is supported for 3D modeling.

See chapter 2 for more on the user interface.

Drafting Tools

The program contains the kind of smart drafting tools you expect to find today's CAD software, such as a multitude of *entity snap* modes, on-screen assistance from *tracking*, and *polar guides*. When drawing in 3D, the program automatically relocates the *coordinate system* to align the construction plane with object faces.

Using *grips*, you can reposition, align, copy, and resize entities. *Filtering* modifies selection sets of entities. The Properties palette lets you manipulate geometry directly, as well as the properties of entities — including their layer names, colors, line styles, and line weights. The Property Painter conveniently copies properties from one entity to others.

See chapter 5 for more on entity snaps.

Dimensions, hatches, and gradient fills all are *associative*, which means they update automatically as you change them. To ensure that drawings conform with industry standards, you create *styles* for dimensions, text, tables, and richlines (multilines). Add notes, tables, leaders, geometric tolerances, and text attributes to annotate drawings. The scale of *annotative* objects can be set to equal the plot scale. Drawing resources are accessed from other computers through the Drawing Resource manager.

See chapter 7 on drawing resources, chapter 9 on text styles, and chapter 10 for more on dimensions.

Collaboration

Drawings can be shared, viewed, and edited on any platform, including tablets and smartphones, Web browsers, and laptop and desktop computers. ARES Commander, Kudo, and Touch support the most popular cloud file storage services, such as Box, Dropbox, Google Drive, and iCloud.

See chapters 11 through 13 on sharing and collaboration.

Drawings and images can be attached to drawings as *references*, which let you share files among workgroups. Referenced drawings can be *clipped* to show just portions of them.

See chapter 18 for more on references.

Mark up drawings with freehand sketches, revision clouds, and mask portions as needed. The Options center manages all drafting and system preferences in a single location. *Profiles* lets you store and restore multiple user settings, while UI profiles let you switch instantly between customized user interfaces.

Flexible Viewing

The program lets you view drawings at different angles in multiple *viewports*, such as with standard orthogonal and isometric views. Three-dimensional models can be shaded differently in each viewport, and rolled about in 3D space interactively.

Display modes can be in wireframe, with hidden lines removed, flat shaded, or Gouraud shaded. Models can be rendered to full-screen or in a separate window. Light sources make the model look more realistic. Entities and layers can be made semi-transparent to emphasize other parts of drawings.

See chapter 16 for more on 3D display modes.

Organized Layouts

Drawings can be organized into multiple tiled *layouts* using tabbed *sheets*. Each tiled viewport can show the model at a different scale and different viewpoint. Drawing borders, title blocks, and equipment lists can be placed on sheets as required.

You can switch between model and paper *space* with just a double-click. Each sheet can determine a different style of printing or plotting. Page layouts save and reuse print settings for printers and plotters.

See chapter 18 for more on viewports and paper space.

Solid Modeling

The program includes a 3D solid modeler based on ACIS, which supports 3D solid primitives, Boolean operations, slicing and intersecting solids. You can edit edges, faces, and bodies of 3D solids. You can extrude or revolve 2D entities into 3D models, sweep 2D entities along paths, and loft between 2D entities. Regions transform areas into 3D solids.

(NEW IN 2019) You can interactively push and pull areas in 2D drawings and the faces of 3D models with the PushPull command.

See chapters 16 and 17 for more on 3D solid modeling.

Parametric Constraints

To make *parametric* drawings, the Windows version of the program includes dimensional *constraints*, 2D geometric constraints, and parameters (expressions) licensed from the Spatial division of Dassault Systemes.

Dimensional constraints let you specify the sizes of objects and distances between them. Geometric constraints act like entity snaps that are sticky, keeping entities parallel or concentric to each other, perfectly horizontal or vertical, and so on.

See chapter 19 for more on constraints.

For Programmers: Application Programming Interfaces

The program provides you with a host of programming languages and interfaces for creating your own functions and plug-ins:

- FxARX
- LISP
- C/C++
- Visual Studio for Applications (VSTA)
- Delphi
- Common Object Model (COM)
- Design Runtime Extension (DRX)
- DraftSight API

To gain some more knowledge about FxARX, see https://www.graebert.com/blog/product-news/new-fxarx-api-enables-compatibility-with-objectarx/.

You can add content from other application using OLE (object linking and embedding) objects in drawings, or through the Drawing Resource palette.

The program has a built-in LISP interpreter, as well as an IDE (integrated development environment) using Microsoft VSTA (visual studio tools for applications), and .NET languages. The program features autoloading of add-ins, conversion, and headers from ADS (AutoCAD development system), C++, a CFx/DRX API including the ARES user interface functions, and the Teigha development platform from the Open Design Alliance, Delphi, COM, and toolbar and menu macros.

ARES OEM is a stand-alone version for third-party developers with an open architecture and a wide range of API features, including ActiveX. Graebert supports partners in building their stand-alone software, and offers a variety of licensing models for Graebert products, such as single licenses, floating licenses, and volume licenses. For more information on using ARES OEM as a CAD engine, contact the Order Desk at oem@graebert.com.

To learn how to migrate ARX code to TX, review the whitepaper available from https://www.graebert.com/blog/white-paper/how-to-migrate-code-from-arx-to-c-tx-with-ares-commanders-api-method-1/.

A VARIETY OF PLATFORMS

Versions of the program are available for five operating systems — Android, iOS, Linux, MacOS, and Windows — as well as running in Web browsers.

The desktop version of the program runs on Windows, Linux, and OS X. It supports Windows 7, 8, and 10 in both 32- and 64-bit versions, and runs on any recent Windows-compatible computer, as well as OS X running on Mac computers and computers operating Linux , such as Suse, Fedora, Ubuntu, Debian, and other distributions.

TIP Throughout this book, references specific to MacOS or Linux are shown in gray text.

The mobile versions of ARES provide you with a smaller set of commands and APIs.

ARES Touch for Mobile Devices

ARES Touch is the version of ARES that runs on Android and iOS tablets and smartphones. Touch is more than a DWG viewer, as it has the largest set of drawing and editing functions of any mobile CAD app. It includes APIs, such as LISP, DCL, and DRX.

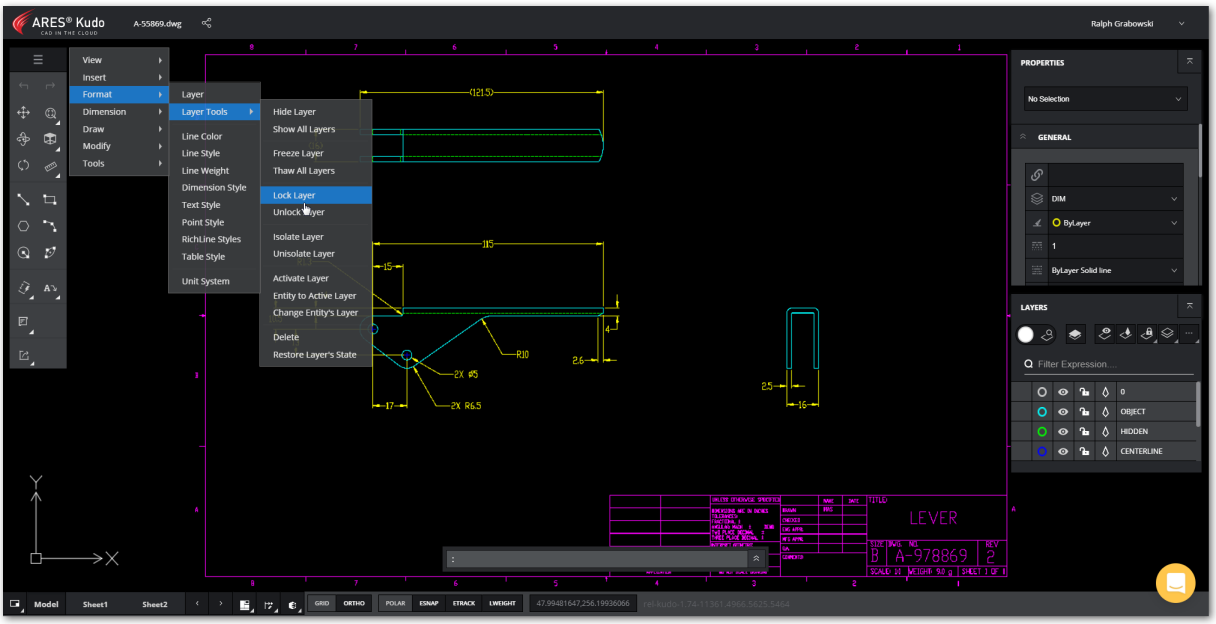
Touch accesses *.dwg* files through Dropbox, Google Drive, email, and other services. Files can be stored locally on the device or automatically synchronized with the cloud. *Synchronization* means that the same version on your desktop computer as on the mobile app.

See chapter 11 for more about ARES Touch.

Left: *ARES Touch on an Android tablet and (**right**) on an Android smartphone*

ARES Kudo for Web Browsers

ARES Kudo is a version of ARES that runs on servers (a.k.a. cloud) and is displayed in Web browsers, meaning it can operate on any hardware that supports a modern Web browser and has an Internet connection. It is meant to be used when a computer does not have desktop ARES Commander installed. It also has many drawing and editing commands, and an API for programmers.



ARES Kudo running on a Web browser

See chapter 12 for more about ARES Kudo.

System Requirements

As a minimum, Linux and Windows computers should have the hardware and software specified below.

Operating Systems	64-bit: Windows 7, 8.1, or 10
	64-bit: Ubuntu, Debian, Fedora, SUSE, or generic Linux
CPU	Intel Core 2 Duo, or AMD Athlon x2 Dual-Core
Free Disk Space	500MB
RAM	2GB RAM
Graphics Board	1280x768-pixels resolution, 3D graphics OpenGL v1.4
Input	Mouse

For better performance, your Linux or Windows computer should run stronger hardware:

CPU	Intel Core i5 or AMD Athlon/Phenon X4
Free disk space	1GB
RAM	8GB
Graphics board	1920x1080 or higher, 3D graphics with OpenGL v3.2
Input	Wheel mouse

Any current model of Macintosh computer runs the MacOS version:

Operating Systems	Mac OS X v10.8 or higher
CPU	Mac with Intel Core 2 Duo processor
RAM	2GB RAM (8GB or more recommended)
Free Disk Space	500MB for installation
Graphics Board	1280x768 or better, with OpenGL v1.4 (OpenGL v3.2 or better recommended)
Input	Mouse

LANGUAGE SUPPORT

The program is available in English, German, and a dozen other languages.

After-sales Service

Graebert continually updates the software. To ensure you are notified of the latest improvements and have full access to its support teams, consider taking out a service subscription.

Perpetual license: one-time payment that includes upgrades and email support for the first year, plus a one-year license of ARES Touch and ARES Kudo.

Maintenance license: renews the subscription for successive terms of one year, with ARES Touch and ARES Kudo licenses included.

Annual licensing: 12 months full license of the program that includes free licenses for ARES Touch for ARES Kudo.

Flex license: add 25% for non-concurrent use of the license on multiple computers on the same network; allows temporary license deactivation for use away from the network. For detailed pricing and contract details, contact Graebert's Order Desk at sales@graebert.com.

Enterprise license: To learn how enterprise licenses work, please review https://www.graebert.com/blog/product-news/graebert-announces-ares-for-enterprise/.

WHAT'S NEW IN ARES COMMANDER 2019

The 2019 edition of ARES Commander offers new functions in the user interface, commands, and system variables. The names of functions and options added in release 2018 SP1 and 2019 are highlighted in red. View the video highlights of this new release at https://www.graebert.com/blog/product-news/ares-commander-2019/.

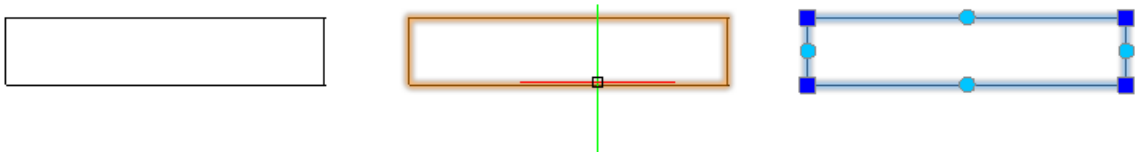
NEW USER INTERFACE ELEMENTS

Enhanced: Icons for 4K Monitors

ARES 2019 adds extra large icons suitable for use with very-high resolution monitors and the light color theme.

New: Rollover Highlighting

When you move the cursor over entities, they are now highlighted with an orange glow to preview the selection set before you click on them. When you select entities, they are now highlighted differently, with a blue glow.



Left to right: *Un-selected entity; rollover highlighting; and selected entity*

Tip When entities overlap, you can select one of them by cycling through them: hold down **Shift**, and then press the **spacebar** until you reach the one you want.

Enhanced: RollView Commands

The **RollView** and **RollViewCenter** commands orbit (or roll) the view in 3D. ARES 2019 adds the following commands for finer control over orbiting:

RollViewContinuous command rotates the view in 3D space with continuous motion about a specified pivot point.

RollViewFree command views models interactively in 3D space without constraining the roll.

RollViewHorizontal command swivels the model view left and right about an imaginary vertical axis.

RollViewVertical command tilts the model view to your model up and down about an imaginary horizontal axis.

RollViewAutoTarget command determines how the target point is detected for these RollView commands

Tip The target looks like a green ball.

Related system variable

OrbitAutoTarget variable specifies how the target point is detected for RollView-related commands:

- **0** Your click point specifies the target point
- **1** (Default) The program centers the target point on visible entities,

Improved: 3D Mouse Support

ARES 2019 detects automatically when a 3D mouse is connected to the computer. Options for ARES are set in the utility software provided by 3dConnexion, and so the ~~3DMouseOptions~~, ~~-3DMouseOptions~~, and ~~3DMouseButtons~~ commands are removed.

NEW DRAWING & EDITING COMMANDS

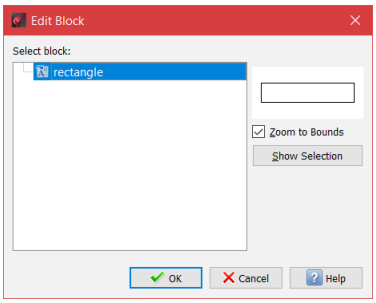
New: Custom Block Editor

A custom block is one that combines one or more blocks into one through visibility states (display one of several blocks), actions, and/or constraints to interactively adjust the size and position of the block with grips — kind of like a "multi-block." You create them in the new Custom Block Editor.

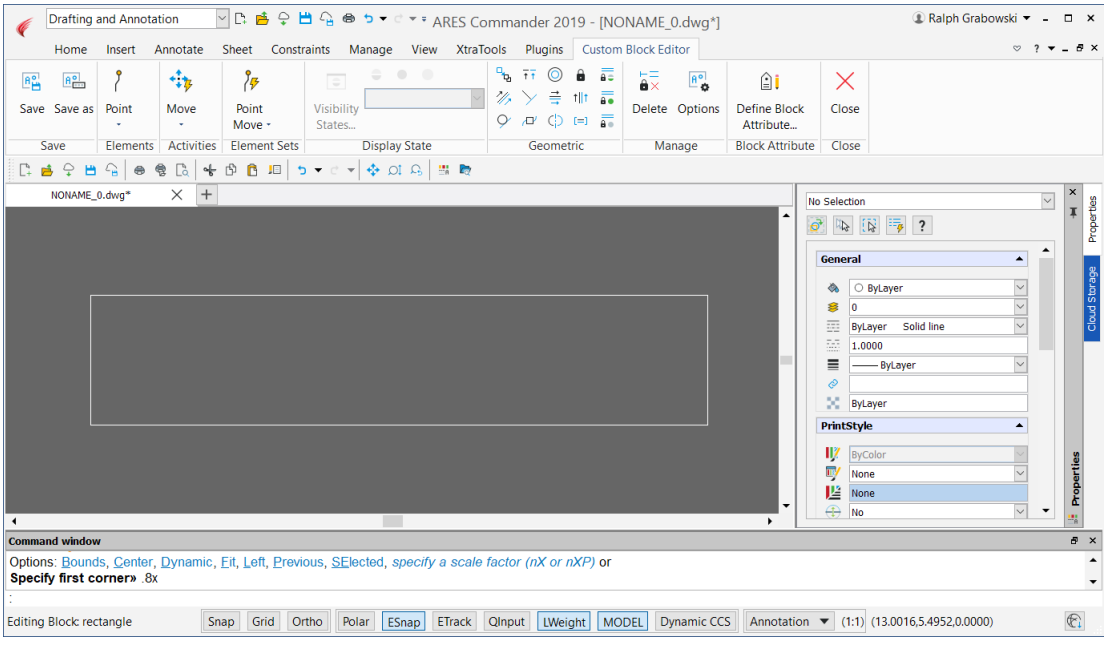
> **Tip** Custom blocks can reduce drafting time by placing one block and then selecting its options afterwards, such as size and rotation angle. Multi-blocks can reduce the number of blocks that you have to manage and maintain by combing, say, several kinds of tree into a single block.

To access this feature:

1. To access the custom block editor, enter the **EditCustomBlock** command, and then choose a single block that already exists in the drawing.
2. In the Edit Block dialog box, confirm your selection, and then click **OK**.



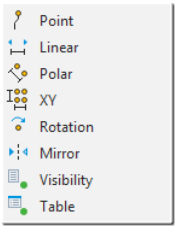
3. The block opens in the Custom Block Editor environment, with the ribbon displaying commands specific to the editor.



> **Tip** The custom block editor can also be used for editing regular blocks in drawings.

All 2D drawing, editing, and view commands work, along with the following commands specific to the editor:

CBElement command adds these elements to custom blocks:



» **Point** element - specifies the x and y coordinates of a grip point on the custom block that lets users stretch or move the block.

Tip To specify whether the use can stretch or move the custom block, you add the Stretch or Move activity to the element with the **CBActivity** command.

» **Linear** element - specifies the distance between two points that lets users move, stretch, scale, or array (by a linear pattern) the block — depending on the activity added to the element.

» **Polar** element - specify a distance and an angle at two points that lets users rotate, move, stretch, scale, or array (by a polar pattern) the block.

» **XY** element - specifies the X and Y distance, so that users can move, stretch, scale, or array (by a rectangular pattern) the block.

» **Rotation** element - specifies an angle that lets users rotate the block by specified amounts.

» **Mirror** element - positions a mirroring axis and a grip point that lets users mirror the block using the mirror grip.

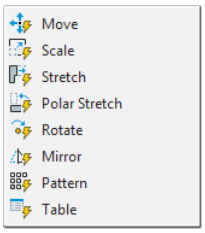
» **Visibility** element - add a visibility element that lets users toggle visibility states on and off.

Tip *Visibility states* are used to display one of several blocks that are combined into a single custom block, such as showing just one of multiple automobile blocks.

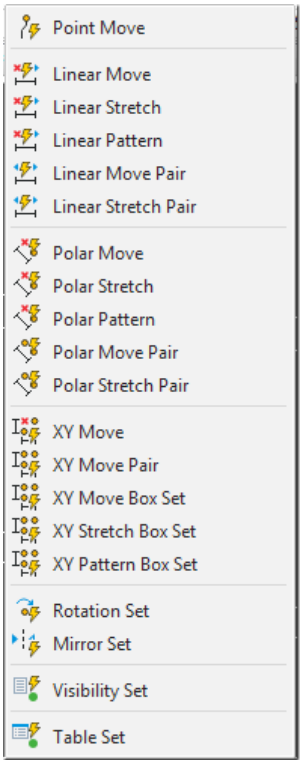
» **Table** element - adds custom properties to the block through the use of a table.

Tip You add properties and values to the table element with the **CBValueTable** command.

CBActivity command adds activities to elements, such as adding Move or Stretch activities to the Point element; the following activities are defined in ARES:

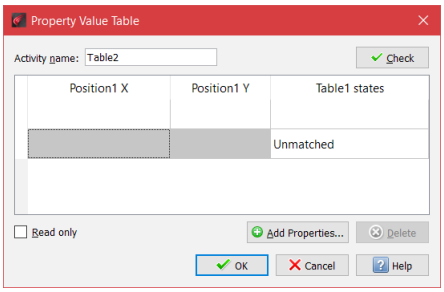


CBElementSet command adds predefined element sets to custom blocks, which consist of a noun (the element) and a verb (the activity), such as Point Move and Polar Pattern. The following element sets are pre-defined in ARES:

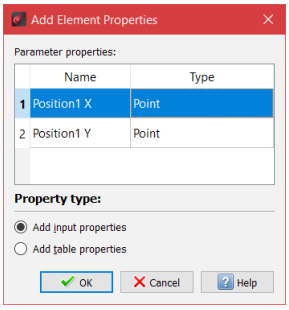


CBActivityTool command applies activities to elements, such as applying the Scale activity to an existing Linear, Polar, or XY element.

CBValueTable command opens the Property Value Table dialog box to add, edit, and delete properties and values:



Click **Add Properties** to add properties and values to the table.



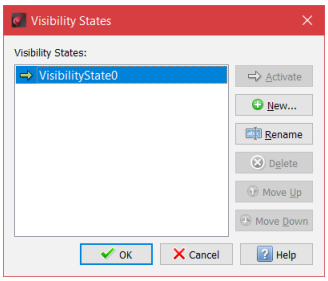
Tip At least one element and activity must exist in the block before you can work with tables.

CBActivityIcon command toggles visibility of activity icons between show, hide, and reset (returns visibility to default).

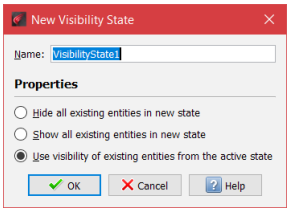
CBHide command sets the visibility state to hidden.

CBShow command sets the visibility state to visible.

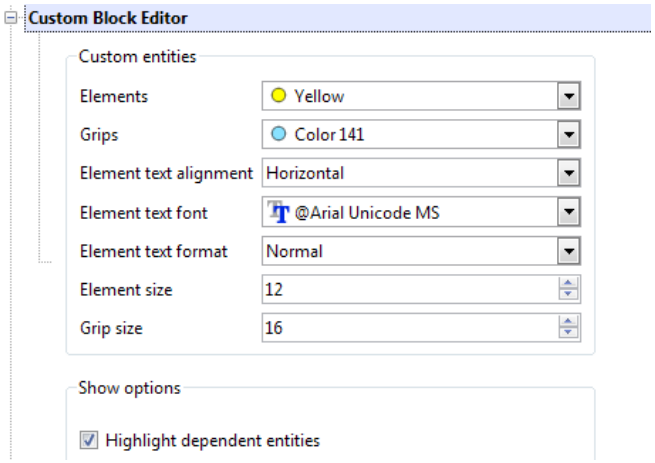
CBVisibility and **-CBVisibility** commands set the default visibility of entities that make up the custom block. For this command to work, you must add at least one visibility state to the block with the **CBElementSet** command.



Add additional states by clicking the **New** button:



CBOptions command displays the Options dialog box at the the Custom Block Editor section.



CBSave command saves the custom block definition.

CBSaveAs command saves the custom block definition by another name.

CBClose command asks if you want to save the changes, and then exits the Custom Block Editor back to the drawing editor

Improved: Trimmed Hatches and Gradients

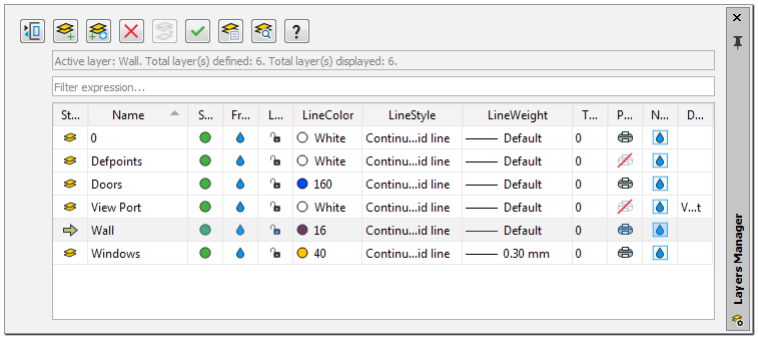
Trimmed hatches and gradients now maintain their associativity when the boundary entity(ies) are changed, even after they are divided into different hatched areas.

> **Tip** By default, divided hatches and divided gradients remain as a single entity. Use the **EditHatch** command to separate them into individual entities. To have separate hatches created, set Create Hatch for each boundary in the Hatch command's dialog box.

Improved: Layer Command

Layer command now opens the new layer palette so that you can control layers and their properties without needing to close it to continue editing and drawing.

A new option (New ViewPort) specifies whether new viewport are initially frozen or thawed



LayerDialog command opens the legacy Layers Manager dialog box.

Related system variable

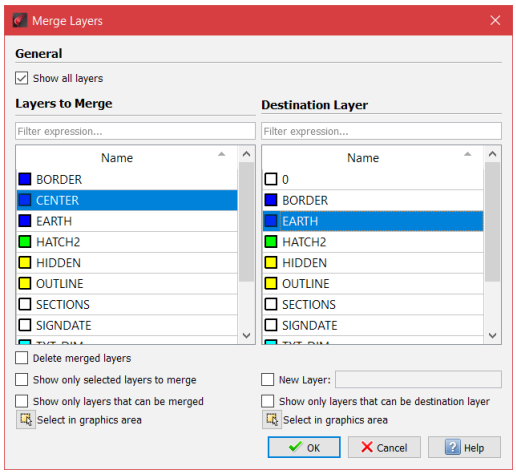
LayerDlgMode variable toggles the interface opened by the Layer command:

0 Layer command displays dialog box, as in ARES 2018 and earlier

1 (Default) Layer command displays the palette

New: Merge Layers

MergeLayer and **-MergeLayer** commands merge one or more layers with a destination Layer (via dialog box and command line).



: **-MergeLayer**

Options: Name or Specify layer to merge by selecting entity» *(Choose an entity on the layer to be merged, or enter its name)*

Options: Name, Remove layer, Undo or Specify layer to merge by selecting entity» *(Enter an option, or else press Enter to continue)*

Options: Name or Specify entity on destination layer» *(Choose an entity on the layer to be the destination, or enter its name)*

Default: No

```
Confirm: Do you want to merge specified layers: "OBJECT" with destination
layer: "DIM". Specify Yes or No» (Enter y or n)
Specified layers: "OBJECT" are merged with destination layer: "DIM" suc-
cessfully. Specified layers are deleted.
```

New: Toggle Hatch Mirroring

MirrorHatch command determines whether hatches and gradient fills are reflected or retain their pattern direction when you apply Mirror and Flip commands:

```
: MirrorHatch
Default: OFf
Options: On, OFf or Specify option»(Enter on or off)
```

» **On** option - hatches and gradients are mirrored (default)

» **OFf** option - not mirrored

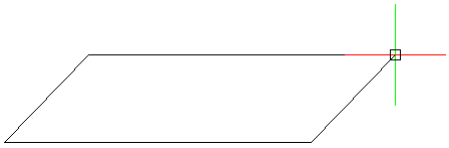
Related system variable

MirrHatch variable toggles whether hatches and gradient fills are mirrored during the Mirror and Flip commands.

Enhanced: Drawing Rectangles

Rectangle command has four new ways to draw rectangles:

```
: Rectangle
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fil-
let, Parallelogram, Thickness, line Width or Specify start corner
```



» **3Corner option** - you specify three points to construct a rectangle by three corner points:

First point - specifies the starting corner of the rectangle
Second point - specifies the width and the angle of the rectangle
Third point - specifies the height of the rectangle

» **3Point center** option - constructs a rectangle by its center point, a midpoint of one size, and a corner point.

» **CEnter** option - constructs the rectangle by its center point and one corner point

» **Parallelogram** option - constructs a rectangle by two points, with the third point specifying the angle of the slanted sides

NEW 3D COMMANDS

New: 3D Entity Snaps

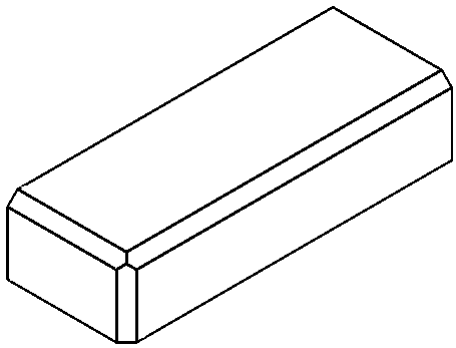
3DEntitySnap and **-3DEntitySnap** commands specify which 3D entity modes to turn on or off (through a dialog box or the command line).

Related system variable

3dOsMode variable toggles the current set of 3D entity snaps.

New: Chamfer Edges

ChamferEdges command cuts edges from the faces of 3D solids.



```
: ChamferEdges
Active chamfer Dist1 = 1.0000, Dist2 = 1.0000
Options: Loop, Face, Distance or Specify edge»
```

» **Specify edge** option - pick an edge to chamfer

» **Loop** option - you pick an edge, and the program selects all edges of the face adjacent to the edge; as there are two faces adjacent to every edge, you also need to specify which face to chamfer

» **Face** option - you pick a face, and then the program selects all edges of the face

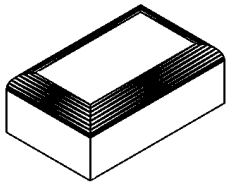
» **Distance** option - the program prompts you to specify the first and second chamfer distances.

Tip This command does not work on round faces, such as on a sphere or the curved part of a cone.

Enhanced: Fillet Edge Options Added

FilletEdges command adds Face, Chain, and Loop options:

```
: FilletEdges
Fillet Radius = 1.0000
Options: Chain, Face, Loop, Radius or Specify edges»
```



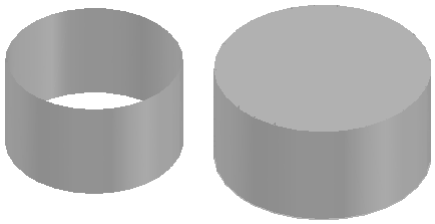
» **Chain** option - you pick an edge, and the program selects all edge connected to that edge

» **Face** option - you pick a face, and the program selects all edges of the face

» **Loop** option - you pick an edge, and the program selects all edges of the face adjacent to the edge; as there are two faces adjacent to every edge, you also need to specify the face.

Enhanced: Extrude Options Added

Extrude command adds Surface and Solid options to determine the type of 3D entity to create.



Left to right: Surface created by extrusion, and a solid

Tip Enter the **SU** or **S** option before you select the entities to extrude.

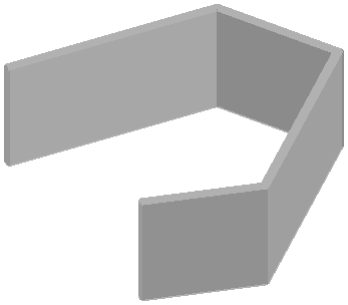
```
: extrude
Current type is SOLID
Options: SUrface, Solid or Specify entities to extrude»
```

» **SUrface** option - creates 3D surfaces from closed entities; open entities always create surfaces, which are open at either end.

» **Solid** option - reverts to creating 3D solids from closed entities (default)

New: Polysolid Walls

Polysolid command draws wall-like 3D solids from points you pick in the drawing, or else converts from 2D entities -- such as lines, polylines, arcs, circles, ellipses, and splines -- into polysolids.



```
: polysolid
Active settings: Height = 4.0000, Width = 0.2500, Justification = Center
Default: Entity
Options: Entity, Height, Width, Justify or Specify start point» (Pick the
starting point, or else enter an option)
Options: Arc, Undo, Enter to exit or Specify next vertex» (Pick the next
point, or else enter an option)
Options: Arc, Undo, Enter to exit or Specify next vertex» (Pick another
point, or else enter an option)
Options: Arc, Close, Undo, Enter to exit or Specify next vertex» (Press
Enter to end the command)
```

- **Specify start point** option - specifies the starting point of the polysolid wall
- **Specify next vertex** option - specifies the next point; the program mitres vertices automatically
- **Entity** option - converts an existing 2D entity into a polysolid entity
- **Height** option - specifies the height of the polysolid, such as 8' for eight-foot tall walls
- **Width** option - specifies the width, such as 4" for four-inch wide walls
- **Justify** option - determines which side of the pick points the polysolid is drawn: left, right, or centered (default)
- **Arc** option - switches to drawing the polysolid segment as an arc
- **Close** option - draws one more segment to join the starting point with the current point
- **Undo** option - undraws the last segment
- **Enter to exit** option - exits the command

Related system variables

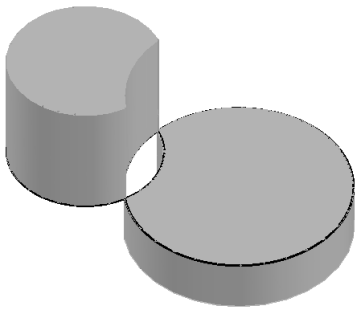
PSolHeight variable sets the default height for 3D solids constructed by the PolySolid command:

```
: psolheight
Default: 4.0000
Enter new value for PSOLHEIGHT» (Enter a new value)
```

PSolWidth variable sets the default width for the 3D solids constructed by the PolySolid comma.

New: Interactive Pushing and Pulling

PushPull command interactively extrudes areas on 3D solids.



```
: pushpull
Specify entity or internal point» (Pick a point)
Options: Multiple or Specify height» (Enter an option)
```

- **Specify entity** option- select a closed 2D entity or face on an existing 3D solid
- **Internal point** option- pick inside a closed area created by 2D entities and/or faces on an existing 3D solid
- **Multiple** option - prompts you to pick additional internal points to extrude
- **Specify height** option - move the cursor to show the height, or else enter a value for the height of the extrusion; enter a negative value for downward extrusions

NEW CLOUD COMMANDS

New: Commenting System

Comments command opens the Comments palette so that you and other users comment on drawings, using the desktop, tablet, and browser versions of ARES.

> **Tip** This command works after (1) when you are signed into your Graebert account and (2) the drawing is stored online in cloud storage with the **SaveToCloudStorage** command.

CommentsHide command closes the palette.

NEW FILE COMMANDS

ARES Commander Release 2019 reads and writes DWG and DXF files up to and including R2018, which is the latest version of this format.

New: Digital Signatures

Digital signatures ensure that the drawing has not been modified by unauthorized persons.

> **Tip** These commands work only with valid digital IDs, which are an optional extra, and may well not exist on your computer.

SignFile command adds a digital signature to the current drawing file.

VerifySignature command checks the digital signature of a drawing for authentication.

RemoveSignature command removes an existing digital signature from the current drawing or from file names that you specify.

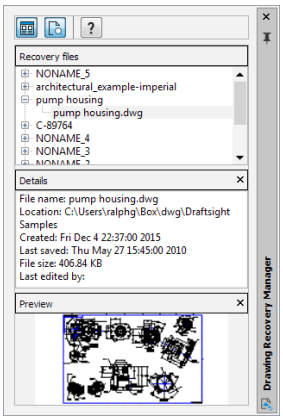
Related system variable

SigWarn variable controls whether the Digital Signature Verification dialog box appears each time you attach external drawings with valid digital signatures.

- 0 Dialog box does not appear when digitally-signed xrefs are attached
- 1 (Default) Dialog box appears

Improved: Drawing Recovery Manager

DrawingRecovery command displays the Drawing Recovery Manager palette to let you recover drawings after unexpected system failures in a more interactive manner than with the command-line **Recover** command. To recover a drawing, double-click its name.



HideDrawingRecovery command hides the Drawing Recovery Manager palette.

> **Tip** The program can stop unexpectedly from a power failure, hardware issue, or operating system problem, preventing it from saving your work. This command helps you recover the last saved version.

Related system variables

RecoveryMode variable determines whether drawing recovery information is recorded when the system terminates unexpectedly:

ISaveBak variable toggles whether backup (*.bak*) files are created each time you save the drawing.

ISavePercent variable specifies when the program performs a full save; partial saves are faster than full saves.

New: Import STL files

ImportSTL command imports stereolithography files (.stl) and then inserts them into the current drawing as a 3D solid.

> **Tip** To export 3D models in STL format, use the **ExportSTL** command. The STL format is commonly used to print 3D models, and exchange models with other programs.

Coming in R2019 SP1

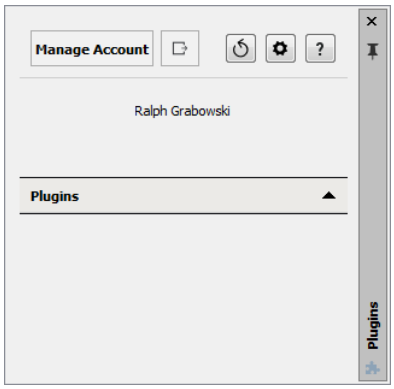
New: Revit and IFC files as underlays

New: Export drawings as RFA (family or blocks) files

NEW API OPTIONS

Enhanced: Plugins Palette

Plugins command opens the Plugins Manager palette to manage application plug-ins provided by third-party vendors. It replaces the Plugins dialog box from ARES 2018 and earlier.



At time of writing, Graebert has two plug-ins available:

» ~~Pointcloud plug-in from Undet~~

» ~~Render plug-in from Graebert~~

HidePlugins command hides the Plugins palette.

Also new in APIs: SDK with samples for plugin developers, installed plugins with account options, and managing licenses via customer portal;

SUMMARY OF NEW SYSTEM VARIABLES

3dOsMode variable toggles the current set of 3D entity snaps:

LayerDlgMode variable toggles the interface opened by the Layer command.

ISaveBak variable toggles whether backup (*.bak*) files are created each time you save the drawing:

ISavePercent variable specifies when the program performs a full save; partial saves are faster than full saves:

MirrHatch variable toggles whether hatches and gradient fills are mirrored during the Mirror and Flip commands:

OrbitAutoTarget variable specifies how the target point is detected for RollView commands:

PSolHeight variable sets the default height for 3D solids constructed by the PolySolid command:

PSolWidth variable sets the default width for the 3D solids constructed by the PolySolid comma:

RecoveryMode variable determines whether drawing recovery information is recorded when the system terminates unexpectedly:

SigWarn variable controls whether the Digital Signature Verification dialog box appears each time you attach external drawings with valid digital signatures:

CHAPTER 02

A Quick Tour Through ARES Commander

In This Chapter

- Starting the program
- Becoming familiar with the user interface
- Drawing lines
- Reversing mistakes
- Accessing online help

This chapter takes you on a whirlwind tour of ARES Commander, where you learn how to start the program, take a look at its user interface, and then get your feet wet by placing a few lines in a new drawing.

To work through the tutorials in this book, you can download a free 30-day demo versions of the software from https://www.graebert.com (after registration).

Starting ARES Commander

If the program is not yet set up on your computer, install it first. Before you use it for the first time, you must register with Graebert Gmbh; see Appendix G. The program runs on any recent computer, as well as many older ones with slower specifications. Your computer must be outfitted with one of the following operating systems.

For desktop computers, laptops, and 2-in1 tablet PCs

» Windows 7, 8 or 10

For desktop and laptop Macs

» Any recent release of macOS

For Linux computers

» Suse Linux
» Ubuntu
» Debian
» Fedora
» Or almost any other recent Linux distribution

For illustrative purposes, this book mostly uses Windows 10, along with MacOS and Debian-based Mint Linux when necessary.

KEY TERMS IN THIS CHAPTER

Clean screen minimize the user interface elements to maximize the drawing area.

Cursors provide feedback from Windows and the program.

Pick button is the left button on the mouse.

Pickbox specifies the points being picked (a.k.a. selected).

ABBREVIATIONS

Alt Alt (or alternate) key

CCS Custom Coordinate System

Ctrl Ctrl (or control) key (**Cmd** on MacOS computers)

F Function key

U Undoes the last command or option

UI User interface

COMMANDS

Command	Windows & Linux Shortcuts *	MacOS Shortcuts	Menu Selections **
Help	? *or* F1	F1 *or* Cmd+?	Help | Help
CleanscreenOn	Ctrl+0	Cmd+0	View | Clean Screen
CsIcon	Alt+v l c	Alt+v l c	View | Display | CCS Icon
Exit	Ctrl+q	Cmd+q	File | Exit
Line	l	l	Draw | Line
Properties	Ctrl+1	Cmd+1	Tools | Properties
(Minimize ribbon)	Ctrl+F1	Cmd+F1	...
Undo	Ctrl+z	Cmd+z	Edit | Undo

*) The F refers to function keys. For instance, **F1** means press function key F1.

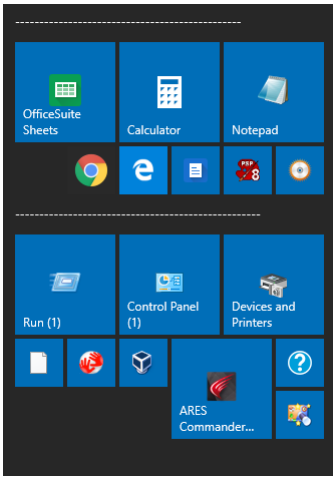
) The vertical bar separates menu selections. For instance, Help | Help means: from the **Help menu, select the **Help** item.

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- New set of high-res (4K) icons for Light theme.
- Added palettes for plugins, drawing recovery manager, layers manager, cloud storage, and comments.
- New button on the status bar for checking for software updates.

STARTING THE PROGRAM IN WINDOWS 8 & 10's MODERN (METRO) INTERFACE

This program was one of the very first CAD programs certified to work on Windows 8; today it works on Windows 10 as well. If your Windows computer or tablet is running the Modern (a.k.a Tablet mode) interface, then tap the program's tile to start it up.

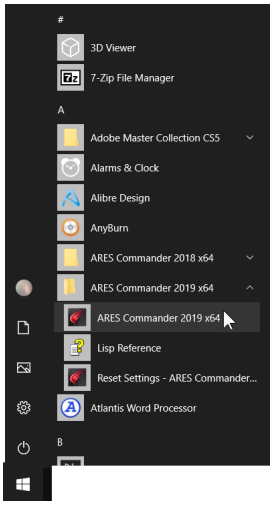


Accessing the program from the Modern interface of Windows 8 and 10

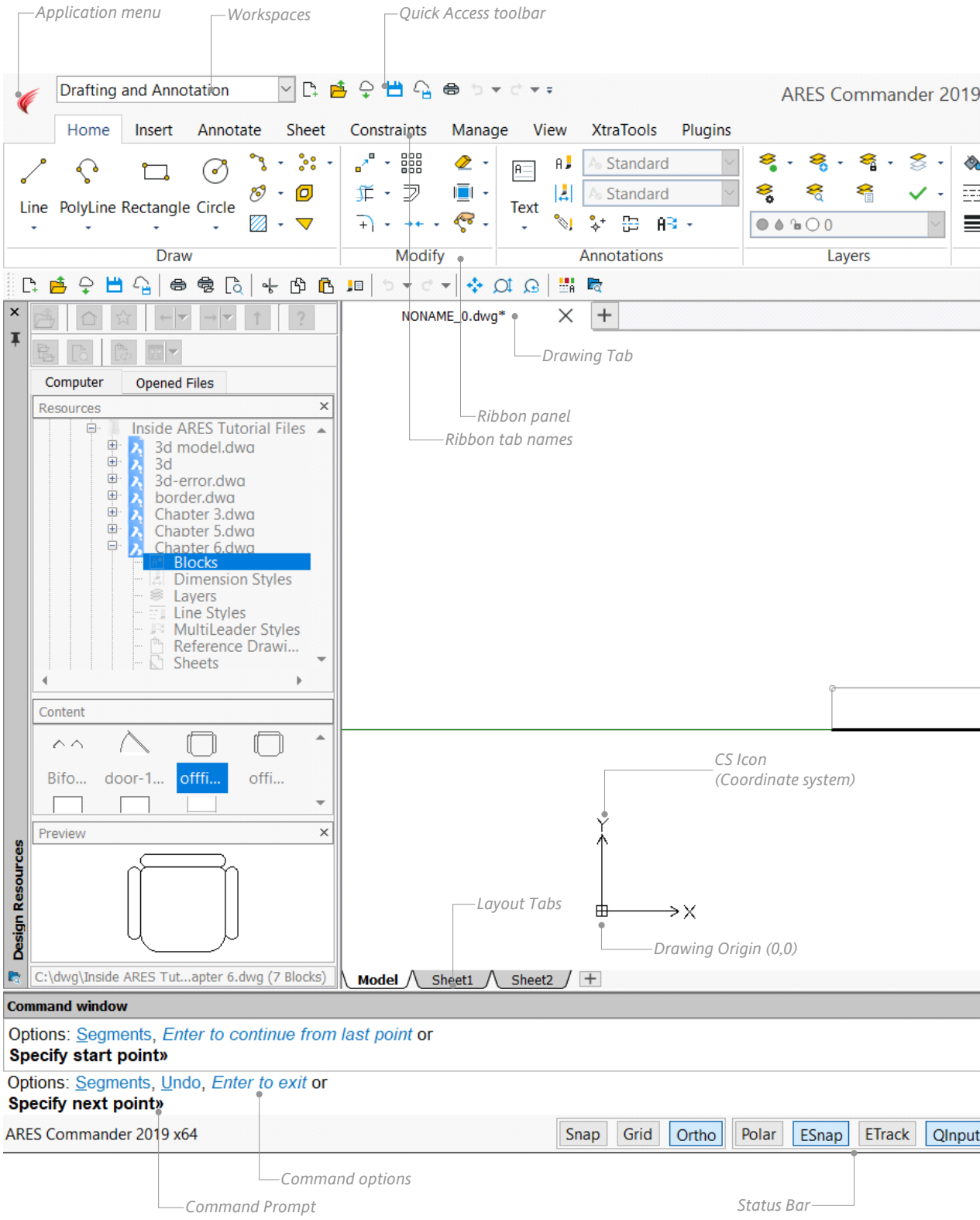
STARTING THE PROGRAM IN WINDOWS 7 AND 10

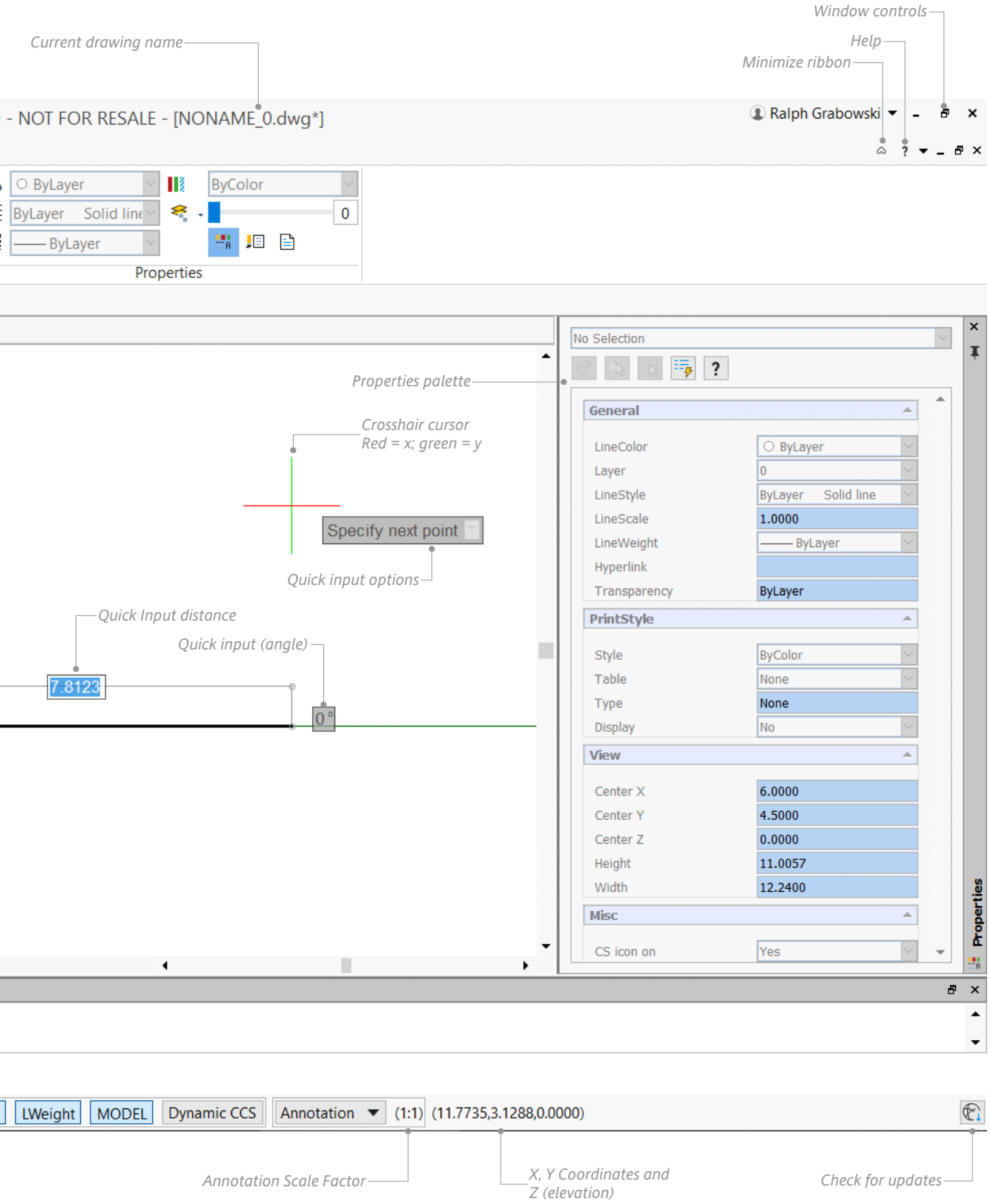
To start the CAD program from the desktop of Windows 7 and 10, double-click the icon found on the desktop. If you do not see the icon (illustrated at left), then follow these steps:

1. Click the taskbar's **Start** button — or the button in Windows 10.
2. Select **All Programs** — select **All Apps** in Windows 10.
3. From the submenu, choose **ARES Commander 2019**, then **ARES Commander 2019**.



Windows 10 showing its Start menu


Application menu
Workspaces
Quick Access toolbar
Drafting and Annotation
ARES Commander 2019
Home
Insert
Annotate
Sheet
Constraints
Manage
View
XtraTools
Plugins
Line
PolyLine
Rectangle
Circle
Text
Standard
Standard
Draw
Modify
Annotations
Layers
NONAME_0.dwg*
Drawing Tab
Ribbon panel
Ribbon tab names
Computer
Opened Files
Resources
Inside ARES Tutorial Files
3d model.dwg
3d
3d-error.dwg
border.dwg
Chapter 3.dwg
Chapter 5.dwg
Chapter 6.dwg
Blocks
Dimension Styles
Layers
Line Styles
MultiLeader Styles
Reference Drawi...
Sheets
Content
Bifo...
door-1...
offfi...
offi...
Preview
Design Resources
C:\dwg\Inside ARES Tut...apter 6.dwg (7 Blocks)
CS Icon
(Coordinate system)
Layout Tabs
Drawing Origin (0,0)
Model
Sheet1
Sheet2
Command window
Options: Segments, Enter to continue from last point or
Specify start point»
Options: Segments, Undo, Enter to exit or
Specify next point»
ARES Commander 2019 x64
Snap
Grid
Ortho
Polar
ESnap
ETrack
QInput
Command options
Command Prompt
Status Bar

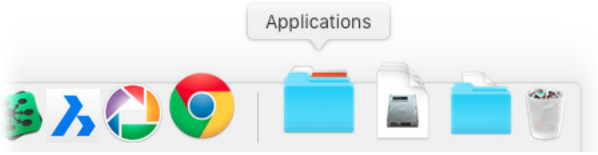

Window controls
Help
Minimize ribbon
Current drawing name
- NOT FOR RESALE - [NONAME_0.dwg*]
Ralph Grabowski
ByLayer
ByColor
ByLayer Solid line
0
ByLayer
Properties
Properties palette
Crosshair cursor
Red = x; green = y
Specify next point
Quick input options
Quick Input distance
Quick input (angle)
7.8123
0°
No Selection
General
LineColor ByLayer
Layer 0
LineStyle ByLayer Solid line
LineScale 1.0000
LineWeight ByLayer
Hyperlink
Transparency ByLayer
PrintStyle
Style ByColor
Table None
Type None
Display No
View
Center X 6.0000
Center Y 4.5000
Center Z 0.0000
Height 11.0057
Width 12.2400
Misc
CS icon on Yes
Properties
LWeight
MODEL
Dynamic CCS
Annotation
(1:1)
(11.7735,3.1288,0.0000)
Annotation Scale Factor
X, Y Coordinates and
Z (elevation)
Check for updates

STARTING THE PROGRAM ON MAC OS X

Before starting the Mac version, your computer must be running a recent release of OS X. If it is not yet set up on your Mac, install it using the instructions in Appendix F.

Once installation is complete, go to the Application folder to start the program, as follows:

1. Click the **Application** icon on the dock.

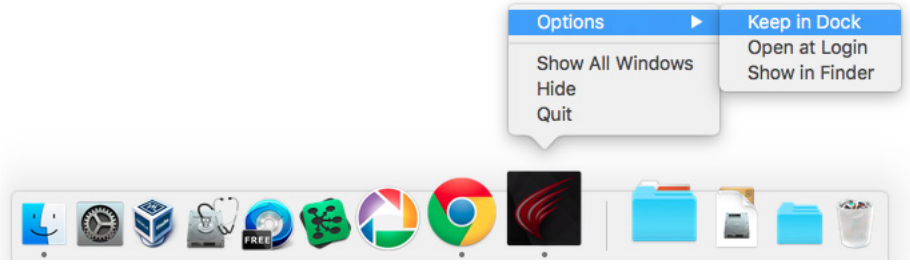


Application icon in the Mac dock

2. Find the program in the folder, and then click the icon to start it.

Locating the icon in the Application folder

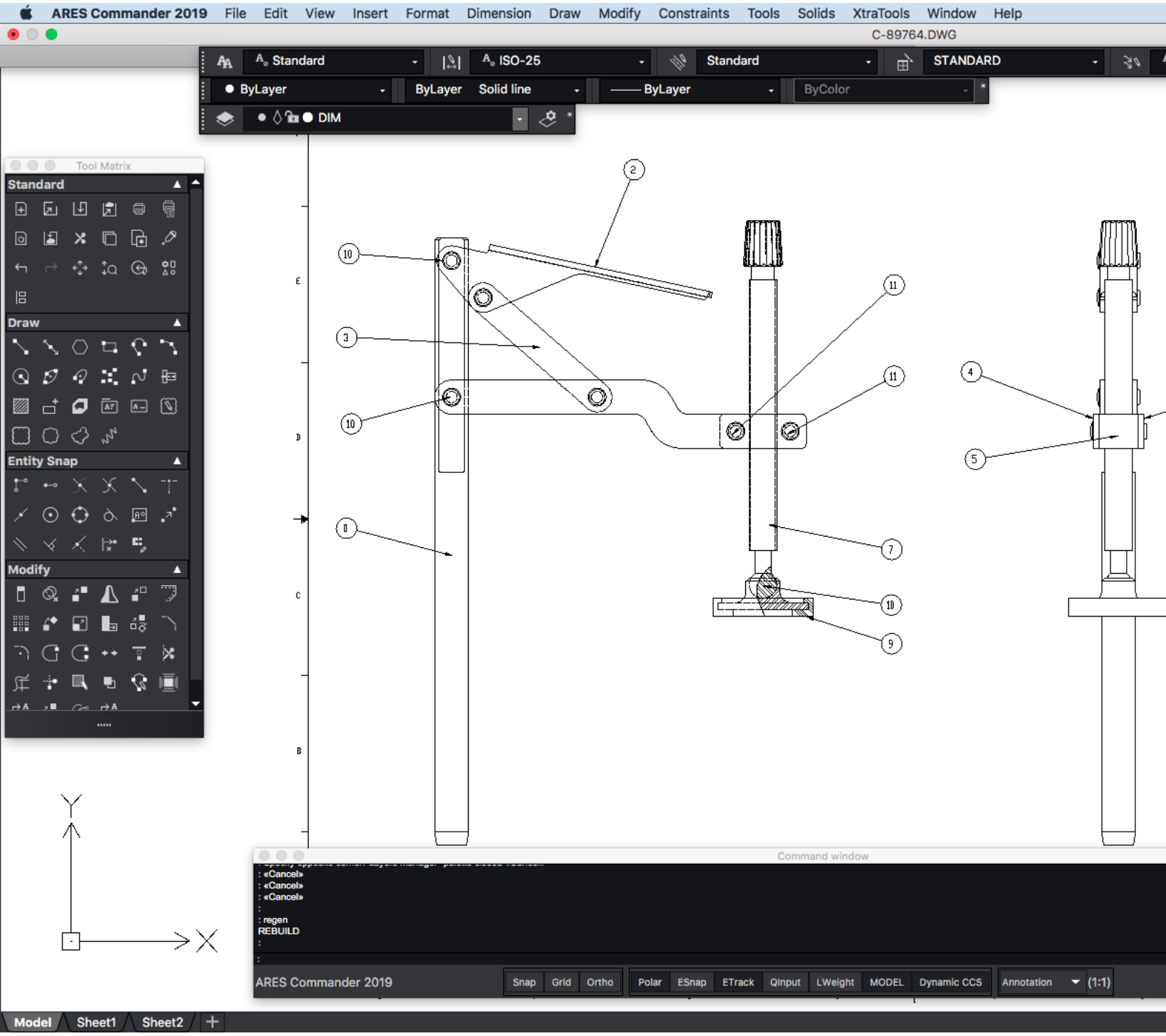
3. Once the program is running, you can keep to the program's icon on the dock to make it easier to start in the future. Right-click the icon, and then choose **Options | Keep in Dock**.



Icon in the Mac dock

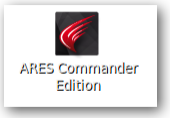
TIP If Mac OS X complains the program is from an unidentified developer, open **System Preferences**, choose **Security and Privacy**. In the **General** tab, click the padlock to allow changes, and then change **Allow Apps Downloaded From** to "Anywhere."

ARES 2019 for MacOS



STARTING THE PROGRAM ON LINUX

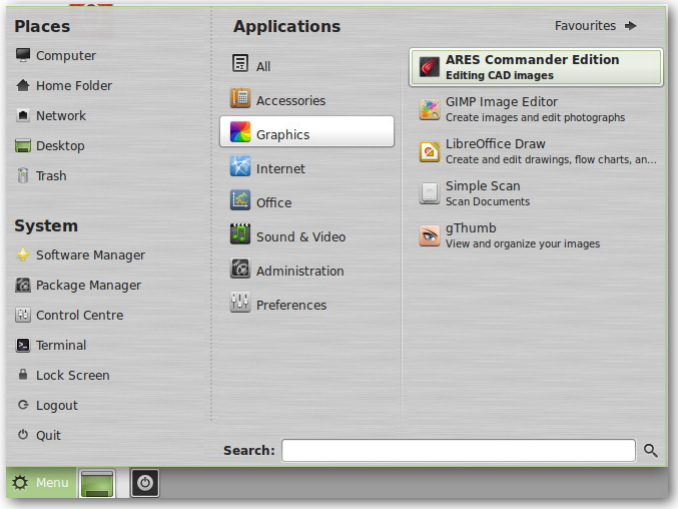
Your computer can be running Suse Linux, Ubuntu, Debian, Fedora, or another Linux distribution. (In this book, Debian-based Mint Linux is used for illustrative purposes.) If the program is not yet set up on your PC, install it following the instructions found in Appendix F.



Once the installation is complete, you may wish to place the program's icon on the desktop; this makes it easy to find and start the program. Then, to start the program, click its icon on the Linux desktop.

Alternatively, follow these steps:

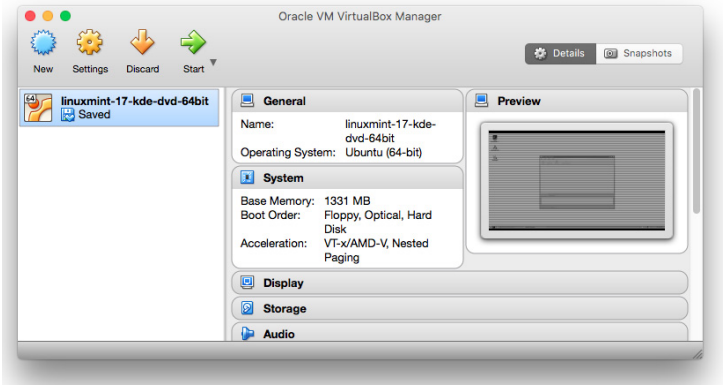
1. Click the **Main Menu** button on the Linux taskbar.
2. Select **Graphics**.
3. Choose **ARES Commander**.



The program in the Linux Mint main menu

TIP You can run Linux and Linux programs on your Windows or Mac computer through the use of virtual machine software, such as VM Virtual Box (free from Oracle) at www.virtualbox.org.

Virtual machine software is a great way to run other operating systems, such as dialects of Linux, old and new releases of Windows, and even x86 versions of Android and Chrome OS. This eliminates the need to dedicate entire computers to a rarely used OS, or to worry about partitioning hard drives, or to install dual-boot software.



THE PROGRAM WINDOW

When the program starts up, it displays a window consisting of a large graphical drawing area surrounded by areas of information on the four sides. The image on the previous page is from Windows, but the program looks almost exactly the same in Linux and MacOS; the primary difference between them is that only Windows has the ribbon interface. While the program's initial interface uses dark elements, this book shows them in light elements as they show better on these pages.

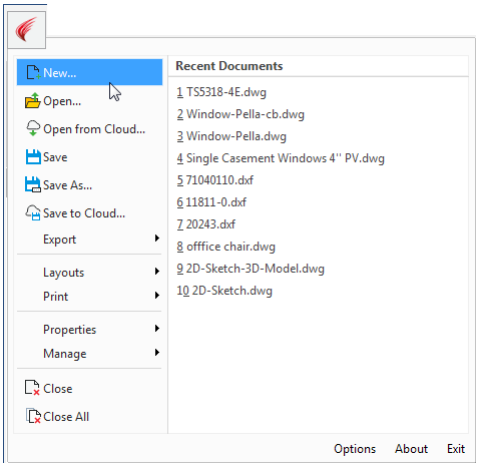
Upper Part of the Window

At the top of the program's window, we see the title bar. It hosts the name of the current drawing.

The title bar showing the program name and current drawing name and other program elements

When the filename sports an asterisk (*), it means that the drawing has changed and should be saved. Should the words "Read-only" appear next to the file name, it means the file cannot be saved under its own name, but must be saved by another name. More on read-only files later.

(WINDOWS ONLY) On the far left of the title bar, you see the ARES icon ( ) that hints at the *Application menu*. It is very important, as it holds all file, print, import-export, and utility functions, as illustrated below. In the Linux and MacOS versions, click the File menu to access the same file and utility functions.



Application menu with file and utility commands

(WINDOWS ONLY) Next door to the Application menu are the Workspace droplist and Quick Access toolbar.

Quick Access toolbar

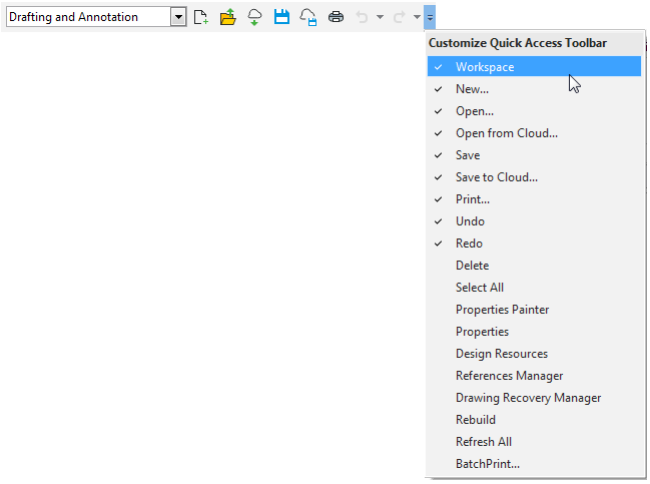
The Workspaces droplist Drafting and Annotation provides quick access to workspaces, which are different user interfaces for the program, such as "3D Modeling" and "Classic." (The Classic workspace displays the menu bar and toolbars.)

The mini toolbar gives you quick access to commonly used utility functions, like saving and printing drawings. In the Linux and MacOS versions, click the File menu to access these functions.

From left to right, the buttons on the Quick Access toolbar perform the following functions:

Button	Meaning	Action
	Starts a new drawing from a template	Displays Specify Template dialog box
	Opens an existing drawing file from the drive or network	Displays Open dialog box
	Opens a drawing file from the cloud	Displays Cloud Storage palette
	Saves the current drawing to disk	Displays Save As dialog box when a new drawing has never been saved, or when the drawing is read-only
	Saves the current drawing to the cloud	Displays Cloud Storage palette
	Prints the current drawing	Displays Print dialog box
	Undoes the last command	...
	Redoes the last undone command	...
	Edits the Quick Access toolbar	Displays dropdown menu, as shown below

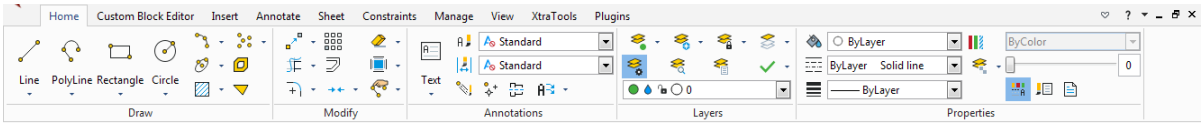
You can change which buttons this toolbar displays by clicking the down arrow ( ), and then choosing commands from the list. A check mark next to a name means the button is displayed by the toolbar; no check mark means it is not displayed. Click a name to turn the check mark on and off.



Changing the buttons shown by the Quick Access toolbar

The Quick Access toolbar can be customized further with the Customize command.

(WINDOWS ONLY) The *ribbon* dominates the upper part of the program, with its tabs and panels. Shown below is the "Drafting and Annotation" ribbon, which offers commands specific to 2D drafting and editing. Instead of the ribbon, the MacOS and Linux versions show the menu bar and several toolbars.



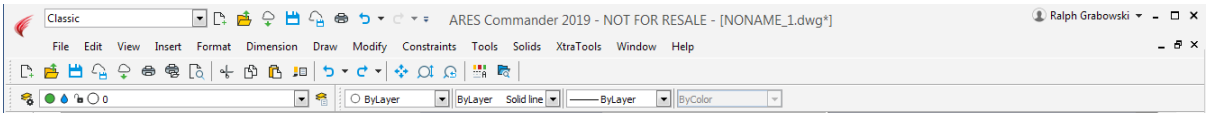
Ribbon with commands for drafting and annotation activities

The ribbon is segregated into several *tabs* that group similar functions, such as **Home** (the most common functions) and **Insert** (inserting entities into the drawing).

Panels segregate tabs into groups of related commands, such as **Draw** (drawing entities) and **Modify** (editing the entities). You can customize the tabs and panels with the Customize command.

Menus and Toolbars

The Windows version of the program displays the menu bar and toolbars when the workspace is set to "Classic." The Linux and MacOS versions display the same menu bar.

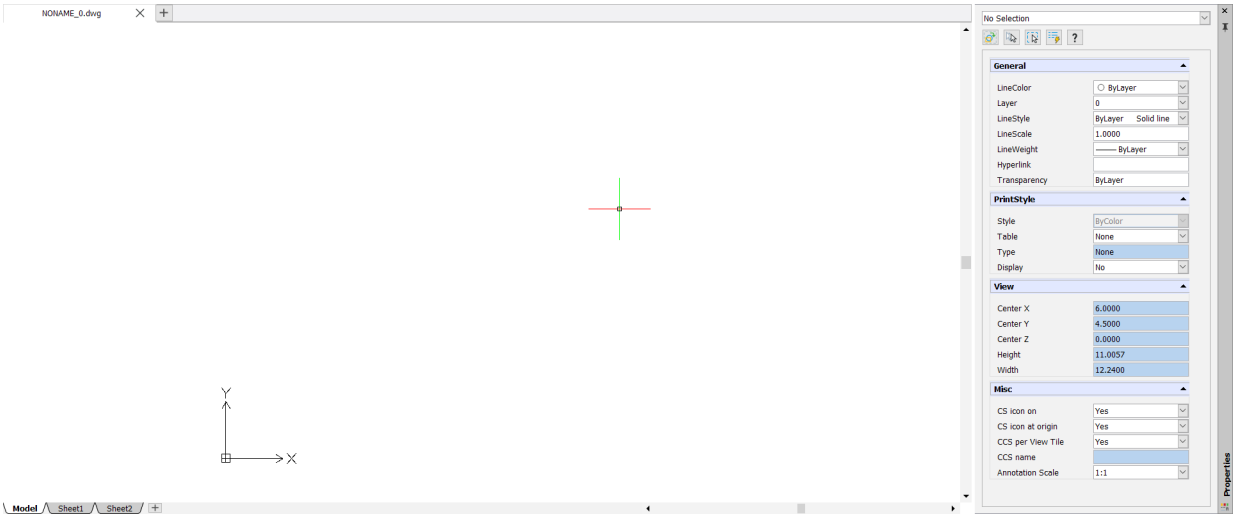


Classic workspace displaying menu bar and toolbars

The menu bar and toolbars operate just like in any other Linux, MacOS, or Windows program.

MIDDLE PART OF THE WINDOW

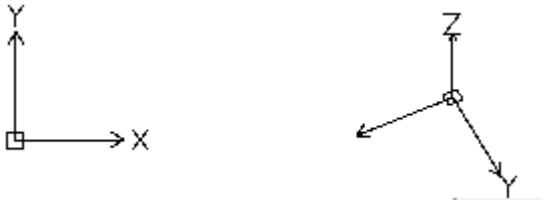
The center of the program is dominated by the large drawing area. Here you see the drawing, if loaded. As you move the mouse, you see an arrow cursor ( ), along with a square pickbox cursor for picking entities in the drawing. This arrow cursor changes to a red-green crosshair ( ) during drawing and editing commands.



The drawing area with drawing tabs (along the top), the crosshair cursor (center), and CCS icon (lower left)

Along the top of the drawing area is a series of tabs that represent the drawings that are open. Click a tab to switch quickly to another drawing. See the details on using these tabs later in this chapter.

In the lower corner of the drawing area is the custom coordinate system (CCS) icon with its x, y, and hidden z axes. The CCS is used in three-dimensional (3D) modeling primarily, because it helps you orient yourself into 3D space.



Left: CCS icon in 2D drafting mode; right: in 3D modeling mode

To the right of the screen is the Properties palette. It displays the properties of the currently selected object(s); when no objects are selected, then it reports the current default values of properties. If you do not see the palette, then press **Ctrl+1** to display it: hold down the **Ctrl** key, and then press **1**. (On Macs, hold down the **Cmd** key, and then press **1**.)

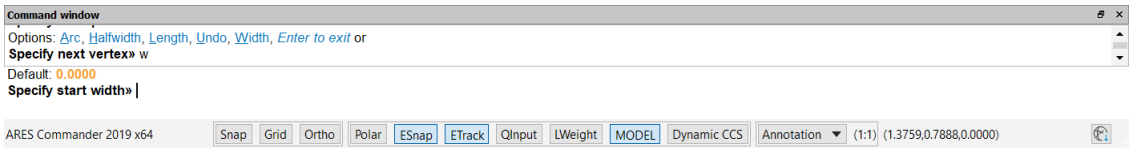
Along the bottom of the drawing area are model and layout sheet tabs, along with optional scroll bars. The tabs switch between model view and layout (sheet) views of the drawing. *Sheets* are used to position and size the drawing prior to printing; see Chapter 18 for how to use this function.

***Left**: Model and sheet tabs; **right:** scroll bars*

TIP Scroll bars are useful for panning the drawing, and so I turn them on through the Options dialog box. To find the setting, enter **Options** at the command prompt, and then type in the **Find** field "status bar."

BOTTOM PART OF THE WINDOW

At the bottom you find the *Command window* and below it, the *status bar*. The Command window is where you enter command names and options, and read responses from the program. See boxed text for more about the Command Window.

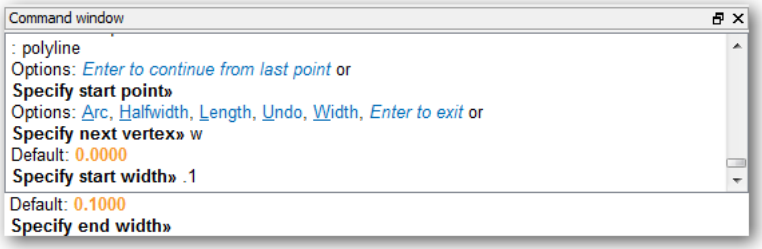


Command bar and status bar

At the very bottom is the *status bar* that reports the state of the program, such as the settings of drawing modes. For instance, Snap is off and is shown with a gray background; ESnap is on and is shown with the **blue** background. At the right end of the status bar is the x,y coordinate position of the cursor.

COLORS OF THE COMMAND TEXT

The program uses colors to help you distinguish between different kinds of text in the command bar. The figure below shows some examples:



Blue text indicates options, as in AutoCAD. You can enter the entire name, or just the option's abbreviation, as indicated by the underlining. For instance, for the **Width** option, you can enter either **width** or just plain **w**.

Blue italicized text indicates the option that the program would execute when you press **ENTER** without entering an option, such as *Enter to continue from last point*.

Amber text displays the current default value, such as **0.000**. The default is either the system's default, or else the value you entered most recently (**0.1000** in the example above).

Navigating the User Interface

So many aspects of the program's user interface can be a bit daunting to learn all at once, and so here we look at just some of the basics:

- Understanding the cursor and its shapes
- Tabbing between drawings
- Using commands in the Command window
- Revealing hidden user interface elements
- Secrets of the menu bar
- Employing the Options toolbar
- (WINDOWS ONLY) Exploring the Info Center
- Accessing information from palettes
- Customizing the Tool Matrix
- Turning off the CCS icon
- Accessing help

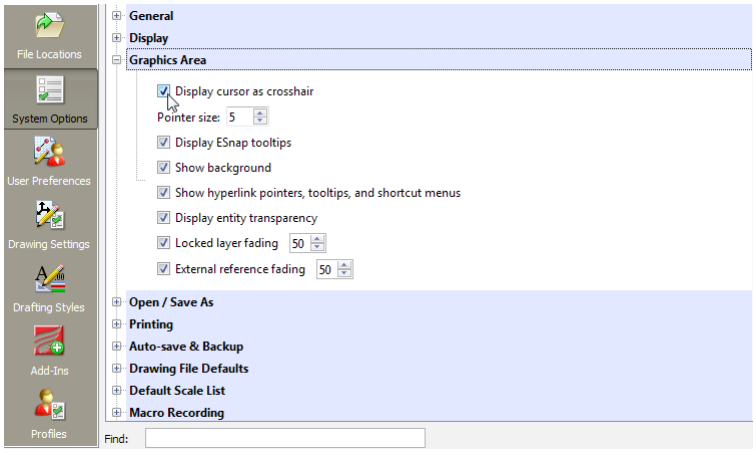
THE CROSSHAIR AND OTHER CURSORS

Cursors give you feedback from Windows, the program, and other software. While in the drawing area, the cursor looks like an arrow. At the tip of the arrow is a small square called the *pickbox*. The cursor shows you the precise location in the drawing, while the pickbox shows you the area in which you can select (or "pick") objects in the drawing.

Arrow cursor and square pickbox

If you prefer to use a cursor that looks like a crosshair, then you can change its look. Follow these steps:

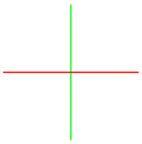
1. Enter the **Options** command.
2. In the panel on the left, choose **System Options**.



Navigating the Options dialog box to the cursor option

3. Click the ⊞ in front of **Graphics Area**. This opens up the section.
4. Select the **Display Cursor as Crosshair** options.
5. Click **OK** to close the dialog box.

Notice that the cursor now takes on the look of a red and green cross. The red indicates the x axis, while the green the y. The pickbox is black and at the center of the cursor.



Red and green crosshairs referring to x and y axes

To move the cursor around the program's window, move your mouse.

When you move the crosshair cursor out of the drawing area, it again changes to an arrow. You are probably familiar with this arrow cursor from other Windows applications. The arrow cursor lets you make menu selections and pick toolbar buttons.

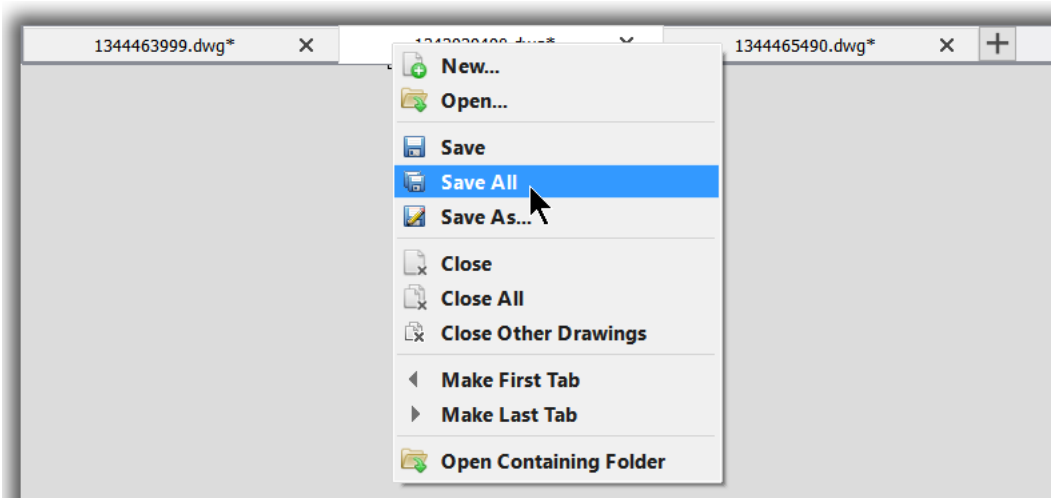
Cursor changing shape

The cursor can change to other shapes, as well. For example, when the cursor turns into a double-ended cursor, you can resize the program's window, palettes, and toolbars.

TABBING BETWEEN DRAWINGS

The program sports tabs along the top of the drawing area that show the names of open drawings. This is useful for switching quickly between drawings. These tabs are initially turned on; the MacOS version displays drawing tabs when UiMode is set to Yes.

Tabs are turned on with the **DrawingTabs** command, and off with **HideDrawingTabs**. Right-click a tab to see a shortcut menu of options.



Shortcut menu for drawing tabs

Some of the options are commands that control drawings, such as **Save All** (saves all drawings) and **Open Containing Folder** (opens the folder in which the drawing file resides).

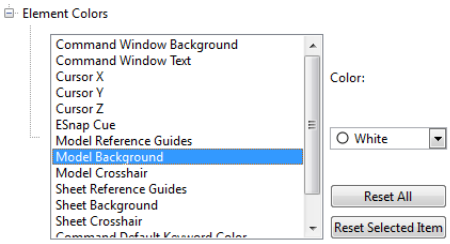
Other options in the shortcut menu control the tabs themselves. For example, **Make First Tab** move the current tab leftward to the start of the list. Click the **X** on the tab to close the drawing; click the **+** at the end of the tabs to start a new drawing.

SECRETS OF CHANGING THE USER INTERFACE

Many user interface options can be changed with the Options dialog box, accessed from the **Options** button in the Application Menu (**Application | Preferences** on Mac).

BACKGROUND COLOR

To change the drawing area's background color from black to white, follow these steps:



1. Open the Options dialog box with the **Options** command.
2. Select the **System Options** section, and then open the **Display** node.
3. Open the **Element Colors** node.
4. Choose **Model Background**, and then select Color "White."
5. Click **Apply** to make the change.

CROSSHAIR CURSOR SIZE

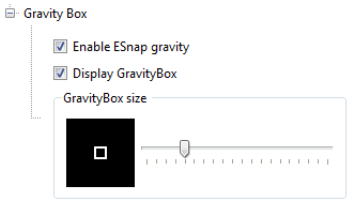
To change the size of the crosshair cursor, follow these steps:

1. Select the **System Option**s section, and then open the **Graphics Area** node.
2. Look for P**ointer Size**.

The default value is 5, which means the length of the crosshair is 5% of the screen's size. When set to 100, the cursor stretches across the entire drawing area.

PICKBOX SIZE

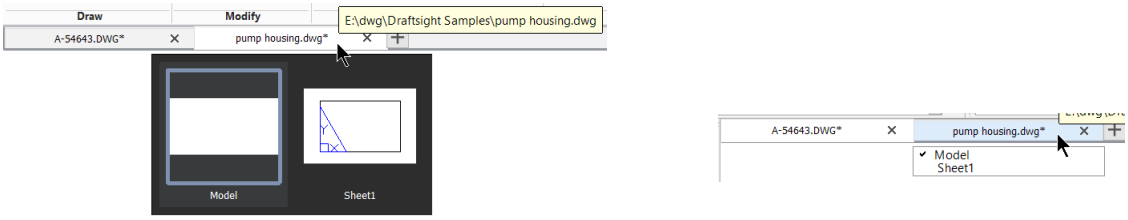
To change the size of the square pickbox, follow these steps:



1. Select the User Preferences section
2. Open the Drafting Options node.
3. Open the Display node and then the Gravity Box section.

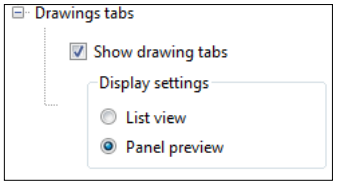
The size ranges from 0 to 50 pixels; the default size is 3.

When you pause the cursor over a tab, the program shows a simplified preview of model space and layouts, as well as the path and filename of the drawing.



*Previewing model and paper spaces with (**left**) Panel Preview and (**right**) List View*

There are two ways of viewing previews, as a panel or as a list; see figure above. You switch between the modes through the Options dialog box (System Options > Display > Drawing Tab) or else with the **FileTabPreview** variable.



Options for drawing tabs

The **FileTabState** variable toggles whether or not the drawing tabs are displayed. I recommend leaving them on, because they are so super useful for switching between drawings!

THE COMMAND WINDOW

Command:	CommandWindow
Aliases:	pr, props, ch, ddmodify, mo
Shortcuts:	Ctrl+9 (Cmd+9 on Macs)
Shortcut menu:	Right-click toolbar and choose Command Window

Along the bottom of the program's window is the Command window. This is pretty much the most important part of the program. Here you type commands — if you are like me and prefer typing over making menu selections with the mouse — and where the program tells you what to do next.

The Command window

Typing command names and options is a fast way to draw and edit when you are a touch typist, but it is slower if you are not. The command area is also the place where the program displays additional options, and prompts you for additional information it might need to complete a command. When you see the ' : ' prompt by itself, like this...

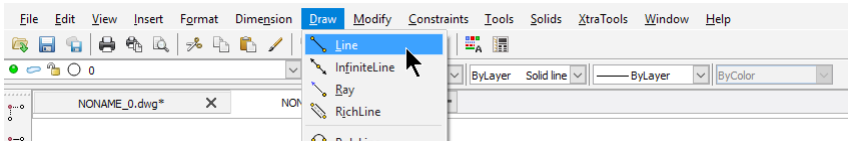
:

...this means that the program is ready for you to enter a command. If you want to enter a command, but there is text after ' : ', press the **Esc** key once or twice to cancel the current command.

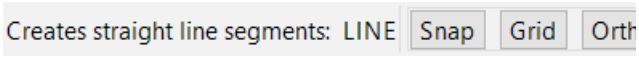
SECRETS OF THE MENU BAR

The program sports a menu bar similar to ones used by other programs. (If you don't see the menu bar in Windows, turn it on through the Quick Access toolbar: from the **Workspaces** droplist, choose "Classic Default.")

When you move the cursor over a menu item, the program helpfully provides a summary of the command's purpose on the status bar, found at the very bottom of the window. This is useful when you are not sure of a command's purpose. For instance, move the cursor over the **Draw** menu's **Line** command...



...to see the following explanation on the status bar:



As you peruse the menus and ribbon, you may come across check marks and keystroke shortcuts associated with menu items. The symbols have the following meanings:

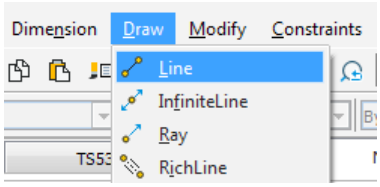
Menu Symbol	Example	When selected...
(none)	Refresh	... executes a command
... (ellipses)	Named Views...	... displays a dialog box
▸ *(arrowhead)*	3D Views ▸	... displays a submenu
✓ *(check mark)*	✓ Normal	... turns off the option
+	Ctrl+0	... a Ctrl shortcut keystroke can be used (Windows and Linux)
⌘+	Cmd+0	...the Cmd key is used on Macs
_ *(underline)*	New	... the shortcut is for use with the Alt key
(black text)	Plot Style	... the command is available
(gray text)	Plot Style	... the command is not available and so cannot be selected

Icons in menus have no function other than to provide a visual reminder of a command's function.

If you pick an item on the menu bar by accident, you can "unselect" it by picking the item a second time.

MOUSELESS MENU ACCESS

- To access menus without a mouse, here's how with the keyboard:



1. Hold down the **Alt** key. Notice each word on the menu bar has a letter underlined: Draw.
2. Press the **D** key to access the Draw menu.
3. Again, every menu item has a letter underlined. Press the **L** key to execute Line command.

CTRL+0 is an example of a *keystroke shortcut*. Instead of selecting items from menus or typing command, you can hold down the Ctrl key and press the 0 (zero) key. (On Macs, press Cmd+0.) Some users find using the keyboard is faster than menus or the ribbon.

Try drawing a few lines now:

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

   ```
   Command: line (Press Enter)
   ```

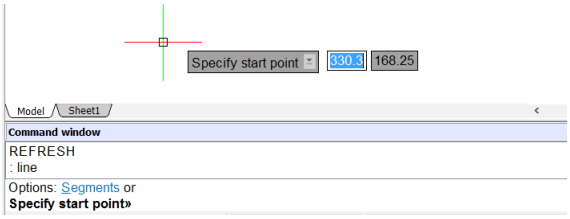
   Type the word **line**, and then press the **Enter** key. Pressing Enter tells the program you are finished entering the command name.

2. The program changes the prompt from ' : ' to the following:

   ```
   Options: Segments or
   Specify start point: (Move the mouse to move the cursor)
   ```

   The program asks you to where it should start drawing the line. One way it is to indicate the starting point with your mouse. Move the mouse, and you see the crosshair cursor move about the drawing area of the window.

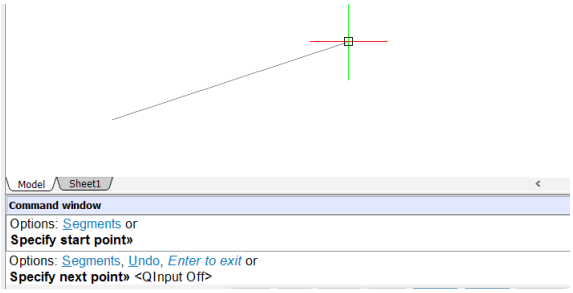
TIP If you find the Quick Input interface getting in the way, turn it off by clicking the **QInput** button on the status bar.



Draw a line begins with picking its starting point (Quick Input is turned on)

(The command window also shows the phrase 'Options: **S**egment or'. If you enter **S**, then the program would switch to *segment* mode, where it draws a single line segment.)

3. Pick a point on the screen by pressing the left button on your mouse. This button is known universally as the "pick button." The program remembers the x,y coordinates of your pick point, making them the start of the line.



Showing the extent of the line by ghosting it (Quick Input is turned off)

4. Move the mouse again. Notice the line stretching from the starting point to the cursor. This line is known as a "ghost line" or as a "rubber band." Before you pick the next point, it previews the position of the line for you.

5. Press the pick button again. You have drawn your first line in the program!

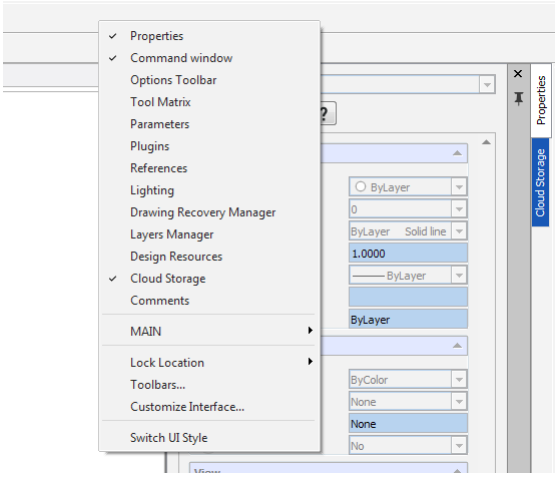
6. In the command window, the program changes the prompt to read:

   ```
   Options: Segments, Undo, Enter to exit or
   Specify next point or [Undo]: (Pick another point)
   ```

SECRETS OF HIDDEN UI ELEMENTS

The program "hides" some user interface elements, because not all users need them. Here is how to reveal them.

1. Right-click the ribbon or any toolbar. Notice the shortcut menu:



2. From the shortcut menu, select the name of a palette. A check mark indicates that the palette is displayed.

Shortcut Menu	Meaning	Related Command
Properties	Toggles the Properties palette	Properties
Command window	Toggles the Command window	CommandWindow
Options Toolbar	Toggles the Options toolbar	...
Tool Matrix	Toggles the Tool Matrix palette	ToolMatrix
Parameters	Toggles the Parameters palette	Parameters
Plugins (NEW IN 2019)	Toggles the Plugins palette	Plugins
References	Toggles the References palette	References
Lighting	Toggles the Lighting palette	LightList
Drawing Recovery Manager (NEW IN 2019)	Toggles the Drawing Recovery Manager	DrawingRecovery
Layers Manager (NEW IN 2019)	Toggles the Layers Manager palette	Layer
Design Resources	Toggles the Design Resources palette	DesignResources
Cloud Storage (NEW IN 2019)	Toggles the Cloud Storage palette	CloudStorage
Comments (NEW IN 2019)	Toggles the Comments palette	Comments
MAIN	Displays submenu of toolbar names	
Lock Location	Locks floating and/or docked toolbars ✔ Floating Toolbars Docking Toolbars • **Floating Toolbars** locks floating toolbars • **Docking Toolbars** locks docked toolbars	LockUI
Toolbars	Displays the Specify Toolbars dialog box	Toolbars
Customize Interface	Displays the Customize dialog box	Customize
Switch UI Style	Switches interface between light and dark	SwitchUI

(NEW IN 2019) When multiple palettes are open, the program nests them so that they do not take up too much screen real estate. Switch between the palettes through tabs found on their edges.

Continue by drawing more lines: moving the mouse and pressing the pick button. Each time you press the pick button, the program draws another line segment.

7. You end the Line command by pressing the **Enter** key, by pressing the **Esc** key, or by pressing the right mouse button, as follows:

```
Specify next point: (Press Enter or Esc)
```

Pressing the **Esc** key cancels *any* command; sometimes, complex commands may need you to press the key two or three times.

TIPS Pressing the **Enter** key can have three different effects, depending on the prompt active at the time. Here are some examples using the Line command:

- At the 'Specify start point:' prompt, pressing **Enter** causes the program to continue drawing from the last picked point, whether line or arc. This ensures the line is drawn tangent to the end of an arc.
- At the 'Specify next point:' prompt, pressing **Enter** terminates the command.
- At the ' : ' prompt, pressing **Enter** repeats the last command, in this case the Line command.

You don't need to type the entire command name. Start typing the first one or two letters, such as "ci"and then press **Tab**. The program finishes the name for you, as follows:

: **ci** *(press Tab)*
CIRCLE

If this is not the command you want, keep pressing Tab and the program works through all other command names starting with "ci" in alphabetical order:

CIRCLE *(press Tab)*
CIRCLERAD

---

8. To erase the lines you just drew, type **U** at the ' : ' prompt (short for "undo").

```
: u
Undo LINE
```

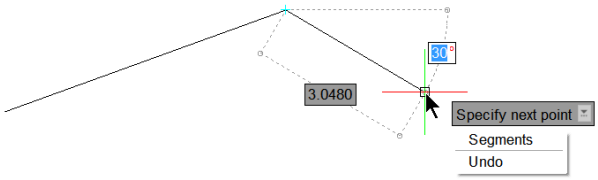
Alternatively, you can execute the U command these ways:

» From the Edit menu, select **Undo**
» Press **Ctrl+z**. (On Macs, press **Cmd+z**.)
» Select the undo icon from the toolbar.

As you can see from this example, the program provides different ways to perform actions. You will probably find yourself using a combination of typing at the keyboard, choosing toolbar icons, making menu picks, and entering keyboard shortcuts — whichever you find most convenient.

QUICK INPUT

When you start to draw, the program lets you work heads-up. Instead of entering commands and options at the command prompt, the new Quick Input user interface allows you to see and enter command options, coordinates, distances, and angles right where you are drawing.

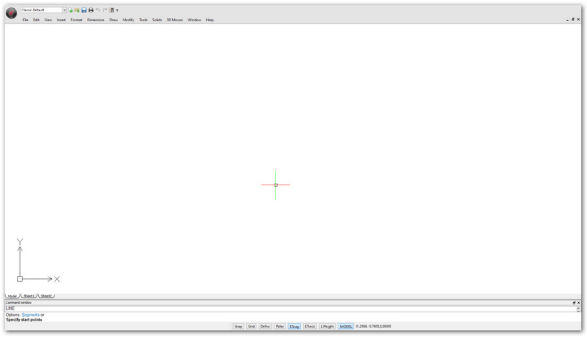


Using Quick Input to enter distances and angles, and select options

To turn Quick Input on and off, click the **QInput** button on the status bar.

TIPS To run the program in full screen mode without most user interface elements, take these steps:

1. From the **View** menu, choose **Clean Screen**. Notice the drawing area is maximized as toolbars, palettes, and other window elements disappear — even the Windows toolbar!
2. To return the program to normal, repeat the menu picks, or else reenter the **CleanScreen** command, or else press **Ctrl+0** (Cmd+0 on Macs).

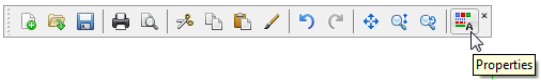


SECRETS OF THE TOOLBAR

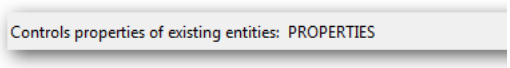
The program makes use of regular toolbars, like those illustrated below. Toolbars collect related commands as buttons; *icons* are picture on the buttons that indicate their purpose. For instance, the Properties toolbar contains droplists for changing layers, line types, and so on.

(Toolbars are not shown in the Windows version when the workspace is set to display the ribbon. To see toolbars, switch the workspace to "Classic Default.")

Sometimes I am unsure about the meaning of an icon, and so I pause the cursor over it. After a couple of seconds, the program displays a yellow tooltip showing the name of the command:



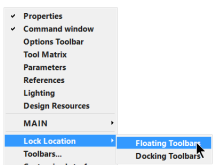
Down on the status bar, a line of helpful text appears:



Toolbar scan be floated, and then dragged to anywhere on the computer screen, or even to a second screen. Once toolbars are floating, you can move them by dragging them around the screen.

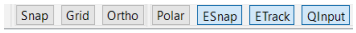
When you double-click a toolbar, it returns to its default docked position. The next time you start the program, it will remember the location of all toolbars, and open them in the same place you left them last, whether floating or docked.

Some people find that as they click buttons the toolbar shifts on them. This can be caused by moving the mouse while clicking the button. To prevent toolbars from moving, lock them by right-clicking the menu bar or ribbon, choose **Lock Location**. The two options in the submenu have the following meanings: **Floating Toolbars** locks floating toolbars in place, while **Docking Toolbars** locks docked toolbars in place.



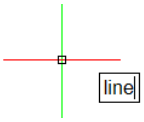
Using Quick Input can be tricky, so here is a tutorial on employing this "heads-up interface":

1. If necessary, click the status bar's **QInput** button to turn on Quick Input mode.



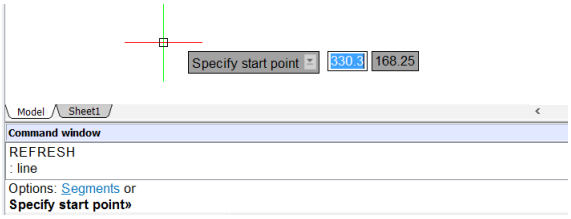
QInput button looks blue when turned on

2. Start the **Line** command. Notice that the command's name appears in the drawing area.



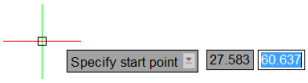
Quick Input displaying the command being entered.

3. Press **Enter**. Notice that Quick Input shows the x,y-coordinates of the cursor location, such as **330.3** and **168.25** illustrated below. The x-coordinate is in blue: this color indicates the value can be edited.



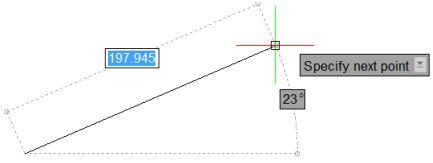
Quick Input displaying starting coordinates

4. The y-coordinate is shown in gray; this color means you cannot edit it — for now. To switch between the coordinates values, press **Tab**. Notice that the y-coordinate now turns blue, and so you can edit it.



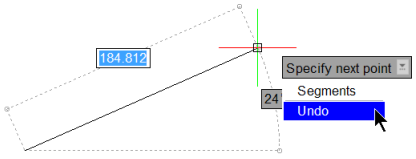
Pressing Tab to edit the other coordinate

5. To place the starting point, click with the mouse or else press **Enter** to accept the x and y values you entered.
6. Move the cursor. As you do, more Quick Input elements appear:
   - » **Length of line** — the relative distance from the last picked point (in blue below)
   - » **Angle of line** — measured from the x-axis (shown in gray)
   - » **Command's options** — in the gray box



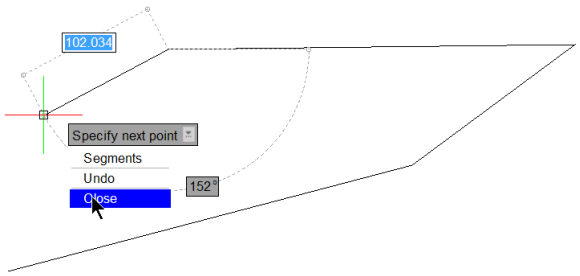
Quick Input showing the relative distance and absolute angle

7. To access a command's options, press the keyboard cursor **Down** arrow.



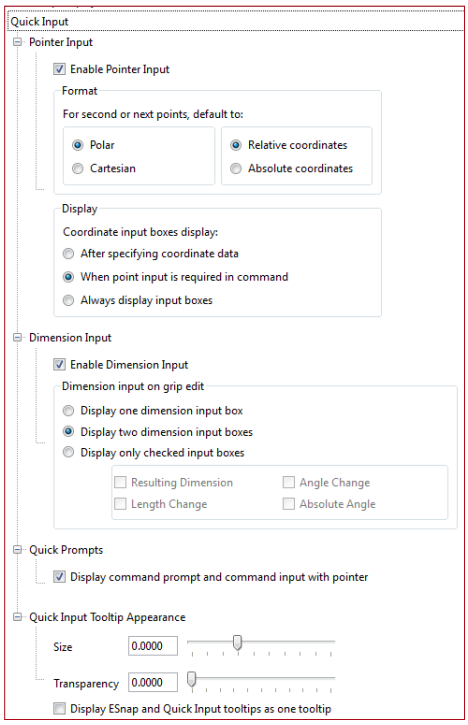
Pressing Down arrow to access command options

8. Continue drawing. Notice that you can enter options and values at the command prompt or in the Quick Input fields.



Drawing with Quick Input

There are a number of system variables available to control the look and function of quick input, but I find it easier to use the Options dialog box to make changes (User Preferences > Drafting Options):

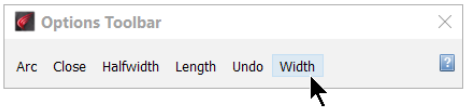


Options for Quick Input

OPTIONS TOOLBAR

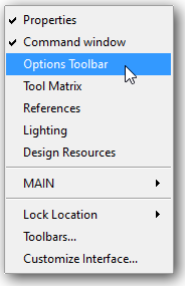
Command: --
Shortcut menu: Right-click the ribbon or any toolbar, and then choose **Options Toolbar**

The Options toolbar displays command options that you can choose with the mouse. For example, when you start the Polyline command, it displays option names like Arc, Close, and so on, illustrated below.



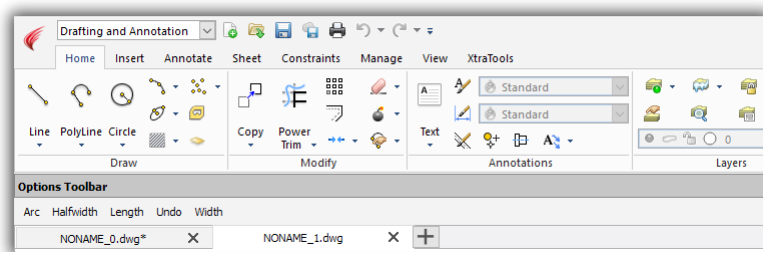
Using the Options toolbar

Normally, this toolbar is turned off, so if you want to use it, you'll have to turn it on. Right-click the menu bar, and then from the shortcut menu, choose **Options Toolbar**.



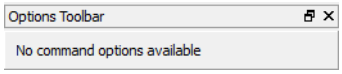
Opening the Options toolbar through a shortcut menu

Notice that the Options toolbar squeezes in below the ribbon (or toolbars) but above the drawing area.



Options toolbar appearing below the ribbon

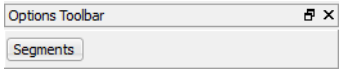
Initially, no command options are listed by it: "No command options available."



State of Options toolbar when no command is underway

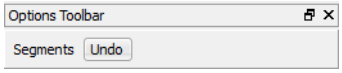
Once you enter a command, however, option names appear here. To activate one, you click on the name of an option, or else enter it as usual in the Command window. Here again is the Line command, this time coordinated with the Options toolbar:

```
Command: line (Press Enter)
Options: Segments or
Specify start point: (Move the mouse to move the cursor.)
```



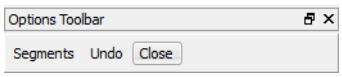
Option name on the Options toolbar mimicking the command prompts

```
Options: Segments, Undo, Enter to exit or
Specify next point or [Undo]: (Pick another point)
```



Option names being added as the command progresses

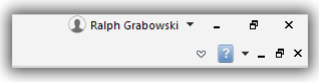
```
Options: Segments, Undo, Close, Enter to exit or
Specify next point» (Enter an option)
```



More option names being added

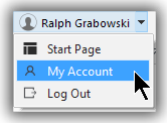
THE INFORMATION CENTER

At the far right end from the Quick Access toolbar is the information center, the location for help.



The information center

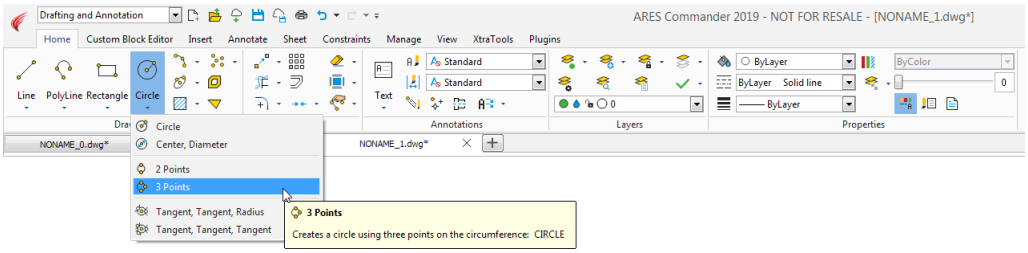
The first row gives you access to your account and other online information.



Account access

SECRETS OF THE RIBBON

The ribbon is a tabbed toolbar-like interface invented by Microsoft, and so is available on Windows only; it is not found on Linux or Mac. If you do not see the ribbon, you can turn it on by choosing "Drafting and Annotation" or "3D Modeling" from the workspaces droplist, or else entering the WsCurrent variable, and then typing the name of a workspace. (There is no "Ribbon" command.)

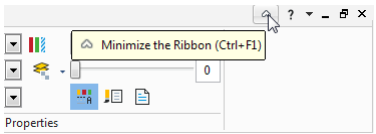


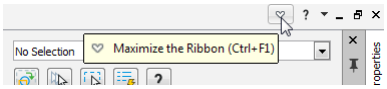
Pause the cursor over a button, and a tooltip appears explaining its purpose, as well as on the status bar.

TOGGLE RIBBON VISIBILITY

The ribbon takes up a fair chunk of screen space, and so you get more drafting space by a couple of ways:

» Click the almost invisible arrowhead, found at the upper right corner of the program, to minimize the ribbon — only the tab names remain visible. Click the arrow again to bring back the full ribbon. Alternatively, press **Ctrl+F1**.



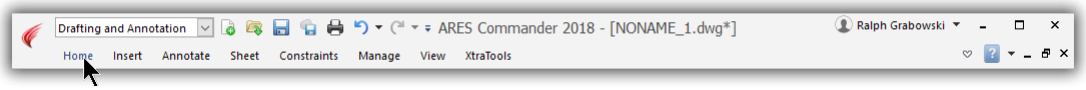


» Press **Ctrl+0** to enter full-screen mode, which removes most interface elements; press **Ctrl+0** again to get the interface back to normal.

You can customize the ribbon with the Customize command.

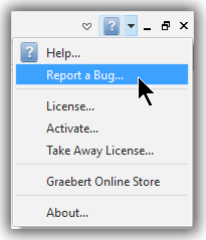
On the second row, from left to right, the buttons perform the following functions:

- Minimize the ribbon; click a second time to bring it back again. When the ribbon is minimized, the names of tabs are still visible, as illustrated below; click a tab name to display it temporarily.



Ribbon minimized to show only the names of tabs

- Access help (see the section later in this chapter)
- Access additional services, as shown below:



Additional sources of help

- Minimize the current window
- Resize the current window
- Close the current window and optionally save the drawing

These last three buttons are also present when the ribbon is turned off, and the toolbars are displayed.

PROPERTIES PALETTE

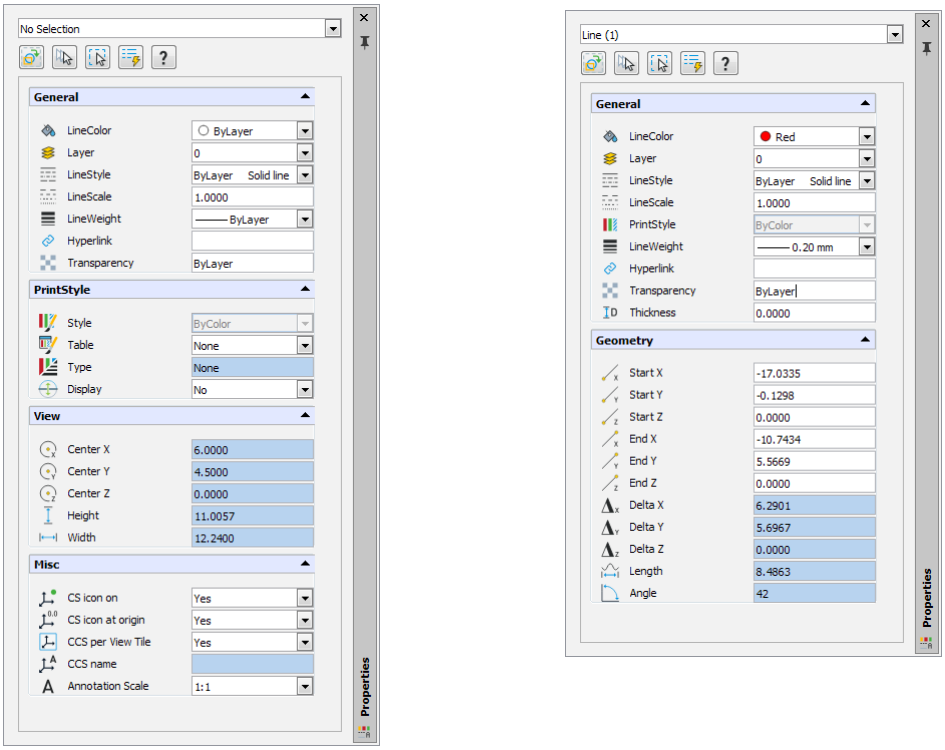
Command:	Properties
Aliases:	pr, props, ch, ddmodify, mo
Menu bar:	Tools
Ribbon:	Right-click ribbon and choose Properties Palette
Shortcuts:	Ctrl+1 (Cmd+1 on Macs)
Shortcut menu:	Right-click menu bar and choose Properties Palette
Toolbar:	

Palettes are like dialog boxes but with one significant difference: you don't have to dismiss them, and so they can always remain open as you draw and edit. (In contrast, you have to dismiss a dialog box before you can continue drawing and editing.)

The program has several palettes, such as Properties, References, and Lighting. Of these, the Properties palette is the most useful, because of the two big tasks it performs: reporting the properties of selected objects, and changing their properties.

If the Properties palette is not displayed, there are many ways to turn it on. Choose one of the methods listed above.

Initially, the Properties palette looks like this:

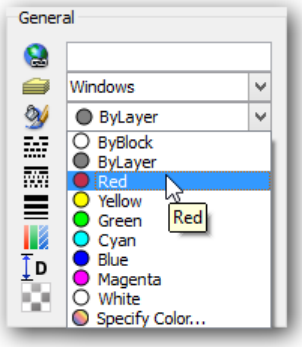


Left: *Properties palette when nothing is selected in the drawing;* ***right:*** *And when a red line is selected*

The palette shows the drawing's default properties, such as the active color, linestyle, and print style. You can change the values of fields in white; those in blue cannot be changed.

To see how the Properties palette works, lets try it on one of the lines you drew earlier in this chapter:

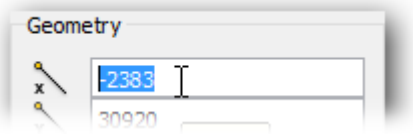
1. Select a line. Notice that the list of properties changes.
2. Use the Properties palette to change the line's color, as follows:
   a. In the General section, click the **LineColor** droplist.
   b. Choose **Red**. Notice that the line turns red.



Changing a line's color to red

3. Now use the palette to change the line's geometry, like this:
   a. In the Geometry section, click the **End X** text box.

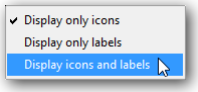
b. Erase the text.



Changing the location of a line's endpoint

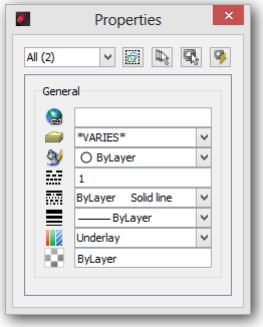
c. Enter a new number, such as **10**, and then press **Tab**. Notice that one end of the line changes its position.

TIP The palette identifies properties through icons and/or text. I find I prefer text over the default of icons-only. To change, right-click the palette and then choose **Display icons and labels:**



---

4. Pick another line. Notice that the palette reports two entities are selected but some properties are now *VARIES*. This means the lines have properties that different between them.

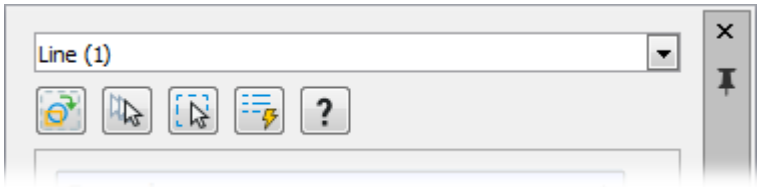


**VARIES* reported when properties are different among selected entities,*

5. Press **Esc** to un-select the two lines. The one line retains its red color.

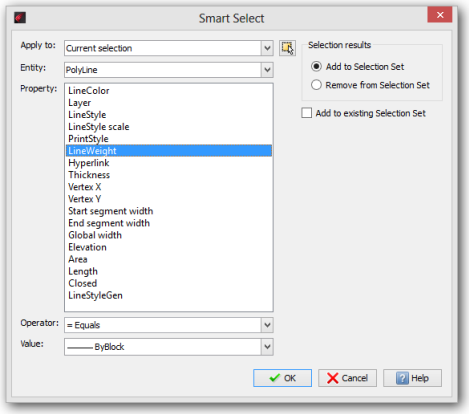
Properties Palette Toolbar

At the top of the palette is a mini toolbar. From left to right, these are:



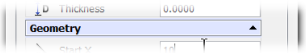
Toolbar in the Properties palette

Button	Meaning	Function
	Disable / enable group selection	Selects individual entities in a group, or else the entire group
	Add to selection set	Add entities to the current selection set
	Select entities	Replaces the current selection set
	Smart select	Displays a dialog box to select entities based on their properties, such as all red lines on layer "Electrical"
?	Help	Displays the Help window at Properties



Smart Select dialog box selects entities by properties

TIP When you find that the Properties palette displays too much information, you can hide sections by clicking the small arrowhead, such as the one shown below.

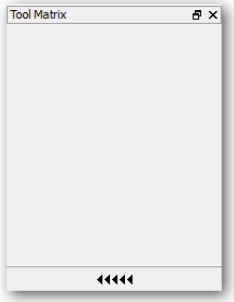


Click it a second time to again reveal the section.

TOOL MATRIX

Command: ToolMatrix
Alias: --
Menu bar: --
Ribbon: --
Shortcut: Right-click any toolbar and choose Tool Matrix
Toolbar: --

The Tool Matrix is a palette that you can customize with toolbars. You open the palette with the ToolMatrix command, and the first time you see it, it may be empty.

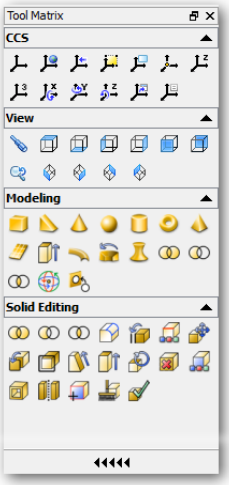


Tool Matrix initially empty

To fill it, drag toolbars into the Tool Matrix; drag toolbars by their ( ) handles. You cannot drag elements from the ribbon.

Dragging toolbars into the Tool Matrix

For the Tool Matrix shown below, I filled it with all the toolbars related to 3D modeling:



Tool Matrix filled with toolbars of command useful for 3D modeling

Here are some additional ways to control the Tool Matrix:

- To remove a toolbar from the Tool Matrix, drag its title bar away from the matrix.
- ▲ (Close) and ▼ (Open) open and close "toolbars" within the palette.
- ◀◀◀◀ (Minimize) collapses the Tool Matrix by making it narrower to free up space in the drawing area. When the Tool Matrix is collapsed, the toolbars reduce themselves to buttons with flyouts, as shown below.

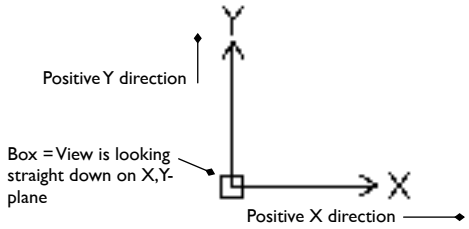
Minimize Tool Matrix shows just one button from each toolbar

▶▶▶▶ (Expand) opens up the palette.

TURNING OFF THE CS ICON

Command: CsIcon
Menu bar: View | Display | CCS Icon | On
Ribbon: --
Toolbar: --

The CS icon is located in the lower-left corner of the drawing area. *CS* is short for "coordinate system."



The parts of the CS icon

Its primary purpose is to help you draw in 3D by locating the origin and indicating the rotation of the x,y-plane.

The X arrow points in the direction of the positive x axis (to the right); the Y arrow points towards the positive y axis (upwards). The square around the origin means you are "looking down" along the z axis straight onto the x,y-plane.

The intersection of the x and y axes is always located at the origin of the drawing, wherever x=0 and y=0 happens to be. In this case, there is a small + at the intersection.

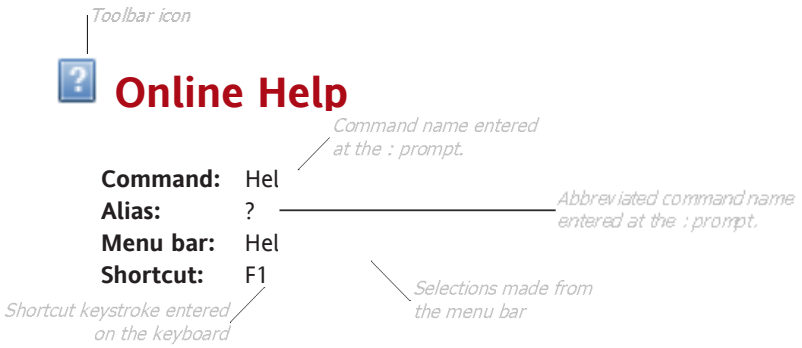
When the origin is not visible, then the CS icon is located in the lower right of the drawing area but the + is turned off.

The CS icon is not that useful in 2D drafting; in fact, I find it gets in the way, and so I recommend turning it off, like this:

1. From the **View** menu, select **Display**.
2. Then choose **CCS Icon | On**. The vertical bar (|) separates menu picks.

Notice that the CS icon disappears. To turn it back on, use the same procedure.

There is one situation where the CS icon is useful in 2D drafting: when you need to draw part of a building that is not orthogonal to another part of the structure, then you use the CCS command's Align to Entity option to rotate the entire drawing to match angle of the building.

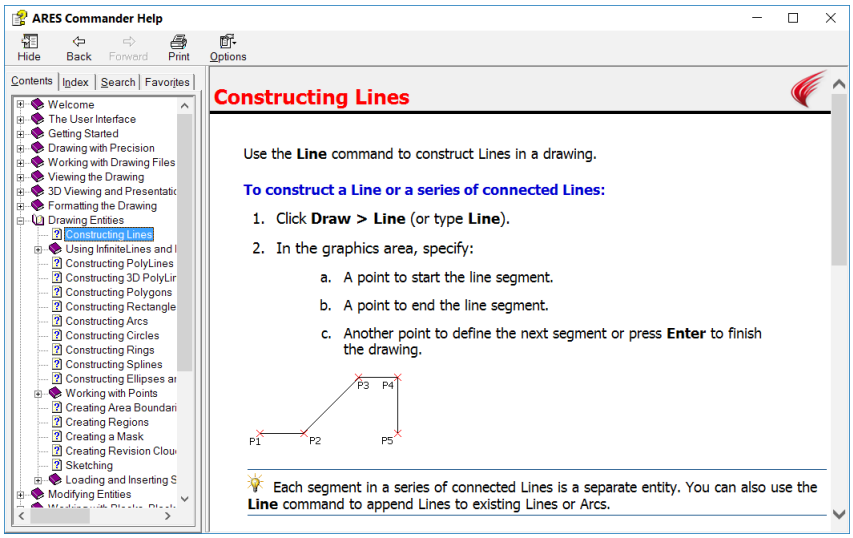


The program provides an easy way to access help: press function key F1 during a command. Here's an example using the Line command:

1. Type the **Line** command, then press the **F1** function key. Pressing F1 invokes context-sensitive help, as follows:

```
: line
Options: Segments or
Specify start point» (Press F1)
```

The program opens the Help window to display helpful information about the Line command.



Whenever you see underlined text, click it to display the definition of a word or to find out more about an option.

2. You can continue using the Line command with the Help window open:

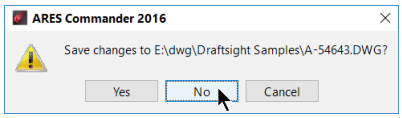
```
Resuming LINE command
Specify start point:
```

TIP To peruse specific subjects, select **Help** from the **Help** menu, and then select one of the tabs, such as **Contents** or **Search**.

Exiting The Program

Command: Exit
Alias: quit
Menu bar: File | Exit
Ribbon: Application button | Exit
Shortcut: Ctrl+q (Cmd+q on Macs)
Toolbar: --

To exit the program, use the **Exit** command. When the program asks if you want to save the drawing, click **No** this time.



Not saving changes to the drawing

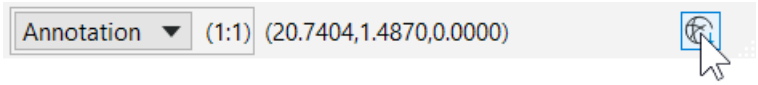
As alternatives, you can use any of the following methods to exit the program:

» Enter the **Alt+F4** or **Ctrl+q** shortcuts (**Cmd+q** on Macs)
» Select **Exit** from the **File** menu

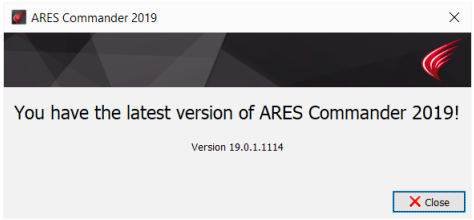
CHECKING FOR UPDATES

(NEW IN 2019) The program is regularly updated by Graebert with new functions and bug fixes. When you purchase a license to the software, you receive updates free for the first year. Following this, you can pay for updates through a maintenance or subscription fee.

To check if an update is available to you, click the Update button at the far right end of the status bar:



A dialog box appear, reporting whether the software is up to date, or whether an update is available to be downloaded.



Summary

This chapter provided you with an overview of the most important elements in the the program's user interface. This is important for finding your way around this CAD program. You will learn about the other user interface elements in the following chapters of this book. These include the layout tabs and the status bar.

In the next chapter, you learn about basic elements that make up every CAD drawings: entities and how they are drawn.

Notes

CHAPTER 03

Basic CAD Concepts

In This Chapter

- Learning how vectors define entities
- Specifying coordinates and selecting entities
- Inputting data with the mouse
- Editing with grips

Here you learn about the fundamental concepts to computer-aided design, such as *vectors*, *coordinates*, and *selection* *methods.* If you are experienced with CAD software, then feel free to skip this chapter, and move to the next one.

This chapter is crucial to understanding how computer-aided design software works, which is quite different from nearly all other kinds of software.

KEY TERMS IN THIS CHAPTER

Axis refers to references lines from which distances and angles are measured; drawings have three axes: x, y, z.

Cartesian coordinate specifies points in planes (and space) by their distance measured from the origin along the x and y axes (and optionally the z axis).

Coordinates refers to two or more numbers or angles that determine the positions of points in space relative to another reference, such as the origin or another point.

Cylindrical coordinate specifies points in space by their distance from the origin along the x and z axes; the a angle is measured counterclockwise from the positive x axis.

Direct distance entry specifies distances by moving the mouse and then entering a distance.

Grips refers to small squares that permit direct editing of entities.

Negative distance refers to distances measured negatively from the origin.

Negative angle refers to an angle measured clockwise from the positive x axis**.**

Origin refers to the coordinate center of the drawing, where x=0, y=0, and z= 0.

Point filter allows mixed input of coordinates by pointing in the drawing and entering values at the keyboard.

Polar coordinate specifies points in space by their distance from the origin along the x axis; the y distance is specified by the angle measured counterclockwise from the positive x axis.

Positive angle refers to an angle measured counterclockwise from the positive x axis**.**

Relative coordinate refers to points measured relative to another point.

Spherical coordinate specifies points in space by their distance from the origin along the x, y, and z axes; the y distance is specified by the angle measured counterclockwise from the positive x axis.

Vector refers to entities drawn by the program, defined by endpoints, radii, and additional geometric functions.

ABBREVIATIONS

@	(pronounced "at") indicates relative coordinates.
<	("angle bracket" or "less than") indicates angles.
-	("negative sign" or "dash") indicates negative values.
.	("point") indicates point filters.
:	("colon") indicates the command prompt.
x,y	indicates 2D Cartesian coordinates.
x,y,z	indicates 3D Cartesian coordinates.
@x,y	indicates relative 2D Cartesian coordinates.
d<a	indicates polar coordinates (distance, angle).
@r<a	indicates relative polar coordinates
x<a,z	indicates cylindrical coordinates
d<a<A	indicates spherical coordinates (distance, angle1, angle2).

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- Selection highlighting shows entities not yet selected.

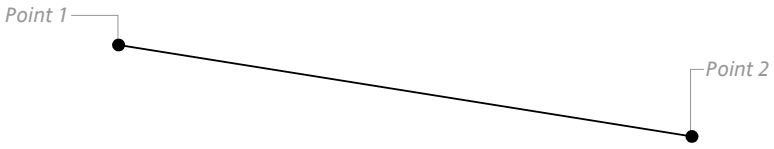
Vector Entities

All CAD software works with *vector entities*. By vectors we mean entities that are defined by their geometric features. Drawings consist of vector entities — lines, circles, arcs, text, dimensions, and so on. You create and manipulate them by inputting commands and specifying coordinates.

It may seem to be curious point, but you never actually draw the entities themselves; instead, you specify their geometric features, and then the program fills them in. For instance, instead of drawing a line, you specify only the two endpoints of the line, like this:

```
: line
Options: Segments or
Specify start point» (Indicate the starting point of the line, point #1.)
Options: Segments, Undo, Enter to exit or
Specify next point» (Indicate the end of the line at point #2.)
```

The program then fills in the segment between the two points, as illustrated below.



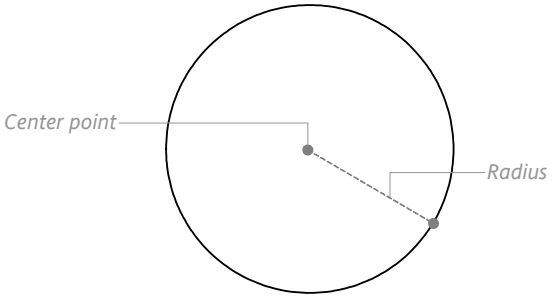
Lines are defined by two end points

The same technique applies to all other entities. The program draws them after you specify geometric features like center points, widths, and radii.

This is how you draw the most common entities in CAD:

Circles are drawn after you indicate the location of the center point and the length of the radius.

```
: circle
Options: 3Point, 2Point, Ttr, TTT, Enter to exit or
Specify center point» (Pick point 1 at the center of the circle.)
Options: Diameter or
Specify radius» (Pick point 2 to specify the size of the circle.)
```



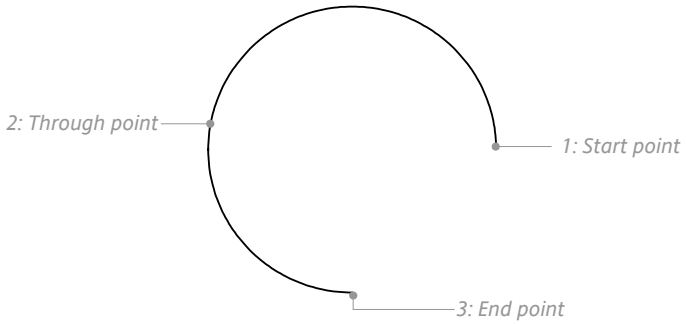
Circles are defined by the center point and the radius

TIPS Usually, there is more than one way to draw an entity. For instance, you can specify the radius or diameter for circles. Or, you can specify two or three points on the circumference. You can also draw them so that they touch other entities tangentially.

The program highlights the names of options in the Command bar with blue text.

Arcs are drawn after you indicate two endpoints and one other point, such as the length of the arc;s radius or indicate its center point. (Of all CAD entities, arcs have just about the largest number of options for drawing them.)

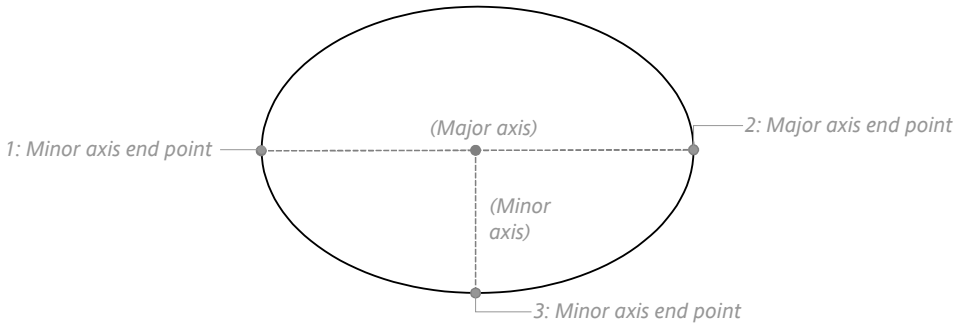
```
: arc
Options: Center, Append, Enter to continue from last point or
Specify start point» (Pick point #1 to start the arc)
Options: Center, End or
Specify through point» (Pick point #2 on the arc's curve)
Specify end point» (Pick point #3 to end the arc)
```



Arcs are defined by the start and end points, and a through point

Ellipses are defined by two axes, one that is wider, called the "major axis," and another that is narrower (the "minor" axis). The program draws the ellipse after you indicate the location and length of each axis.

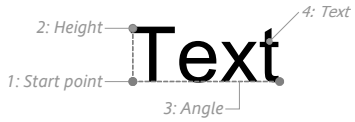
```
: ellipse
Options: Elliptical arc, Center or
Specify axis start point» (Pick point #1 to start the major axis)
Specify axis end point» (Pick point #2 to define the length and angle of the major
axis)
Options: Rotation or
Specify other axis end point» (Pick point #3 to define the length of the minor
axis)
```



Ellipses are defined by the lengths of major and minor axes

Text is one of the few entities that you actually create, after indicating its starting point, height, and angle. The program uses the -SimpleNote command to place text in drawings.

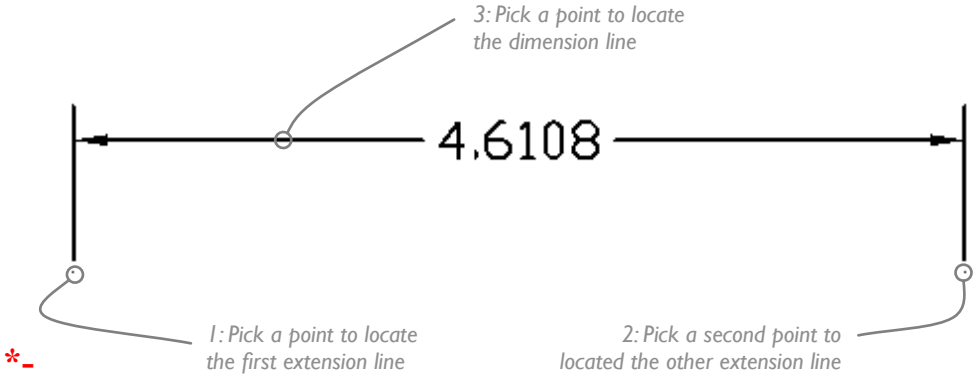
```
: -simplenote
Options: Justify, textSTyle or
Specify start position» (Pick point #1)
Default: 2.5
Specify height» (Specify the height at #2)
Default: 0
Specify text angle» (Specify the angle at #3)
Specify text» (Enter the text at #4)
Specify text» (Press Enter to end the command)
```



The position of text is determined by the starting point, height, and angle

Dimension are drawn after you indicate the location of the two extension lines, the dimension line, and the text. The LinearDimension command draws horizontal, vertical, and rotated dimensions; other commands in the program draw additional types of dimensions.

```
: lineardimenion
Default: Entity
Options: Entity or
Specify first extension line position» (Pick point #1 to start the dimension)
Specify second extension line position» (Pick point #2 to specify the dimension's
length)
Options: Angle, Horizontal, Note, Rotated, Text, Vertical or
Specify dimension line position» (Pick point #3 to locate the dimension line)
Dimension Text : 4.6108
```



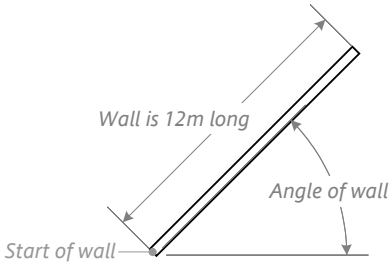
Left: *Using dimensions to specify an entity*

In addition to the basic entities described above, the program draws many others. You can see the complete list in the Draw, Dimension, and Solids menus.

ALTERNATIVE WAYS TO SPECIFY ENTITIES

While you can certainly draw lines by specifying two endpoints, this may not be the most convenient method. For example, in drawing a line that represents a wall, you often know the length, and where it starts, but not where the other end is located.

The program provides multiple methods of defining entities. To draw the wall, for instance, you can specify its starting point, its length, and its angle to other walls. As before, you don't draw the wall itself, but specify its parameters. See the figure below; note that the angle is measured from the x axis.



Drawing a line representing a wall, by its start, angle, and length

The same is true for nearly all entities created by the program. Often, there are two or more ways to draw each of them. The program provides, for example, more than a dozen ways to draw arcs.

Specifying Coordinates

To draw entities, you need to specify the coordinates (locations) of geometric features. The program provides two methods of specifying geometric features: through the keyboard, and with a mouse (cursor). The keyboard is used to enter specific numbers, distances, and angles; the mouse is used to show locations, distances, and angles in drawings.

KEYBOARD INPUT

You can use the keyboard to enter the names of commands and options, as well as to specify exact x,y coordinates, distances, angles, and amounts. Let's take the Line command as the example.

```
: line
Options: Segments, Enter to continue from last point or
Specify start point»
```

At the 'Specify start point' prompt you can do any of the following at the keyboard:

- Press **Enter** to continue the line from the last point entered in the drawing.
  `Specify start point» (Press Enter)`

  The program then starts the line at the end of the last line or arc that was drawn earlier.

- Press **S** and then **Enter** to specify the Segments option.
  `Specify start point» s (...and then press Enter)`

  The program presents the prompts for drawing a single line segment.

- Enter x,y coordinates, such as **1,2**, and then press **Enter**.
  `Specify start point» 1,2 (...and then press Enter)`

  The program starts drawing the line at a point defined by x=1 and y=2.

TIP Another way to enter commands and coordinates is through **Quick Input** See chapter 2 for how to use this input system.

HOW COORDINATES ARE SPECIFIED

Although entering x,y coordinates is very common in CAD programs, it is not the only way; the program supports a variety of coordinates types.

It can be tricky to remember, but positive angles are measured *counterclockwise* from the x axis, although the direction can be reversed with the Units command, as can the location from which they are measured.

The program uses commas (,) to separate distances, and the angle bracket (<) to signify angles. It uses additional symbols to modify coordinates:

» Negative distances and angles are shown with a dash (-). Negative angles are measured clockwise from the x axis, and are indicated by the dash prefix.

```
Specify start point» -1,-2<-45
```

» **Relative** coordinates use the at symbol (@). Most coordinates are relative to the origin of the drawing (0,0) and so are called "absolute coordinates." By placing the @ symbol in front of the coordinate, you measure distances and angles relative to the last point picked in the drawing.

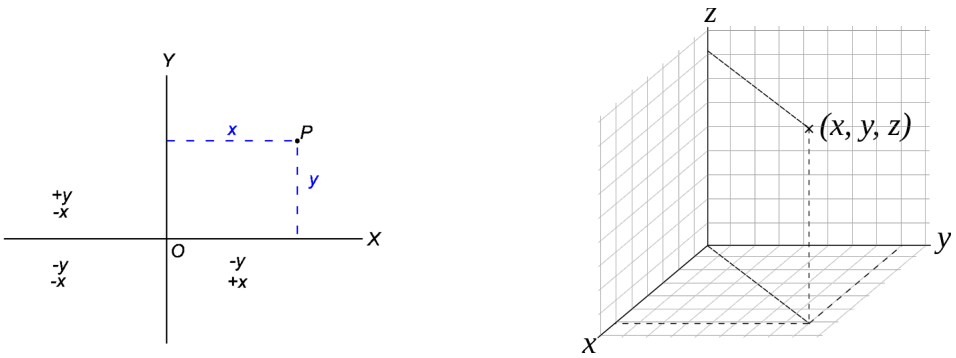
```
Specify start point» @1<45
End point: @7,5
```

2D Cartesian Coordinates

x,y specify two-dimensional coordinates used for 2D drafting. (*Cartesian* is named after the French mathematician, Rene Descartes.) Note that a comma (,) is required to separate the x and y values.

```
Specify start point» 1,2
```

The z coordinate is fixed by the Elevation system variable, and is usually set to 0.



***Left**: Specifying 2D Cartesian coordinates by two distances;*
***right**:... by 3D ones by three distances image source Wikipedia*

TIP You can enter coordinates with imperial or metric units. Imperial units use **'** for feet and **"** for inches, such as **1'2",5".** Metric units use no symbols, just the numbers.

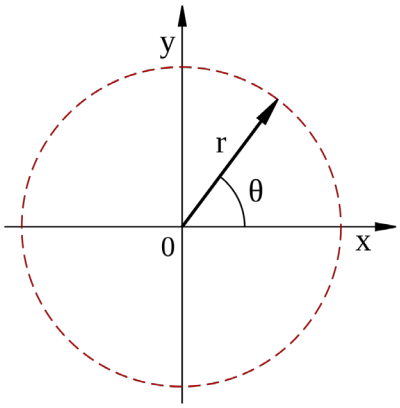
3D Cartesian Coordinates

x,y,z specify three-dimensional coordinates, also known as Cartesian coordinates.

```
Specify start point» 1,2,3
```

Polar Coordinates

d<a specify polar coordinates, which describe the distance and angle from the origin (0,0). Think of being at the North Pole, and finding any point on the earth by a distance (from the North Pole) and an angle around the North Pole.



Specifying polar coordinates by distance (radius) and angle; image source Wikipedia

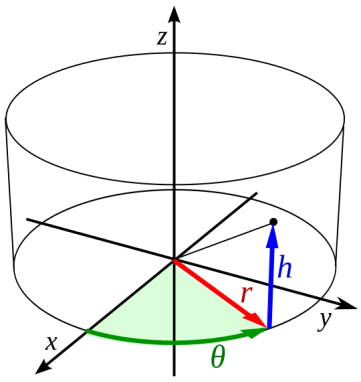
```
Specify start point» 23<45
```

The angle bracket (**<**) prefix indicates the angle.

TIP You can specify the angle in degrees, radians, grads, or surveyor units. Radians are indicated by **r**, grads by **g**, and surveyor units use **N** and **E**; degrees use no symbols. For example, 360 (degrees), 2.17r, 400g, and 24N45E.

Cylindrical Coordinates

x<a,z specify cylindrical coordinates, which combine an x-coordinate with an angle and a z-height.



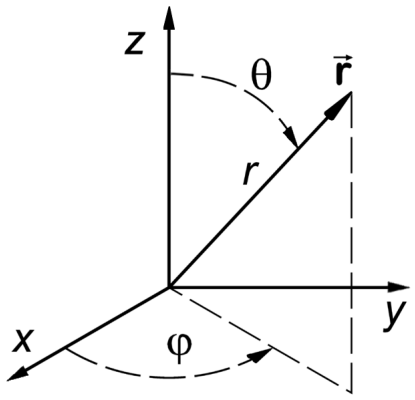
Specifying cylindrical coordinates by distance (radius), height, and angle; image source Wikipedia

Think of being inside a pop can: the **x** measures the radius of the can, the **z** measures the height of the can, and the **a** measures the angle around the pop can.

```
Specify start point» 1<45,1
```

Spherical Coordinates

d<a<A specify spherical coordinates, the 3D version of polar coordinates. Here you specify a distance and two angles.



Specifying spherical coordinates by distance (radius) and two angles; image source Wikipedia

Think of being in the center of the Earth: the **d** measures the distance from the center, the **a** measures the angle left-and-right around the equator (x, y plane), while the **A** measures the angle up-and-down along the International Data Line (y, z plane).

```
Specify start point» 1<45<90
```

POINT FILTERS

Finally, you can enter partial coordinates by using *point filters*. The word "point" refers to the points being placed in the drawing , while "filter" means that the coordinates are being provided partially.

Typically, you enter the points in two steps. First, you enter the x or y or some other coordinate, and then the missing portion, as illustrated by the table below:

Point Filter	First specify...	...and then specify
.x	x coordinate	y,z coordinates
.y	y coordinate	x,z coordinates
.z	z coordinate	x,y coordinates
.xy	x,y coordinates	z coordinate
.xz	x,z coordinates	y coordinate
.yz	y,z coordinates	x coordinate

When you enter a point filter, the program holds the following conversation with you:

```
Specify start point» .xy
of (Pick a point)
(need Z) 4
```

NAVIGATING WITH THE MOUSE

Like almost all other desktop software, the program is best navigated by mouse. Buttons perform these functions.

Button #	Action
1	Picks entities
2	Displays entity grips menu
3	Displays entity snap menu
Shift+2	Displays entity snap menu
Shift+3	Orbits drawing transparently

Roller wheel (#3)	
Roll forwards	Zooms into the drawing
Roll backwards	Zooms out of the drawing
Hold down	Pans the drawing

Double-clicking an entity with button #1 edits the entity through the Properties palette or an entity-specific editing command. The function of all mouse buttons can be changed through the **Customize** command.

Photo credit: The Motion Report

Photo credit: 3Dconnexion

3D MICE

The Windows version also supports 3D mice from 3dconnexion, which interactively roll, twist, zoom, and perform other view changes useful in three-dimensional modeling. When a 3D mouse is plugged into the computer, the program adds the **3D Mouse** item to the menu bar automatically. The buttons of these mice are customized through the **3DMouseButtons** and **3DMouseOptions** commands.

QUICK INPUT ENTRY

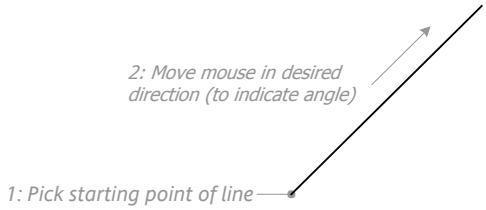
the program has a "hidden" form of coordinate entry known as "quick input." In this method, you don't enter coordinates; instead, you move the cursor in a specific direction, and then type in the distance:

- Move the cursor in a specific direction — indicates the angle
- Type the distance — indicates the distance to the next end point
- Press **Tab** to switch between entering the angle and the distance from the keyboard

This is an interactive form of relative polar coordinates (@d<a).

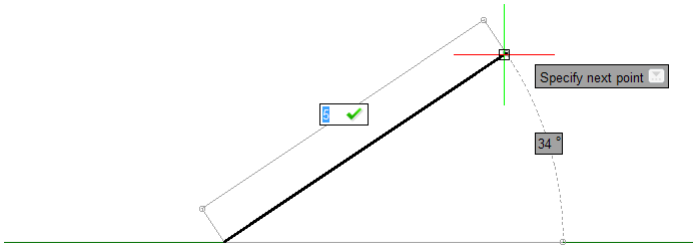
To turn on direct distance entry, click the **QInput** button on the status bar. Here's how to use quick input during the Line command:

```
Specify start point» (Move mouse in any direction, and then enter a distance:) 5
```



Directly entering the angle and distance

As you move the cursor, on-screen assistants show you the distance and angle from the x axis. In the figure below, these are 5 units long and 34 degrees. To enter precise values, press the **Tab** key to switch between distance and angle measurements.



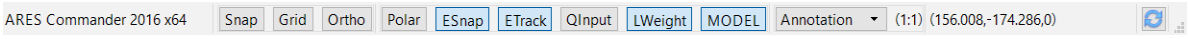
Quick input guiding the distance and angle

Mouse Input

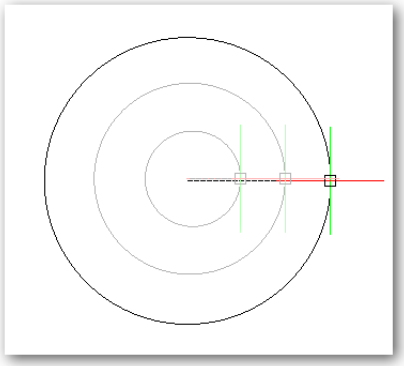
You can use the cursor to show the program locations in drawings; the mouse (or digitizer) controls the cursor. The problem with using the mouse is that it is not particularly accurate by itself. You would find it difficult to draw, for example, a 30mm-diameter circle using just the mouse.

For this reason, the program provides several aids for more accurately drawing with the mouse and cursor.

Status Bar shows the distance and angle from the last pick point. As you draw, you could keep an eye on the status bar to see the size of the entity being drawn. (Notice the x,y,z coordinates at the right end of the status bar.) But the drawback is that watching the status bar is too slow and imprecise for production use.



Snap specifies the cursor resolution. For instance, when you set the snap distance to 10 units, then mouse movements in the drawing area are accurate to the nearest ten units. Now you can draw that 30mm-circle easily with the mouse:

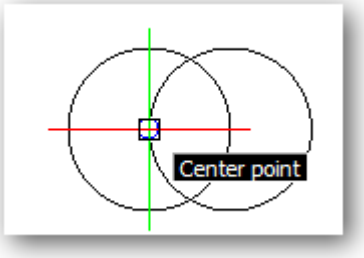


Size of circle changed by increments of the snap distance

(Snap does not affect the cursor outside of the drawing area.)

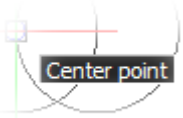
Grid is an array of evenly-spaced dots. The snap distance is invisible, however, so many users also turn on the grid to match the snap distance.

Entity Snap causes the cursor to snap to geometric features of entities, such as their end points, mid points, or tangent points. This is useful when you want to draw from one geometric feature to another. For example, you could locate the circle's radius at the precise center of another circle, as illustrated below.



Entity snap finding the center of a circle

Entity snaps are called "esnaps" for short. To help you further, the program displays tooltips to identify geometric features, such as "Center point," as illustrated below.



Tooltip identifying the name of the entity snap

SELECTING ENTITIES

When it comes to editing drawings, you need a way to select the entities to be edited. Selection tells the program which group of entities you need to modify. In most cases, you must use the mouse to select entities; in a few cases, you use the keyboard instead.

There are two ways to select entities:

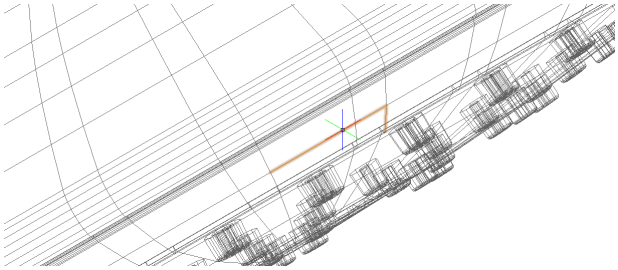
- *Before* starting an editing command (called "grips editing").
- *After* starting an editing command (called "entity selection").

You will probably find both methods useful, but for different kinds of editing operations.

GRIPS EDITING

Grips editing is the kind of editing you may be familiar with from other drawing programs. It starts when no command is active, and you select one or more entities by picking them with the cursor.

(new in 2019) When you move your cursor over an entity, the program highlights it in orange. This way you know before you pick the entity, which one you want, as shown below:



Orange highlight show pre-selected entity

NAVIGATING BY TOUCHSCREEN

Windows has long supported touch screens — remember 1990s Windows for Pen Computing? Touching was done with a pen. Not until Windows 8 did Microsoft support screens that recognized human touches.

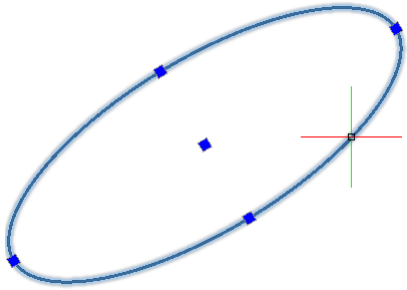
The program does not specifically support touch screens, because support comes from the operating system. The following touch motions work:

Touch Action	Reaction
Tap	Selects UI elements, picks entities (like clicking left mouse button)
Hold	Displays context menu (like clicking right mouse button)
Drag and tap	Windowed selection of entities (like click and drag with a mouse)
Double-tap	Selects file, opens folder (like double-clicking with a mouse)

Holding the finger down on the screen acts like pressing the right mouse button. In this case, Windows displays a square to show that it has registered the action. When we left our finger, the context menu appears.

There were, however, other common touches that do not work, such as pinching two fingers to zoom in or dragging two fingers to pan.

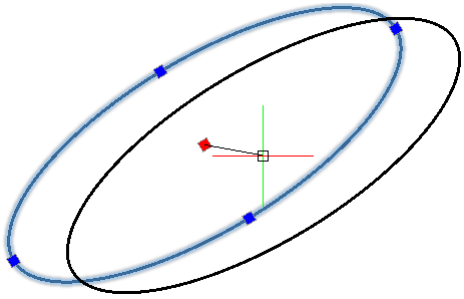
Notice that the entities selected change to a blue glow, as shown below. (The blue glow is **NEW IN 2019**.) The program responds in this way to let you know which entity you picked.



Blue squares indicate the locations of editing grips

In addition to the blue glow, one or more blue squares appear. These squares are the *grips*. (Blue grips are also known as "cold grips.") Their purpose is to show you where grips are located on the entity.

Move the cursor over a grip, and then click on it. Notice that the grip turns red.



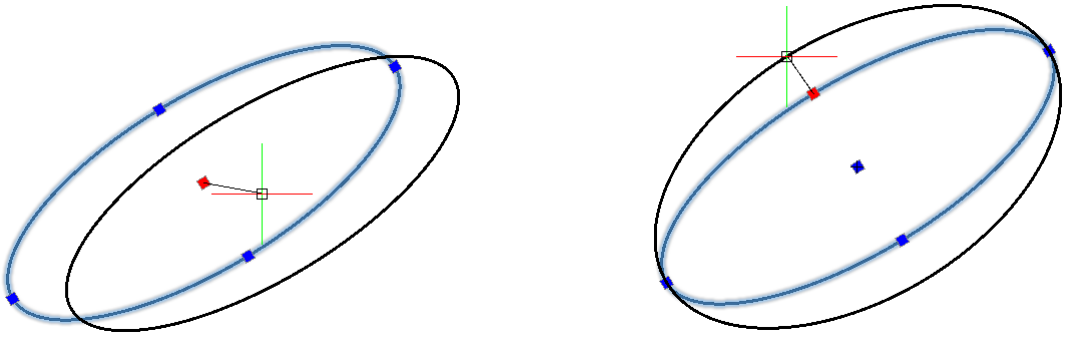
Red square indicating the "hot" grip, which edits the entity

This is known as a "hot grip," the purpose of which is to edit the entity. The editing function varies, depending on which grip you chose; most often, a hot grip moves or stretches the entity, as described next.

Moving and Stretching with Grips

Drag the hot grip. Notice that the action edits the entity in some way. The exact type of editing depends on the location of the grip. In general:

- **Centrally**-located grips tend to move the entity.
- **Peripherally**-located grips tend to stretch the entity.



Left: *Using the hot center grip to move the ellipse* ***Right:*** *Using the hot edge grip to stretch the ellipse*

Additional editing options are available in the command window:

```
STRETCH
Options: Base point, Copy, Undo, eXit or
Stretch point»
```

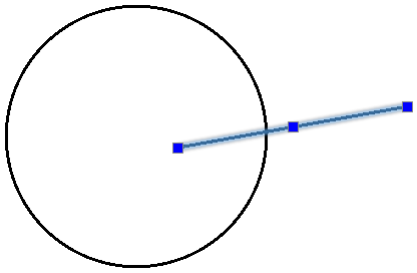
- **Base point** relocates the base point (the "start point").
- **Copy** copies the entity, instead of moving it.
- **Undo** undoes the last action.
- **eXit** exits grips editing. You can also press **Esc** to exit.

Erasing with Grips

To erase selected entities, press the **Del** or **Back** key (**Delete** on Macs) on the keyboard.

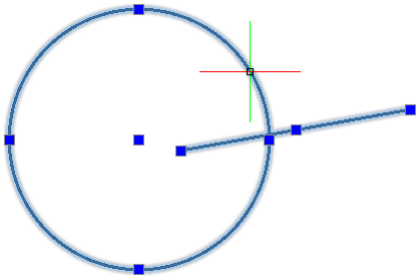
SELECTING MULTIPLE ENTITIES

You select a single entity by picking it with the cursor. Ensure part of the entity is within the cursor's pickbox.



Picking an entity with the cursor

To select additional entities, just keep right on picking 'em:



Picking a second entity (blue squares show the other entities that were selected)

Selecting Multiple Entities by Rectangle

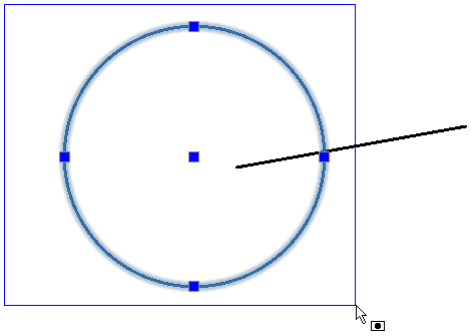
After a while, picking entities one-by-one becomes tedious, and so the program provides ways to pick groups of entities. One method is the selection rectangle. It selects entities within an area specified by a rectangle, which is created like this:

1. Pick in a blank point in the drawing away from any entities.

2. Form a rectangle by moving the cursor diagonally. The program reacts differently, depending on whether you move the cursor to the right — or the left:

   Right — moving the cursor right forms a *windowed* selection. The program selects all entities that fall *entirely within* the selection rectangle. It ignores entities that cross the rectangle or fall outside the rectangle. The program colors the selection rectangle blue.

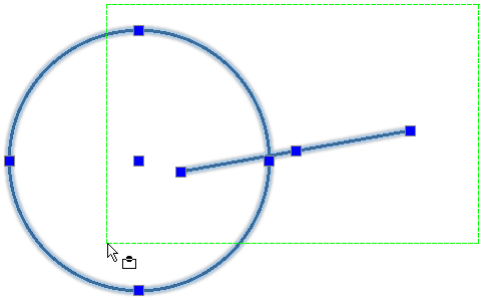
   Entities selected by windows selection mode are highlighted by the program after you pick the second point of the selection rectangle. In the figure below, only the circle is selected, because part of the line lies outside the selection window (blue rectangle).



Selecting multiple entities entirely within the rectangle

   Left — moving the cursor left forms a *crossing selection*. The program selects all entities within the selection rectangle, as well as those crossing the rectangle. It colors the selection rectangle green and makes it dashed.

   In the figure below, both the circle and line are selected, because they are either inside or crossing the selection window (dashed green rectangle).



Selecting multiple entities within and crossing the rectangle

Whichever one you use depends on which selection mode works better for you in a editing situation.

SELECTING ENTITIES BY LOCATION

But there are more selection modes! Selection areas need not be rectangles; they can also be polygons and fences. These other selection modes become useful when editing commands ask you to specify entities.

For example, the Copy command prompts you, as follows:

```
: copy
Specify entities» (Enter the name of a selection type or mode)
```

At the 'Specify entities' prompt, you can pick entities with the mouse, or enter one of the following options. I've highlighted each option's abbreviation with underlined, boldfaced text.

Selection Option	Meaning	Color
ALL	Selects all non-frozen entities in the drawing	...
AUto	Selects an entity when picked directly; picking a blank spot starts Window or Crossing mode	Blue, Green
BOX	Starts Windows and Crossing mode, depending on how the cursor is moved: Right to left = Crossing mode Left to right = Window mode	 Green Blue
Crossing	Selects entities contained within or crossing the boundary of a rectangular selection area	Green
CPolygon	Selects entities contained within or crossing the boundary of a polygonal selection area	Green
Fence	Selects entities crossing a series of line segments	Black
Last	Selects the entity most recently added to the drawing	...
Previous	Selects entities included in the previous selection set	...
Window	Selects entities contained entirely within a rectangular selection window	Blue
WPolygon	Selects entities contained entirely within a polygon selection window	Blue

Modes	
Add	Adds one or more entities to the selection set; used after the Remove option
EGroup	Selects entities of a named group (created by the EntityGroup command)
Remove	Removes one or more entities from the selection set
Multiple	Selects multiple entities without highlighting them
SIngle	Selects one entity, then carries on with the command; the 'Specify entities' prompt doesn't return
Undo	Undoes the last selection

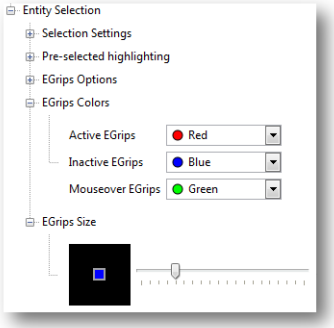
You can create a selection set with the **Select** command. Then, during an editing command use the Previous option to employ the stored set, like this:

```
: select
Specify entities» (Choose one or more entities)

: copy
Specify entities» p
```

TIPS To see a list of selection options, enter a question mark at the 'Specify entities' prompt, like this:

```
: copy
Specify entities» ?
«Invalid selection» Specify a point or Window, Last, Crossing,
BOX, ALL, Fence, WPolygon, CPolygon, EGroup, Add, Remove,
Multiple, Previous, Undo, AUto, SIngle
```

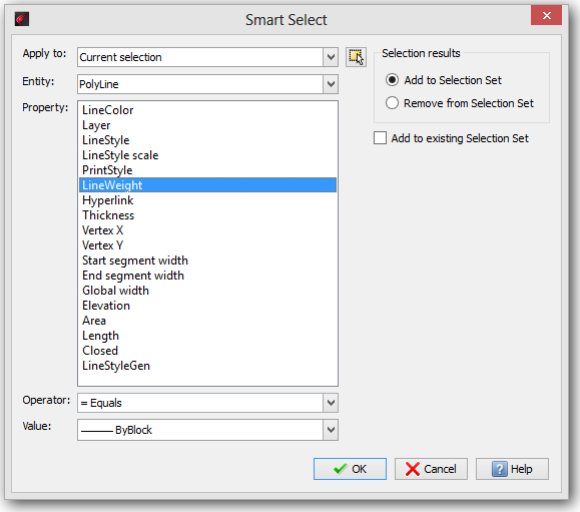
To change the size and color of grips, enter the **DraftingOptions** command, and then expand the **Entity Selection** node (illustrated below). Sizes are changed with the **EGrips Size** node, colors with the **EGrips Colors** one.



SELECTING ENTITIES BY PROPERTIES

The selection options listed in the table above are meant for selecting entities based on their location in the drawing. A different method is to select them by their properties, such as all entities colored red or with a specific hyperlink.

The **SmartSelect** command displays a dialog box that lets you choose entities based on any of their properties. The most common properties include color, Layer name, line style, line style scale, print style, lineweight, or hyperlink.



Selecting entities by common properties

(This dialog box is also available through the Properties palette by clicking the Smart Select icon.)

You can mix keyboard and mouse entries. For instance, the prompts below show a line drawn using point filters, esnap modes, mouse/cursor picks, and relative coordinates:

```
: line
Specify start point» .xy
of: mid
of (Pick an entity with the mouse/cursor)
(need z) 3
Specify next point» @1<-45
```

Modifying Properties of Selected Entities

Once entities are selected, their properties can be changed, as described in later chapters. Properties are things like color, assigned layer, and line type.

AutoCAD Command Compatibility

The program's command names are compatible with those in AutoCAD and IntelliCAD, for the most part. Some command names are identical, such as Line, Copy, and Stretch.

Others are different, such as Erase (Delete, in this program), PEdit (EditPolyline), and DimLinear (LinearDimension).

The program ensures compatibility through the use of aliases. Appendix C, "Command Aliases," lists commands and corresponding aliases, and boldfaces those that are identical with AutoCAD.

Summary

Here you learned concepts crucial to understanding and using CAD well. With these ideas under your belt, you are ready to begin drawing — which you start doing with the next chapter.

Notes

PART II

2D Architectural Drafting

Notes

CHAPTER 04

Starting New Drawings

In This Chapter

- Preparing drawings for the first time
- Understanding how layers organize drawings
- Saving drawings
- Learning the importance of automatic backups

Here you learn how to prepare the program for new drawings. By the end of this chapter, you will know how to save your work to disk and to exit the program.

Before You Begin

To learn how to use the program, you work with a drawing of an office floor plan. The example used for the 2D drafting portion of this book is illustrated on the next page.

PREPARING FOR DRAWING

Before creating any new drawing, you must prepare it by carrying out the following tasks:

i. Name the drawing by saving it
ii. Specify the units of measurement
iii. Set the spacings of the snap and the grid
iv. Indicate the bounds of the drawing
v. Create layers by naming them

KEY TERMS IN THIS CHAPTER

File name refers to names used for drawing files and other documents.

Grid provides a visual guide consisting of an array of evenly-space dots.

Layer organizes drawings by segregating in-common entities.

Bounds specify the nominal limits of drawings and constrains the range of grid marks.

Snap constrains cursor movement to discrete distances.

Unit specifies forms of linear and angular measurements, such as metric, architectural, and engineering.

Zoom enlarges and reduces the visual size of drawings.

COMMANDS

Command	Windows & Linux Shortcuts	Mac Shortcuts	Menu Selections
DrawingBounds	bounds	bounds	Format | Drawing Bounds
Grid	F7 *or* Ctrl+g	F7 *or* Cmd+g	...
LayerDialog	...	...	...
New	Ctrl+n	Cmd+n	File | New
Options	op	op or Cmd+,	Tools | Options Application | Preferences (on Mac)
Save	Ctrl+s	Cmd+s	File | Save
Snap	F9 *or* Ctrl+b	F9 *or* Cmd+b	...
UnitSystem	un, units	un, units	Format | Unit System
Zoom	z	z	View | Zoom

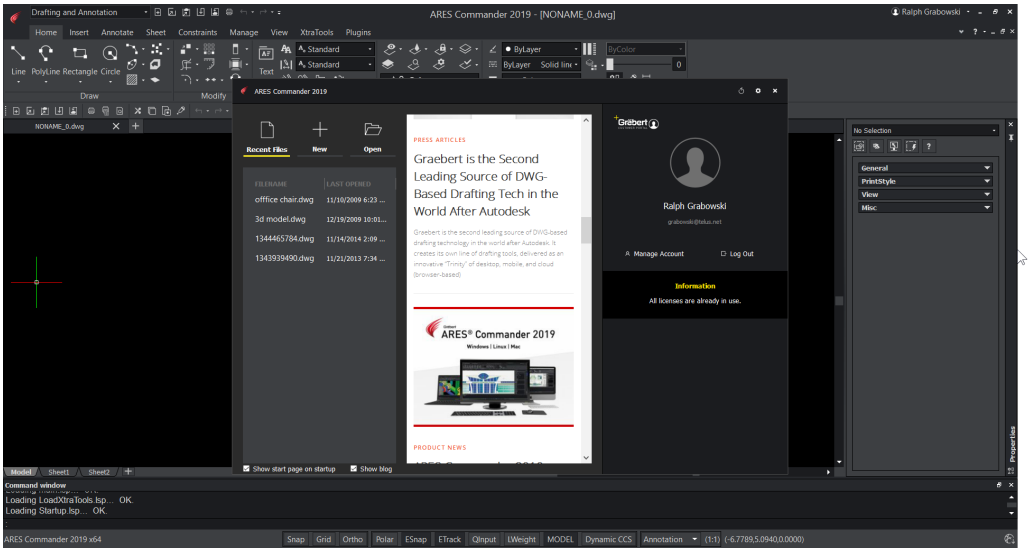
WHAT'S NEW IN 2019 FOR THIS CHAPTER

- **LayerDialog** command replaces the Layer command for displaying the layers dialog box.
- **Layer** command displays the new layers palette.
- **MergeLayers** and **-MergeLayers** commands
- **DrawingRecovery** and **HideDrawingRecovery** commands display and hide the Drawing Recovery Manager palette to recover drawings after unexpected system failures; the palette opens automatically afgter restarting the program following a crash. To recover a drawing, double-click its name.
- **SignFile** command adds a digital signature to the current drawing file, but works only with valid digital IDs, which are optional and may well not exist on your computer. Digital signatures ensure that the drawing has not been modified by unauthorized persons.
- **VerifySignature** command checks digital signatures for authentication.
- **RemoveSignature** command removes digital signatures from drawings.

STARTING WITH A NEW DRAWING

Start the program by one of the methods described in the previous chapter, such as double-clicking the icon (shown at left) on the desktop.

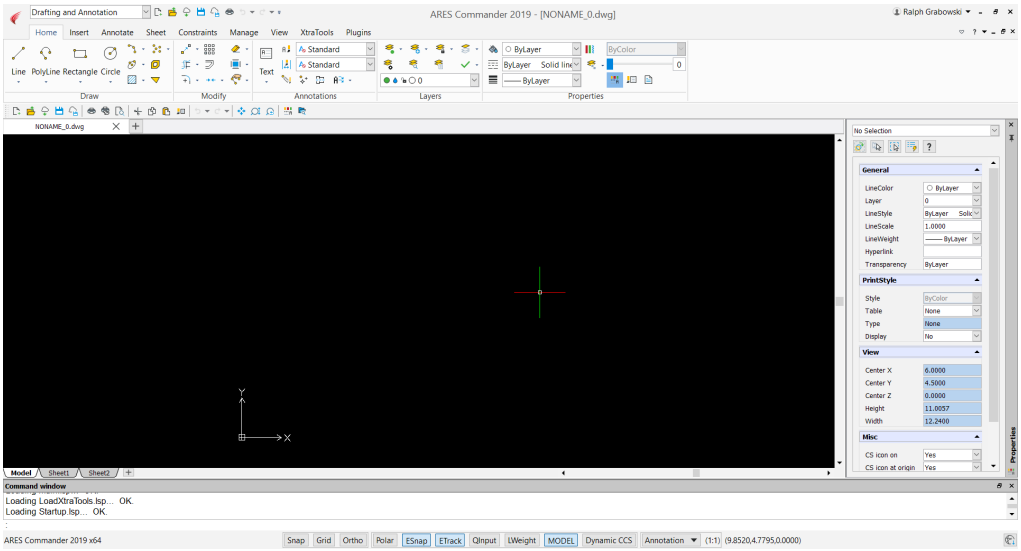
Notice that the program starts with what looks like a largely blank screen. This is what the program looks like with a new drawing in Windows 10. It looks similar in other releases of Windows, while the Mac OS and Linux show toolbars instead of the ribbon.



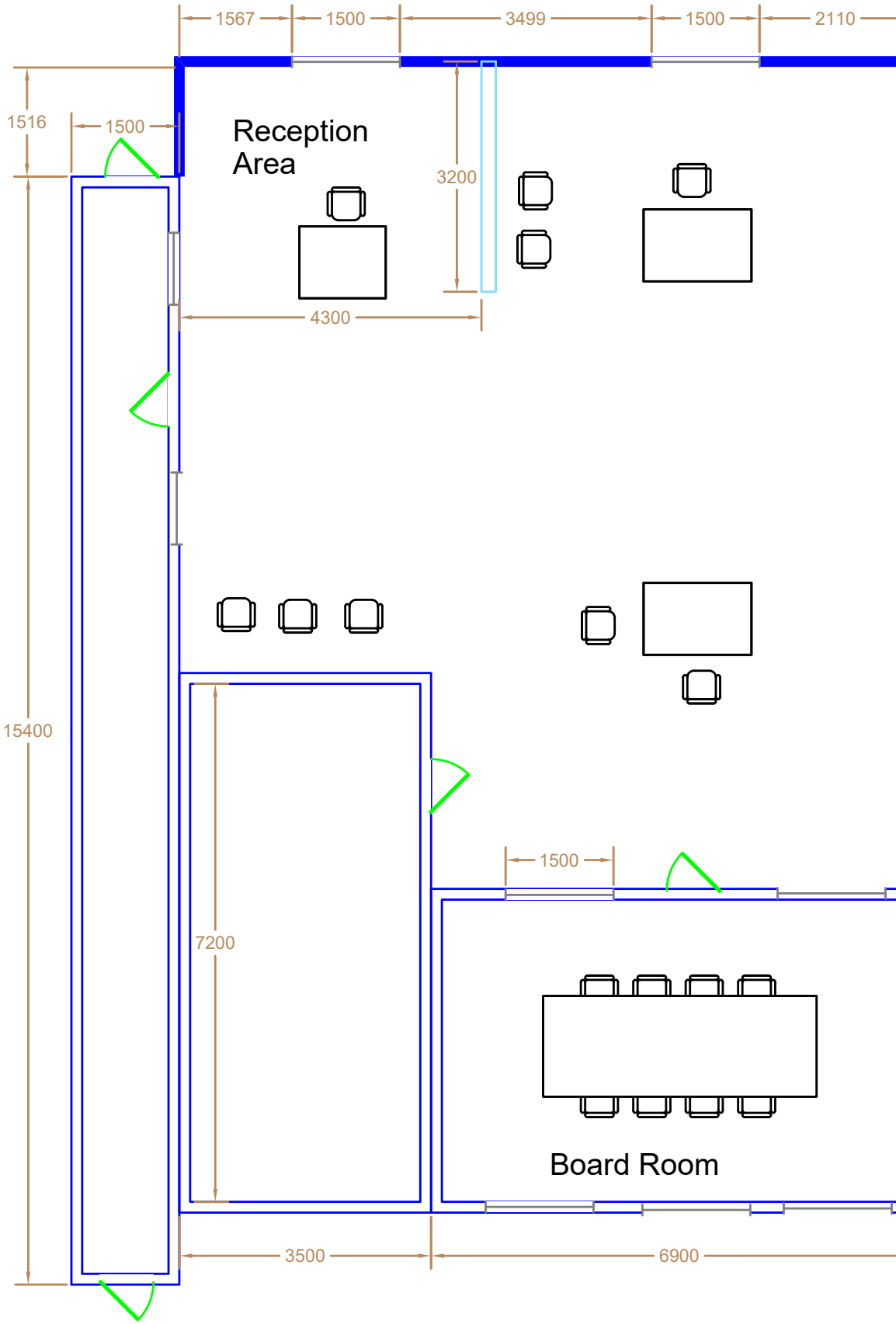
The program starting with a new, blank drawing

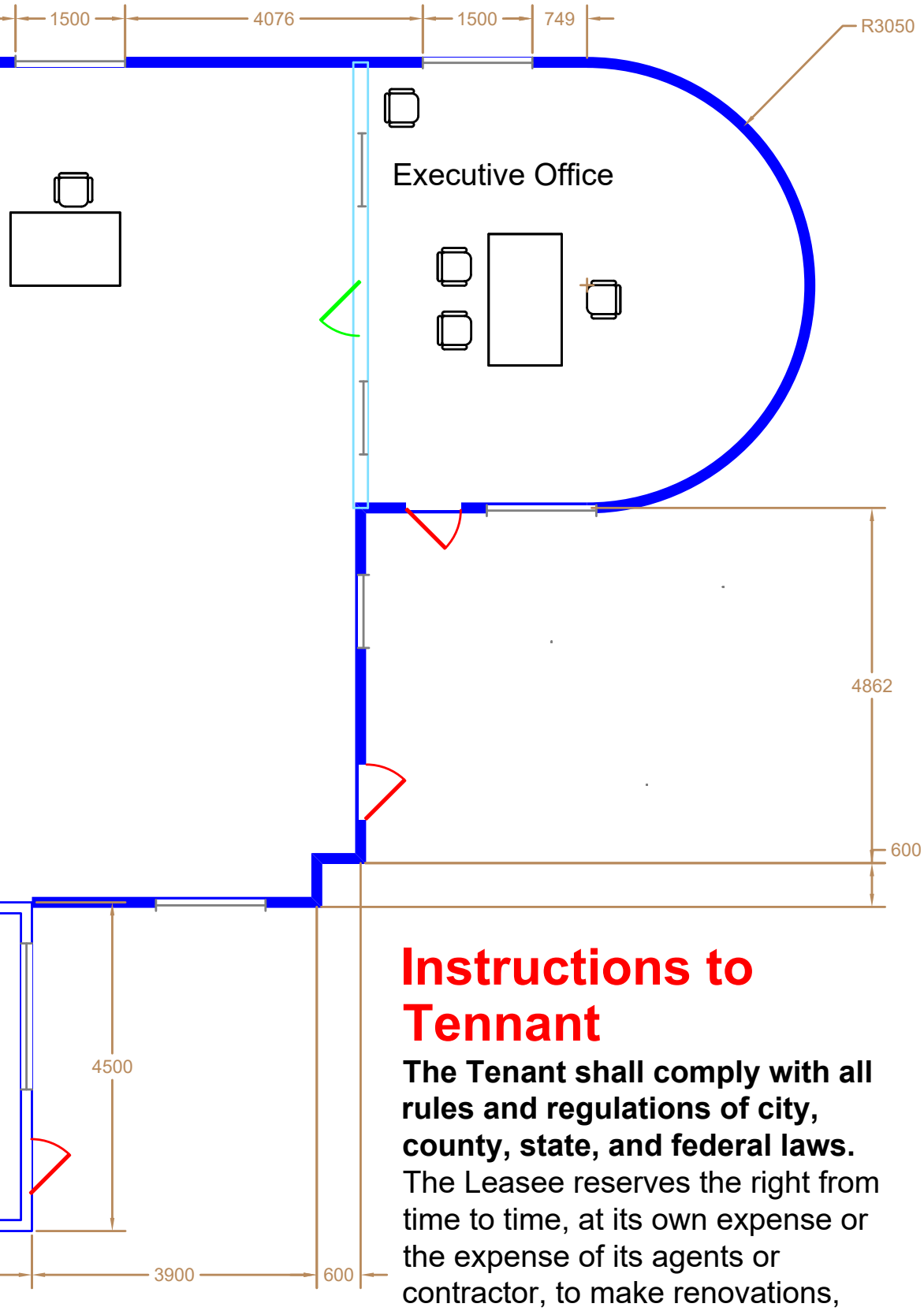
For most of this book, I'll show images from the Windows version of the program, because the Linux and Mac OS versions operate identically, for the most part. When there is a significant difference in functions or interface elements, then I illustrate them separately and write about them in grey colored text. Examples of differences include start-up and plotting.

The initial user interface is called "Dark," as shown above. If you prefer a light user interface (as I do, and it is necessary for images of the screens in this book), then you can change it as follows: Enter the **UiStyle** command, click **OK**, and then restart the program for the change in color to take place.



ARES Commander in the light user interface


1567
1500
3499
1500
2110
1516
1500
Reception
Area
3200
4300
15400
7200
1500
Board Room
3500
6900



Instructions to Tennant

The Tenant shall comply with all rules and regulations of city, county, state, and federal laws. The Leasee reserves the right from time to time, at its own expense or the expense of its agents or contractor, to make renovations, repairs and alterations in and about the leased premises, and may enter upon the premises at a reasonable hour for these purposes.

SAVING DRAWINGS

Command: Save
Alias: qsave
Menu bar: File | Save
Ribbon: Application button | Save
Shortcut: Ctrl+s (Cmd+s on Macs)
Toobar:

The most important task in CAD is to save your drawing work to the computer's hard drive. Fortunately, the program does this for you automatically every ten minutes. Still, you need to know how to from time to time change names of drawings, and this is done at the same time as saving it.

Along the title bar at the top of the program, you can see that the first drawing always starts with a generic name, *NONAME_0.dwg*. This is a name assigned by the program, but you should change it to something meaningful. This is done by saving the drawing.

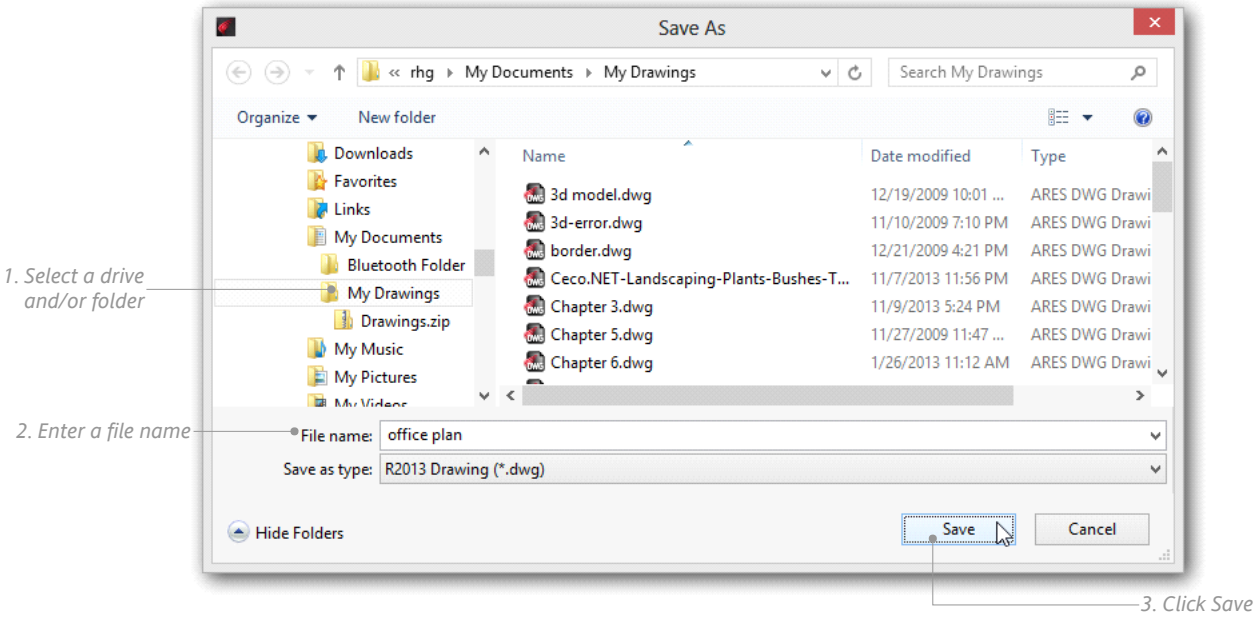
You save drawings with the **Save** command like this:

1. To start saving the drawing, enter the **Save** command, as follows:

   `: save` *(Press Enter)*

2. Because the drawing has the generic name "NONAME_0.dwg," the program knows to automatically display the Save As dialog box. See the figure below.
3. At this point, you decide on a *name* for the file. Sometimes, the name is decided for you by a client or by office CAD standards; other times, you can make up any name that makes sense to you. Since these tutorials are about the floor plan of an office, you'll call it "Office Plan."

   Also, consider the name of the *folder* in which you store it. You might have a project folder on your computer or on an office server for all your drawings.



Using the Save As dialog box

The default folder used by the program is named "My Drawings." Its location varies, depending on the operating system you are employing:

- In Windows 7 through 10, the *My Drawings* folder is found under the *C:\Users\<login>\Documents* folder.
- In Linux, *My Drawings* is found in the *\home\<login>\documents* folder.
- In MacOS, *My Drawings* is found in the *Macintosh\Users\<login>\Documents* folder.

(Replace *<login>* with the name by which you log into the computer.)

4. For this tutorial, use the following settings: rename the "NONAME_0" file name with **office plan**.
5. Store the drawing in the *My Drawing* folder, the default used by the program.
6. Click the **Save** button.

You know that the program saved the drawing with the new *office plan.dwg* name, because the title bar changes the name to [office plan.dwg]. From now on, each time you use the Save command, the program saves the drawing by this name; it no longer displays the Save File dialog box.

TIP Should you wish to save this drawing under a different name, use the **SaveAs** command. You save drawings by other names when you want to reuse them for other purposes.

SAVING DRAWINGS IN OTHER FORMATS

When you enter the **SaveAs** command, the Save File dialog box lets you change drawings to a number formats read by other CAD programs. To see the complete list, click on the **Save As Type** droplist, shown at right.

Drawing (*.dwg) files are read by the program, AutoCAD, IntelliCAD, and quite a few other CAD programs. It is crucial, however, that you match the version number. This is because older releases of CAD software cannot read newer releases of DWG format. (Newer versions of CAD software can, however, always read older DWG files.)

Warning! When saving drawings to older DWG releases, the program might erase or alter some entities if they are not supported by early formats of DWG.

DXF is read by many CAD, CAM, analysis, and graphical programs; short for "drawing interchange format." Always use the ASCII version, unless you know for sure that the Binary version can be read by the recipient.

DWS attaches CAD standards to drawing files; short for "drawing standards."

DWT saves drawings in *.dwt* format for use as templates; see Chapter 15.

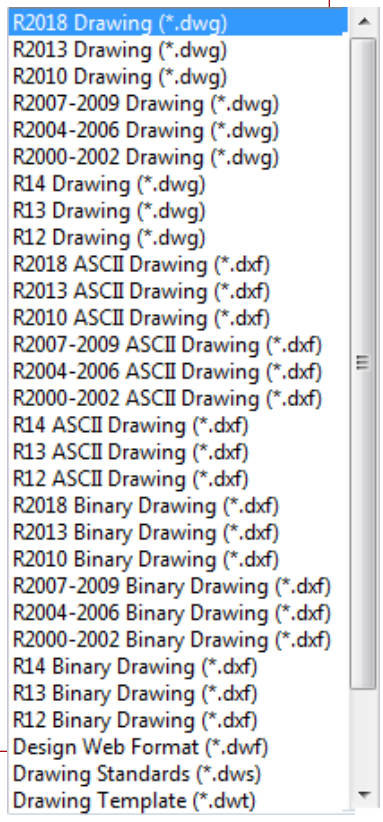
DWF is a simplified format for displaying drawings on Web sites; short for "design Web format."

ESRI Shape SHP is a file format used by ESRI's mapping software.

PowerCAD FLX is the file format used by PowerCAD, a predecessor to ARES.

Additional formats are available through the **Export** command: BMP (Bitmap), EMF (enhanced metafile), EPS (ecapsulated PostScript), JPEG, PNG, SLD (AutoCAD slide), STL (stereolithography), SVG (scalable vector graphics), TIFF (tagged image file format), and WMF (Windows metafile).

To save in PDF format, us the **ExportPDF** command.



SETTING UP UNITS

Command: UnitSystem
Aliases: un, units
Menu bar: Format | Unit System
Ribbon: Mange | Drawing | Drawing Units (Drafting and Annotation)
Toobar: --

The program displays measurement in a variety of styles, including metric, fractional, and exponential. (The latter expresses very large numbers). You can change the way by which the program displays lengths and angles, as well as the precision, with the **UnitSystem** command.

Initially, the program starts new drawings with the following measurement styles:

- **Length** in decimal units (another term for "metric") with four decimal places of precision; for example: `1.3456`
- **Angles** in decimal degrees with zero decimal places of precision, measured counterclockwise; for example: `90`

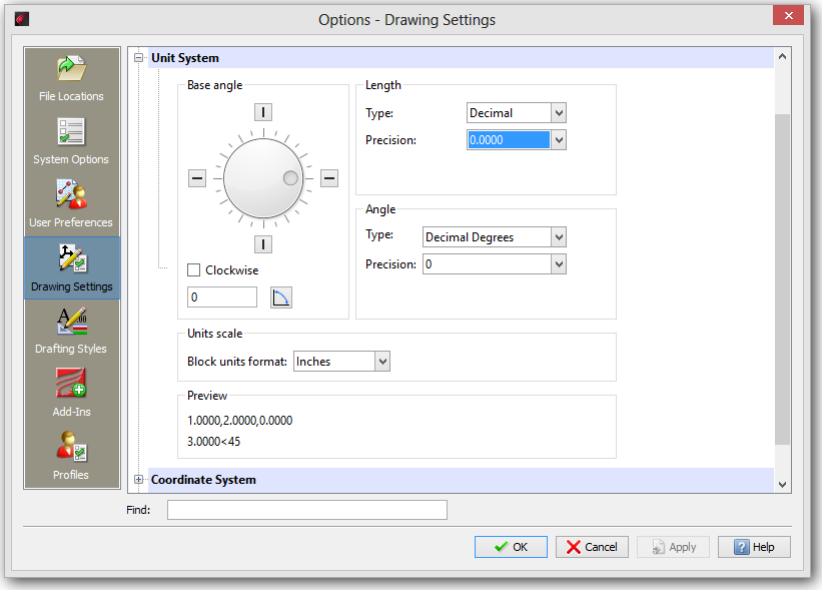
These initial units are *almost* good enough for the office layout drawing, because it was measured in millimeters. The only change you need to make is to set the precision of lengths to zero, since four decimal places is four too many.

To make this change, follow these steps:

1. Start the **UnitSystem** command by one of the methods listed above (at the start of this section). These include the following ones:

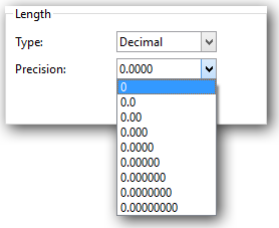
   » At the command prompt, enter **unitsystem.**
   » Or, enter one of the aliases, such as **un** or **units**. (*Aliases* are abbreviations of commands, such as "un," or mimic the command name of other CAD systems, such as AutoCAD or IntelliCAD, such as "units.")
   » Or, in the ribbon's **Manage** tab, look in the **Drawing** panel for the **Drawing Units** button ("Drafting and Annotation" workspace).
   » Or, from the **Format** menu, choose **Unit System**.

   Notice that the Options dialog box opens to the Drawing Settings | Unit System section:



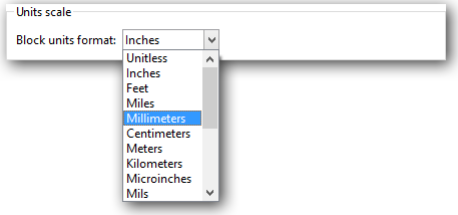
Settings units in the Options dialog box

2. The dialog box has many options, but you are interested in only the *precision* of lengths. Because the measurements of the floor plan were made to the nearest millimeter, the precision should be changed to 0 — no decimal places — as follows:
   a. Locate the **Precision** droplist in the **Length** section.
   b. Click the down arrow, and then select **0**.



Choosing precision from 0 to eight decimal places

3. Now move your attention to the Units Scale section of the dialog box:
   a. Click the droplist next to **Block Units Format**,
   b. Change it from "Inches" to **Millimeters**. (This setting becomes important when inserting blocks during a later chapter.)

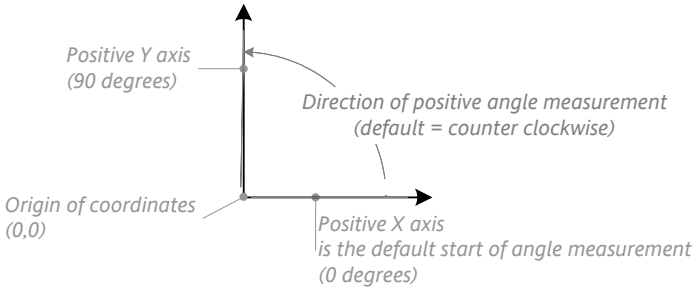


Choosing the type of linear units

UNITS OF ANGLE MEASUREMENT

Normally, the program measures angles this way:

- **0 degrees** starts from the positive x axis (East).
- **Angles** are measured using decimal degrees, such as 45.6789 degrees.
- **Positive angles** are measured counterclockwise, as illustrated below.



The **UnitSystem** command can change the default settings so that 0 degrees starts at any angle; angles are measured in degrees-minutes-seconds, grads, radians, or surveyor's units; and/or clockwise.

TIP There are 400 grads and 2*pi radians in a 360-degree circle.

4. Click **OK** to dismiss the Options dialog box.

You are free to enter distances more accurate than 1 millimeter, because the program remembers distances to 14 decimal places of accuracy. When the program displays coordinates on the screen, it rounds them up or down to the nearest millimeter.

After you specify the measurement style, the program will display all measurements in that style. But don't worry: you can switch to another measurement styles at any time, again with the UnitSystem command.

LIMITING THE DRAWING BOUNDARY

Command: DrawingBounds
Aliases: bounds, limits
Menu bar: Format | Drawing Boundary
Ribbon: --
Toobar: --

There is just about no limit to the size of drawings you can create with the program. You could draw the entire solar system full size if you wanted, and it would fit in the program.

The office plan is, however, a great deal smaller that a solar system, and so it can be useful to give boundaries to drawings. These constrain the extent of things like the grid marks and the Zoom Bound command. (More on these later.)

This is how to determine the size of the boundary:

1. Examine the size of the office plan in the sketch. Notice that the building is 22.5m wide and 16.4m deep. Leaving a bit of "breathing room" around the plan, the drawing will need about 25m of width and 20m of depth.
2. Start the **DrawingBounds** command, and then enter the values at the prompts:
   a. Specify the x,y coordinates of the lower left corner as **0,0**

```
: drawingbounds
Options: OFf, ON or
Specify lower left corner» 0,0
```

   b. Specify the coordinates of the upper right corner as **25000,20000**

```
Default: (12,9)
Specify upper right corner» 25000,20000
```

You won't notice any changes to the drawing until after you turn on the grid, as described next.

SETTING THE SNAP AND GRID SPACING

The primary advantage to drafting with CAD is that it permits you to create *excruciatingly* accurate drawings, and so the program has several features that help you draw with this level of accuracy. One such feature is called "snap mode."

Snap can be thought of as the drawing resolution. For instance, a snap of 10mm means the cursor moves in precise 10.0000mm increments.

Another is called the "grid," which helps us visualize distances. It does this by displaying an array of dots. For instance, a grid spacing of 1000mm means that a dot appears every 1m. While grid dots are displayed on the screen, they are not plotted on paper.

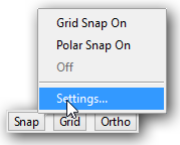
Here you set both of these accuracy features through the Options dialog box; in later chapters, you learn about additional accuracy aids that the program offers you.

Setting the Snap Spacing

Command: Snap
Alias: sn
Status bar: *right-click* Snap
Toobar: --

Follow these steps to set the snap spacing:

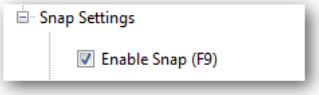
1. On the status bar, *right-click* the **Snap** button. (Right-click this means to move the mouse cursor over the Snap button, and then press the *right* button on the mouse.)



Right-clicking the Snap button to access the Settings dialog box

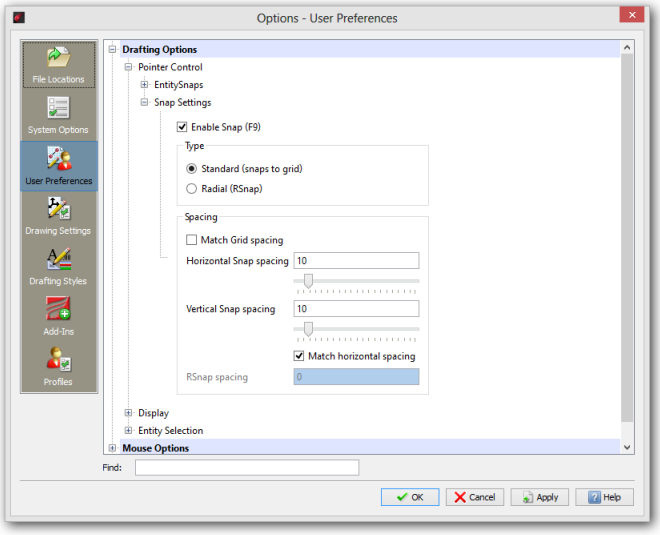
TIP Why not use the Snap command? This is a case where *not* using the command is the easier approach. The Snap and Grid commands display options through lines of text on the command bar, which can be more difficult for new users to understand than using a dialog box.

2. In the shortcut menu, choose **Settings**. Notice that the Options dialog box opens at the Snap Settings section. The advantage to right-clicking the status bar's Snap button is that it takes you directly to this part of the dialog box!
3. To move the cursor in ten-millimeter increments, make the following changes to the dialog box:
   a. Click the white square (called a "check box") next to **Enable Snap (F9)**.



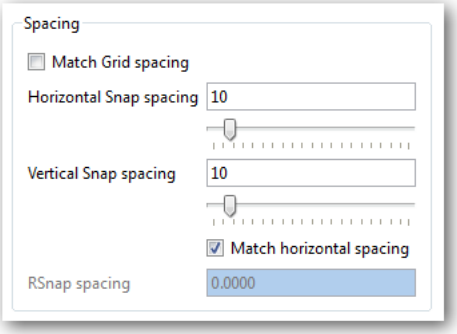
Turning on Snap mode

(The "F9" text reminds you that can turn snap on and off anytime by pressing **F9**.)



Changing the Snap settings

b. Change the setting of **Horizontal Snap Spacing** to **10**.



Setting snap distance to 10 units

Notice that next door **Vertical Snap Spacing** matches the change to 10, because the **Match Horizontal Spacing** option is turned on.

c. Leave other snap settings, and then click **OK** to close the dialog box.

4. The drawing looks no different, but reacts differently. Start the **Line** command, and then move the cursor about. Notice that it jumps, rather than moves smoothly.

5. Press **Esc** to cancel the Line command.

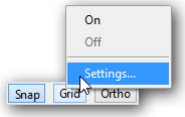
TIP When entering text and numbers in a dialog box, pressing the **Tab** key is a quick way to get to the next field. "Fields" are dialog box elements like buttons, text entry boxes, and list boxes. To return to a previous field, press **Shift+Tab** (hold down the **Shift** key, and then press **Tab**).

Grid Spacing

Command: Grid
Aliases: --
Menu bar: --
Status bar: *right-click* Grid
Toobar: --

To have a visual guide on the screen, turn on the grid. Follow these steps to set the grid spacing to 1000 and turn it on:

1. On the status bar, right-click the **Grid** button.

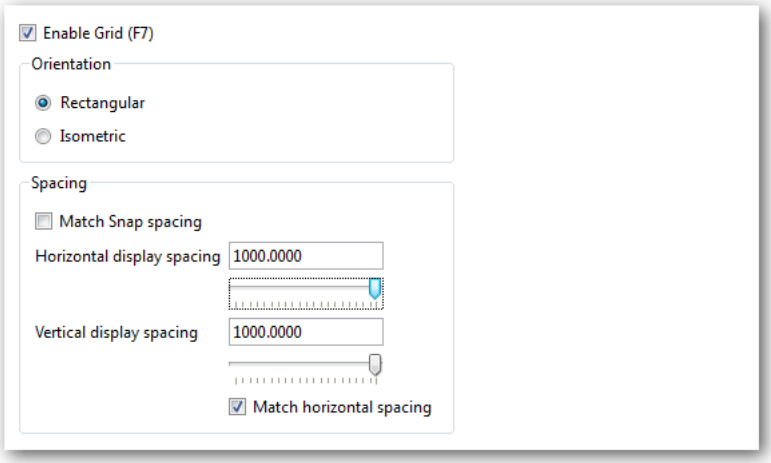


Accessing settings from the Grid button

2. In the shortcut menu, choose **Settings**.

Notice that the Options dialog box opens at the Grid Settings section (User Preferences | Drafting Options | Display | Grid Settings).

3. To display grid dots in 1000-millimeter increments, make the following changes to the dialog box:

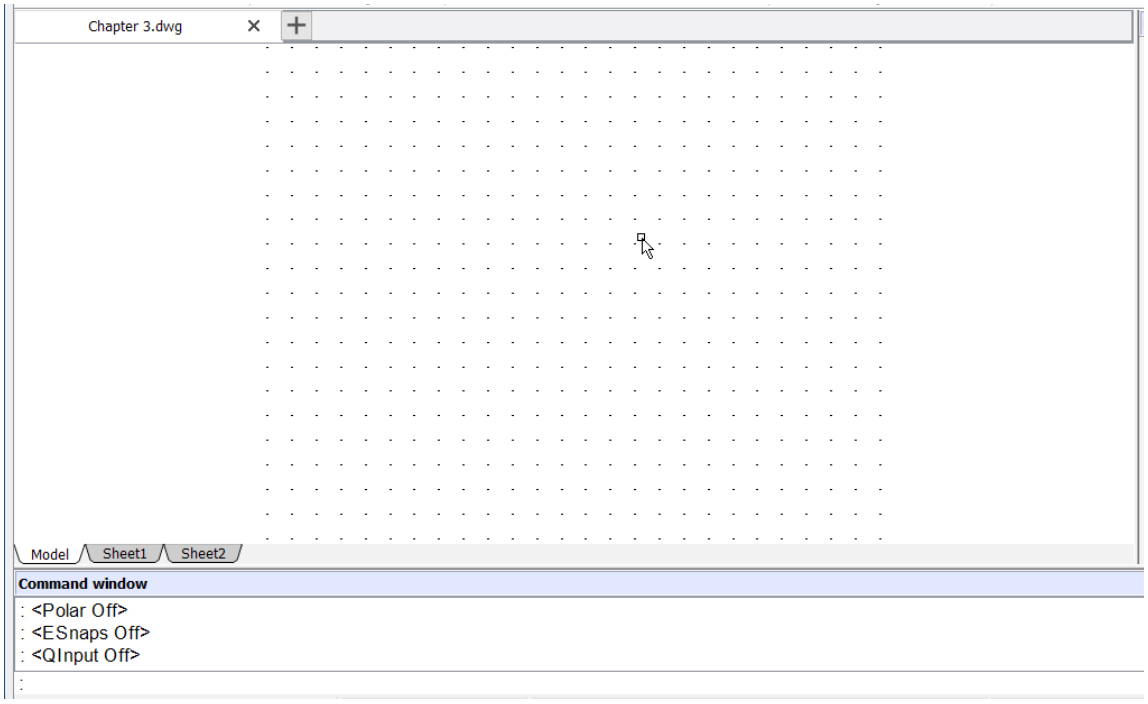


Specifying the spacing of the grid dots

   a. Click the white square (a "check box") next to **Enable Grid (F7)**. See figure below. ("F7" reminds you that the grid can be turned on and off anytime by pressing function key F7.)

   b. Change **Horizontal Snap Spacing** to **1000**. Notice that the **Vertical Snap Spacing** options matches the change to 1000.

   c. Leave the other grid settings, and then click **OK** to close the dialog box. A grid of dots should appear in the drawing area (a.k.a. graphics area).



Displaying grid dots

The grid is meant as guide; the default spacing of 0.5mm gets in the way, because it spaces the dots too closely. In fact, the program won't even display grid dots in this case, because there would be too many on the screen. By changing the spacing to 1000mm, you get just enough to help you get a feel for 1-metre distances.

SEEING THE ENTIRE DRAWING

Command: Zoom Bounds
Alias: z b
Menu bar: View | Zoom | Bounds
Ribbon: View | Navigate | Zoom Bounds (Drafting and Annotation)
Toobar: 

To change your view of the drawing, the Zoom command is useful. The Bounds option lets you see the entire grid. The edge of the graded area will give you a visual indication of the extents of the drawing limits.

Enter the **Zoom** command...

```
: zoom
Default: Dynamic
Options: Bounds, Center, Dynamic, Fit, Previous, SElected, specify a scale factor
(nX or nXP) or
```

...and then the **B** option:

```
Specify first corner» b
```

Don't be intimidated by the many options of this command; you use just a couple for the most part.

TIP For many command options, you need only type the first letter of the option. Above, you typed **b** as the abbreviation for the Bounds option. When two options begin with the same letter, you need to type the first two characters of the option.

When you began, the program displayed an area of about 9 by 12; now it displays an area of 25m by 20m. The Zoom command let you see the "big picture," as well as zooming in for detailed looks.

Adding Layers

If you ever worked with overlay drafting, then you are familiar with the concept of *layers*. In overlay drafting, you draw the base plan on a clear sheet of Mylar (plastic drawing media), draw the electrical plan on another sheet, the structural on a third, and so on. Since the Mylar is transparent, you can overlay the three drawings to create a single blueprint.

Layers in CAD programs operate similarly. You draw parts of drawings on different layers. Then you can turn layers off and on to display the content of the drawing in different ways. For example, the electrical contractor would be interested in seeing only the layers showing the base plan and the electrical.

A further advantage to layers is that properties can be applied to entities on a per-layer basis. For instance, all entities on a layer called "grass" could be colored green (or maybe brown, depending on the time of year!). Changing the layer color to brown instantly changes the color of all entities assigned to that layer.

You can also assign lineweights, linestyles, and print styles to layers, as well as determine whether they should be printed, hidden from sight (frozen), or be locked (uneditable).

It is possible to create hundreds and even thousands of layers in drawings for complex projects, but it is more common to work with just a dozen or so layers. In this book, you work with a mere half-dozen layers to segregate items like the text, different types of walls, and so on.

CREATING LAYERS BY NAMING THEM

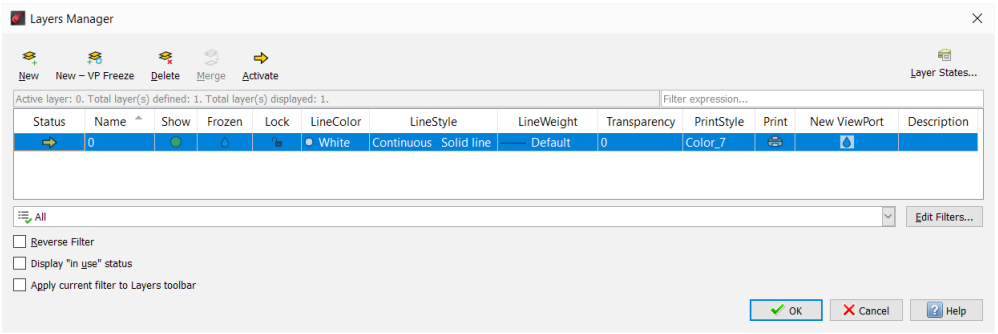
Command: layerdialog
Alias: ...
Menu bar: ...
Ribbon: ...
Toobar: ...

Layers are created by naming them. The Layers Manager dialog box lets you give layers names up to 255 characters long.

But first think about the layers should handle by looking at the sketch. You can see a number of parts that would benefit from being segregated by layers: there are the external and internal walls, furnishings, doors, and windows.

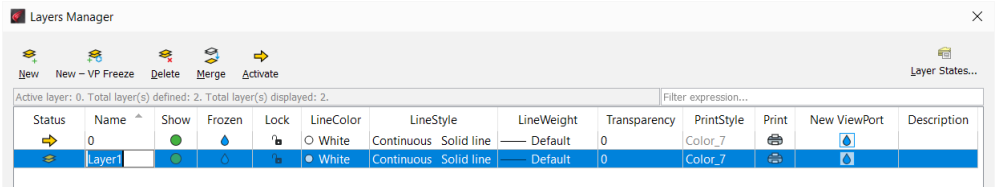
1. Enter the **LayerDialog** command. Notice that the program displays the Layers Manager dialog box, which lets you control almost every aspect of the drawing's layers.

   The drawing already has a layer, 0. Every new drawing has layer 0, which is fixed and cannot be removed.



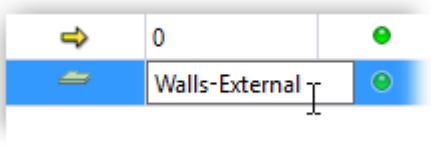
Layer 0 found in every drawing

2. Create a layer by clicking the New **New** button. Notice the new layer called "Layer1."



Creating a new layer

3. Change the name to **Walls-External**.



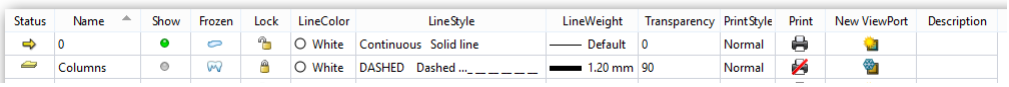
Renaming the layer

4. It is easier to determine which lines belong to the layer by assigning a color. All entities assigned to a layer take on this color, although the color and other properties can be overridden, if necessary. To change the color of the Walls-External layer to dark blue, follow these steps:

   a. Across from the layer name Walls-External click the circle under the **LineColor** column.

ABOUT LAYERS

The program provides you quite a bit of control over layers and their properties. Every layer has the same set of properties, as displayed by the header bar in the Layers Manager dialog box or palette:



Status reports the status of each layer; the arrow indicates the active layer.

Name identifies layers, and can be up to 255 characters long. You can use numbers, letters, and some punctuation: 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 layer names in drawings.

Show displays entities assigned to the layer. When show is off, you cannot see or edit entities nor are they plotted.

Freeze prevents entities on layers from being seen, edited, or plotted. In addition, the entities are not included when the program performs drawing rebuilds; this is the only manner in which Freeze differs from Show being turned off.

When layers are not frozen, entities can be seen, edited, and plotted. (To *thaw* means to turn off the frozen status.) It is better to freeze layers than turn off their Show status.

Lock prevents them from being edited. To *unlock* layers means to make the entities available for editing.

LineColor specifies the color for entities placed on the layer. The default color is white, which is displayed as white or black depending on the background color. The color assigned by the layer can be overridden for individual entities with the LineColor command.

LineStyle specifies the linestyles for entities placed on the layer. The default is Continuous Solid Line. The style assigned by the layer can be overridden for individual entities with the LineStyle command.

LineWeight specifies the line width for entities on the layer: default is 0.00mm; maximum is 2.11mm (0.08").

Transparency specifies the "see-throughness" of entities: 0 is opaque, while 90 is the maximum transparency.

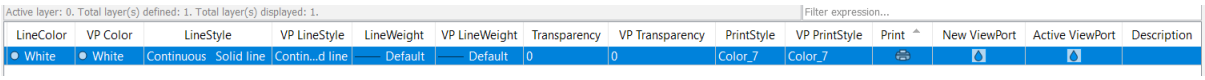
PrintStyle determines the named plot style with which to print entities assigned to the layer. Print styles define colors, widths, and other properties to be used during printing.

Print prints layers; when print is off, entities are not printed.

Description includes explanatory text for each layer. To add a description, click the field twice.

LAYOUT MODE PROPERTIES

When drawings are in layout mode (paper space), this dialog box displays many more columns:



VP Color specifies the color of entities in the viewport, allowing them to look different from model space.

VP LineStyle specifies the linestyle of entities in the viewport.

VP LineWeight specifies the lineweight of entities in the viewport.

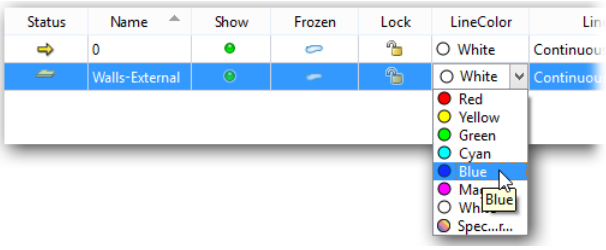
VP Transparency specifies the transparency of entities in the viewport.

VP Print Style specifies the print style of entities in the viewport, if print styles are enabled.

New ViewPort immediately freezes/thaws the specified layer the moment new viewports are created in layouts.

Active ViewPort independently freezes/thaws the specified layer of the active (current) viewport in layouts.

b. Notice the droplist that appears. Select **Blue**.



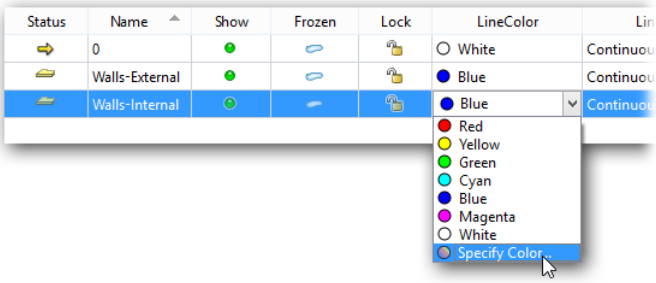
Choosing a color for the selected layer

Notice that the color of the circle changes to blue.

Layer with changed color

5. Create another layer, naming it "Walls-Internal" and assign it a light blue color. To select a color other than the seven shown in the LineColor droplist, follow these steps:

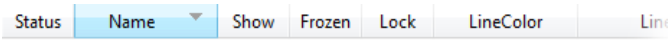
   a. Click **Specify Color**.



Choosing Specify Color to access additional colors

CONTROLLING LAYER LISTINGS

To control drawings with many layers, you sort can layers in a variety of ways and display selected groups of layers.

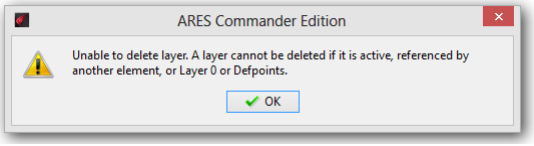


To sort layers alphabetically, click headers such as Name or Freeze. Notice the arrowhead. Click the headers a second time to sort in reverse order (Z to A).

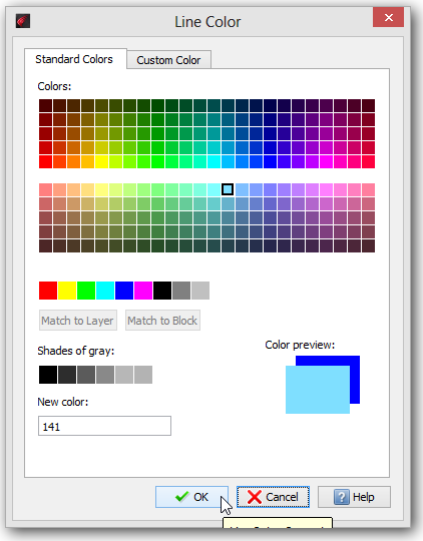
ERASING LAYERS

To erase one or more layers from the drawing, select them and then press the **Del** key. The program changes the layers status icon to  , indicating the layer will be removed once you click **OK**. To prevent accidently deletion of layer(s), click **Cancel**.

You can erase layers that are not empty, not active, not part of externally referenced drawings, and not layer 0. If you try, the program displays a warning dialog box.



b. Notice the Line Color dialog box. Click on a light blue square, such as 141.



Line Color dialog box for choosing colors

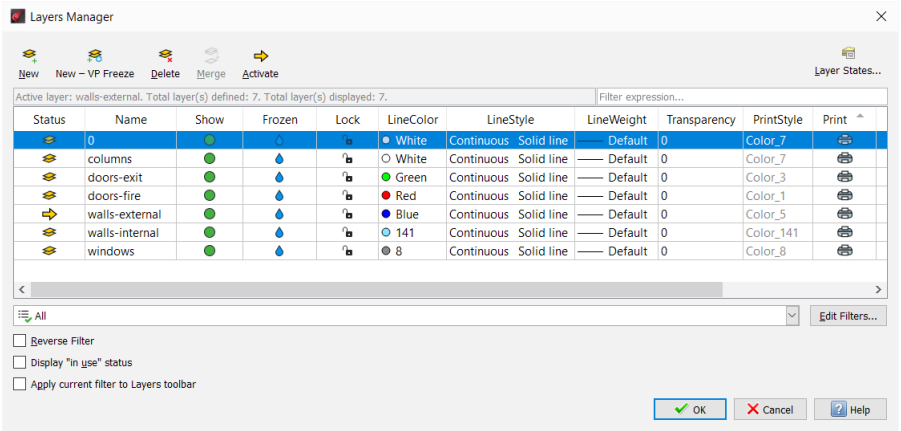
c. Click **OK**.

Notice that the color for layer Walls-Internal changes to light blue.

6. Continue adding layers and assigning colors according to the table below:

Layer Name	LineColor
Furnishings	White
Doors-Exit	Green (color # 4)
Doors-Fire	Red (color # 1)
Windows	Gray (color # 8)

When done, the Layers Manger dialog box should look like this:

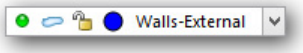


All layers assigned in the drawing

If you make a mistake in spelling a layer name, just (a) click the name twice, and then (b) enter the correction.

7. When you finish assigning colors to layer names, pick the "Walls-External" layer, and then click **Activate**. From now on, all drawing takes place on the Walls-External layer — at least until you activate another layer.

8. Click **OK** to exit the Layers Manager dialog box. Notice that the name of the layer on the Layers toolbar or ribbon changes from 0 to "External-Walls" and the color to blue.



Color and name of the default layer

MAKING AUTOMATIC BACKUPS

Command: options
Aliases: --
Menu bar: Drafting Tools | Options
Ribbon: Application button | Options
Toobar: --

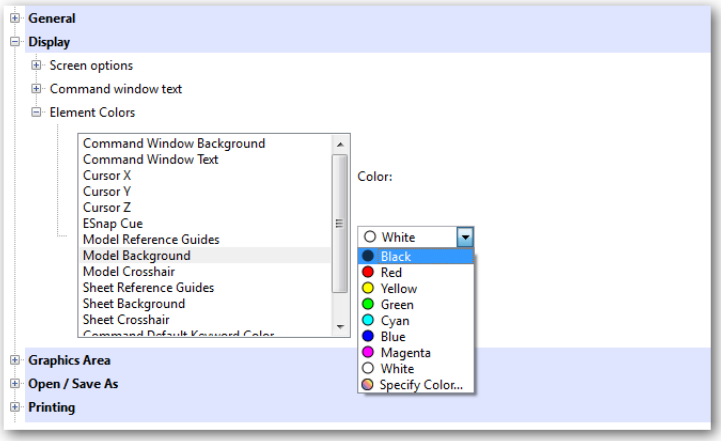
To make software run as quickly as possible, all programs keep their data in the computer's memory (RAM). The program is no different. The drawback to this approach is that when Windows crashes or when the power is cut to the computer, you lose your work. Because Windows crashes are not uncommon, it is an excellent idea to save drawings every 5 or 10 minutes.

To do so, get into the habit of pressing **Ctrl+s** regularly (Cmd+s on Macs). This shortcut keystroke is for the Save command.

The program also quietly saves your drawings automatically every ten minutes to a folder named *c:\users\<login>appdata\local\temp\ares commander edition_autosave* (where <login> is your Windows login name). Automatically-saved drawings are given the extension of *.ds$*. (The $ sign is traditionally used in computing to signify temporary files.) The backup file of this tutorial's drawing is *office plan_19172.ds$*.

TIPS There can be confusion over the color "white" — or is it black? The program switches the color entities between white and black depending on the drawing area's background color. When black, the program displays white entities; when white, the program displays them in lines. Nevertheless, the program always calls the color "white." To change the background color of the drawing area, follow these steps:

1. From the **Tools** menu, select **Options**. (Or click the Application button, and then click Options.)
2. In the Options dialog box, click the **System Options** tab.
3. Open the **Display** node, and then the **Element Colors** node.
4. Select the window element (**Model Background**), and then choose a color.=



To enable automatic backups, follow these steps:

1. From the **Tools** menu, select **Options**. The program displays the Options dialog box.
2. Select the **System Options** tab.
3. Open the **Auto-save & Backup** node, and then the **Auto-save/backups** node.

Automatic backup options

4. If necessary, click the check box next to **Enable Auto-save** to turn on the option.

   The **Save Document Every** box specifies how much time elapses before the program saves the drawing. Don't set this number too low, or the computer spends so much time saving to disk that you cannot do other work.

   If haven't already done so, you might want to turn on **Save Backup at Each Save**. This means that the program makes backup (*.bak) copies of drawing files with each save. The older version is renamed *office plan.bak.*

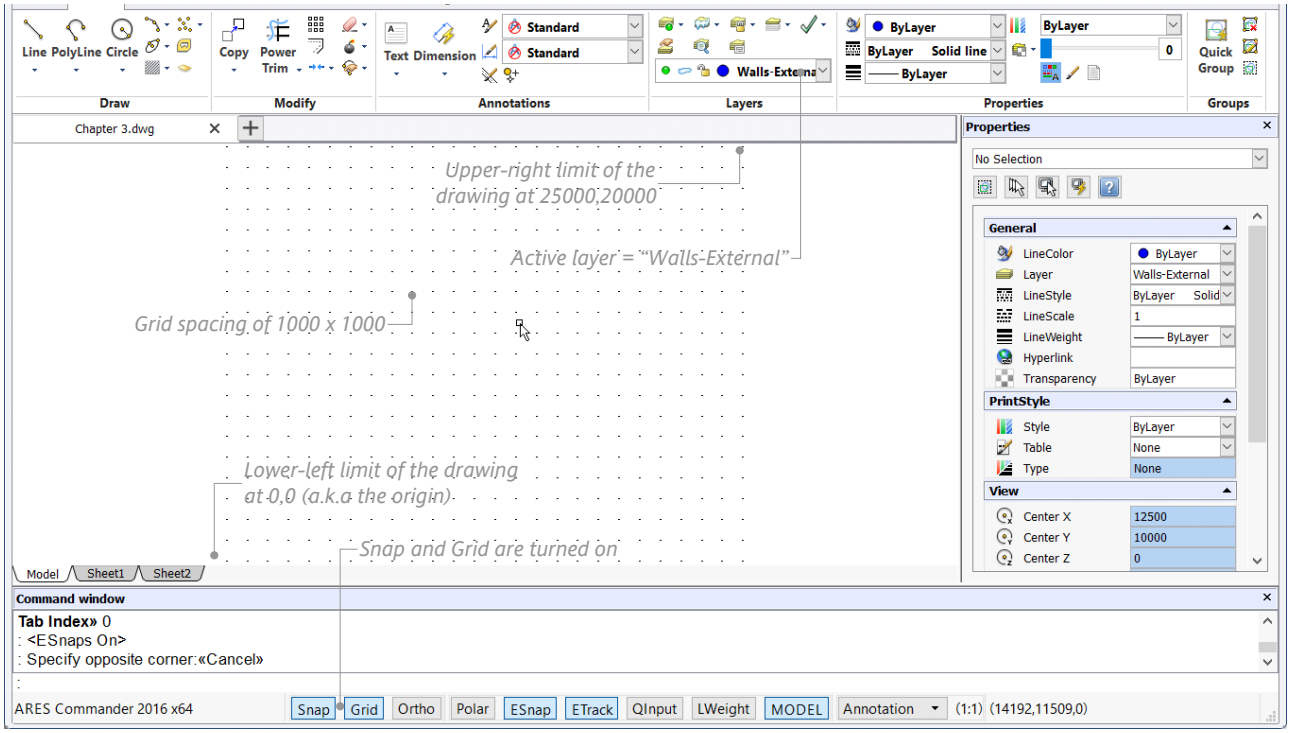
5. Click **OK**. Although the program will automatically save your drawings every ten minutes, it is still a good idea for you to save your work with **Ctrl+s** (**Cmd+s** on Macs) after finishing significant amounts of editing.

TIP Automatic saves and backups are not made to the original files. During automatic saves, the program saves drawings with the extension of *.ds$* and backed up files with *.bak*.

6. If you need to stop at this point, use **File | Exit**. The program asks to save open drawings, and then closes down.

Summary

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



The tutorial drawing at the end of this chapter

On the toolbar, you see that the color of the current layer is blue and its name is Walls-External. On the status line, you see that the coordinates are displayed in millimeters, and that drafting modes snap and grid are turned on.

In the next chapter, you begin drawing the external walls of the floor plan.

Notes

CHAPTER 05

Drawing Walls

In This Chapter

- Drawing walls with lines, rectangles, polylines, and richlines
- Creating richline styles, and modifying polyline widths
- Offsetting and trimming entities
- Understanding absolute and relative distances

The previous chapter was where you learned how to start the program, set up new drawings, and save drawings to the computer's hard drive. In this chapter, you learn how to draw lines accurately, make simple changes to drawings, and produce copies of drawings on your printer.

KEY TERMS IN THIS CHAPTER

Absolute coordinates are measurements made relative to the drawing's origin.
Entity snaps snap the cursor to geometric features, such as the ends of lines or centers of circles.
Origin locates 0,0, which is usually found at the lower-left corner of drawings.
Polylines are connected lines and/or arcs that are treated as a single entity.
Read-only refers to drawings that cannot be saved to their original files.
Relative coordinates are measurements made relative to the last point.
Richlines draw up to 16 parallel lines, with optional color fill and end caps.

COMMANDS

Command	Shortcuts	Menu Selection	Status Bar
Cancel	Esc	...[1]	...
EditPolyline	pedit, edpl, pe, edpline	Modify | Entity | PolyLine	...
Lineweight	lw, lweight	Format | Line Weight	LWeight
Open	Ctrl [2] +o	File | Open	...
Ortho	F8 *or* Ctrl+l	...	Ortho
EntitySnap	es, esnap, os, osnap	...	ESnap | Settings
Offset	o, qoffset	Modify | Offset	...
PolyLine	pl, pline	Draw | PolyLine	...
Rectangle	rec, rect, rectang	Draw | Rectangle	...
Redo	Ctrl+y	Edit | Redo	...
RichLine	rl, ml, mline	Draw | RichLine	...
RichLineStyle	rls, rlstyle, mlstyle	Format | RichLine Style	...
Trim	tr	Modify | Trim	...
Save	Ctrl+s	File | Save	...
U	Ctrl+z	Edit | Undo	...
Zoom	z	View | Zoom	...

[1] No menu selection

[2] Press Cmd key on Mac computers

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- **Rectangle** command receives additional options: 3Corner, 3Point center, CEnter, and Parallelogram.
- **Trim** command now trims hatches and gradients so that they maintain their associativity, even when they are divided into different hatched areas.

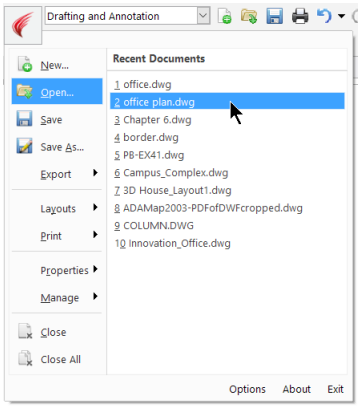
Bringing Back the Office Drawing

Command: Open
Menu bar: File | Open
Application menu: Open
Shortcut: Ctrl+o (Cmd+o on Macs)
Toolbar:

If you exited the program at the end of the last chapter, you need to restart it, and then load the office plan drawing. Here's how to do this:

1. Start the program by double-clicking its icon on the desktop.

2. The program remembers the names of the last ten drawing files you saved. You access them like this:
   a. Click the **Application** button.
   b. Notice the **Recent Documents** list. Click on the *office plan.dwg* file name. Notice that the drawing is opened in the program.



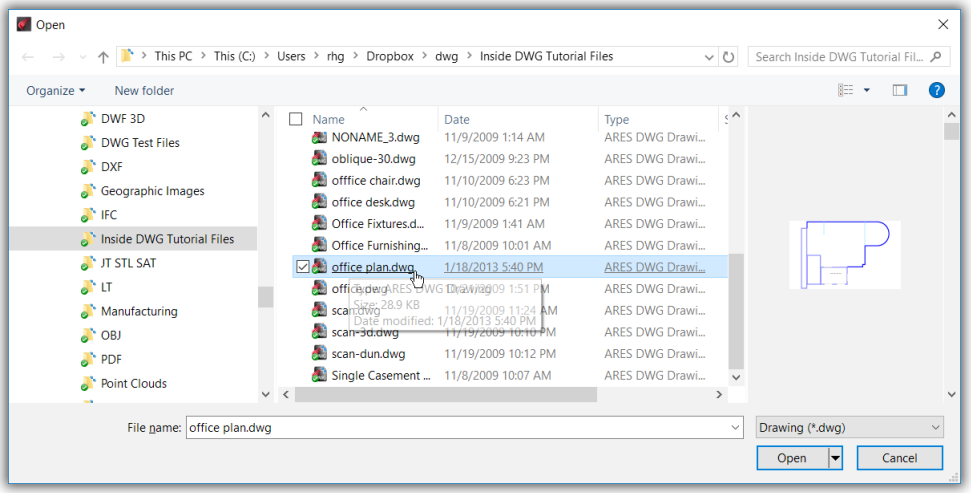
Selecting a recent drawing from the Recent Documents list

The drawing should look exactly the same as when you last saw it, that is to say, blank, except for the grid.

OPENING DRAWINGS WITH THE OPEN DIALOG BOX

When drawing names are not listed in the Recent Documents list, then you have to use the Open dialog box. Here is how to do this:

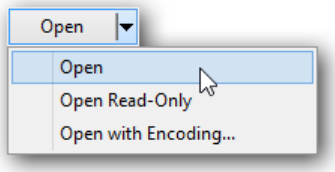
1. Enter the **Open** command, or click the Open button. Notice the Open dialog box.



2. Navigate to the folder in which drawing files are stored.
3. (Optional.) Choose a file format other than DWG, such as DWF, DXF, DWT, DWS, FLX, or SHP.
4. Select a file name. Notice that a preview image of it appears to the right of the dialog box.
5. Click **Open**. The drawing appears in the program.

OPEN OPTIONS

The **Open** button is actually a droplist with some options:



- **Open Read-only** opens the drawing in *read-only* mode, which means you cannot save (*write*) changes to the original file; instead, you have to save the drawing by a different name.
- **Open with Encoding** opens drawings with alternate character sets. "Coding" refers to the language code page number software uses to determine character encoding, such as Baltic or Arabic. This option displays a dialog box that allows you to choose the appropriate code page number.

TIP You can open the same drawing in the program more than once. Second and subsequent copies of the drawing are opened read-only.

Drawing Walls

Let's get some lines on the screen! The office floor plan consists largely of straight lines, some of which form rectangles. For the remainder of this chapter, you draw straight lines and rectangles using a variety of methods.

Line command draws independent line segments.

Rectangle command draws rectangles.

PolyLine command draws connected line and arc segments with optional width.

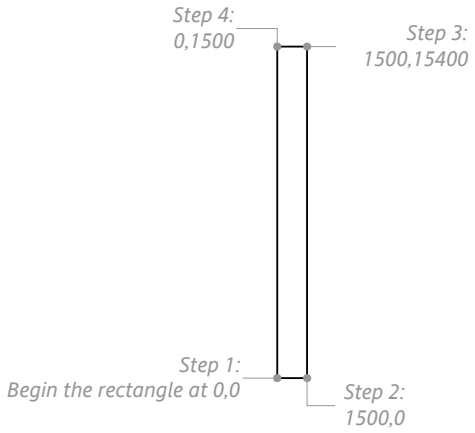
RichLine draws connected parallel lines.

Using each one gives you experience with the different methods of drawing lines in the program. You gain exposure to the pros and cons of each, and then you can decide which is best for particular situations.

DRAWING LINES WITH ABSOLUTE COORDINATES

Command: Line
Alias: l
Menu bar: Draw| Line
Ribbon: Home | Draw | Line (Drafting and Annotation)
Toolbar:

You start the drawing the common hallway with the Line command. The lines begin at the lower-left corner known as the "origin," where x = 0 and y= 0, and then are drawn using absolute coordinates. The x,y coordinates are shown in the following illustration.



Coordinates for drawing the hallway

Absolute coordinates are just that: the actual x,y coordinates for each point in the drawing. (Later, you draw with the other kind of coordinates, relative.)

1. Start the **Line** command using one of the methods listed above. That is...
   - » Enter **line** at the ':' prompt
   - » Or enter its alias **l** at the ':' prompt
   - » Or from the **Draw** menu, select **Line**
   - » Or in the ribbon's **Home** tab, choose **Line** in the **Draw** panel
   - » Or click the **Line** button on the **Draw** toolbar

   No matter which method you choose, notice that the program prints "_LINE" at the command line as confirmation to your action.

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

```
: _LINE
Options: Segments or
Specify start point» 0,0
```

3. If the Quick Input interfaces appears, turn it off by clicking **QInput** on the status bar.

4. To draw the lower 1500-long line, you need to tell the program that the other end of the line is located at the x,y coordinates of 1500,0:

```
Options: Segments, Undo, Enter to exit or
Specify next point» 1500,0
```

TIPS Notice the words **Segments**, **Undo**, and ***Enter*** in the command prompt. They are colored blue to indicate that these are options of the Line command, things that you can do instead of drawing lines. Here is what they mean:

- **Segments** draws single-line segments; one line is drawn, and then command ends
- **Undo** undraws the last line segment; the previous line is removed from the drawing
- ***Enter*** exits the command. *Enter* is italicized to indicate it is a keystroke: you press the **[Enter]** key; you do not type E-n-t-e-r.

Some of the blue letters are underlined. This indicates the abbreviation. You can enter **Undo** or **u**.
In addition, I underline the prompts that apply to you, to help them stand out from the many options.

---

5. The next line is 15400mm north. Its endpoint is located at coordinates 1500,15400:

```
Options: Segments, Undo, Enter to exit or
Specify next point» 1500,15400
```

Remember that these numbers are large because they represent millimeters.

6. Draw the topmost segment:

```
Options: Segments, Undo, Close, Enter to exit or
Specify next point» 0,15400
```

TIP If you make a mistake entering coordinates, simply type **u** to undo the last segment, and then re-enter the coordinates.

---

7. To finish the lot boundary, use this shortcut: type **c** (short for "Close"), instead of typing the final coordinates (0,0).

```
Options: Segments, Undo, Close, Enter to exit or
Specify next point» c
```

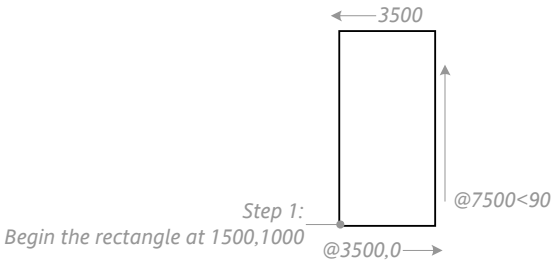
The program automatically draws a line from the current endpoint to the beginning of the first line.

8. Save your work by pressing **Ctrl+s** (**Cmd+s** on Macs).

DRAWING LINES WITH INPUT & RELATIVE COORDINATES

You drew the hallway with *absolute coordinates*, coordinates based on measurements from origin (at 0,0). The program can also draw lengths and angles, known as *relative coordinates*. In this case, the program draws a line from the current point relative to the last point.

It is easier to see relative coordinates in action, rather then to read about them. Draw the next rectangle using the Line command and relative coordinates. (The next rectangle is the 3500x7500mm storage room, located 1000mm up from the x axis.)



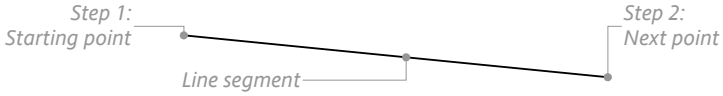
Direct distance instructions for drawing a rectangle

1. Turn on ortho mode by clicking the Ortho **Ortho** button on the status bar. (It turns blue when turned on.) The program acknowledges this at the command prompt:

```
: <Ortho On>
```

SUMMARY OF LINE COMMAND OPTIONS

```
: line
Options: Segments, Undo, Enter to exit or
Specify next point» (Pick a point, or enter an option)
```



Specify start point — specifies the coordinates of the starting point of the line segment. You can specify the point by any of the following methods:

- Pick a point in the drawing using the cursor.
- Enter coordinates at the keyboard.
- Obtain a point from existing geometry using entity snaps.
- Press **Enter** to continue the line from the last-drawn line or arc.

Segments — draws single line segments, prompting for start and end points of each segment.

Enter to exit — exits the command when you press the **Enter** key.

ADDITIONAL OPTIONS

```
Options: Segments, Undo, Close, Enter to exit or
Specify next point» (Pick a point, or enter an option)
```

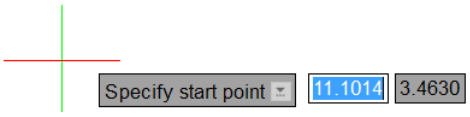
Undo — undraws the last line segment; repeat this option to undraw previous segments.

Close — closes the polygon by drawing a segment that connects the last and first points automatically.

Specify next point — specifies the endpoints of line segments.

Ortho is short for "orthographic." This mode forces the program to draw all lines precisely to the horizontal or vertical — no angled lines, something that is useful for drawing rectangles.

2. Quick Input works naturally with relative coordinates. Turn on the feature by clicking the QInput **QInput** button on the status bar.
3. Start the **Line** command. Notice that the command name line appears next to the cursor.
4. The line should start at coordinates of x,y = 1500,1000. These are is the absolute coordinates for the lower left corner of the rectangle. Notice that the Quick Input interface is now displaying three fields near the cursor:



Elements of the quick input interface

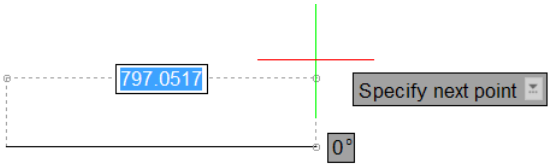
» **Prompt** is shown by "Specify start point." This is the same wording of the Line command's prompt you would see in the Command window when Quick Input is turned off.
» **Options** are accessed by clicking the down arrow; see Tip later.
» **X coordinate** is shown by the first input field, which happens to be "11.1014" in blue now. See figure above.
» **Y coordinate** is the second input field, which happens to be colored gray now.

Enter **1500** in the X coordinate field.

5. To enter the value for the Y coordinate field, press **Tab**. Notice that it turns blue.

TIP You can press **Tab** repeatedly to go back and forth between the two coordinate fields. As an alternative to pressing Tab, you can press ',' (comma) just as when you enter coordinates at the command prompt, like **1500,1000**.

6. Enter **1000** in the Y coordinate field, and then press **Enter**.
7. Notice that the look of Quick Input changes with two different coordinate input fields:



» **Distance** is shown by the first input field, which looks like "797.0517" in blue now (see figure above). This is the distance measured from the last point (1500,1000) and so is a *relative* distance.
» **Angle** is the second input field, which happens to be colored gray now. This angle is measured from the positive X axis, and so is *not* a relative measurement!

Enter **3500** in the Distance field.

You don't need to change the Angle field, as long as it read 0 degrees. If it does not, move the cursor to the right, and ortho mode takes care of the rest. *(If you don't want to use Quick Input, then you enter relative coordinates at the command prompt like this:*

CONTROLLING QUICK INPUT

The look of the Quick Input interface can be changed through variables separately when creating entities or editing using grips.

DRAWING CREATION

The following variables affect the look of Quick Input during drawing creation, such as placing lines and circles.

DynPrompt toggles (turns on or off) the prompt field; default = 1, on.

DynTooltips determines which tooltips are affected by appearance variables: (1) all or (0) Quick Input only; default = 1.

TooltipMerge toggles whether entity snap tooltips and Quick Input tooltips are merged into a single tooltip: (1) yes or (0) no; default

DynMode controls the use of Quick Input pointer input and dimension input methods:

DynMode Bitcode	Meaning
0	All quick input features turned off; like turning off **QInput** at the status bar
1	Only pointer input on; no command input, no options
2	Only dimensional input on; no QInput with first pick
3	Pointer input and dimensional input on (default)

DynPiCoords ("pi" = pointer input) determines the type of coordinates used by Quick Input:

DynPiCoords	Meaning
0	Displays relative coordinates with second pick point (default)
1	Displays absolute coordinates with all pick points

DynPiFormat determines the format of second and subsequent coordinates:

DynPiFormat	Meaning
0	Displays polar coordinates (default)
1	Displays Cartesian coordinates

DynPiVis determines when the Quick Input pointer is displayed:

DynPiVis	Meaning
0	Only when the user enters a point at a prompt
1	When the program prompts for a point
2	Always (default)

GRIPS EDITING

DynDiGrip ("di" = dimension input) determines which input boxes are displayed during grips editing; the default has all of them turned on:

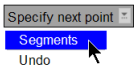
DynDiGrip	Meaning
0	None
1	Absolute length
2	Change in length
4	Absolute angle
8	Change in angle
16	Arc radius dimension

continued...

```
Options: Segments, Undo, Enter to exit or
Specify next point» @3500,0
```

The @ symbol tells the program to draw the line to the right by 3500 units from the last point of 1500,1000, and 0 units up.

TIPS To access the command's options, press the down arrow on your keyboard's cursor pad. The currently-available options appear below the prompt field:



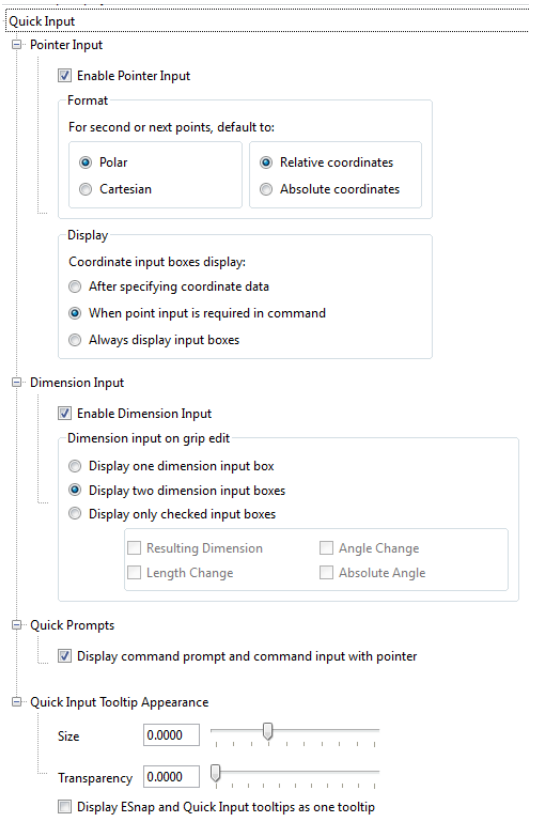
Choose an option with your cursor, or move the highlight with the cursor keys and then press **Enter**.
To cancel a Quick Input session, press **Esc**.

DynDiVis determines which of the Quick Input dimensions are displayed during grips editing:

DynDiVis	Meaning
0	First quick input dimension in cycle order
1	First two quick input dimensions (default)
2	All dynamic dimensions

OPTIONS DIALOG BOX

All of the options are available to customize the look and operations of Quick Input through the Options dialog box **User Preferences > Drafting Options > Quick Input**, as shown below.



8. To draw the next line, move the cursor upwards. Notice that ortho mode forces the Angle field to change to 90 degrees.

In the Distance field, enter **7500** and then press **Enter**. *(When not using Quick Input, you specifying the relative coordinates like this:*

```
Options: Segments, Undo, Enter to exit or
Specify next point» @7500<0
```

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

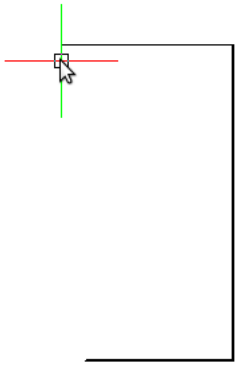
Notation	Meaning
@	Use relative coordinates.
7500	Distance is 7500mm from the current point.
<	Draw the line at an angle...
0	...of 0 degrees

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

9. You need to enter 3500 to draw the top of the rectangle. This time, though, use a different method called "direct distance entry" to indicate the length. Follow these steps:

   a. Turn off Quick Input by clicking the **QInput** button on the status bar. It turns gray.

   b. Move the mouse to the left. Notice a ghost line that appears.



Ghost line aids in determining the line's length and direction

   c. Enter **3500**, and then press **Enter**. Now that Quick Input is off, the prompts appear back in the Command window:

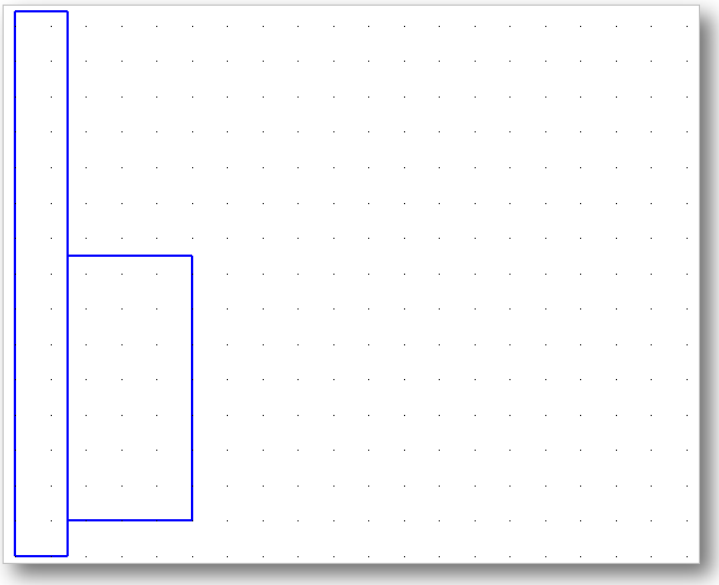
```
Options: Segments, Undo, Close, Enter to exit or
Specify next point» 3500
```

10. Use the **Close** option to close the rectangle:

```
Options: Segments, Undo, Close, Enter to exit or
Specify next point» c
```

"Closing the rectangle" means that the program draws the fourth segment for you.

11. Save your work by pressing **Ctrl+S** (Cmd+s on Macs). The drawing now contains two rectangles, representing the hallway and the storage room.



The hallway and a room placed in the drawing as rectangles

Unlike drafting on paper, you are drawing the floor plan 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 to use a scale ruler or to divide distances by a scale factor as in manual drafting.

USING ENTITY SNAPS

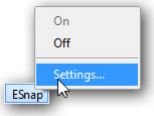
Command: EntitySnap
Aliases: es, esnap, os, osnap
Status bar: ESnap | Settings
Ribbon: --
Status bar: 

Next to be drawn is the conference room. Notice that its lower right corner starts at the storage room's lower left corner. You can take advantage of that geometry to start the rectangle precisely there using entity snaps.

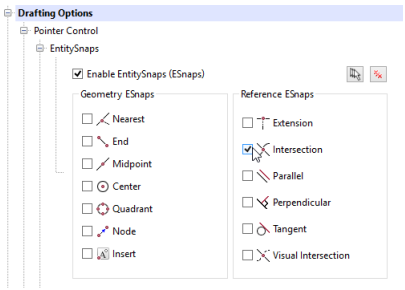
Entity snaps ("esnaps" for short) snap the cursor to specific geometric features, such as to the intersections of lines, the centers of circles, and the midpoints of lines. Here you use Intersection esnap. To ensure the Intersection esnap is turned on, follow these steps:

1. Click the **ESnap** button. Notice that it turns blue.

2. Right-click the **ESnap** button, and then choose **Settings** from the shortcut menu. This takes you to the entity snap section of the Options dialog box quickly.



3. In the Options dialog box, ensure that **Intersection** is checked. Turn off all other snap modes.



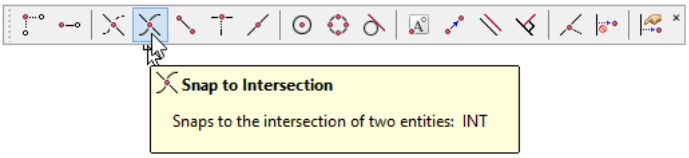
Selecting entity snaps in the Options dialog box

4. Click **OK** to exit the dialog box.

With Intersection esnap turned on, go on to drawing the rectangle.

ALL ABOUT ENTITY SNAPS

The Entity Snap toolbar illustrates all of the entity snaps available in the program: Left to right, these are:



Snap from snaps at a distance from geometry.

Inference point snaps to temporary points.

Virtual intersection snaps to the apparent intersection (used in 3D drawings).

Intersection snaps to the point where two entities cross.

Endpoint snaps to the endpoints of lines, polylines, arcs, and other open entities, including end points of planes and solids.

Extension snaps to the imaginary endpoint, if an open entity were to be extended that far.

Midpoint snaps to the midpoints of lines, polylines, and arcs.

Center point snaps to the center points of circles, ellipses, rings, polyarcs, and arcs.

Quadrants snaps to the 90-degree points (quadrants) of arcs and circles.

Tangent snaps to the point of tangency on circles and arcs.

Insertion point snaps to the insertion point of text, attributes, blocks, and shapes.

Node snaps to points.

Parallel snaps parallel to the geometry.

Perpendicular snaps perpendicularly to the nearest geometry.

Nearest snaps to the nearest geometry.

Off turns entity snapping off.

Entity Settings open the Options dialog box.

DRAWING RECTANGLES

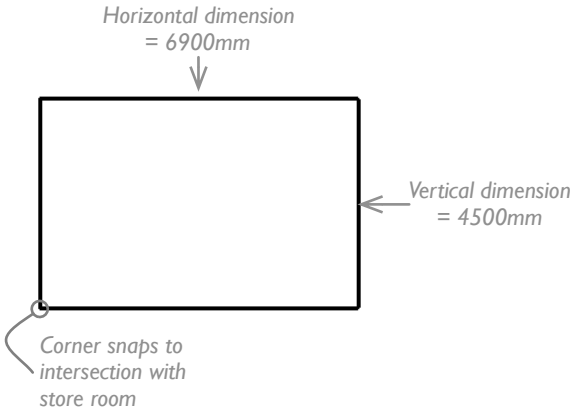
Command: Rectangle
Aliases: rec, rect, rectang
Menu bar: Draw| Rectangle
Ribbon: Home | Draw | Rectangle (Drafting and Annotation)
Toolbar:

You have drawn two rectangles with the Line command, one with absolute coordinates, the other with relative ones. The program has a command that draws rectangles directly, the **Rectangle** command.

There is a difference, however, in *what* the two commands draw: Line draws line segments, whereas Rectangle draws polylines. *Polylines* are connected lines. Move one segment and the entire polyline moves.

You will use the Rectangle command to draw the conference room.

1. Start the **Rectangle** command using one of the methods listed above.



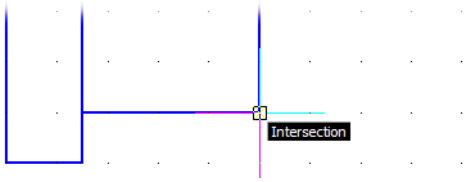
Aspects of drawing with the Rectangle command

2. At the prompt, move the cursor to the intersection indicated by the figure below

```
: rectangle
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Par-
allelogram, Thickness, line Width or
Specify start corner» (Click at the intersection)
```

3. Enter d to specify the dimensions of the rectangle:

```
Options: Area, Dimensions, Rotation or
Specify opposite corner» d
```

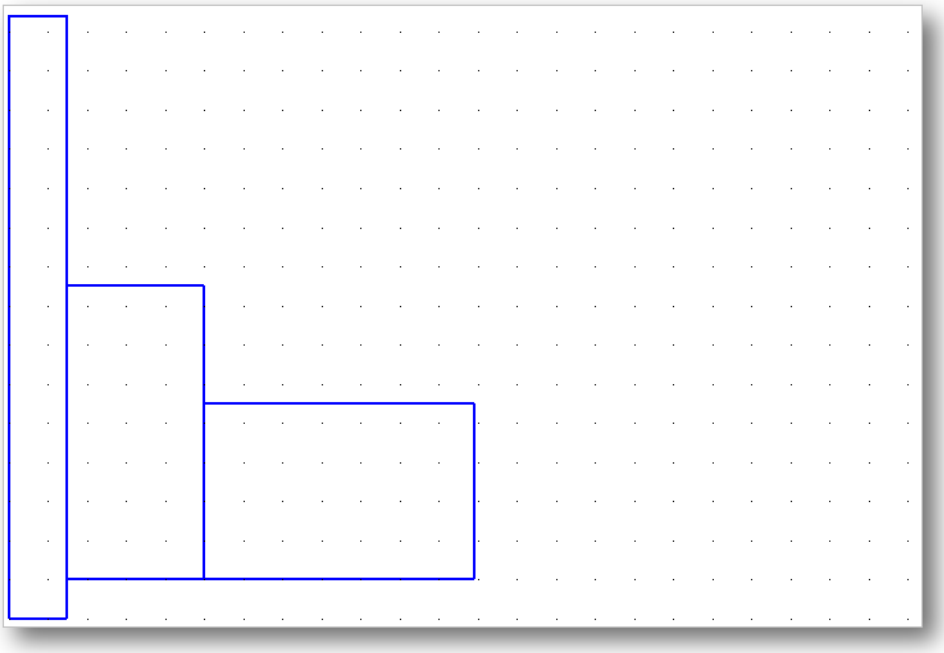


Locating the intersection with entity snaps

4. The dimensions of the rectangle are 6900mm horizontally and 4500 vertically:

```
Default: 10
Specify horizontal dimension» 6900
Default: 10
Specify vertical dimension» 4500
```

Notice that the program draws the rectangle.



The floor plan thus far

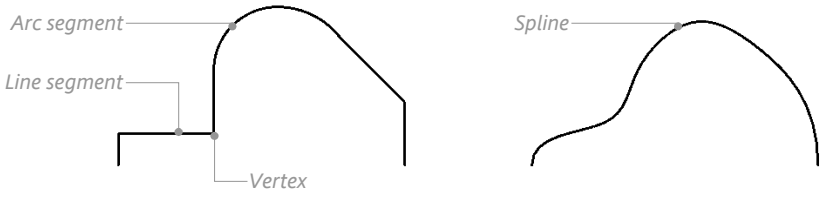
DRAWING POLYLINES

Command: PolyLine
Aliases: pl, pline
Menu bar: Draw| PolyLine
Ribbon: Home | Draw | Polyline (Drafting and Annotation)
Toolbar:

When you used the Rectangle command earlier in this chapter, you drew a rectangle made of polylines — lines connected together into a single object. As the prefix "poly" suggests, polylines are made up of many features — lines and arcs, or splines of varying widths — all connected as a single entity.

In addition to Rectangle, the program has a command specifically for drawing any shape of polyline. (It also has the special Polygon and Ring commands for creating regular polygons and filled circles from polylines.)

The PolyLine command is good for drawing entities like the ones illustrated below, such as a complex one made of lines and arcs, or of splines.



Aspects of polylines

SUMMARY OF RECTANGLE COMMAND OPTIONS

```
: rectangle
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Parallelogram, Thickness, line Width or
Specify start corner»  (Enter an option, or pick a point)
```

(NEW IN 2019) 3Corner constructs angled rectangles by specifying three corner points:

```
Specify first corner» (Pick the starting corner of the rectangle)
Specify second corner» (Pick a point to specify the horizontal width and the angle of the rectangle)
Specify third corner» (Pick a point to specify the vertical height of the rectangle)
```

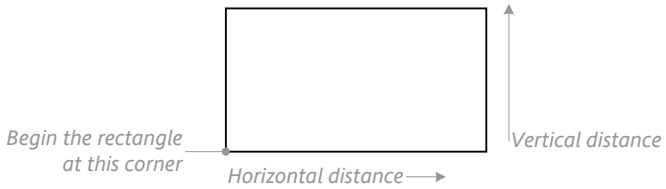
(NEW IN 2019) 3Point center constructs rectangles by specifying its center point, the midpoint of one side, and a corner point, kind of like constructing a circle by center, tangent, radius:

```
Specify center point» (Specify the center of the new rectangle)
Specify mid-point of side» (Specify the width of the rectangle by picking a point at the midpoint of one of the sides)
Specify corner point» (Specify the height of the rectangle by picking a corner point)
```

(NEW IN 2019) CEnter constructs the rectangle by specifying its center point and one corner point, kind of like constructing a circle by center, radius:

```
Specify center point» (Specify the center of the new rectangle)
Specify corner» (Specify the size of the rectangle by picking a point at one of its corners)
```

(COrner) Specify start corner specifies one corner of the rectangle using one of these input methods:

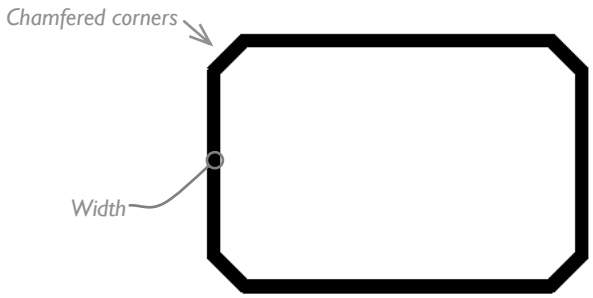


- Pick a point in the drawing using the cursor
- Enter coordinates at the keyboard
- Obtain a point from existing geometry using entity snaps

Chamfer specifies the two distances for bevelling the corners of the rectangle; the same size chamfer is applied to all four corners.

```
Specify first chamfer length» (Enter a distance)
Specify second chamfer length» (Enter a distance)
```

Rectangle Command: Options continued...

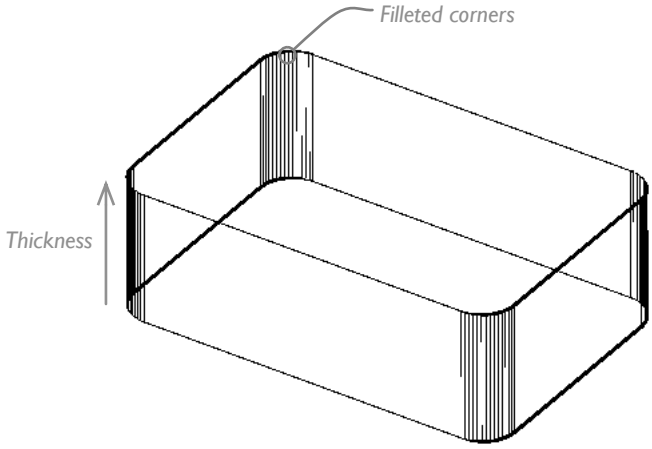


Elevation specifies the height of the rectangle above or below the x,y-plane; positive numbers locate the rectangle above the plane; negative ones below.

```
Specify elevation» (Enter a height)
```

Fillet specifies the radius of rounded corners; the same fillet is applied to all four corners.

```
Specify fillet radius» (Enter a radius)
```



(NEW IN 2019) Parallelogram constructs a rectangle by two points, and then uses a third point to specify the angle of the slanted sides:

```
Specify first corner» (Specify one corner of the rectangle)
Specify second corner» (Specify the opposite side of the horizontal width
of the rectangle)
Specify third corner» (Specify the vertical height and slant of the paral-
lelogram)
```

Thickness specifies the thickness of the rectangle (height in the z direction; see figure above):

```
Specify thickness» (Enter a thickness)
```

line Width specifies the width of the rectangle's lines:

```
Specify line width» (Enter a width)
```

Rectangle Command: Options continued...

OPPOSITE CORNER OPTIONS

The following options appear when you are prompted for the opposite corner:

```
Options: Area, Dimensions, Rotation or
Specify opposite corner»
```

Area specifies the size of the rectangle based on its area and the length of one side.

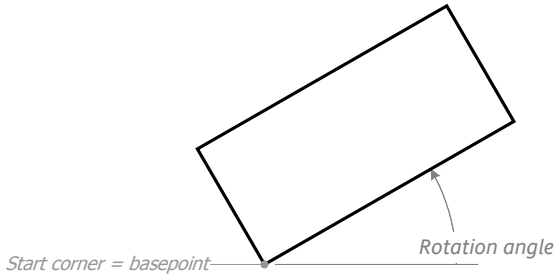
```
Specify total area» (Specify the area of the rectangle)
Options: Horizontal, or Vertical
Specify known dimension» (Enter H or V)
Specify horizontal dimension» (Enter a length)
```

Dimensions specifies the length and width of the rectangle, measured from the start corner:

```
Specify horizontal dimension» (Enter a length)
Specify vertical dimension» (Enter a width)
```

Rotation specifies the angle of the rectangle, measured counter clockwise from the positive x axis. The rectangle rotates about the start corner.

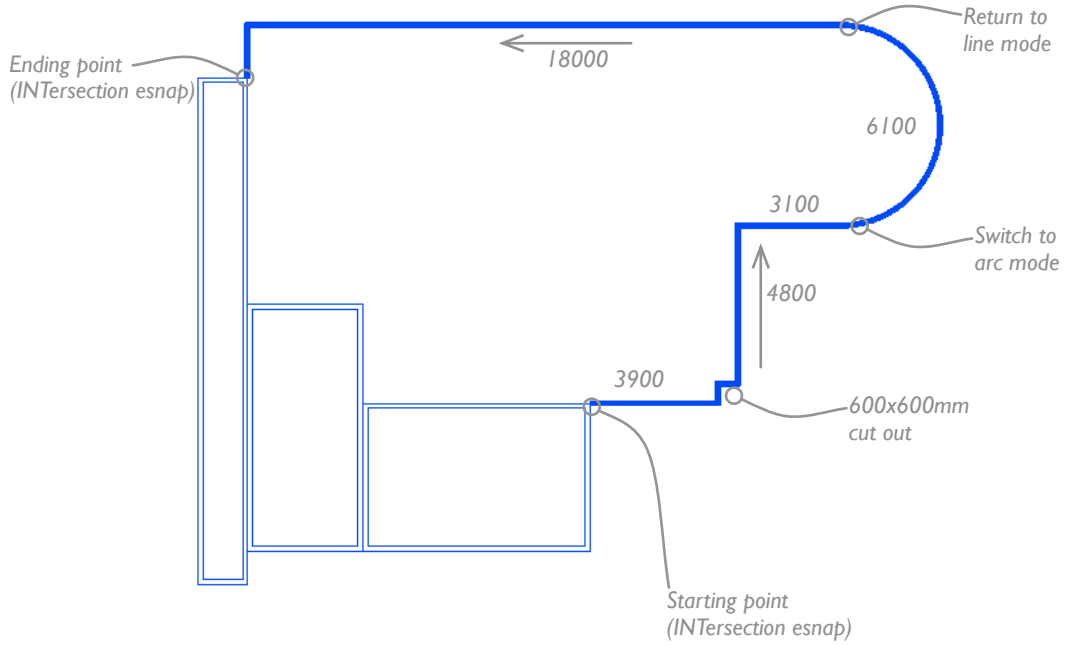
```
Options: Pick points or
Specify rotation» (Specify an angle)
Options: Area, Dimensions, Rotation or
Specify opposite corner» (Pick the other corner, or enter an option)
```



TIPS

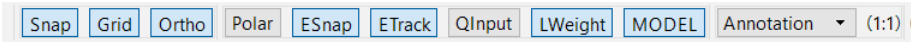
To change the thickness and elevation, use the **Modify** command.
To change the line width, use the **EditPolyline** command.
Chamfers and fillets cannot be changed.

You use the PolyLine command to draw the remaining external walls, shown in black by the figure below:



Drawing the remaining walls with polylines

1. Ensure the drawing aids are turned on: snap, grid, ortho, and esnap.



2. Start the **PolyLine** command using one of the methods listed above.

```
: polyline
```

3. Start at the conference room rectangle's upper right corner. The Intersection snap ensures a precise pick.

```
Options: Enter to continue from last point or
Specify start point» (Pick the upper right corner of the conference room)
```

4. Move the cursor to the right, and then enter 3900. This draws the first polyline segment 3900mm long.

```
Options: Arc, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» 3900
```

5. Continue drawing the polyline using the dimensions shown in the figure.

```
Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» 600

Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» 600

Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» 4800

Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» 3100
```

6. To draw the arc around the executive office, switch to arc drawing mode. To do so, enter **a** at the prompt, as follows:

```
Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» a
```

7. There are many ways to define an arc. Here, you will define the end point of the arc:

```
Options: Angle, CEnter, CLose, Direction, Halfwidth, Line, Radius, Through point,
Undo, Width, Enter to exit or
Specify arc end point» 6100
```

8. With the arc in place, you can switch back to line drawing mode. Enter **l** at the prompt:

```
Options: Angle, CEnter, CLose, Direction, Halfwidth, Line, Radius, Through point,
Undo, Width, Enter to exit or
Specify arc end point» l
```

9. You are almost done! Draw the final two polyline segments:

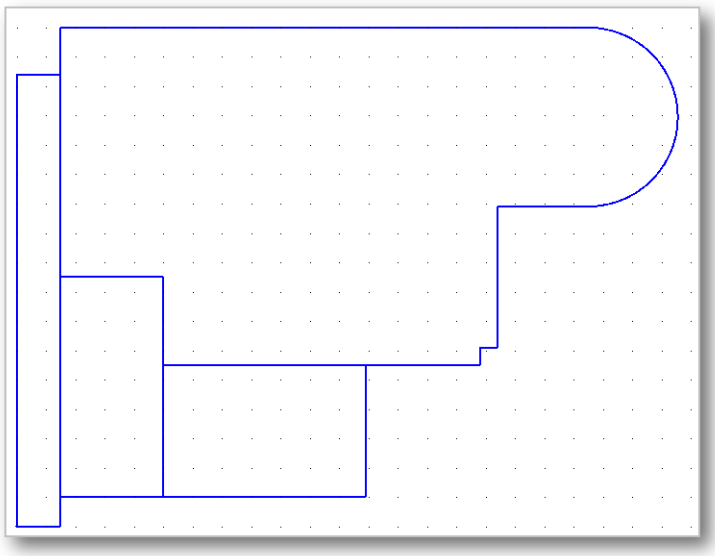
```
Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» 18000

Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» (Move the cursor down to meet the corner of the hallway, and
then use Intersection entity snap to complete the segment.)
```

10. Press **Enter** to exit the PolyLine command:

```
Options: Arc, Close, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» (Press Enter)
```

11. Press **Ctrl+s** to save your work (**Cmd+s** on Macs).

The drawing should now look like the figure illustrated below.



The floorplan drawn with lines, rectangles, and polylines

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 **EditPolyline** command.

Drawing Internal Walls

The interior walls are drawn on a layer different from the external ones. In the following section, you change layers, and then use yet another kind of line command to draw the interior walls.

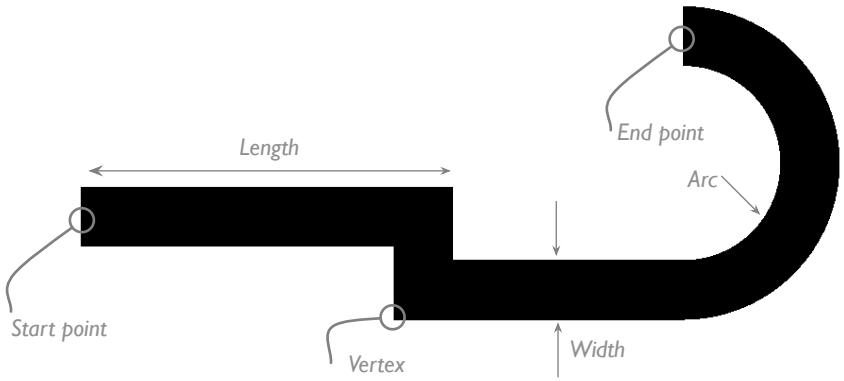
CHANGING LAYERS QUICKLY

By placing entities on different layers, you ensure that they automatically take on the properties of the layers. Properties include things like color, line style, and visibility. When you drew lines drawn on the Walls-Exterior layer, they were colored blue, automatically. When you make the Wall-Interior layer active, lines will be drawn light blue.

In the previous chapter, you used the Layer Manager dialog box to create new layers, and then to set Walls-Exterior as the active (or working) layer. You could use this dialog box to change the active layer, but I find the Layers toolbar more convenient.

SUMMARY OF POLYLINE COMMAND OPTIONS

Polylines consist of the following elements:



The PolyLine command has the following options for drawing polylines:

```
: polyline
Options: Enter to continue from last point or
Specify start point» (Press Enter, or pick a point.)
Options: Arc, Halfwidth, Length, Undo, Width, Enter to exit or
Specify next vertex» (Enter an option, or pick a point.)
```

Enter to continue starts the polyline at the end of the last drawn line, arc, or polyline.

Specify start/next vertex specifies the location of polyline's next *vertex* (endpoint of a polyline segment).

Arc draws polyline arcs using prompts similar to the Arc command; you can switch between line and arc mode.

Close joins the last endpoint with the starting point.

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

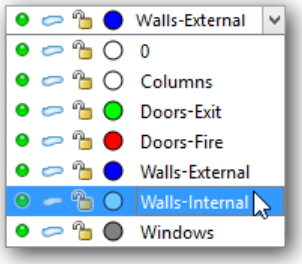
Length specifies the length of the segment in the manner similar to direct distance entry.

Undo undoes the last polyline drawing operation.

Width specifies the width of polyline segments; with different starting and ending widths, the program draws tapered polylines.

To make the Walls-Interior layer active, follow these steps:

1. Move the cursor over the layer droplist in a toolbar or on the ribbon, and then click. Notice the list of layer names.



Listing the names of layers in the drawing

2. Move the cursor down to **Walls-Interior**, and then click again. The layer droplist now displays Walls-Internal as the active layer.

"Walls-Interior" layer is the working layer

TIP If you prefer to use the keyboard, then you might find the CLayer system variable to be a quicker way to change layers. Enter **CLayer**, and then the name of the layer to be made active, like this:

```
: clayer
Enter new value for CLAYER: walls-internal
```

SUMMARY OF LAYER DROPLIST ICONS

The layer droplist prefixes each layer name with four icons:

On / Off — click the green dot to turn off layers; click the gray dot to turn them on again.

Thaw / Freeze — click the puddle icon to freeze layers; click the snow icon to thaw the layers.

Unlock / Lock — click the open padlock icon to lock layers; click the locked padlock to unlock.

Color — colored dot indicates the default color of layers (performs no action).

Off and frozen layers are invisible. Locked layers are visible but cannot be edited. Layer names in gray indicate frozen layers.

DRAWING WALLS WITH RICHLINES

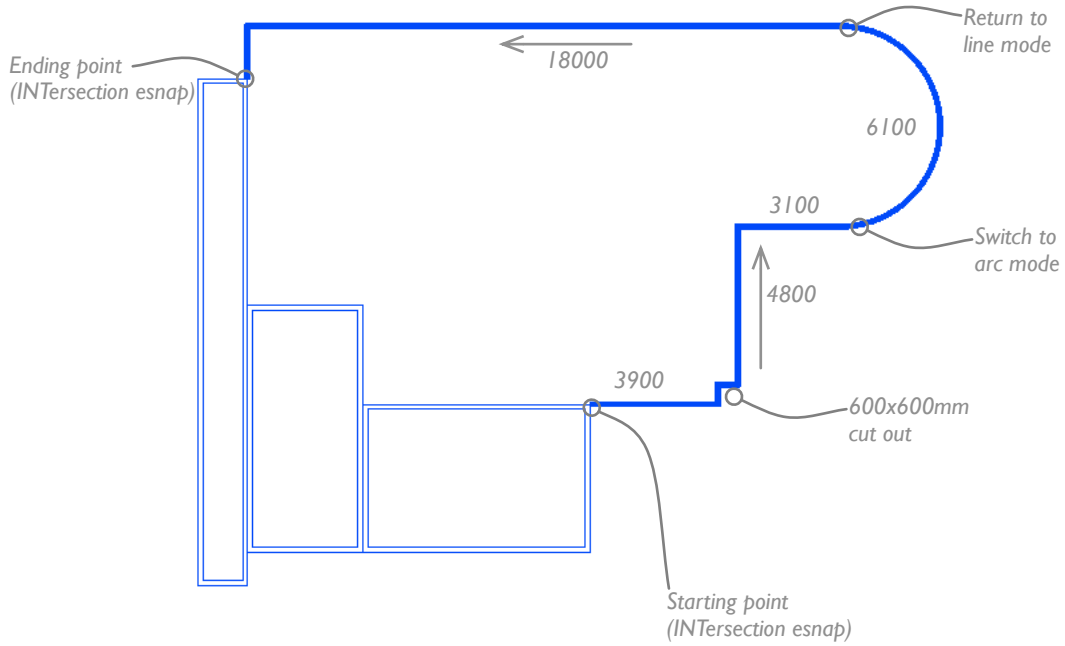
So far you have drawn walls with lines, rectangles, and polylines. There is one more way to draw straight lines in the program, the richline. Richlines are made of one or more parallel lines. The space between the lines can be filled with color, and the ends can be capped. You can define many kinds of rich lines, and apply them with *styles*.

There are two primary drawbacks to rich lines: they can only be straight lines, no arcs; they do not clean up their intersections automatically, where pairs of lines overlap.

Placing RichLines

Command: RichLine
Aliases: rl, ml, mline
Menu bar: Draw | RichLines
Ribbon: Home | Draw | RichLine (Drafting and Annotation)
Toolbar:

There are two interior walls you draw with rich lines. One defines the reception area; the other walls off the executive office. Let's do the executive office first. This is the path that the rich lines will trace:



Route that the rich lines take in this tutorial

1. Start the **RichLine** command with one of the methods listed above.
2. At the prompt, pick the intersection indicated by the figure.

```
: richline
Active settings: Justification = Top, Scale = 1, Style = STANDARD
Options: Justification, Scale, STyle or
Specify start point» (Pick a point at the intersection)
```

3. Move the cursor up to the upper wall.

4. Enter "per" and then press **Enter**.

```
Specify next point» per
```

Per is the abbreviation for "perpendicular," another one of the program's entity snaps. Entering it at the 'Specify next point:' prompt makes the esnap temporary; it is in effect for the next pick point only. (This is different from using the Options dialog box to set Intersection esnap, which remains in effect until you turn it off.)

5. The program asks "to," meaning where the lines should be drawn to:

```
to (Pick on the wall)
```

6. Press **Enter** to exit the command.

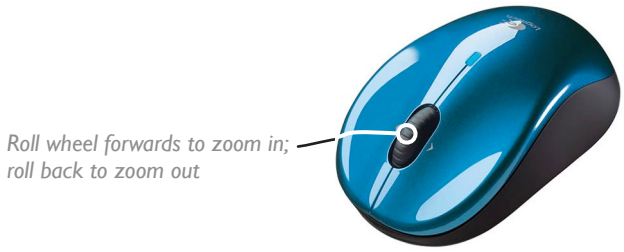
```
Options: Undo or
Specify next point» (Press Enter)
```

Zooming In and Out

The richline probably looks to you like any other line (or polyline) in the drawing — other than being light blue in color. In fact, it consists of two closely spaced lines, just 0.5mm apart.

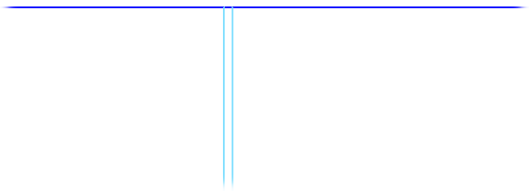
You can see them by zooming into the drawing:

1. Position the cursor over the richline.
2. Roll the mouse's scroll wheel forward. Under the cursor, the drawing should become larger.



Locating the roller wheel on a mouse

3. Keep rolling the wheel until you see two distinct lines, as illustrated below.



Zooming into a pair of lines

4. To return to the original view, enter the **Zoom Bounds** command as an alias:

```
: z b
```

(To zoom out, you could roll the scroll wheel backwards, but it is faster to enter "z b".)

Defining RichLine Styles

Command: RichLineStyle
Aliases: rls, rlstyle, rlinestyle, mlstyle
Menu bar: Format | RichLine Styles
Ribbon: Home | Annotations | RichLine Style (Drafting and Annotation)
Toolbar: 

Rich lines can consist of as many as 16 parallel lines located a specified distance from the center. In addition, each can have its own color and linestyle. Rich lines can also have the following optional properties:

Display Color — a solid fill color that floods the entire richline.

Junctions — a diagonal line at each intersection.

Cap ends — a line or arc at either end; they can be at any angle.

The default richline style consists of two parallel lines, each 0.5 mm from an imaginary center line. Interior walls are often 100mm thick, so the current richline of 0.5mm is way, way too thin.

SUMMARY OF RICHLINE COMMAND OPTIONS

```
: richline
Active settings: Justification = Top, Scale = 1, Style = STANDARD
Options: Justification, Scale, STyle or
Specify start point» (Pick a point)
```

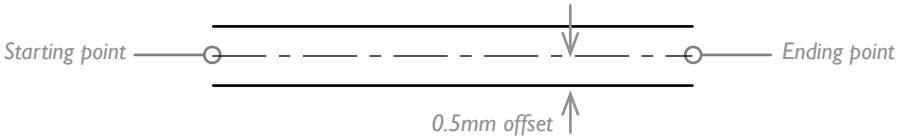
Justification aligns the rich lines to the right, left, or center of pick points. The options are named Bottom, Top, and Zero.

Scale enlarges or reduces the width of the richline by a scale factor. (The width of the richline is defined by its style.) Enter a number at the 'Enter RichLine scale' prompt:

- Smaller than 1.0 makes the richline narrower.
- 1.0 makes the richline the same width as defined by its style.
- Larger than 1.0 makes the richline wider.
- Negative numbers flip the richline about its center line.

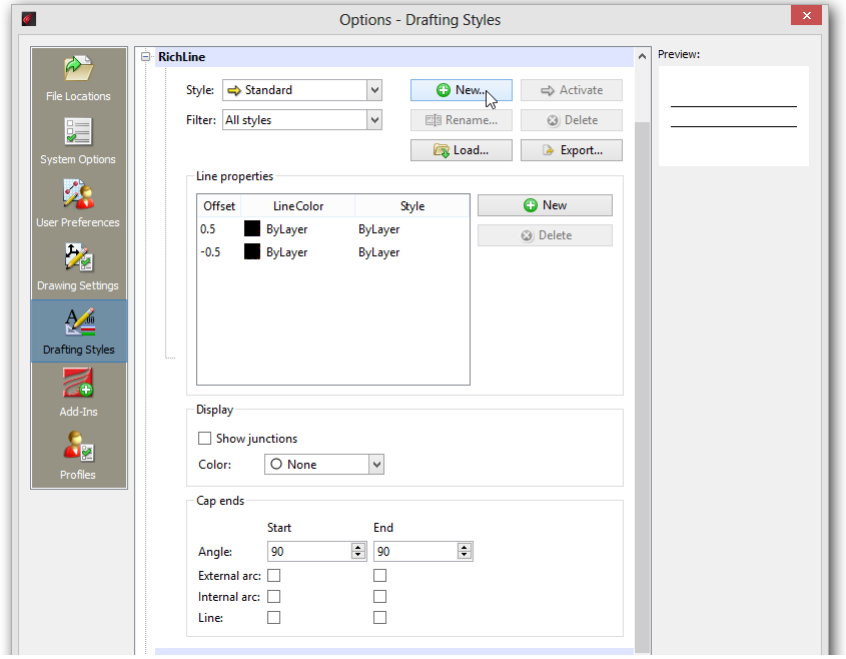
Style selects a predefined style for the richline. Styles are created and edited with the RichLineStyle command.

You change the width with the RichLineStyle command. (The center line path is defined by your pick points in the drawing.)



Aspects of a richline

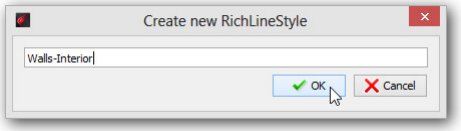
1. Start the **RichLineStyle** command using one of the methods listed above. Notice that the Options dialog box opens to the Drafting Styles | RichLine section, featuring a preview window to the right.



Defining richline styles through the Options dialog box

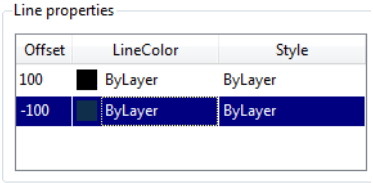
2. To create a new style, click 

New. Notice the Create New RichLineStyle dialog box.



Naming the new richline style

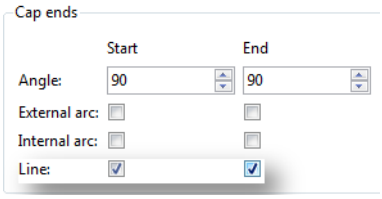
3. Change the name from STANDARD1 to "Walls-Interior", and then click **OK**.
4. With the new style named, you define two lines spaced 100mm apart, as follows:
   a. In the Line Properties section, under Offset, click 0.5, and then change it to **100**.



Specifying the distance between richline elements

b. Change the second line from -0.5 to **-100**.

5. Add end caps to the richline. In the Cap Ends section, check both boxes next to **Line**.



Specifying the richline end cap treatment

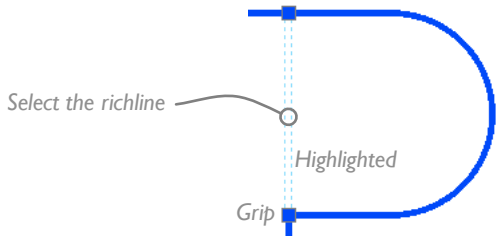
6. That's all you need to do for this richline style. Click **OK** to exit the dialog box.

Erasing Entities

Command: Delete
Aliases: e, erase, del
Menu bar: Edit | Erase *or* Modify | Delete
Ribbon: Home | Modify | Erase (Drafting and Annotation)
Shortcut: Del (**fn+Delete** on Mac)
Toolbar:

A flaw in rich lines means that you cannot simply apply new styles to existing lines, as you might to other entities that accept styles, such as text and dimensions. This means that you need to erase the existing richline, and then redraw it with the new style. You erase entities as follows:

1. Select the richline. Notice that it is highlighted, as illustrated below.



Selecting entities

Grips allow you to edit entities directly, as described in a later chapter.

Highlighting indicates the selected entities.

2. Press the **Del** key. The entity disappears from the drawing.

Oops! Unerasing Entities

Command: U
Menu bar: Edit | Undo
Ribbon Undo (Quick Access toolbar)
Shortcut: Ctrl+z (Cmd+z on Macs)
Toolbar:

When you erase objects by accident, you can return them to the drawing with the U command, as follows:

```
: u
```

TIP To undo an undo, use the **Redo** command, or press **Ctrl+y** (Cmd+y on Macs).

Replacing RichLines

With the unwanted richline removed, you now redraw the wall defining the executive office with the newly-defined richline style.

1. Start the **RichLine** command.

```
: richline
```

2. Notice that the style is set to "Standard." Enter **st** to change the style, as follows:

```
Active settings: Justification = Top, Scale = 1, Style = STANDARD
Options: Justification, Scale, STyle or
Specify start point» st
```

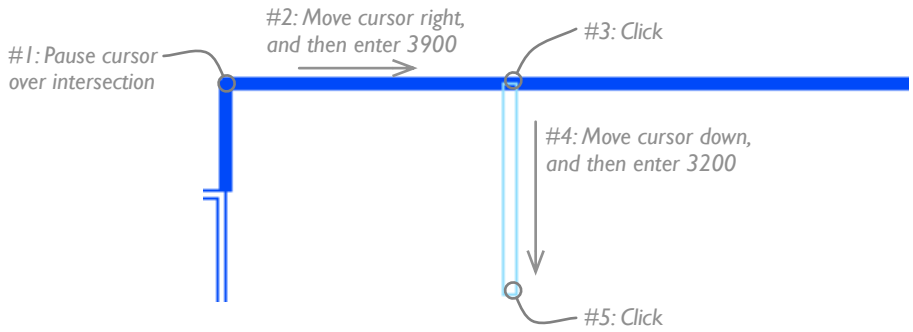
3. Enter "walls-interior" for the new style name:

```
Options: ? to list or
Specify name» walls-interior
```

4. Draw the richline as before, from the intersection to the perpendicular of the upper wall. Notice that the lines are 100mm apart.

```
Specify start point» (Pick point at intersection)
Specify next point» per
to (Pick on the wall)
Options: Undo or
Specify next point» (Press Enter to end the command)
```

5. Add the wall that defines the reception area. (If style is not set to "walls-interior," you may need to enter the **STyle** option again.



Starting richlines at an offset distance

To repeat the previous command, just press **Enter**:

```
: (Press Enter to repeat the RichLine command)
Active settings: Justification = Top, Scale = 1, Style = WALLS-INTERIOR
```

6. Use direct distance entry to locate the 3900mm-offset, as follows:
   a. Position the cursor over intersection #1.
   b. Move the cursor to the right, and then enter the offset distance, **3900**.

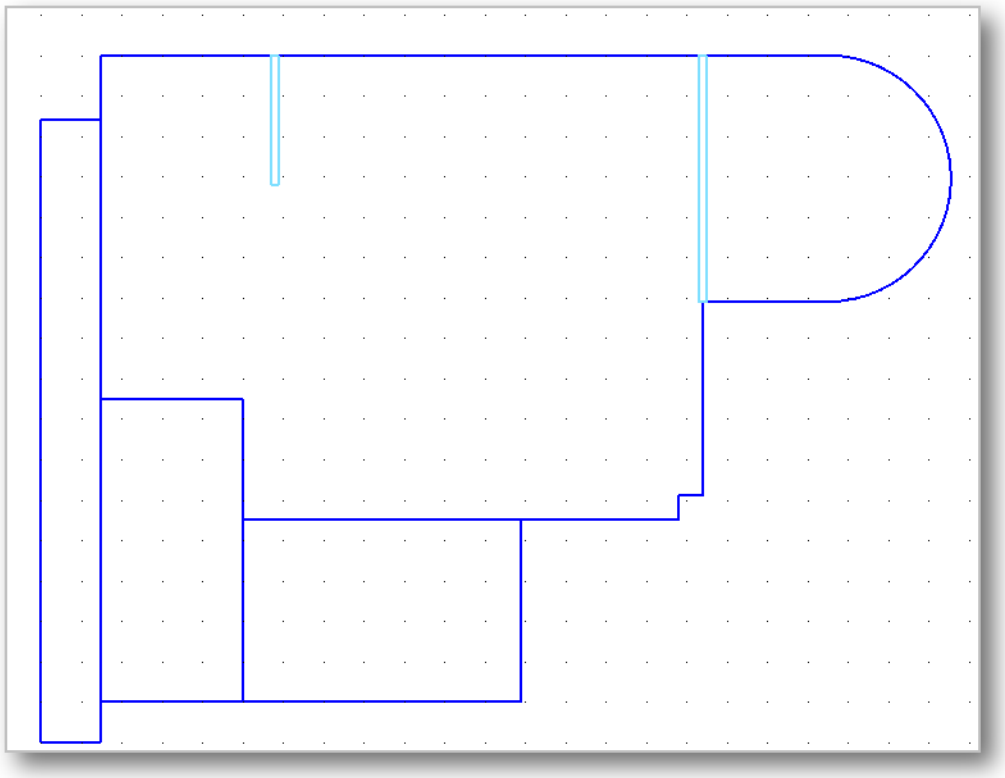
c. Click to position the starting point of the richline.

```
Options: Justification, Scale, STyle or
Specify start point» (Position cursor at intersection #1; do not click!)
(Move cursor right:) 3900 (And then click to start the richline)
```

d. Move the cursor down, and then enter the length of the wall, **3200**.

e. Click to position the endpoint of the richline.

7. Press **Enter** to end the command. The result should look like the figure below.
8. Press **Ctrl+s** to save your work (**Cmd+s** on Macs).



The drawing thus far

Thickening the Look of Walls

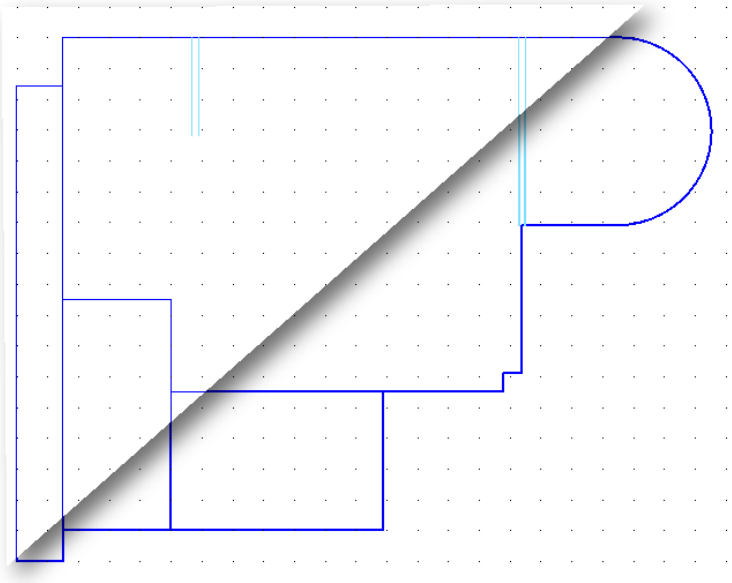
You have drawn wall lines four ways now, with the Line, Rectangle, PolyLine, and RichLine commands. Only the rich lines have the correct wall thickness. In this section, you learn how to make walls thicker when they are made of lines and polylines.

CHANGING THE LINEWEIGHT

Command: Lineweight
Aliases: lw, lweight
Menu bar: Format | Line Weight
Ribbon: Home | Properties | LineWeight (Drafting and Annotation)
Toolbar: --
Status bar: LWeight

Lineweights change the visual width of lines and all other entities. They don't actually thicken entities. I find it easier to view drawings, however, when lineweights are turned on, so I always keep them on.

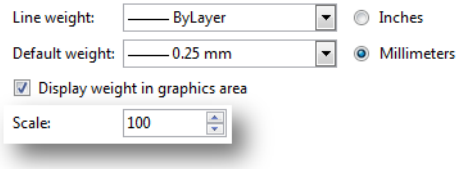
To turn on lineweights, click the **LWeight** button on the status bar. Entities should thicken, as illustrated by the composite image:



Upper left: *Regular lines of "0" thickness*
Lower right: *Lines fattened through lineweights*

Lines can look too thin or too thick. You can change the relative thickness, as follows:

1. Right-click **LWeight** on the status bar, and then choose **Settings** from the shortcut menu.
2. In the Options dialog box's Line Weight section, change the value of **Scale**.



Specifying lineweight parameters

3. Click **Apply** to see if the weights are now appropriately scaled.
4. Click **OK** to exit the dialog box.

WIDENING POLYLINES

Command: EditPolyline
Aliases: pedit, edpl, edpline, pe, polyedit
Menu bar: Modify | Entity | PolyLine
Ribbon: Home | Modify | Edit Polyline (Drafting and Annotation)
Toolbar: 

Unlike lineweights, the widths of polylines are real. You can assign the width while drawing them with the PolyLine command, or else change the width afterwards with the EditPolyline command. Follow these steps:

1. Start the **EditPolyline** command using one of the methods listed above.

```
: editpolyline
```

2. Choose the long polyline.

```
Options: Multiple or
Specify polyline» (Pick the polyline)
```

3. Enter **w** to choose the Width option.

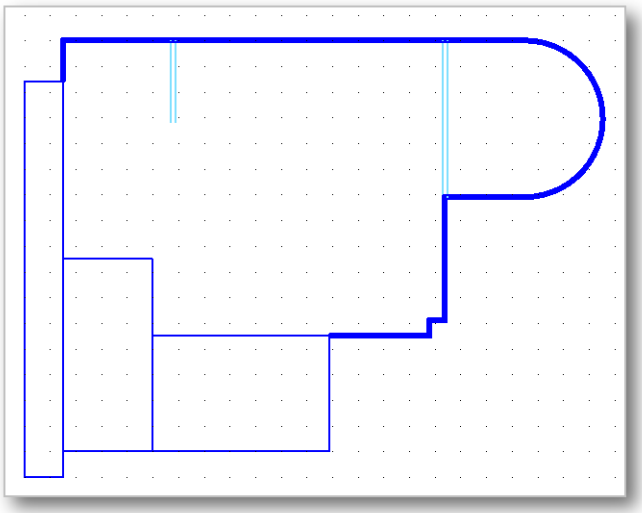
```
Options: Close, Decurve, Edit vertex, Fit, Join, Linegen, Spline, Undo, Width, or
eXit
Specify option» w
```

4. Specify a new width. Outside walls are typically 150mm, so enter **150**:

```
Specify new width» 150
```

5. Press **Enter** to end the command. Notice that the polyline instantly thickens.

```
Specify option» (Press Enter to end the command)
```



Polylines looking wider through the application of width

6. Press **Ctrl+s** to save the drawing (Cmd+s on Macs).

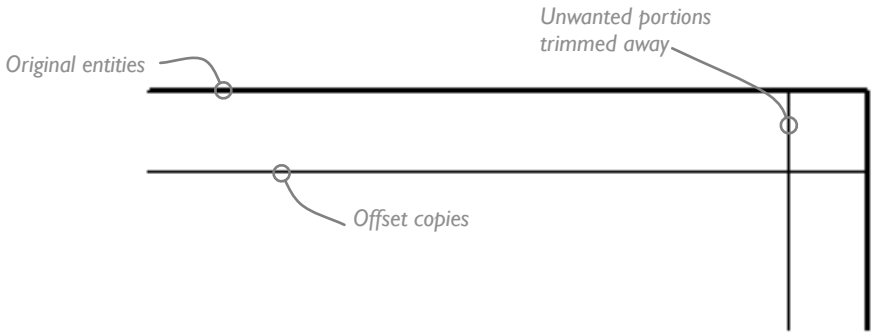
OFFSETTING LINES

Command: Offset
Aliases: o, qoffset
Menu bar: Modify | Offset
Ribbon: Home | Modify | Offset (Drafting and Annotation)
Toolbar:

While the EditPolyline command nicely thickens polylines, it make them solid filled; they are not hollow, like the rich lines. You can create an effect similar to rich lines by using the Offset and Trim commands:

Offset creates parallel entities.

Trim cuts away the excess where entities overlap.



Offsetting and trimming lines

In this tutorial, you apply the Offset command to the polyline drawn with the Rectangle command and to the rectangle drawn with the Line command — because the Offset command works slightly differently with polylines and lines.

1. Start the **Offset** command using one of the methods listed above.

```
: offset
```

2. Specify the offset distance, 150mm.

```
Active settings: Delete source=No Layer=Source OFFSETGAPTYPE=0
Default: 1
Options: Delete, DIstances, destination Layer, Through point or
Specify distance» 150
```

3. Select the rectangle representing the conference room, which was drawn with the PolyLine command.

```
Default: Exit
Options: Exit, Undo or
Specify source entity» (Pick the polyline)
```

4. Indicate on which side you wish the parallel copy made; for this tutorial, click inside the conference room.

```
Default: Exit
Options: Both sides, Exit, Multiple, Undo or
Specify side for destination» (Pick inside the polyline)
```

Notice that the program draws parallel polylines instantly.

SUMMARY OF EDITPOLYLINE COMMAND OPTIONS

The **EditPolyline** command has very many options; here is an overview of the important ones:

```
: editpolyline
Options: Multiple or
Specify polyline» (Pick a polyline.)
```

Multiple works with more than one polyline at a time.

Specify polyline selects the polyline to be edited; select a line or an arc, the program offers to turn it into a polyline.

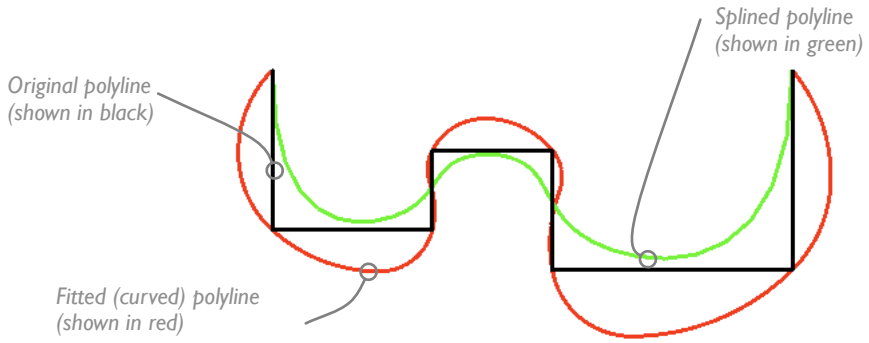
```
Options: Close, Decurve, Edit vertex, Fit, Join, Linegen, Spline, Undo,
Width, or eXit
Specify option» (Enter an option.)
```

Close closes the polyline: a segment is drawn between the last and first endpoints.

Decurve removes curve information from fitted and splines polylines.

Edit vertex edits the vertices: adds, changes, and removes vertices, and changes the widths in individual segments. The edit vertex options are the following: Break, Insert, Move, Next, Previous, REgen, Straighten, Tangent, Width, and eXit.

Fit changes all straight line segments to curves.



Join joins other polylines and lines to the current polyline; other entities must be touching.

Linegen (short for "linestyle generation") determines how linestyles are treated at vertices, continuous or dashed.

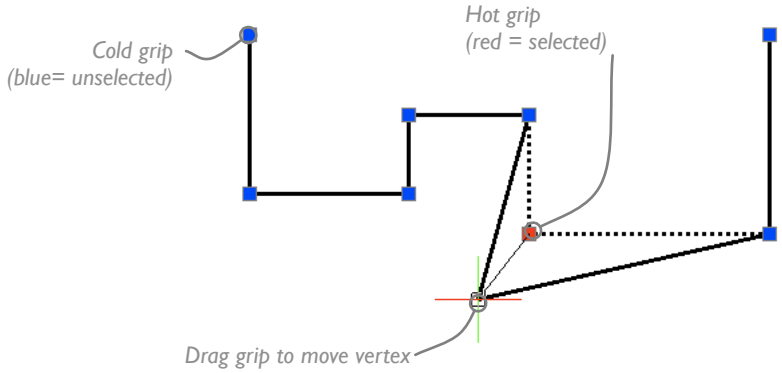
Spline changes the polyline into the equivalent of a spline curve.

Undo undoes the last editing changes.

Width changes the width of the entire polyline; constant width only, no tapers.

eXit exits the command.

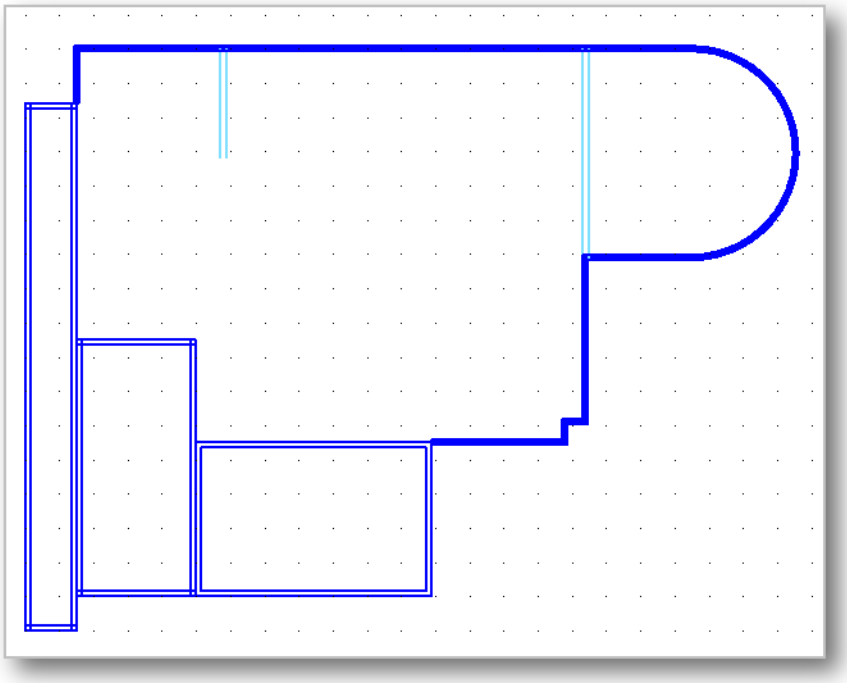
TIP Polylines can also be edited with the Properties palette, as well as directly through grips editing.



5. Now carry on to make offsets of the storage room and hallway, which were drawn with the Line command. This will be more work, because you have to select all eight lines making up the two rooms, one at a time.

```
Specify source entity» (Select a line)
Specify side for destination» (Pick inside the rectangle)
```

. . . and repeat seven more times until the drawing looks like this:



Lines doubled through offsetting

SUMMARY OF OFFSET COMMAND OPTIONS

```
: offset
Active settings: Delete source=No Layer=Source OFFSETGAPTYPE=0
Default: 1
Options: Delete, DIstances, destination Layer, Through point or
Specify distance» (Enter an option.)
```

Delete deletes the original after making the offset copy.

DIstances specifies multiple offsets.

destination Layer determines if offset copies are placed on the active (current) layer or remain on the source's layer.

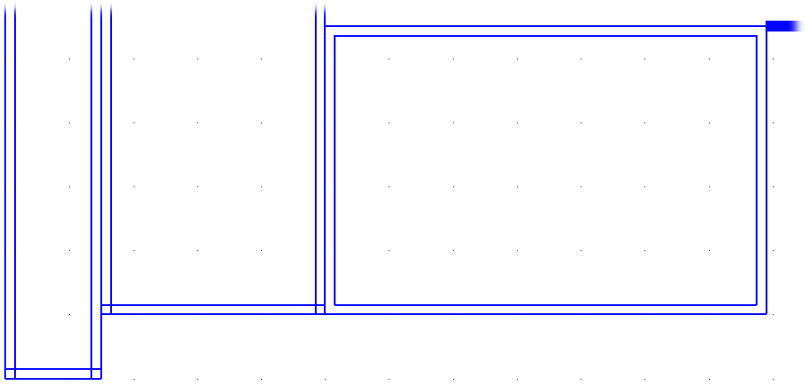
Through point places the offset copies at the cursor's pick point.

Specify distance places the offset distance.

Trimming Lines

Command: Trim
Alias: tr
Menu bar: Modify | Trim
Ribbon: Home | Modify | Trim (Drafting and Annotation)
Toolbar: 

The drawback to applying offsets to lines is overlapping lines. I have enlarged part of the drawing to illustrate the problem more clearly:



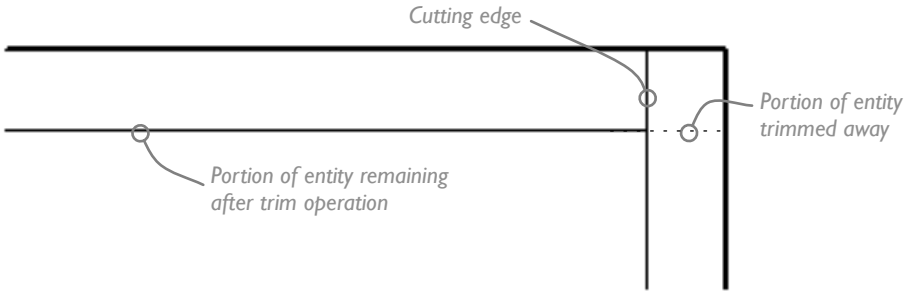
Offset lines needing trimming

The rectangle to the right has clean intersections, because it was drawn with the polyline; those to the left need cleaning up, because they were drawn with lines.

The solution to the mess is to trim the overlaps, cleaning up the drawing with the Trim command. There are two types of entities you select during the command:

Cutting edges are entities that determine where other entities are trimmed.

Trimmed edges are the portions of entities to be removed.



Line ends to be trimmed

Entities can be both cutting edges and trimmed.

1. Start the **Trim** command.

```
: trim
```

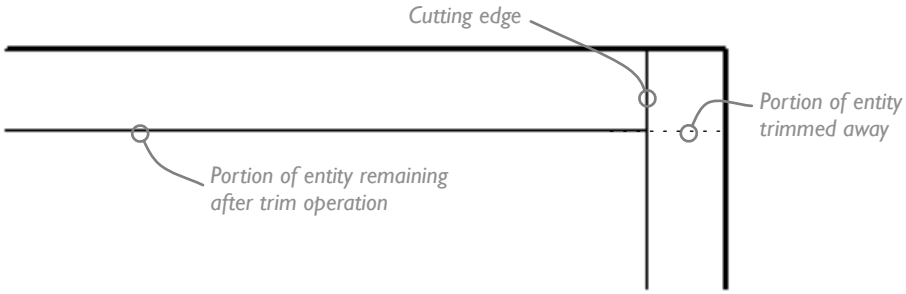
2. Press **Enter** to select all entities in the drawing as cutting edges. This is a safe move, because cutting edges are not modified by this command.

```
Active settings: Projection=CCS, Edge=None
Specify cutting edges ...
Options: Enter to specify all entities or
Specify entities» (Press Enter to select all the entities in the drawing)
```

TIP Note that richline entities are not supported by this command.

3. One by one, pick the portions to be trimmed. *Important!* Pick the part of the entity you want removed. If you make an error, use the Undo option.

```
Options: Crossing, croSsline, Project, Edge, eRase, Undo or
Specify segments to remove» (Select the portion to remove)
```



Cutting edge determines where line is trimmed

TIP Because it can be difficult to pick small entities accurately, employ these tricks:

- Zoom into the area where you make the trims. (Roll the scroll wheel.)
- Turn off **Snap** on the status bar for unrestricedt cursor movement.

SUMMARY OF TRIM COMMAND OPTIONS

```
: trim
Active settings: Delete source=No Layer=Source OFFSETGAPTYPE=0
Active settings: Projection=CCS, Edge=None
Specify cutting edges ...
Options: Enter to specify all entities or
Specify entities» (Press Enter to select all entities.)

Options: Crossing, croSsline, Project, Edge, eRase, Undo or
Specify segments to remove» (Select portion to remove.)
```

Specify Entities selects the entities that will be used to indicate trim points; it is safe to select all entities in the drawing.

Crossing selects entities within a crossing rectangle selection set.

Crossline selects entities with fence selection mode, and then trims entities along fence line.

Edge trims to extended cutting edges.

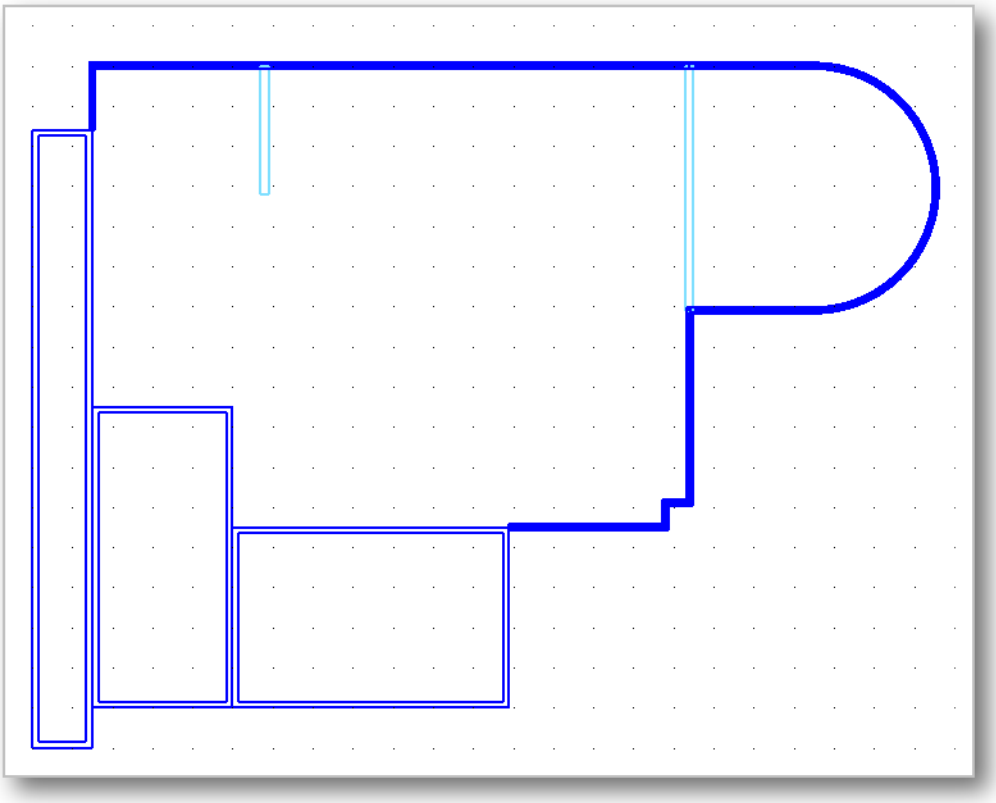
Erase erases unneeded entities without exiting this command.

Project determines the projection mode as CCS (uses the x,y-plane of the active custom coordinate system), None, or View.

Undo undoes the last trim operation.

4. Continue with the other lines needing trimming. When done, press **Enter** to exit the command.

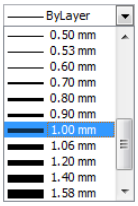
```
Specify segments to remove» (Press Enter to end the command)
```



5. Press **Ctrl+s** to save the drawing (**Cmd+s** on Macs).

TIPS You can assign lineweights individually to layers and entities. For layers, use the LineWeight column in the Layer Manager dialog box (Layer command). For entities, use the Lineweight droplist, as follows:

1. Select the entities.
2. From the Lineweight droplist, choose a lineweight value.



Summary

Congratulations! You've drafted your very first drawing using a computer.

In this chapter, you learned how to represent walls by lines, polylines, and richlines — thereby learning the pros and cons of each type of entity. You also learned how to make them look more like walls by giving them styles, weight, width, and parallel lines.

In the next chapter, you print the drawing on paper.

Notes

CHAPTER 06

Printing Drawings

In This Chapter

- Printing drawings
- Previewing drawings before printing them
- Reviewing detailed print options
- Creating page layouts and print styles

The most common result of a CAD drawing is to be printed on paper or electronically. In this chapter, you learn how to print, preview, and export drawings by a variety of methods.

Because the printing interface of MacOS is so different from Linux and Windows, this chapter places some of the Mac printing commands in separate sections.

KEY TERMS IN THIS CHAPTER

Default printer specifies the printer used by the program when no other printer is specified.

Dither simulates more colors through combinations of colors.

Grayscale prints colors as shades of gray.

Page layout refers to preset print settings.

Plot refers to an older term for printing drawings on paper.

Plotter refers to an old style of CAD printer that used pens to draw lines.

Preview shows how the drawing will fit the paper before printing.

Print prints drawings on paper.

Print range determines which part of the drawing is printed.

Print scale specifies the size of the drawing relative to the paper.

Print style defines how entities are printed.

Print to file sends the drawing to a file instead of a printer.

Quick print prints drawings scaled to fit the paper using the default printer.

Screening prints lines with less ink to make them look faded.

Virtual pen simulates pens in non-pen printers.

COMMANDS

Command	Windows & Linux Shortcuts *	Mac Shortcuts	Menu Selections **
PageLayout	pagesetup	pagesetup	File | Page Layout Manager
Preview	pre	pre	File | Print Preview
Print	Ctrl+p, plot	Cmd+p, plot	File | Print
PrintStamp	...	...	...
PrintStyle	plotstyle	plotstyle	Format | Print Style

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- **ImportSTL** command imports stereolithography files (*.stl*) and inserts them into the drawing as 3D solids.
- The following file-related functions are planned to be added to ARES Commander 2019 service pack 1:
  - » Import IFC files as underlays
  - » Import Revit as underlays
  - » Export drawings as RFA (Revit family) files

Printing the Office Drawing

Command: Print
Alias: plot
Menu bar: File | Print
Ribbon: Print (Quick Access toolbar)
Shortcut: Ctrl+P (Cmd+P on Macs)
Toolbar:

The Print command sends the drawing to printers, plotters (oversize printers), and files. It can print a drawing "quickly" or with a great deal of control by you:

Quick Print — prints the current view of the drawing fitted to the default paper size using the operating system's default printer.
Best suited for draft printouts, because the drawing is not scaled.

Manual Setup — prints after you choose among all of the settings available in the Print dialog box.
Best suited for having the greatest control over print jobs.

Use Previous Settings — prints using the same settings from the previous print job. This option works only after a manual setup.
Best suited for repeated printouts of the same drawing.

Use Page Layout — prints using a named Page Layout, previously defined with the Page-Layout command or imported from page layouts stored in other DWG, DXF, and DWF files. *Page layouts* memorize print settings.
Best suited for assigning corporate or client standards to print jobs.

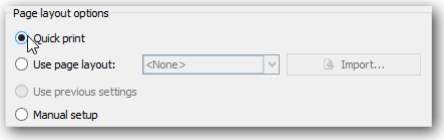
The tutorials in this chapter require that your computer be attached to a printer. This chapter uses the *office layout.dwg* file you completed in the last chapter.

Important Note Printing on MacOS is sufficiently different from Windows and Linux that the instructions regarding the Mac version are provided separately later in this chapter

MAKING DRAFT PRINTS (WINDOWS AND LINUX)

In this first tutorial, you produce a draft printout of the office layout drawing through the program's Quick Print option.

1. Start the **Print** command using one of the methods listed above. Notice the Print - Model dialog box.
2. Choose **Quick Print** mode.



Choosing the Quick Print option

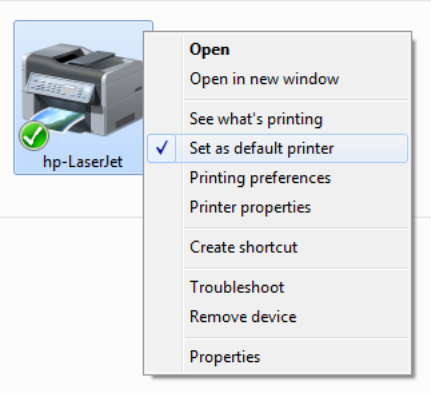
SELECTING THE DEFAULT PRINTER

Quick Print mode uses the printer designated as the *default printer* by the operating system, and not the printer assigned to the sheet (if any). The default printer is the one that is used by the program for printing when no other printer is specified.

The only way to change the default printer is through the operating system; it cannot be set by the program. Here are the steps to changing the default printers. Then next time you use Quick Print mode, the program will use the new default printer.

CHANGING THE DEFAULT PRINTER IN WINDOWS 10

1. On the Windows taskbar, click the **Start** button.
2. Start typing **Printers**.
3. From the "Best Match" column, choose **Devices and Scanners**.
4. In the Devices and Printers dialog box, the green check circle indicates the default printer. Click the icon of the printer you wish to make the default, and then right-click it.



5. From the shortcut menu, choose **Set as Default Printer**.
6. Close the dialog box.

WINDOWS 8.X

1. Swipe horizontally from the right edge of the screen.
2. Choose **Search**. (Choosing "Devices" does not work!)
3. In the search field enter **Devices and Printers**.
4. From the list returned by the search, select **Settings.**
5. In the Settings dialog box, select **Devices and Printers.**
6. In the Devices and Printers dialog box, the green check circle indicates the default printer. Click the icon of the printer you wish to make the default, and then right-click it.
7. From the shortcut menu, choose **Set as Default Printer**.
8. Close the dialog box.

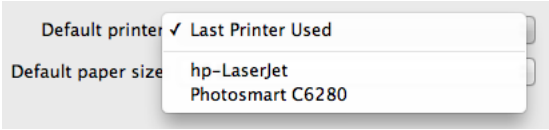
WINDOWS 7

1. On the Windows taskbar, click the  **Start** button.
2. Choose **Devices and Printers**. (In Vista and earlier, choose **Printers**.)
3. In the Devices and Printers dialog box, the green  check circle indicates the default printer. Click on the icon of the printer you wish to make the default.
4. Right-click the printer icon, and then choose **Set as Default Printer** from the shortcut menu.
5. Close the dialog box.

MACOS

On Mac computers, the default printer is usually the one that was used last. But if you want to make one specific printer the default for every print job, then follow these steps:

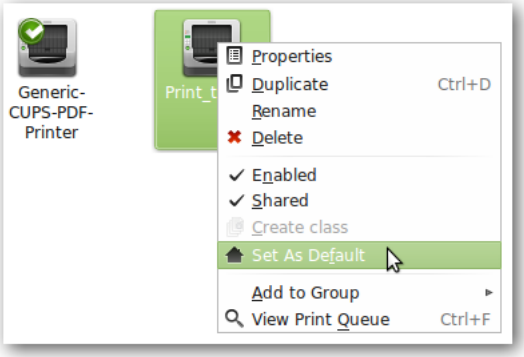
1. In the dock, choose **System Preferences**.
2. Click **Print & Scan**.
3. From the **Default Printer** droplist, choose the printer you wish to be the default one.



4. Close the dialog box.

LINUX

1. On the Linux taskbar, click the **Menu** button.
2. In the search field, enter "printers", and then choose the **Printing** application.
3. Right-click the icon of the printer you wish to make the default, and then choose **Set as Default Printer**.

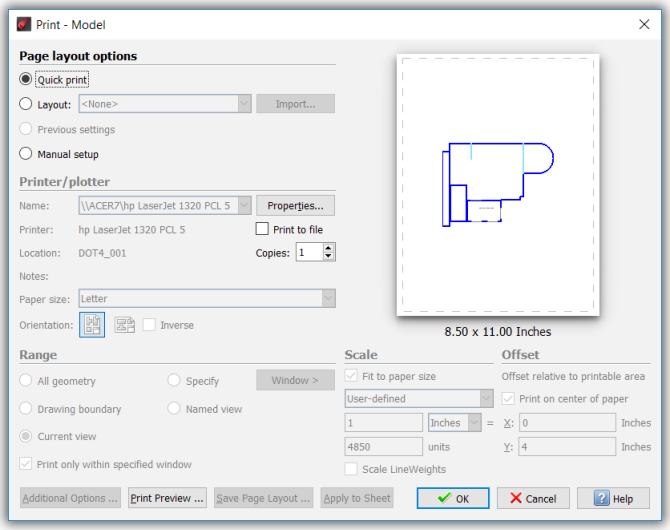


4. Close the dialog box.

Notice that nearly all options become grayed out, because the program takes over the task of choosing printing options.

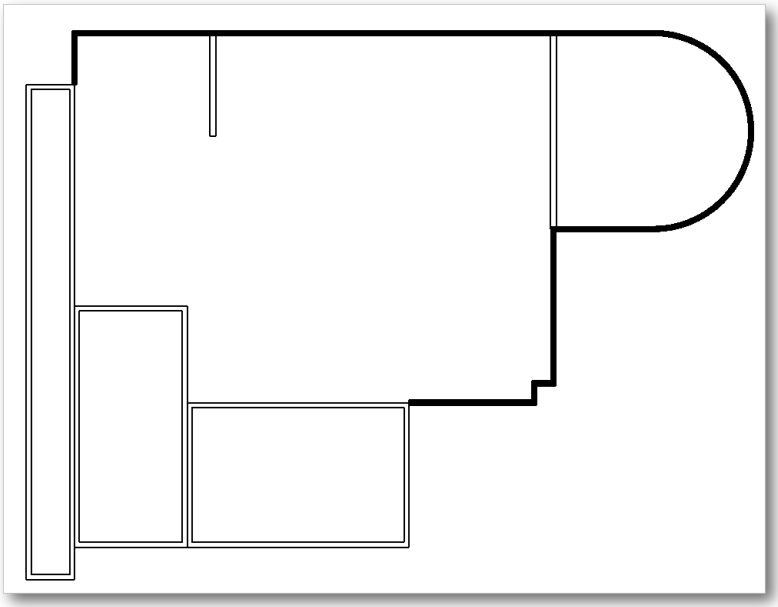
Quick Print mode sets the following print options; all others are grayed out. The settings are designed to ensure that the current view of the drawing is plotted to fit the size of the paper that's in the default printer.

Print Setting	Value	Comment
Printer/plotter Name	Default printer	Default determined by operating system
Paper Size	Default	Paper supplied by the default printer
Print Scale	Fit to Paper Size	Drawing cannot be scaled
Print Range	Current View	Print view cannot be specified



Dialog box with options grayed out in Quick Print mode

3. Click **OK**. Within a few moments, the drawing should emerge from the printer looking something like this.



Quickly printed drawing

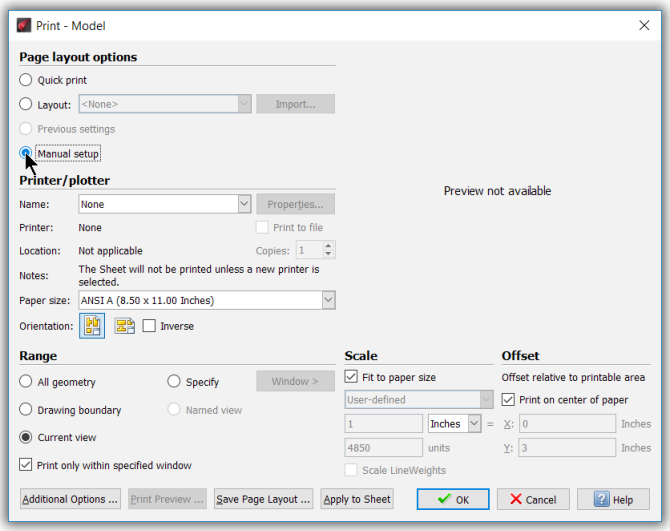
PRINTING THE OFFICE DRAWING MANUALLY (WINDOWS AND LINUX)

What Quick Print gives you in speed, it takes away in control. For example, it does not let you specify a scale factor, choose a different printer, or select another size of paper.

When you change the printing mode to Manual Setup, however, you can arrange the print to meet your needs. You can choose the printer, the type of paper, the scale factor, and the drawing view.

Follow these steps to set the options in Manual Setup mode:

1. Start the **Print** command
2. In the Print dialog box choose **Manual Setup**. Notice that most options become available that previously were grayed-out.



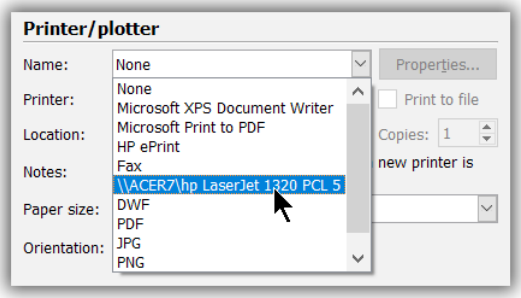
Print options available during manual setup

Here is a guide to all of the sections of this dialog box:

Printer/Plotter Options

The very first thing you need to do is select a printer, because this affects many other options in the dialog box. The program reads information provided by the operating system about the printer, and then presents some of it in the dialog box.

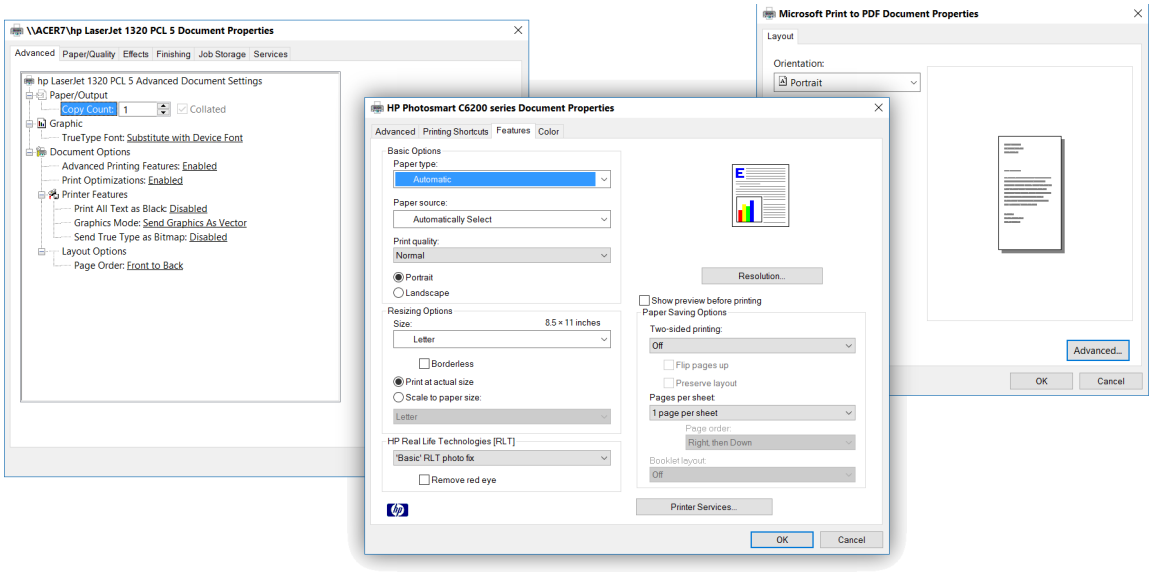
In the Printer/Plotter section, click the **Name** droplist. Notice the names of printer drivers and file formats. Some printer drivers are installed by the operating system, such as HP LaserJet and Adobe PDF. Others are *formats* provided by the program, such as DWF and JPG. You learn about these formats later in the chapter.



Choosing the printer from the list of available printers

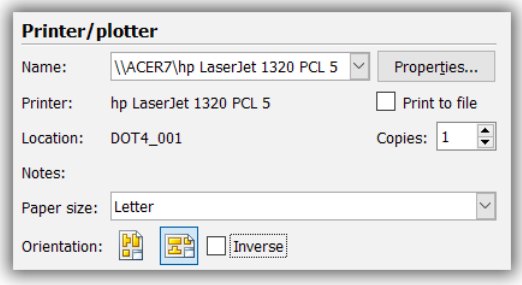
If you need to specify other parameters, such as the type of paper, the printer resolution, or two-sided printing, then you need to click the **Properties** button and make changes in the printer's Properties dialog box.

The dialog box you see depends on the printer model, and so below I should the Properties dialog boxes for several different printers in Windows.



Variety of dialog boxes for setting printer-specific options

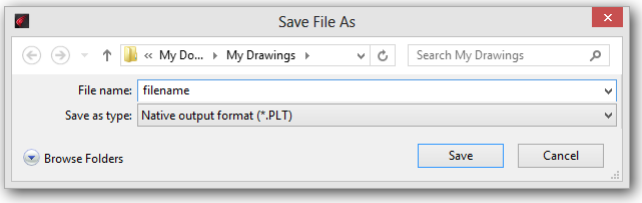
Here is the meaning of the other options in the program's Printer/plotter section:



Options in the Printer/plotter section

Print to File. The **Print to File** check box saves the print data to a file. You use this with printer drivers that are suitable for files, such as PDF and JPEG. You could use this option for regular printers, such as an HP LaserJet, but then you have to know what to do with the resulting *.plt* file.

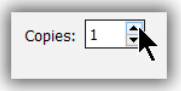
1. Choose the **Print to File** option.
2. Click **OK** to "print" the drawing. Notice that the Save File As dialog box appears.



Saving the plot data to a file

3. Click **Save**. The program saves the *.plt* file to the *c:\users\<login>\documents\my drawings* folder.

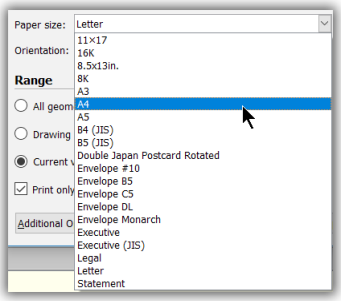
Copies. The Copies option lets you get more than one copy of the drawing printed at a time. You could have the program print like a 100 copies at a time.



Specifying the number of copies to print

Paper Size Option. The second most-important setting in this dialog box is choosing the size of the paper on which the drawing is printed. The paper is sometimes called “media” (plular) or “medium” (singular), because it might not be paper; the medium could also be made from vellum, parchment, transparent plastic, and so on.

The **Paper Size** droplist lets you choose the size of paper on which to print the drawing. The sizes listed are those supported by the printer. For instance, the largest size my HP LaserJet supports is 11x17.



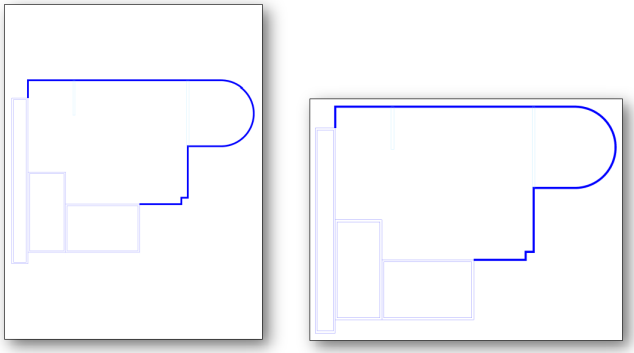
Choosing the paper size from among the options available

Some of the paper sizes in the list adhere to international standards:

Standard	Units	Used In
ANSI	Imperial	North America, mechanical designers
ARCH	Imperial	North America, architects
ISO	Metric	Europe and some parts of North America
JIS	Metric	Japan

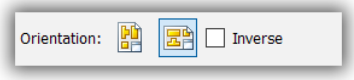
When you choose a different printer, then Paper Size lists other sizes. For example, choose JPG format:, and the list changes to commonly-used raster sizes, such as 1600x1200 pixels.

Orientation and Inverse. Traditionally, small A-size (or A4) drawings were drawn in portrait orientation, while larger drawings were in landscape orientation.



***Left**: Geometry on paper in portrait mode; **right**: Geometry on paper in landscape mode*

But with CAD, orientation doesn't matter. You can have the drawing printed one way or another. The program gives you the choice with the **Orientation** option:



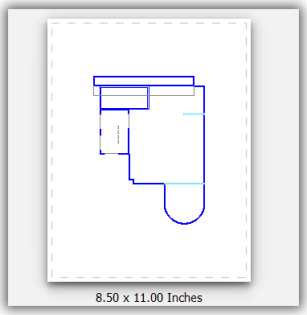
Changing the orientation of the geometry on the paper

- **Portrait** orients the paper uprightly; see figure below
- **Landscape** rotates the paper sideways by 90 degrees

The **Print Inverse** checkbox mirrors the drawing, which is useful for printing on the "underside" of mylar sheets.

Real-time Preview

When you make changes to how the drawing will be printed, it's a good idea to see what the changes look like before committing to print. This saves paper and time. Viewing the output ahead of time is called "print preview." The program offers a small thumbnail preview in the upper right corner of the dialog box.

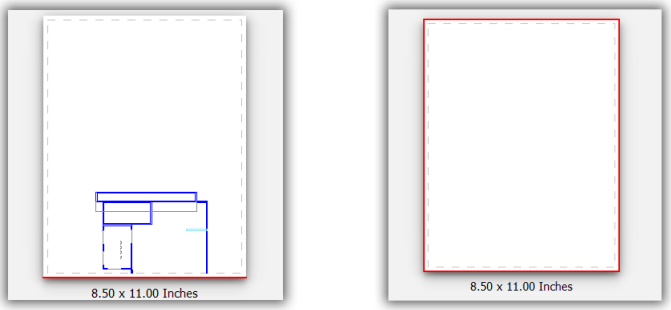


Real-time preview of the plot

As you change options in the dialog box, the changes are reflected in the preview in just a second or two:

- Position of drawing on the paper
- Landscape and portrait orientation
- Range of drawing, such as printing a named view or a windowed area
- Margins shown by dashed lines
- Size of the paper

The preview warns you with a red border when part (or all) of the drawing is not on the paper.

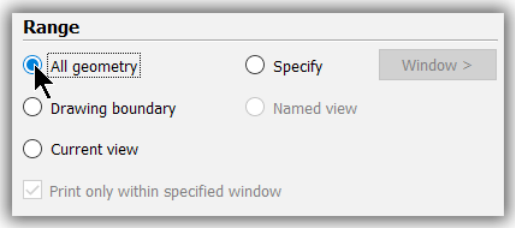


***Left:** Part of the drawing won't be printed; **right:** none of the drawing will be printed*

The red border lines do not prevent you from printing; they are a warning that something might be amiss.

Range Options

You can print drawings the way they look on the screen at the time you invoke the Print command, or else you can print only a portion. The options in the **Range** section look like this.



Choosing how much of the drawing to print

All Geometry prints all visible entities, no matter the zoom level. Technically, the program creates an invisible bounding rectangle that encompasses all non-frozen entities.

TIP Entities are not printed when they are on the following layers:

- Layers that are frozen
- Layers that are set to no-print
- Layer "Defpoints," which is always set to no-print automatically

---

Drawing Boundary prints the area of the drawing defined by the DrawingBounds command. When you select this option, the **Print Only Area Within Specified Window** option becomes available.

Named View prints the area of the drawing specified by a named view; this option is available only when the drawing contains views created by the Views command.

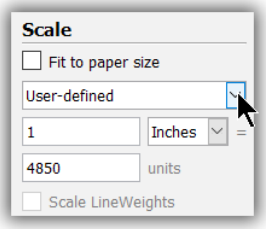
Specify prints the area of the drawing defined by a rectangular window. This option is good for printing details or portions of drawings.

Current View prints the area of the drawing currently visible in the drawing area.

Print Only Area Within Specified Window cuts off entities located outside of the area defined by the Drawing Boundary, Named View, Specify window, and Current View options.

Scale Options

Drawings can be printed to fit the paper, or they can be printed to scale. The program gives you the choice in the **Scale** section.



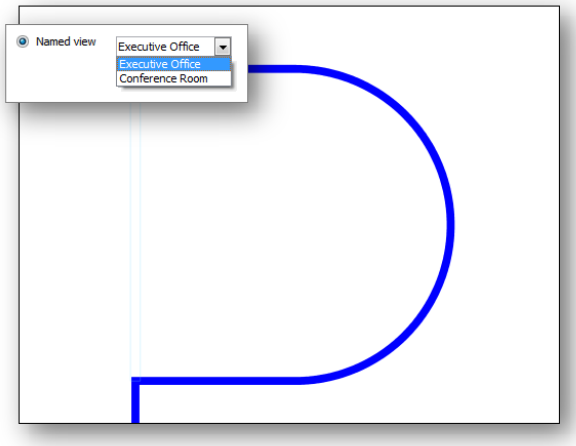
Specifying the scale at which the drawing will be printed

This is what the options mean:

Fit to Paper Size ensures the drawing fits the paper; the drawing area is determined by the Print Range options described below. Turn off this option to specify the scale factor at which the drawing is printed.

PRINTING NAMED VIEWS

1. Before starting the Print command, create one or more named views with the **Views** command.
2. Enter **Print**, and then choose the **Named View** option. Notice the droplist that appears.

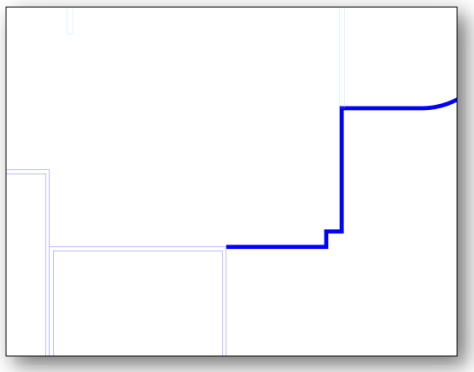


3. Select a named view from the droplist.
4. CHeck the **Preview** window to ensure the area to be plotted matches that of the named view.

PRINTING A WINDOWED SELECTION

1. Choose the **Specify** radio button.
2. Click the **Window**. Notice that the dialog box disappears so that you can pick two points in the drawing.
3. Define the rectangular area you want to print by picking two points, as the program prompts you:

```
Specify first corner: (Pick a point, or enter x,y coordinates)
Specify second corner: (Pick another point, or enter x,y coordinates)
```

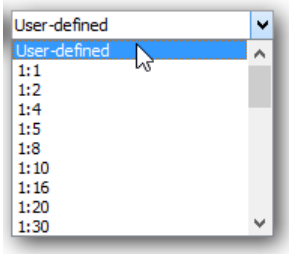


4. After you pick the second point, the dialog box returns and you can set other options.

TIP The program does not actually fit the drawing to the edges of the paper, but rather to the *margins*. Recall that the margin is area along the edges of the paper on which the printer cannot print; the printer uses the four edges for handling the paper. Some printers, however, can print "borderless" right up to the edge of paper, which means there are no margins.

Scale specifies the printed drawing's scale factor. The **Fit to Paper Size** option must be turned off.

Scale droplist contains common scale factors in both Imperial and metric units.



Choosing the print scale from a predefined list

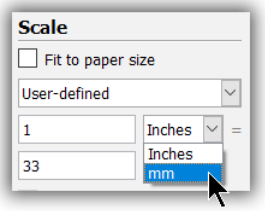
Example Scale	Units	Meaning
1:2	Metric	One mm (or cm or m) on the paper equals 2 in the drawing (the drawing is plotted half-size)
1/2" = 1'	Imperial	A half-inch on the paper equals a foot in the drawing (the drawing is plotted 24 times smaller)

Scale LineWeight to Print Scale option scales the width of lineweights to match the print scale. This ensures that the weights do not overwhelm drawings that are printed at small scale factors.

6. Choose a predefined scale factor from the droplist, or else choose User-defined to specify your own custom scale factor.

SPECIFYING A CUSTOM PRINT SCALE

To specify a custom scale factor, such as 1:33mm, follow these steps:

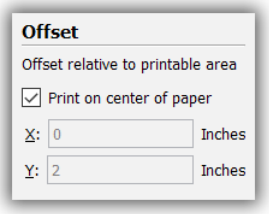


1. Select "User-defined" from the **Scale** droplist.
2. Enter **1** as the number of millimeters, the distance measured on the paper.
3. Select **mm** from the droplist of units.
4. Enter **33** as the number of units in the drawing.

TIP Remember to click **Print Preview** to see that the drawing properly fits the paper.

Offset Options

The **Offset** section determines where the drawing is placed on the paper:

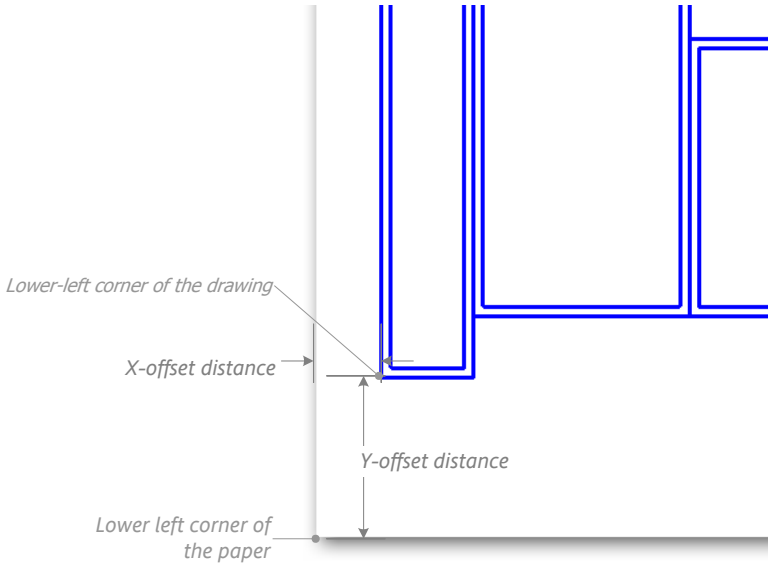


Options for moving the drawing on the paper

Print on Center of Paper centers the drawing on the paper. This is the option you use the most often, unless you have a reason to move the drawing over.

X and **Y** specify the distance from the lower left corner of the paper to the lower left corner of the drawing, as illustrated above. Use these distances to move the drawing out of the way of title blocks and other pre-printed areas of the paper.

Be careful, as it is possible for the drawing to be moved over so far that it ends up not being printed! Use the Preview to make sure the drawing is positioned properly.

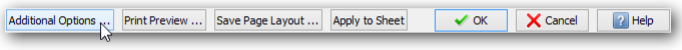


Specifying the offset distance between the geometry's and paper's corners

TIP You can use layout mode to position the drawing visually relative to the paper. See chapter 18.

ADDITIONAL BUTTONS

Along the bottom of the dialog box is a row of buttons that lead to additional options:



Additional Options displays the Additional Print Options dialog box, which allows you to specify other printing options. See the Additional Options section below.

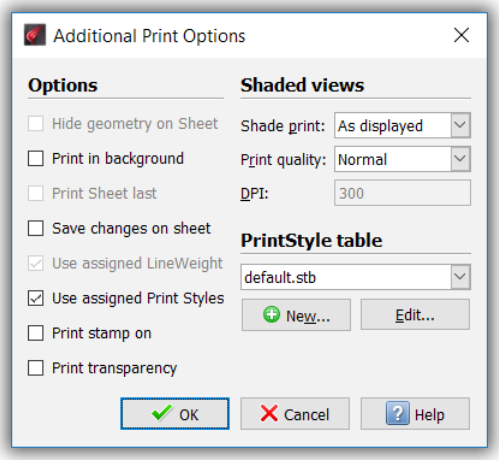
Print Preview displays the Print Preview window, which allows you to check the drawing's position and size before it is printed. See the Preview command later in this chapter.

Save Page Layout displays the Create Page Layout dialog box, which allows you save the print settings for use with the PageLayout command. See the PageLayout command later in this chapter.

Apply to Sheet saves the current settings to the standard page layout of the current sheet.

Additional Options

The program keeps a second set of options "hidden" away in a second dialog box. Click **Additional Options** to see them.



Additional print options

The following options come into effect when they are turned on:

Hide Geometry on Sheet does not print hidden lines of 3D entities shown through viewports in sheets (layouts). This option is unavailable (grayed out) when you plot from the Model tab.

Print in Background prints the drawing in the background, allowing you to return to the drawing editor almost immediately. Only turn off this option when printing is too slow.

Print Sheet Last prints paper space entities before printing model space entities shown in viewports. This option is unavailable (grayed out) when plotting from Model tab.

Save Changes on Sheet saves the Print dialog box's settings with the active sheet. This is handy to make sure that every time you come back to the sheet, the Print dialog box shows the settings you used last time.

Use Assigned LineWeight prints entities with the LineWeights assigned to them in the drawing. This option is unavailable (grayed out) when Use Assigned Print Styles option is turned on, because print styles assign lineweights to the print out independently of the program.

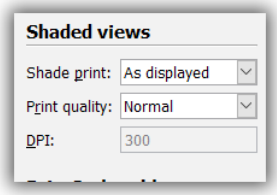
Use Assigned Print Styles prints entities with the PrintStyles assigned to them. This option is unavailable (grayed out) when print styles were not assigned to the drawing.

Plot Stamp On adds a plot stamp to the edge of the paper; see the PrintStamp command later in this chapter.

Plot Transparency prints transparent objects with simulated transparency, where entities underneath show through entities that are on top of them. Normally this setting is turned off, because it can take a long time to generate the transparency look. Some printers are unable to simulated transparency.

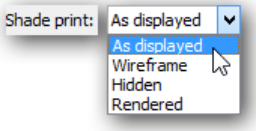
Shaded Views

The options in the Shaded Views section change the way 3D entities are printed:



Shaded views options

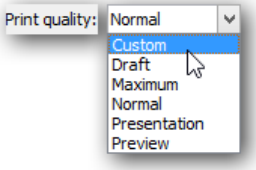
Print Style specifies how 3D entities are printed. Choose a style from the droplist:



Choosing the way to print 3D models

- **As Displayed on Sheet** mode prints the way it looks.
- **Hidden** mode removes hidden lines.
- **Rendered** mode prints the model as rendered.
- **Wireframe** mode prints with no rendering and no hidden lines removed.

Print Quality specifies the quality of printing, according to the capabilities of the printer; choose the quality level from the droplist:



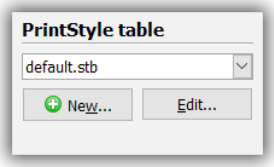
Choosing the print quality

- **Custom** mode specifies the resolution in the DPI text box.
- **Draft** prints drawings at 50dpi; this resolution is crude but fast.
- **Maximum** prints drawings at the printer's maximum resolution (typically 600 or 1200dpi).
- **Normal** prints the drawing at the printer's default resolution (typically 300 or 360dpi).
- **Presentation** ode prints the drawing at the printer's maximum colors and resolution.
- **Preview** prints drawings better than draft quality but worse than normal.

DPI specifies the number of dots per inch at which to print the drawing; this option is available when Print Quality is set to Custom.

PrintStyle Table

These options assign a print style table to the drawing:



Choosing the print style

PrintStyle table selects a print style (*.ctb*) file name from the droplist; CTB is short for "color-based table."

New creates new print styles; see the section on print styles later in this chapter.

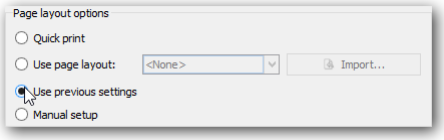
Edit modifies the selected print style.

PRINTING THE OFFICE DRAWING AGAIN (WINDOWS AND LINUX)

When you set up a print job for a drawing with the Manual Setup mode, you can reuse the settings with the **Use Previous Settings** mode. It prints the drawing with the same settings used in the previous print job. This is suited to repeated printouts of the same drawing.

The mode works like this:

1. Start the **Print** command.
2. Choose **Use Previous Settings** in the Print dialog box.



Printing the drawing using the settings from the previous time

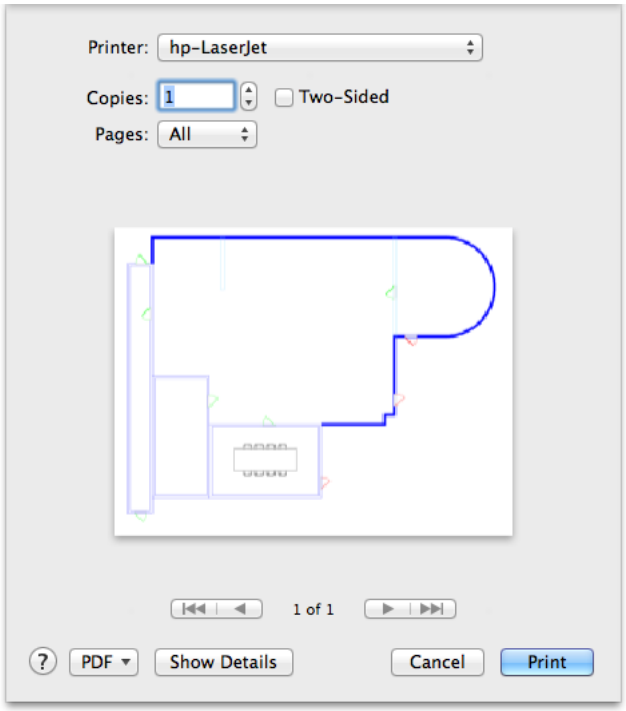
Notice that most options are still available, allowing you to make changes.

3. Click **OK**. The drawing is printed.

TIP Linux and Windows users can now jump past the next several pages of gray text, which describe how to print with the MacOS version.

DRAFT PRINTS (MACOS)

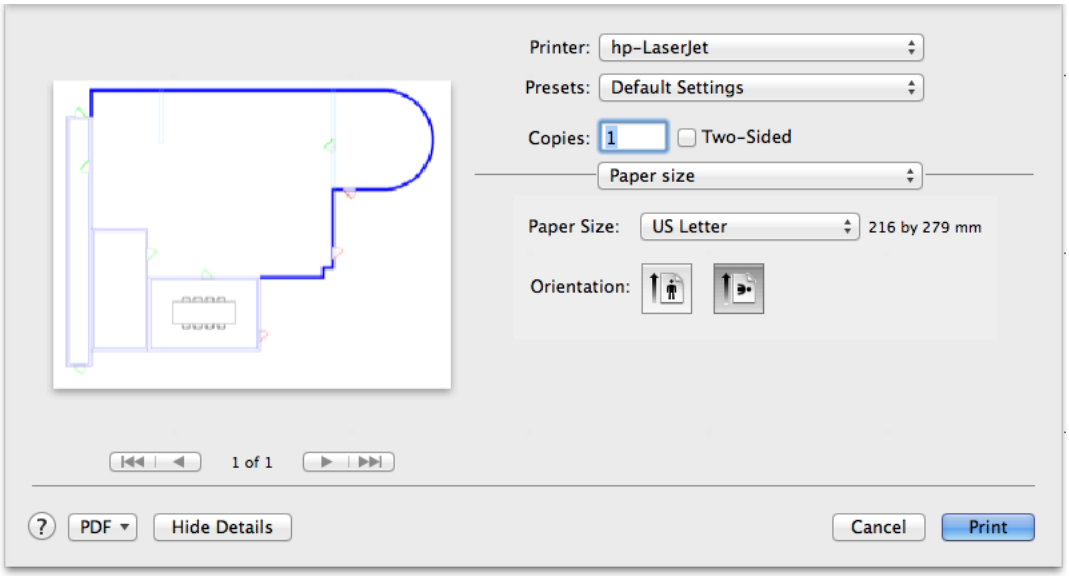
The MacOS version uses Apple's interface guidelines for the Print dialog box, and so it looks very different from the one used by the Linux and Windows versions:



Print window in Mac showing preview

The primary difference is that the preview window always shows the true drawing, not the rectangular approximation found in the Linux and Windows version. (The Mac version also has the Preview command and its full-screen preview.)

If you so not see the full dialog box, click the **Show Details** button.



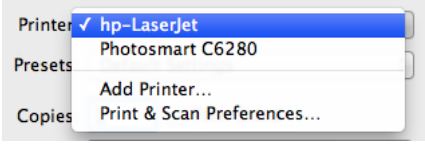
Expanding the dialog box to see all options

The second big difference is that all of the program's print options are accessed through this droplist:

Droplist of options, beginning with Paper Size

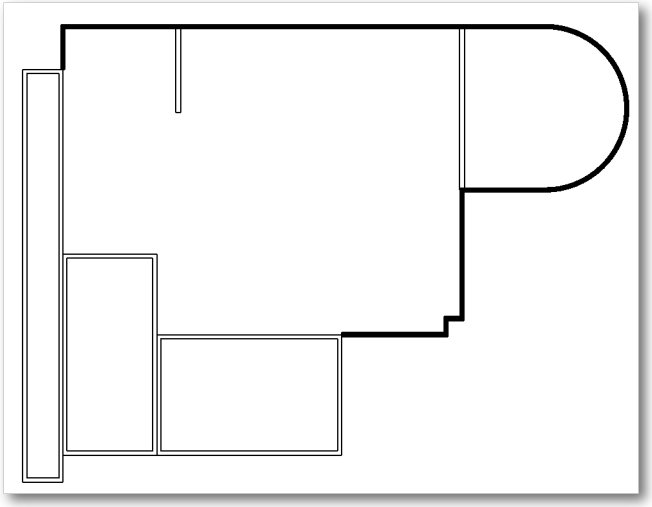
In this first printing tutorial for MacOS, you produce a draft (or quick) printout of the office layout drawing.

1. Start the **Print** command using one of the methods listed above. Notice the dialog box.
2. Choose the printer from the **Printer** droplist; you can select only from those that are set up on your Mac.



Choosing the printer

3. Click **Print**. Within a few moments, the drawing should emerge from the printer.



Result of the quick print setting

If the printout does not appear as you expect, then you can adjust printing options manually, as described next. Once you establish the settings to your satisfaction, you can save them, and then recall them for future printing jobs through the Presets droplist.

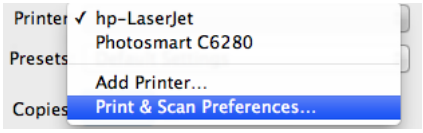
PRINTING THE OFFICE DRAWING MANUALLY (MACOS)

In this tutorial, let's go through the options presented by the Print dialog box, from top to bottom. I leave detailed configuration for later in this chapter.

Printer Options

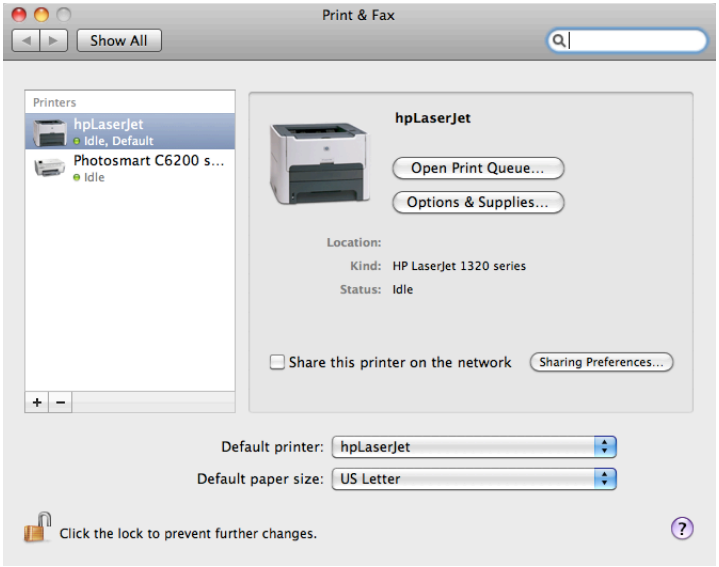
As mentioned earlier, to change the printer or its settings, choose its name from the Printer droplist. To change the MacOS settings of the printer, follow these steps:

1. To change the printer's settings, choose **Print & Fax Properties** from the Printer droplist.



Changing the printer's settings

2. In the Print & Fax dialog box, choose a printer from the list.



Dialog box for printers

3. Click the **Options and Supplies** button. The properties available will depend on the printer model.
4. Make changes as necessary, and then click the close button to close the Print & Scan dialog box. You are returned to the Print dialog box.
5. **Copies** specifies the number of copies to print; maximum is 9,999 copies.

Setting the number of copies to print

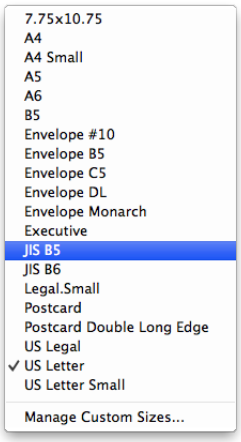
Collated option applies only when printing sets of drawings. Instead of printing 10 copies of page 1 followed by 10 of page 2, the Collated option prints pages 1 and 2, followed by 1 and 2 again.

Two-Sided option applies only to duplex printers that can print on both sides of the paper.

6. The **Paper Size** droplist lets you choose different sizes of paper on which to print the drawing.

Selecting the size of paper

Shown below are the standard sizes supported by HP's ubiquitous LaserJet laser printer.



Paper sizes supported by the selected printer

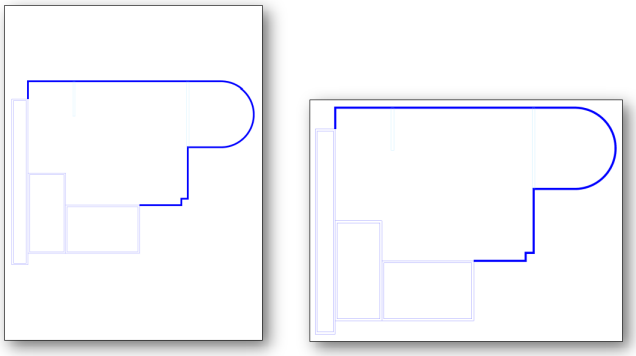
7. The **Orientation** section offers these options:

Choosing the page orientation

From left to right, these are:

Portrait orients the paper vertically; commonly used with A and A4 size paper. See figure below.

Landscape rotates the paper horizontally by 90 degrees; commonly used with all other sizes.

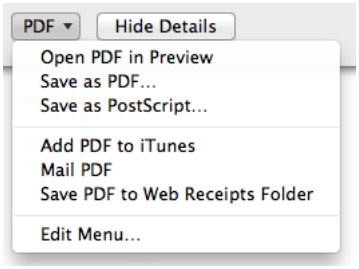


Left: *Paper in portrait mode*
Right: *Paper in landscape orientation*

8. The pages buttons lets you view multi-page documents before they are printed. The program at this point has the ability to only print one drawing at a time, and so these buttons are inactive.

Viewing additional pages in multi-page documents

9. The **PDF** droplist lets you output the drawing as a PDF file — or to iTunes (really?). The difference between PDF and PostScript is that PDF is more like a graphical file, like JPEG, while PostScript is more like a programming file. In most cases, you would choose PDF, unless you have a specific need for PostScript.



Determining the destination of the PDF file

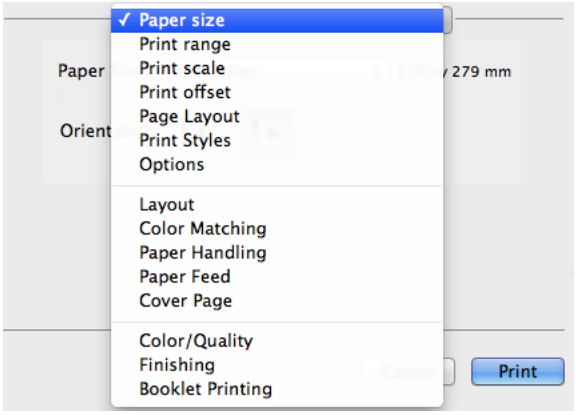
10. Click **Print** to print the drawing.

DETAILED PRINT CONFIGURATION (MACOS)

Let's now go through all the remaining options available for printing drawings.

By clicking the 

droplist, you can set the printout to meet your needs. For instance, you can change the printer's settings, specify a scale factor, and select a drawing view to print.



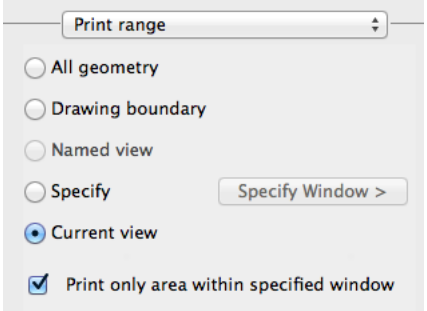
Accessing all of the print options

The **Paper Size** droplist was described earlier. you choose different sizes of paper on which to print the drawing.

Print Range

You can print drawings the way they look on the screen at the time you invoke the Print command, or you can print only a portion. These options affect the size of the printed drawing, when scale is set to Fit.

Choose the **Print Range** option from the droplist to see the following items:



Specifying the area of the drawing to print

All Geometry prints all visible entities, no matter the zoom level; technically, the program creates an invisible bounding rectangle that encompasses all non-frozen entities.

Drawing Boundary prints the area of the drawing defined by the DrawingBounds command. (When you select this option, the Print Only Area Within Specified Window option becomes available.)

Named View prints the area of the drawing specified by a named view; this option is available only when the drawing contains views named by the Views command. See tutorial later.

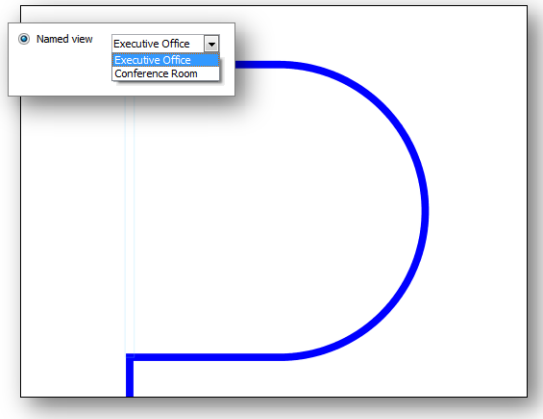
Specify prints the area of the drawing defined by a rectangular window. This option is good for printing details or portions of drawings. See tutorial later.

Current View prints the area of the drawing currently visible; this option is affected by the current zoom level and pan position.

Print Only Area Within Specified Window cuts off entities located outside of the area defined by the Drawing Boundary, Named View, Specify window, and Current View options.

How to Create and Print a Named View

1. Before starting the Print command, create one or more named views with the **View** command.
2. Enter **Print**, and then choose the **Named View** option. Notice the droplist that appears.
3. Select a named view from the droplist.



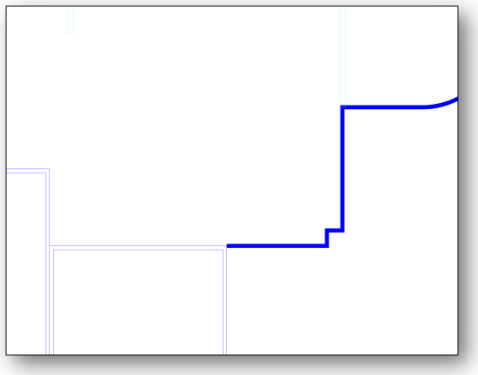
Selecting the named view to print

\4. Click **Print Preview** to see that the area to be plotted matches that of the named view, as illustrated above.

How to Print a Specified Window

1. Click **Specify Window.**
2. Define the printable area by picking two opposite points, as the program prompts you:

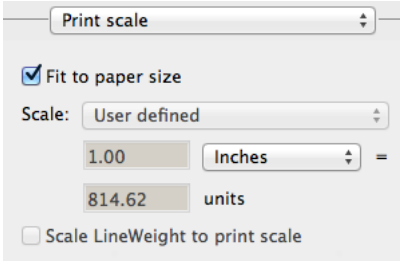
```
Specify first corner: (Pick a point, or enter x,y coordinates.)
Specify second corner: (Pick another point, or enter x,y coordinates.)
```



Picking two points to print a windowed area

Print Scale

Drawings can be printed to fit the paper, or they can be printed to scale. Choose the **Print Scale** option from the droplist to see the following items:

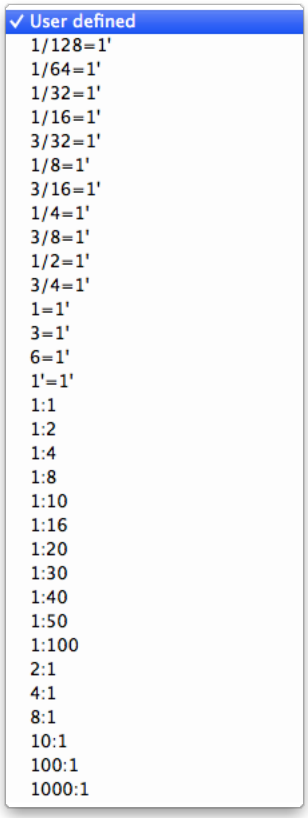


Specifying the scale at which to print the drawing

Fit to Paper Size ensures the drawing fits the paper; the drawing area is determined by the Print Range options described above. Turn off this option to specify the scale factor at which the drawing is printed.

Scale specifies the printed drawing's scale factor. (To access the scale factors, you must turn off the Fit to Paper Size option.) Choose a predefined scale factor from the droplist, or else choose User-defined to specify your own scale factor. See tutorial following.

The droplist contains common scale factors for Imperial units (such as 1/4”=1’) and metric units, such as 1:20.

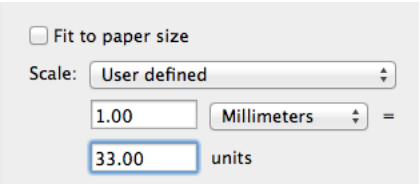


Choosing from the list of predetermined print scales

Scale LineWeight to Print Scale scales lineweights to match the print scale. This ensures that the width of lineweights does not overwhelm drawings printed at small scale factors.

How to Create a Custom Plot Scale

To specify a custom scale factor, such as 1:33mm, follow these steps:

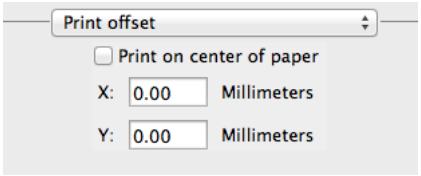


Defining your own print scale

1. Ensure **Fit to Paper Size** is turned off.
2. Select **User-defined** from the Scale droplist.
3. Enter **1** as the number of millimeters. This is the distance measured on the paper.
4. Select **Millimeters** from the droplist.
5. Enter **33** as the number of units in the drawing. Remember to check the preview area to see how the drawing now fits the paper.

Print Offset

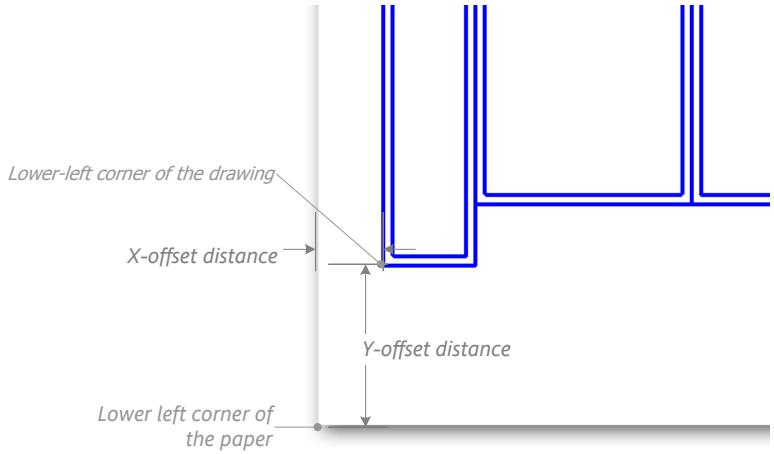
The Print Offset section determines where the drawing is placed on the paper. Choose the **Print Offset** option from the droplist to see the following items:



Parameters to print the drawing offset on the paper

Print on Center of Paper centers the drawing on the paper. This is handy for making sure the drawing is nicely centered.

X and **Y** specify the distance from the lower left corner of the paper to the lower left corner of the drawing, as illustrated below. Use these distances to move the drawing out of the way of title blocks and other pre-printed areas of the paper.

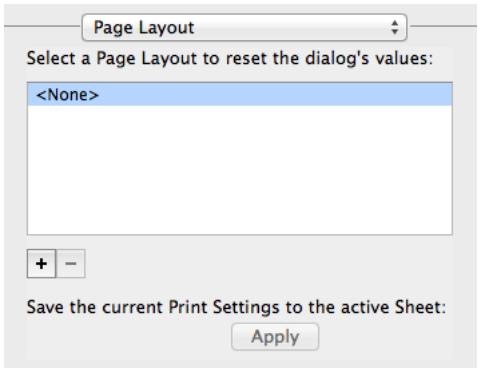


Determining the offset distance

TIP You can use layout mode to position the drawing visually relative to the paper. See chapter 18.

Page Layout

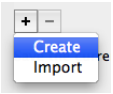
Page Layout applies previously made settings to the entire Print dialog box. Choose the **Page Layout** option from the droplist to see the following items:



Choosing a layout to print

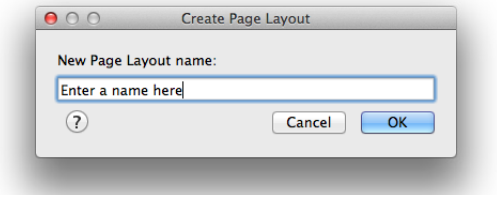
Select a Page Layout lists the names of page layouts loaded into this drawing. See more information about page layouts later in this chapter.

+ creates or imports page layouts.



Clicking the + button to create or import page layouts from other drawings

- **Create** option prompts you to adds enter the name of new page layout. The current settings of the Print dialog box are saved to this name, which you can then later reuse.



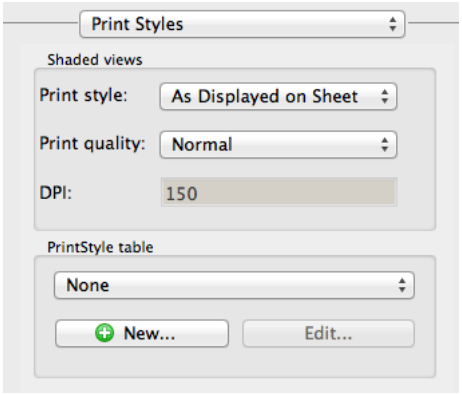
Naming the new page layout

- **Import** option opens the Open dialog box, from which you can select a drawing file that yo know contains print layouts. The files can be DWG drawings, DXF interchange, or DWT template files.

Apply saves the settings to the active sheet, should you be printing a sheet at this time.

Print Styles

Print Styles specifies how 3D entities are printed. Choose the **Print Styles** option from the droplist to see the following items:



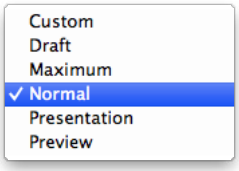
Options for printing 3D models

Shaded Views controls the look of the entire 3D drawing. Choose a style from the droplist:

Determining how 3D models look when printed

- **As Displayed on Sheet** prints the way it looks.
- **Hidden** removes hidden lines.
- **Rendered** prints the model as rendered.
- **Wireframe** prints with no rendering and no hidden lines removed.

Print Quality specifies the quality of printing, according to the capabilities of the printer; choose the quality level from the droplist:

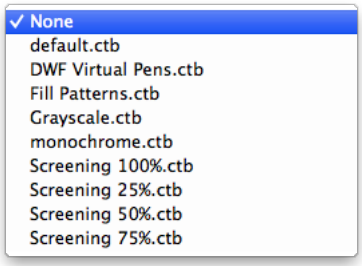


Specifying the quality of printing

- **Custom** mode specifies the resolution in the DPI text box.
- **Draft** prints drawings at 50dpi; this resolution is crude but fast.
- **Maximum** prints drawings at the printer's maximum resolution (typically 600 or 1200dpi).
- **Normal** prints the drawing at the printer's default resolution (typically 300 or 360dpi).
- **Presentation** ode prints the drawing at the printer's maximum colors and resolution.
- **Preview** prints drawings better than draft quality but worse than normal.

DPI specifies the number of dots per inch at which to print the drawing; this option is available when Print Quality is set to Custom.

Print Style Table controls how individual entities and layers are plotted, assuming a plot style table has been attached. Choose a print style (*.ctb*) file name from the droplist. (CTB is short for "color-based table.")



Choosing a print style

New creates new print styles; see the section on print styles later in this chapter.

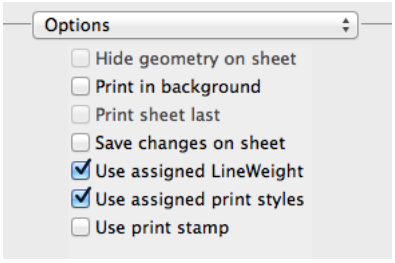
Edit modifies the selected print style.

Options

The following items come into effect when Options is selected from the droplist.

Hide Geometry on Sheet does not print hidden lines of 3D entities shown through viewports in sheets (layouts). This option is unavailable (grayed out) when you plot from the Model tab.

Print in Background prints the drawing in the background, allowing you to return to the drawing editor almost immediately. Only turn off this option when printing occurs too slowly.



Additional printing options

Print Sheet Last prints paper space entities before printing model space entities shown in viewports. This option is unavailable (grayed out) when plotting from Model tab.

Save Changes on Sheet saves the Print dialog box's settings with the active sheet.

Use Assigned LineWeight prints entities with the LineWeights assigned to them in the drawing. This option is unavailable (grayed out) when Use Assigned Print Styles option is turned on, because print styles assign lineweights.

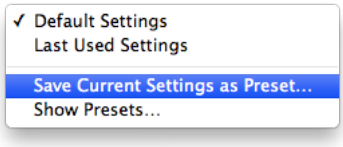
Use Assigned Print Styles prints entities with the print styles assigned to them. This option is unavailable (grayed out) when print styles were not assigned to the drawing.

Use Print Stamp applies the print stamp information defined by the **PrintStamp** command, as described at the end of this chapter.

Saving Settings as Presets

After you change one or more settings, you may want to preserve them for future prints. I tend to print drawings (a) with all geometry, (b) scaled to fit the paper, and (c) centered.

To save customized settings, click the **Standard** drop list. Above the horizontal line are the names of saved settings. Initially, there are two: Standard (the default settings) and Last Used Settings.



Saving the current printing settings for reuse

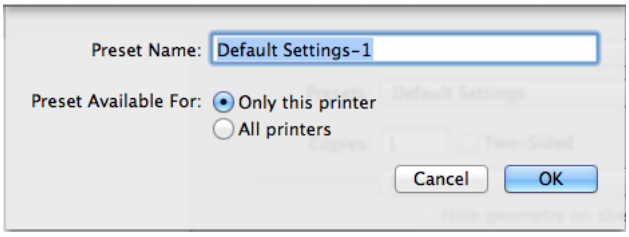
Below the horizontal line are the controls:

Save Current Settings as Preset prompts you to enter a name by which you can recall the preset for future print jobs. You specify whether the preset will apply to the current printer or to all printers. See tutorial following.

Show Presets brings up the Presets dialog box, which is generated by MacOS.

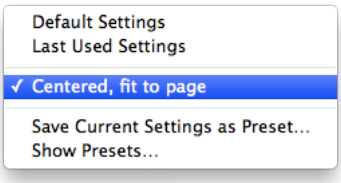
How to Save Print Settings

To save print settings for reuse, following these steps:



Naming the preset

1. From the **Presets** droplist, choose the **Save Current Settings as Preset** item. Notice the dialog box.
2. Enter a name that describes the settings, such as "Centered, fit to page."The program prevents you from using a name that already exists.
3. Decide whether the settings apply to the current printer, or to all. In my case, I always want my prints to be centered and fitted, so I would choose **All Printers**.
4. Click **OK**. Notice that the name is added to the list of preset names.

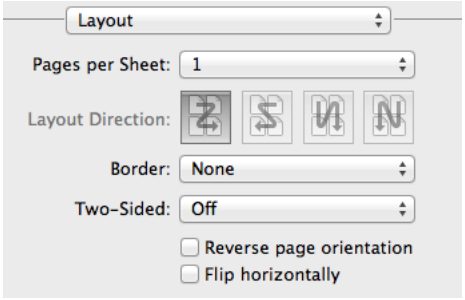


Selecting a preset by name

From now on, you need only select the preset name, instead of going through the five sections.

MacOS Options

A second set of options is common to the Macintosh, not specific to the program. Many of these are not usually of interest to CAD users, but some of them may be useful from time to time. Here is a summary of the options:



Printing options available through MacOS

Layout lets you print the drawing over two or more pages. The Flip Horizontally option mirrors the print, which can be useful when you need to print the drawing on the underside of transparent media.

Color Matching ensures the colors you see on the screen are reproduced by the printer. This is of no consequence to monochrome laser printers, and matters little for color prints of drawings.

Paper Handling prints multi-page output in forward, reverse, odd-only, or even-only order.

Paper Feed specifies which paper tray to use, useful for printers that have two or more trays holding different sizes of paper.

Cover Page prints a cover page, useful for separating multiple print jobs, especially for networked printers serving numerous computers.

Color/Quantity Options vary according to the printer; the settings illustrated above are for the HP LaserJet. A higher resolution produces a higher quality print, but takes longer to output.

Finishing specifies the type of medium, such as paper or transparency.

Booklet Printing is the opposite of Layout; here, multiple drawings are printed on both sides of single sheets of paper.

TIP For the remaining topics in this chapter, the procedures and dialog boxes are pretty much identical for all three of the operating systems that the program runs on — Linux, MacOS, and Windows.

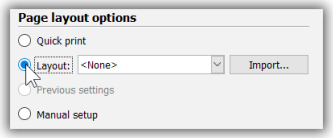
PRINTING THE OFFICE DRAWING WITH PAGE LAYOUTS (LINUX, MacOS, WINDOWS)

For the rest of this chapter, the printing procedures and dialog boxes are identical for all three operating systems — Linux, MacOS, and Windows.

———

Page layouts let you reuse print options that are stored in the drawing. They define everything about how the drawing should be plotted. Think of page layouts as preset plot configurations.

You cannot, of course, use page layouts unless at least one has already been defined. In the figure below, the '<None>' indicates the current drawing contains no page layouts. The adjacent **Import** button lets you import page layouts from other *.dwg* drawing files, or from *.dwt* template files that should contain your office's drafting standards.



Choosing the Page Layout option

Page layouts are usually defined with the **PageLayout** command before you start any print jobs. Page layouts are stored in the drawing file. (They should not be confused with print styles. Page layouts are also known as "page setups" and just plain as "layouts.")

- If the drawing contains a page layout, then you can choose it from the **Layou**t droplist.
- If the drawing contains none, then you have two ways to proceed:

  a. Borrow a page layout from another drawing by clicking the **Import** button; see how to do this with the "Borrowing Page Layouts" topic below

  b. Exit the Print dialog box, and then create a page layout with the **PageLayout** command; see how to do this with the "How to Create a Page Layout" section later in the chapter

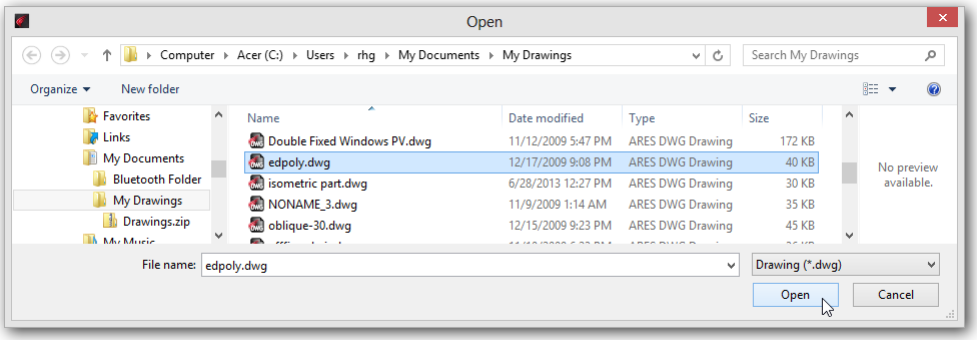
Borrowing Page Layouts

You borrow page layouts from other drawings when the current drawing does not have one, or else when you want to reused page layouts from other drawings. Follow these step to borrow them:

1. In the Print dialog box, click **Import**.

Clicking the Import button

2. Notice the Open dialog box. Go to the folder containing drawing files with page layouts.



Picking a drawing file from which to borrow the page layout

3. If necessary, select a file type. The program imports page layouts from the following types of files, including those produced by other CAD systems:

   DWG drawing files from the program, AutoCAD, and so on

   DXF interchange files from many graphical programs

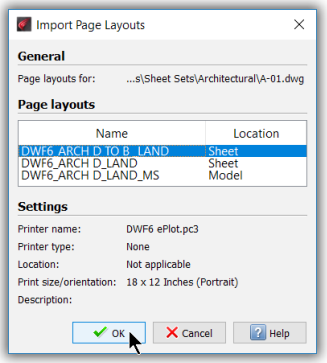
   DWF drawing Web files

4. Select the file.

TIP You may well face this problem: most drawings lack page layouts, but the dialog box does not indicate this. You learn of your success only after clicking Open, and then seeing if the Import Page Setups dialog box is full — or empty.

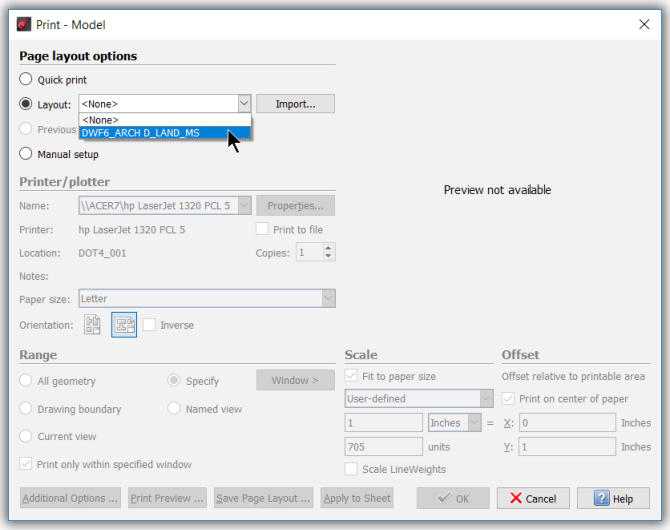
---

5. Click **Open**. Notice the Import Page Layouts dialog box.



Page layouts found in the drawing file

6. Choose one or more — or all — of the page layouts listed in the dialog box.
7. Click **OK**. Notice that the Print dialog box reappears, and that the page layouts are added to the droplist in the Print dialog box.



Print dialog box's options populated by the borrowed page layout

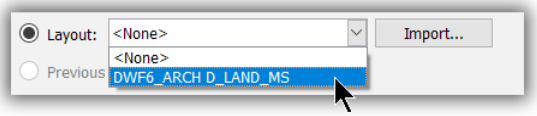
8. Choose one from the droplist. Notice that the dialog box's options become available so that you can override the page layout settings. If you wish to save the changed page layout settings, then click the **Save Page Layout** button.

9. Check **Preview** to confirm the result, and then click **Print** to print the drawing using the imported page layout.

Using Page Layouts

To apply a page layout to the current print job, follow these steps:

1. Start the **Print** command, and then choose **Layout**. Notice that most options of the Print dialog box are grayed out; they are unavailable until you choose a named page layout.
2. From the Layout droplist, select the name of a layout.



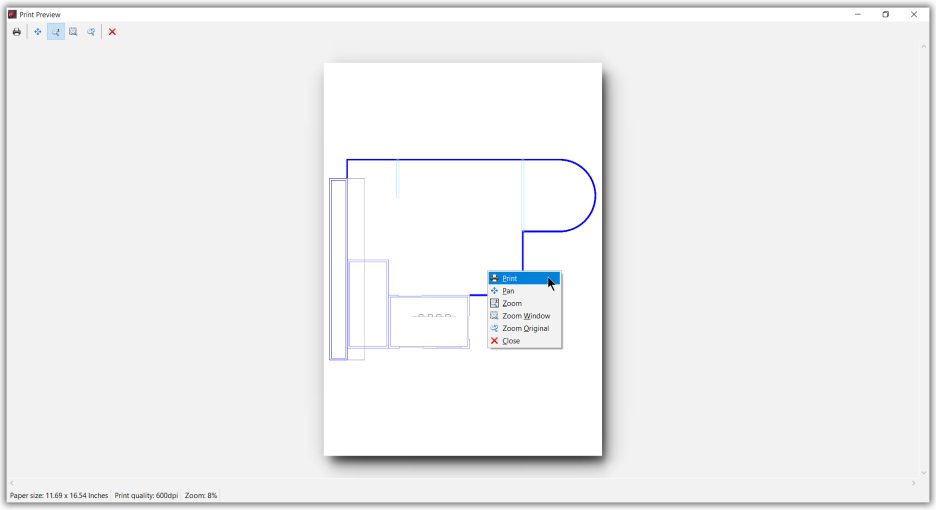
Choosing the page layout name

3. Click **Page Preview** to check the look of the drawing before it is printed.
4. Click **Print**.

Previewing the Print

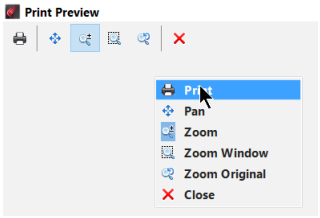
Command: Preview
Alias: pre
Menu bar: File | Print Preview
Ribbon: Print | Print Preview (Application button)
Toolbar:

You can use the real-time preview window, but it's pretty small. So the alternative is to click the dialog box's **Print Preview** button to see a preview of the print full-screen, or else enter the **Preview** command from outside of the dialog box.



Window displaying a preview of the plot

The preview allows you to correct errors before you waste a sheet of paper and a bunch of ink, errors such as incorrect size or offset. You control the preview image through the toolbar or the right-click menu.



Controls for the print preview window

The Pan and Zoom options do not affect the printed output; that is, they do not move, offset, enlarge, reduce, or scale the drawing printed on the paper.

Print closes the preview window and then prints the drawing.

Pan moves the preview image; this works only after you zoom into the preview image. During panning, the hand cursor appears (shown at left).

Zoom enlarges and reduces the preview image as you roll the mouse's scroll wheel. Roll forward to zoom in, roll back to zoom out.

Zoom Window enlarges the preview image defined by a rectangle. Pick two points to define the rectangle.

Zoom Original returns the preview image to its original size and position, following zoom and pan operations.

Cancel closes the preview window.

TIP Upon entering the Preview command, you might receive the following error message:

```
Unable to preview print because no printer has been assigned to
the Page Layout.
```

To get around this problem, enter the **Print** command, and then click the **Print Preview** button.

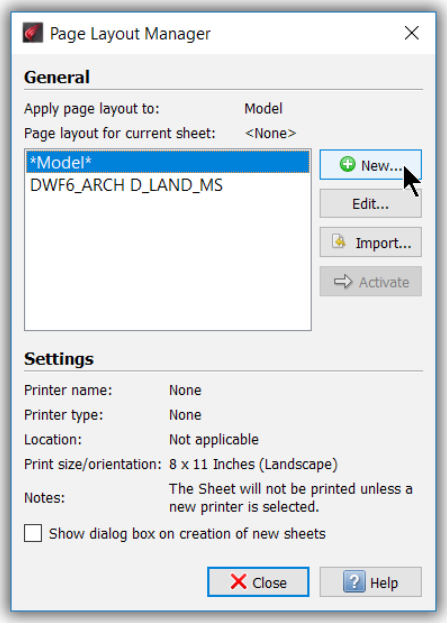
Creating Page Layouts

Command: PageLayout
Alias: pagesetup
Menu bar: File | Page Layout Managers
Ribbon: Layouts | Page Layout Manager (Application button)
Toolbar:

The **PageLayout** command creates and edits page layouts. Page layouts are stored in the drawing file, and should not be confused with print styles. Page layouts are also known as "page setups."

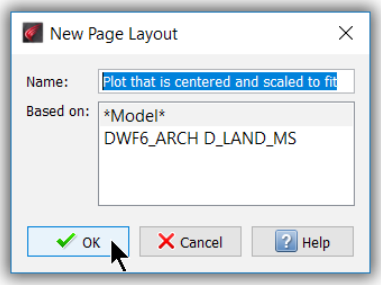
To create a page layout, follow these steps.

1. Start the **PageLayout** command using one of the methods listed above. Notice the Page Layout Manager dialog box. (It looks identical to the Import Page Layout dialog box you saw earlier in the "Borrowing Page Layouts" section of this chapter.)



Page Layout Manager dialog box

2. To create a new page layout, click **New**. Notice the New Page Layout dialog box.

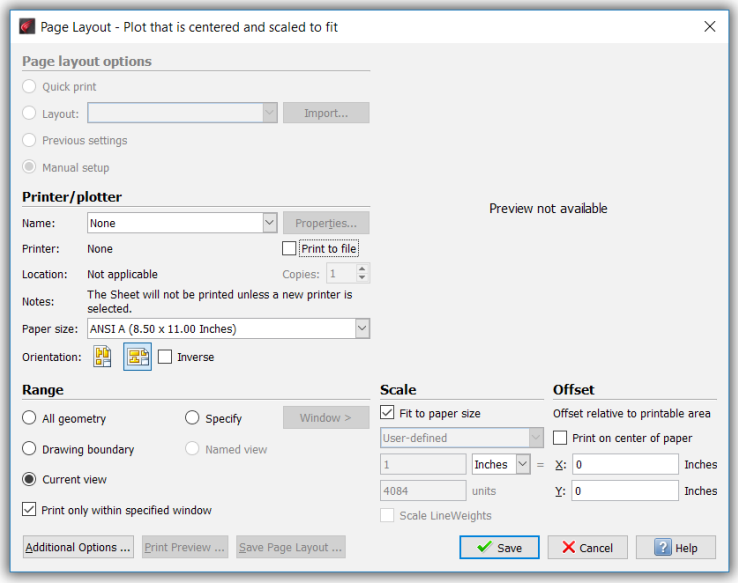


Naming a new page layout

3. In the **Name** field, replace the generic "Setup1" name with a descriptive name, such as "Plot that is centered and scaled to fit."

   If the drawing already contains page layouts, you can choose its name from the **Based On** list. Otherwise, the new one uses the default settings indicated by "*Model*."

4. Click **OK**. Notice that the Page Layout dialog box appears and that it looks almost identical to the Print dialog box.



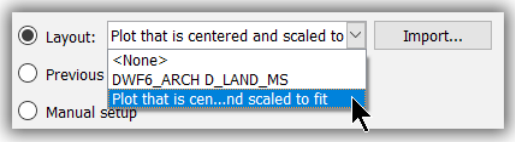
Options for specifying page layouts

They are, in fact, almost identical except that you only get to choose options; you do not print from this dialog box.

5. Change the settings as needed. For this tutorial, ensure the following:

Page Setup Option	Setting
Fit to Paper Size	**On** (found in Print Scale section)
Print On Center of Paper	**On** (found in Plot Offset section)

6. Click **Save** to close the dialog box. The dialog box closes, saving the page layout settings; it does not print the drawing.
7. Start the **Print** command, and then select the **Layout** option.



Choosing the page layout

8. From the droplist, choose **Plot that is centered and scaled to fit**. Notice that the settings in the dialog box change to match those you chose earlier while creating the page layout.
9. Click **OK** to print the drawing.

TIP You can make changes to the settings in the Print dialog box. If you wish to save the changes to the named page layout that is current, then click **Save Page Layout**.

Assigning Print Styles

Command: PrintStyle
Aliases: plotstyle, pstyle
Menu bar: Format | Print Style
Ribbon: --
Toolbar:

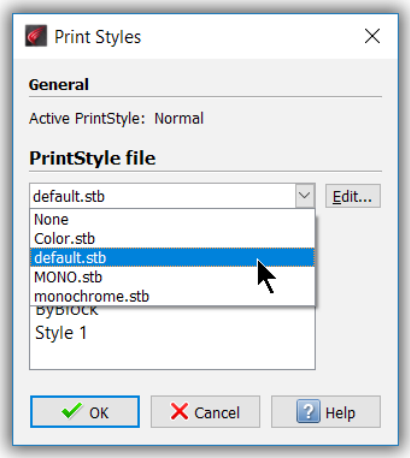
The **PrintStyles** command defines how entities are printed. In general, you can ignore print styles, unless you require extreme control over the printing process.

With the many print style options, you can specify the manner in which every layer and/or entity be printed. Later in this chapter, I describe how every one of the options works. First, here is a tutorial on using them.

CREATING PRINT STYLES

If you want fine control over the look of printed drawings, then this is how you create print styles:

1. Start the **PrintStyle** command using one of the methods listed above. Notice the Print Style dialog box.

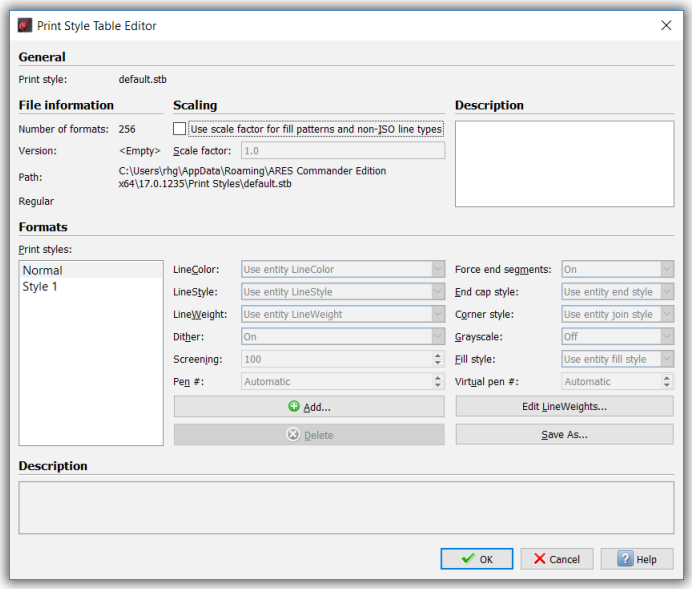


The Print Style dialog box

2. From the **PrintStyle File** droplist, choose a print style name. For this tutorial, choose *default.stb.*

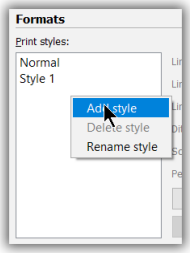
   (Whereas page layouts are stored in drawings, print styles are stored in external *.stb* files. That's because page layouts are specific to drawings, while print styles can be used with any drawing.)

3. Click **Edit**. Notice the Print Style Table Editor dialog box.

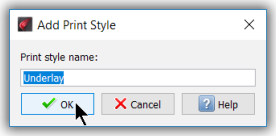


The Print Style Table Editor dialog box

4. In the **Formats** area, right-click, and then choose **Add Style** from the shortcut menu.

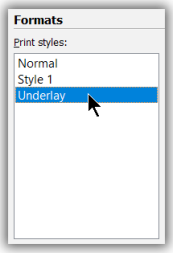


5. In the Add Print Style dialog box, enter a name for the new print style. For this tutorial, enter "Underlay," and then click **OK**.



Naming a new print style

6. Notice that the new "Underlay" print style is added to the list of **Print Styles**.



New print style added to the program

You can now change the settings. For this tutorial, change screening to 30%.

Changing the screening setting

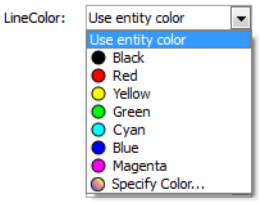
7. Click **OK**.

You have created the new print style named Underlay. In the next tutorial you apply the print style to this book's tutorial drawing.

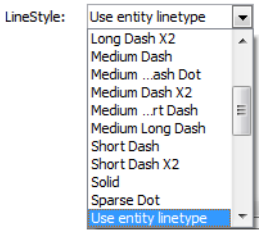
PRINT STYLE OPTIONS

Here is a catalog of the many options for specifying print styles.

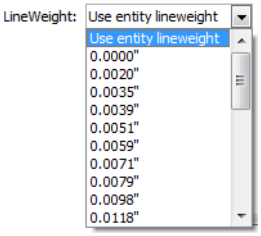
Line Color prints the entity with the color assigned in the drawing (with the Color or Layer commands), or overrules the assigned color with another one. Usually, this option should be left at "Use Entity Color," unless you have a reason to overrule the drawing color.



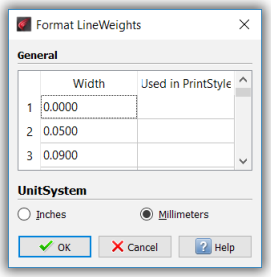
Line Style prints the entity with the linestyle (linetype) assigned in the drawing (with the LineStyle or Layer commands), or overrules the assigned style with another one. Usually, this option should be left at "Use Entity Linetype" unless you have a reason to overrule the drawing linestyle.



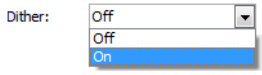
Line Weight prints the entity with the weight assigned in the drawing (with the LineWeight or Layer commands), or overrules the assigned lineweight with another one. Usually, this option should be left at "Use Entity Lineweight," unless you have a reason to overrule the drawing lineweight.



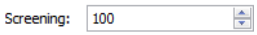
TIP Whereas the widths of lineweights are fixed in the drawing editor, you can edit their values in the Print Style Table Editor dialog box. Click **Edit Lineweights,** and then enter new values.



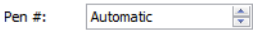
Dither toggles the use of dithering, which is useful when the printer outputs only a few colors. Turning on dithering helps the printer simulate more colors and shades of gray. Usually, this option should be left **off**, unless you know the printer needs it.



Screening adjusts the amount of color. For example, 50% screening prints half white and half color. This is useful for making colors less intense. Usually, this option should be left at **100%**.

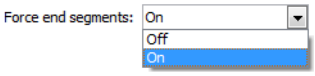


Pen # allows you to specify the pen number the printer should use. This is a holdover from the 1980s, when pen plotters were popular. These plotters had between 1 and 8 pens, which could have different colors or different thicknesses. You can specify that certain layers or entities use a specific pen in the plotter.

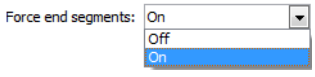


The range is 1 to 32. Usually, however, you keep this option set to "Automatic," which means the plotter uses the pen number that corresponds to the color number. For instance, red is color #1 and so the plotter uses pen #1 — even if it is not red.

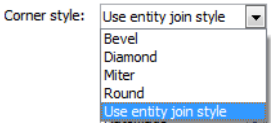
Force End Segments determines if the linestyle scale is adjusted to complete the linestyle pattern along the length of the entity. Usually, this option should be turned **on**.



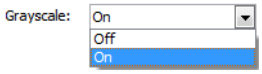
End Cap Style determines how the ends of lines (and other open entities) should be printed. You can choose from Butt, Diamond, Round, Square, or just leave it alone with "Use Entity End Style," which is typically square.



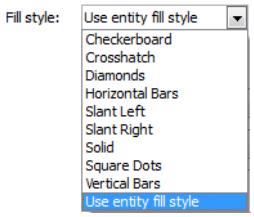
Corner Style determines how the intersections of lines (and other entities) should be printed — and not just corners. You can choose from Bevel, Diamond, Miter, Round, or just leave it alone with "Use Entity End Style," which is typically overlapping.



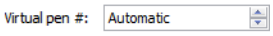
Grayscale toggles the use of shades of gray, instead of color. Use this option if your monochrome printer prints all colors as black, or if you want a color printer to print the drawing colors in gray. Usually, this option should be left **off**.



Fill Style specifies the pattern with which filled areas should be printed. Usually, this option should be left as "Use Entity Fill Style," which is a solid fill. However, you do have a choice of patterns, which is required by some disciplines.



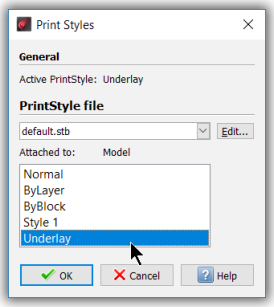
Virtual Pen # is just like the Pen Number option, but is meant for printers that support the concept of pen numbers. For a laser printer, for example, virtual pen #1 might correspond to printing solid lines, while pen #2 might cause it to print crosshatched lines. Usually, you keep this option set to "Automatic," unless you know that the printer supports virtual pens.



APPLYING PRINT STYLES

You've created a print style and now in this tutorial, you learn how to apply print styles to drawings — well, not drawings. Print styles are applied to layers and entities, not entire drawings. To do so, follow these steps:

1. Open the *office plan.dwg* file in the program.
2. Enter the **PrintStyle** command.

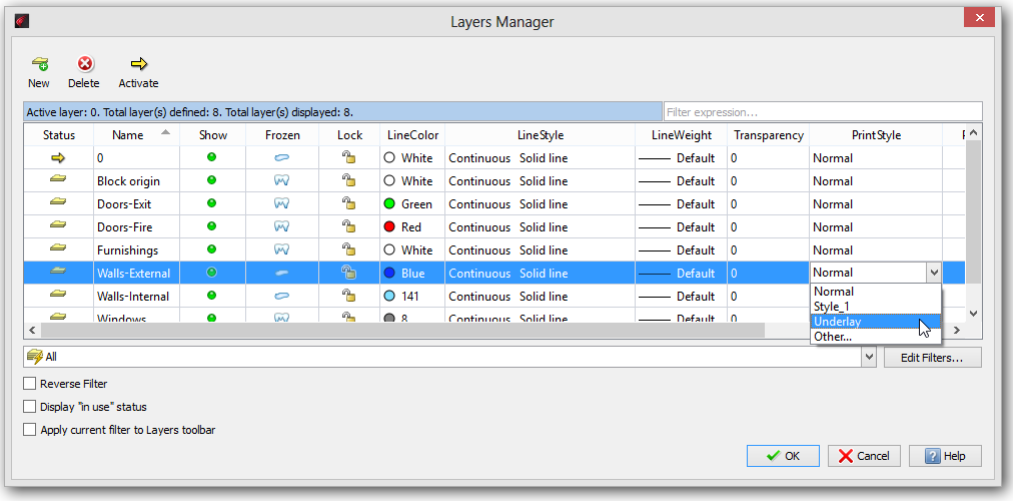


Selecting the print style to apply to the plot

3. In the PrintStyle dialog box, attach the "Underlay" print style like this:
   a. From the PrintStyle File droplist, choose **Default.stb**.
   b. In the list of print style names, choose **Underlay**.
   c. Click **OK**.

   This action attaches the *Underlay* print style to the *office plan.dwg*. The drawing does not look any different. That's because you now apply the print style to *entities* through the Layer command.

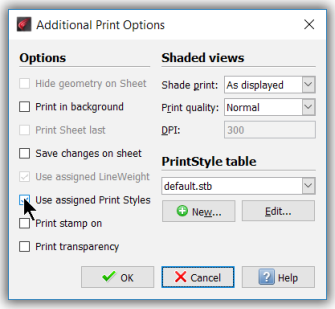
4. Start the **Layer** command.



Assigning print styles to layers

5. In the Layers Manager dialog box, choose the **Walls-External** layer.
6. Notice the PrintStyle column. Click "Normal," and notice that it turns into a droplist.
7. From the PrintStyle droplist, select **Underlay**. (The **Other** item opens the Print Style dialog box.)
8. Click **OK**. Again, the drawing looks no different. That's because print styles apply to prints, not drawings.
9. Start the **Print** command.
10. Click **Additional Options**.
11. In the Additional Print Options dialog box, turn on the **Use Assigned Print Styles** option. (Notice that the Use Assigned LineWeight option is turned off automatically; that's because print styles override lineweights.)

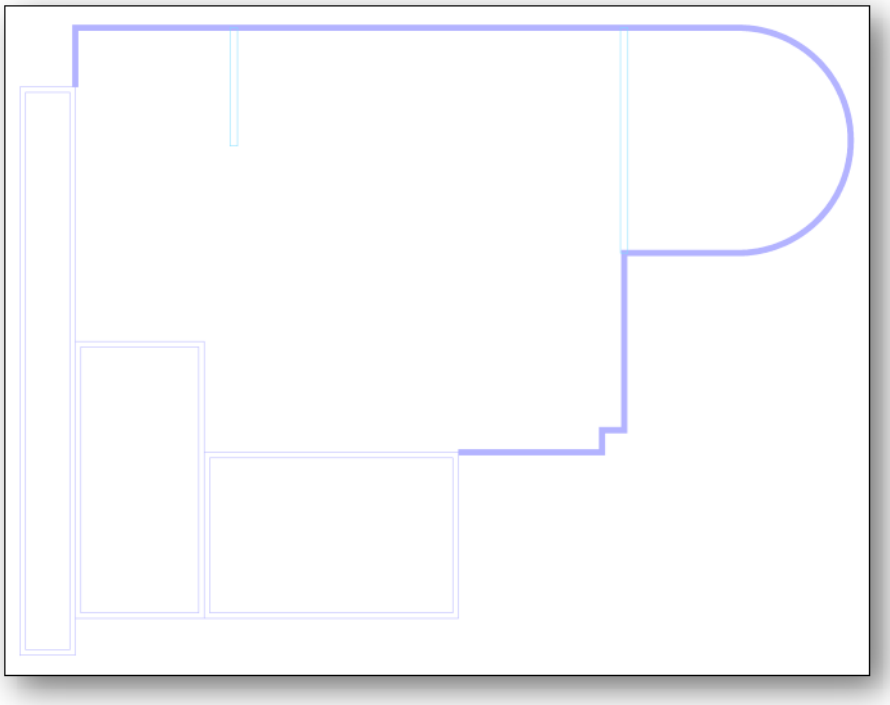
    If necessary, choose "default.stb" from the **Print Style Table** droplist.



Turning on the print style options for the next plot

12. Click **OK** to close the dialog box for additional options.

13. Click **Print Preview**. Notice that the external walls look fainter.

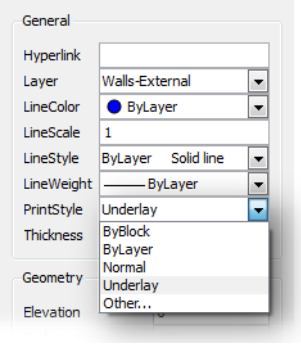


Results of applying the print style to the plot

14. Click **Print**, and the resulting printout should also have fainter wall lines.

TIP To apply print styles to individual entities, use the Properties palette, as follows

1. Press **Ctrl+1** (Cmd+1 on Macs) to open the Properties palette.
2. Select the entities to which you wish to apply a print style.
3. In the Properties palette, choose a print style from the General section.



Use Print Preview to ensure the style has been applied.

Print Stamps

Command: PrintStamp
Aliases: --
Menu bar: --
Ribbon: --
Toolbar: --

While drawings have data, time, and other data automatically applied by the operating system, the same is not so for printed drawings. It can be useful to know when a drawing was printed and from which file. This information can be added through the **PrintStamp** command.

Applying a print stamp takes two steps:

i. Define the print stamp with the PrintStamp command.

ii. Turn on the print stamp option in the Print dialog box.

DEFINING PRINT STAMPS

To define the content of the print stamp, start the **PrintStamp** command. The command works at the command-line, which is less convenient than a dialog box.

1. Enter the PrintStamp command.

```
: printstamp
```

2. Notice that it first lists all of the current settings. To make it easier to read, I added tabs:

```
Active print stamp settings:
    Print stamp:                Off
Fields:
    Stamp drawing name:         No
    Stamp sheet name:           No
    Stamp date and time:        No
    Stamp login name:           No
    Stamp print device name:    No
    Stamp paper size:           No
    Stamp print scale:          No
Comment to stamp
Log file:
    Write to log file:          No
    Log file path:              plot.log
Location:
    Location:                   Bottom left
    Orientation:                Horizontal
    Offset:                     0.10,0.10
    Offset relative to:         Printable area
Text properties:
    UnitSystem:                 Inches
    Font:                       Tahoma
    Text height:                0.20
    Stamp on single line:       No
```

3. Then the command displays a prompt line for changing the settings:

```
Options: Fields, LOCation, Log file, OFF, On, Text properties, UNits or User fields
Specify option» (Enter an option)
```

The options have the following meaning:

PrintStamp Option	Meaning
Fields	Stamps the drawing name, sheet name, date and time, login name, printer name, paper size, and/or print scale.
LOCation	Place print stamp at TL (top left), TR (top right), BL (bottom left), or BR (bottom right); relative to the page orientation, upside-down, offset by a distance, and whether the offset is relative to the paper's border or the printable area
Log file	Write the print stamp data to a log file
OFF	Disables the print stamp
On	Enables the print stamp during printing
Text properties	Specifies the font name, height, and whether the stamp is on a single line
UNits	Changes units between inches, millimeters, and pixels
User fields	Allows two user-defined fields to be defined

4. For this tutorial, turn on the following fields:

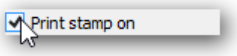
```
Specify option» f
Default: No
Confirm: Include drawing name?
Specify Yes or No» y
Default: No
Confirm: Include sheet name?
Specify Yes or No» n
Default: No
Confirm: Include date and time?
Specify Yes or No» y
Default: No
Confirm: Include login name?
Specify Yes or No» y
Default: No
Confirm: Include printer name?
Specify Yes or No» y
Default: No
Confirm: Include paper size?
Specify Yes or No» n
Default: No
Confirm: Include print scale?
Specify Yes or No» y
```

5. Press **Enter** once more to exit the command.

USING PRINT STAMPS

To apply a print stamp to a printed drawing, turn on the option in the Print dialog box, like this:

1. Enter the **Print** command.
2. Ensure that you select a printer from the **Name** droplist. If the printer is lists as "<None>," then the print stamp option is unavailable.
3. Click **Advanced Options**.
4. In the Options section, turn on **Print Stamp On**.



Turning on print stamps for the next plot

5. Click **OK** to exit.
6. Click **Print Preview** to check how the print stamp looks.

TIP If no print stamp appears on the printed drawing, this means no stamp was defined by the **PrintStamp** command.

Summary

You learned in this chapter all there is know about printing drawings — probably too much information! In the next chapter you return to learning drafting, specifically how to draw and edit reusable symbols.

Notes

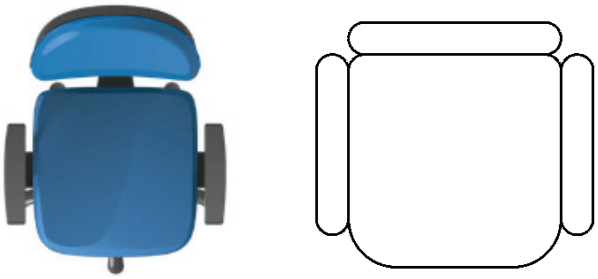
CHAPTER 07

Adding Symbols

In This Chapter

- Drawing symbols
- Converting symbols to blocks
- Inserting blocks
- Downloading blocks from Web sites
- Extracting blocks from drawings

With the walls of the office in place, it is time to populate the office with furnishings, like desks, chairs, windows, and doors. These are usually drawn as symbols, which look like the real objects, but seen from above, such as this office chair:



Left: *Office chair;* ***right:*** *symbol as a simplified representation of the chair*

Symbols (a.k.a. blocks) can shared by inserting them into other drawings as reusable blocks. In this chapter, you learn how to draw a variety of symbols common to office layouts, and then add them to the *office layout.dwg* drawing as blocks.

KEY TERMS IN THIS CHAPTER

Block refers to a collection of entities that is treated as one (a.k.a. symbol).

Block Definition defines a block; invisible in the drawing.

Display Order defines which overlapping entities are visually "on top."

Explode refers to the reduction of complex entities to basic equivalents.

Exploded Block refers to inserting constituent entities, instead of the block.

Fillet rounds corners.

Insert places blocks in drawings.

Mask refers to colored areas (usually white or black) meant to hide objects underneath them.

Mirror mirrors copies of entities.

Pattern copies entities in evenly-spaced horizontal, vertical, or circular patterns (a.k.a. arrays).

Symbol refers to a graphical representation of a real world object (a.k.a. block).

Uniform Scale refers to blocks that have the same x, y, and z scale factor.

Unit Block refers to a block that fits inside a 1x1-unit square.

COMMANDS

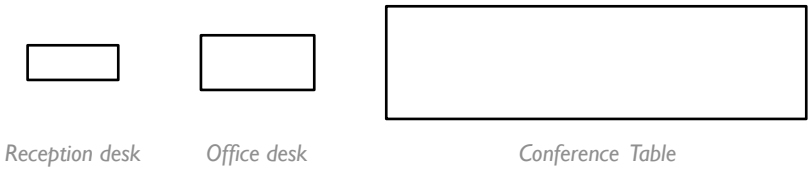
Command	Shortcuts	Menu Selection
Arc	a	Draw | Arc
DesignResources	Ctrl+2	Tools | Design Resources
DisplayOrder	dr, draworder	Tools | Display Order
Explode	x	Modify | Explode
ExportDrawing	w, wblock	File | Export | Export DWG
Fillet	f	Modify | Fillet
InsertBlock	i, insert	Insert | Block
MakeBlock	b, block	Draw | Block | Make Block
Mask	wipeout	Draw | Mask
Mirror	mi	Modify | Mirror
Pattern	ar, array	Modify | Pattern

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- **EditCustomBlock** opens a new block editor that creates and edits custom blocks, which features visibility states, element sets, and constraints. See the next chapter on how to use them. Custom blocks are also known as dynamic blocks.
- **CbClose** closes the block editor.
- **MirrorHatch** command determines whether hatches and gradient fills are reflected or retain their pattern direction when the Mirror and Flip commands are applied.

Drawing Desk Symbols

The desk is perhaps the most basic of symbols, and so the easiest to draw. It consists of just a rectangle! To represent desks that are small, large, or even as enormous as a conference table, the rectangle is placed at various sizes, as illustrated below.



A simple rectangle representing a variety of table types

If you were to draw one symbol for every possible size of desk, you would end up with many, many rectangles — and such a large number of items becomes unmanageable. Instead, you need only draw one, and then use a trick that I'll show you in this chapter to make it suitable for any size.

The desk symbol is drawn with the **Rectangle** command, and then an opaque filling is added with the **Mask** command. The mask becomes useful later when you add chair symbols to the desks)

> **TIP** The program uses the term "block" in place of *symbol*, following the practice of other CAD programs, like AutoCAD and IntelliCAD. In this book, I use "symbol" when speaking of a group of objects before they are unified into a block.

DRAWING UNIT-SIZE DESKS

Command: Mask
Aliases: wipeout
Menu bar: Draw | Mask
Ribbon: --
Toolbar: --

The trick to placing different sized symbols is called the "unit block." This is a symbol drawn to fit a square, usually sized 1mm by 1mm. Later, when we insert the block in a drawing, we specify the actual size of the desk, such as 600mm by 900mm. The CAD program resizes the 1x1mm block automatically. In this way, a single block can represent a desk, cabinet, or other rectangular shape of any size!

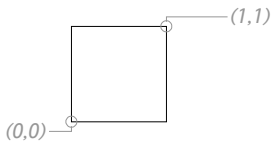
To draw a unit block, follow these steps.

1. Start a new drawing with the **New** command, and then choose *standardiso.dwt* from the Specify Template dialog box.
2. Use the **SaveAs** command to save the drawing with the file name of *office desk.dwg.*
3. Enter the **Rectangle** command, and then draw a rectangle sized 1mm by 1mm, as follows:

```
: rectangle
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Par-
allelogram, Thickness, line Width or Specify start corner
Specify start corner» 0,0
Options: Area, Dimensions, Rotation or
Specify opposite corner» 1,1
```

4. To see the rectangle, zoom the drawing so that it fits the window. (Feel free to use aliases, as I do to reduce the number of keystrokes.)

```
: z
ZOOM Default: Dynamic
Options: Bounds, Center, Dynamic, Fit, Previous, SElected, specify a scale factor
(nX or nXP) or
Specify first corner» f
```



A unit block sized 1 x 1 units

TIP If you prefer entering the options of the **Zoom** command as found in AutoCAD or IntelliCAD, the this program accepts them as well. For example, to zoom a drawing to its *extents*, you can enter **z e**.

5. Now fill the rectangle with a solid white mask using the **Mask** command:

```
: mask
```

a. This command creates new mask outlines with the 'Specify start point' option, or else can use existing entities to define the boundary of the mask. In this tutorial, you use the rectangle to define the mask. Because the rectangle is made of a polyline, enter **p** for the Polyline option:

```
Default: Polyline
Options: Frames, Polyline or
Specify start point» p
```

b. Select the rectangle:

```
Specify entity» (Select the rectangle)
```

c. The program can keep or erase the source entity, the rectangle. For this tutorial, keep the polyline, because it represents the outline of the desk:

```
Default: No
Confirm: Keep polyline?
Specify Yes, or No» y
```

6. Press **Ctrl+s** to save your work (**Cmd+s** on Macs).

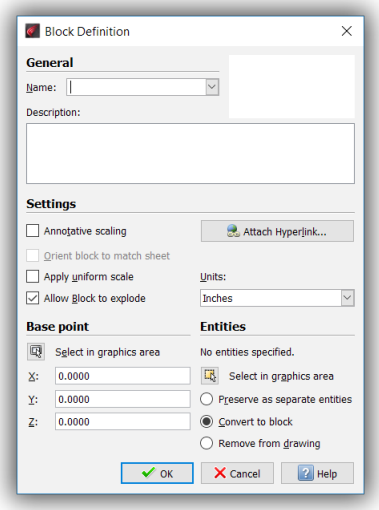
The desk symbol looks no different, because the mask is white. But it will come in handy later in this chapter when you place chairs that rest partially under the desks.

CONVERTING SYMBOLS INTO BLOCKS

Command: MakeBlock
Aliases: b, block, mblck, partdef, bmake
Menu bar: Draw | Block | Make Block
Ribbon: Insert | Block Definition | Block Description (Drafting and Annotation)
Toolbar:

The desk symbol consists of two entities — a rectangle and a mask — and so it is useful to combine them into a single entity. It is useful, because you can use it repeatedly without having always be drawing rectangles and filling them with masks. The program converts symbols to blocks with the **MakeBlock** command.

1. Start the **MakeBlock** command using one of the methods listed above. Notice the Block Definition dialog box.



Dialog box for converting entities into blocks, before any info has been entered

   The dialog box has many options, but you need provide only three pieces of information, as a minimum. This is all the program needs to convert a selection of entities into a block:

   i. The name, by which the block is identified

   ii. The entities, to be converted into the block

   iii. The insertion spot, where the block is later placed in drawings

2. In the **Name** field, enter "unit desk." This name identifies the block later when it comes time to insert it into the office plan drawing.

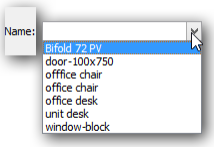
3. To choose the entities to be converted into a block, follow these steps:

   a. Click the **Select in Graphics Area** button found in the Block Entities section.

   b. Notice that the dialog box disappears, and that the program prompts you in the command window. Select the two entities, and the press **Enter**.

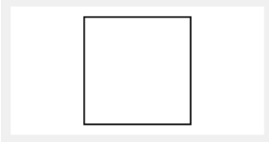
```
Specify entities» (Choose the rectangle and mask)
Specify entities» (Press Enter to return to the dialog box)
```

TIP When a drawing contains two or more blocks, you can click the droplist arrow to see their names. This ensures that you do not accidently use an existing block name.



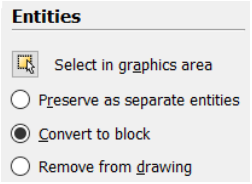
If you do reuse an existing block name, the block associated with the name is redefined with the new entities you select. Every instance of the block is redefined with the new one — a powerful but dangerous feature.

c. Up in the upper right corner, notice that the program displays a thumbnail image of the entities you selected.



Preview of entities selected

4. There are several options in the Block Entities section of which you should take note:



Options for dealing with entities

Preserve as Separate Entities leaves the entities in the drawing as entities; the block definition is stored in the drawing, but is not visible until you use the InsertBlock command.

Use this option when you plan to make several similar blocks, reusing and modifying the existing sybmols.

Convert to Block replaces the entities with a block insertion. This is a convenient option, because it combines the following commands: Erase to remove the entities, followed by InsertBlock to insert the block.

Use this option when you are creating one-of-a-kinds blocks for use in the current drawing.

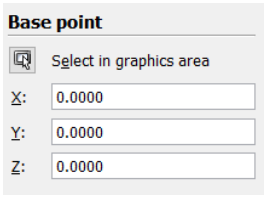
Remove from Drawing removes the entities from the drawing, leaving only the block definition (invisible until inserted with the InsertBlock command). This option tends to dismay new users, who think their new block has disappeared!

Use this option when you are creating a block library, a drawing that contains a lot of blocks that you use repeatedly in other drawings.

For this tutorial, keep the default — **Preserve as Separate Entities** — because you will use this drawing file as the source for the block. (Later, you see how to insert blocks from other drawing files.)

5. Choose the *base point* for the block. There has to be a spot in the block that designates where it is inserted in the drawing. For this table block, specify the lower left corner, as follows:

a. Click the **Select in Graphics Area** button found in the Base Point section.



Picking the block's base point

b. Notice that the dialog box disappears, and that the program prompts you in the command bar. Pick the lower left corner with the help of the **Intersection** entity snap, and the press **Enter**.

```
Specify insertion base point» int
of (Pick the lower left corner of the rectangle)
```

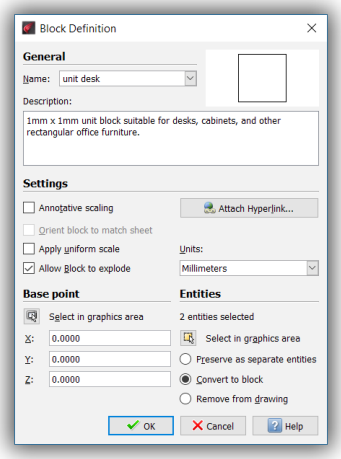
c. When the dialog box returns, notice that the **X**, **Y**, and **Z** coordinates are filled in. (Because you drew the rectangle starting at 0,0, the values reported by X, Y, Z are likely to be 0,0,0.)

6. Ensure the following options are left at their default values:

Make Block Option	Setting
Apply Uniform Scale	Off
Allow Block to Explode	On
Attach Hyperlink	Ignore
Units	Millimeters

All other items in this dialog box are optional. For instance, you can describe the purpose of the block in the **Description** field, if you wish.

7. Now the content of the dialog box should be like the one illustrated below:



Make Block dialog box with information filled in

8. Click **OK** to close the dialog box and define the block.

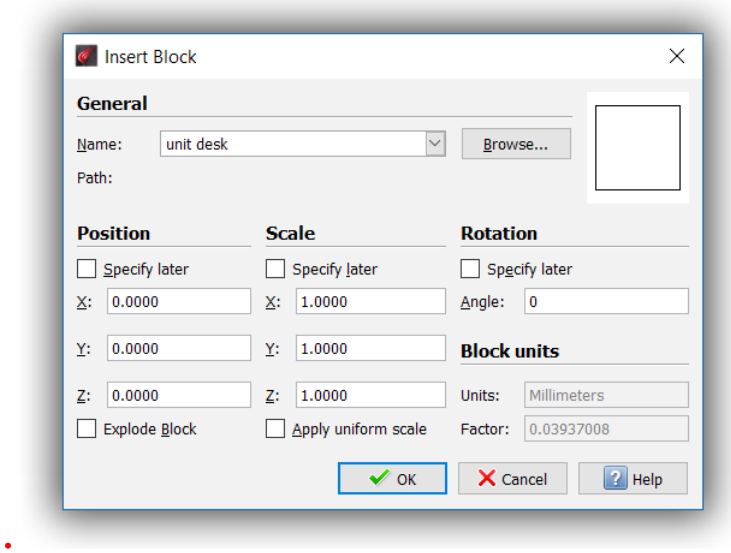
The drawing looks no different. In the next tutorial, you learn how to insert this newly-created block into the drawing.

INSERTING BLOCKS INTO DRAWINGS

Command: InsertBlock
Aliases: i, insert
Menu bar: Insert | Block
Ribbon: Insert | Insert Block (Drafting and Annotation)
Toolbar:

Blocks are placed in drawings with the **InsertBlock** command.

1. Start the **InsertBlock** command using a method from above. Notice the Insert Block dialog box:



Dialog box for inserting blocks into drawings

Like the Block Definition dialog box, this dialog also has many options, but there are just two bits of info that you must provide as minimum before the program can insert a block:

i. The name of the block to be inserted

ii. The position in the drawing at which place the block

Because you are working with a unit block, you must also specify the following:

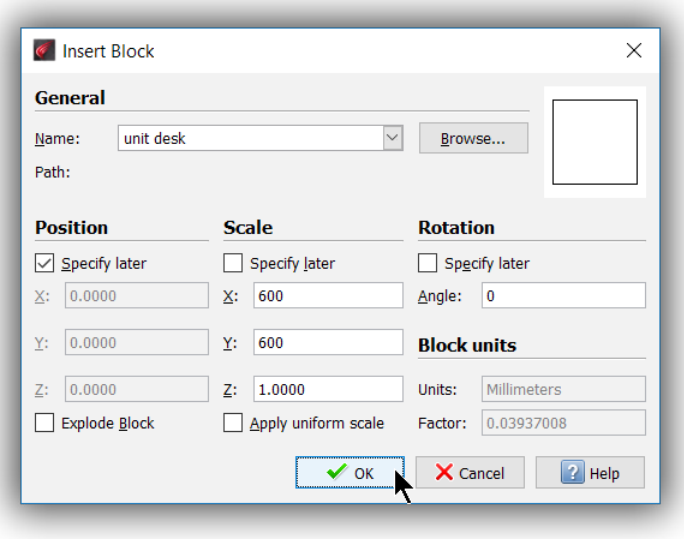
iii. The scale (size) of the block

2. From the **Name** droplist, choose "unit desk."

When a drawing contains just one block, its name is automatically shown. When a drawing contains many blocks, the Preview image helps you distinguish them visually.

TIP You can insert any other drawing as a block. To do so, click **Browse**, and then in the Open dialog box choose a DWG, DXF, or FLX file. The drawing is treated just like a block. (The name of folder in which the file resides is listed next to **Path** in the Insert Block dialog box.)

3. To specify a value for the **Position** of the block, turn on the **Specify Later** option. (After you click **OK**, the program will prompt you to place the block.)
4. To place this 1mm x 1mm unit block as a 600mm x 900mm desk, change the values in the **Scale** section.



Insert Block dialog box with parameters filled in

You can leave all other options are they are:

Insert Block Option	Setting
Scale X	600
Scale Y	900
Apply Uniform Scale	Off (When on, this option forces the Y and Z scale factors to be the same as X)
Rotate Specify Later	Off (When on, this option prompts you for the block's rotation angle in the command window after exiting the dialog box)
Angle	0 (When no 0, this is the value by which the block is rotated)
Explode Block	Off (When on, this option causes the constituent entities to be inserted, instead of the block itself. This is useful if you later want to edit the block)

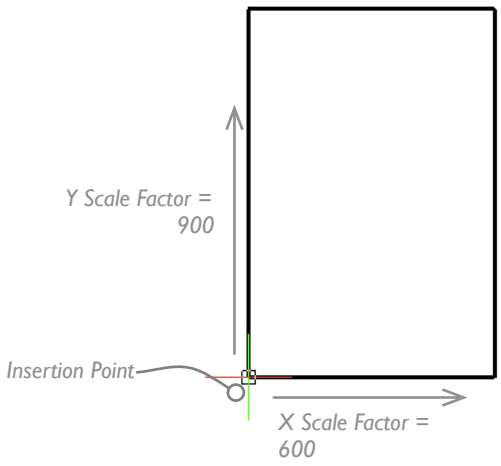
At this point, the options in the dialog box should look like the one illustrated above.

TIP To make the unit block smaller, use **Scale X** and **Y** with values less than 1. For instance, a scale factor of 0.5 makes the block half as large. You cannot enter 0 as a scale factor; you can, however, enter negative numbers, such as X = -600 to mirror the block about the x axis.

5. Click **OK** to close the dialog box.
6. Notice the prompt in the command window:

```
Options: Angle, reference Point, uniform Scale, X, Y, Z or
Specify destination» (Pick a point in the drawing)
```

Move the mouse. Notice that the block moves around the drawing. The cursor is stuck at the block's insertion point, the location at which you had defined with the MakeBlock command.



Applying parameters to place the block in the drawing

7. When you pick a point in the drawing, the program places the block.

TIP You are prompted for a destination point, because you turned on that option in the Insert Block dialog box. No other prompts are displayed, such as scale factor or rotation angle, because you left those options turned off. You can, however, override the defaults set in the dialog box by entering the **Angle**, **reference Point**, **uniform Scale**, **X**, **Y**, and/or **Z** options.

8. Press **Ctrl+s** to save the drawing file (**Cmd+s** on Macs).

With the block placed, the InsertBlock command ends. In later tutorials, you draw an office chair and other symbols, and then add them to the office layout.

Creating Symbols, Step-by-Step

The following tutorials build blocks representing a chair, door, and window. Here you learn how to use additional editing commands, such as Trim, Mirror, and Fillet.

CHAIR BLOCK

Command: Explode
Alias: x
Menu bar: Modify | Explode
Ribbon: Home | Modify | Explode (Drafting and Annotation)
Toolbar:

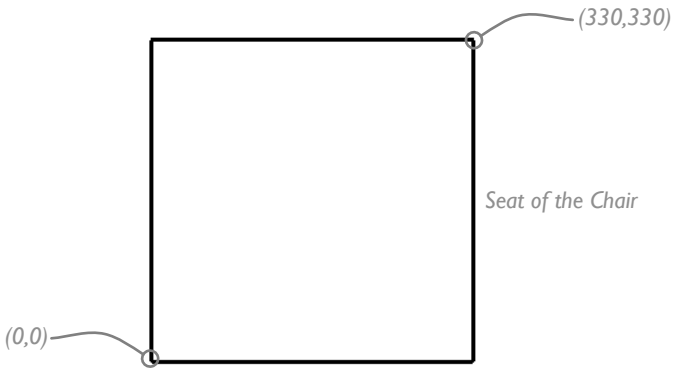
Unlike table symbols, chair symbols must be drawn full-size. They don't work well as unit blocks, because of their shape. A chair inserted with scale factors of 200 and 300 just plain looks wrong. In this tutorial, you draw an office chair full-size using commands that may be new to you.

1. Start a new drawing with the *standardiso.dwt* template file.

> **TIPS** There is no need for the UCS icon, so turn it off with the **CsIcon Off** command.
> The drawing is easier to see when lineweights are turned on, so click the **LWT** button on the status bar. The other status bar buttons can be left off, including Snap, Grid, Ortho, and ESnap.

2. Draw the seat of the chair as a rectangle between 0,0 and 330,330.

```
: rectangle
Options: Chamfer, Elevation, Fillet, Thickness, line Width or
Specify start corner» 0,0
Options: Area, Dimensions, Rotation or
Specify opposite corner» 330,330
```



First step in drawing a chair symbol

3. Use the Zoom command twice to see the rectangle.
   a. First, use the **Fit** option to have the rectangle fill the screen:

```
: z
ZOOM Default: Dynamic
Options: Bounds, Center, Dynamic, Fit, Previous, SElected, specify a
scale factor (nX or nXP) or Specify first corner» f
```

b. Second, make the drawing 70% smaller to add some space around the rectangle:

```
: (Press spacebar to repeat the Zoom command)
ZOOM Default: Dynamic
Options: Bounds, Center, Dynamic, Fit, Previous, SElected, specify a
   scale factor (nX or nXP) or Specify first corner» 0.7
```

4. You need to explode the rectangle to change its polyline into lines, so that they can be correctly offset later in this tutorial.

   a. Start the **Explode** command using one of the methods listed above:

```
: explode
```

   b. At the 'Specify entities' prompt, you select the rectangle. You can move the mouse over to pick it, but I find it's faster to enter **L** to select the last-drawn entity:

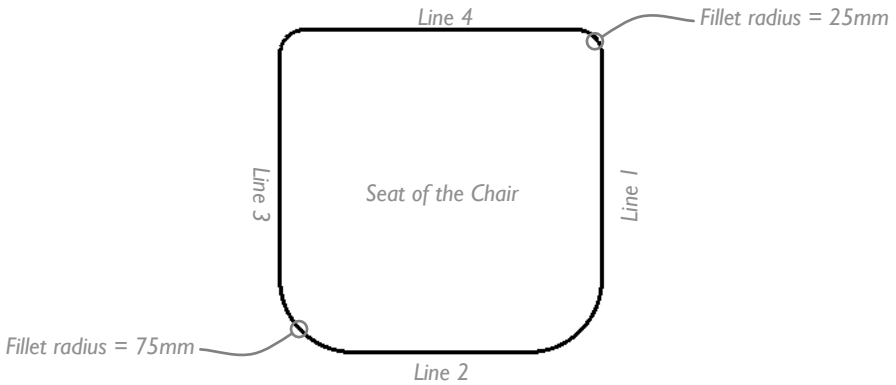
```
Specify entities» l
1 found, 1 total
Specify entities» (Press Enter to end the command)
```

   The rectangle looks no different after being exploded.

ROUNDING CORNERS

Command: Fillet
Alias: f
Menu bar: Modify | Fillet
Ribbon: Home | Modify | Trim | Fillet (Drafting and Annotation)
Toolbar: 

Use the **Fillet** command to round the corners of the seat. In this tutorial, you apply two different radii to the corners of the seat.



Rounding corners with the Fillet command

1. Enter the Fillet command. Notice that the default radius is 10mm:

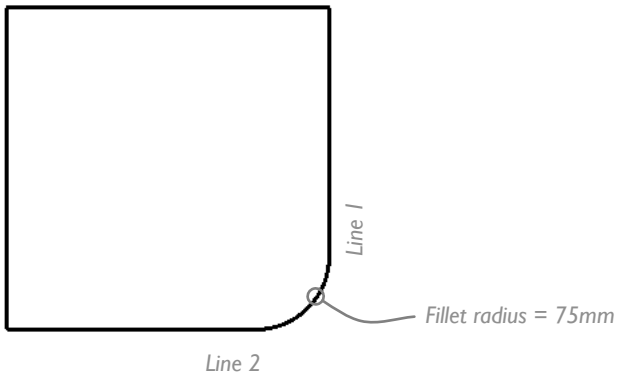
```
: fillet
Mode = TRIM, Radius = 10
```

2. Enter 'r' to change the radius to 75 (mm):

```
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity» r
Default: 10
Specify radius» 75
```

3. Select the two lines indicated by the figure:

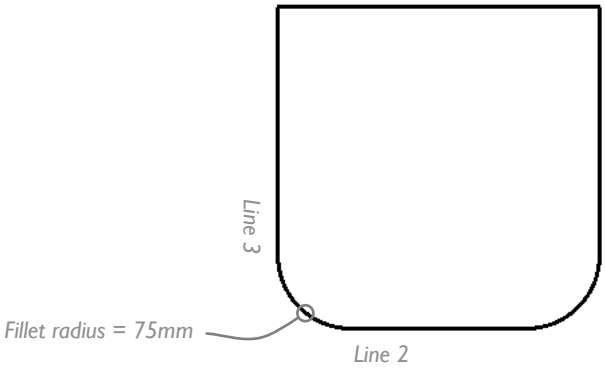
```
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity» (Pick line #1, as shown below)
Options: Shift + select to apply corner or
Specify second entity» (Pick line #2)
```



Placing a 75mm fillet radius

4. The Fillet command ends automatically, so you need to restart the command to place the next fillet. The easiest way to do this is to press the spacebar:

```
: (Press the spacebar to repeat the Fillet command)
FILLET Mode = TRIM, Radius = 75
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity» (Pick line #2, as shown below)
Options: Shift + select to apply corner or
Specify second entity» (Pick line #3)
```



Adding the second fillet

5. The seat has two more fillets, but of a different radius. First, though, use the **Multiple** option to make the command repeat itself:

```
: (Press the spacebar to repeat the Fillet command)
FILLET Mode = TRIM, Radius = 75
```

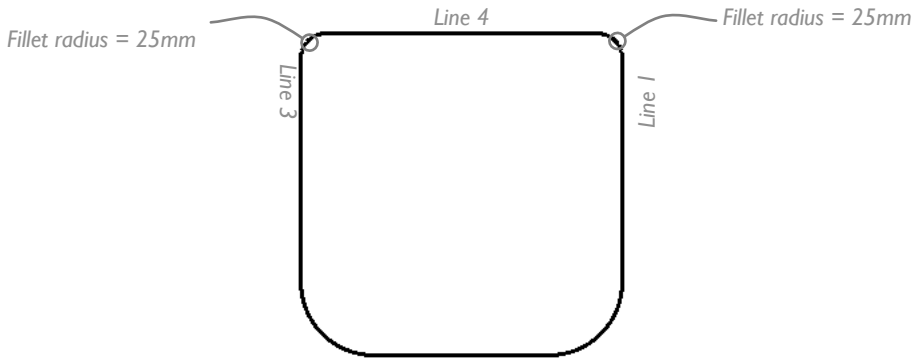
```
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity» m
```

6. Use the **Radius** option to change the radius to 25mm:

```
Specify first entity» r
Specify radius» 25
```

7. Fillet the remaining corners:

```
Specify first entity» (Pick line #3, as shown below)
Specify second entity» (Pick line #4)

Specify first entity» (Pick line #1)
Specify second entity» (Pick line #4)
```



Adding two 25mm fillets

8. Save the drawing as *office chair.dwg* with the **SaveAs** command.

```
: saveas
```

MIRRORING ARMS AND BACKS

Command: Mirror
Alias: mi
Menu bar: Modify | Mirror
Ribbon: Home | Modify | Copy | Mirror (Drafting and Annotation)
Toolbar: 

The two arms and the back of the chair are drawn by offsetting existing lines, filleting parallel lines, and then mirroring some of the entities.

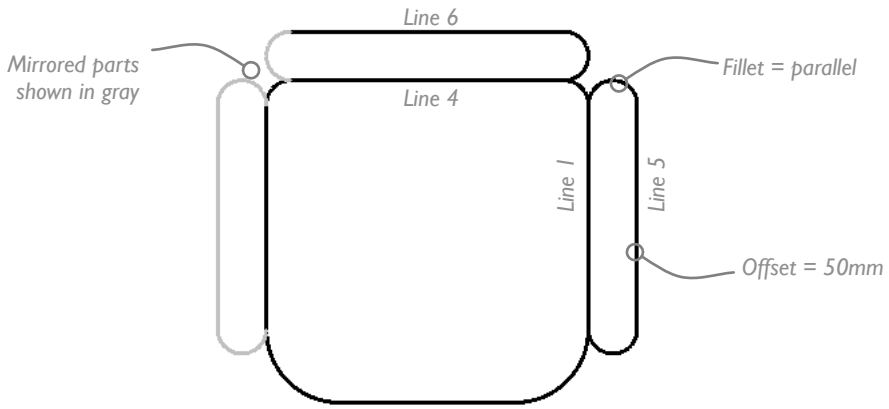
As you saw in an earlier chapter, the Offset command makes parallel copies. You use it here to draw the straight portions of one arm and the back.

1. The arms are 50mm wide. Draw one of them. (The second arm is copied later with the Mirror command.)
   a. Start the **Offset** command:

```
: offset
Active settings: Delete source=No Layer=Source OFFSETGAPTYPE=0
```

b. The default offset distance is 1, so change it to 50:

```
Default: 1
Options: Delete, DIstances, destination Layer, Through point or
Specify distance» 50
```



Drawing an arm (right) which will be mirrored at the left

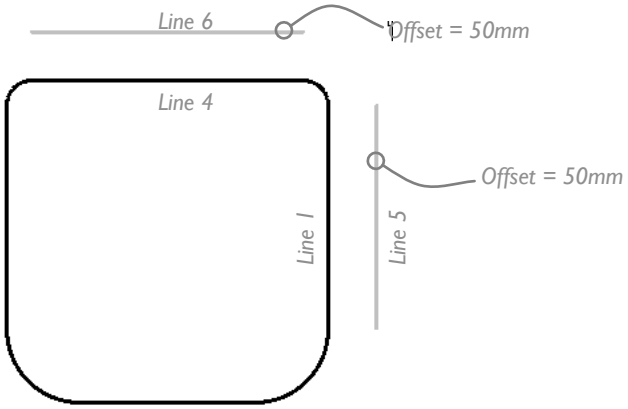
c. Select line #1, as shown above:

```
Default: Exit
Options: Exit, Undo or
Specify source entity» (Pick line #1, as shown below)
```

d. And then pick a point to the right:

```
Default: Exit
Options: Both sides, Exit, Multiple, Undo or
Specify side for destination» (Pick a point to the right, as shown below)
```

Notice that the program makes the offset copy 50mm to the right.



Creating lines by offsetting existing lines

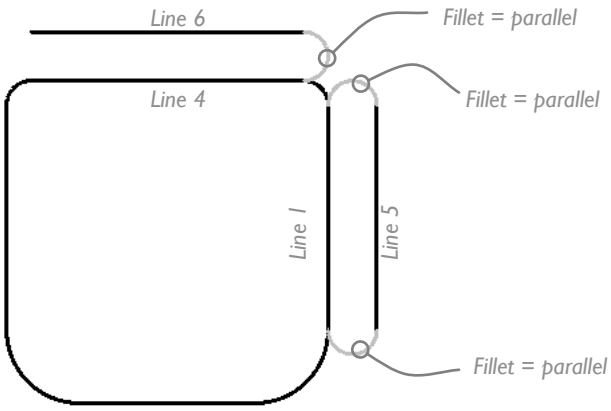
e. The **Offset** command repeats itself, unlike the Fillet command. Show the program where to place the second offset line:

```
Default: Exit
Options: Exit, Undo or
Specify source entity» (Pick line #4)
Default: Exit
Options: Both sides, Exit, Multiple, Undo or
Specify side for destination» (Pick a point to the top)
```

f. Press **Enter** to exit the command:

```
Default: Exit
Options: Exit, Undo or
Specify source entity» (Press Enter to end the command)
```

2. The curves of the arm and back are made with the Fillet command. In this case, you do not specify the radius, because your are filleting parallel lines; the distance between them determines the radius, automatically.



Placing fillets in parallel mode

```
: fillet
Mode = TRIM, Radius = 25
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity»  (Pick line #1, as shown above)
Options: Shift + select to apply corner or
Specify second entity» (Pick line #5)
```

3. Repeat the **Fillet** command twice more to place two more parallel fillets:

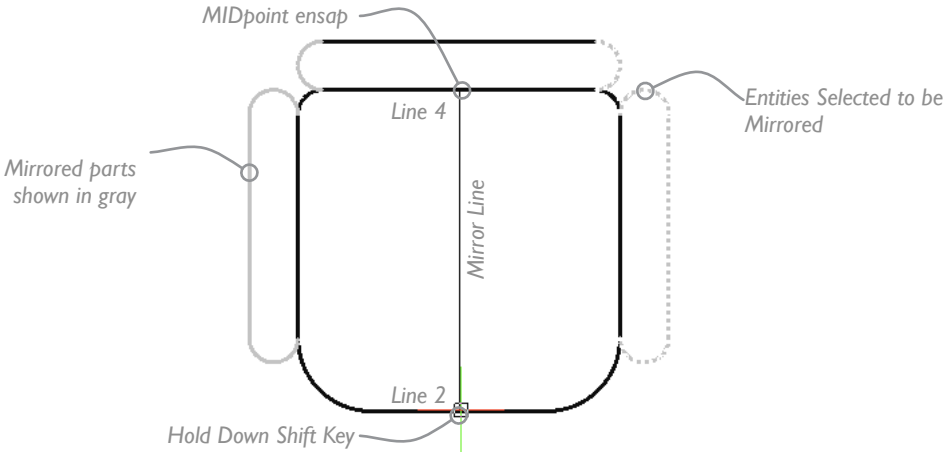
```
: (Press Enter to repeat the Fillet command)
FILLET Mode = NOTRIM, Radius = 25
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity» (Pick line #1)
Options: Shift + select to apply corner or
Specify second entity» (Pick line #5)

: (Press spacebar or Enter to repeat the Fillet command)
Mode = NOTRIM, Radius = 25
Options: Multiple, Polyline, Radius, Trim mode, Undo or
Specify first entity» (Pick line #4)
Options: Shift + select to apply corner or
Specify second entity» (Pick line #6)
```

4. Because the chair symbol is symmetric, you don't have to draw all of it. The chair can be completed with the **Mirror** command, which makes mirrored copies. This command needs you to specify two points that define an invisible line about which the entities are mirrored.

a. Start the **Mirror** command using one of the methods listed above.

```
: mirror
```

b. Select the entities to be mirrored, as illustrated by dotted lines in the figure below:



Selecting entities that will be mirrored

```
Specify entities» (Select the four entities shown in the figure above)
4 found, 4 total
Specify entities» (Press Enter to end entity selection.)
```

c. Pick the two points that indicate the mirror line. In this case, the line should be vertical and lie along the precise center of the chair. Use **Midpoint** entity snap to ensure accuracy:

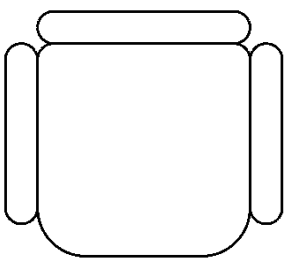
```
Specify start point of mirror line» mid
of (Pick a point roughly in the middle of line #4)
```

d. It is good enough to pick the second point precisely below the first. The easiest way to do this is with ortho mode. You turn on ortho mode temporarily by holding down the **Shift** key, like this:

```
Specify end point of mirror line» (Hold down the Shift key, and then
pick a point below the start point)
```

e. Since you want to keep the source entities (the ones you selected earlier in step b), enter 'n' at the final prompt:

```
Default: No
Confirm: Delete source entities?
Specify Yes, or No» n
```



Chair symbol completed

5. Press **Ctrl+s** to save the drawing file (**Cmd+s** on Macs).

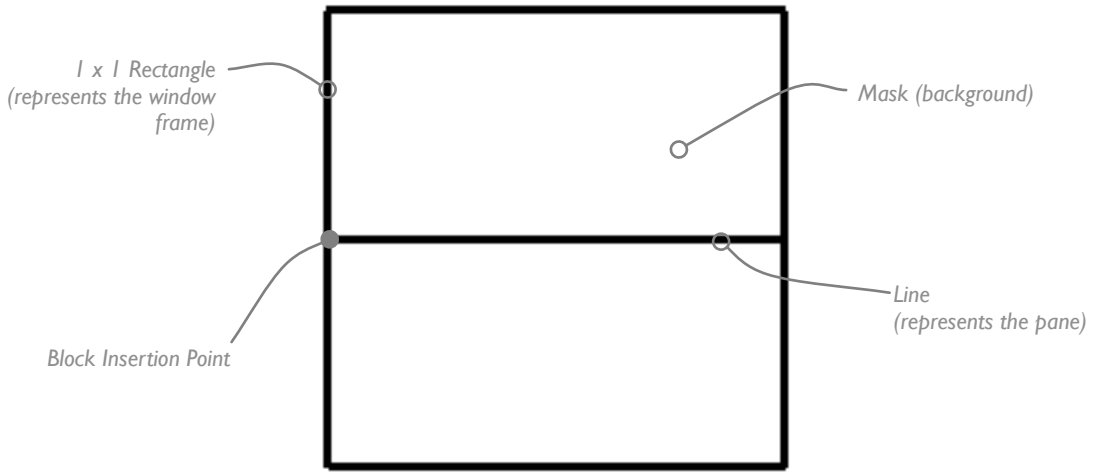
MAKING WINDOW AND DOOR BLOCKS

Use the commands you have learned so far to create the window and door symbols for the walls.

Windows typically are drawn as simple blocks, especially for the fixed windows that are commonly used for offices. One line represents the pane of glass, and two more represent the sides. The block should be drawn as a unit block using the **Line** and **Mask** commands. (The mask hides wall lines.)

Follow these guidelines:

1. To ensure the unit block is 1mm tall and wide, first draw a rectangle sized 1x1 with the **Rectangle** command.



Elements of a unit window block

2. Convert the rectangle to a mask with the **Mask** command's Polyline option.
3. Use the **Line** command to draw the pane and frame.
4. Save the drawing as *window-block.dwg*.

TIP Always draw blocks on layer 0. This ensures the block will be inserted on the active layer of the drawing. (If you draw the block on a layer other than 0, such as "blocks," then the program will always insert it on layer Blocks.)

When inserting the block, use the x and y scale factors to size the block correctly. Remember that there are two wall thicknesses, 150mm (outer walls) and 100mm (inner walls):

X scale factor sizes the window for the specified **width**, such as 1200mm.
Y scale factor makes the window **deep** enough for the wall (150 or 100mm)
Rotate rotates the block by 90 degrees for walls running **vertically**.

Top: *X scale factor = 150, Y scale factor = 1000*
Bottom: *X scale factor = 100, Y scale factor = 1200*

Drawing Door Symbols

Command: Arc
Alias: a
Menu bar: Draw | Arc
Ribbon: Home | Draw | Arc (Drafting and Annotation)
Toolbar:

Command: DisplayOrder
Aliases: dr, draworder
Menu bar: Tools | Display Order
Ribbon: View | Order (Drafting and Annotation)
Toolbar:

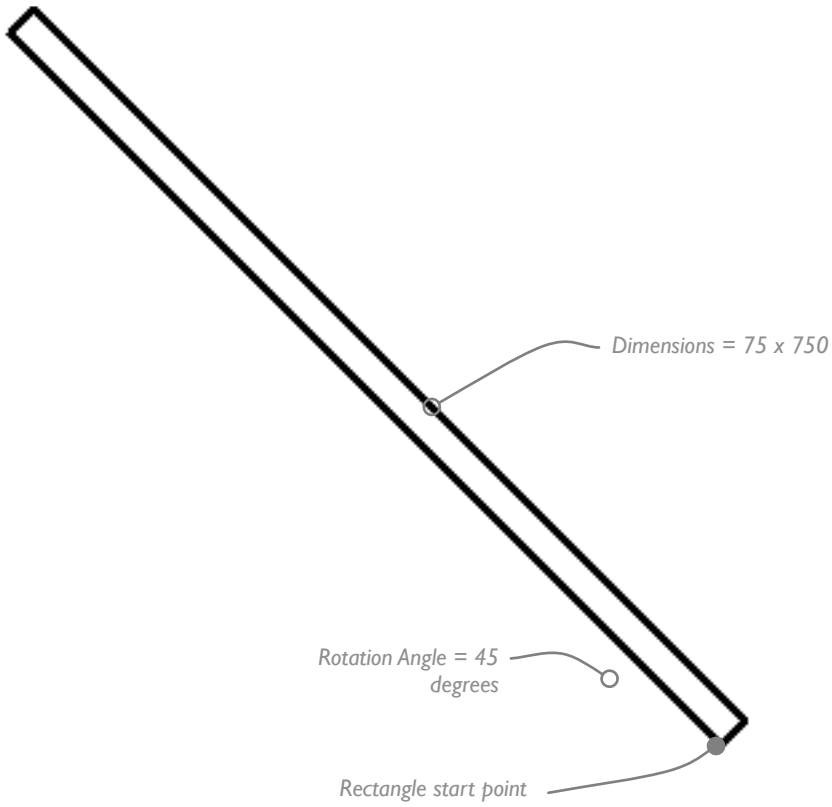
Door blocks use an arc to show the door swing. The arc, however, prevents doors from being drawn as unit blocks, because circular arcs end up looking elliptical when placed with unequal x and y scale factors.

Instead, you need to draw one door block for each size and type of door. For example, a regular door is 750mm wide, while a narrow door is 650mm. A bi-fold door is drawn like a W. Doors that swing in or out can be flipped using a negative scale factor. Below, I have illustrated some door styles.

Follow these guidelines:

1. Start the door symbol with **Rectangle** command:

```
: rectang
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Par-
allelogram, Thickness, line Width or Specify start corner
Specify start corner» (Pick a point in the drawing)
```



Elements of a door block

a. To show the door in a partially open position, use the **Rotation** option to rotate the door by 45 degrees:

```
Options: Area, Dimensions, Rotation or
Specify opposite corner» r
Default: 0
Options: Pick points or Specify rotation» 45
```

b. Draw the door with the **Dimensions** option. Specify a thickness of 25mm and width of 750mm:

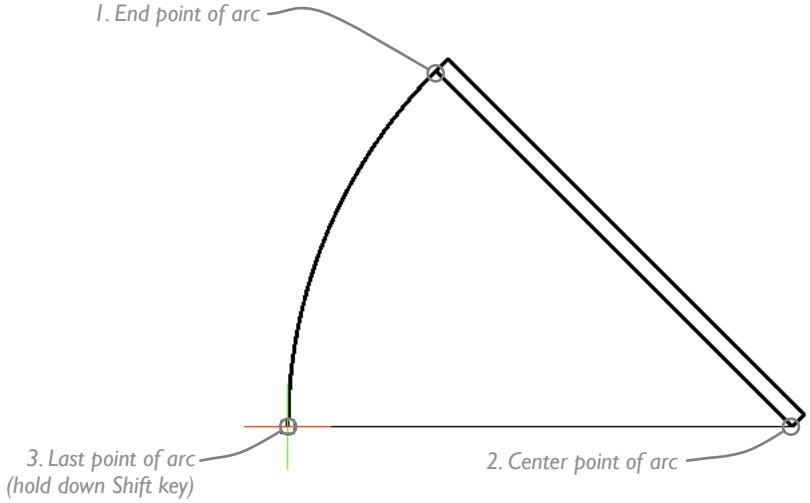
```
Options: Area, Dimensions, Rotation or
Specify opposite corner» d
Default: 10 Specify horizontal dimension» 25
Default: 10 Specify vertical dimension» 750
```

2. Draw the door swing with the **Arc** command:

```
: arc
```

a. To specify the arc's start point accurately, use **Endpoint** entity snap:

```
Options: Center, Append, Enter to continue from last point or
Specify start point» end
of (Pick corner #1 of the rectangle)
```



Three points for drawing an arc

For the next point of the arc, use the command's **Center** option, and then use **Endpoint** entity snap to position it:

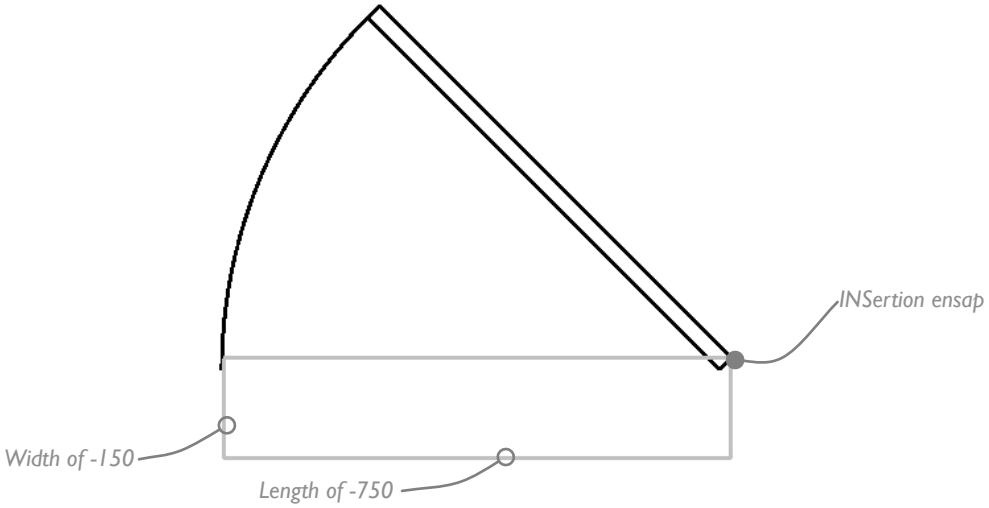
```
Options: Center, End or
Specify through point» c
Specify center point» end
of (Pick corner #2 of the rectangle)
```

For the third point of the arc, turn on Ortho mode to position the point exactly across from the arc's center point:

```
Options: Angle, chord Length or
Specify end point» (Click ORTHO on the status bar, and then click)
```

3. Place a second rectangle in the wall position. For 150mm-thick walls, dimension the rectangle as **Horizontal** = -750 and **Vertical** = -150; use negative dimensions to draw the rectangle to the left and downwards.

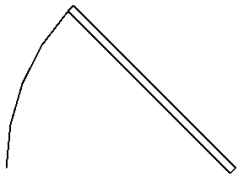
   Remember to reset the **Rotation** parameter to 0 degrees. Use **Insertion** entity snap to start the rectangle at door corner 2:



Elements of the mask that hides the wall

4. Convert the rectangle to a mask with the **Mask** command's Polyline option.
5. The mask is on top of the door, obscuring it partially. Use the **DisplayOrder** command to move the mask visually below the door with the **Back** option:

```
: displayorder
Specify entities» (Select the mask entity)
Specify entities» (Press Enter to end entity selection)
Default: in Back
Options: Above entities, in Back, in Front, or Under entities
Specify option» b
```



The completed door symbol

6. Save the drawing as *door-100x750.dwg.*, which identifies it as a door that fits 100x750mm openings.
7. Create additional door blocks for 150mm-thick walls and of varying widths.

TIP The **GetBlockInfo** command reports information about blocks in drawings. The **ReplaceBlock** command replaces one block with another one, while the **ReplaceBlockByReference** replaces blocks with xrefs (externally-referenced drawings). All of these commands can be found in the XtraTools menu and ribbon tab.

Placing Many Furnishings at Once

Command: Pattern
Aliases: pat, ar, array
Menu: Modify | Pattern
Ribbon: Home | Draw | Pattern (Drafting and Annotation)
Toolbar:

You have created two furniture symbols, one of a table that can be used for any size of table, and one of a chair. Let's place them into the office drawing. (Later, you add the doors and windows.)

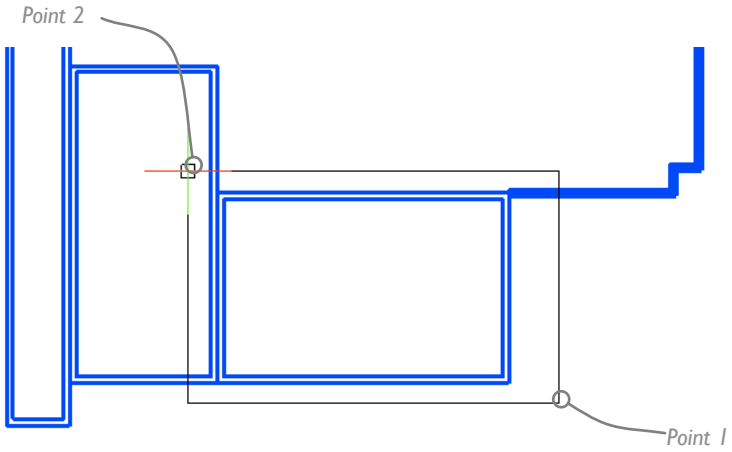
In this tutorial, you populate the conference room with a conference table and several chairs.

1. Open the *office plan.dwg* drawing file in the program.
2. Change the active layer to **Furnishings**.
3. Use the **Zoom Window** command to get a closer look at the conference room. The Window option is not listed in the prompt, but you can enter it anyhow:

```
: z
ZOOM Default: Dynamic
Options: Bounds, Center, Dynamic, Fit, Previous, SElected, specify a scale factor
(nX or nXP) or
Specify first corner» w
```

   The Zoom Window option requires you to pick two points, forming a rectangle. The rectangle specifies the zoomed-in view:

```
Specify first corner» (Pick point #1)
Specify opposite corner» (Pick point #2)
```

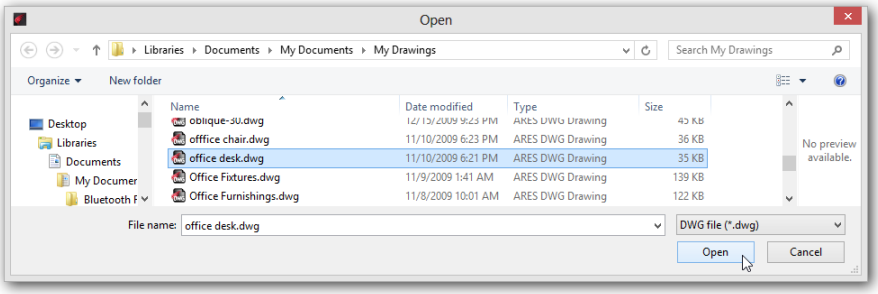


Picking two points to define the windowed zoom

4. Recall that blocks are placed with the InsertBlock command. Place a conference table sized 3800mm x 1400mm by following these steps:
   a. Start the **InsertBlock** command.

b. You need to access the "unit table" block stored in the *unit-table.dwg* drawing file. To do so, click **Browse**.

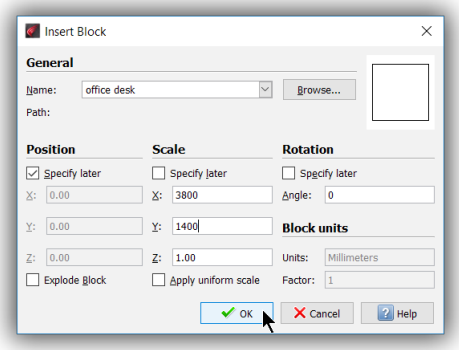
c. Notice the Open dialog box. Select the *office desk.dwg* file, and then click **Open**.



Browsing for blocks stored on the computer's drive

d. In the Insert Block dialog box, set the options for the block, as follows:

Insert Block Options	Setting
Position	Specify later
Scale X	3800
Scale Y	1400
Angle	Off

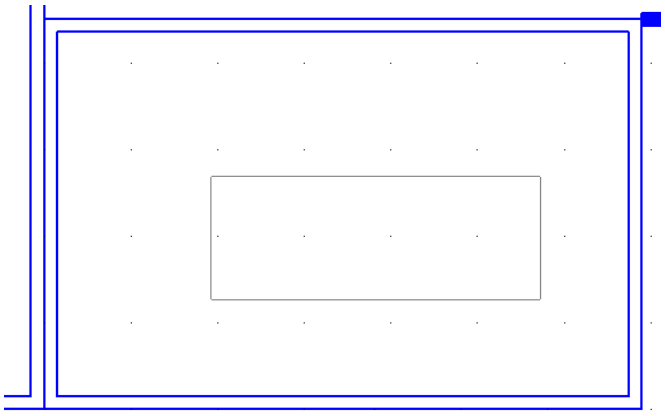


Insert Block dialog box filled out for inserting a desk

e. Click **OK**.

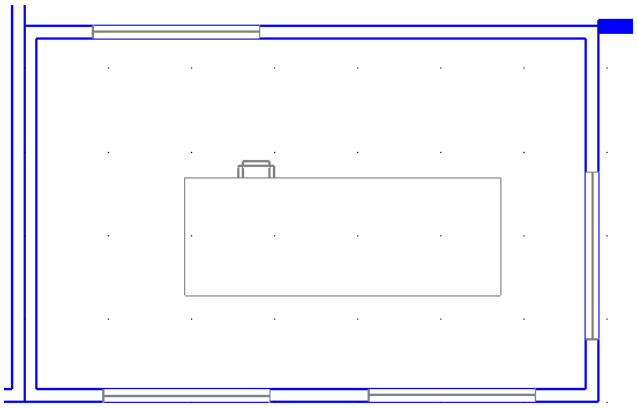
f. The program prompts you in the command bar:

`Specify destination» (Pick a point inside the conference room)`



Rectangle block placed as a table

5. Place the other blocks, such as chairs around the table, and doors and windows in the walls.

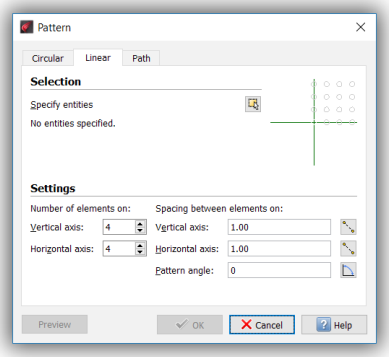


A chair, table, and window blocks added to the drawing

6. You could use the Insert command repeatedly to place more chairs. But the more efficient method is to use the **Pattern** command, because it makes multiple copies at a time. It can make the copies in straight lines, in a rectangular pattern, or around a circle. Follow these steps to add the three chairs:

   a. Start the **Pattern** command using a method listed above. Notice the dialog box.

   b. Click the **Linear** tab:



Dialog box sets parameters for creating linear patterns

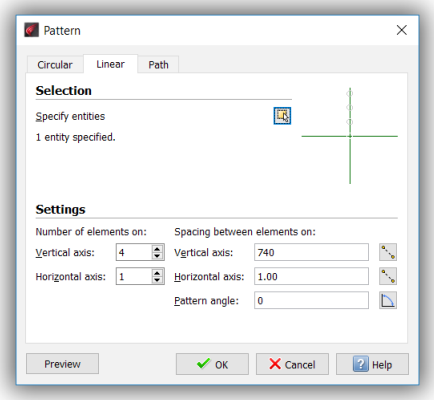
   c. Specify the following options; the preview window shows what the pattern looks like.

Array Options	Setting
Number of Elements	
Vertical Axis	4
Horizontal Axis	1
Spacing Between Elements	
Vertical	750 (value is in millimeters)
Horizontal Axis	1 (value does not matter)
Pattern Angle	0

TIP When you want three copies, you enter 4 for the number of elements. That's because the program counts the first element as the one already in the drawing — just like entering 1 for elements on the horizontal axis.

d. Click the **Select Entities** button. At the prompt, select the chair block:

```
Specify entities» (Select the chair block.)
1 found, 1 total
Specify entities» (Press Enter to return to the dialog box.)
```



Options enters for creating copies of chairs as a pattern

e. Click **OK** to add the chairs.

Three chair copies placed as a linear pattern

7. If necessary, use the **DisplayOrder** command to place the chair under the table.
8. Save your work.

Sourcing Ready-drawn Blocks

Even though blocks make it easy to place the same items into drawings repeatedly, it can become tedious drawing them from scratch. For this reason, there are many libraries of pre-drawn blocks available on the Internet. Your firm may have a collection of blocks. No matter on which computer they are located — or even in other drawings on your computer — the program accesses them through its Design Resources palette.

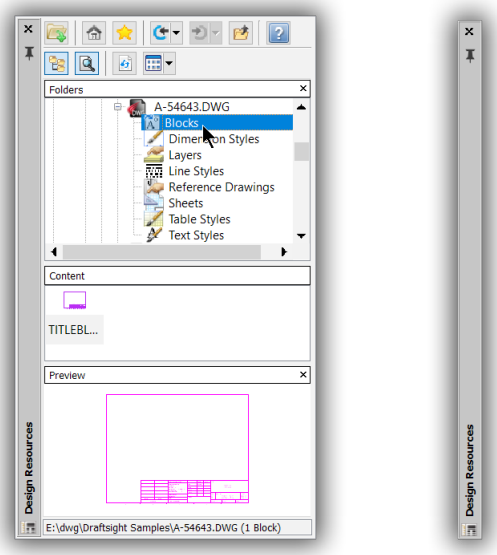
In addition, numerous companies and government agencies provide pre-drawn symbols free. Companies often provide symbols free in the hope that you will specify their products, boosting sales. Government agencies provide them free to encourage standardization. These you access through the Internet.

ACCESSING DESIGN RESOURCES

Command: DesignResources
Aliases: adcenter, drs
Menu: Tools | Design Resources
Ribbon: Insert | Resources | Design Resources (Drafting and Annotation)
Shortcut: Ctrl+2
Toolbar: 

The Design Resources palette allows the program to access the these resources from other drawings:

- Blocks
- Dimension, line, table, and text styles
- Layer names and associated properties
- Reference drawings (xrefs)
- Sheets (layouts)

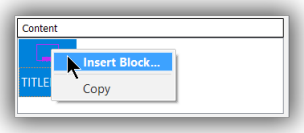


***Left:** Design Resources palette showing blocks stored in drawings; **right:** palette minimized (collapsed)*

All these resources are stored in drawing files, and this palette access them without you having to open the *.dwg* files. As well, the drawing currently open in the program, or on another computer networked to yours can be accessed.

- To locate a drawing file, look for it in the Folders pane of the palette
- Alternatively, you can access folders associated with a cloud-based storage service, such as Dropbox and Box through the **Folders** pane
- To access files on networked computers, go to **My Network Places** (on Windows) or **Net on Mac** in the Folders pane

Once you locate the drawing file, click its node — such as a block — and then look at the resources it has.



Inserting blocks from Drawing Resources palette

For all elements, you can insert the item or else copy it to the Clipboard. For instance, in the case of blocks, you choose from the following options:

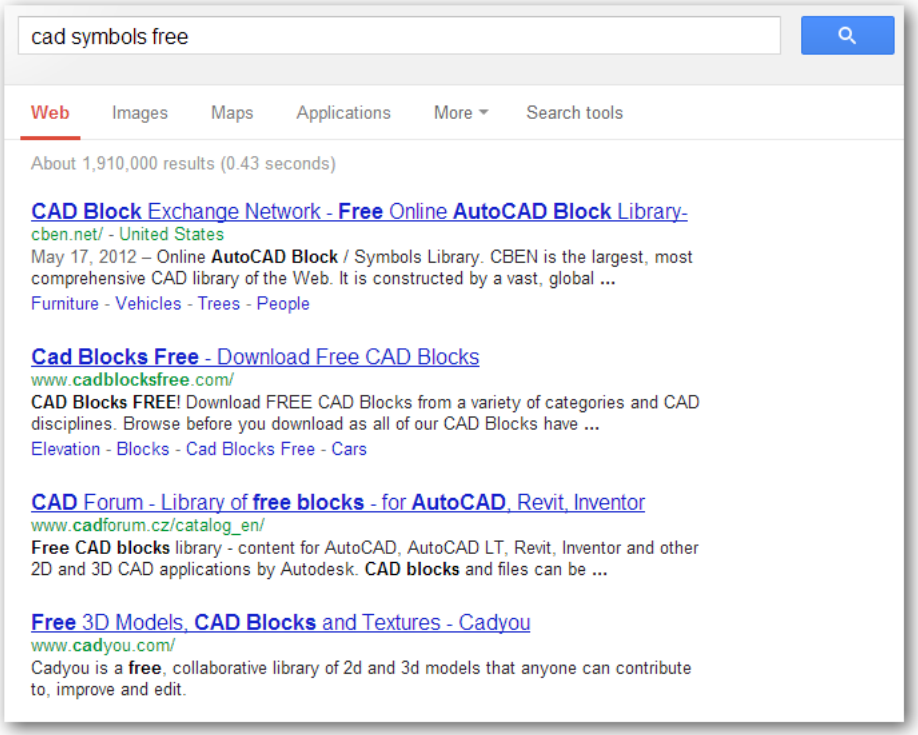
Insert Block shows the Insert Block dialog box

Copy copies the block to the Clipboard, after which you can paste into the drawing or other documents with the **Paste** command; prompts are display at the command line

Alternatively, you can ignore the shortcut menu and drag the block directly into the drawing; prompts are display at the command line

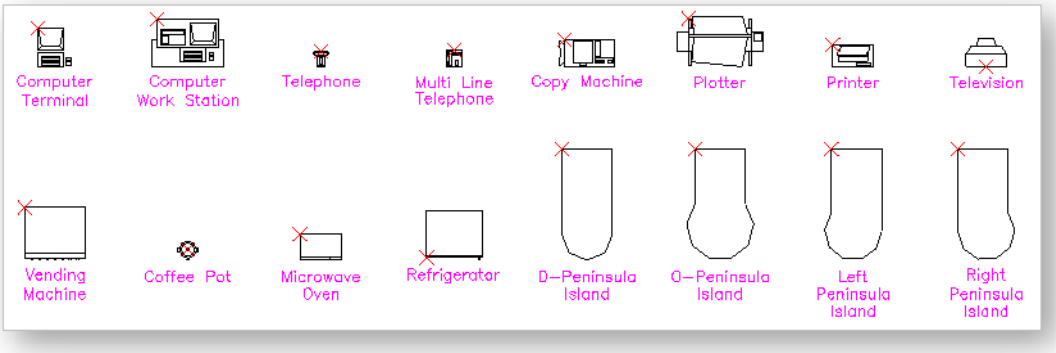
SOURCE OF BLOCKS FROM WEB SITES

You can find CAD symbols on the Internet by searching Bing or Google with terms like “CAD symbols free” and “DXF blocks free.”



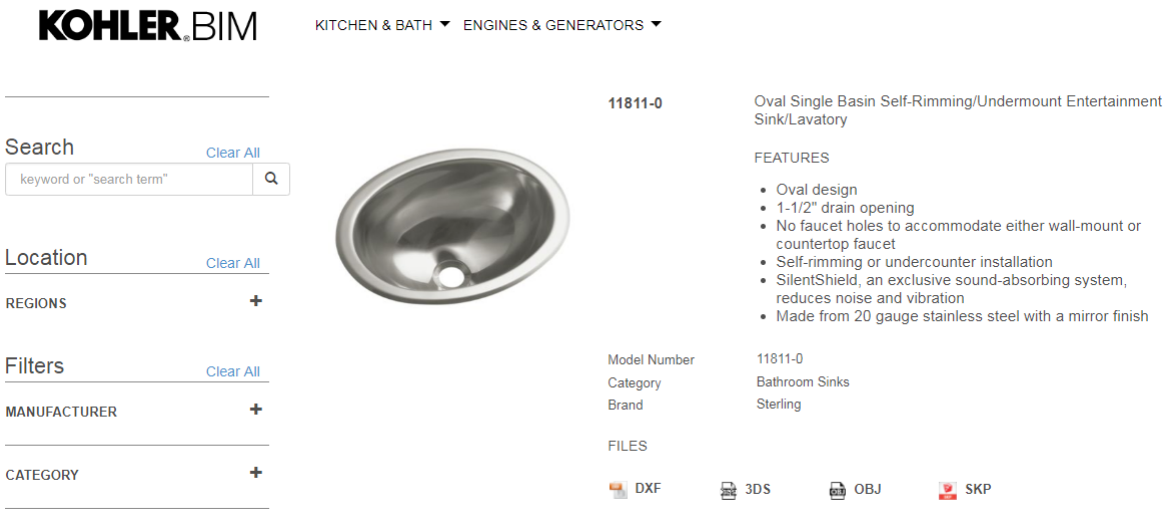
Internet search engine listing sites with blocks

CAD Corner Canada offers DWG block libraries for architects and mechanical engineers at http://cad-corner.com/. Groups of blocks are stored in *.dwg* files, such as the one illustrated below.



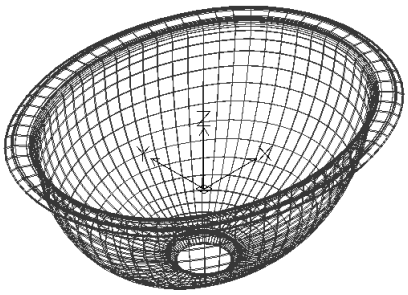
Blocks offered by CAD Corner Canada

Kohler specializes in residential and commercial fixtures for bathrooms, kitchens, and engines. It has over 3,000 2D and 3D symbols in several formats, including DXF format. They are freely available from http://www.us.kohler.com/tech/cadsymbol/cadsymbol.jsp.



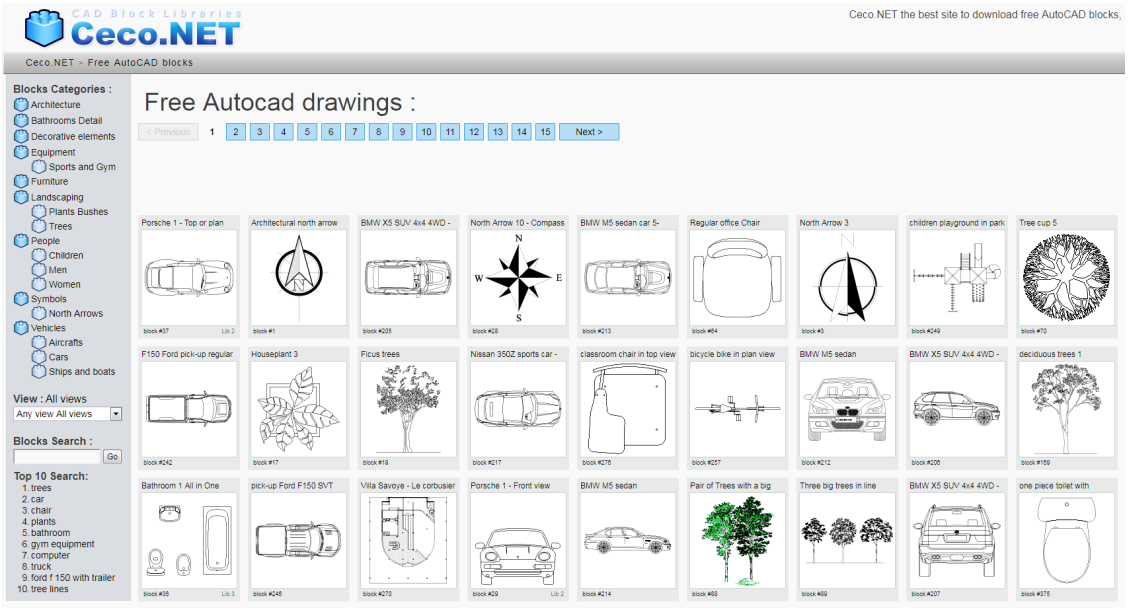
A block offered by Kohler

Here is the DXF file of the basin (shown above) in ARES Commander:



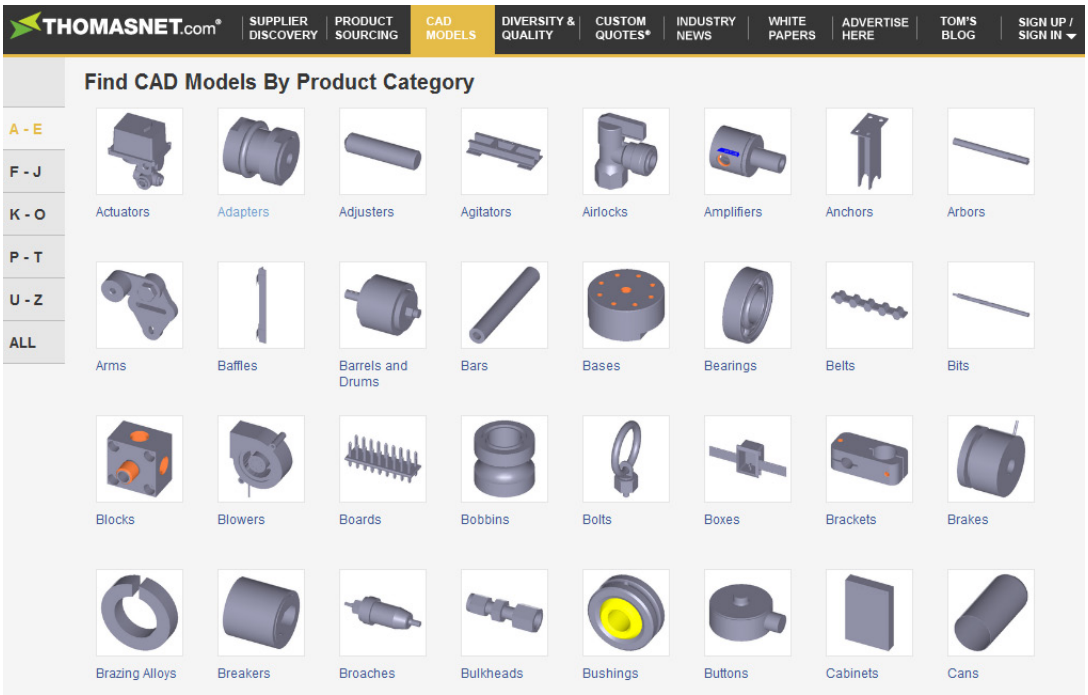
"Cut-out templates" are instructions to contractors how to cut out countertops to fit the sinks and basins.

Ceco.net has thousand of symbols for sale, as well as a selection of free blocks at http://www.ceco.net. Click on **Free AutoCAD Blocks**.



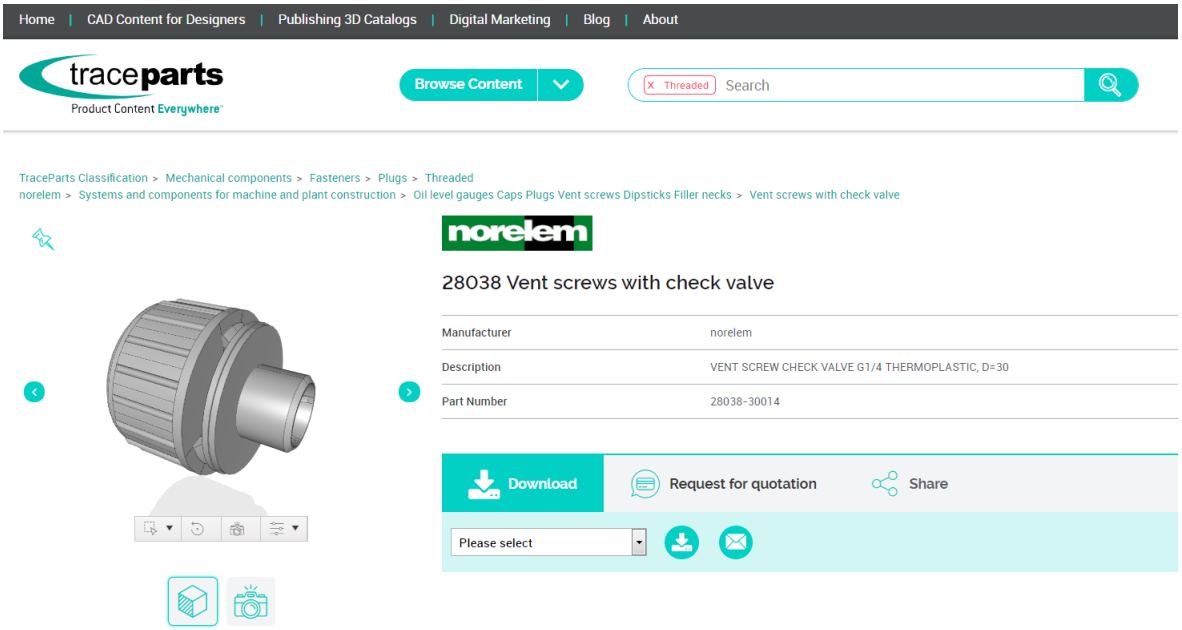
Blocks offered by Ceco.net

Thomas Register has millions of CAD drawings from hundreds of manufacturers at http://cad.thomasnet.com/cadmodels.html. The symbols are specific to brand name manufacturing products, such as Daikin air conditioning units, Tsubaki chains, and Parker valves. The manufacturers provides the drawings free in the hope that you will specify their products. You must register with the site to access symbols.



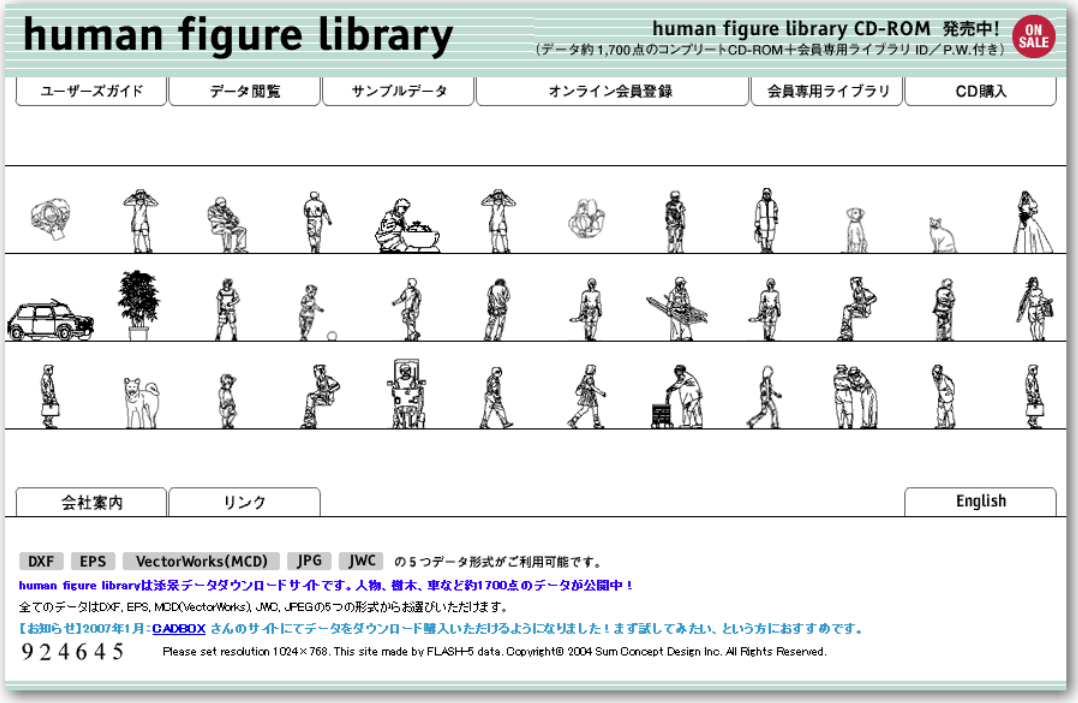
Blocks offered by Thomas.net

Traceparts is like Thomas Register, also offering has millions of CAD drawings from manufacturers. Register for free at https://www.tracepartsonline.net.



3D models from Traceparts

Human Figure Library at http://www.sum-cd.com/hfl/ has DXF files of people, cars, trees, and so on. A selection of 36 people in multiple poses are available free. The text at the Web site is mostly Japanese, but there is enough English for others to navigate it.



Blocks offered by Sum-CD

DOWNLOADING AND INSERTING BLOCKS

The process of accessing blocks from the Internet typically involves these steps:

i. Find a Web site with the type of blocks your drawing needs.

ii. Navigate the site to get to the block's download page. Sites normally segregate blocks into categories, and you may need to select several options to arrive at the desired block.

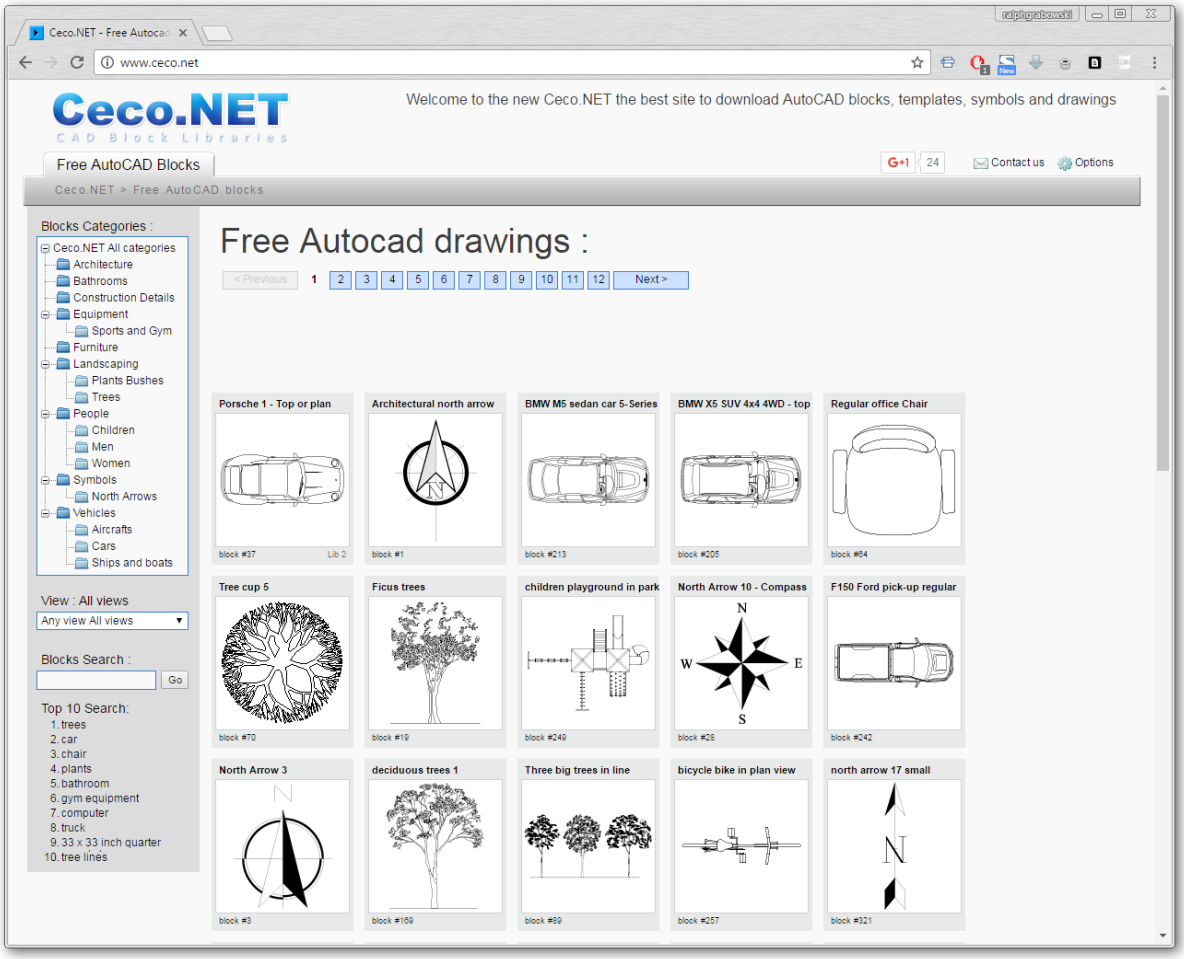
iii. Determine the variation to download. Sites often offer blocks in a variety of file formats (DWG, DXF, 3DS, and so on) and styles, (2D front view, 2D top view, or 3D).

iv. Download, and then insert them into your drawing.

Downloading Blocks

In this tutorial, you go to a Web site that offers free blocks, and then download one of them.

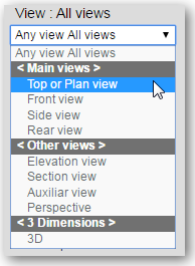
1. Open your computer's Web browser, and then enter http://www.ceco.net.
2. If necessary, click the **Free AutoCAD Blocks** tab. Notice the "Free AutoCAD Drawings" page:



Accessing free blocks from Ceco.net

3. Since the office plan drawing is a floor plan that shows the top view, you want blocks that show the top view only. This site provides a way to narrow down your selection:

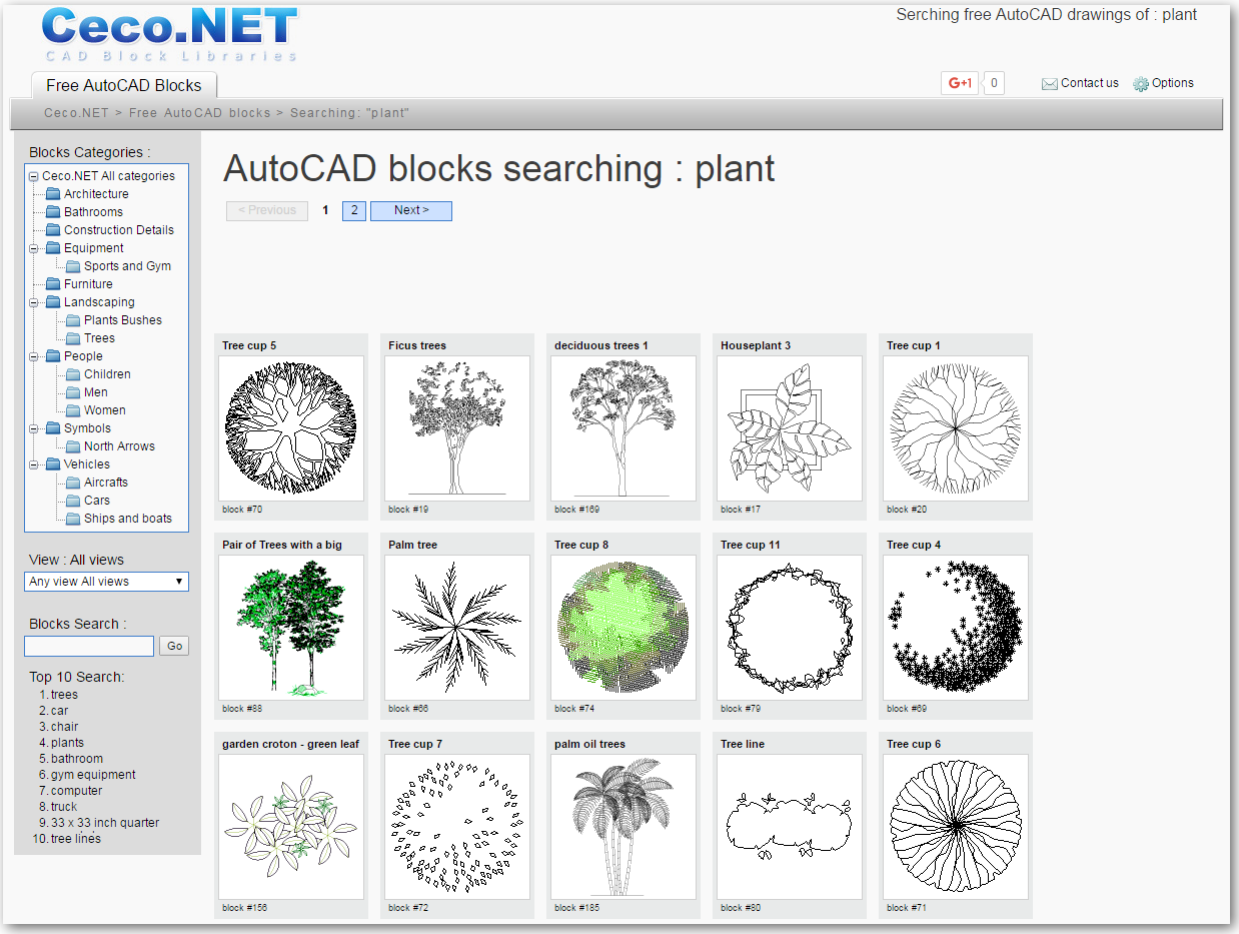
   a. In the **View: All Views** droplist, choose "Top or Plan View."



Choosing the view

   b. Click **Go**. (Other Web sites may operate differently from the one used for this tutorial.)

4. Refine the search further by specifying the type of block. In **Block Search**, enter "plant," and then click **Go**. Notice that just the top views of plants are shown.



Narrowing the search to plant blocks

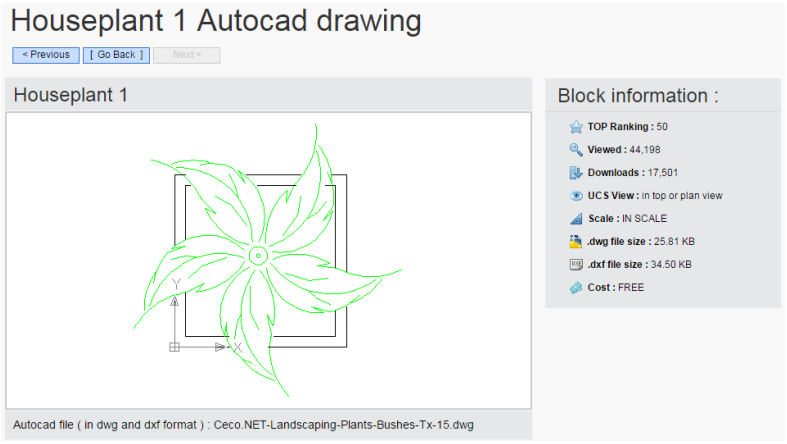
5. Click **Houseplant 3**. Notice the new window. It lists options for downloading this block:

   » DWG — for most CAD systems
   » DXF — for CAD systems that cannot read .dwg files

The window provides you with additional information, such as these useful tidbits:

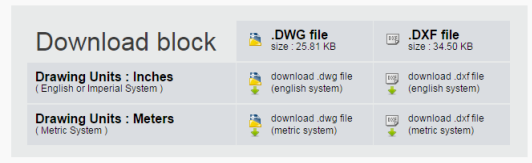
» **File size** — 25-35KB, which for this kind of block is small; file size becomes a consideration only for complex 3D blocks, where downloading could take a long time on slow Internet connections

» **Scale** — In-scale means its will match the scale of drawings in which you place it



Options to downloading this plant block in DWG or DXF format

6. Scroll down the Web page to find the download link, and then click **download .dwg file (metric system)**.



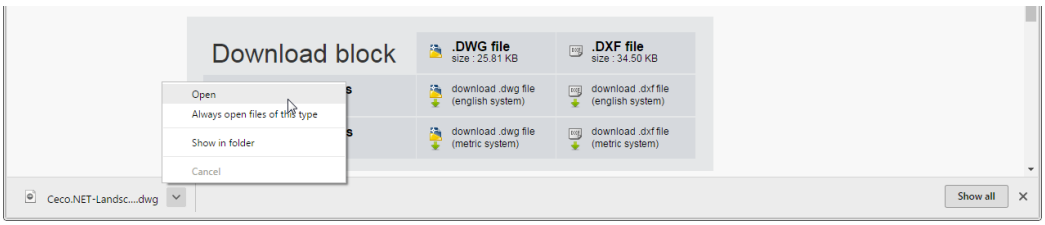
7. At this point in the tutorial, what happens next depends on your browser.

   Some display a dialog box. If so, click **Open** to instruct the browser to downloads the file and then open it in the program.

TIP If you find that the drawing does not open in the program automatically, then return to the Web page, click **Save** in the dialog box, and then choose the folder in which to place the DWG file. Locate the file with the program, and then open it.

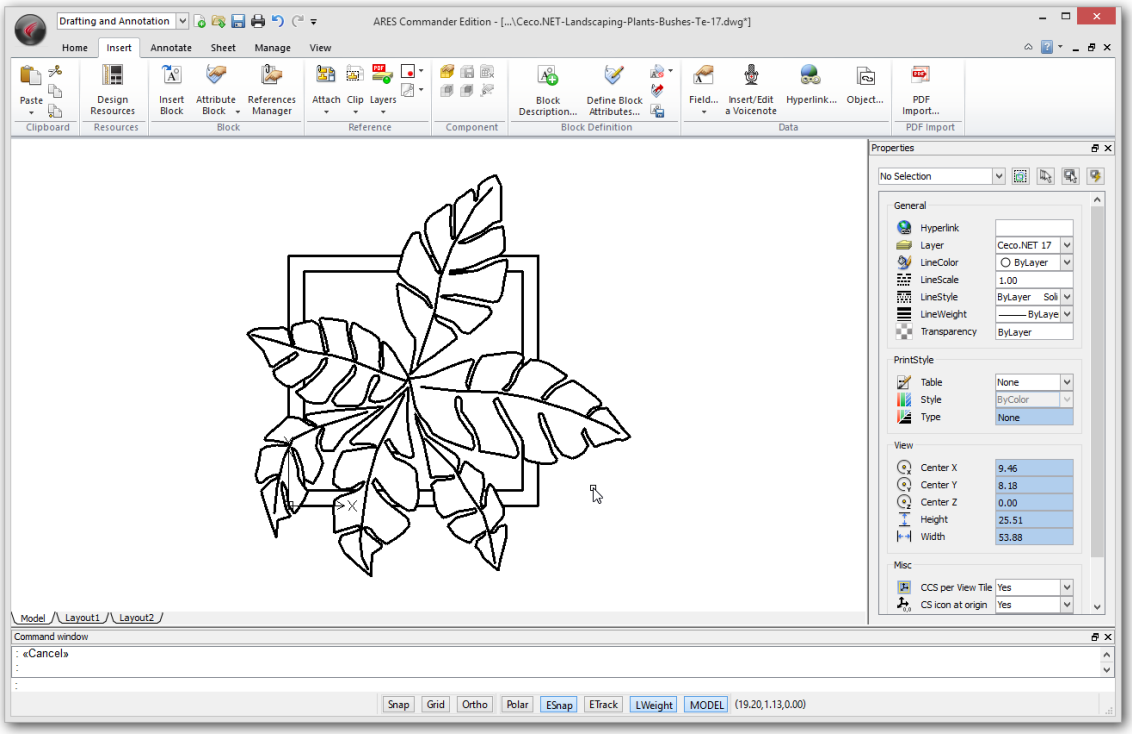
Google Chrome only: If the file does not open in the program, click Show in Folder, and then drag the DWG file into the program.

The Google Chrome browser used in this tutorial is different: it immediately downloads the file. Click the button that represents the downloaded file, and then choose **Open**.



Opening the downloaded block from Chrome Web browser

8. Opening the downloaded drawing in the program lets you see that you have the block.



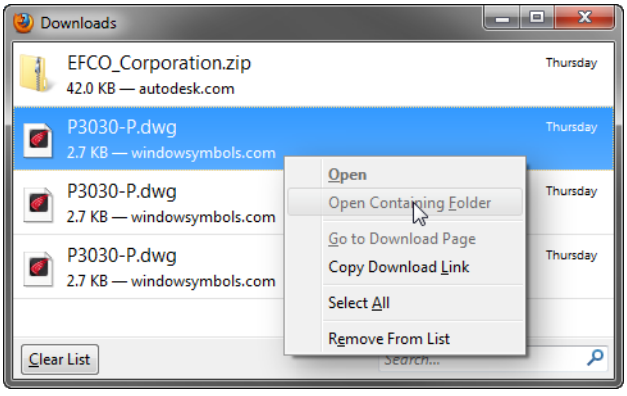
Plant block from Web site inserted in a drawing

TIP Some blocks are downloaded in ZIP files. This is done to make the file size smaller. ZIP files cannot be opened in the program. Instead, you must first use Windows Explorer (or a program like 7-Zip) to drag the DWG file out of the ZIP folder and into the folder that holds your block collection.

Inserting Downloaded Blocks

Sometimes, the most difficult aspect of downloading blocks can be determining the folder in which the Web browser stored them. Most browsers have a command that opens Windows Explorer with the download folder visible:

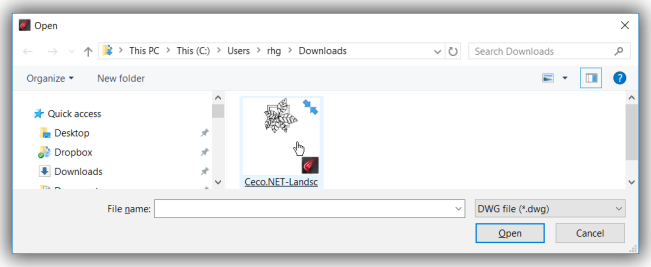
- **Google Chrome** — click the download bar, and then choose **Show in Folder**.
- **Opera** — right-click the file in the Downloads window, and then choose **Open Folder**.
- **Mozilla Firefox** — right-click the Downloads window, and then choose **Open Containing Folder,** as illustrated below.



Accessing downloaded files in Firefox

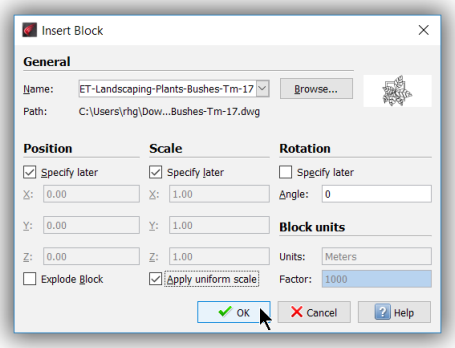
In this tutorial, you locate the downloaded block, and then insert it into the office plan drawing.

1. Switch to the office plan drawing. (In Windows or Linux, press **Ctrl+Tab** in the program until you see it.)
2. Enter the **InsertBlock** command.
3. Click **Browse**, and then in the Open dialog box, choose the downloaded file.



Selecting the downloaded .dwg file

4. Click **OK**. Notice that the Insert Block dialog box returns.
5. In the dialog box, make the following changes:
   - » Set **Position** to "Specify Later." This will let you position the plant anywhere in the office.
   - » Set **Scale** to "Specify Later." This will lets you size the plant to suit.
   - » Turn on the **Apply Uniform Scale** option. This will ensure the block maintains its aspect ratio correctly.



Insert Block dialog box

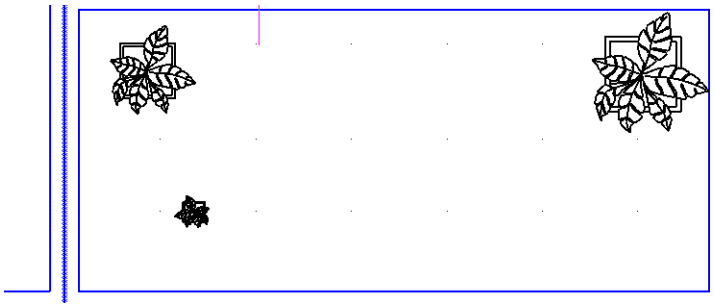
6. Click **OK**. Notice the prompts in the command window:
   a. At the 'Specify destination' prompt, pick a point in the drawing to place the plant.

      ```
      Options: Angle, reference Point, uniform Scale or
      Specify destination» (Pick a point in the drawing)
      ```

   b. At the 'Specify scale' prompt, drag the cursor. Notice that the plant becomes larger. When its size suits you, click. This positions the plant block.

      ```
      Default: 1
      Specify scale» (Drag the cursor, and then click)
      ```

7. Repeat the command (by pressing the spacebar) to place several more plants, each in a new location and at a different size (scale factor).



Placing plant blocks of varying sizes

8. Save the drawing.

TIPS FOR PLACING BLOCKS

Here are some more tips for placing blocks:

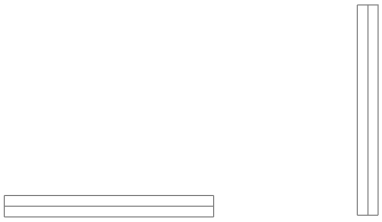
» You might come across two similar looking terms, *block definition* and *block reference*. This is the difference between them:

When a symbol is defined as a block by the **MakeBlock** command, it becomes a "block definition." Block *definitions* are invisible.

When the block definition is inserted with the **InsertBlock** command, the block you see into a drawing is known as the "block reference." Block *references* are visible.

» The window block is a unit block; when inserting it, use these scale factors and rotation angles:

Orientation	Rotation	X Scale Factor	Y Scale Factor
Horizontal	0 degrees	Window width, like 1000mm	Wall width (100 or 150mm)
Vertical	90 degrees	Wall width of 100 or 150mm	Window width (1000mm)



***Left:** Horizontal window at 0 degrees, 1000 x 100mm; **right:** Vertical window at 90 degrees, 100 x 1000mm'*

» Doors should swing out from the inside. This is sometimes a fire code requirement, so that it is easier for people to flee a damaged building.

» There should be windows in all of the walls of the office, especially in the executive office and the conference room.

» There should be no windows in the storage room, for reasons of security.

» Place desks in the executive office, the reception area, and elsewhere.

» Add at least one chair per desk.

» To place the same block many times, use the Copy or Pattern commands. The Copy command is better for placing chairs and windows all over the drawing; the Pattern command is useful for placing chairs in a uniformly-spaced row.

» Use the InsertBlock command's **Rotate** option to place blocks at 90-degree increments.

Extracting Blocks From Drawings

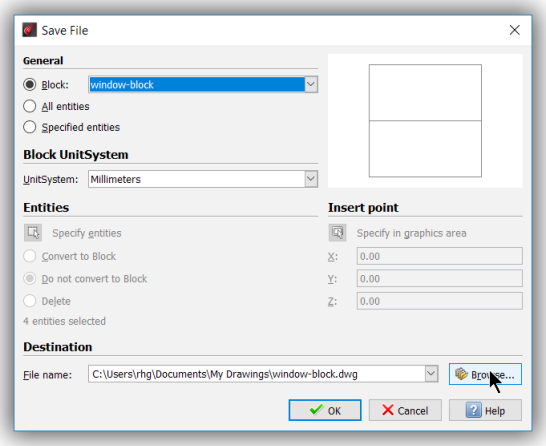
Command: ExportDrawing
Aliases: wblock, w, dwgout, exportdwg
Menu bar: File | Export| Export DWG
Ribbon: Application | Export | Export Drawing (Drafting and Annotation)
Toolbar:

When you create blocks in a drawing like *office plan.dwg*, you might come across this problem: how to use the blocks in another drawing? You cannot use the InsertBlock command, because it would insert the entire office floor plan as a block. The solution is to extract the blocks into their own DWG files with the ExportDrawing command. This flexible command lets you extract individual entities, blocks, or the entire drawing (**All Entities** option).

EXPORTING BLOCKS

In this tutorial, you extract the window-block block.

1. Enter the **ExportDrawing** command using one of the methods listed above.
2. Notice the Save File dialog box. Enter the following options to export the window block:
   a. In the Source section, choose **Block**.
   b. From the droplist, choose **window-block**.

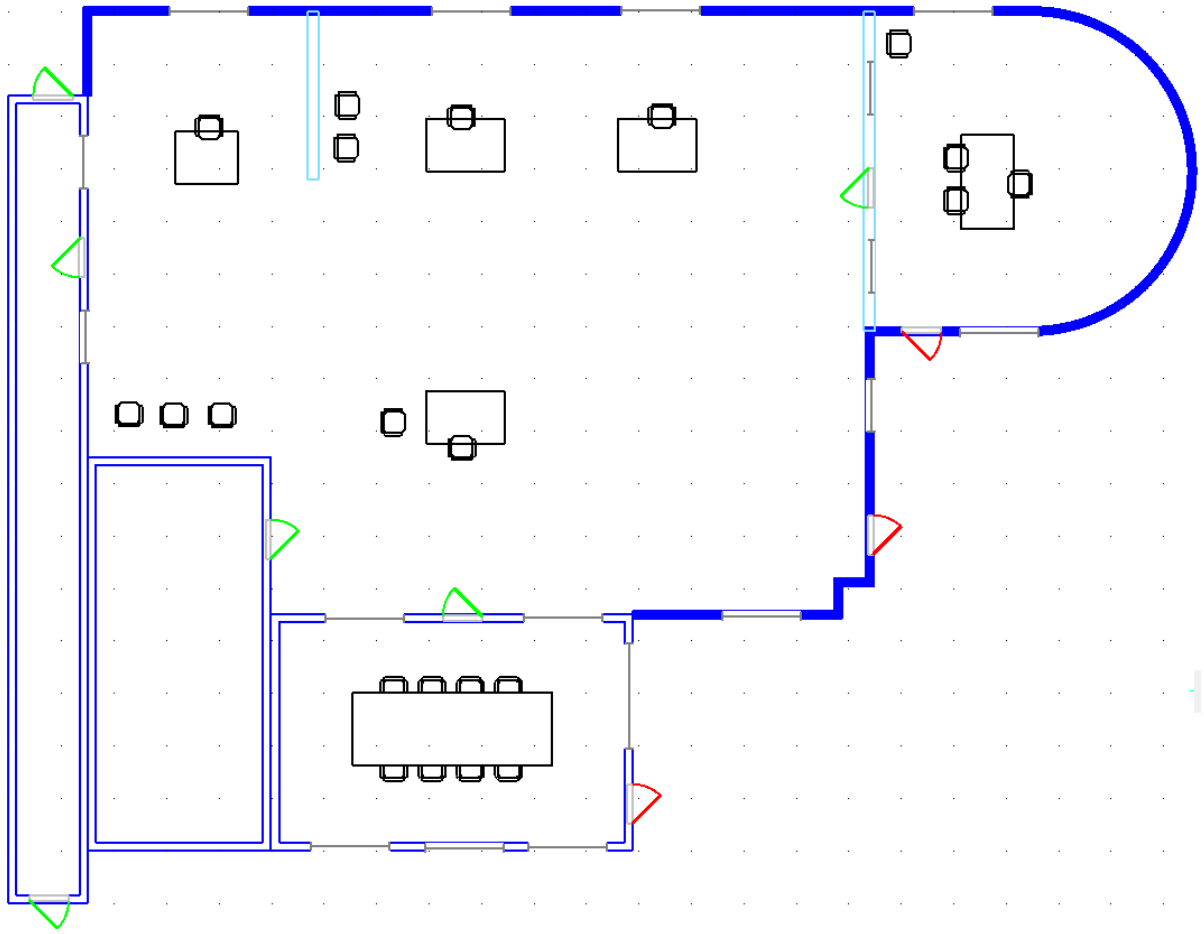


Saving blocks as .dwg files

3. The name of the exported drawing file does not need to match the name of the block. In fact, the program gives the exported block a generic name, such as "NewBlock.Dwg." To change the name and to specify the folder in which to store the exported block, follow these steps:
   a. In the Destination section, click **Browse**. Notice the Save Entities As dialog box.
   b. If you wish, change the file name from "NewBlock.Dwg" to **Window-Block-Dwg**.
   c. Choose the folder in which to store the exported block.
   d. Click **Save** to return to the Save File dialog box.
4. Click **OK**.

The program appears to do nothing, but when you check the folder, you should find the exported block under the name of *Window-Block.Dwg*.

You can now use the InsertBlock command's **Browse** option to select and insert the file into other drawings.



Summary

The use of blocks is a powerful way to increase your productivity by reducing the amount of drafting needed to complete a drawing. In the next chapter, you learn about a more powerful version of symbol, called the custom block.

CHAPTER 08

Custom Blocks from Scratch

In this Chapter

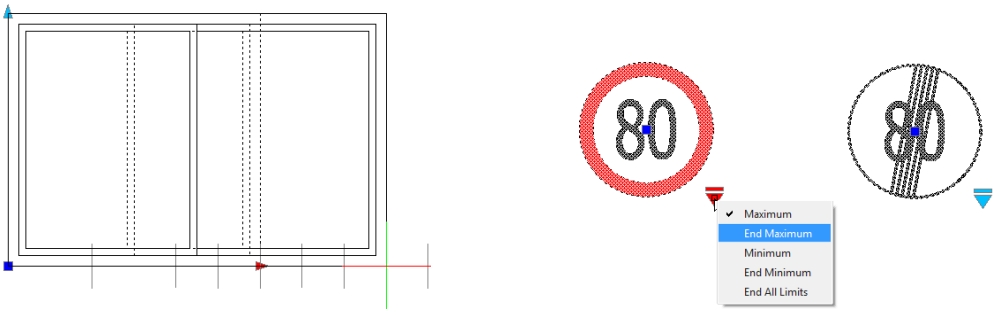
- How custom blocks differ from regular blocks
- Applying visibility states to custom blocks
- Understanding elements and activities
- Adding element sets to custom blocks

Most of the time, you use blocks that are *static*. You specify a few parameters when you insert them, such as their scale factor and angle, but once in the drawing they stay looking the way they were when you inserted them.

A different kind of block is called a *custom* block, because it can change its looks after it is inserted. The looks are "programmed" into it ahead of time and are quite specific, such as changing its size or its look.

To learn about creating custom blocks through videos, see the tutorial series at Graebert's blog site:
https://www.graebert.com/blog/tutorial/dwg-block-editor-create-a-custom-block-with-resizable-shape/

Here are examples of some custom blocks. One changes its size by specific increments (at left), while the other changes its look (at right).



Left: *Window as a custom block changing its width*
Right: *European road sign as a custom block changing its face*

KEY TERMS IN THIS CHAPTER

Activities connect elements to entities.

Block Editor is the environment in which custom blocks are created and regular blocks are edited.

Custom Block contains constraints that restrict size and movement, as well as visibility toggles that allow one definition to display multiple blocks.

Elements define actions, such as move, rotate, and array.

Element Sets combine elements with activities.

Visibility States determine whether portions of a custom block are visible, or hidden.

COMMANDS

In addition to all of the program's regular drawing and editing commands, you can use the following ones with custom blocks:

Command	Shortcuts	Ribbon Selection
CbActivityTool	...	Block Editor | Activities
CbClose	...	Block Editor | Close | Close
CbElement	...	Block Editor | Elements
CbElementSet	...	Block Editor | Element Sets
CbHide	...	Block Editor | Display States | Hide
CbOptions	...	Block Editor | Manage | Options
CbSave	...	Block Editor | Save | Save
CbSaveAs	...	Block Editor | Save | Save As
CbShow	...	Block Editor | Display States | Show
CbVisibility	...	Block Editor | Display States | Visibility States
EditBlock	bedit, be, editcustomblock, eb	...

WHAT'S NEW IN 2019 FOR THIS CHAPTER

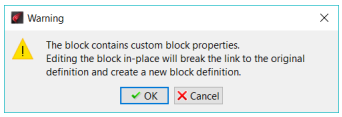
- **Custom Blocks** are created in the new Block Editor environment.
- Numerous new commands for creating custom blocks.
- Settings added to the Options dialog box that affect custom blocks.

Custom blocks are adjusted using grips. The grips are cyan in color to distinguish them from regular grips, which are dark blue. The shape of the custom grip tells you its function; see “About Custom Grips” boxed text.

The program creates custom blocks inside a Block Editor environment. You can create your own, or else source interactive blocks from online sites, such as www.cadforum.cz/catalog_en/?q=dynamic.

TIP Dynamic blocks, such as those found at the CadForum portal, were probably created in AutoCAD, and will operate correctly in ARES Commander: you can adjust the size, visibility, and parameters. The definition of these dynamic blocks cannot, however, be modified with the Block Editor in ARES Commander.

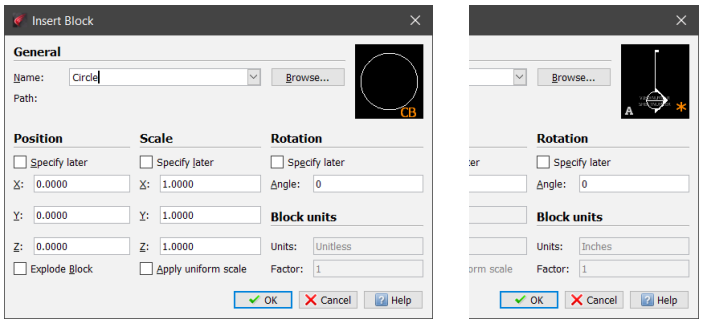
When you want to open a non-ARES interactive block in Block Editor, you are warned:



Clicking **OK** converts the dynamic block into a simple block, which can then be edited by Block Editor.

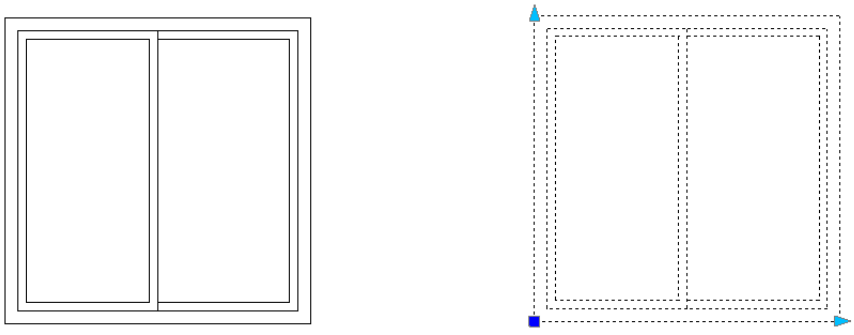
Custom blocks and dynamic are inserted with the **InsertBlock** command, just like static blocks. The only difference is that the preview window shows a symbol:

- **‘CP’ logo** — interactive block was created by ARES Commander; short for “custom block”
- **‘*’ logo** (asterisk) — interactive block not created in ARES Commander
- **‘A’ logo** — annotative block; contains two or more scale factors



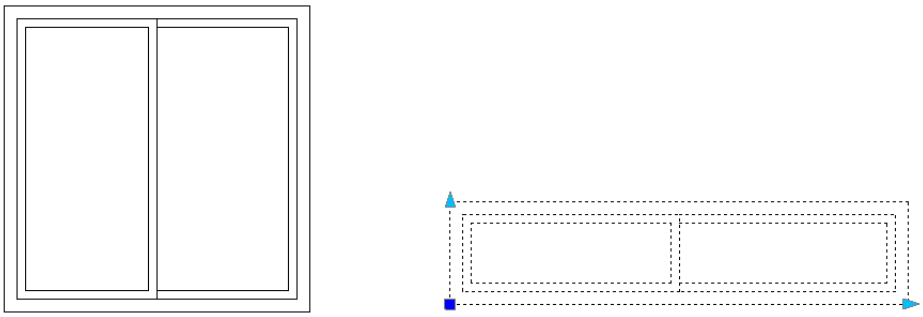
Insert dialog box’s preview area reporting the source of interactive blocks

When a custom block is inserted, it looks no different from any other block. Until, that is, you select it: then the cyan-colored custom grips come into view, letting you see the aspects that you can manipulate.



Left: *Custom block inserted into drawing looking normal*
Right: *Selected custom block showing its parameters through the cyan-colored grips*

Referring to the table on the previous page, you can see that the cyan arrow grips resize the window, vertically and horizontally — independently. (The dark blue grip indicates the block's insertion point, and is used to move the window.) In the figure below, I resized the window, making it shorter and wider than the original.



Left: *Custom block as originally inserted*
Right: *Block interactively changed to be shorter and wider*

The advantage of custom blocks is that one block can represent many varieties, such as stoves with options, a large range of bolt lengths or diameters of pipes, or a variety of flowers and cars.

ABOUT CUSTOM BLOCK COMMANDS

(NEW IN 2019) All 2D drawing, editing, and view commands work, along with the following commands specific to the editor:

CBActivity command adds activities to elements, such as adding Move or Stretch activities to the Point element.

CBActivityIcon command toggles visibility of activity icons between show, hide, and reset (returns visibility to the default state).

CBActivityTool command applies activities to elements, such as applying the Scale activity to an existing Linear, Polar, or XY element.

CBClose command asks if you want to save the changes, and then exits the Block Editor back to the drawing editor.

CBElement command adds elements to custom blocks.

CBElementSet command adds predefined element sets to custom blocks, which consist of a noun (the element) and a verb (the activity), such as Point Move and Polar Pattern.

CBHide command sets the visibility state to hidden.

CBOptions command displays the Options dialog box at the Block Editor section.

CBSave command saves the custom block definition.

CBSaveAs command saves the custom block definition by another name.

CBShow command sets the visibility state to visible.

CBValueTable command opens the Property Value Table dialog box to add, edit, and delete properties and values:

CBVisibility and **-CBVisibility** commands set the default visibility of entities that make up the custom block. For this command to work, you must add at least one visibility state to the block with the CBElementSet command.

A Tour Through the Block Editor

Command: EditBlock
Aliases: editcustomblock, bedit, be, eb
Shortcut: *double-click a block*

(NEW IN 2019) Custom blocks are created and edited in a special Block Editor environment. This is in contrast to regular blocks, which are created in the regular drawing environment of model space. The commands for making blocks interactive work only in the Block Editor.

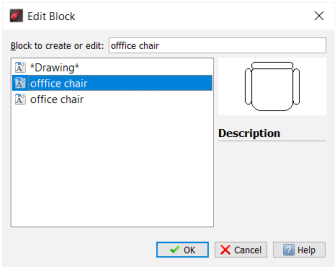
You can start the block editor with these options:

- Enter a name into the B**lock to Create or Edit** field to start a new block from scratch.
- Use an existing block that's already in the drawing, or else access one elsewhere (from the current drawing, on your computer, off the Internet) with the Insert command; we use this approach in this tutorial.

1. Start the program, and then use the **Insert** command to access the *office-chair.dwg* file that you created in the previous chapter.
2. To access the block editor, enter the **EditBlock** command using a method listed above.

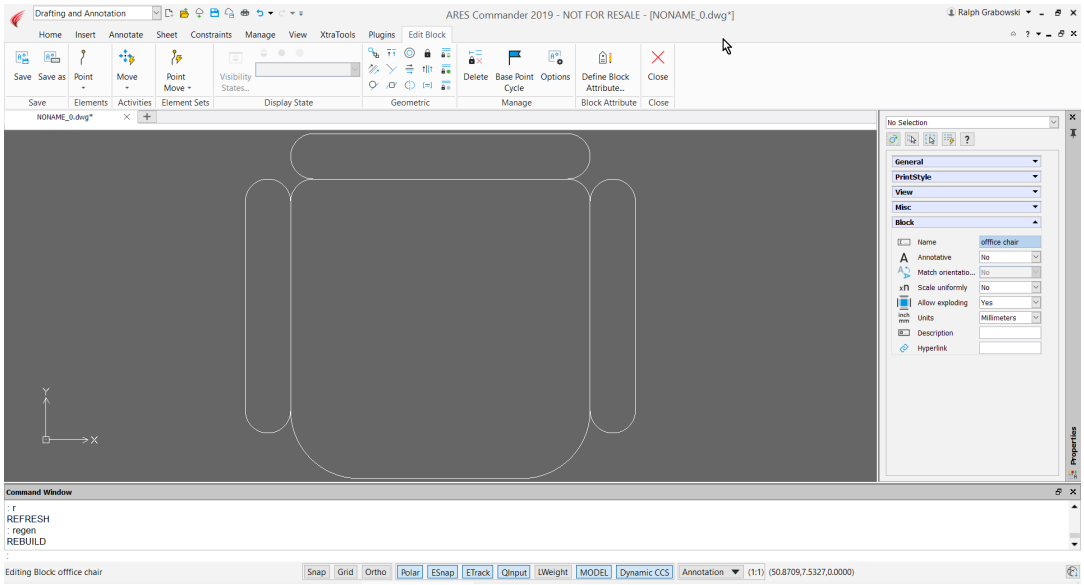
   `: editblock`

3. Notice the Edit Block dialog box. Choose "office chair" from the list, and then click **OK**.



Edit Block dialog box

4. The block opens in the Block Editor environment. Notice that the background color of the drawing area is dark gray, and that the ribbon displays commands specific to the editor. The status bar reads, "Edit Block: office-chair."

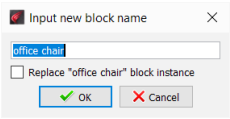


Block Editor environment

All drawing and editing commands work in this environment, such as drawing lines and moving entities around. What doesn't work in the Block Editor are commands that involve files, such as Open, WBlock, and Print, other than file-related commands that begin with Cb-, such as CbSave.

I find the ribbon to be the best interface for working with custom blocks. (The additional commands listed in the "About Custom Block Commands" boxed text operate only in the Block Editor.)

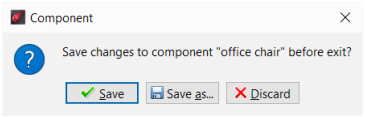
5. To save your work as you are editing the custom block, follow one of these options:
   - » For a regular save, click the **Save** button (CbSave command), and the block is saved in the drawing.
   - » To save the block by another name in the drawing, click the **SaveAs** button (CbSaveAs command). In the Input New Block Name dialog box, enter a new name, and then click **OK**. Notice that you have to option to replace existing instances (insertions) of the block with the renamed one.



Input New Block Name dialog box

   - » To save the block to disk for sharing by other CAD users, exit the Block Editor and then enter the **Export-Drawing** command; see the previous chapter for how to use this command.

7. When you are done editing the custom block, click **Close** (CbClose command), and the program asks if you want to save the changes you made. Here tou have a second chance to rename the custom block with the dialog box's Save As option.



Component dialog box

The program exits to the usual drawing editor's model space.

ABOUT CUSTOM BLOCK GRIPS

Custom blocks are adjusted using special grips that are cyan in color to distinguish them from regular, dark blue. The shape of the custom grip tells you its function:

Grip	Function
	Resize a part of the block (scale); arrow indicates direction
	Increment markers (the series of horizontal lines) indicate resize restrictions; toggle the display of them with the **BtMarkDisplay** system variable
	Move the block in any direction
	Align the block to another entity; arrow indicates the alignment edge
	Flip the block (mirror); arrow indicates the mirroring direction
	Choose an alternative representation from a shortcut menu

TIP You change the color of custom grips with the **GripDynColor** system variable.

Creating Custom Blocks, Step-by-Step

There are two things you can do to a block to make it a custom block: set up visibility states and/or apply element sets. Other operations shown by the ribbon's Block Editor tab, such as applying geometric constraints and adding attributes, are also available for regular blocks.

SETTING UP VISIBILITY STATES

Command: CbVisibility
Alias: bvstate
Menu bar: ...
Ribbon: Block Editor | Display State | Visibility States
Toolbar: Block - Visibility | Visibility States

You use visibility states to show and hide portions of a block. This lets you create custom blocks that perform the following functions in drawings:

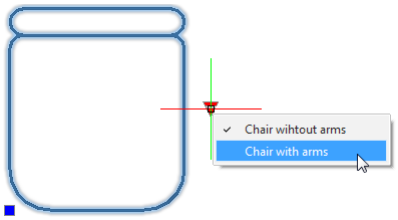
- Make a single custom block with two or more visibility states, such as a chair with arms, without arms, or even hide the chair entirely.
- Make a complex custom block that combines several symbols that are shown one at a time, such as a chair, a desk, a computer, and so on.

There are two visibility states: on and off, although it could be argued there are three: all entities turned on, some turned on, and all turned off. To accomplish this, you use visibility states in the Block Editor environment. To set up visibility states take these steps:

i. Add the Visibility element to the block. The Visibility element tells the program that the custom block is capable of this action, and assigns the custom grip.

i. Determine which entities should be hidden.

ii Define two (or more) visibility states based on entities that are hidden or visible. The states become a droplist of visibility options attached to the custom grip. See figure below.

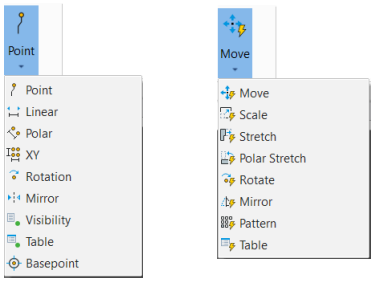


The Visibility element creates this custom grip and its droplist of visibility options

The following tutorial builds a single custom block that represents two types of office chair, one with arms and one without.

1. Open the *office-chair.dwg* file in the program.
2. Double-click the block, and then click **OK** in the Edit Block dialog box. The block appears in the Block Editor environment.
3. Add the visibility action to the custom block like this;
   a. In the ribbon's Block Editor tab, go to the Element Sets pane, and then choose **Visibility Set**. This executes the CbElementSet command with the Visibility option, and so you can ignore all of the chatter in the command bar, except for the last prompt:

QUICK SUMMARY OF ELEMENTS & ACTIVITIES



Buttons on the Block edit ribbon: ***left****: Elements* ***right****: Activities*

ELEMENTS

Elements are applied to custom blocks with the **CbElement** command.

Editing Elements

Point specifies the x,y coordinates from which to move or stretch the custom block.

Linear specifies the distance by which to move, scale, stretch, or array the custom block — uni- or bidirectionally.

Polar specifies the base point, distance, and angle by which to move, scale, stretch, or polar-array the custom block — uniformly or incrementally.

XY specifies the base point, and horizontal and vertical distances by which to move, scale, stretch, or array the custom block — uniformly or incrementally.

Rotation specifies the base point, radius, and angle by which to rotate the custom block — uniformly or incrementally.

Mirror specifies the mirror line about which to flip the custom block; formerly named "Flip."

Utility Elements

Basepoint specifies the x,y coordinates of the base point.

Visibility specifies the parts of the block which are visible and invisible.

Table specifies a lookup table that defines the look, size, and shape of the block.

ACTIVITIES

Activities are linked to elements and applied to entities in custom blocks with the **CbActivityTool** command.

Editing Activities

Move moves entities in linear or polar directions.

Scale resizes entities in linear, polar, or x,y (array) directions.

Stretch stretches entities in and (point), linear, polar, or x,y (array) directions.

Polar Stretch stretches entities by distances or angles.

Rotate rotates entities.

Mirror flips (mirrors) entities about a mirror line.

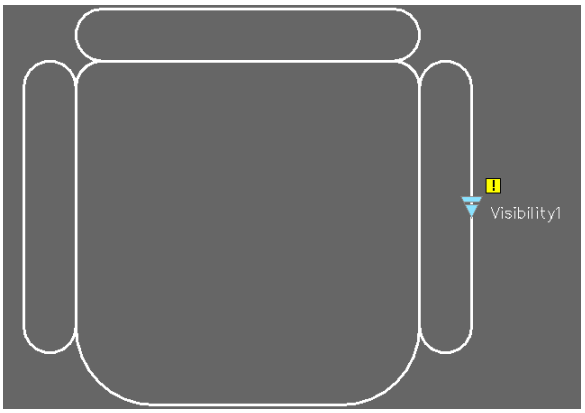
Pattern arrays entities rectangularly.

Table applies actions, user-defined parameters, constraint parameters, parameters, and attributes to objects.

```
: _CBELEMENTSET
Options: Base, pOint, Linear, Polar, Xy, Rotation, mIrror, Visibility or taBle
Specify element type» _Visibility
Options: Name, Label, Description or Palette
```

b. At the prompt, pick a suitable location, such as next to an arm.

```
Specify element position» (Pick a suitable location)
```

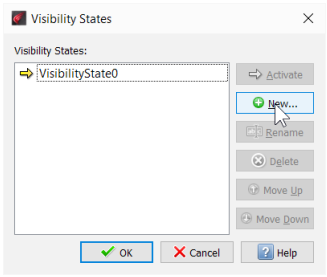


Placing the visibility element

The yellow exclamation mark means that the visibility state needs some data to work with. Let's add it by specifying which entities to show and which to hide.

3. You'll be creating two visibility states, one that shows the arms and one that does not. The first visibility state is the default arrangement, the chair with arms.

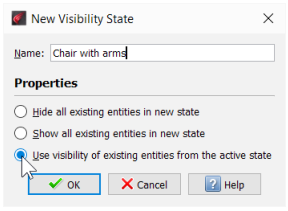
   a. To define the visibility states, click **Visibility States** in the Display State panel. Notice that the Visibility States dialog box opens with one visibility state already set up. Ignore it, because "VisibiltyState0" is only a placeholder. (In fact, you erase it later.)



Visibility States dialog box

TIP Ignoring "VisibiltyStateo" is crucial. If you try to use it, you will become hopelessly confused.

   b. To create a new visibility state, click **New**. Notice the New Visibility State dialog box.



New Visibility State dialog box

c Replace the default "VisibiltyState1" name with a more descriptive one, such as "Chair with arms." In fact, the name should be very descriptive, because this is the text that appears in the droplist of visibility options, later on in the drawing editor.

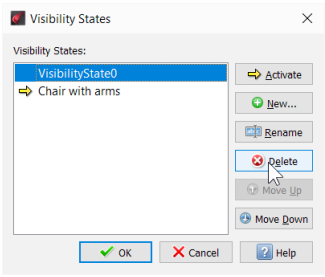
d. Under Properties, click the radio button next to **Use visibility of existing entities from the active state**. This means "Chair with arms" displays exactly what you see right now in the block editor.

e. Click **OK** to close the dialog box.

4. You can now erase the unneeded visibility state:

a. Select "VisibiltyState0".

b. Click **Delete**.



Removing an unneeded visibility state

5. Now set up the other visibility state, the one with the arms hidden.

DYNAMIC BLOCKS vs. CUSTOM BLOCKS

AutoCAD creates and uses *dynamic blocks*. ARES Command creates interactive blocks that operate similarly, which it calls "custom blocks."

USING AUTOCAD'S DYNAMIC BLOCKS IN ARES

ARES Commander reads and uses AutoCAD's dynamic blocks just like AutoCAD does. When someone shares with you dynamic blocks created in AutoCAD, they can be used inside ARES Commander with the same set of interactivity functions, including modifying their shape, size, and configuration.

After making interactivity changes to them and saving the drawing with ARES Commander, the blocks are still seen as dynamic blocks back in AutoCAD.

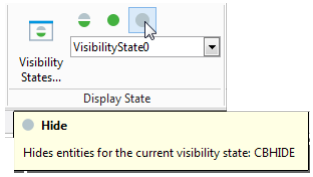
When, however, you want to edit an AutoCAD dynamic block with ARES' Block Editor, you will be required to redefine grips and actions from scratch.

USING ARES CUSTOM BLOCKS IN OTHER CAD PROGRAMS

Custom blocks created with ARES Commander can be read and displayed by other CAD programs, such as AutoCAD, but only as simple blocks. When one of ARES' custom blocks is modified in the other program, it loses the interactivity that was specified with ARES. Fortunately, you may modify all the rest of the drawing with the other program without breaking the interactivity of custom blocks, as long as you don't touch the custom blocks.

When you want to edit an ARES custom block with AutoCAD's Block Editor, you are also required to redefine grips and actions from scratch.

a. To hide the arms, click the gray circle button in the Display State panel. (This executes the **Cb-Hide** command.)

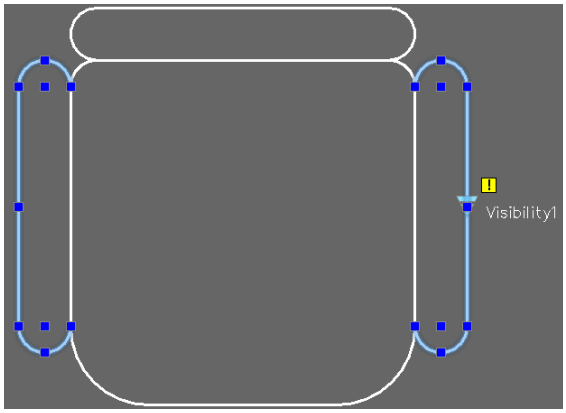


Hiding elements

```
: _CBHIDE
Default: Current
Options: Current or All
Specify affected visibility states» _Current
```

b. Select the entities that make up the two arms:

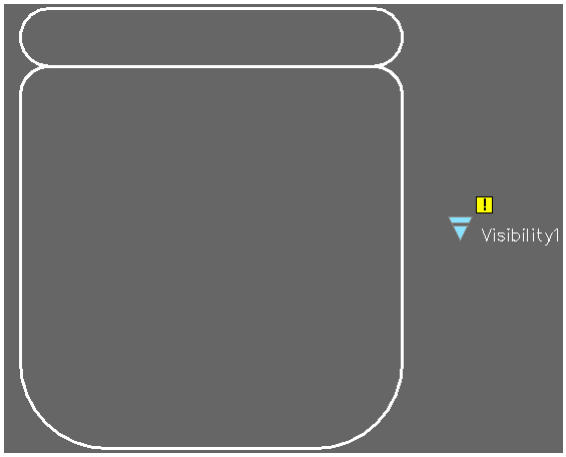
```
Specify entities to make hidden» (Pick the six arm elements)
6 found, 6 total
```



Selecting entities to hide

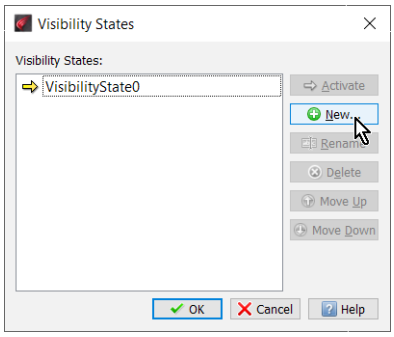
c. After you press **Enter**, the arms disappear. This represents the other visibility state.

```
Specify entities to make hidden» (Press Enter to end the command)
```



Arms are hidden

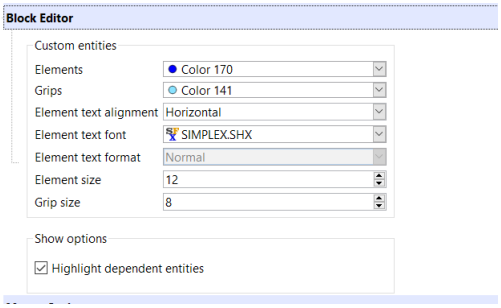
4. Go through steps similar to before:
   a. Click **Visibility States** in the Display State panel.
   b. Click **New**.



Defining a new visibility state

CUSTOM BLOCK OPTIONS

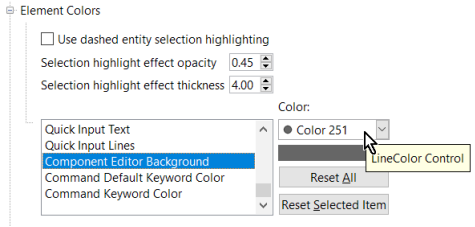
The **CbOptions** command takes you directly to the Options dialog box's section for custom blocks so that you can customize their look:



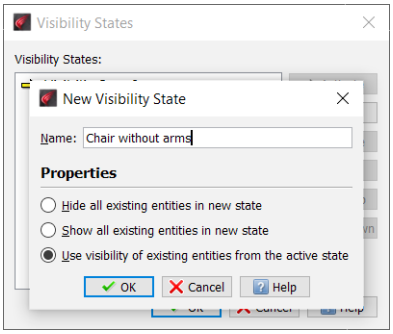
The one change I recommend is to **Grip Size**: double it to 16 pixels to make it easier to see.

BACKGROUND COLOR

To change the background color away from the default dark gray of the Block Editor environment, go in the Options dialog box to **System Options | Display | Element Colors**, and then select a lighter color for **Component Editor Background**.



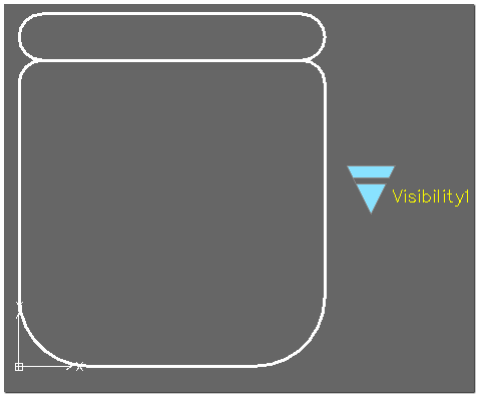
c, Replace the default "VisibiltyState2" name with a descriptive name like "Chair without arms."



Renaming the visibility state

d. Click **OK** to close the Nwq Visibilith dialog box, and then again click **OK** to close the Visibility States dialog box.

Notice that the yellow exclamation mark is gone! This means you save satisfied its requirements, which is that a visibility element must have at least two states.



Visibility element with requirements satisfied

5. You test the visibility states inside the Block Editor like this:

a. In the ribbon's Display State panel, click the droplist.

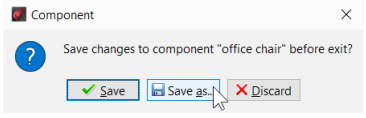
Changing visibility states inside the Block Editor

b. Choose "Chair with arms" to see the chair block with its arms.

c. Now from the droplist, choose "Chair without arms" to see the chair hide its arms.

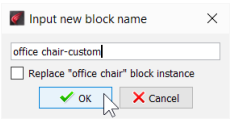
6. Now that you have checked that the custom block is working correctly, you can exit the Block Editor back to the drawing editor. As you do, rename the block to indicate it is a custom block.

a. Click **Close** in the Close pane (CbClose command).



Saving by another name

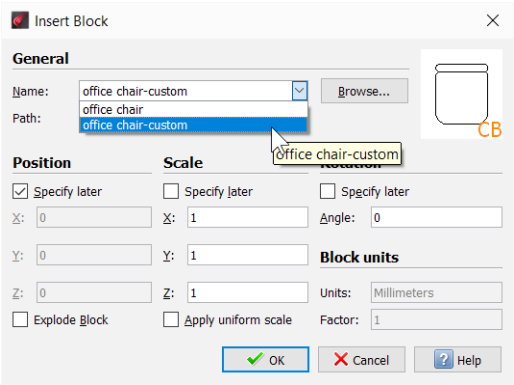
b. In the Component dialog box, click **Save As** to change the name of the block. After all, this is now a different block from the one with which you began.



The new name

c. In the Input New Block Name dialog box, change the name to "office chair-custom," and then click **OK**.

7. The renamed block is stored as a block definition in the drawing, and so don't see it. Use the Insert command to add it to the drawing:



Selecting the custom block to insert into the drawing

BLOCK EDITOR vs. COMPONENT EDITOR

ARES Commander provides two environments for editing blocks, the Block Editor and the Component Editor.

ABOUT THE BLOCK EDITOR

The Block Editor (**EditBlock** command) is designed to create and edit custom blocks, but it also edits simple, non-interactive blocks. The advantage to using the Block Editor for simple blocks is that it lets you edit the *definition* of the block, which affects all blocks inserted by that name -- instead of only the one insertion of this block.

You exit the Block Editor with the **CBClose** command.

ABOUT THE COMPONENT EDITOR

You use the Component Editor (**EditComponent** command) to edit a single *instance* of an inserted block or an xref (externally referenced drawing attached to the current drawing). This lets you make changes to the way one block looks, without affecting any other block inserted with the same name.

The Component Editor has the added benefit of working in-situ: you see the rest of the drawing, whereas the Block Editor shows only the block being worked on.

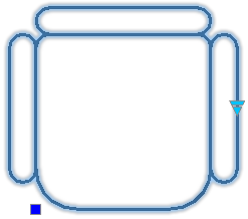
The Component Editor cannot edit custom blocks. You exit the Component Editor with the **CloseComponent** command.

a. In the Insert Block dialog box, choose "office chair-custom," from the **Name** droplist. Notice that the preview window displays a red CB in the lower corner. This alerts you that the block is a custom block.

b. Set the parameters in the dialog box, and then click **OK**.

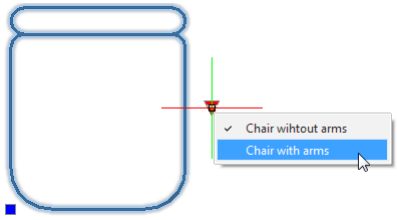
8. The block is inserted with arms, as this is the default visibility state.

a. With the custom block inserted in the drawing, select it. Notice the triangle grip, which indicates the block has visibility states.



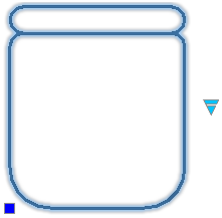
Custom block selected

b. Click the cyan-colored visibility grip. Notice that a droplist appears, showing the two visibility options:



Visibility options

c. Choose the option to display the arms:



Chair with arms

The arms appear. The custom block works correctly! You can go on to create multiple visibility states in a custom block. In summary then, you set up visibility states by following these steps:

i. **CbElementSet** command adds the visibility element to the block.
ii. **CbHide** command determines which entities should be hidden.
iii **CbVisiblity** command names and defines at least two visibility states.

Let's look now at adding another action to the office chair custom block, rotation.

SETTING UP ROTATION ELEMENTS

Command: CbElement Rotation
Ribbon: Block Editor | Elements | Rotation
Toolbar: Block - Elements | Rotation

Custom blocks can perform actions, such as move, rotate, and pattern (create an array of repeating patterns). These movements might not seem impressive, because you can edit any entity in your drawings with the related commands of Move, Rotate, and Pattern.

But here's the deal with custom blocks: you can constrain those movements. For instance, you could...

- Move a block only horizontally, making a sliding motion
- Rotate a block only in 45-degree increments
- Array part of a block, such as creating a picket fence without needing the Array command; to see how to do this, watch the video tutorial mentioned at the start of this chapter

Even better, you carry out these actions interactively, without needing to choose a command or enter options.

To attach actions to custom blocks, you go through these steps:

i. Chose an *element*, such as Rotation.
ii. Connect the element to the block's entities with an *action*, in this case, Rotate.
iii. Specify the rotation properties in the Properties panel's Value Set section.

TIP You can attach more than one activity to a custom block.

Tutorial: Rotating Blocks by 90 Degrees

In this tutorial, you learn how to rotate a custom block — after it is inserted — by 90-degree increments. To accomplish this, you use rotation sets in the Block Editor environment. Follow these steps:

1. Double-click the office-chair-custom block to open it in the Block Editor.
2. 

Let's add the rotation element: from the Elements panel, choose **Rotation**, and then ignore the first set of prompts:

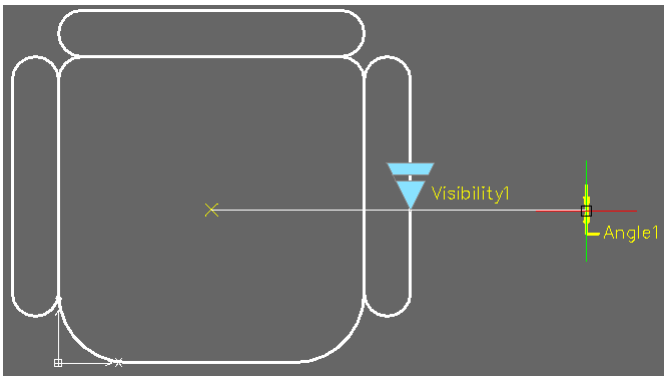
```
_CBELEMENT
Options: Base, pOint, Linear, Polar, Xy, Rotation, mIrror, Visibility or taBle
Specify element type» _Rotation
Options: Name, Label, Chain, Description, Palette or Value Set
```

   a. The first prompt asks you for the *base point*. This is the point about which the chair will rotate. The center of the chair is a good place to specify; see the small yellow x in the figure below. Pick center of the chair using the **m2p** command modifier, and then the **MIDpoint** entity snap twice:

```
Specify base point» m2p
First point of mid mid
of (Pick one line)
Second point of mid mid
of (Pick other line)
```

b. The next prompt wants a *radius*, which means how far the element label "Angle1" is placed from the base point. The best place is outside of the block, away from the visibility grip:

```
Specify radius of element» (Pick a point)
```



Placing the radius of the element

c. Specify the default angle as 0, which means the custom block initially appears with a rotation angle of 0:

```
Options: Base angle or Specify default rotation angle» 0
```

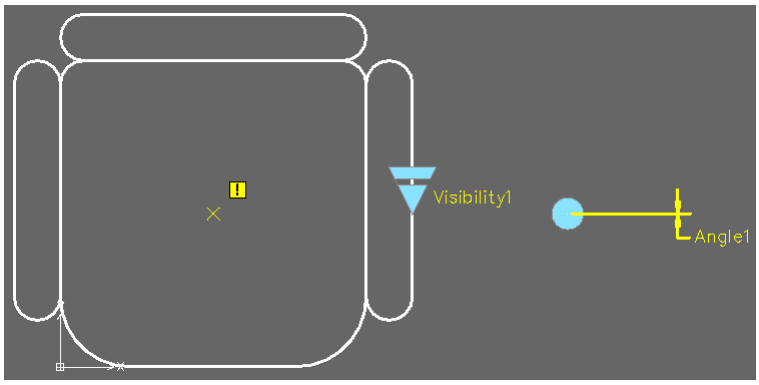
d. Next, pick a point to locate the label. The location is not important, except that it shouldn't be on top of other custom block controls.

```
Specify label location» (Pick a convenient location)
```

e. The *grip count* refers to the number of grips that this element will display on the custom block. Specify 1 grip, as 0 grips means there will be no grips displayed, and so you would be unable to rotate the chair!

```
Default: 1
Options: 0 or 1
Specify grip count» 1
```

The grip appears as a cyan-colored dot.



Rotation element in place

4. With the rotation element in place, notice the yellow exclamation mark. It reminds you that you need to tell the program which entities it applies to. All of them, naturally. Here's how:

a. 

From the Activities panel, select **Rotate**.

```
: _CBACTIVITYTOOL
Options: pattErn, taBle, mIrror, Move, Rotate, Scale, sTretch or Polar
stretch
Specify activity type» _Rotate
```

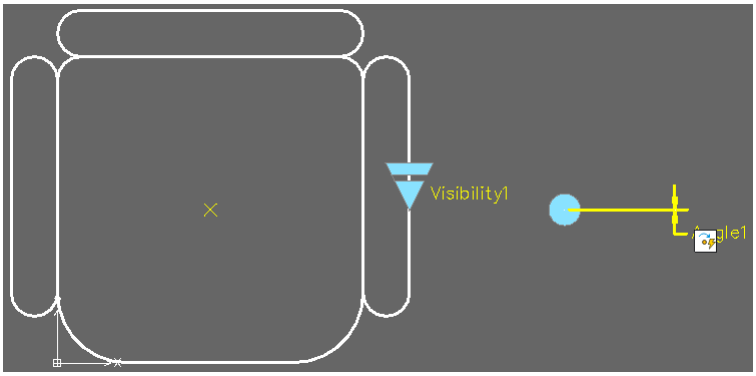
b. Select the element to which this activity applies. It is the rotation element:

```
Specify element» (Pick the yellow exclamation mark)
```

c. Now select the entities to be rotated, which is all of them:

```
Specify entities to rotate» all
22 found
Specify entities to rotate» (Press Enter to end the command)
```

So now you have applied the Rotation element and associated it all the entities to it via the Rotate activity. Together, the two are known as an *element set*. The yellow exclamation mark is gone, indicating you have satisfied the requirements of the element.

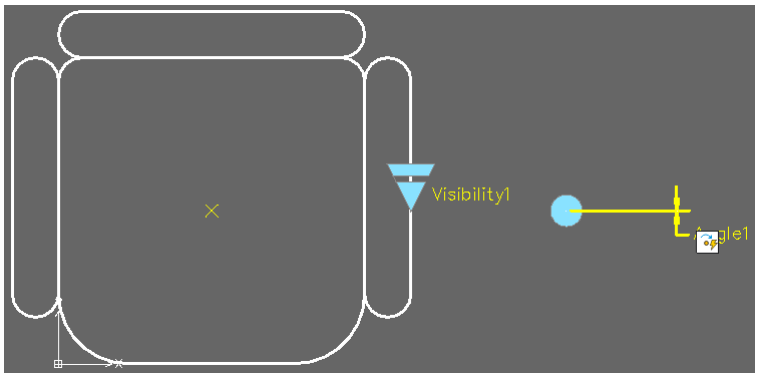


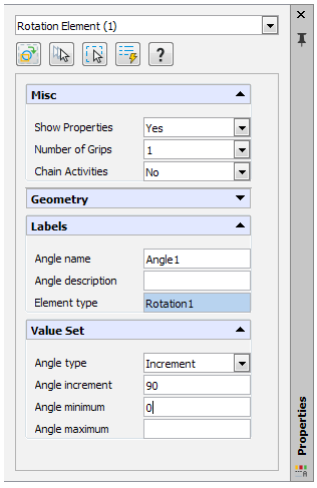
Rotation element satisfied

5. The final step is to define the rotation parameters. This is done with the Properties panel:

a. If the panel is not visible, press **Ctrl+1**.

b. Select the rotation element so that you can modify its properties.





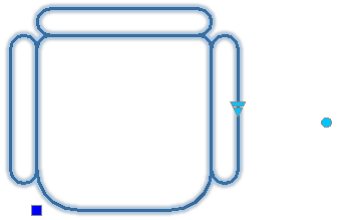
Editing the properties of the Rotation element

c. In the Properties panel, notice that you can change nearly all of the element's properties. We are just interested in the **Value Set** section, near the bottom. Make the following changes:

Property	Value	Meaning
Angle type	Increment	The block will be rotated in increments, instead of smoothly
Angle increment	90	The block will rotate in increments of 90 degrees

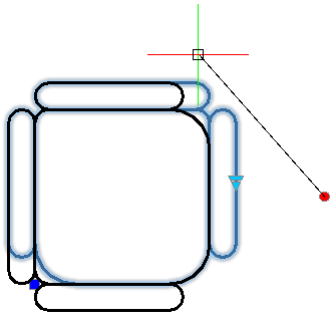
Leave the **Angle Minimum** at 0 and **Angle Maximum** blank

6. As this custom block has changed, you should change its name so that its definition does not overwrite the earlier version.
   a. Click **Close**.
   b. Choose **Save by Another Name**.
   c. Enter *office-chair-custom-rotation*
   d. Click **OK**
7. Time to test it!
   a. Use the **Inert** command to insert the new *office-chair-custom-rotation* block in the drawing.
   b. Select the block.



Custom block with two grips

c. Notice the cyan-color circle grip. It means that it rotates the block when you drag it. As you drag the round grip, notice that you are changing its angle after it was inserted, and that the rotation occurs in 90-degree increments. You don't need to use the Rotate command or enter any options.



Rotating a custom block in 90-degree increments

In summary, to attach an action to a custom block:

i. Chose an element, such as Rotation.

ii. Connect the element to the entities with a related action, in this case Rotate.

iii. Specify the angle's properties in the Properties panel's Value Set section.

SETTING UP STRETCH ELEMENTS

Command: CbElement Stretch
Ribbon: Block Editor | Elements | Stretch
Toolbar: Block - Elements | Stretch

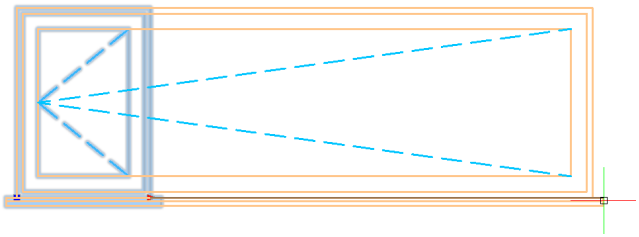
You saw how custom blocks can hide portions of themselves and rotate in specific increments. Let's turn to another useful action: stretching in increments.

Prefabricated windows come in standard sizes, two to six inches apart. Here is a typical range of sizes:

Width in inches	Height in inches
18, 20, 24, 28, 30, 32, 36, 42, 48, 54, 60	18, 20, 24, 28, 30, 36, 42, 48

You could draw 88 blocks, one for each combination of width and height dimensions. Or you can construct one custom block that handles all of them. You wouldn't use visibility states, because that would mean you still would have to draw 88 blocks, and then toggle their visibility, from a very long droplist. Impractical!

The solution is to stretch the window frame, in increments, using this procedure:



Stretching a custom window block from 18 to 60 inches

i. To show the direction of the stretch, apply the linear element.

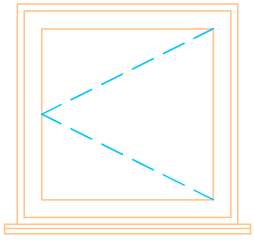
ii. To specify which entities move, which ones stretch, and which stay in place, add the stretch activity.

iii. To limit the stretching to specific widths (18, 20, 24, 28, 30, 32, 36, 42, 48, 54, and 60 in this case), create a distance list.

Tutorial: Stretching Blocks in Increments

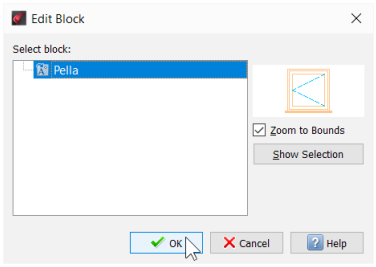
In this tutorial, you learn how to construct a custom block that changes the width of a window by specific amounts.

1. Start the program, and then open the *window-pella.dwg* file. This window measures 18" x 18", the smallest standard size.



The window-pella drawing with 18"x18" window block

2. Double-click the *Pella* block, and then in the Edit Block dialog box click **OK**.



Selecting the Pella block to edit

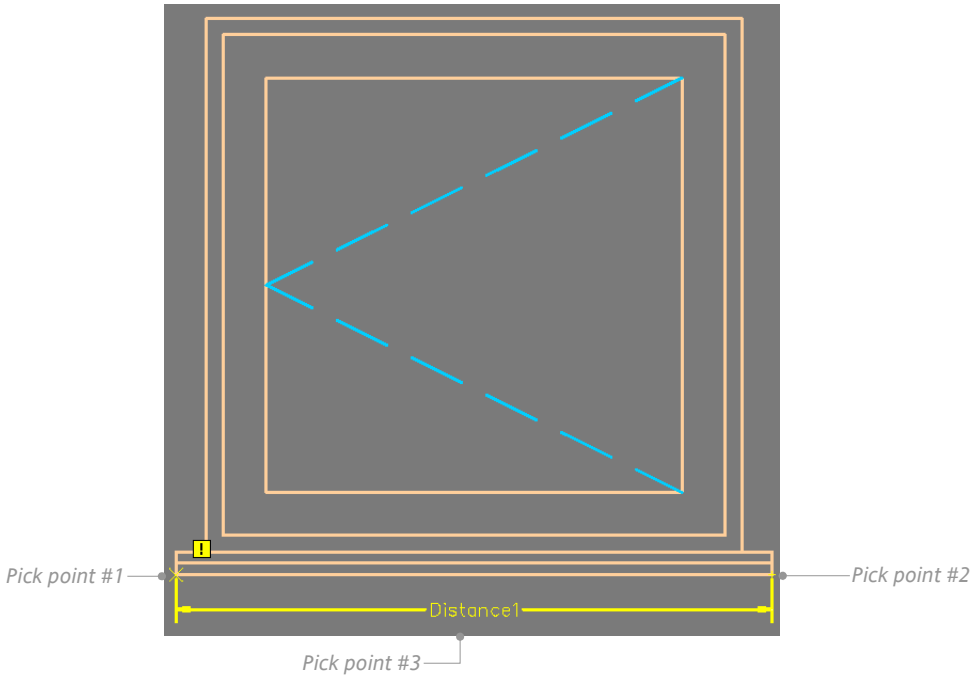
3. Apply the linear element:

   a. 

In the ribbon's Block Editor tab, choose **Linear** from the Elements panel. As before, ignore the first several lines of prompts:

```
: _CBELEMENT
Options: Base, pOint, Linear, Polar, Xy, Rotation, mIrror, Visibility or taBle
Specify element type» _Linear
Options: Name, Label, Chain, Description, Base, Palette or Value Set
```

   b. Specify the location of the linear element, which looks kind of like a linear dimension and is placed exactly like one:



Placing the linear element

```
Specify start point» (Pick intersection at #1)
Specify end point» (Pick intersection at #2)
Specify label location» (Pick a convenient location, such as at #3)
```

c. Specify the number of grips. Here is the difference between the three values:

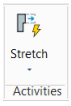
Grips	Meaning
0	**No grips** are displayed; not that useful 
1	**One grip** is displayed, so that the window block is lengthened in one direction only 
2	**Two grips** are displayed, so that the window can be lengthened in both directions 

For this tutorial, enter 1:

```
Default: 2
Options: 0, 1 or 2
Specify grip count» 1
```

4. Apply the stretch activity, which works much like the Stretch command. You associative the stretch activity to two things to the linear element in two steps:

   » Attach the stretch activity to the linear element, so that the program knows what to do when you move the linear grip.
   » Select the entities of the window symbol to be stretched, moved, or left in place.

a. 

In the ribbon's Block Editor tab, choose **Stretch** from the Activities panel. As always, ignore the first several lines of prompts:

```
: _CBACTIVITYTOOL
Options: pattErn, taBle, mIrror, Move, Rotate, Scale, sTretch or Polar stretch
Specify activity type» _sTretch
```

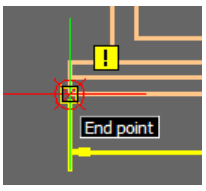
b. Associate stretching with the linear element by selecting it:

```
Specify element» (Select the linear element)
1 found
```

Stretch activity selecting the linear element, in orange

c. Now the program wants to know where the stretching starts. Pick the start point, so that it stays in place, and the rest of the window stretches:

```
Default: Second
Options: sTart point or Second point
Specify element point to associate with activity» (Select the start point)
```



Selecting the start point

d. You might recall that the Stretch *command* uses a CWindow (crossing window) selection set to determine which elements are stretched, moved, or not moved. Entities are...

- **Stretched** when crossing the selection window
- **Moved** when fully inside the selection window
- **Not moved** when fully outside the selection window

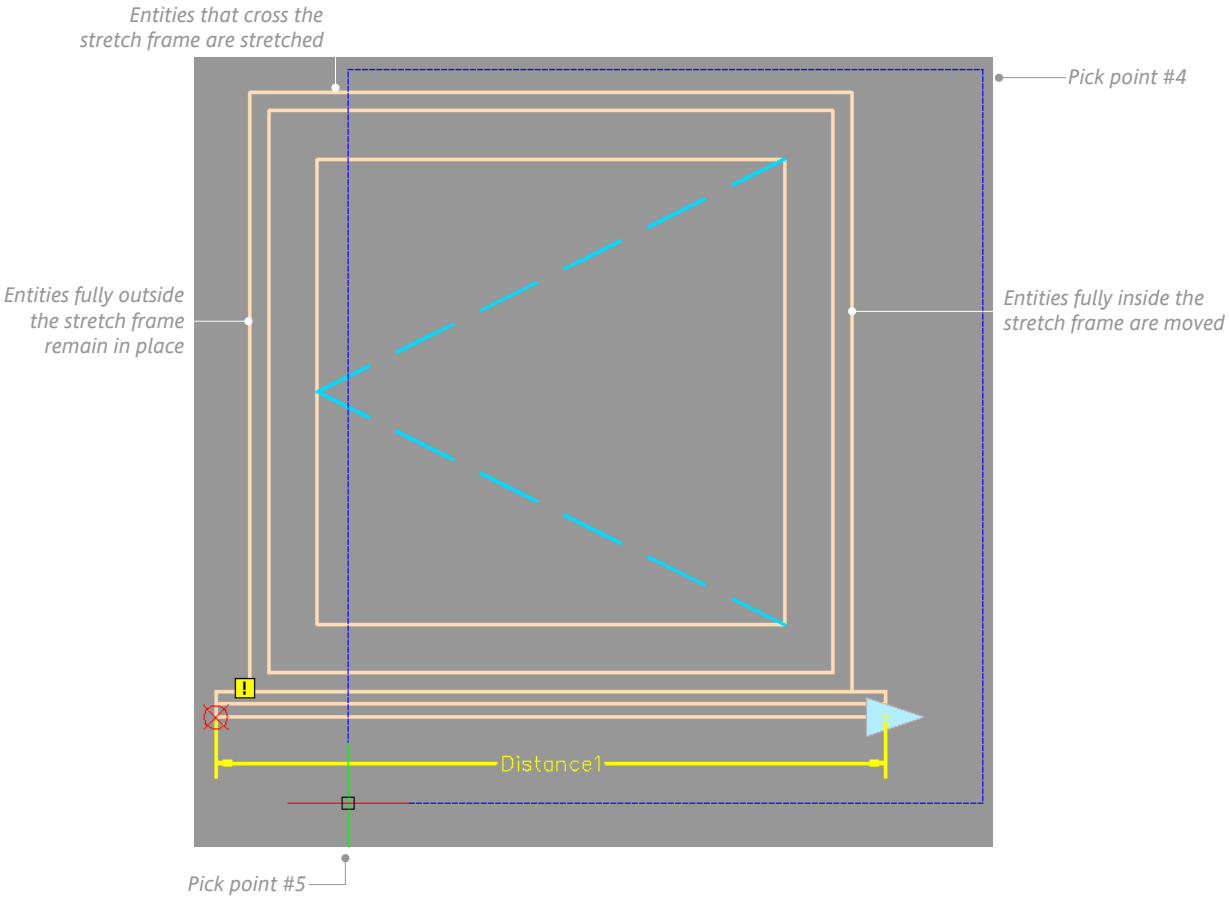
Now, the stretch *activity* uses a "stretch frame," in addition to the selection window. In many cases, the frame is the same as the selection window.

For added flexibility, however, the frame adds a second set of conditions. The first two conditions are just like the selection window. Entities are...

- **Stretched** when crossing the frame
- **Moved** when fully inside the frame

The additional conditions imposed by the stretch activity are different. Entities are...

- **Moved** when in the selection set, but outside the frame
- **Not moved** when not in the selection set, but within or crossing the frame



Placing the stretch frame, carefully

```
Options: CPolygon
Specify first corner of stretch frame» (Pick a point near #4)
Specify second corner of stretch frame» (Pick a point near #5 so that the high-
lighted entities are selected)
```

In the figure, the stretch frame is the blue rectangle. Notice carefully which entities are fully inside, which cross the frame, and which are outside of it.

e. Now you can select the entities that are part of the selection set. Click at the same two places but with this important consideration: *make sure that the crossing window encompasses the blue stretch frame!*

```
Specify entities to stretch» (Pick a point just outside of #4)
(Pick a point just outside of #5)
25 found
Specify entities to stretch» (Press Enter to end the command)
```

5. Before adding the increment data, try testing the stretch activity:

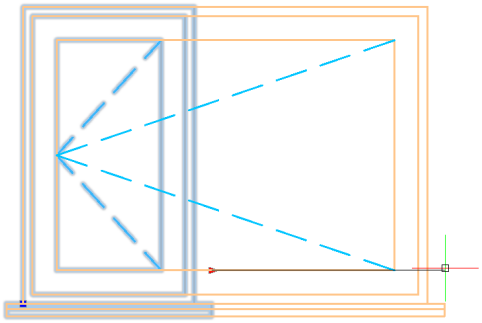
   a. Click **Save**.

   b. Click **Close**.

   c. Pick the custom block. Notice the small, cyan-colored triangle grip: this is the linear grip.

Triangle grip

   d. Drag the grip. Notice that the window changes its width, smoothly.



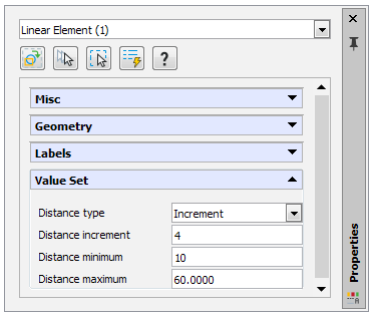
Draggin' the grip

   e. Press **Esc** to end the block selection.

   f. Double-click the block to get back to the Block Editor environment.

6. The next step is to add distance increments to the stretching. First we'll do the easy version, where the increment distance is the same (four inches, an average change in window sizes).

   a. Select the linear-stretch set.



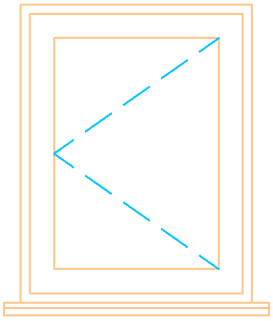
Value Set section of the Properties palette

b. Open the Properties palette to the **Value Set** section, and then make the following changes:

Value Set		Value	Comment
Distance Type	Increment		
Distance Increment		4	An average change in window size
Distance Minimum		8	The smallest pre-fab window size
Distance Maximum		60	The largest window size

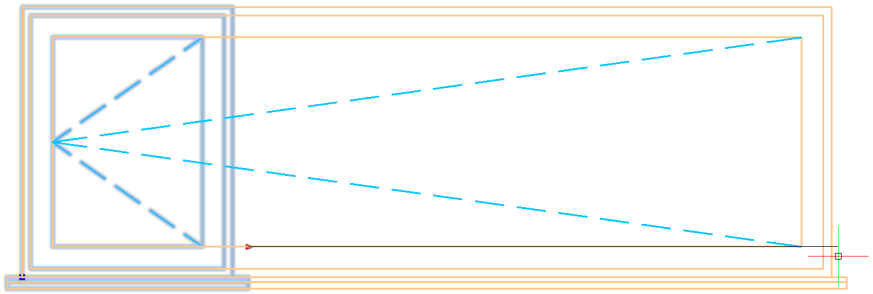
7. Test the changes you made:

a. Click **Save**, and then click **Close**. Notice that the window looks narrower, because the program is showing it at the smallest distance of 18".



Window frame at its narrowest width of 18"

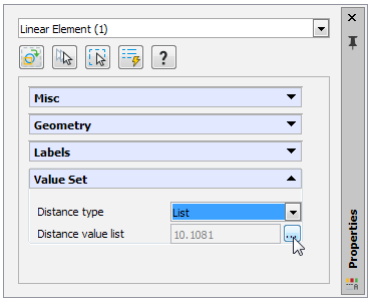
b. Pick the custom block, and then drag the grip. Notice that the window changes its width in 4-inch increments.



Window frame at its narrowest (18") and widest (60" extents

c. Press **Esc** to end the block selection, and then double-click the block to return to the Block Editor.

8. But the increments range from 2" to 6". So we need to change the increments from a regular number (4) to a list of widths. This is done by creating a list of all the widths:



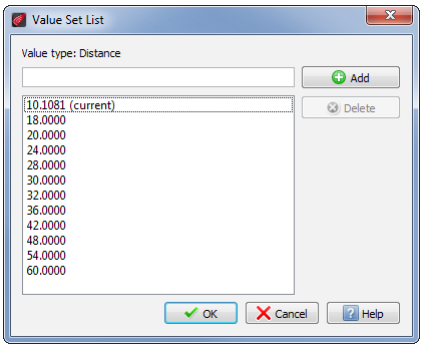
Accessing the distance value list

a. Select the linear-stretch set

b. Open the Properties palette to the **Value Set** section.

c. Change the Distance Type to **List**.

d. Click the **Browse** button (...) in the Distance Value List field. Notice the Value Set List dialog box.



Entering width values into the Value Set List dialog box

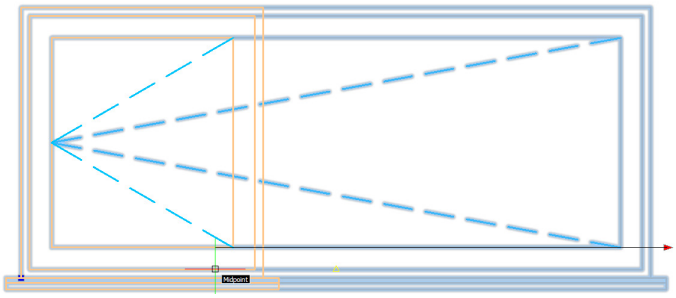
e. And then add the standard widths to the dialog box, one by one. Enter a value, and then click **Add**:

```
18, 20, 24, 28, 30, 32, 36, 42, 48, 54, 60
```

9. Test, for the final time hopefully, the changes you made:

a. Click **Save**, and then click **Close**.

b. Pick the custom block, and then drag the grip. Notice that the window changes its width in 2- to 6-inch increments, according to the sizes you specified.



Window frame changing its width in specific increments

In summary, you construct a stretchable custom block using this procedure:

i. Apply the linear element to show the direction of the stretch,

ii. Add the stretch activity to the linear element to specify which entities move, which ones stretch, and which stay in place.

iii. Create a distance list to limit the stretching to specific widths.

You could add a second linear-stretch set to change the height of the window.

Summary

Custom blocks are the next level in using symbols to increase your productivity.

Meanwhile, the floor plan drawing is nearly complete. It has all its graphical elements, such as walls and desks. There is one more type of drafting that your drawing needs: annotations. This includes items like text for descriptive labels and dimensions to show sizes. The next two chapters explain how to place text and add dimensions.

CHAPTER 09

Placing Text and Attributes

In This Chapter

- Placing notes in drawings
- Creating text styles
- Editing text
- Adding attributes to blocks
- Exporting attributes to spreadsheets
- Importing spreadsheet data as tables
- Editing table text and cells

Notes are added to drawings to explain the designer's intent. For instance, it is common to label the rooms of floor plans with text, also known as "callouts" and "annotations." The text explains the functions of the room.

The program supports two types of text: simple and paragraph:

» *Simple text* consists of a single line of words, sometimes called a "string"; it is placed with the SimpleNote command

» *Paragraph text* consists of multiple lines of text arranged like paragraphs, sometimes called a "block" of text; it is placed with the Note command

Make a mistake? You edit simple and paragraph text with the EditAnnotation command, and then use the SpellCheck command to find *missteaks* you may have missed. Styles are used to determine the look of the text. It is not uncommon to attach text data to blocks so that they can be counted by a spreadsheet.

In this chapter, you place text in the *office layout.dwg* drawing, and then change its look through styles. In addition, you learn how to attach attribute data and then export it to a spreadsheet for further analysis. Once in the spreadsheet, the data is brought back into the program as a table, and then the table can be edited.

KEY TERMS IN THIS CHAPTER

Annotation refers to notes placed in drawings.

Attribute refers to textual data included with blocks.

Bounding box specifies the margins of paragraph text.

Extract exports attribute data to a file.

Fixed attribute refers to an attribute value that is constant.

Justification refers to horizontal positioning of text, such as left, right, or center.

Name (or Tag) identifies attributes by name.

String refers to a single line of text.

Style specifies the font and other properties of text.

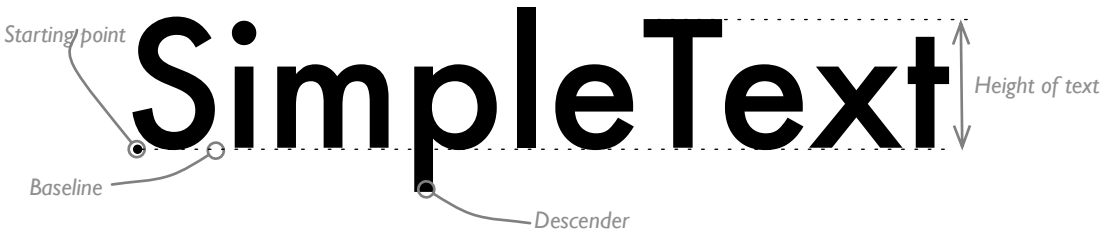
COMMANDS

Command	Shortcuts	Menu Selection
BlockAttributeOutput	attout	Tools | Attribute Export
EditAnnotation	ed	Modify | Entity | Annotation
EditBlockAttribute	attedit	Modify | Entity | Block Attribute | Single
EditNote	mtedit	Modify | Entity | Annotation
MakeBlockAttribute	attdef	Draw | Block | Define Attributes
Note	t, mtext	Draw | Text | Note
NoteOptions	...	...
-SimpleNote	text	Draw | Text | Simple Text
TextStyle	style	Format | Text Style

Placing Simple Text

Command: -SimpleNote
Aliases: text, dtext, snote, dt
Menu bar: Draw | Text | Simple Text
Ribbon: Annotate | Text | Note | SimpleNote (Drafting and Annotation)
Toolbar:

In this tutorial, you label the reception area and the storage room of the drawing with pieces of text. The figure below points out the most important parts of text in CAD drawings:



Text with the parameters that define it

Starting point — the point at which the text is placed; also known as the *start position* or *insertion point*

Baseline — the line along which text is placed; excludes descenders

Height of text — the distance from the baseline to the top of uppercase letters

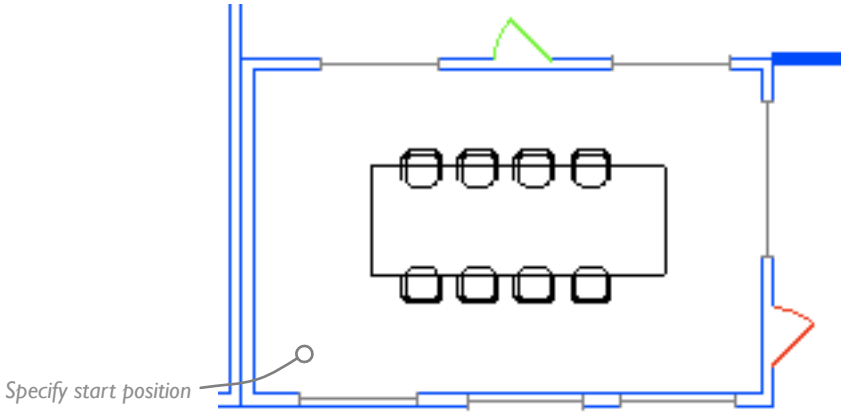
Descender — portions of lower-case letters that go below the baseline

Follow these steps to place some text in the drawing:

1. Start the **-SimpleNote** command using one of the methods listed above.
2. In the command bar, the program prompts you to pick a point in the drawing:

```
: -simplenote
Options: Justify, textSTyle or
Specify start position» (Pick a point at which to start the text)
```

3. Pick a point below the conference table.



Picking the point at which to begin the line of text

TIPS You may wish to zoom into the room to see it more clearly. You can zoom and pan during commands, such as during the -SimpleNote command: just roll the mouse wheel until the drawing is a good size.

The dash prefix in **-SimpleNote** means that it operates in the Command window; the related SimpleNote command displays a dialog box.

4. The command would like to know the height of the text. You can enter a height in millimeters, but in this tutorial pick two points:

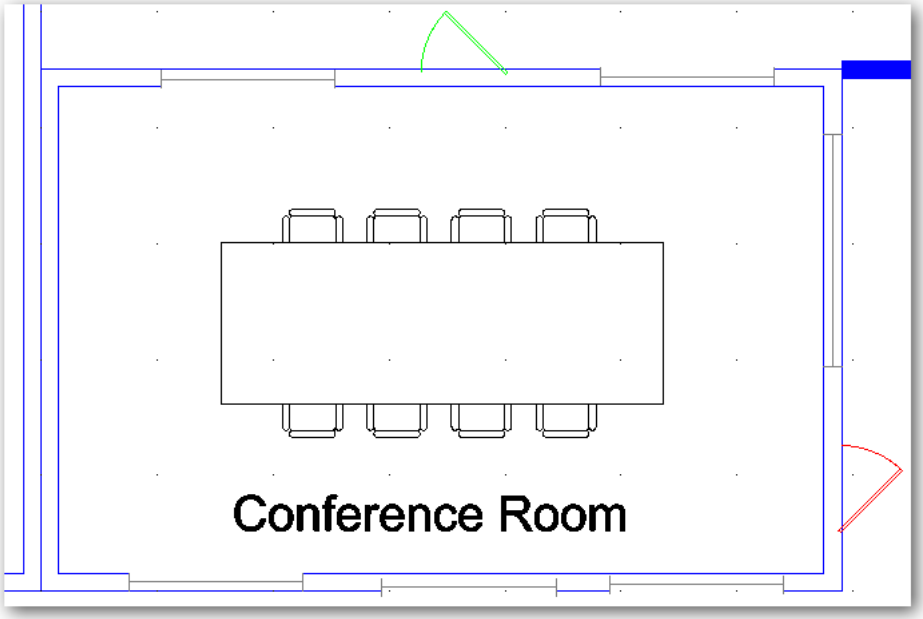
```
Default: 0.20
Specify height» (Pick two points to indicate the height)
```

5. Text can be placed at any angle. For horizontal text, use the default value of 0:

```
Default: 0
Specify text angle» (Press Enter to accept the default angle)
```

6. Finally, you get to specify the text. Type "Conference Room," and then press **Enter**.

```
Specify text» Conference Room (Press Enter to indicate the end of the line of
text)
```

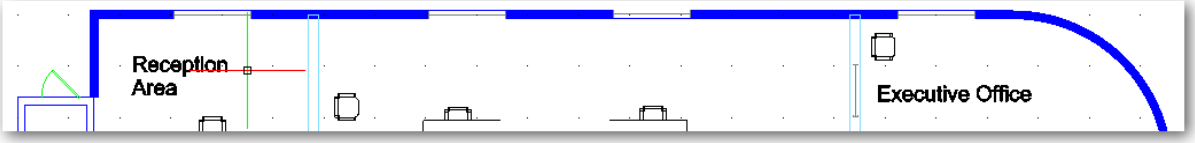


Text placed in drawing

7. Press **Enter** one more time:7

```
Specify text» (Press Enter to end the command)
```

Go ahead and repeat the **-SimpleNote** command to place text in the reception area and executive office, as illustrated below.



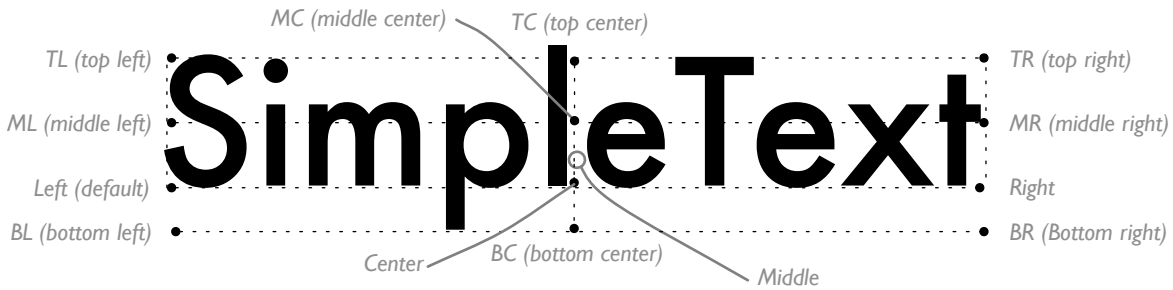
Additional lines of text added to the drawing

SUMMARY OF -SIMPLENOTE COMMAND OPTIONS

```
: -simplenote
Options: Justify, textSTyle or
Specify start position» (Enter an option.)
Specify height» (Enter a height.)
Specify text angle» (Enter an angle.)
Specify text» (Enter text, and then to exit press Enter twice.)
```

You can specify the start position, height, and angle by picking point(s) in the drawing, or by entering appropriate values, such as coordinates, a number, or an angle. **Justify** specifies the text justification:

```
Options: Align, BC, BL, BR, Center, Fit, Middle, MC, ML, MR, Right, TC, TL, or TR
Specify option» (Enter an option.)
```



Justify	Meaning
Align	Text is fitted between two points; height is proportional to length of text
BC	Bottom center (see figure below)
BL	Bottom left
BR	Bottom right
Center	Center of the text's baseline
Fit	Text is fitted between two points
Middle	Center of the text's bounding box
MC	Middle center
ML	Middle left
MR	Middle right
Right	Right end of the text's baseline
TC	Top center
TL	Top left
TR	Top right

textSTyle specifies the name of a style predefined by the TextStyle command:

```
Default: Standard
Options: ? to list or
Specify name» (Enter the name of a style, or type ? to list the styles
stored in this drawing.)
```

You will find yourself in the program pressing **Enter** repeatedly for different reasons. The reasons depend on the prompt displayed by the program. Here are the most common occurrences:

» Pressing **Enter** at the ':' prompt repeats the command:

```
: (Press Enter to repeat the -SimpleNote command)
Options: Justify, textSTyle or
Specify start position» (Pick a point in the reception area)
```

» Pressing **Enter** at the following prompts accepts *default* values stored by the program:

```
Default: 300.88
Specify height» (Press Enter to accept default height, 300.99 in this case)
Default: 0
Specify text angle» (Press Enter to accept default angle, 0 in this case)
```

» Pressing **Enter** indicates the end of the line of text:

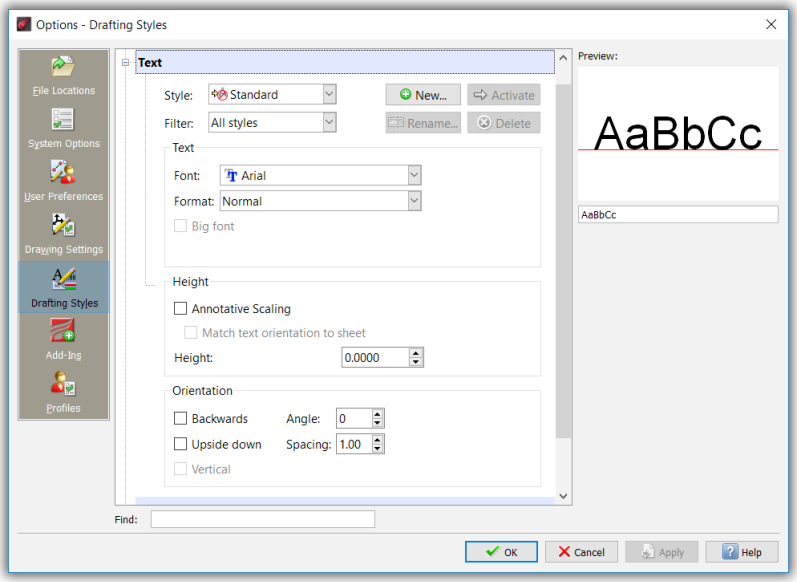
```
Specify text» Reception (Press Enter to end line of text)
Specify text» Area (Press Enter to end line of text)
```

» Pressing **Enter** ends the command at the end of a command:

```
Specify text» (Press Enter to end the command)
```

SETTINGS TEXT STYLES

Command: TextStyle
Aliases: st, style, txs
Menu bar: Format | Text Style
Ribbon: Annotate | Text | Text Style (Drafting and Annotation)
Toolbar:



Options dialog box displaying settings for text styles

The text is being drawn with the Arial font, but the program can use just about any font found on your computer. Specifically, it understand fonts in TrueType (TTF) and AutoCAD (SHX) formats, but not in Post-Script formats. The Windows and Mac OS X versions use Microsoft's Arial font as the default; the Linux version uses ARSimp.Shx, a font installed by the program.

In CAD, you do not apply fonts directly to text as you would with a word processor. Instead, you specify the name of a font indirectly through a *text style*. The text style is then applied to the text. Styles define other properties as well, such as the angle of the text or its *orientation* (whether it should be drawn backwards or upside down). Change the style, and all text given that style also changes its look.

Text styles are defined through the **TextStyle** command. (The program also uses other types of styles for predefining the look of dimensions and tables.)

In this tutorial, you change the font in the drawing below to Times New Roman.

1. Start the **TextStyle** command. Notice that the Options dialog box opens to the Text section:

SUMMARY OF TEXTSTYLE DIALOG BOX OPTIONS

```
: textstyle
(Displays Options dialog box at the Drafting Styles | Text section.)
```

Style lists the text styles stored in the active drawing.

New names new text styles; displays the Create new TextStyle dialog box.

Activate activates the selected style, making it the default.

Rename renames the selected style.

Delete erases the selected style from the drawing; you cannot erase the style named Standard, or styles used by text in the drawing.

TEXT OPTIONS

Font lists the file names of TTF (TrueType) and SHX (shape) fonts available on your computer.

Format assigns normal, *italic*, **bold**, or *bold italic* format to TrueType fonts.

Big Font toggles the listing of SHX "big fonts" that accommodate extended character sets.

HEIGHT OPTIONS

Annotative Scaling scales the text automatically to match the sheet (layout) and plot scales.

Match Text Orientation to Sheet rotates the text so that it is right-side up in sheets (layout mode).

Height specifies the height; leave this value at 0 if you wish to specify the height during the -SimpleNote command. (The Note, -Note, and SimpleNote commands can override the height.)

ORIENTATION OPTIONS

Backwards draws text backwards.

Angle specifies the slant at which to draw the text; positive angles slant the text forward; negative angles backwards. Range is -84.9 to 84.8 degrees.

Upside Down draws text upside down.

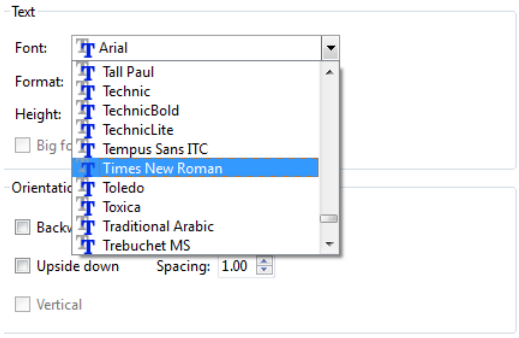
Spacing changes the aspect ratio; values larger than 1 expand the width of the text, while smaller values compress it.

Vertical draws certain SHX fonts vertically.

2. Change the font as follows:

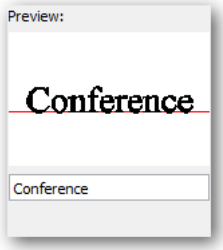
   a. Click the droplist next to **Font**.

   b. Scroll through the list of font names until you come to **Times New Roman** and then select it. (The fast way to get there is to press the letter 't'.)



Selecting the new font

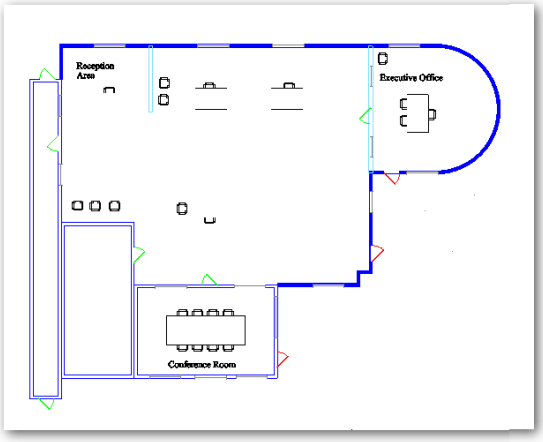
3. Notice that the Preview image updates to show the changed font. In addition, you can enter any text you wish to preview. For example, enter **Conference** and it looks like this:



Preview of the changed text style

4. Click **OK** to dismiss the dialog box.

   Notice that the text strings in the drawing change from Arial to Times New Roman font. From now on, all text uses this font — until you again change the style.



Text reflecting the changes to the style

While it may seem indirect to use styles for specifying fonts, they allow for very quick global changes.

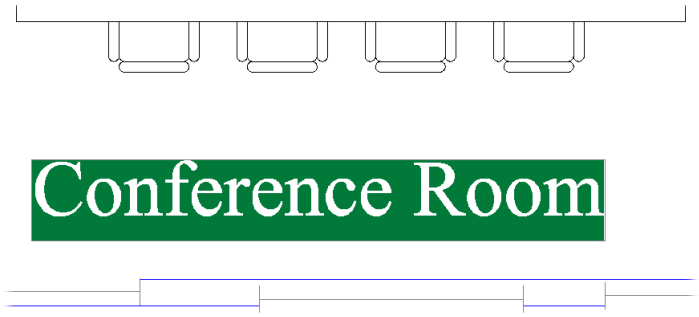
EDITING TEXT

Command: EditAnnotation
Aliases: edittext, ddedit, ed, edanno
Menu bar: Modify | Entity | Annotation
Ribbon: Annotate | Text | Text Edit (Drafting and Annotation)
Shortcut: *Double-click the text*
Toolbar:

Text is edited with either the **EditAnnotation** command or the Properties palette — your choice. Of the two, you might find the EditAnnotation command more straightforward, because it has fewer options than the Properties palette.

In this tutorial, you change the label Conference Room to "Board Room".

1. Double-click the text. Notice that the program highlights the text in green.



Double-clicked text highlighted in green

(In the background, the program launches the EditAnnotation command.)

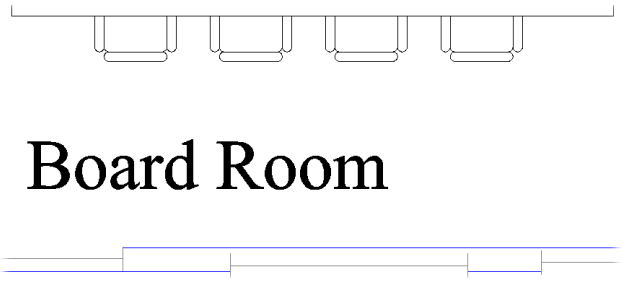
2. Double-click the word "Conference."

Choosing the text to edit

3. To erase the highlighted text, press the **Backspace** key.
4. Enter the new text:

```
Board
```

5. Exit the editing mode by pressing **Enter** twice.



New text replacing old text

This style of editing is called "in-place" editing, and is useful, because it lets you concentrate on correcting text or changing it.

Switching to Dialog Box Editing

Command: NoteOptions
Aliases: --
Menu bar: --
Ribbon: --
Toolbar: --

Should you wish to make other changes to the text like the height of the text or its justification, then you need to switch to the dialog box version of EditAnnotation. The **NoteOptions** command tells the program what kind of interface you prefer:

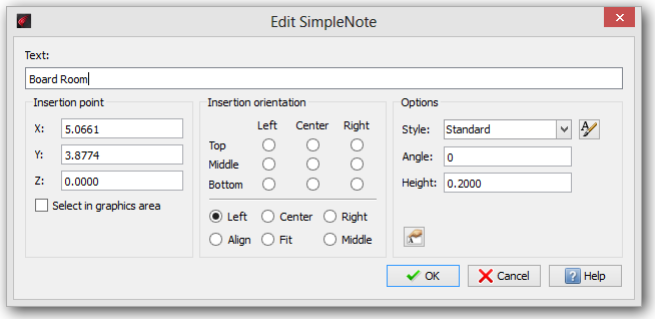
1. Enter the **NoteOptions** command.

```
: noteoptions
```

2. Specify the **Simplenote editor**, and then choose the **Dialog box** option, because this option turns on the dialog box.

```
Default: Note editor
Options: Note editor, Simplenote editor, Display options or sample Text
Specify option» s
Default: Advanced
Options: Dialog box, In-place or Advanced
Specify option» d
```

3. Now when you double-click the text (or enter the EditAnnotation command), a dialog box appears instead.



Parameters that modify text

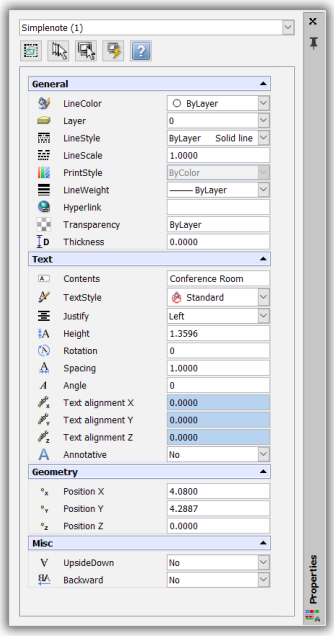
It lets you change the following parameters of the selected text:

- Text wording
- Insertion point through specifying coordinates or by picking a point in the drawing
- Justification mode (a.k.a. insertion orientation)
- Style name, and editing styles through the Drafting Styles dialog box
- Angle and height
- Insert and edit field text

4. Click **OK** to close the dialog box.
5. Press **Esc** to exit the EditAnnotation command.

Editing Text with Properties

Another way to edit text is through the Properties palette. To access the palette, single-click the text. (If the palette is not visible, turn it on with the Properties command.) In the palette, notice that the text-oriented properties are in three sections: Text, Geometry, and Misc.



Properties palette showing settings that can be changed for text

The Properties palette is easier to get into (a single click of the text) and out of (press **Esc**) than the Edit Simple Note dialog box. Which one you use is a matter of personal preference!

PLACING PARAGRAPH TEXT

Command: Note
Aliases: n, t, mtext, mt
Menu bar: Draw | Text | Note
Ribbon: Annotate | Text | Note (Drafting and Annotation)
Toolbar:

The -SimpleNote command places just one line of text at a time. To enter entire paragraphs of text, use the **SimpleNote** (without the hyphen prefix) or **Note** commands. Here is the difference between the two:

SimpleNote

- Displays a dialog box that looks like Edit Simple Note, but with room to write more than one line of text at a time
- Limits the formatting options to style, justification, height, and angle, and applies to all text
- Places independent lines of text, and so does not produce true paragraph text

Note

- Displays a toolbar or ribbon
- Makes available almost any kind of text formatting, which can be applied to all or some of the text, even to individual characters
- Places text as true paragraphs that reflow when the text boundary changes

In this section, we'll concentrate on the Note command. You can edit and spell check notes, as well as apply styles, just as with simple text. But better yet, you can override the properties of notes, like underlining or and boldfacing individual characters, even change colors and sizes.

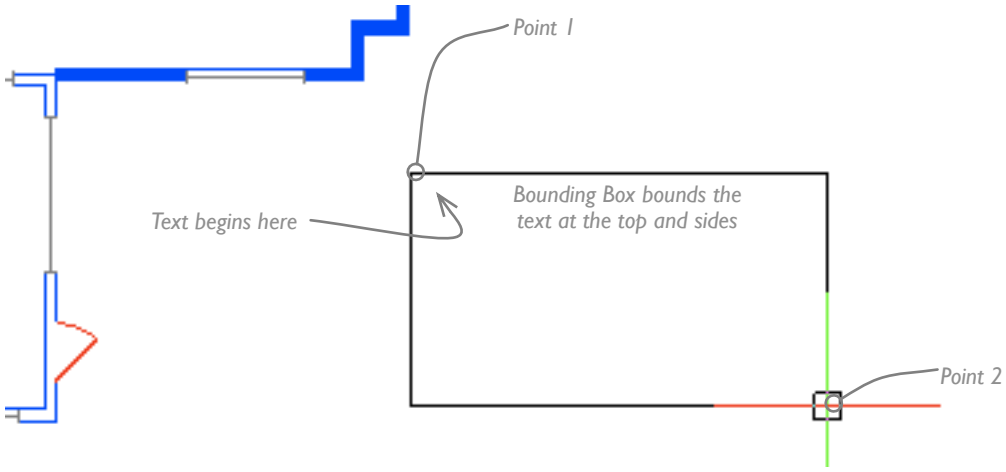
Note text is placed within an invisible rectangle called the "bounding box." As you make the bounding box wider or narrower (to fit space in the drawing), the program makes the note reflow and automatically shortens or lengthens the box.

In the following tutorial, you add a paragraph of text to the lower right area of the drawing:

1. Start the **Note** command using one of the methods listed above.

   ```
   : note
   ```

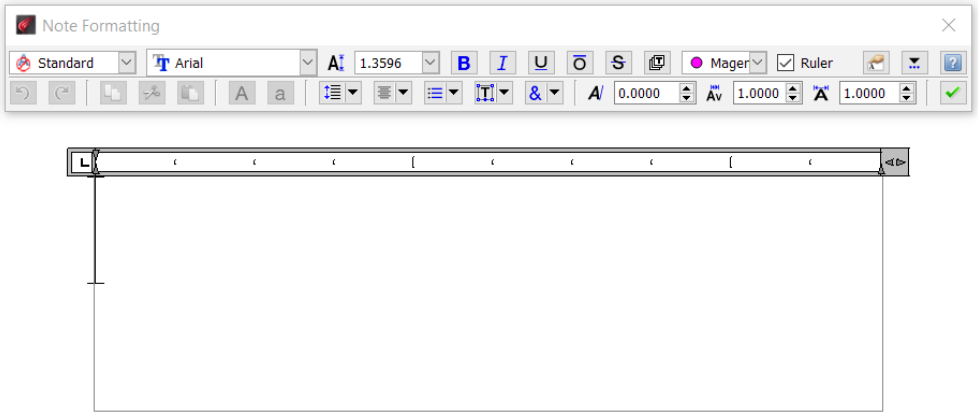
2. Notice that the program prompts you to pick two points that define the bounding box. (Later the program fits the text to the box, starting at the top, constrained by the sides, and extending as far down as necessary.)



Picking two points as the first step in placing notes

```
Active TextStyle: "Standard" Text height: 300.88
Specify first corner» (Pick a point in the drawing)
Options: Angle, Height, Justify, Line spacing, textSTyle, Width or
Specify opposite corner» (Pick another point to indicate the width of the para-
graph)
```

3. After you pick the second point, the program displays the Note Formatting toolbar.

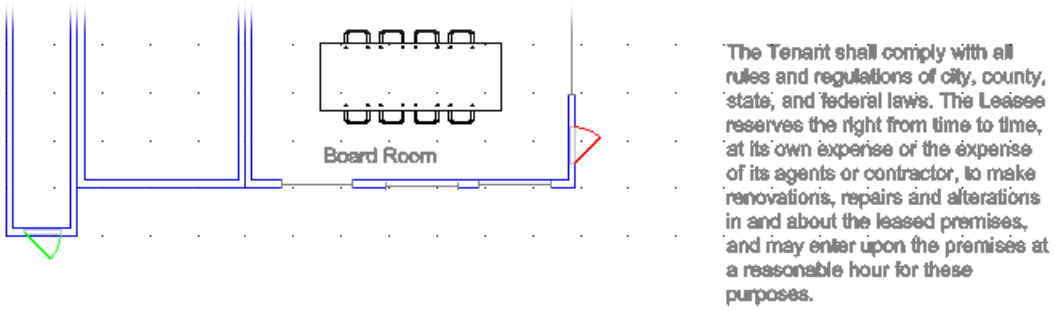


Entering note text

Enter text, such as the sample provided below.

```
The Tenant shall comply with all rules and regulations of city, county, state, and
federal laws. The Leassee reserves the right from time to time, at its own expense
or the expense of its agents or contractors, to make renovations, repairs and al-
terations in and about the leased premises, and may enter upon the premises at a
reasonable hour for these purposes.
```

4. Click **Create Note**. The text is placed in the drawing.



Completed note text

TIPS You can switch the Note user interface between a toolbar and the ribbon. Set the value of the **DTextEd** system variable as follows:

0 = Use toolbar interface (compatible with release 2013 and earlier)

1 = Use ribbon interface

The **ConvertSimpleNotes** command converts regular text into notes (mtext), while the **CurvedText** command places text along curves. Both of these commands are found in the XtraTools menu and ribbon tab.

SUMMARY OF NOTE COMMAND OPTIONS

```
: note
Active TextStyle: "Standard" Text height: 6.00
Specify first corner» (Pick a point.)
Options: Angle, Height, Justify, Line spacing, textSTyle, Width or
Specify opposite corner» (Enter an option.)
```

Angle specifies the angle of the bounding box, rotated about the first corner.

Height specifies the height of the text.

Justify specifies the text justification mode relative to the bounding box.

Line spacing specifies the spacing between lines of text.

textSTyle specifies the name of the text style.

Width specifies the width of the bounding box.

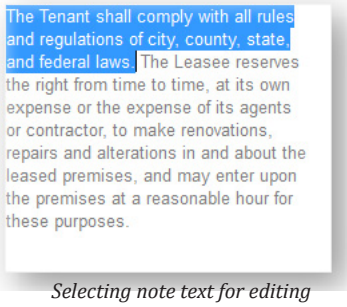
MODIFYING PARAGRAPH TEXT

Command: EditNote
Aliases: mtedit, edn
Menu bar: Modify | Entity | Annotation
Ribbon: Home | Annotate | Note (Drafting and Annotation)
Shortcut: *Double-click the text*

You can click the text to modify it with the Properties palette, or double-click to edit with the Edit Note toolbar. It looks just like the Note Formatting toolbar.

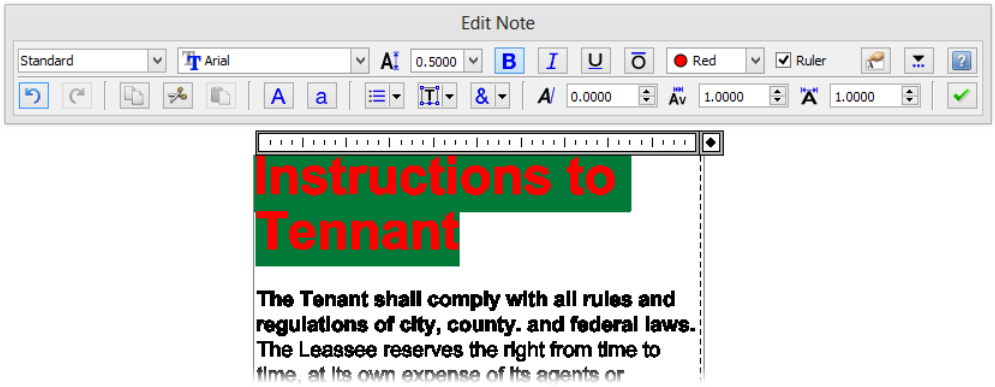
In this tutorial, you selectively change the properties of some of the text.

1. Double-click the paragraph text. Notice that the Edit Note toolbar appears, containing the text you entered earlier.
2. Apply boldfacing to the first sentence, as follows:
   a. Select the text to be boldfaced. (Click at the start of the sentence, and then drag the cursor to the period at the end of the sentence.)

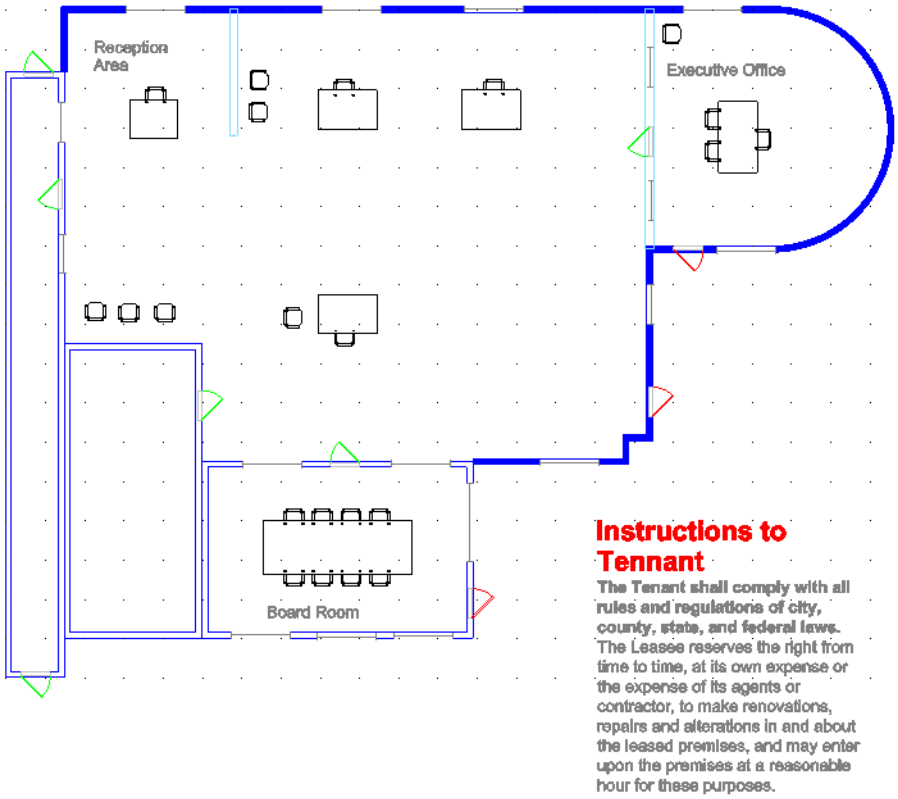


Selecting note text for editing

   b. Click the **B** **Boldface** button. Notice that the text becomes bold.
   c. Click anywhere to remove the highlighting.
3. To add a title, follow these steps:
   a. Position the cursor at the start of the text.
   b. Press **Enter** to create a blank line.
   c. Change the height to **500**mm:
   d. Change the color to **Red**.
   e. Type the title text, **Instructions to Tenant**.

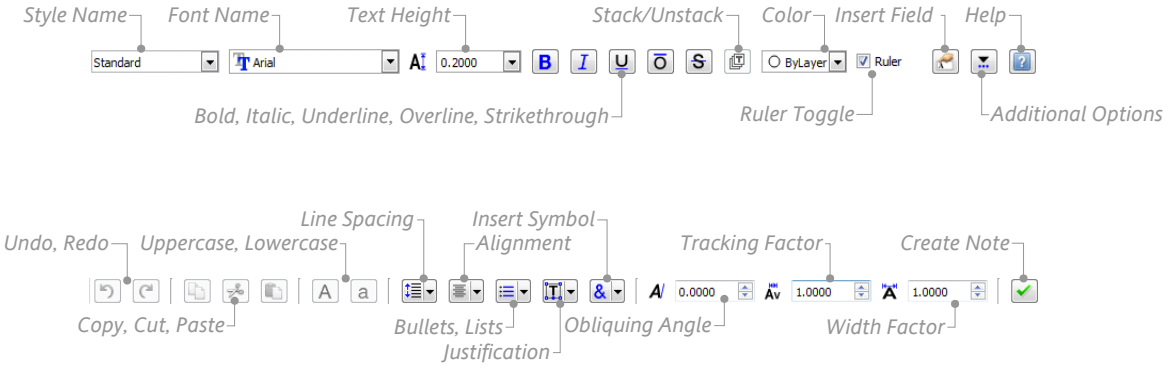


4. Click **OK** to apply the changes, and then press **Ctrl+s** to save the drawing (**Cmd+s** on Macs).

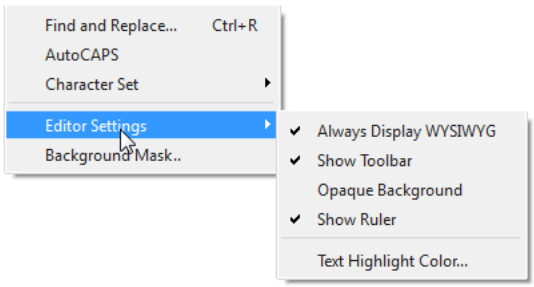


Formatted note text in drawing

SUMMARY OF EDITNOTE TOOLBAR OPTIONS



ADDITIONAL OPTIONS



Annotative Scaling

A problem related to text (and other annotative entities) rears its head when it comes time to print drawings that don't print at a scale of 1:1. Text that was perfectly legible in model space becomes microscopically tiny when printed at scale factors like 1:100 or 1" = 50'. The same problem applies to other annotative entities that depend on scale, such as linetypes, hatch patterns, dimensions, and so on.

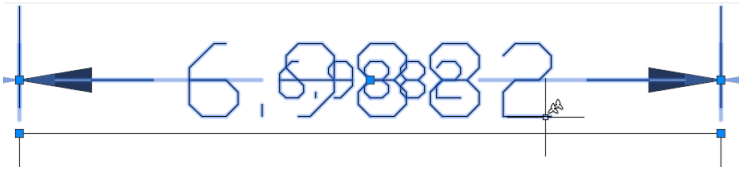
The program provides a solution in the form of a master scale factor that affects only annotative entities. The value of this scale factor is used in three areas of drawing production: annotative entities, in viewports, and on plots. This makes sense, because the whole point of viewports is to create sheets that match plot sizes 1 to 1. And so annotative scaling makes annotative objects the correct size when plotted. To understand annotative scaling, all you need is to memorize this rule:

```
Annotative scale = Viewport scale = Plot scale
```

Simple enough! The size (scale) at which you plot the drawing determines the size of the drawing in the viewport, and so determines the scale factor of annotative entities.

But there is a downside. Annotative scaling can cause objects to disappear when the viewport scale factor no longer matches the entities' annotative scale factor. This, however, also makes sense. CAD drawings can have multiple sheets, each with a different scale factor for printing drawings at a variety of paper sizes. (This is why all drawings have just one Model tab but many Sheet tabs: each Sheet tab is assigned to a different plot scale and/or a different view of the drawing.) Thus, annotatively-scaled entities should only appear at the correct scale factor.

To repeat: Annotatively-scaled objects are displayed only when their assigned scale factor matches the scale factor of model space — whether in Model tab or a layout tab's viewport. Well, there is one exception...



All annotative scale factors of an entity being displayed

If the **AnnoAllVisible** variable is turned on, then when we happen to select an annotative entity, all its variations appear, as illustrated above. This might lead to some confusion as you see multiple-sized versions of text and other entities. The function is meant for debugging: it's handy when we need to see all scales of an annotative entity. To keep us from seeing in double, triple, and quadruple vision, the program normally leave AnnoAllVisible turned off.

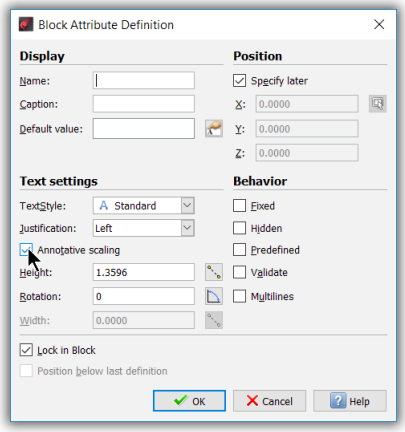
Because of these factors (or "problems"), annotative scaling has not proven particularly popular among drafters. Nevertheless, we press on, because the feature is powerful when used correctly.

USING ANNOTATIVE SCALING

Annotative scaling is only available to drawings when it is turned on; it is not on by default. We turn it on by applying annotatively-scaled styles to entities. (Annotative scaling does not apply to layers.)

Some new drawings may contain text styles and other entity styles named "Annotative," in addition to the default "Standard" non-annotative styles. To make a style annotative, typically we simply turn on the Annotative check box in the style's definition dialog box.

The figure below shows the check box in an attribute definition dialog box.

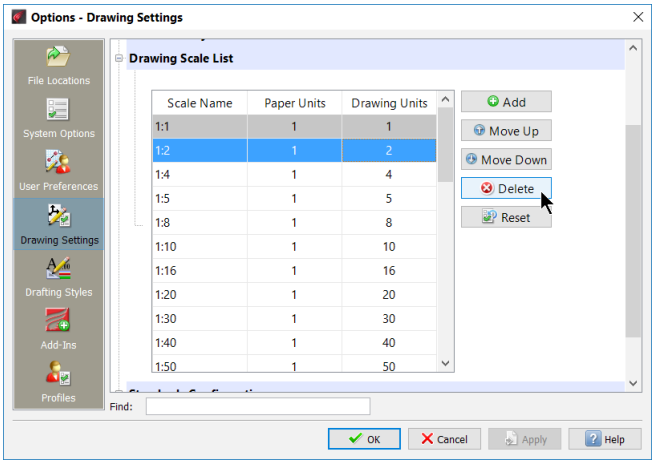


Noticing the Annotative Scaling check box

Entities affected by annotative scaling show an A icon when we pass the cursor over them. Entities can have multiple scale factors, one for each size of sheet on which you intend to plot the drawing. So when an entity shows a double-A instead, this means it has more than one annotative scale applied.

Text that is annotatively-scaled being flagged by the A icon

Annotative scale factors are limited to ones specified by the program; you can't just type one in. Fortunately, the list of available scale factors is quite long, and so you have many to choose from. Should you find the list too long, use the **EditScaleList** command to remove the ones you never use (this reduces complexity) and adding one you do use (this adds flexibility.) When it comes time to assign an annotative scale factor, you choose one from the list.



Editing the list of scale factors

When annotative scaling is turned on, commands for placing text and other entities change their wording for the height prompt. The word "paper" is added to indicate that the text height will be scaled automatically by the program:

```
Specify paper height <0.2000>:
```

We enter the height at which we wish the text to appear when it is plotted on paper. If we specify 1/8" (0.125 units), then the program will plot the text 1/8" tall; it calculates the correct display size for us. The

need to figure out the scale factor of text and other annotative entities is eliminated. This is the beauty of annotative scaling.

Annotative Scaling Tutorial

As there are several nuances involved with annotative scaling, please follow this tutorial to better understand how it works.

1. In the program, open a drawing.
2. Switch to layout mode by clicking the **Sheet1** tab.
3. Enter model space by double-clicking inside the viewport border. On the status bar, look for the tools that work with annotative scaling. These appear at the right end, just before the coordinates.

Annotative scaling controls on the status bar

SUMMARY OF ANNOTATIVE VARIABLES

The **AnnoAllVisible** toggles the display of annotatively-scaled entities showing all (1) or showing only those that match the current annotation scale (0); default = 1.

The **AnnoAutoScale** determines which annotative entities are assigned newly set annotation scales; default = -4. When the value is negative, the setting is turned off but the setting is remembered:

AnnoAutoScale	Meaning
1 *or* -1	All Except entities on layers that are turned off, frozen, *locked*, or are in frozen viewports
2 *or* -2	All Except entities on layers that are turned off, frozen, or are in frozen viewports
3 *or* -3	All Except for entities on layers that are locked
4 *or* -4	All annotative entities

AnnotativeDwg variable toggles whether (1) or not (0) the current drawing acts as annotatively-scaled block when inserted into another drawing; default = 0.

CAnnoScale variable sets the current annotation scale for the active workspace; must be one of the scales listed by the EditScaleList command.

CAnnoScaleValue variable reports the value of the current annotation scale.

DimAnno variable reports whether the current dimension style is annotative (1) or not (0); default = 0.

HpAnnotative variable controls whether new hatched areas are annotatively-scaled (1) or not (0); default = 0.

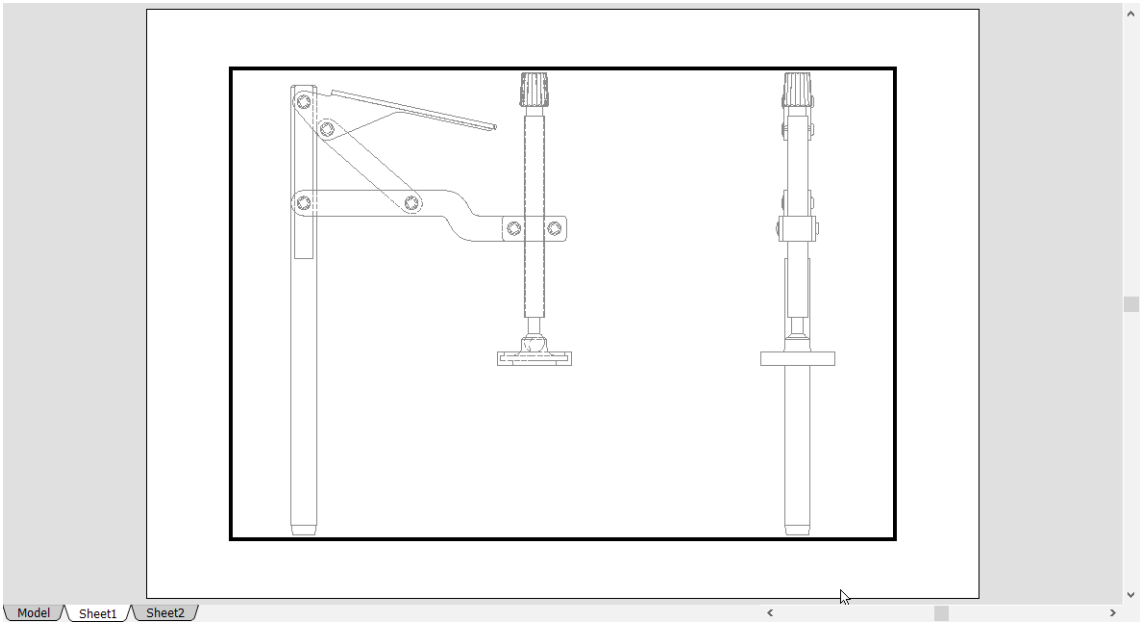
MsLtScale variable determines whether linestyles in Model tab or in layout viewports are scaled annotatively (1) or not (0); default = 1.

SaveFidelity variable determines whether annotative entities are saved in 2007 and earlier DWG formats without annotative scaling (0) or saved to separate layers for each annotative scale used (1); default = 1.

SelectionAnnoDisplay variable specifies whether selected entities show all annotative scalings dimmed (1) or regular (0); default = 1.The XFadeCtl variable determines the level of dimming.

4. When first opened in a viewport, drawings typically are not displayed at a normal scale. This is because the program fits the model space extents to the area of the viewport.

   In the drawing I am using for this tutorial, the scale happens to be 0.13225 (or 7.5614, expressed as a scale factor).



Entities from model space precisely fitting the viewport border

For this reason, we must resize drawings in layouts in that manner that meets these criteria:

» The whole drawing must fit inside the viewport, and usually is smaller than the viewport
» The drawing must be scaled to a standard factor

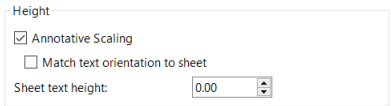
To select an appropriate scale factor, first use the **Zoom Extents** command to fit the drawing to the viewport. This gives you the largest scale permissible.

5. Notice the reported scale. On my drawing, it reads 0.13225. This is the *scale*; but you need to work out the scale *factor*. So, get out your calculator, and then divide 1 by the scale. In my case, I divided 1 by 0.13225. The answer is 7.5614.

   For the scale factor, I pick the next largest whole number, 8, because (1) scale factors are usually whole numbers, and (2) the next largest number makes the drawing slightly smaller in the viewport. We can't have some of the drawing hidden by the edges of the viewport!

TIP If the drawing does not contain an annotative text style, then create one, like this:

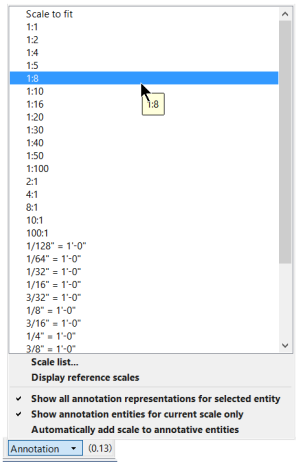
1. Enter the **TextStyle** command to modify text styles.
2. Make a copy of an existing style by clicking the **New** button.
3. Check the **Annotative Scaling** option. It's that simple!



4. Give the style a name, such as "Annotative," and then save it.

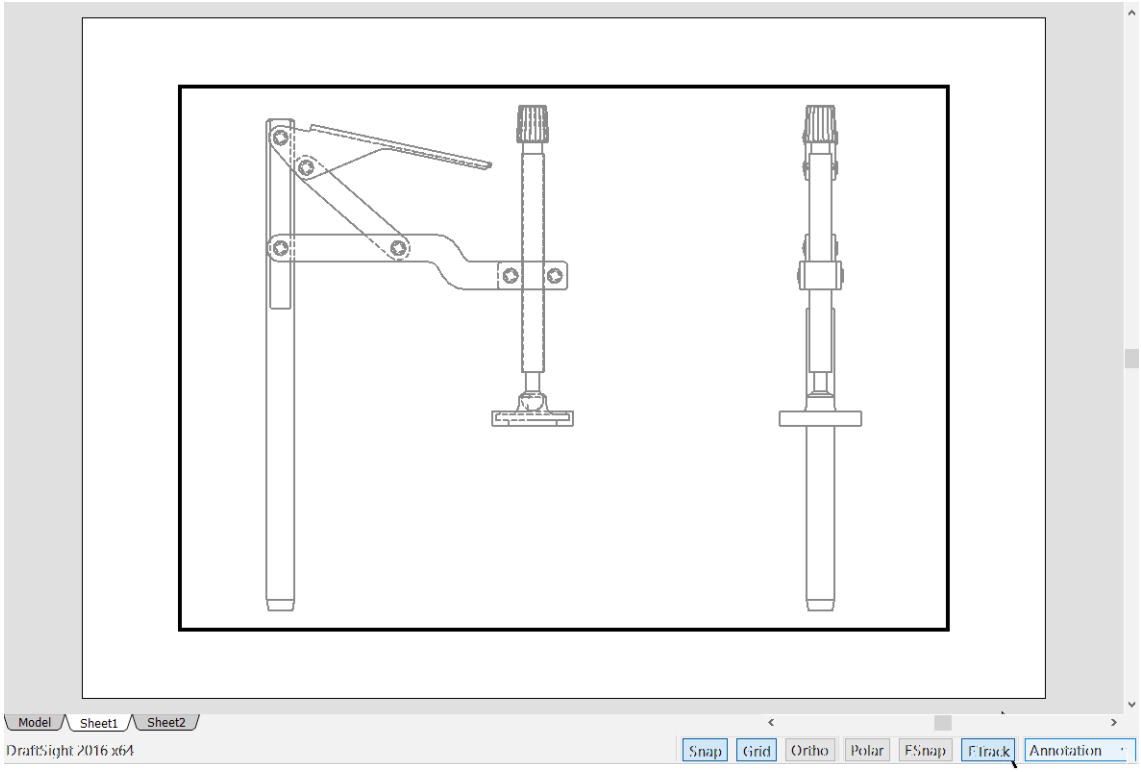
---

6. Click the annotative scale list, and then select a factor of the nearest value that results in a smaller looking drawing. In my drawing, the nearest standard scale factor to 8 is 1:8.



Picking the closest scale factor

You know that you have picked the correct scale when the drawing becomes slightly smaller, but not too much smaller. Notice also that the reported scale matches the annotation factor you selected.



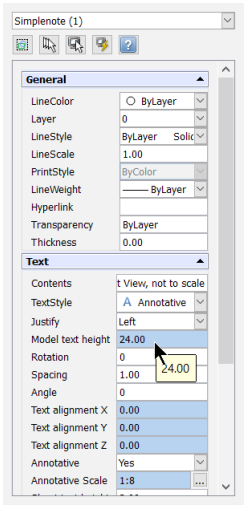
Status bar reporting the scale factor of 1:8

7. With the annotative scale factor set up, let's test the annotation feature. Place two kinds of text in Model tab, standard and annotative.

   Why place the text in model tab? (Why not in the sheet tab?) My AutoCAD-expert friend Bill Fane explains, "If you zoom and pan while entering text in a sheet 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."

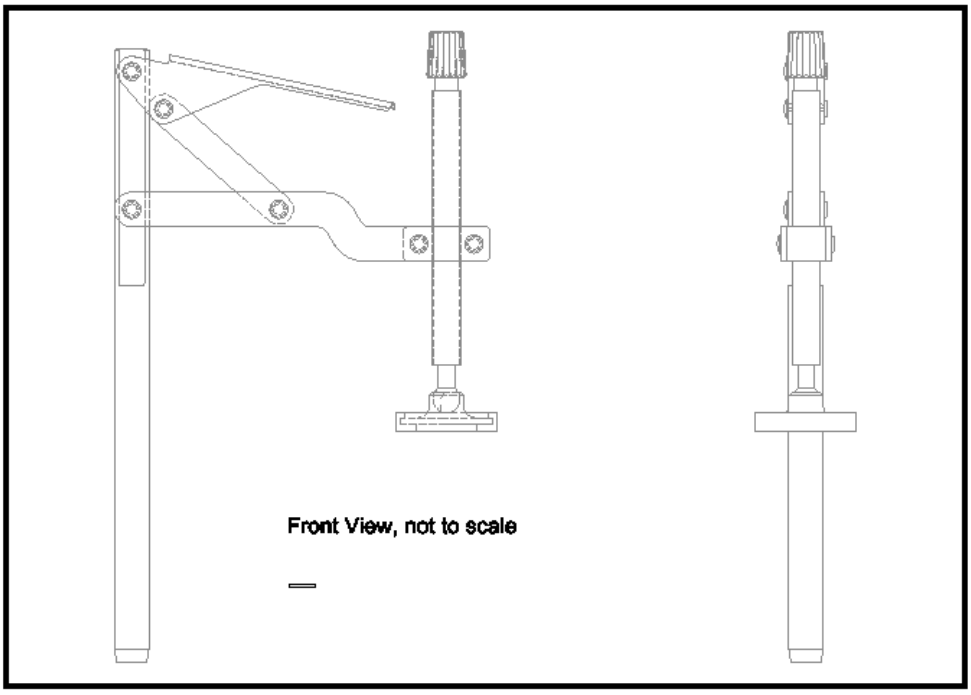
Click on the **Model** tab, and then place text with a standard (non-annotative) style at a height of 1/8" or 3mm. Notice that the text looks very tiny.

8. Change the text style to an annotative one, and again place text with a height of 1/8" or 3mm.
9. Notice that the text looks a reasonable size. To learn why, open the Properties palette, and then choose the annotatively-scaled text. Notice that the height is reported as 24 units — exactly 8x larger than the 3mm we specified.



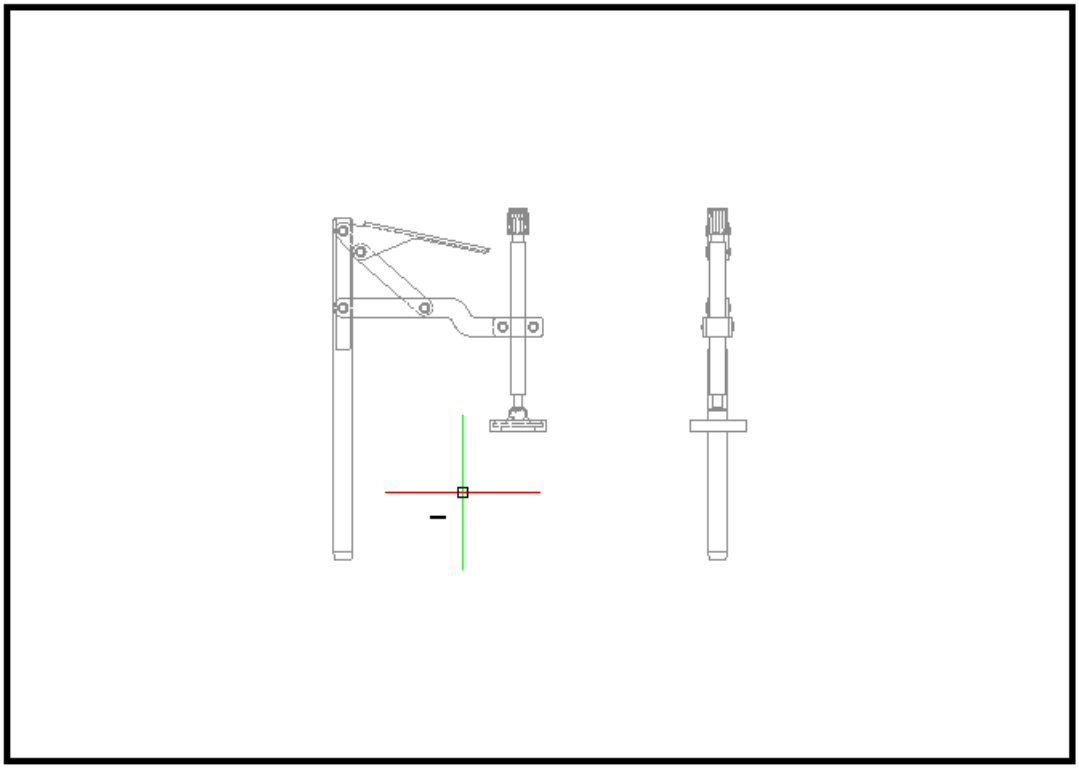
Properties palette reporting the model height of annotatively-scaled text

10. Any text you place now 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 **Sheet1**. The tiny rectangle visible underneath "Front View, not to scale" is what the text looks like when it is not annotatively-scaled.



Examples of text with and without annotative scaling

11. Remember earlier I said that we only see annotative text when its scale factor matches that of the viewport. To see this effect, change the scale of the viewport to something like 1:16. Notice that the text disappears; it should be where the crosshair cursor is located in the figure below.



Annotatively-scaled text disappearing after the viewport scale changes

Second Scale, Second Sheet

Should you need to plot the drawing at a second scale, then I recommend you set up a second layout (sheet) tab, set the new annotation scale, and then go back to model tab. Select the text and other annotations needing scaling, and then apply the new annotation scale factor. See Chapter 18 for a tutorial on using a second annotative scale factor in a drawing.

TIP The annotative scale factor system works differently for line styles. To ensure their pattern is the right size, you turn on the **MsLtScale** system variable.

Facilities Management

Once the design of a floor plan is complete, managers can extract data to keep track of the facilities. This is known as *facilities management*. It allows corporations and agencies to track spaces and inventory, such as offices, desks, computers, and entire buildings.

The program has the ability to include text data with blocks. This can be any kind of useful data, but in practice tends to be descriptions of items. For instance, a desk block might be assigned its purchase price, model number, room number, depreciation schedule, and so on. This data can be extracted to a spreadsheet, creating a list of all desks in the drawing or building. The spreadsheet can list all desks by room number, it can add up the total number of desks and the total purchase cost, and so on.

The data attached to blocks are called "attributes." Attributes cannot, unfortunately, be attached to any other kind of entity, such as circles or polylines — just blocks. The workaround is to turn those entities into blocks.

Working with attributes in the program involves these stages:

Stage i: **Define each attributes** with a name and a default value
Stage ii: **Attach the attributes** to blocks; each block can have up to 255 attributes
Stage iii: **Extract attributes** to spreadsheets and other data files

All attributes consists of two parts:

- An identifying *name* (a.k.a. *tag*), such as "Price," which allows the program, spreadsheet software, and you to identify values
- A *value*, such as the tag "Price" having the value "199.00"

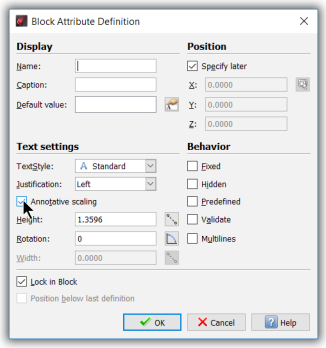
STAGE 1: DEFINING ATTRIBUTES

Command: MakeBlockAttribute
Aliases: att, attdef, mblkatt
Menu bar: Draw | Block | Define Attributes
Ribbon: Insert | Block Definition | Define Block Attributes (Drafting and Annotation)
Toolbar: 

Attributes are defined with the **MakeBlockAttribute** command. It can become tedious working with many attributes, so for this tutorial you attach just two attributes to the chair block: its purchase price (how much it cost), and its owner (who sits in it).

Follow these steps:

1. Start the **MakeBlockAttribute** command. Notice that the Block Attribute Definition dialog box has many options, but fortunately you need to deal with just a few of them.



Block Attribute Definition dialog box

2. Attributes are created one by one. The first one is Purchase Price, which you create like this:
   a. In the **Name** field, enter:

      ```
      PurchasePrice
      ```

      Remember: the *name* is the identifier for this attribute.

SUMMARY OF MAKEBLOCKATTRIBUTE DIALOG BOX OPTIONS

```
: MakeBlockAttribute
(Displays the Block Attribute Definition dialog box)
```

DISPLAY OPTIONS

Name (or Tag) identifies the attribute; use letters, numbers, blanks, underscores, and dollar signs.

Caption (or Prompt) displays a prompt during block insertion; maximum of 256 characters.

Default Value specifies the default value for the attribute.

Field button opens the Field dialog box.

INSERTION POINT OPTIONS

Specify later prompts you for the insertion point after you close this dialog box.

Select in Graphics Area button pick the insertion point in the drawing.

X, **Y**, and **Z** specify the coordinates of the insertion point.

TEXT SETTINGS OPTIONS

TextStyle specifies the style for the attribute text; text styles are created and modified by the TextStyle command.

Justification specifies the justification mode.

Height specifies the height of the text.

Select in Graphics Area button picks two points in the drawing.

Rotation specifies the rotation angle.

Select in Graphics Area button picks two points in the drawing.

Annotative Scaling scales the text to automatically match the scale factor of sheets (layouts).

BEHAVIOR OPTIONS

Fixed gives the attribute the constant value defined by Default Value; the user cannot enter a different value.

Hidden hides the attribute text; the user can make hidden attributes visible with the DisplayBlockAttributes command.

Predefined uses the default value automatically; you are not prompted for a value when the block is being inserted. Values can be edited later using the EditBlockAttribute command.

Validate prompts you to verify that the value is correct.

OTHER OPTIONS

Lock in Block prevents the attributes from being moved away from the block reference.

Position Below Last Definition places the next attribute definition below the previous one automatically.

b. In the **Caption** field, enter a prompt:

```
Purchase price of chair
```

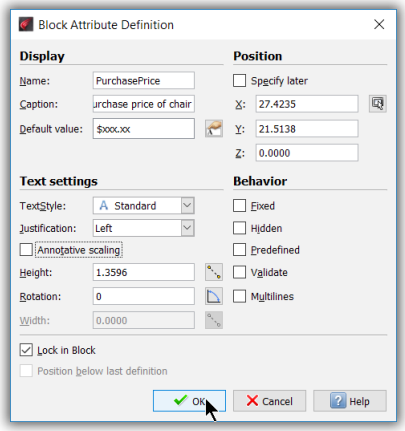
The caption appears later in the Command window to remind users of the purpose of this attribute.

c. In the **Default Value** field, enter:

```
$xxx.xx
```

The default contains either a common value or a format hint. Above, the $xxx.xx hints at the format, in dollars.

The contents of the dialog box should look like this:



Filling in parameters for defining the attribute

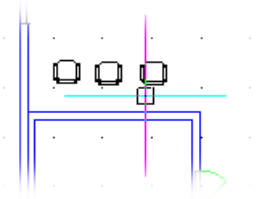
3. Attribute data is located typically near the block:

a. Ensure that the **Specify Later** option is turned off.

b. Click the **Specify Position** button.

c. Choose a point near a chair block in the drawing, as prompted:

```
Options: press ESC to return to dialog or
Specify position» (Pick a point near a chair block)
```



Placing the attribute near the block

4. As soon as you pick a point, the dialog box comes back. You can ignore all other options in the dialog box, except for this one: keep **Lock in Block** turned on, for it prevents users from moving the attribute away from the block. See figure above of how the dialog box should look before you click OK.

5. Click **OK**. Notice that the attribute text appears near the block.

Attribute definition next to chair block

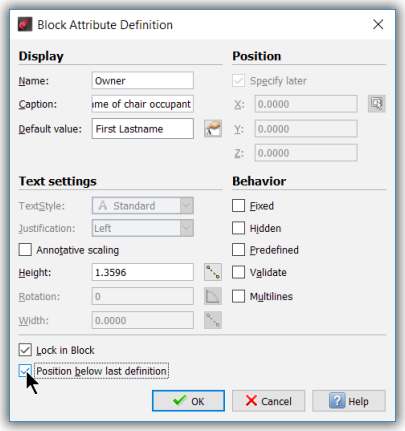
6. To add the second attribute, press the **spacebar**. This action repeats the MakeAttributeBlock command and brings back the dialog box.

7. Enter the following parameters for the second attribute:

Display Setting	Value
Name	Owner
Caption	Name of chair occupant
Default Value	First LastName

8. The program can place the new attribute definition below the first one automatically: turn on the **Position Below Last Position** option.

   The Block Attribute Definition dialog box should look like this:



Settings for placing the second attribute definition

9. After you click **OK**, the second attribute appears:

Both attribute definitions placed next to chair block

TIP The **ImportBlockAttributes** command imports attribute values in blocks, while the **RedefineBlockAttribute** command redefines attributes in blocks. Both of these commands can be found in the XtraTools menu and ribbon tab.

STAGE 2: ADDING ATTRIBUTES TO BLOCKS

The text that you see is an *attribute definition*; the "PurchasePrice" and "Owner" text is not the attribute data, which you define here in stage 2. Attribute data are things like $199.00. This concept of attribute definitions and data is one of the trickier ones in the program, but I think you will understand once you complete these tutorials.

So far, you have placed the attribute definitions in the drawing near the chair block. The block and the adjacent attribute definitions are only "geographically convenient," as Bart Simpson once said of his next-door-neighbour friend.

Therefore, in this part of the tutorial, you attach the attribute definitions to the chair block. By redefining the block, the attributes become part of the block, you specify the attribute data, and then the data is displayed in the drawing.

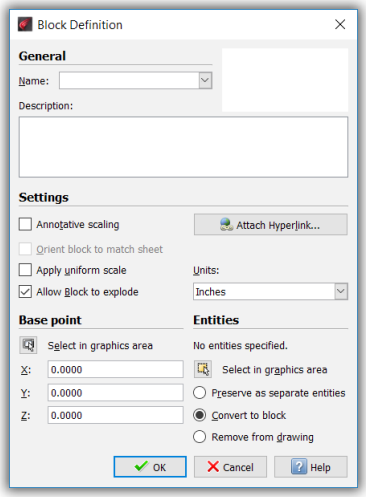
1. To redefine the Office Chair block, you must first explode one of its insertions. Start the **Explode** command, and then choose the block next to the attribute definitions:

```
: explode
Specify entities» (Select the chair block)
Specify entities» (Press Enter to end the command)
```

The chair looks no different, except that it is no longer a block. (You could move each line and arc individually — but don't. You'll just make a mess of it.)

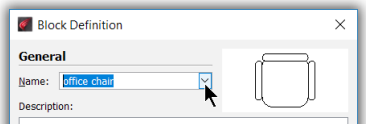
TIP This procedure **also** applies to entities that are not already blocks. The only change being that you don't need to explode them, because they are not blocks yet; all of the following steps apply.

2. With the block reduced to its constituent parts, you can now redefine the Office Chair:
   a. Start the **MakeBlock** command. Notice the Block Definition dialog box.



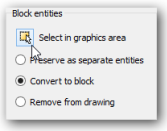
The Block Definition dialog box

   b. In the **Name** droplist, choose "Office Chair." By doing this, you redefine the block. You are redefining the block so that you can add the two attributes to it.



First step in redefining a block: choosing it

   c. In the Block Entities section, click the **Select In Graphics Area** button.



Preparing to select entities in the drawing

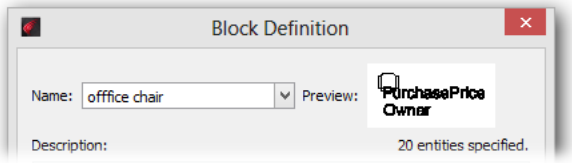
d. When the dialog box disappears, *window*-select the two attribute definitions and all the entities making up the chair. Don't miss any: there are 20 parts to the chair!

```
Select entities: (Select all attributes and all the entities of the exploded
chair)
20 found, 20 total
```

e. And then press **Enter** to return to the dialog box.

```
Select entities: (Press Enter to return to dialog box)
```

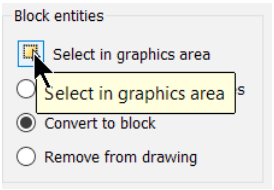
Notice that the preview window now shows the chair entities and the attribute definitions.



Preview showing block and attribute definitions

TIP The order in which you select the attributes determines the order in which the prompts are displayed later while inserting the block.

f. In the Base Point section of the dialog box, again click the **Select In Graphics Area** button.

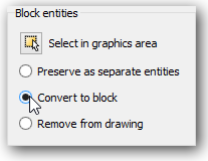


Choosing the Select in Graphics Area button

The dialog box disappears, allowing you to choose a point on the chair:

```
Specify insertion base point: (Pick a logical point, such as the center of
the chair)
```

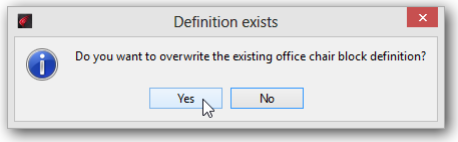
g. As soon as you pick a point, the dialog box returns. Ensure the **Block Entities** option is set to "Convert to Block." This ensures that all existing blocks are replaced by the new definition.



Choosing Convert to Block

3. Click **OK** to affect the redefinition.
4. Notice that the program warns that you are engaging in a dangerous enterprise, the redefinition of existing blocks. The program warns you, because the changes you made to the

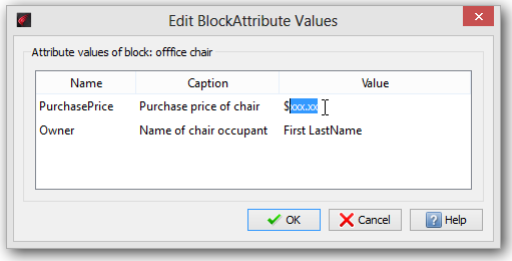
Office Chair definition will affect all office chair blocks in the drawing.



Agreeing to overwrite the old block definition

Click **Yes**. After all, if you make a mistake, there always is the Undo command.

5. Now comes the tedious part: entering the Purchase Price and Owner data for each one of the eighteen chairs. In practice, this is a job for which people are paid; for this tutorial, you need only enter data for two chairs.



Entering the values into the attribute database

TIP If the dialog box does not appear, then you will find the prompts in the Command bar. The dialog box is more convenient, and so to force it to appear, change the value of system variable **AttDia** to 1.

In the Edit Block Attribute Values dialog box, change the default values to real-world value:

Name	Value
PurchasePrice	$249.99
Owner	John Smith

Notice that one block now displays this attribute *data*.

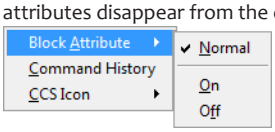
6. Repeat for at least one other chair block:

Name	Value
PurchasePrice	$199.99
Owner	Mary Doe

7. Save your work.

TIPS Don't worry if attribute data spill over on top of other entities. Since attributes are not meant to be seen in drawings, you can easily turn off their display through the **DisplayBlockAttribute** command, or through the menu:

1. From the **View** menu, choose **Display**, and then choose **Block Attribute**.
2. Choose **Off**. Notice that the attributes disappear from the drawing.



The **On** and **Off** options are straightforward; the **Normal** option treats the display of attributes as defined by the Block Attribute Definition dialog box: one of its options is Hidden, which means the attribute is invisible by default. The Normal option displays only non-hidden attributes, while the On option displays all, including those marked as hidden.

When set to 1, the **AttDia** system variable causes the Insert command to display dialog boxes for inserting blocks and entering attribute values; when 0, Insert displays all prompts in the Command window.

Editing Attribute Data

Command: EditBlockAttribute
Aliases: ate, attedit, attxedit, ddatte, edblkatt
Menu bar: Modify | Entity | Block Attribute | Single
Ribbon: Insert | Block | Attribute Block | Single (Drafting and Annotation)
Toolbar:

Should you need to change the values of attributes, you can do so with the Edit Block Attribute Values dialog box.

STAGE 3: EXTRACTING ATTRIBUTES

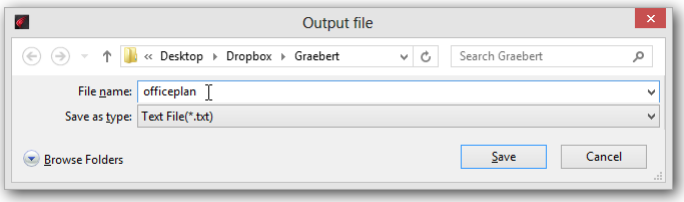
Command: BlockAttributeOutput
Aliases: attout, battout
Menu: Tools | Attribute Export
Ribbon: Insert | Block Definition | Export Attributes (Drafting and Annotation)
Toolbar: --

Attributes are of limited use in drawings; they can be viewed but not analyzed. To do so, extract data from the drawing and place them in a spreadsheet: (i) export attributes to a text file, and then (ii) open the text file in a program. Straightforward; no template needed! Follow these steps:

1. Enter the **BlockAttributeOutput** command.
2. Choose all entities in the drawing, because the program will filter out non-blocks automatically:

```
Specify entities: all
Specify entities: (Press Enter to end entity selection)
```

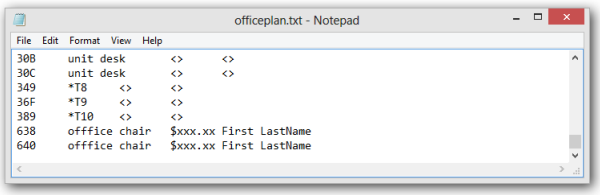
3. In the Output File dialog box, give a name to the file, such as *officeplan.txt*.



Specifying the output file name for exporting attribute data

4. Click **Save**. In a flash, the program extracts all of the attribute data, and then stuffs it into the text file. The file is an ordinary text file, which you can view with Notepad in Windows, or TextEdit on Mac or GEdit with Linux.

   For the technically-inclined, the file is in *tab-delimited format*, and looks like this:



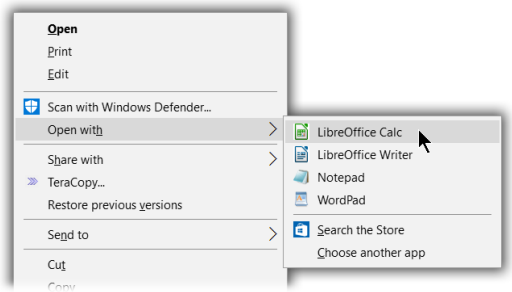
Tab-delimited data being separated by tabs

Exporting Data to Spreadsheets

The best way to manipulate data is with a spreadsheet program, like Excel. It can do things like count items and add up numbers. If you do not have one, you can download the free LibreOffice program (that includes the Calc spreadsheet program) from http://www.libreoffice.org. It is available for Windows, Mac, and Linux.

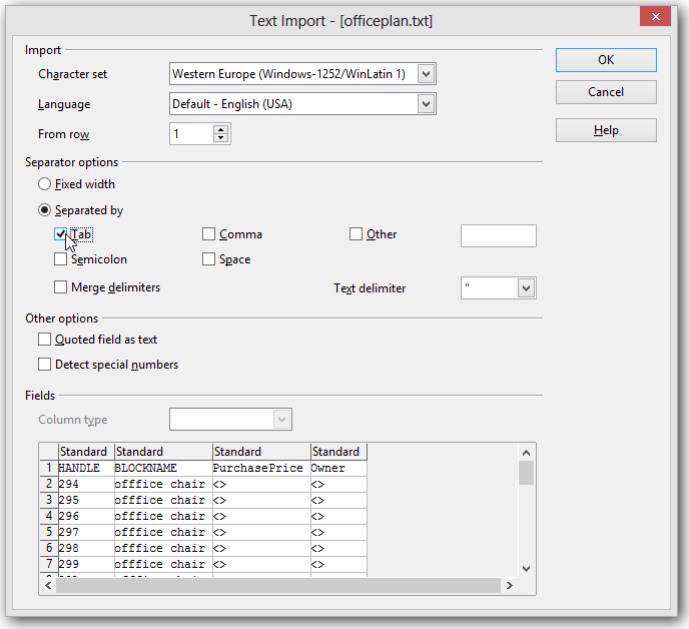
To open the data file in a spreadsheet, follow these steps:

1. Go to the folder that contains the *officeplan.txt* file.
2. Right-click the file, and then from the shortcut menu choose **Open With > LibreOffice Calc**.



Opening the text file with Libre Calc

3. Notice Libre Calc opens and then displays the Text Import dialog box.

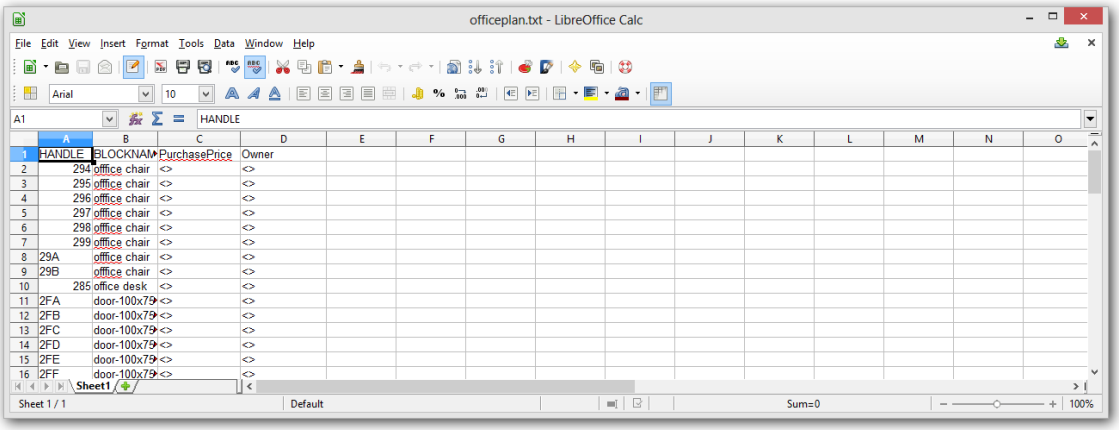


Selecting the Tab format while importing attribute data into LibreCalc

The purpose of this dialog box is to specify the format of text files being imported, so that columns of text are correctly separated into columns of cells. The defaults are Comma, Tab, and Semi-colon.

Our data file uses tabs to separate columns. Ensure **Tab** is checked on. Notice that the dialog box's preview has the text jumped into columns.

4. Click **OK**. The data is entered into the spreadsheet automatically.



Attribute data displayed in a spreadsheet

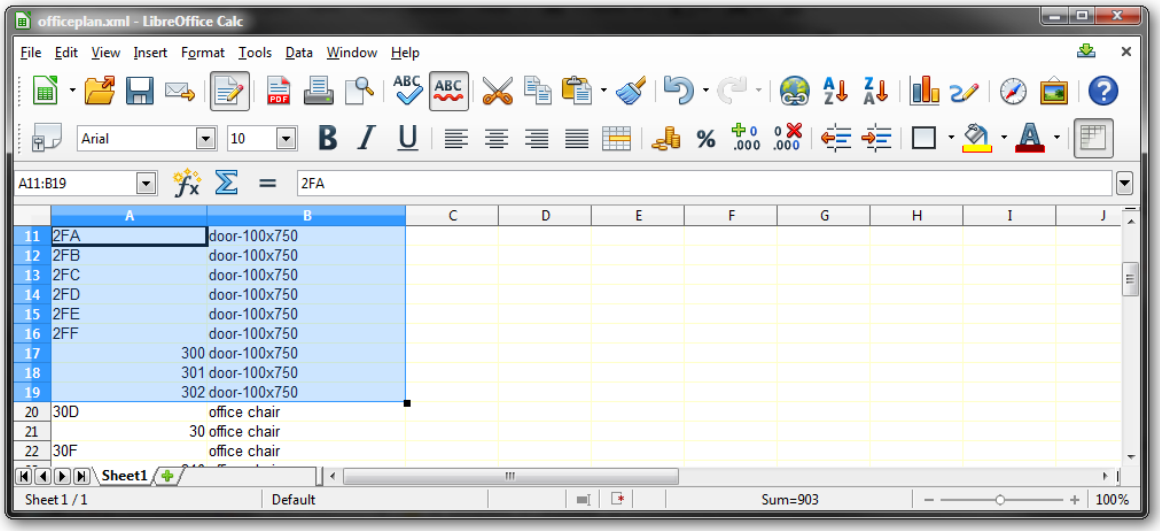
5 You can now apply spreadsheet functions, such as counting, adding up, and multiplying.

Importing Spreadsheet Data as Tables

Command: Table
Aliases: --
Menu: Draw | Table
Ribbon: Annotate | Tables | Table (Drafting and Annotation)
Toolbar: --

The attribute data does not have to stay in the spreadsheet: you can bring it back into the drawing as a bill of materials. Bills of materials are displayed in tables, and so here is how to put the spreadsheet data into a CAD table:

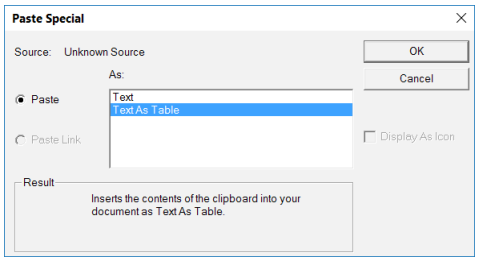
1. In the spreadsheet, highlight the cells to be copied.



Selecting cells in the spreadsheet

2. Press **Ctrl+c** (Cmd+c on Macs) to copy data from the spreadsheet to the clipboard.
3. Switch to the program, and then to paste the data into the drawing using **Edit | Paste Paste Selected Format** from the menu (or enter the PasteSelected command).

4. In the dialog box, choose "Text as Table," and then click **OK**.

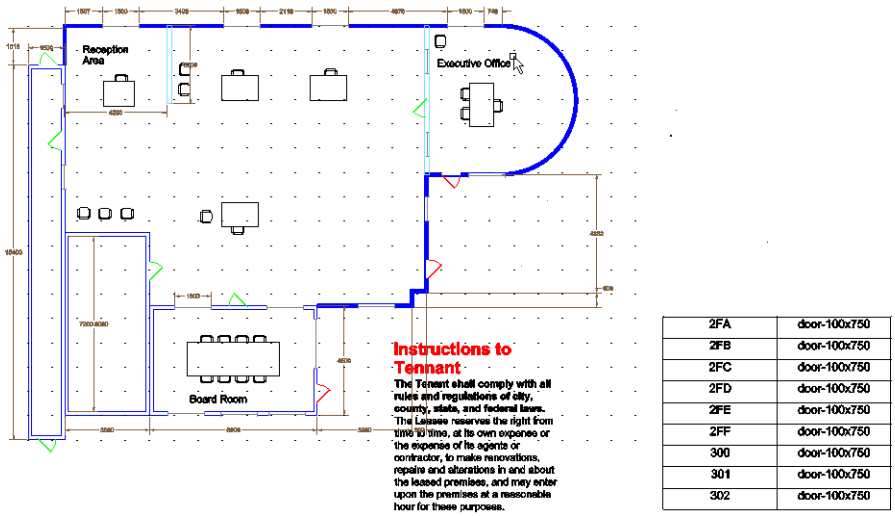


Choosing the format in which to paste the text

5. Notice that the program prompts you for the location and size of the table:

```
Specify position» (Pick a point for the table's upper left corner)
Specify opposite corner» (Pick a second point to indicate the size of the table)
```

The table appears in the drawing.



Spreadsheet cells inserted as a table in the drawing

EDITING TABLES

Command: EditTable
Alias: TablEdit
Menu: Modify | Entity | Table
Ribbon: Annotate | Tables | Table Edit (Drafting and Annotation)
Toolbar: --

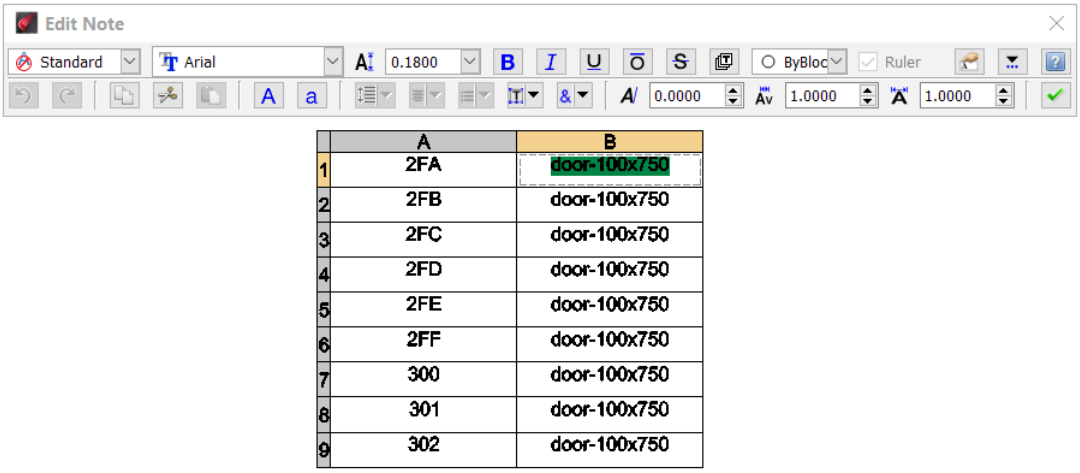
Tables are not static; all aspects of them can be edited, both the look of the lines and the content and size of the cells. The look of cells is changed by editing them directly; the content of cells is edited with the TablEdit command.

Editing Cell Text

To edit text in a cell, enter the EditTable command, and then choose a cell, as follows:

```
: edittable
Specify table cell» (Pick a cell)
```

Notice that the cell is highlighted and that the Edit Note toolbar appears.



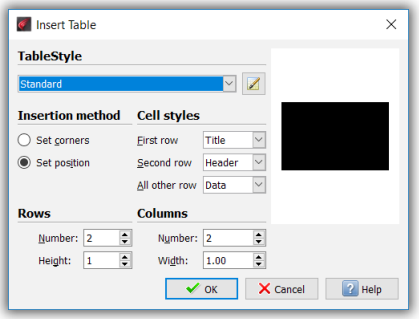
	A	B
1	2FA	door-100x750
2	2FB	door-100x750
3	2FC	door-100x750
4	2FD	door-100x750
5	2FE	door-100x750
6	2FF	door-100x750
7	300	door-100x750
8	301	door-100x750
9	302	door-100x750

Editing text in a cell

To edit another cell, press the **Tab** key to reach it; pressing **Shift+Tab** moves the focus upwards among the cells. When done editing, click the  button in the toolbar.

SUMMARY OF TABLE COMMAND OPTIONS

```
: table
```



TableStyle droplist selects a style that determines the look of the table (specified by the TableStyle command)

Insertion method specifies how the table is to be placed in the drawing:

- **Set corners** prompts you to specify the two opposite corners to size the table
- **Set position** prompts you to pick a point for the upper left corner; overall size is determined by the Height option

Cell Styles specifies the style applied separately to the first row (Title style), the second row (Header style), and all other rows (Data style)

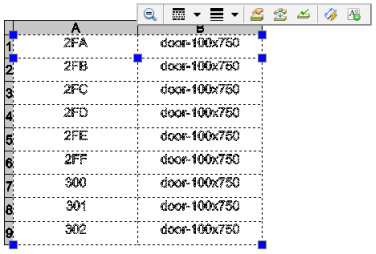
Rows specifies the initial number of rows and their default height

Columns specifies the initial number of columns and their default width

Editing the Look of Cells and Tables

Editing cells and tables involves no commands; just a steady hand to click the right spot:

Edit a table — click one of the lines making up the border. Once a table is selected, blue grips appear. They change the size of the table and of the individual columns.

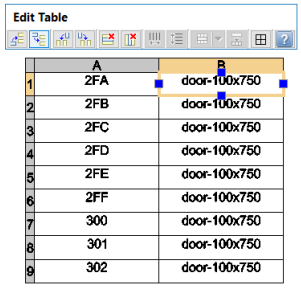


	A	B
1	2FA	door-100x750
2	2FB	door-100x750
3	2FC	door-100x750
4	2FD	door-100x750
5	2FE	door-100x750
6	2FF	door-100x750
7	300	door-100x750
8	301	door-100x750
9	302	door-100x750

Picking a border to edit the table

TIP The **TableIndicator** system variable toggles the headers that show the row numbers and column letters when you edit cells in-place; 0 = off; 1 = on, default.

Edit a cell — click a blank part of the cell, touching neither the text nor the lines. Once a cell is selected, it also gains blue grips, but also a formatting toolbar.



	A	B
1	2FA	door-100x750
2	2FB	door-100x750
3	2FC	door-100x750
4	2FD	door-100x750
5	2FE	door-100x750
6	2FF	door-100x750
7	300	door-100x750
8	301	door-100x750
9	302	door-100x750

Picking emptiness to edit the cell

To work with more than one cell at a time, hold down the **Shift** key and then pick a blank spot in any other cell. Here is what the buttons on the toolbar mean:

Toolbar for editing the selected cell

Insert Row above or below or **Insert Column** to the left or right of the current cell

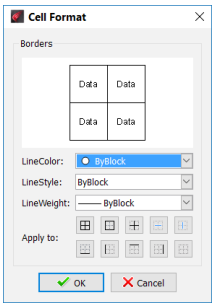
Remove Row (or **Column**) associated with the selected cell(s)

Size Rows (or **Columns**) **Equally** associated with the selected cells; at least two cells must be selected for these commands to operate

Merge Cells horizontally or vertically or all selected cells. At least two cells must be selected for this command to operate; the content of the first selected cell is retained, but the content of all other merged cells is erased.

Unmerge Cells returns rows and columns, based on the table indicator's size and number of rows and columns; content erased from cells from a previous Merge Cell operation is not recovered.

Cell Format displays the Cell Format dialog box for changing the color, style, weight, and visibility of the lines surrounding cells:



The Cell Format dialog box

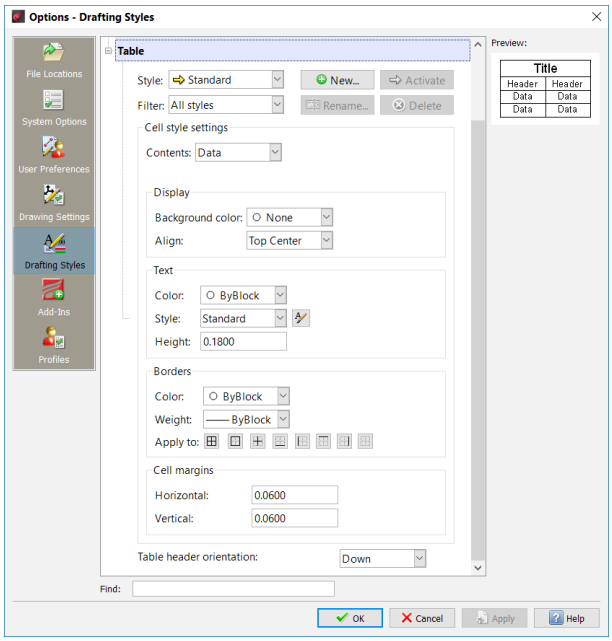
TIPS To edit other cells, use the cursor keys for navigation.

The **TableToolbar** system variable toggles the Edit Table toolbar: 0 = off; 1 = on (default).

To create table styles, use the **TableStyle** command.

SUMMARY OF TABLESTYLE COMMAND OPTIONS

: `tablestyle`



Style lists names of table styles stored in the drawing

Filter displays just the table styles used by the drawing

Cell Style Settings specifies the look of cells:

- **Data** is the default formatting for all cells
- **Head** is the formatting of cells used for header rows, typically one row below the title
- **Title** is the formatting for the title row, typically the first row of the table

Tutorial: Adding and Formatting Rows

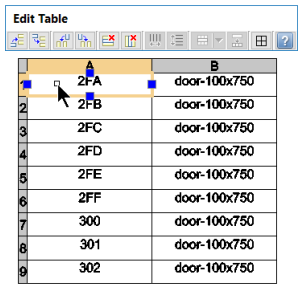
In this tutorial, you add rows to the top of a table, and then format them as title and header rows.

1. Open the *spreadsheet.dwg* file. Notice that it contains a two-column table with no header rows.

2FA	door-100x750
2FB	door-100x750
2FC	door-100x750
2FD	door-100x750
2FE	door-100x750
2FF	door-100x750
300	door-100x750
301	door-100x750
302	door-100x750

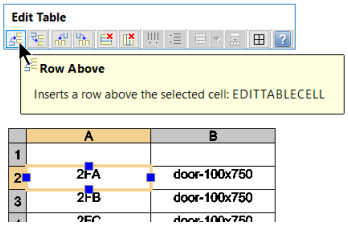
Example table open in the drawing

2. Click inside one of the cells in the top row, avoiding border lines and text. Notice that the cell is highlighted in orange, that blue grips appear, and that the Edit Table toolbar is displayed.



Inserting a new row to the top of the table

3. In the toolbar, click the **Insert Row Above** button. Notice that the program adds a blank row.

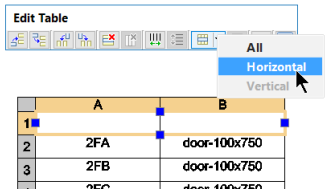


Blank row added above the selected cell

4. Select both of the two new cells that make up the blank row, and then merge them into a single cell. Follow these steps:

   a. Click one new cell.

   b. Hold down the **Shift** key and then pick the other new cell.

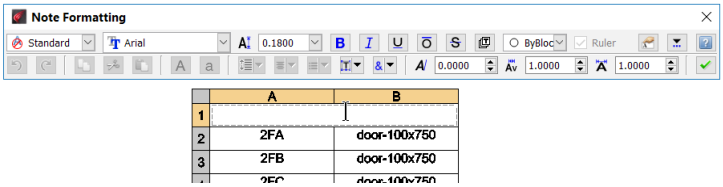


Merging two cells

   c. From the Edit Table toolbar, click the **Merge Cells** button.

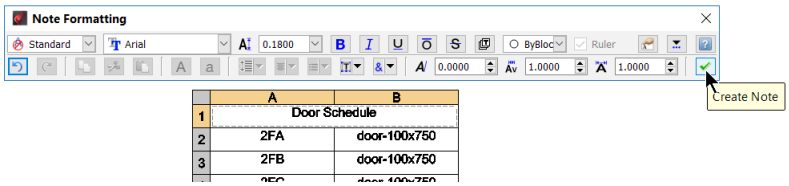
d. From the droplist, choose **Horizontal** to merge the two cells horizontally. Notice that one cell spans the width of the table.

5. To add text to the cell, double-click it. Notice the Note Formatting toolbar.



Adding text to the merged cells

6. Enter text, such as "Door Schedule," and then click the **Create Note** button.



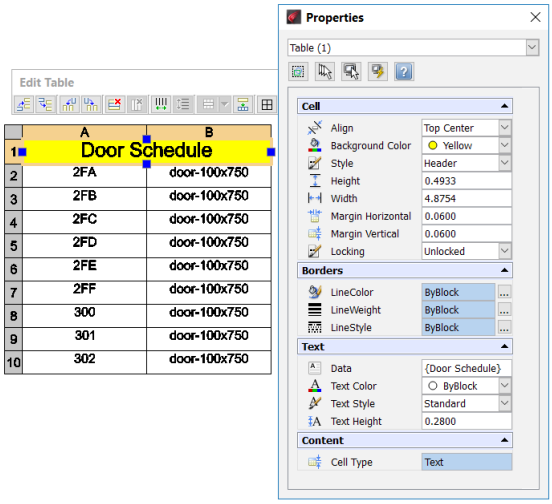
Creating the note

7. Here you use table styles to format the cell to make it look more like a title. Follow these steps:

   a. Ensure the Properties palette is visible; if not, then press **Ctrl+1** (Cmd+1 on Macs).

   b. Choose the topmost cell.

   c. In the Cell section of the Properties palette, change the color and height of the text.



Modifying the look of cells through the Properties palette

8. Press **Esc** to exit table editing.

Tutorial: Performing Calculations and Formatting Rows

The program can perform simple calculations on the content of tables. You can place formulas in table cells. Specifically, these are the four arithmetic operators (+, -, /, *), and the Sum, Average, and Count functions.

To start a formula, enter = (equals) and then the cell references with the operator. The figure shows a formula being added to cell C4:

	A	B	C
1	Table of Operators		
2	2	4	6
3	3	5	15
4	14	4	=average(A2:B3)

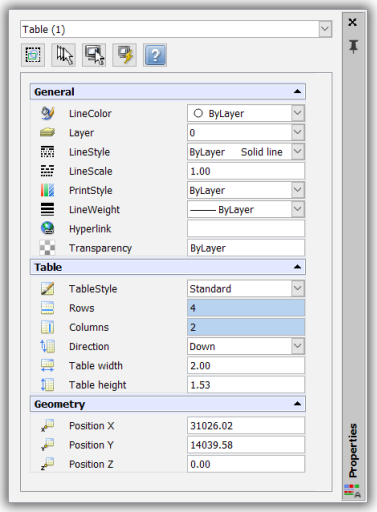
Adding a formula to a table cells

Here is a reference guide to all of the functions the program supports, along with examples of how they are used. Refer to the figure above for the examples, equivalents, and results.

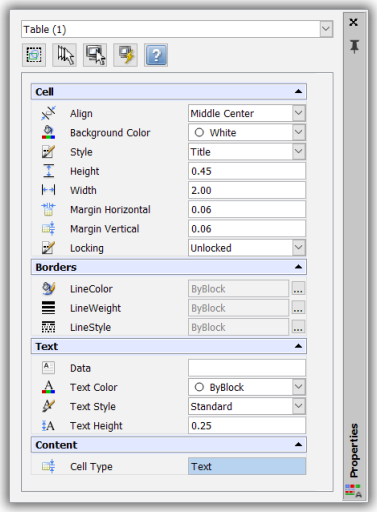
Formula	Function	Example	Equivalent & Result
+	Adds the contents of two or more cells	=A2+B2	2 + 4 = 6
-	Subtracts the content of one cell from another cell	=A2-B2	2 - 4 = -2
/	Divides the content of one cell into another cell	=A2/B2	2 / 4 = 0.5
*	Multiplies the contents of two or more cells	=A2*B2	2 * 4 = 8
Average	Finds the average of a range of cells	=average(A2:B4)	3.5
Count	Counts the number of cells in the specified range	=sum(A2:B4)	4
Sum	Adds the content of a range of cells	=sum(A2:B4)	14

TIPS To create tables from scratch, use the **Table** command. At this time, text is the only content that can be placed in cells.

To change the look of the table and the content of its cells, you can edit it table directly or else use the Properties palette, which changes some of the properties of an table as a whole (see figure below at left) or else individual cells (shown at right, below).



Editable properties of tables



Editable properties of a cells

TIPS Use parentheses with functions, as well as to indicate operator order; by default, ARES places multiplication and division operates ahead of addition and subtraction.

Use : *(colon)* to specify a range of cells.

The formulas are placed as field text, which is why the text has a gray background. When you make a mistake in a formula, the program displays the field text error symbol — #### — in the afflicted cell. The mistakes can be due to one of the following problems:

- The function name is misspelled
- The cell reference was entered incorrectly, or does not exist
- An operator is missing, such as (or :

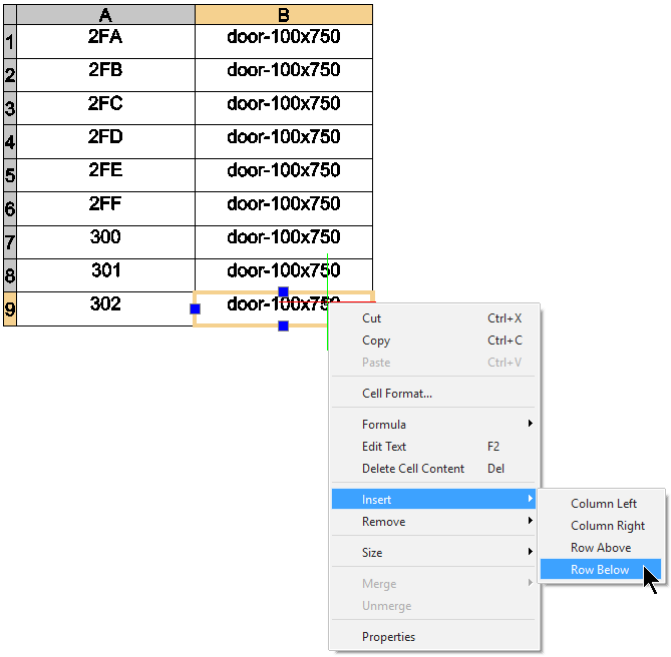
In this tutorial, you add a row to the bottom of a table to count the number of parts.

1. Open the *spreadsheet.dwg* file, and notice the table.

2FA	door-100x750
2FB	door-100x750
2FC	door-100x750
2FD	door-100x750
2FE	door-100x750
2FF	door-100x750
300	door-100x750
301	door-100x750
302	door-100x750

Example table open in the drawing

2. Click inside a cell in the last row. The cell is highlighted in orange with grips appear.
3. Add a row as follows:



   a. Right-click the cell.
   b. From the shortcut menu, choose **Insert**.
   c. From the sub-menu, choose **Row Below**.

4. Click in the first new cell and enter the following descriptor:

```
Total Doors
```

5. Click in the other new cell and then enter the formula for counting cells:

```
=count(B1:B9)
```

6. Exit the text editor. Notice that the cell reports the number of rows.

2FA	door-100x750
2FB	door-100x750
2FC	door-100x750
2FD	door-100x750
2FE	door-100x750
2FF	door-100x750
300	door-100x750
301	door-100x750
302	door-100x750
Total Doors	9

Formula reporting the number of doors

Tutorial: Exporting Tables

Command: EditTable
Alias: TablEdit
Menu: Modify | Entity | Table
Ribbon: Annotate | Tables | Table Edit (Drafting and Annotation)
Toolbar: --

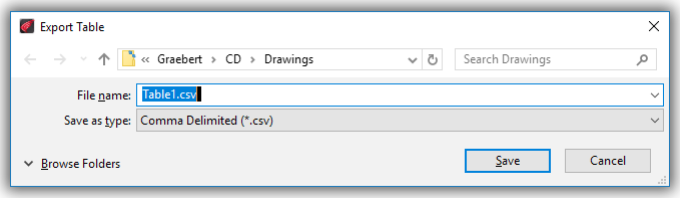
The program can export the contents of tables for use by other software. The **ExportTable** command prompts you to select the table, and saves it as a CSV (comma-separated format) file. As you saw earlier, CSV files an be imported into spreadsheets and databases.

In this tutorial, you export the table from the previous tutorial.

1. Start the **ExportTable** command using one of the methods listed above.

```
: exporttable
Specify table: (Select the table in the drawing)
```

2. After you pick the table in the drawing, the Export Table dialog box appears.



Naming the CSV file

Enter a file name, and choose the folder in which to store the file. Click **OK**.

3. Open the CSV file in a spreadsheet program, as described earlier.

Only the data in the cells are exported, no formulas or formatting. In a way, we have come full circle: export a drawing's attribute data in a CSV file, import and massage the data in a spreadsheet, import it back into the program as a table and massage the data some more, and then export the table out of the program, again as a CSV file.

Summary

You now know how to add text to drawings, format the text through styles, and override styles with properties. Attributes can be useful for tracking inventory and creating parts lists. Tables present data in a clear form. They can be created in the drawing from scratch, or else pasted from spreadsheets.

In the next chapter, you learn about working with another type of annotation: how to place dimensions.

CHAPTER 10

Placing Dimensions

In This Chapter

- Adding dimensions to drawings
- Creating dimension styles
- Formatting dimension text with dimension palette

Drawings almost always carry dimensions to show the size of objects, such as the height of a ceiling or the length and width of rooms. Strictly speaking, however, CAD drawings do not need dimensions. This is because everything is drawn full size, accurate to more than a dozen decimal places.

If you need to know the length or diameter of anything drawn by the program, use the GetDistance or Properties commands; if the drawing is on paper, use a scale ruler.

(Hand made drawings often included the warning, DO NOT SCALE. This meant that no one should use a scale ruler to measure distances on the paper drawings, because the drafter could not guarantee that every pencil line was accurate. Instead, drafters added dimensions sufficient to annotate the size of every part.)

Still, it is useful to add dimensions to important elements of CAD drawings so that anyone can quickly gauge the sizes of objects at a glance — without needing to start up the GetDistance command or hunting through the drawer for a ruler.

In this chapter, you learn how to apply linear and radial dimensions. Linear dimensions measure distances, while radial ones measure radii and diameters. You also learn how to control the size and look of dimensions.

KEY TERMS IN THIS CHAPTER

Baseline dimension refers to dimensions that have a common starting extension line.

Center mark indicates the centers of arcs and circles.

Continued dimension refers to dimensions that carry on from the last dimension.

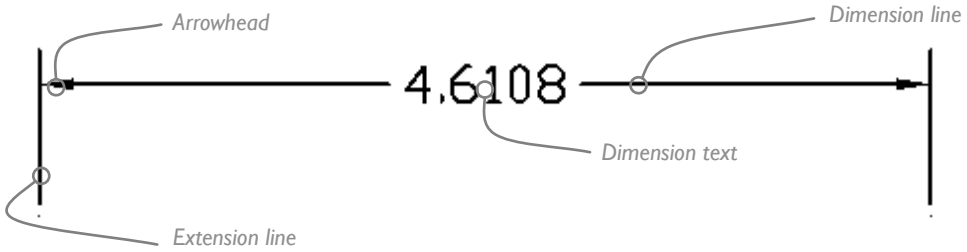
Dimension style specifies the properties of dimensions.

Dimensions show lengths, distances, and angles between entities in drawings.

Dimension line refers to the line that spans the distance between extension lines.

Extension line refers to the two lines that indicate the distance being measured.

Linear dimension shows horizontal, vertical, and rotated lengths of entities.



COMMANDS

Command	Shortcuts	Menu Selection
BaselineDimension	dimbase, dimbaseline	Dimension | Baseline
ContinueDimension	cdim, dimcontinue	Dimension | Continue
DiameterDimension	dimdia, dimdiameter	Dimension | Diameter
DimensionStyle	d, dimstyle	Format | Dimension Style
DimScale	...	...[1]
LinearDimension	dimlin, dimlinear	Dimension | Linear
ParallelDimension	dimali, dimaligned	Dimension | Aligned
RadialDiamension	dimrad, dimradial	Dimenion | Radius

[1] No menu selection.

Linear Dimensions

Command: LinearDimension
Aliases: dimlim, dimlinear, dli, ldim
Menu bar: Dimension | Linear
Ribbon: Annotation | Dimensions | Dimension | Linear (Drafting and Annotation)
Toolbar: 

Linear dimensions report the straight-line distance between two points, as shown by the following tutorial. The **LinearDimension** command draws horizontal, vertical, and rotated dimensions.

HORIZONTAL DIMENSIONS

You'll first practice drawing dimensions in a blank drawing; then later you dimension the office plan drawing.

To draw a linear dimension, start the program with a new drawing, and then follow these steps:

1. Start the **LinearDimension** command using one of the methods listed above.

   ```
   : lineardimension
   ```

2. Notice the prompts in the Command window. They prompt you to pick a point in the drawing. This is the starting point for the dimension:

   ```
   Default: Entity
   Options: Entity or
   Specify first extension line position» (Pick a point)
   ```

3. Hold down the **Shift** key to keep the dimension horizontal.

4. Move the cursor, and then pick a second point in the drawing. This is the ending point for the dimension:

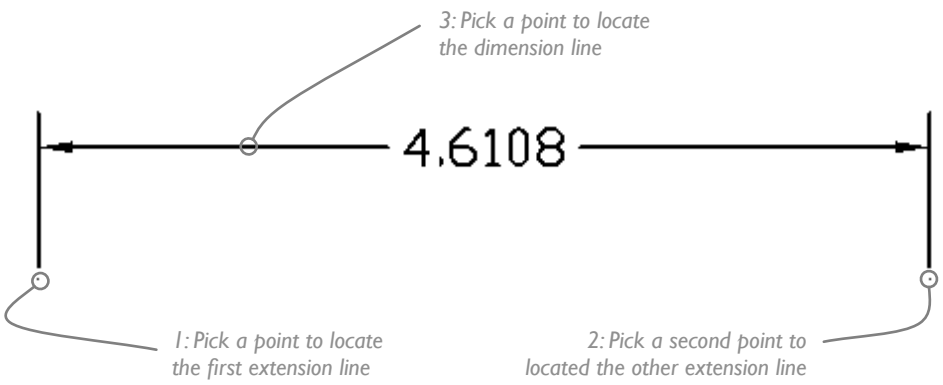
   ```
   Specify second extension line position» (Hold down Shift, and then pick a second point)
   ```

5. With the two pick points, you have defined the distance to be dimensioned. There is one more distance to specify: the one that locates the dimension line, which also determines the length of the extension lines.

   ```
   Options: Angle, Horizontal, Note, Rotated, Text, Vertical or
   Specify dimension line position» (Pick a third point)
   ```

   Notice that the program reports the distance on the dimension and in the Command window.

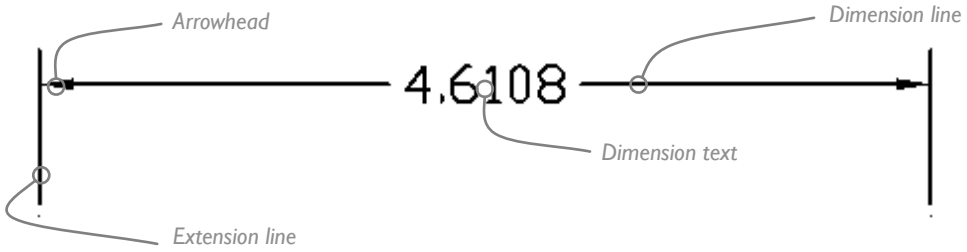
   ```
   Dimension Text : 4.6108
   ```



The entire linear dimension defined by three pick points

Notice that the program draws all the elements of the dimension automatically — three lines, two arrowheads, and the text. All you do is indicate the distance to be measured, and then locate the dimension line.

With some exceptions, dimensions always look the same, being made of the entities illustrated below:



Entities that make up dimensions

Extension lines — point to the start and endpoints of the distance being measured.

Dimension line — links the two extension lines.

Arrowheads — point to the two extension lines.

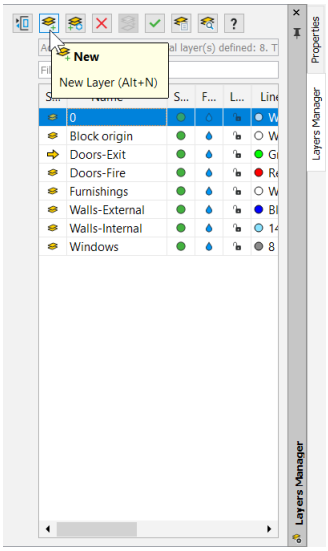
Dimension text — reports the distance between the two points.

There are variations, such as radial dimensions and leaders, some of which you see in this chapter or in Chapter 15. The look of dimension elements is customized through the DimensionStyle command, such as the color of the lines, type of arrowheads, and where the text is positioned.

Vertical Dimensions

In the tutorial, you drew a horizontal dimension, which is probably the most common type of dimension. the program also draws vertical dimensions, rotated dimensions, and aligned ones.

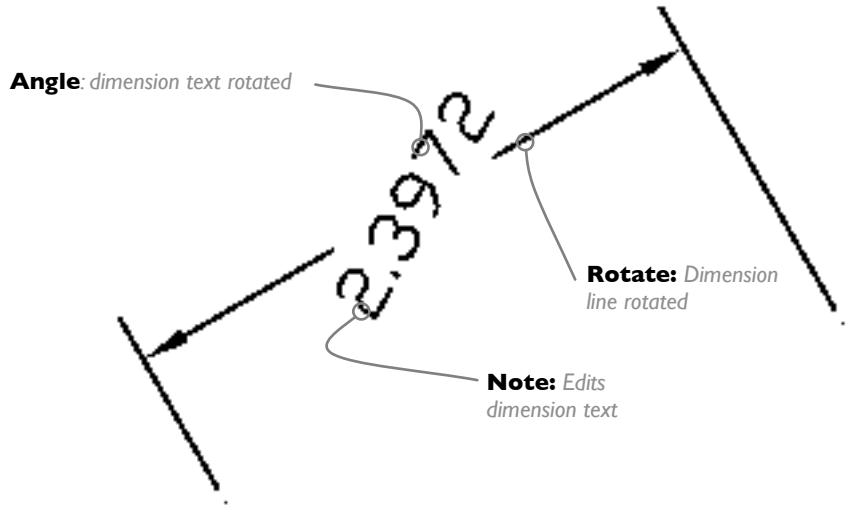
1. Open the *office plan.dwg* file.
2. Ensure entity snaps are turned on; the **ESnap** button should be glowing blue on the status bar. Entity snaps are necessary to ensure that you place dimensions accurately.
3. **(NEW IN 2019)** Use the **Layer** command to open the Layers Manager palette. Notice that this command opens a palette, not a dialog box. The palette hangs on the edge of the program. Create a new layer, as follows:



Layers Manager palette

SUMMARY OF LINEARDIMENSION COMMAND OPTIONS

```
: lineardimension
Default: Entity
Options: Entity or

Specify first extension line position» (Pick a point.)
Specify second extension line position» (Hold down Shift, and then pick a
second point.)
Options: Angle, Horizontal, Note, Rotated, Text, Vertical or
Specify dimension line position» (Pick a third point.)

Dimension Text: 6.3469
```



Entity dimensions the selected entity; extension line positions are not used.

Angle specifies the angle of the dimension text; the default is 0 degrees:

```
Specify text angle» (Enter an angle, and then press Enter)
```

Horizontal forces the dimension line to be horizontal.

Note customizes the dimension text with the Edit Note dialog box.

Rotated specifies the angle of the dimension and extension lines:

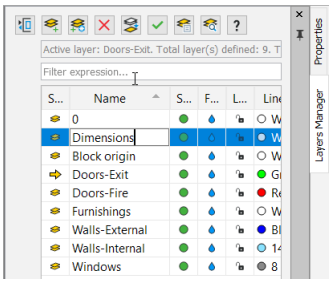
```
Specify dimension line rotation» (Enter an angle, and then press Enter)
```

Text customizes the dimension text in the Command window; allows you to override the text calculated by the program automatically:

```
Specify dimension text » (Enter the dimension text, and then press Enter)
```

Vertical forces the dimension line to be vertical.

a. Click the **New Layer** button, and then name the new layer "Dimensions."

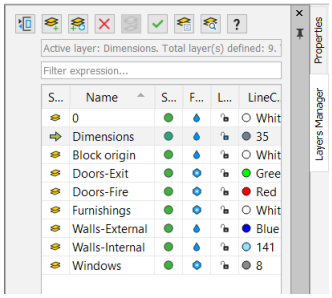


Adding the "Dimensions" layer to the drawing

b. Change the layer's color to one that contrasts with the others, such as brown (color #35).

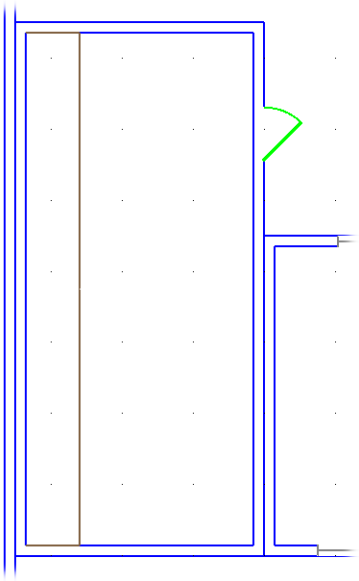
c. Make the Dimensions layer active by clicking the **Activate** button.

d. Freeze unnecessary layers, such as Furnishings and Doors-*, by clicking the appropriate box next to them in the **Frozen** column. The icon changes from a raindrop to a snowflake, and so the dialog box should look like this:



Layers Manager with newly added layer "Dimensions"

4. Click **OK** to exit the dialog box.
5. You'll first dimension the storage room. Zoom into the area of the storage room with the **Zoom W** command.
6. To draw a vertical dimension, start the **LinearDimension** command. This single command places three kinds of dimensions, horizontal, vertical, and rotated.
7. Follow the prompts to dimension the inside length of the storage room:



Dimensioning the length of the storage room

```
Default: Entity
Options: Entity or
Specify first extension line position» (Pick one intersection)
Specify second extension line position» (Pick other intersection)
Options: Angle, Horizontal, Note, Rotated, Text, Vertical or
Specify dimension line position» (Pick a point away from the wall)
Dimension Text : 7200.0000
```

Oops! You can see the dimension line, but not the text and arrowheads! Dimensions are like text; they need to be scaled large enough to see them. The dimension you drew had a scale of 1, which is too small for such a large floor plan drawing. The 1:1 scale is suitable for drawings made at millimeter scale, but your drawing is at the meter scale.

Scaling Dimensions

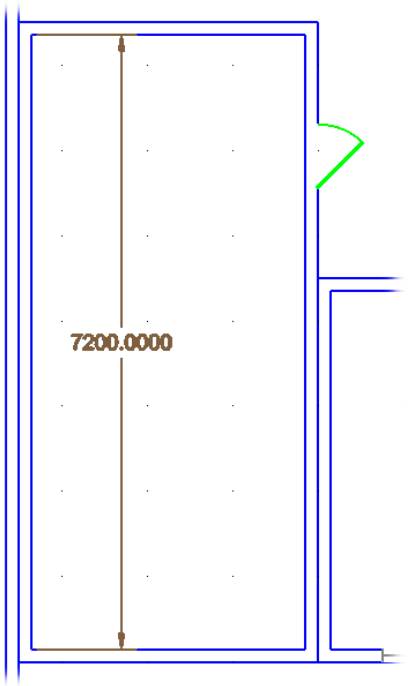
System variable: DimScale

To fix the size problem, use the **DimScale** system variable to change the dimension scale to 1000, making the dimensions a thousand times larger.

1. DimScale is, unfortunately, not retroactive and so you need to erase the vertical dimension. Select the dimension, and then enter the **Delete** command — or just press the **Del** key on the keyboard.
2. Enter **DimScale**, and then specify the scale factor at 1000, as follows:

```
: dimscale
Default: 1.00
Enter new value for DIMSCALE» 1000
```

3. Repeat the **DimLinear** command to re-dimension the storage room.



Linear dimension visible after adjusting the dimension scale

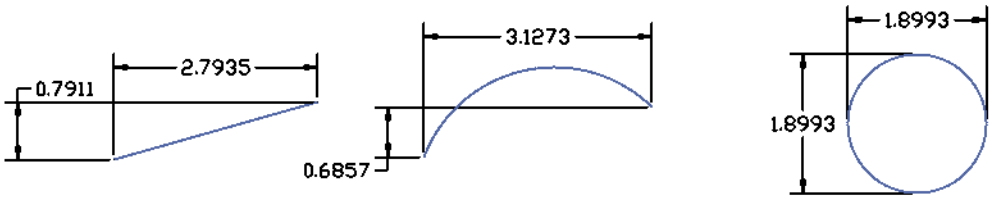
That's better!

Dimensioning Entities

So far, you have been picking two points to indicate the location of the extension lines. There is an alternative method that is more efficient in some cases: you can select an entity, and the program will dimension it automatically.

You make just two picks: select the entity, and then position the dimension line. The dimensioning command's **Entity** option places a horizontal or vertical dimension, depending on how you drag the cursor away from the entity.

The figure below illustrates how a line, an arc, and a circle are dimensioned by the LinearDimension command's Entity option:



Linear dimensions applied to line, arc, and circle entities

- For horizontal dimensions, drag the cursor up or down; ortho mode does not need to be on
- For vertical dimensions, drag the cursor left or right

In this tutorial, you use Entity mode to dimension a window.

1. Restart the **LinearDimension** command.
2. Enter '**e**' at the prompt to switch to entity-dimensioning mode:

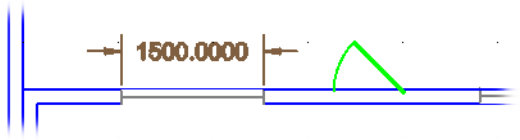
```
: lineardimension
Default: Entity
Options: Entity or
Specify first extension line position» e
```

3. The program prompts you to select an entity. In this case, pick a window block:

```
Specify entity» (Pick a window block)
```

4. Move the cursor, and then position the dimension line.

```
Options: Angle, Horizontal, Note, Rotated, Text, Vertical or
Specify dimension line position» (Pick a point for the dimension line)
Dimension Text : 1500.0000
```



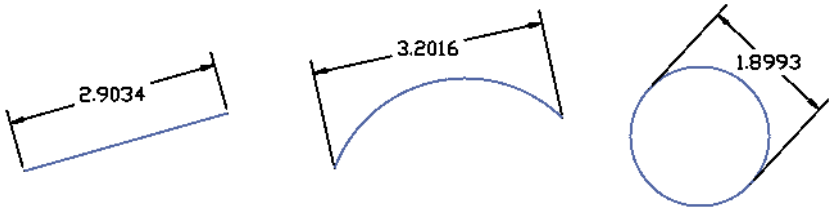
Window gets the entity dimension treatment

Notice that the program found the opposite ends of the block, and then positioned the extension lines automatically.

Parallel Dimensioning

Command: ParallelDimension
Aliases: dal, dimali, dimaligned, pdim, paralledim
Menu bar: Dimension | Aligned
Ribbon: Annotation | Dimensions | Dimensions | Parallel (Drafting and Annotation)
Toolbar:

The program also has the **ParallelDimension** command (a.k.a. aligned dimensioning), whose Entity option places dimensions parallel to entities. In the examples illustrated below, notice how the values of the lengths differ from those shown on the previous page by horizontal and vertical dimensions.



Parallel dimensions aligned to line, arc, and circle entities

- For arcs, the command places parallel dimensions between the endpoints of arcs.
- For circles, the command places the parallel dimension at the first pick point.

This command ignores the setting of ortho mode.

CONTINUED DIMENSIONS

Command: ContinueDimension
Aliases: cdim, dco, dimcont, dimcontinue
Menu bar: Dimension | Continue
Ribbon: Annotation | Dimensions | Continue (Drafting and Annotation)
Toolbar:

Once you draw one dimension, you can have the program draw more, all perfectly aligned. This is accomplished with the **ContinueDimension** command, which "chains" additional dimensions onto the first one.

The first dimension of a continuous series must be placed with the LinearDimension command; thereafter, you use the ContinueDimension command. All you need to do is pick points for the second extension lines; the program places the dimension line at the same distance as previous ones. (The first dimension line is not drawn, because it would overlap with the second extension line.)

I find it is easier to see continued dimensions at work, rather than to read about them. And so in the following tutorial, you place a string of continued dimensions along the top wall of the office layout.

1. Currently, the dimension text displays four decimal places when none is required. To change the display accuracy, use the DimDec system variable, as follows:

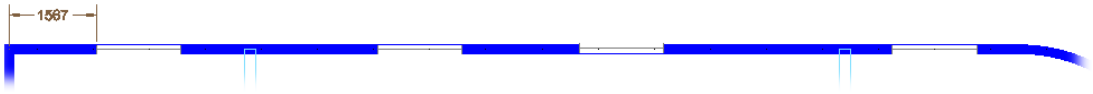
   a. Enter **DimDec** at the command prompt.

   ```
   : dimdec
   ```

   b. Type **0**, and then press **Enter**.

   ```
   Default: 4
   Enter new value for DIMDEC» 0
   ```

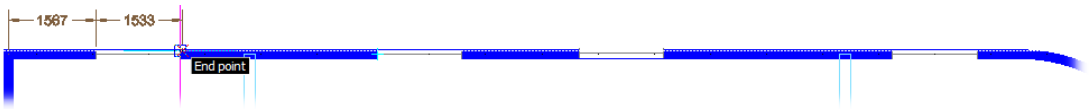
2. The first dimension of a continuous series must be placed with the LinearDimension command; thereafter, you use the ContinueDimension command. Thus, enter the **LinearDimension** command, and then place the initial dimension, as illustrated below.



Placing the initial linear dimension

3. Now you can switch to continued dimensions. Start the **ContinueDimension** command using one of the methods listed above.

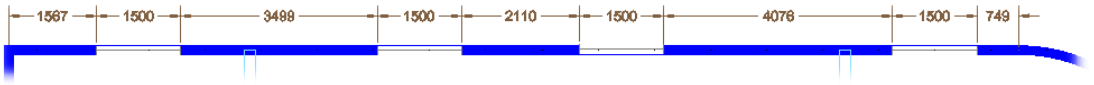
```
: continuedimension
Default: Select dimension
Options: Select dimension, Undo or
Specify second extension line position» (Pick a point)
Dimension Text : 1500
```



Continuing the linear dimension and an additional one

You just make one pick point (the location of the second extension line), and the program draws the rest.

4. The **ContinueDimension** command repeats automatically so that you can continue placing dimensions.



Entire wall dimensioned with continued dimensions

```
Default: Select dimension
Options: Select dimension, Undo or
Specify second extension line position» (Pick another point)
Dimension Text: 3489
```

5. When done, press **Enter** to end the command.

```
Default: Select dimension
Options: Select dimension, Undo or
Specify second extension line position» (Press Enter to exit the command)
```

This command continues linear dimensions created by the LinearDimension, ParallelDimension, and ArcLengthDimension commands; it does not work with radial dimensions made by RadiusDimension or DiameterDimension.

TIPS To continue a dimension that was drawn earlier, use the ContinueDimension command's **Select dimension** option, which prompts you to select a linear dimension in the drawing:

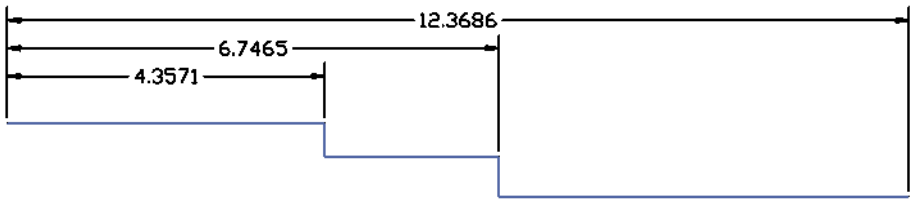
```
Specify dimension: (Pick a linear dimension)
```

The **BreakLine** command places the break symbol in lines. This command can be found in the XtraTools menu and ribbon tab.

Baseline Dimensions

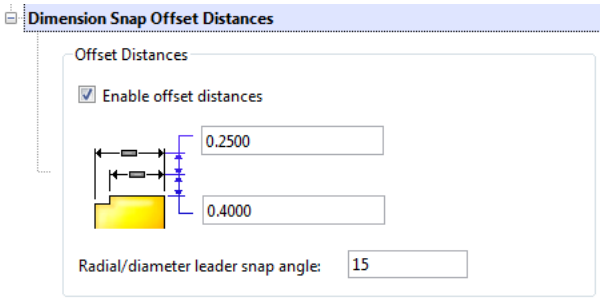
Command: BaselineDimension
Aliases: basedim, bldim, dba, dimbase, dimbaseline
Menu bar: Dimension | Baseline
Ribbon: Annotation | Dimensions | Baseline (Drafting and Annotation)
Toolbar:

The program also has the **BaselineDimension** command, which continues dimensions from a common base line, as illustrated below.



Baseline dimensions applied to a stepped wall

TIP You can customize the distance between dimension lines using the Dimension Offset Distances option, as well as the angle between offset radial dimensions:



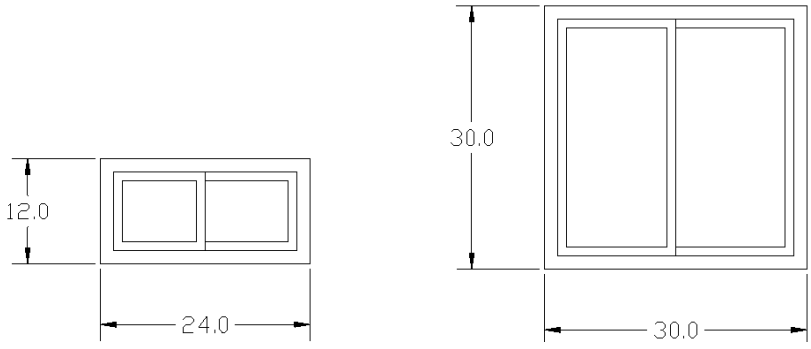
You access this features through the **Options** command: choose **Drawing Settings** and then go into **Dimension Snap Offset Distances**.

MORE TIPS Notice that both the **RadiusDimension** and **DiameterDimension** commands place a center mark (+) to indicate the center of the arc or circle. The **CenterMark** command draws the mark, as well.

For large-radii arcs, you can use the **JoggedDimension** command, which allows you to relocate the center mark to more convenient locations.

Other useful dimensioning commands include **AngularDimension** (for measuring angles), **Leader** (for placing call-outs), and **Tolerance** (for placing geometric tolerances).

Dynamic blocks can be dimensioned, and when the block's size is changed dynamically, the associative dimensions update, as illustrated below:



Radial Dimensions

Command: RadiusDimension
Aliases: dimrad, dimradius, dra, rdim
Menu bar: Dimension | Radius
Ribbon: Annotation | Dimensions | Dimension | Radius (Drafting and Annotation)
Toolbar:

Placing radial and diameter dimensions is like using the Entity option of linear dimensions: you pick a circle or arc, and the program dimensions it immediately.

In the following tutorial, you use the **RadiusDimension** command to dimension the arc-shaped executive office:

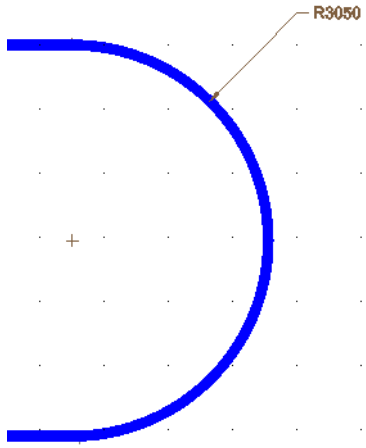
1. Start the **RadiusDimension** command:

   ```
   : radiusdimension
   ```

2. Select the curved wall of the executive office.

   ```
   Specify curved entity» (Pick the arc)
   ```

   This command works with arcs, circles, and polyarcs; it does not work with ellipses or splines because their curves are not of constant radius.



Placing a radial dimension on an arc

3. Locate the leader line. Notice that it can be placed inside or outside the arc; the standard is to place it outside of the drawing.

   ```
   Options: Angle, Note, Text or
   Specify dimension position» (Pick a point)
   Dimension Text : 3050
   ```

Diameter Dimensions

Command: DiameterDimension
Aliases: ddi, dimdia, dimdiameter
Menu bar: Dimension | Diameter
Ribbon: Annotation | Dimensions | Dimension | Diameter (Drafting and Annotation)
Toolbar:

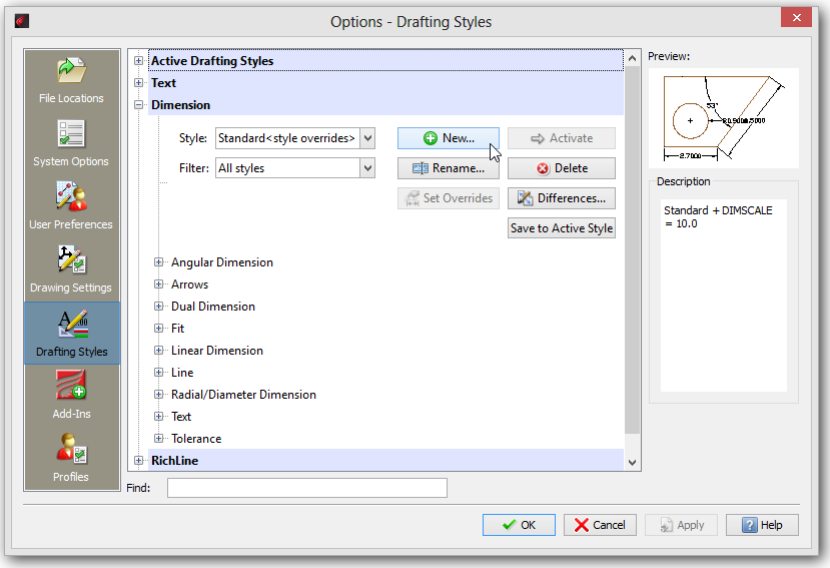
The program also has the **DiameterDimension** command, which dimensions the diameters of circles and arcs, as illustrated below.

Placing diameter dimensions on arcs and circles

Modifying Dimension Styles

Command: DimensionStyle
Aliases: d, dimstyle, dimsty, dst, ddim
Menu bar: Format | Dimension Style
Ribbon: Annotation | Dimensions | Dimension Style (Drafting and Annotation)
Toolbar:

The look of dimensions is controlled globally by dimension styles, just as the look of text is controlled by text styles. The difference is that there are many, many more options for dimensions than for text.



The Dimension Styles section of the Options dialog box

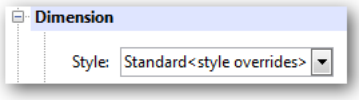
When earlier you used the DimScale and DimDec commands, you *overrode* their settings in the default dimension style found in every drawing, named “Standard.”

There are three ways you can access dimension styles: (1) through the DimensionStyle command's Options dialog box, (2) through the Properties palette, and (3) at the command line.

In this tutorial, you familiarize yourself with the Options dialog box, because it is the most common way of customizing the look of dimensions.

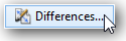
1. Enter the **DimensionStyles** command using one of the methods listed above. Notice that the Options dialog box opens up and displays the Drafting Styles section.

   Notice that the name of the active style is "Standard<style overrides>."

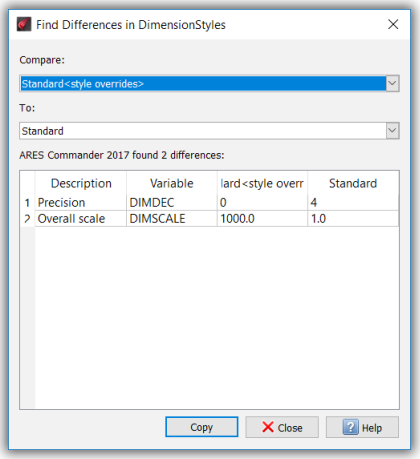


Current dimension style "Standard" with some settings overridden

   The **<Style overrides>** text alerts you to the fact that the style was overridden in one or more areas. Indeed, this is what you did when you changed the values of the DimScale and DimDec system variables. In fact, all aspects of dimensions controlled by this dialog box are stored in system variables.

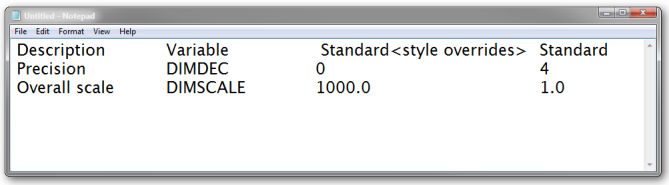
2. To see the difference between the Standard style and the overrides, click the 

Differences button. Notice that the Find Differences in Dimension Styles dialog box lists the differences between any two dimension styles whose name you choose from the droplists.



Examining what's changed in the "Standard" dimension style

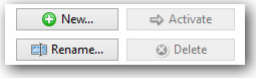
TIP You can copy the list of differences to the Clipboard by clicking the **Copy** button. Then you can paste the data into a word processor or spreadsheet (as illustrated below) with the **Edit | Paste** command.



---

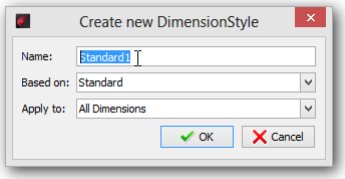
3. In the Find Differences dialog box, click **Close** to return to the Options dialog box.

4. Now look near the top of the Options dialog box for the quartet of buttons illustrated below. They create new styles, rename them, and so on — just like the way you do with text styles.



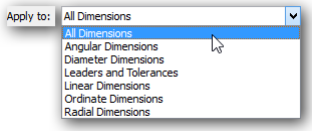
Buttons for making new dimension styles, renaming them, and so on

5. To create a new dimension style, select an existing style and then modify it, like this:
   a. Click  **New** button. Notice the Create New Dimension Style dialog box.
   b. In the **Name** field, enter a name for the new style.

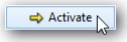


Creating a new dimension style

   c. If necessary, choose an existing style from the **Based On** droplist to use as the basis for the new one.
   d. In the **Apply To** droplist, decide if the new style applies to all dimensions or just certain types, as illustrated below; if you are not sure, choose **All Dimensions**.



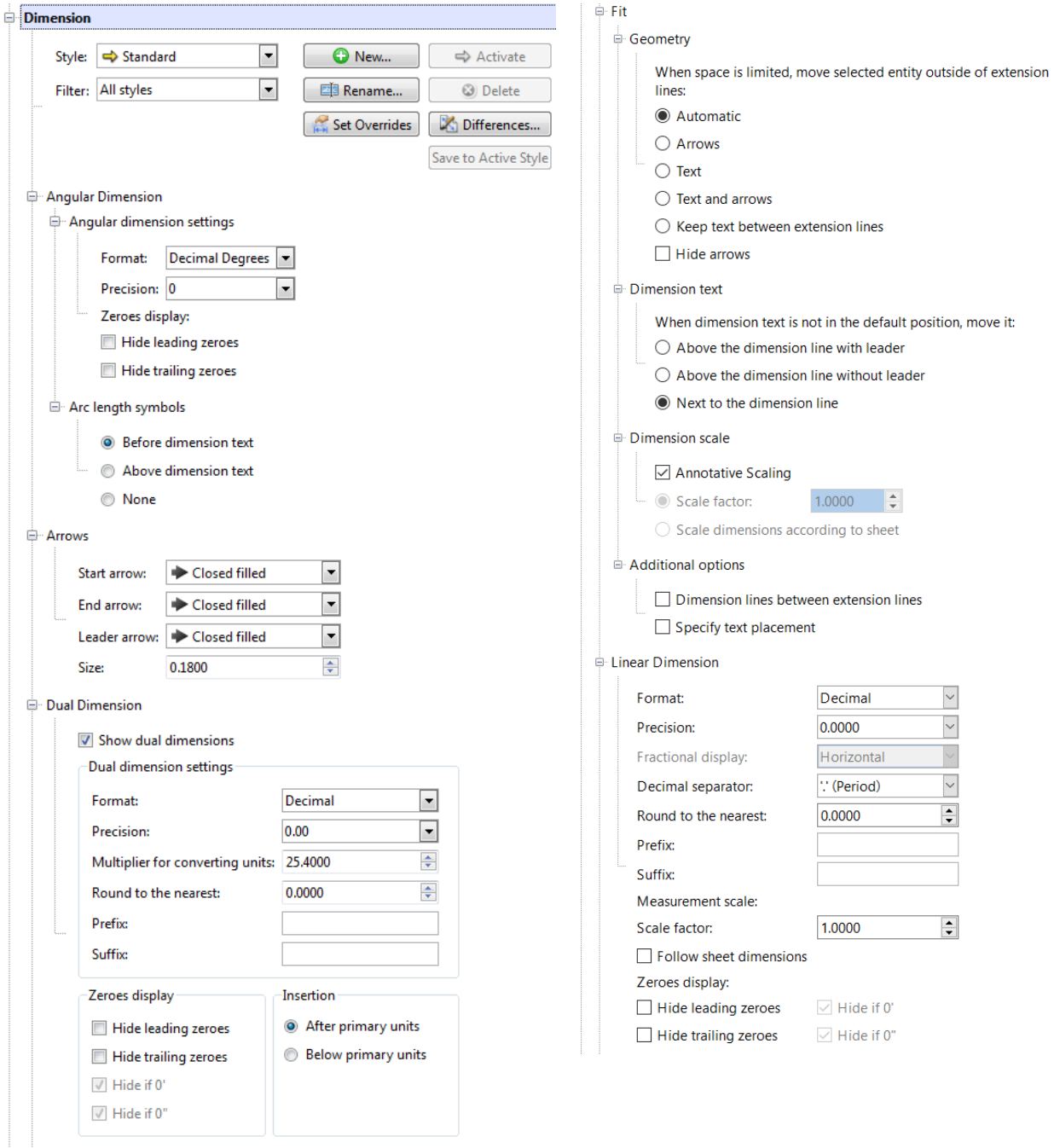
Choosing which dimension parameters to include

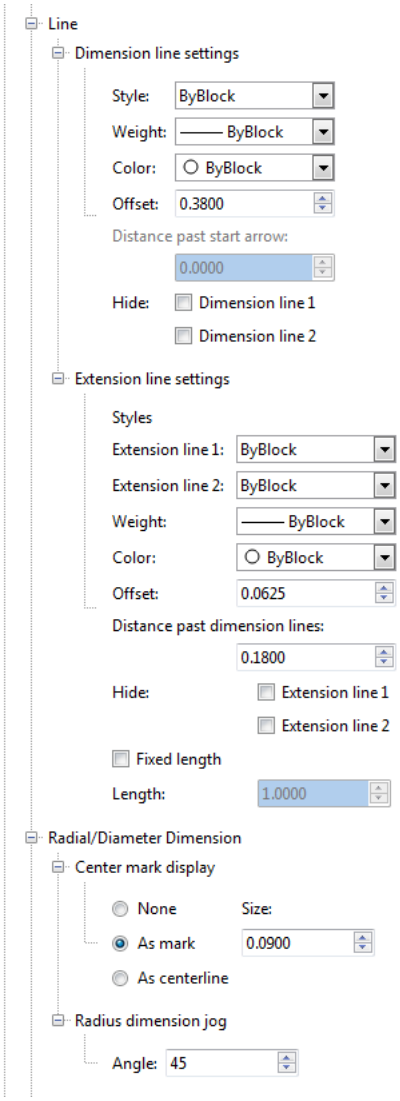
   e. Click **OK** to complete the naming.
   f. Back in the Options dialog box, change the dimension options as necessary. I show all of them on the following pages.
   g. Click  **Activate** button to use the new style for all dimensions from now on
6. Click **OK** to close the dialog box.

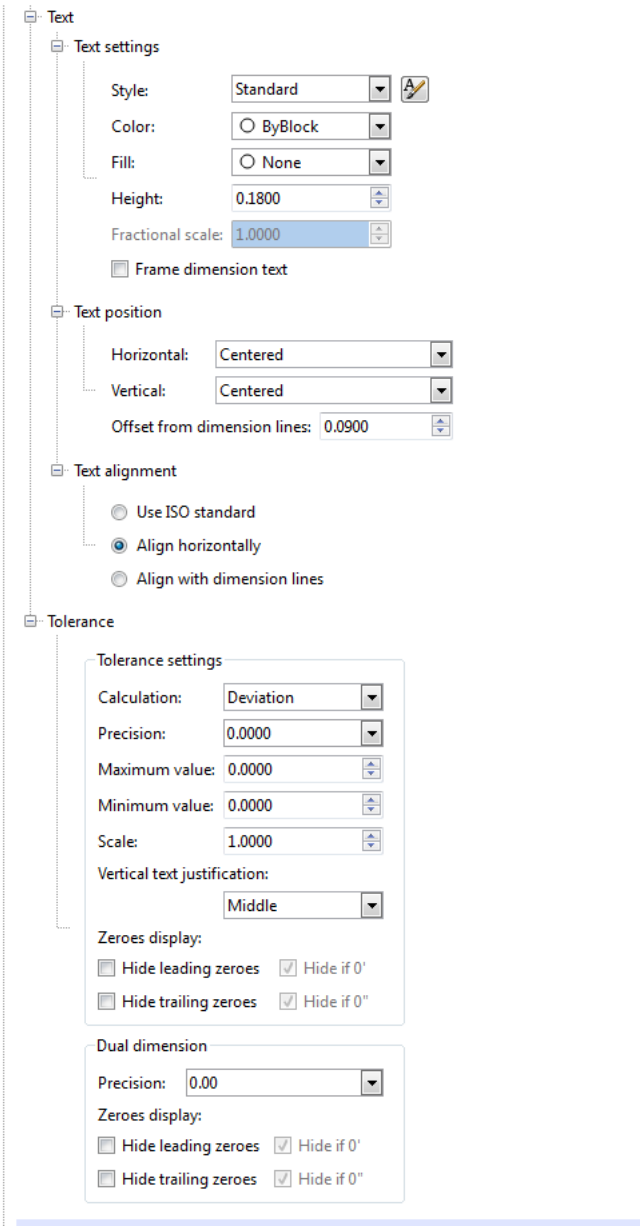
Any dimension you draw now takes on the new style; existing dimensions keep their old style. You can use the **RebuildDimension** command to update dimensions.

Dimension Style Guide

These pages display all dimension style settings available in the program's Options dialog box.



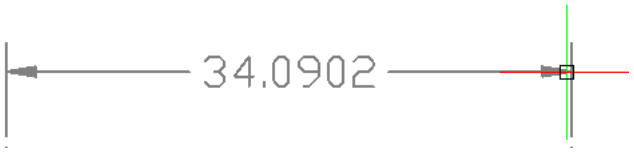

Line
Dimension line settings
Style: ByBlock
Weight: ByBlock
Color: ByBlock
Offset: 0.3800
Distance past start arrow:
0.0000
Hide: Dimension line 1
Dimension line 2
Extension line settings
Styles
Extension line 1: ByBlock
Extension line 2: ByBlock
Weight: ByBlock
Color: ByBlock
Offset: 0.0625
Distance past dimension lines:
0.1800
Hide: Extension line 1
Extension line 2
Fixed length
Length: 1.0000
Radial/Diameter Dimension
Center mark display
None
As mark
As centerline
Size:
0.0900
Radius dimension jog
Angle: 45


Text
Text settings
Style: Standard
Color: ByBlock
Fill: None
Height: 0.1800
Fractional scale: 1.0000
Frame dimension text
Text position
Horizontal: Centered
Vertical: Centered
Offset from dimension lines: 0.0900
Text alignment
Use ISO standard
Align horizontally
Align with dimension lines
Tolerance
Tolerance settings
Calculation: Deviation
Precision: 0.0000
Maximum value: 0.0000
Minimum value: 0.0000
Scale: 1.0000
Vertical text justification:
Middle
Zeroes display:
Hide leading zeroes
Hide if 0'
Hide trailing zeroes
Hide if 0"
Dual dimension
Precision: 0.00
Zeroes display:
Hide leading zeroes
Hide if 0'
Hide trailing zeroes
Hide if 0"

Customizing Dimension Text

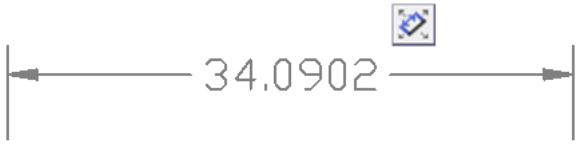
The dimension palette appears when you place dimensions in drawings, as well as when you choose a dimension for editing. The palette changes the properties and formatting of dimension text: There is no command to open the palette; it appears automatically like this:

1. Draw a dimension or select an existing one.



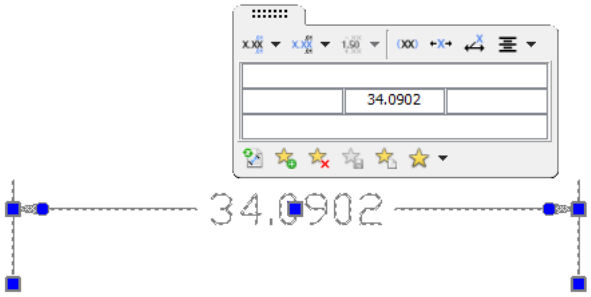
Placing a new dimension

2. As you complete the dimension (or select an existing one), an icon appears that floats near the dimension.



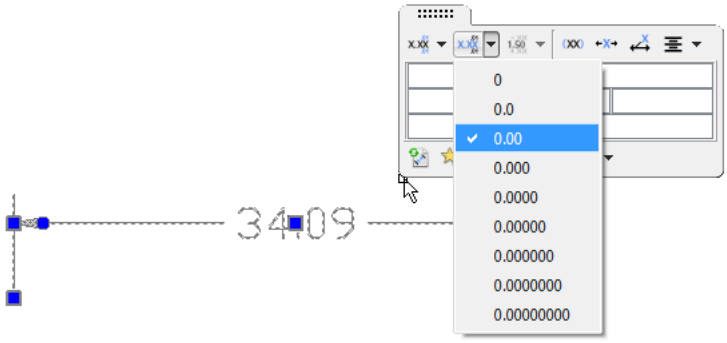
Floating icon accesses the dimension palette

3. Move the cursor over the icon (no need to click it). Notice that the dimension palette appears.



Dimension palette adjusting text of dimensions

4. Adjust settings in the palette. For example, click the **Unit Precision** button to change the number of decimal points displayed.

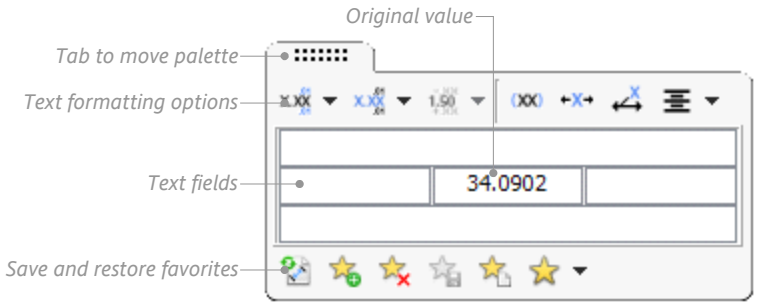


Changing decimal places to 2

5. When done, press **Esc** to exit the palette.

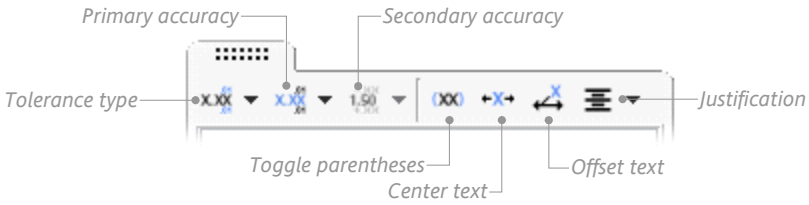
TOURING THE DIMENSION PALETTE

This is a tour of the dimension palette and its many useful functions.

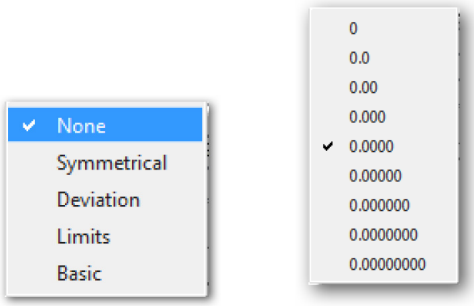


TEXT FORMATTING OPTIONS

The top toolbar formats dimension text:



Tolerance Type button chooses the type of tolerance, as listed by the illustration below. This is like the Tolerance section of the Dimension Style dialog box; the difference is that the changes you make here override the style and apply only to the dimension with which you are working.



***Left:** Choosing the style of tolerance; **right:** Changing the number of decimal places*

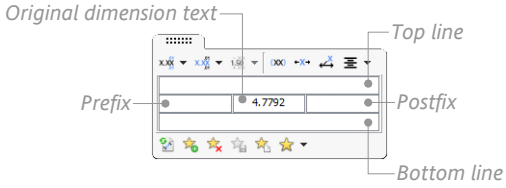
Primary and **Secondary Accuracy** droplists let you choose the number of decimals displayed; see figure above. The first button is for the primary dimension, and so can be used for regular, non-tolerance dimensions; the second button for the tolerance values.

Format Text buttons perform these functions:

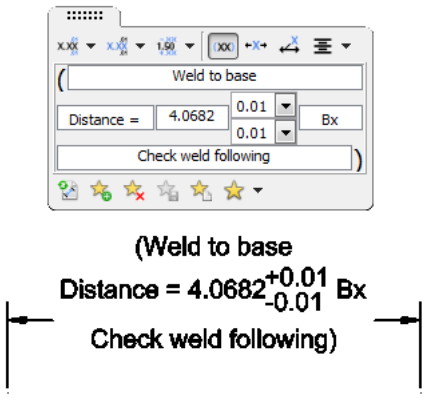
Toggle parentheses button places parentheses around the text
Center text button re-centers text if it was moved from the dimension line
Offset text button moves the text away from the dimension line
Justification button positions the text relative to a bounding box

TEXT FIELDS

The three rows of text field show the original value in the center; you can add text in the other fields.



Here I add text to all the fields and turn on almost all of the options:

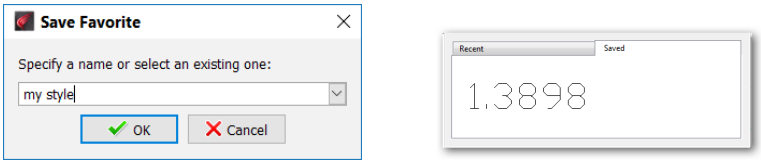


FAVORITES

The toolbar at the bottom saves and restores favorites, kind of like dimension styles that are independent of the DimensionStyle command. From left to right, these are:

Apply Default button applies the default favorite formatting to the dimension text

Add or Update Favorite button adds the current style as a favorite for reuse; prompts you to name the style



***Left:** Saving the current settings as a favorite; **right:** Showing the favorites list*

Delete Favorite button removes a stored favorite style through a dialog box

Save Favorite button saves the style to a *.dimfvt* file

Load Favorite button loads styles from *.dimfvt* files

Favorites List button displays the styles that have been saved; choose one to load

Summary

This chapter showed how to scale, place, and format dimensions. In Chapter 15, you learn about other forms of dimensions, such as ones with both metric and imperial text.

For information on using dimensional constraints and parameters, see Chapter 16.

Notes

PART III

ARES on Mobile Devices and Web Browsers

Notes

CHAPTER 11

ARES Touch for Phones and Tablets

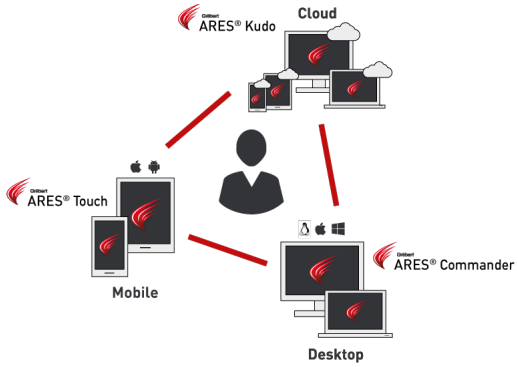
In This Chapter

- Understanding how mobile apps differ from desktop programs
- Touring the user interface of ARES Touch
- Using commands and entering options
- Learning about the differences between ARES Touch and ARES Commander

Graebert GmbH refers to its CAD offering as "Trinity," because they offer software, platforms, and licenses as triads: three desktop operating systems, three operating platforms, and three licenses styles. ARES was written to run on all popular platforms today — both hardware and operating systems. The user interface, command sets, and APIs are similar on all three.

The company offers these versions of ARES CAD software for each operating platform:

» ARES **Commander** runs on desktop and laptop computers (as described in previous chapters)
» ARES **Touch** runs on handheld tablets and phones (in this chapter)
» ARES **Kudo** runs inside Web browsers (see the following chapter)



Versions of ARES for multiple platforms

KEY TERMS IN THIS CHAPTER

Android is the most popular operating system for handheld devices; made by Google

App is a program that runs on handheld devices, such as tablets and smartphones; short for "application"

ARES Kudo is Graebert's CAD software that runs in Web browsers from remote servers ("cloud")

ARES Touch is Graebert's CAD app for handheld devices running Android and iOS

iOS is the second most-popular operating system for handheld devices; made by Apple

HOW TO INSTALL ARES TOUCH

Touch is available in 14 languages: Brazilian Portuguese, Czech, English, French, German, Italian, Japanese, Korean, Polish, Russian, simplified Chinese, Spanish, traditional Chinese, and Turkish. To install ARES Touch on your Android device, go to Google's online Play store (54MB) at https://play.google.com/store/apps/details?id=com.graebert.aresbeta

For iOS devices, go to Apple's online App store (139MB) at https://itunes.apple.com/us/app/ares-touch/id988848336

ARES Touch initially runs 30 days in full drawing and editing mode. After that time, you pay for a subscription, or else use Touch for viewing and simple annotations only.

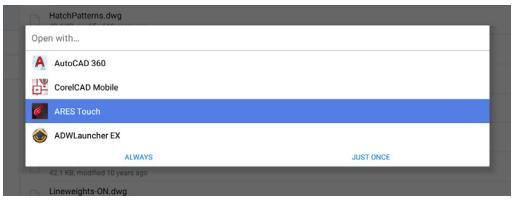
TIP To avoid getting a virus on your device, it is important that you download the apps from the official online stores, and not from third-party stores or third-party installers.

ARES Touch can run on desktop computers when you use an Android emulator.

WORKING WITH CLOUD STORAGE

You can open drawings directly to ARES Touch from inside services such as Dropbox, Box, Google Drive, and iCloud (Apple users only). Here's how:

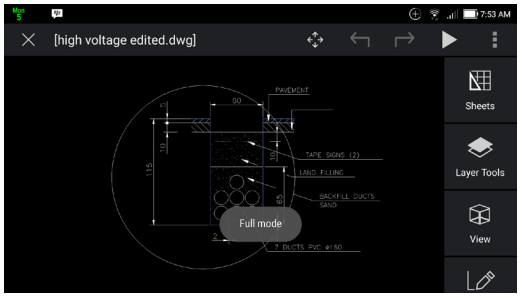
1. Open the Dropbox, Box, or Google Drive app on your portable device.
2. Tap the *.dwg* file you want to open in Touch.
3. Notice that Android (or iOS) ask you to select the app in which to open the *.dwg* file:



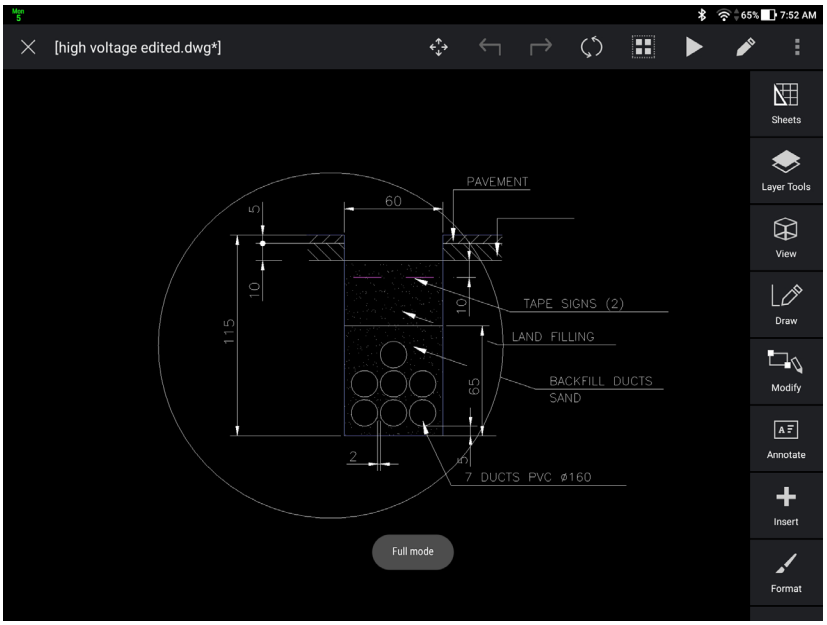
4. Chose **ARES Touch.**
5. If you want ARES Touch to always open the file, then tap **Always;** if you want to choose each time, then tap **Just Once.** Notice that the drawing opens in Touch.

About ARES Touch

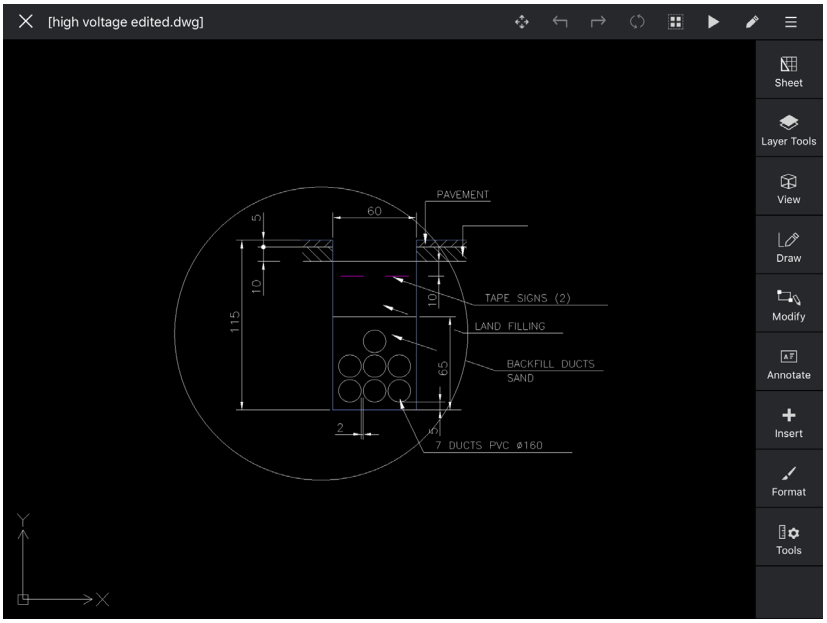
ARES Touch is the version of ARES that runs on Android and iOS tablets and smartphones. Here is what it looks like running on these devices:



ARES Touch running on an Android smartphone



ARES Touch running on an Android tablet



ARES Touch running on an iOS tablet

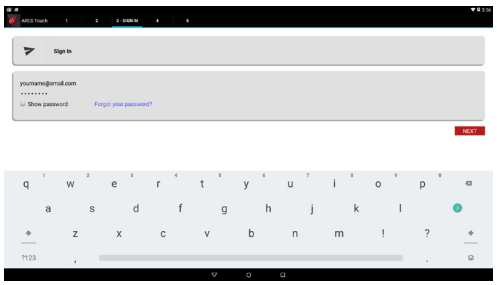
STARTING ARES TOUCH THE FIRST TIME

The first time you run ARES Touch on your tablet or smartphone, you have to go through registration. Here is how to navigate through it.

1. Activate the license:
   - » If you have a license for ARES, then click **Activate**.
   - » If not, then click **Free Trial**.

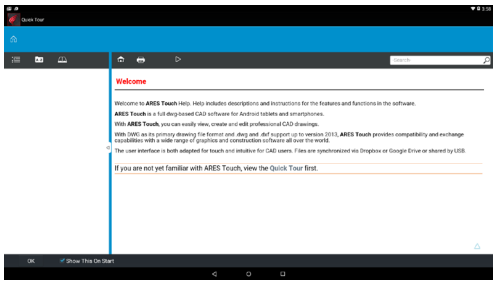
2. Sign in to your account:
   - » If you already have an account with Graebert, then click **Sign In**.
   - » If not, then click **Register**.

3. Enter your email address and password:



4. You receive confirmation. Click **Start** to continue.

5. The next screen that greets you is the Quick Tour. To continue to the CAD program, click **OK** (button is found in the lower left corner).



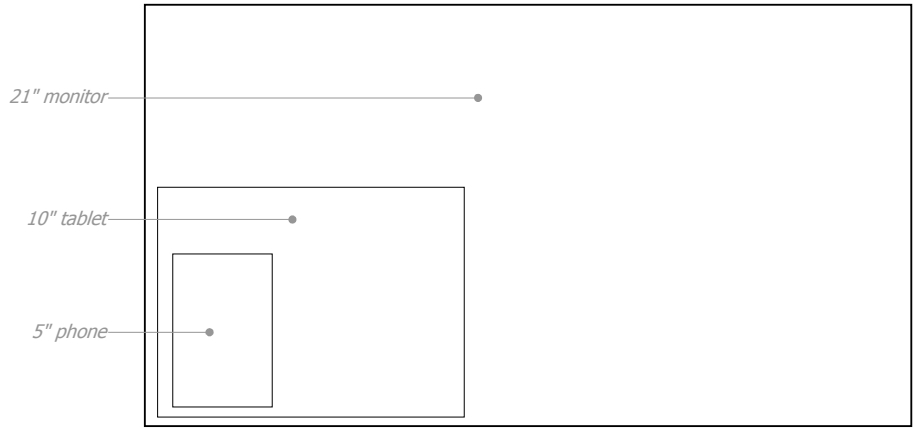
Unlike similar apps from other CAD vendors, Touch is more than a DWG viewer or markup app: it has the largest set of drawing and editing functions of any mobile CAD app. It reads and writes DWG files without translation, and offers several hundred commands for 2D drawing and editing, 3D viewing, entity snap and selection modes, and dimensioning.

Most portable devices are used with a finger, and so Touch includes a "loupe" that shows the area under the finger enlarged and offset. For those who like to customize their CAD system, Touch supports C++, LISP, Tx and DCL; it is possible to migrate add-ons written for the desktop to the field.

Files are stored locally on the device or automatically synchronized with the cloud. *Synchronization* means that you work with the same version of the drawing on your desktop computer as on the mobile app. Touch accesses *.dwg* files through Dropbox, Google Drive, email, and other online services.

In this chapter, you learn how to operate Touch on tablets and in conjunction with ARES Commander on desktop computers.

When you see ARES Touch for the first time, you will notice that it has no menu bar, no ribbon, no command bar. This is because the program's interface must fit a much smaller area — as small as the area of a smartphone. The figure below illustrates the comparative screen areas of a 21" desktop monitor with typical 10" tablets and 5" phones.



Comparison of screen sizes

In this chapter, all of the screen grabs are of ARES Touch running on a 10" Android tablet. The iPad version of the app looks similar.

Walking Through the Touch UI

Using ARES Touch is quite different from ARES Commander, because software that runs on portable devices must accommodate touch-only interfaces and smaller screen sizes. This means that there are almost no hints for what buttons do, unlike the tooltips and other guides found in software for desktop computers. (There is no way in Android or iOS to hover a cursor over a button and wait for an explanatory tooltip so show up — sadly!)

In short, you spend time guessing the purpose of buttons until you get them memorised; no fault of Graebert, that's just how Android and iOs impose their design guidelines on developers.

WHY MOBILE CAD DIFFERS FROM DESKTOP CAD

Graebert has one CAD program for running on Windows and MacOS and Linux. Why does it need *different* programs for running on Android and iOS? Because handheld computers, like smartphones and tablets, are designed very differently from desktop computers and laptops. Here is an overview of some of the differences.

USER INTERFACE

Handheld devices depend on us using our finger to interface with the user interface, instead of the mouse cursor of desktop and laptop computers. Elements like buttons and menus must be larger for tablets and smartphones, because our fingertips are fat compared to the cursor.

Graebert includes icons and other UI elements in five sizes to accommodate different resolutions and screen sizes sported by portable devices.

MEMORY AND STORAGE

Portable devices have less memory than desktop and laptop computers in which to run programs and less storage for files. This is a problem only when you need to access large drawings. Still, it is good to be aware that some smartphones have as little as 0.5GB RAM, although most today have 4GB to 8GB, and as I write this 12GB models are coming online. By comparison, today's desktop computers start with 4GB and commonly go to 16GB or more — so overall, they have 2x-4x more space in which programs can run.

Being portable, mobile devices have less storage space than desktop computers. Most phones and tablets today have 16GB or 32GB, although amounts like 128GB are becoming more common; some phones allow you to add a microSD card, which in theory could expand storage by up to 2TB (even though such memory cards do not yet exist!). By comparison, desktop computers typically start at 1TB (1,024GB) and it is not unusual to have many more terabytes additional storage, with 12TB hard drives now affordable.

For this reason, programs written for portable devices must be very compact. It helps to close other apps if you need more RAM for opening large drawings in ARES Touch.

About Fonts. Some fonts are copyright by Microsoft for use on Windows. To get around this problem, you can specify which fonts to substitute by ARES. Many fonts look similar, and so the substitution works well, in most cases.

PROCESSORS AND GRAPHICS

Portable devices use a processor (called "ARM," short for *Advanced RISC Machines*) that is different from the ones used by desktop computers, which use Intel-compatible processors. ARM processors are very efficient, using far less power than Intel processors, which is important because portable devices run primarily on batteries.

Similarly, portable devices use a different method of displaying graphics (called "Mali") than desktop computers, which use graphics processors from AMD, Intel, and Nvidia. The reasons are, again, low memory and low power consumption: processing graphics with desktop computers can consume even more memory and more power than just running software itself. Then there is a third reason: portable graphics have not been optimized for CAD the way that desktop graphics have.

And so programs for portable devices have to be written for different processors and different graphics systems. There is one exception: some tablets form China are equipped with low-power Intel processors can run both Android and Windows, although a reboot is required to switch between the two operating systems.

Android runs on CPUs designed by ARM and Intel, but iOS runs only on CPUs designed by Apple.

ACCESSING FILES

Apps on portable devices access files in a manner similar to desktop computers. But, because mobile devices are by nature usually disconnected from high-speed networks, large files like CAD have to be handled differently. Files can be accessed from the device, via a USB cable connection to the desktop computer, or else by using a file sharing service like Dropbox.

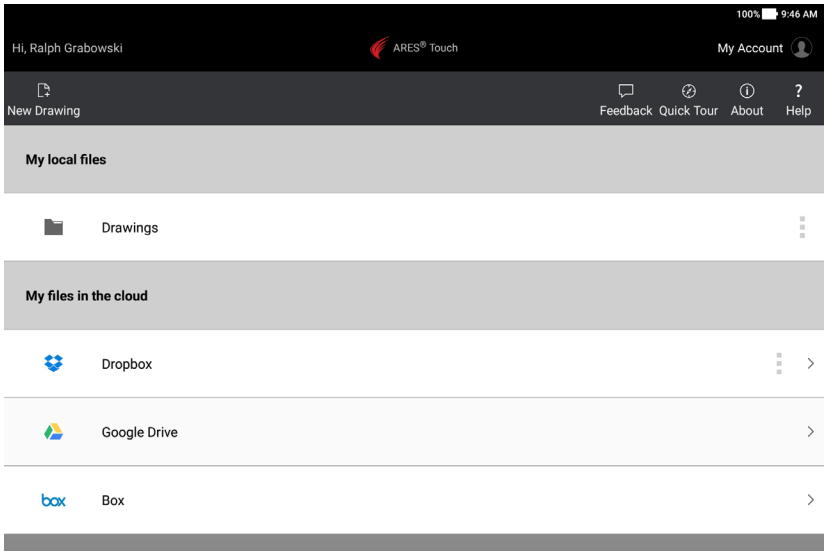
In this section, I show you how to operate ARES Touch. First, here are the parts of Touch's user interface:

TUTORIAL: STARTING TOUCH AND OPENING A DRAWING

In this tutorial you learn how to start Touch, navigate its user interface, and then enter commands and options. The instructions are the same for Android and iOS devices.

1. To start ARES Touch, tap its icon 

.
2. Notice that Touch starts in a file explorer screen. Here you open drawing files stored on the device and from file sharing services, such as Dropbox, Box, Google Drive, and iCloud (Apple only). The file manager is split into one section for local files (stored on the table) and one for cloud storage locations:



File manager for local and cloud storage

My Local Files (offline or **local)** — stores drawings that you need to access offline for when the device has no connection to the Internet, or else you do not want them synchronized to your desktop computer for whatever reason. You copy files to the tablet using a USB cable or wireless transfer utility, storing them in the device's folder named */ARES Touch/Drawings*. The files remained unsynchronized until you choose to do so.

My Files in the Cloud (online or **remote)** — stores drawings that you want to access online and want to have synchronized with other locations, such as your office's central file server. ARES Touch creates a folder into which you copy drawing files from the cloud, and you have the option to make them available offline. Note that some users cannot use cloud storage due to corporate security policies.

TIP ARES software can access drawings and other files through the following online services:

Apple iCloud (5GB free storage; for Apple devices only)
Box
Dropbox
Google Drive (17GB free storage)
Microsoft OneDrive (5-10GB free storage; built into Windows)
Microsoft OneDrive for Business
OnShape
Trimble Connect
WebDAV (for private cloud services)

Graebert may add other file sharing services at a later date.

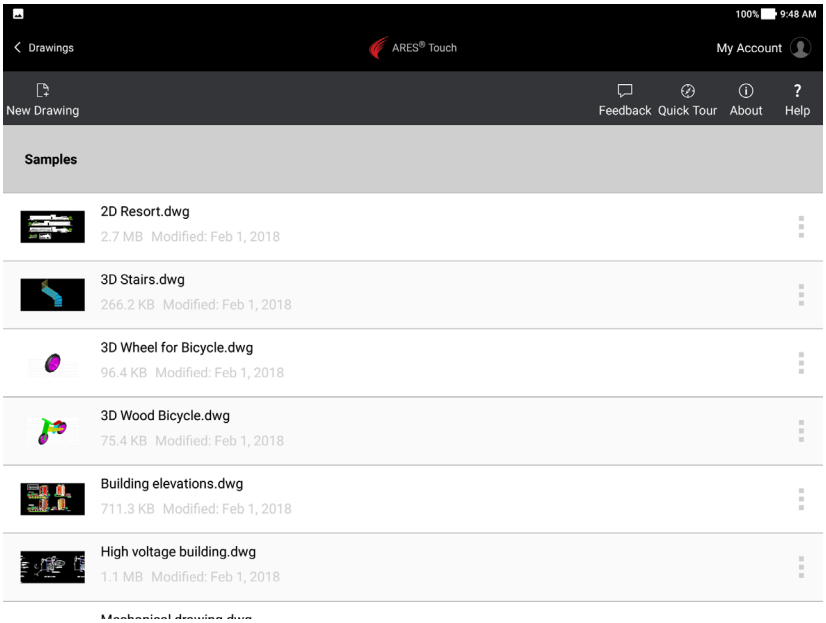
---

3. The installation includes a few 2D and 3D demo files. To access them:

   a. Tap **Drawings**.

   b. Tap **Samples**. Notice that the *.dwg* files are listed alphabetically with a thumbnail of each one.

   (To start a new drawing, tap the ⊕ NEW **New** button.)



Sample drawing files provided with ARES Touch

4. To open a drawing file, tap its name. For this tutorial, tap **2D Resort.dwg**. Wait a moment as Touch loads the drawing.

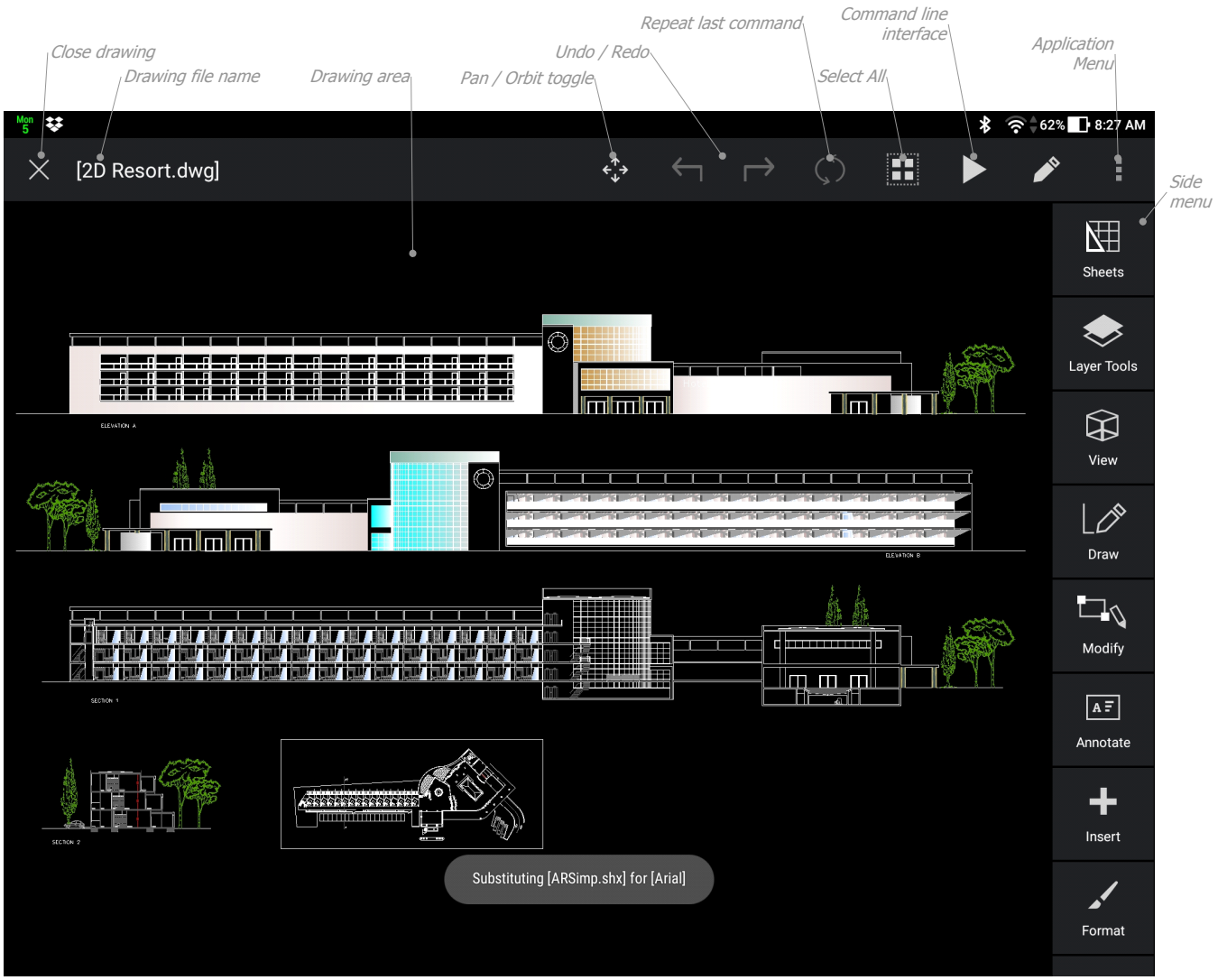
TIP ARES Touch itself does not limit maximum drawing size that it can open. Because mobile devices have smaller amounts of memory in which to operate, there is a limit to the largest or most complex drawing size that can be opened — unlike desktop computers, which use *paging* to overcome memory limitations.

It turns out that there is no easy way to determine the maximum drawing size. Complexity affects size, such as drawings with many hatch patterns. Naturally, the more memory (operating RAM) in your mobile device, the larger a drawing it can accommodate. Look for devices with 3GB or more RAM. Once the drawing is loaded into Touch, the more you zoom in, the faster it operates.

---

5. Once a drawing is open, notice drawing and the user interface that surrounds it. (See figure on next page.)

   Near the bottom, a message tells you that fonts are being substituted. This is because tablets and smartphones don't carry the broad line of fonts found on desktop computers; Touch makes an intelligent guess at replacing fonts with the most similar ones it can find.



Viewing Controls Through Touch

To change the view of the drawing, Touch relies on touch controls, like nearly all other mobile apps. (So as to not confuse verb touch with the name Touch, I'll use the word "tapping" from now on.) The tapping actions work the same for Android and iOS, and are as follows:

Tap — selects an entity in the drawing, or one of the buttons

Tap and hold — creates selection windows

Double-tap — zooms the drawing to its extents

Drag one finger — pans the drawing in Pan mode, or 3D orbits the drawing in Orbit mode; to switch between modes, tap the **Pan/Orbit** button on the main toolbar

Pinch two fingers — zooms out

Spread two fingers — zooms in

Tapping performs these operations much faster than moving the mouse into position. Tapping can also be faster for some drawing and editing commands, such as dragging a boundary on the screen during the PowerTrim command, or tapping an area to hatch.

WHY ARES RUNS ON MULTIPLE PLATFORMS

The ARES "trinity of CAD" software combines the best of what each platform offers. Desktop, mobile and cloud have synergy across all devices with Trinity.

A. ARES COMMANDER ON ON THE DESKTOP

For Graebert, ARES Commander for Windows, macOS, and Linux computers remains the flagship product. You keep using your laptops and desktop computers for the heavy work, creating and modifying drawings in 2D and 3D. Working on a computer offers the larger screen, full-size keyboard, mouse, and a familiar look and feel. It is obviously more comfortable than working with a mobile device.

- ARES Commander is fully installed on your computer, and so you can work offline like you always have.
- Graebert's unique licensing system allow you to work with your ARES Commander license on multiple computers, whether at the office, at home, or on the go.

At the end of the day, it is clear that you draw faster with the desktop than with mobile or cloud.

B. ARES TOUCH ON SMARTPHONES

ARES Touch is the extension of ARES Commander for smartphones and tablets. Graebert subscribers can install ARES Touch on not just one device, but on all their Android and iOS mobile devices at no additional cost. More than 150 viewing, drawing, and editing commands are available to read, modify, and annotate drawings on the go.

Although the features are the same for smartphones and tablets, their use cases vary according on your needs and the size of the device. Smartphones have smaller screens but fit into your pocket to follow you around all day long. Some popular use cases for ARES Touch on your smartphones include the following:

- Read and share drawings received by email, such as from a customer to validate a new project. The *.dwg* and *.dxf* attachments are opened in ARES Touch on your phone.
- Take drawings alone for offline use, or else download them as needed from cloud storage services. You have access to your projects at any time, without returning to the office or even anticipating you will need them.
- Cameras on smartphones effortlessly annotate drawings with pictures and recordings of your voice on site. Pictures and voice comments are inserted in the drawing with the **PictureNote** command at the location of the problem; as soon as your phone synchronizes the project, the picture and voice notes show up in ARES Commander back in the office.
- Using the **FreeSketch** command is like using a pencil on paper: pick a color and then use your finger to sketch on the smartphone's screen. Sketches are placed automatically on a dedicated layer so that they do not mess the rest of the content. This feature is intuitive enough that it can be used without any experience in CAD.
- Dimensioning and enquiry tools measure distances and angles not already shown in the drawing, which is better than using a ruler on paper or a PDF.

TIP The **QuickModify** command execute operations, such as Copy, Move, Rotate, and Scale, sequentially during the same selection.

Continued...

Action Bar

Along the top of the Touch window is the main toolbar, called the "Action Bar."

Above: *Fill size action bar for large screens, like tablets;* ***below:*** *shortened action bar for small screens, like phones*

From left to right, the functions of the buttons are as follows:

- **Close** the drawing; when you tap this button, Touch asks if you want to exit the drawing

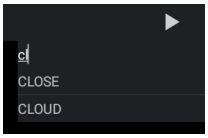
[2D Resort.dwg] **File name** of the open drawing (tapping it does nothing)

- **Toggle** between Pan and Orbit modes:
  - » **Pan** — drag one finger to move the drawing
  - » **Orbit** — rotate the drawing in 3D to view 3D models from all sides
- **Undo** the last command; you can undo all the way to the start of the drawing session (unlimited undo)
- **Redo** the last undo

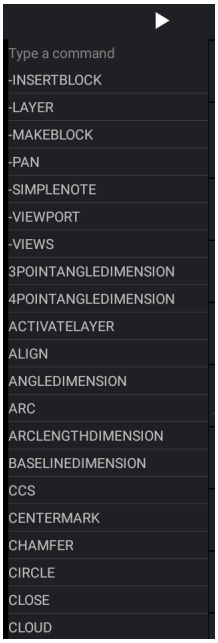
(On phones, most of the following commands are in the menu.)

- **Repeat** last command (like pressing the spacebar in ARES Commander)
- **Select All** selects all entities in the drawing; to un-select, tap **Finish** in the Select menu
- **Commands** button slides open a pop-out that lists all commands available in Touch. Scroll down the list to find a command name, or else type the name in the "Type a command" field.

  When you tap the "Type a command" field, the tablet's on-screen keyboard appears automatically. As you tap letters on the keyboard to spell the command's name, Touch shortens the list to show just the commands that begin with the letters you tapped — such as "cl" shown below. No need to enter the entire name: as soon as you see the command you want, just tap its name.



Finding a command name



Continued...

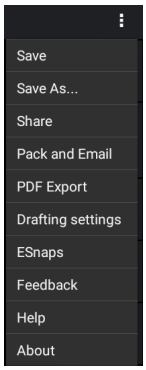
ARES TOUCH ON TABLETS

Tablet offer a larger screen while are still much lighter to carry on than a computer. They switch on faster than your computer, and so are great for showing projects (connected to a projector) and replace paper.

- » For surveys and inspections, tablets are good for looking at project drawings, inserting blocks, and rearrange content.
- » With more than 150 commands, ARES Touch sketches ideas and modifies any drawing on the go, before continuing the project back in the office with ARES Commander.
- » And tablets have all the sharing and annotation features found in smartphones, of course.

When you select a command, its input options appear in blue at the top of the drawing area, as described later.

- **Mode** toggle between Full, Simple, and View modes:
  - **Full mode** has all commands for drawing and editing. This mode is available for the first 30 days after installing the software, and when you pay the annual subscription fee.
  - **Simple Mode** is meant for viewing drawings and making simple annotation. This mimics the most common use of CAD apps, viewing and markup. This mode is available to users who don't pay. (To encourage users to pay, all commands from the full version are shown but are grayed out.)
  - **View Mode** is meant for viewing drawings; it minimizes the user interface so that the drawing area is maximized.
- **Application** menu slides down a menu with the following items:



Application menu

  - **Save** saves changes to the drawing to its original location, whether on the device or to the online storage from which it was opened, such as Dropbox.

TIP If you don't want to save changes to the drawing, then tap **x** (upper left corner) to exit without saving.

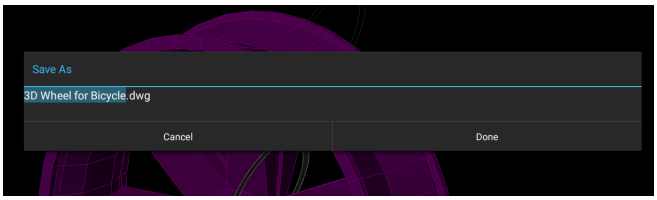
A NOTE FROM HISTORY

Running a CAD system on a mobile device is not new for Graebert Gmbh. In 2000, company owner Wilfried Graebert asked his son, "Can't CAD run on Windows CE?" Back then, Windows CE was a version of Windows that Microsoft provided for portable devices to counter the success of PalmOS.

"Of course," said his son, and wrote the mobile CAD program as his Masters thesis. No other CAD vendor did this. And so Graebert had the first DWG-based mobile CAD, known as SiteMaster, which today continues to be popular with surveyors. It has since turned into a service business that Graebert operates under another name.

The experience of packing a CAD program into a tiny memory footprint served Graebert well when writing ARES for Android and iOS devices 15 years later.

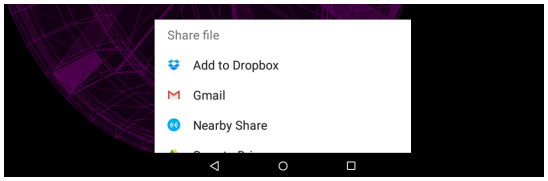
» **Save As** allows you to change the name of the drawing, before saving it.



Renaming a drawing before saving it

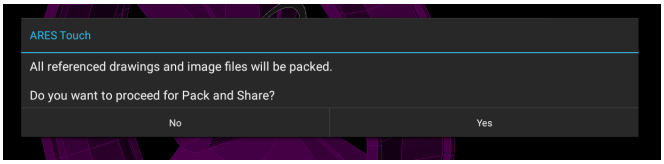
TIP Changing the name is useful when making revisions to drawings and you want a record of the changes.

» **Share** sends the drawing to apps installed on your device. This is a standard function on mobile devices, and so may repeat other functions in Touch.



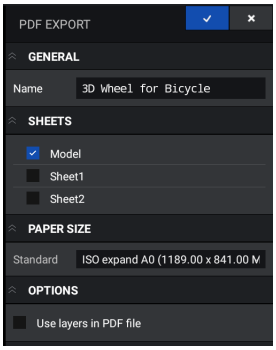
Options for sharing files

» **Pack and Email** collects the drawing file and all its support files (such as fonts and line types) in a ZIP file and then sends it to the email app.



Packing a drawing with all of its support files

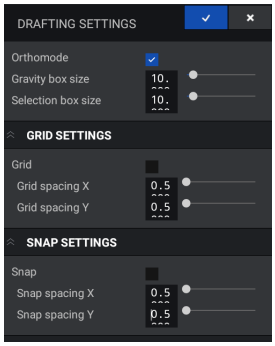
» **PDF Export** opens a dialog box with options for creating a PDF file of the drawing. After you tap the blue check mark, Touch asks you how you want to send the PDF file, such as by email or saved to Dropbox, just like the Share command above.



Options for exporting drawings as PDF files

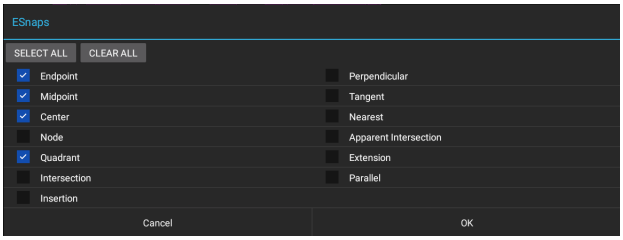
TIP Tap the blue check mark to accept changes and close the dialog box; tap the black **X** to cancel changes before leaving the dialog box.

» **Drafting Settings** opens a dialog box with settings for drafting. These settings for ortho, grid, and snap work the same way as in ARES Commander for the desktop.



Settings for drafting in ARES Touch

» **ESnaps** opens a dialog box for choosing which entity snaps modes should be on or off. Entity snaps work the same in Touch as in ARES Commander for the desktop.



Selecting active entity snaps

» **Feedback** sends an email with the device's email app to support@graebert.com.

» **Help** displays the help screens.

» **About** displays information about the app; tap anywhere to dismiss the dialog box.

APPLICATION TOOLBAR

The toolbar on the right side is called the "Application Toolbar." It gives you access to commands via icons, instead of by name. Commands are grouped by category, rather than listed alphabetically. The toolbar changes as you select a category, a command, and in some cases a sub-command.

The toolbar operates just like the sidebar menus from the old DOS and Unix days of CAD. Once you get used to it, it is a very efficient system for entering commands with your finger. For instance, here is how to start the **Polygon** command:

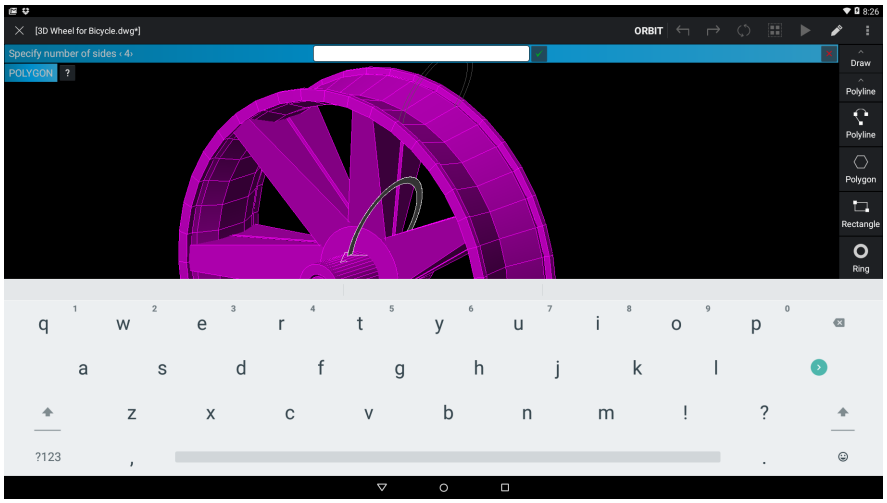
1. Tap **Draw**. Notice that the side toolbar changes to show commands for drawing entities.

Draw menu

2. Tap **Polyline**. Again, the side toolbar changes, but this time to list the kinds of entities that can be drawn as polylines.

Polygon sub-menu

3. Tap **Polygon**. Notice that the input options appear in blue at the top of the drawing area.



Input options for drawing polygons

TIP To back out of the menu system, tap the button at the top, the one with the up arrowhead: Draw

The catch is that you have to know that polygons are drawn as polylines, and so to look for the Polygon command under Polylines.

Command Options

As you select commands by name (from the **Commands** slide-out) or by icon (from the side toolbar), input options appear in blue at the top of the drawing area. See figure above for the Polygon command. The number of prompts you see depend on the needs of the command. The sequence of options matches the order in ARES Commander for the desktop.

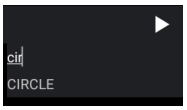
There is a difference, however, in the manner in which options are presented, depending on how you start the command:

- Start a command with the Commands slide-out — options are immediately displayed in blue
- Start from the side toolbar — first one or two options might be displayed by the toolbar

Let's see how options work with the **Circle** command, first with the Commands slide out, and then with the side toolbar.

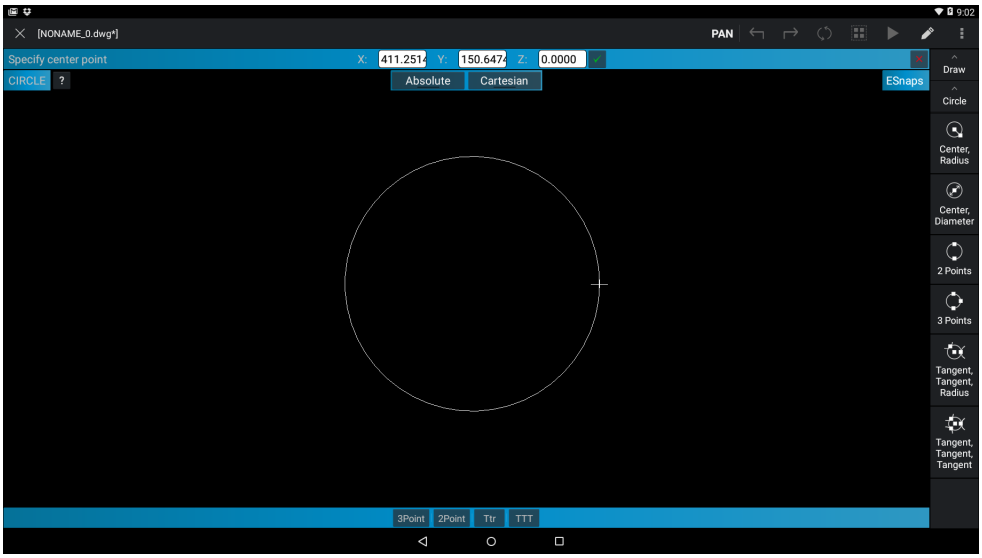
Entering Options with Commands Slide Out

1. Tap the > **Commands** button to access the slide-out.
2. In the "Enter a Command" field, begin tapping out the letters for **circle**.



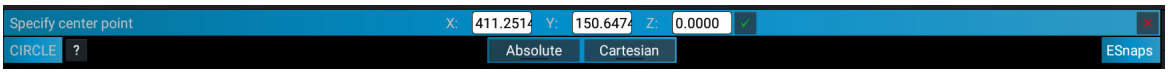
Finding the Circle command

3. Tap **Circle** in the list presented by Touch. Notice the blue bands that appear, one at the top of the drawing area and one at the bottom.



Options for the Circle command

The options you see vary by command. Here, for instance, is what they look like for the Circle command:



Inputting coordinates

On the first row of the blue bar are the following interface elements:

Specify center point — the prompt

X, Y, Z — specify the values of the x, y, and z coordinates. When you tap one of these, the on-screen keyboard appears so that you can enter values.

TIPS Use the **Absolute** and **Cartesian** buttons (found below the coordinates) to change to relative and polar modes.

Alternatively, you can tap a point in the drawing, possibly using entity snaps to make the tap accurate.

Green checkmark — tap to confirm the coordinates as the starting point of the circle

Red X — tap to cancel the command

On the second row are the following interface elements:

Circle — reports the name of the current command (tapping it does nothing!)

? — tap to display the help screen for this command.

Absolute — tap to toggle coordinate values between absolute and relative modes

Cartesian — tap to toggle coordinate system between modes:

- **Cartesian** coordinates are in the form of x, y, and optionally z
- **Polar** coordinates are in the form of distance and angle

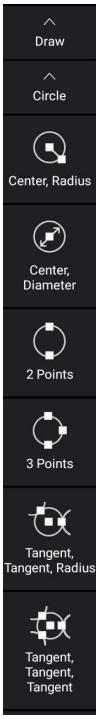
Esnaps — opens a list of entity snap modes, from which you chose the modes you want active or not.

A blue prompt bar appears along the bottom only when the command has options, such as the Circle command. It has options like 3Point, 2Point, Ttr, and TTT. When you tap one of these names on the prompt bar, the upper prompt bar changes to meet the new requirements.

Circle command options

Entering Options with Side Toolbar

When you start a command at the side toolbar, the blue prompt bars differ in one respect. To see this, tap **Draw** in the side toolbar, and then tap **Circle**. Notice that the side toolbar presents a list of circle construction types: Tap one.



Circle construction methods

COMMANDS IN ARES TOUCH

DRAW

Arc
Centermark
Circle
Cloud
Ellipse
Fillet
FreeSketch
Hatch
InfiniteLine
InsertBlock,
-InsertBlock
InsertImage
InsertShape
Leader
Line
MakeBlock,
-MakeBlock
PictureNote
Point
Polygon
Polyline
Polyline3D
Ray
Rectangle
Region
-SimpleNote
Spline
Tolerance
VoiceNote

FILE

Close
InstallPlugin
LoadApplication
LoadShape
PackAndShare
PdfExport
Save
Saveas
UninstallPlugin

DIMENSIONS

AngleDimension
ArcLengthDimension
BaselineDimension
ContinueDimension
DiameterDimension
EditDimension
EditDimensionText
HorizontalDimension
JoggedDimension
LinearDimension
MoveDimensionText
ObliqueDimension
OrdinateDimension
OverrideDimensionStyle
ParallelDimension
RadiusDimension
RebuildDimension
RelateDimension
ReplaceDimensionText
ResetDimensionText
SmartDimension
UnrelateDimension
VerifyDimensions
VerticalDimension
3PointAngleDimension
4PointAngleDimension

ID

GetArea
GetDistance
GetXy
Statistics

EDIT

Align
Chamfer
Copy
Delete
EditAnnotation
EditHatch
EditLength
EditPolyline
EditTolerance
Explode
Extend
MarkDivisions
MarkLengths
Mask
Mirror
Move
Offset
PowerTrim
PropertyPainter
QuickModify
Redo
Scale
SelectAll
Stretch
Trim
U
UndoN
Weld

SETTINGS

DraftingSettings
GeneralSettings
SetVariable

VIEWS

Ccs
DisplayOrder_Above
DisplayOrder_Back
DisplayOrder_Front
DisplayOrder_Under
DrawingBounds
HideView
ModelMode
-Pan
PanDown
PanLeft
PanRight
PanUp
PlanView
Rebuild
RebuildAll
Refresh
RefreshAll
Shade
ShadeView
SheetMode
ViewDirection
-Viewport
-Views
Zoom

LAYERS

ActivateLayer
DeleteLayer
FreezeLayer
HideLayer
HideLayers
IsolateLayer
-Layer
LockLayer
MatchLayer
ShowLayers
ThawLayers
ToActiveLayer
UndoLayer
UnIsolateLayer
UnlockLayer
ViewportLayer

PRECISION DRAWING

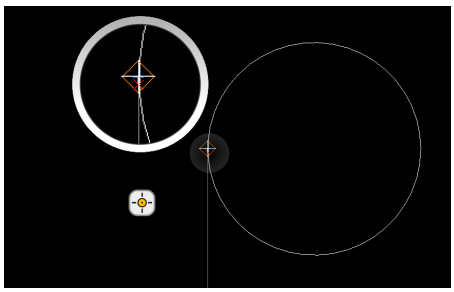
For drawing precisely with ARES Touch, the app offers the *Loupe*, an enlarged area of the drawing under your finger — along with the entity snap modes, snap spacing, and grid display that we have come to expect in CAD programs.

Using the Loupe

During drawing and editing commands a visual aid appears, called the "loupe" (a.k.a. magnifying glass). It shows an enlarged view of what is under your figure. The reason for it is that our fingers obscure the very area we need to see the most: the point where drawing and editing functions are taking place.

To activate the loupe, follow these steps:

1. Start a drawing or editing command. For instance, start the **Line** command (Draw > Line > Line).
2. Hold your finger down on a point where you want to draw or edit. In the example below, I am drawing a line that starts at the quad point (QUADrant entity snap) of the circle.

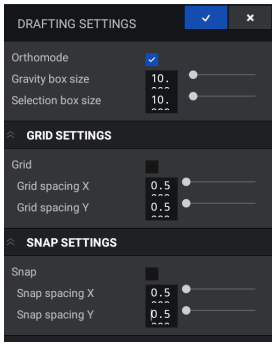


Viewing through the loupe

3. Move your finger to grab the entity snap at the circle's quadrant point. Notice that the loupe moves with it.
4. Release your finger from the screen.
5. Tap the screen near where the line should end, holding down your finger to bring up the loupe.
6. Drag your finger to locate the line's endpoint precisely, such as at another quadrant point of the circle.

Snap Spacing and Grid Display

ARES Touch lets you set the spacing for snap and grid distances. Access the dialog box through **Application Menu > Drafting Settings** action.



Drafting settings

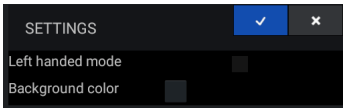
There are no shortcut keystrokes to toggle snap, grid, and ortho modes while drawing, unlike with desktop CAD software. This is because shortcut keystrokes — like Ctrl+B to toggle snap mode, F7 to toggle the grid, or holding down the Shift key to toggle ortho mode — are not possible with touch-oriented mobile devices.

Entity snap modes were discussed earlier. In brief, you can turn them on full-time with the **Application Menu > Esnaps** action, or else temporarily during a command by tapping the **ESNAPS** button.

Changing the Background Color

ARES Touch has only a few settings for changing the user interface. The default color scheme, for instance, is black for both the user interface and the background of the drawing area. You cannot change the color of the UI, but you can change the background color. Here is how:

1. Enter the **GeneralSettings** command. Notice the dialog box.

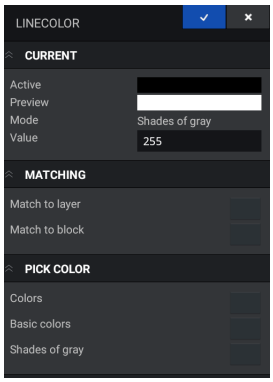


Settings dialog box

2. Tap **Background Color**.
3. Notice the new dialog box. In the **Pick Color** section, tap **Shades of Gray**. Notice the Select Color dialog box.

Select Color dialog box

4. Tap the color **White**. and then tap **OK**. Notice that the color next to **Preview** turns white.



Previewing color changes

5. Tap the blue check mark, twice. Notice that the background color of Touch turns white.

Recording Sketches, Pictures, and Voice Notes

For decades, engineers used cameras, voice recorders, and hand-written notes to document site conditions and problems with the construction. When I worked as a civil engineer, I would take photos with my 35mm film camera, and then rush them to the one-hour photo lab.

Today, however, photos, voice notes, and sketches can be embedded into the actual drawings. ARES Touch makes use of the cameras, microphone, and touch screens in tablets and smartphones as follows:

FreeSketch command records freehand sketches made on the touchscreen

PictureNote command takes, places, and annotates pictures at specific points in drawings

VoiceNote command records, places, and plays back voice notes

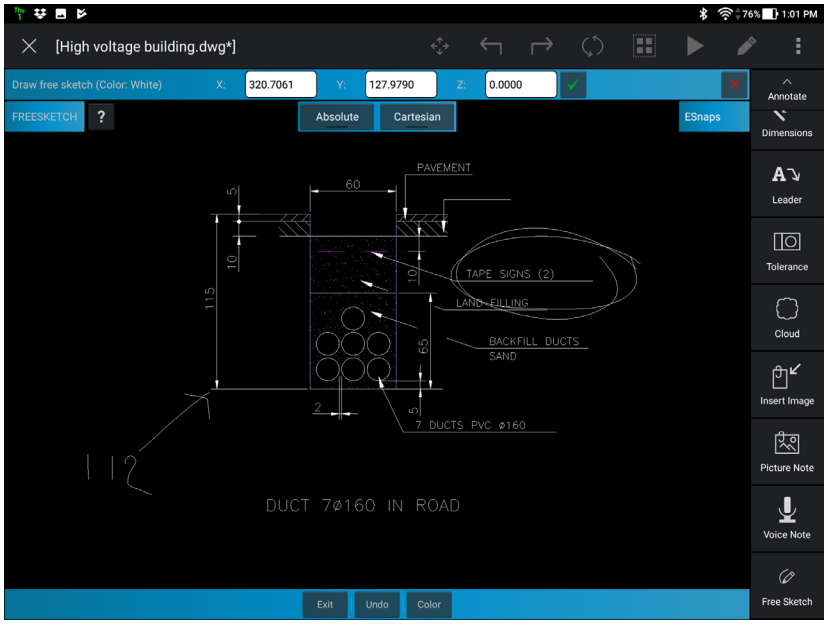
Sketches are placed on a layer named "Sketches," while picture and voice notes are placed on a layer named "Collaboration." This allows you to toggle them into and out of view easily. ARES Commander and Kudo don't have these functions, but they do display the images and sketches, and play back the recordings.

FREEHAND SKETCHING

The FreeSketch command lets you draw freehand on top of a drawing, to circle items, and scrawl hand-written notes. The sketches are made of polylines and are placed on their own layer, "Sketches." You can specify the color, and use entity snaps to start and end the sketch accurately.

To sketch on top of a drawing, use the **FreeSketch** command, as follows:

1. Start the FreeSketch command in one of these ways:
   - » Click the **Command** expander, and then enter the command name at the prompt:
     ```
     Type a command: freesketch
     ```
   - » Tap the **Annotate** button, and then the **Free Sketch** button.
2. Notice that you are prompted to start sketching:
   ```
   Draw free sketch (Color White): (Start drawing with your finger)
   ```



Sketching in Touch

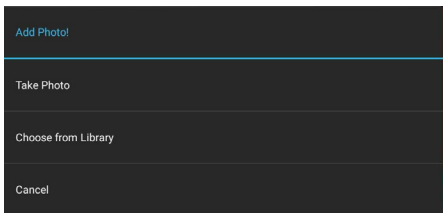
During freehand sketching, you can

- Change the color by tapping **Color** (at the bottom of the screen) and then choosing a different one.
- Start the sketch from a geometric feature by tapping **ESnaps** and then choosing an entity snap mode.
- Record all your sketches until you tap **Exit**.

TAKING PICTURES

The PictureNote command lets you place photographs in drawings, either taken with the tablet or smartphone's camera or opened from the picture library on the device. To place photographs in drawings, start the **PictureNote** command, and then follow these steps:

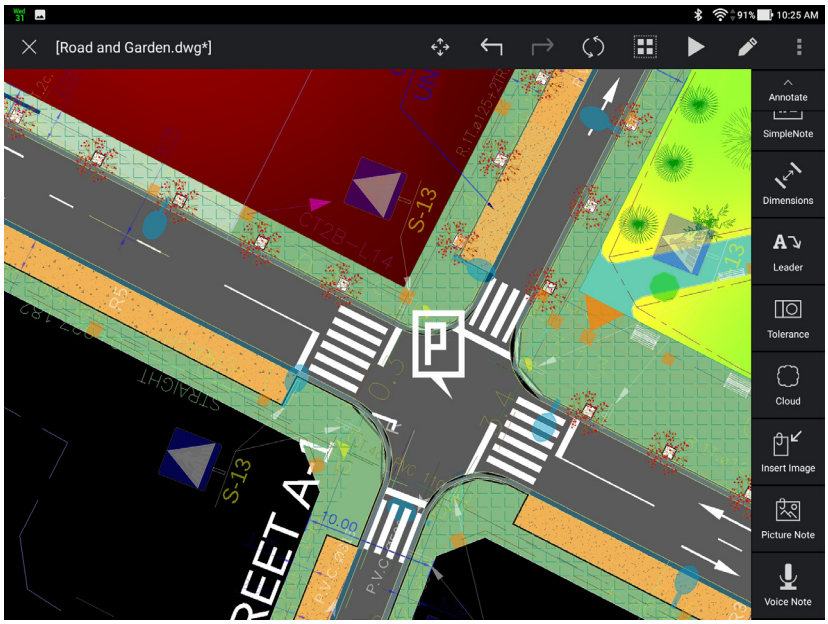
1. Start the PictureNote command in one of these days:
   - Click the **Command** expander, and at the prompt enter the command's name:
     ```
     Type a command: picturenote
     ```
   - Tap the **Annotate** button, and then the **Picture Note** button.
2. Notice that you are prompted to pick a point in the drawing.
   ```
   Specify PictureNote position: (Pick a point in the drawing, or else enter coordinates)
   ```
3. Choose whether you want to take a photograph with the device's camera, or choose a picture from the library of photos on the device.



Deciding where to get a picture from

   For this tutorial, choose **Take Photo**.
4. Notice that Touch disappears from the screen, replaced by the camera's view. Take the photograph, and then tap the checkmark to return to Touch.
5. Back at the prompt, enter a name for the photograph. You can leave this blank, if you wish.
   ```
   Specify name of PictureNote: (Enter a name, such as 'Pedestrian Sign')
   ```
6. Enter a description, or leave it blank:
   ```
   Specify description of PictureNote: (Enter some text, such as 'Existing northbound sign')
   ```

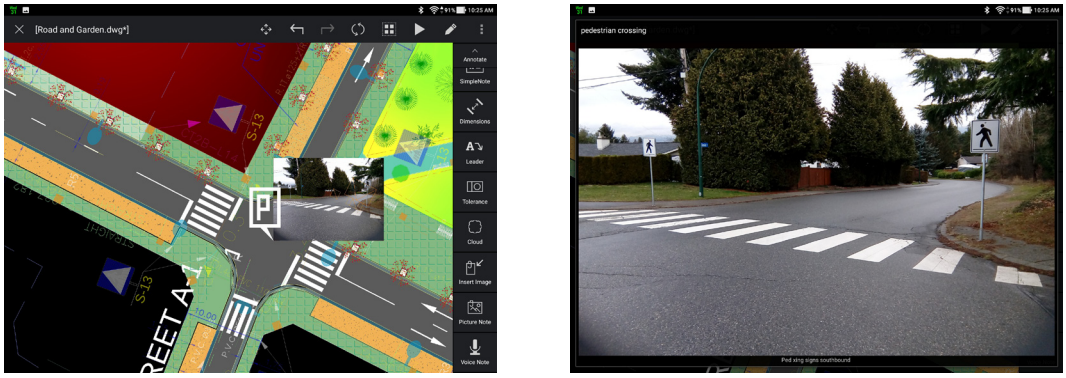
7. Notice that Touch places a 'P' icon at the insertion point of the photograph. It uses this icon to avoid cluttering the drawing with images.



Picture icon added to drawing

To view the image:

a. Tap the 'P' icon once to see a thumbnail image of the photograph.

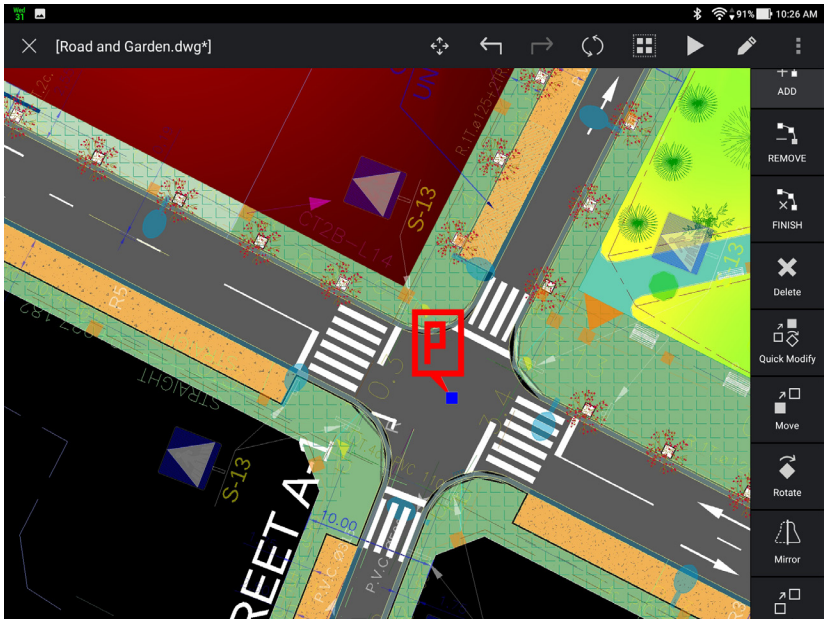


Left: *Thumbnail of photograph;* ***right:*** *Full size image*

b. Tap it a second time to see the image in a larger size.

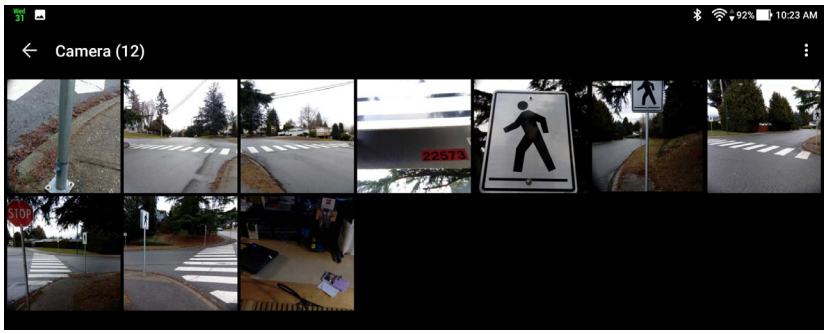
c. Tap the back button to dismiss it.

8. To edit the picture note, hold your finger down on it. Notice that it turns red, and that the toolbar shows a list of actions that can be performed on it, such as rotate or delete it. The blue grip moves the icon to a different location.



Editing the picture icon

When you choose **Choose from Library**, the tablet or smartphone opens the device's default image viewing app. (I happen to use QuickPic.) This method is faster when you take and then want to place a large number of pictures.



Viewing photos stored on the tablet

The photographs that you take from inside Touch are stored in the device's *My Local Files > Drawings* folder. Photos you attach from the library are copied and stored in the same folder as the *.dwg* files.

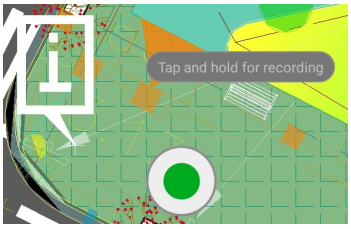
RECORDING VOICE NOTES

The **VoiceNote** command lets you record your voice using the tablet or smartphone's microphone, and the attaching it to a spot in the drawing. To record voice notes and place them in drawings, follow these steps:

1. Start the VoiceNote command in one of these ways:
   - » Click the **Command** expander, and at the prompt:
     `Type a command: voicenote`
   - » Tap the **Annotate** button, and then the **Voice Note** button.
2. Notice that you are prompted to pick a point in the drawing.

   `Specify VoiceNote position: (Pick a point in the drawing, or else enter coordinates)`
3. A green and white circle appears at the pick point; the prompt floats in a gray bubble near by:

   `Tap and hold for recording`



Starting recording the voice note

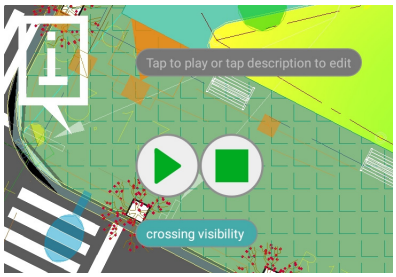
   For as long as you hold the green button, your device's microphone records your words.
4. When you are done speaking, let go. Notice that Touch prompts you to describe the recording:

   `Specify description for VoiceNote: (Enter a description or leave blank)`

Entering a description for the voice note

   Tap the green check mark when done.
5. To hear the recording, tap the 'I' icon. Notice the green playback icon, along with the floating prompt:

   `Tap to play or tap description to edit`



Listening to your voice

Audio files are stored in the drawing's extended entity data.

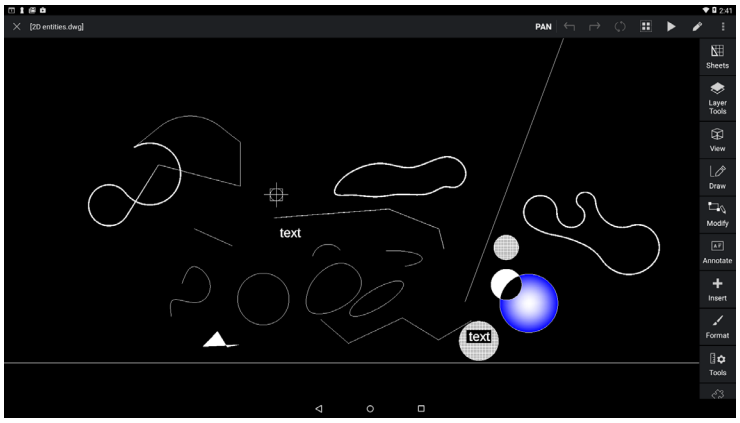
Compatibility with ARES Commander

Clearly the user interface between the mobile and desktop versions of ARES are different, because Touch has to take into account a smaller screen area and a finger-based touch interface. Underneath, however, Touch is fully compatible with Commander for the desktop; the exceptions are certain things that are missing, the topic of this section.

DRAWINGS

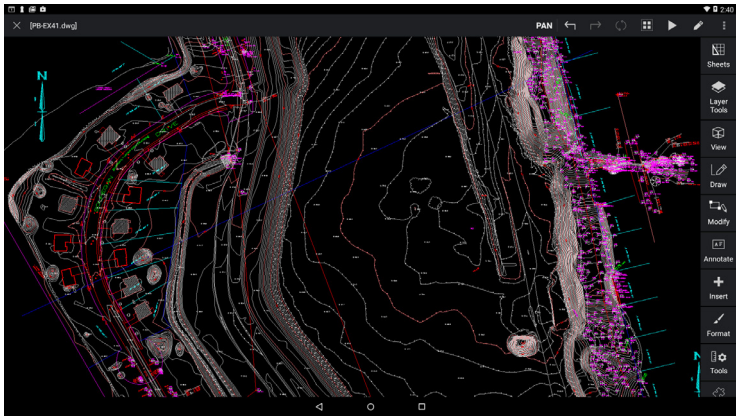
I tested the accuracy of ARES Touch using a set of *.dwg* files I used to test DWG compatibility among various CAD programs. I found that drawings created by ARES Commander are displayed 100% accurately in Touch — with some exceptions, as noted below. This is better performance than some competitors, such as Autodesk, in which its AutoCAD 360 app for mobile devices makes errors in displaying certain entities (at time of writing).

Shown below is one of the many test drawings I employ to gauge DWG accuracy.



Drawing for testing compatibility

Here are the exceptions I found. Very large drawings and very complex are not loaded, due to the memory constraint of the mobile device. I was able to open a 5MB drawing in Touch, and it ran slowly, processing the 124,000 entities. Going smaller, I opened the 1.5MB drawing (shown below) on a tablet with 4GB operating RAM. Touch was able to process the 12,990 entities in the drawing at a normal speed.



Dense drawing with large file size

OLE (object linking and embedding) objects are not displayed, because they are specific to Microsoft and its Windows operating system; OLE objects are also not displayed by the Linux and MacOS versions for the same reason.

Some fonts are displayed incorrectly, some complex linetypes are not displayed, and proxy objects are not displayed, such as from architectural software and generated drawings.

COMMANDS AND SYSTEM VARIABLES

ARES Commander for the desktop has over 500 commands and 700 system variables. Graebert Gmbh has indicated that it wants Touch to support many of those commands and their options. It makes sense that certain commands be excluded, such as those involving the text screen (there is none), 3D drawing and editing, and OLE.

ABOUT GRAEBERT LICENSE PLANS

When you purchase a license from Graebert, it allows you to operate all versions of ARES. There are several versions of license available (information current at time of writing):

- Permanent License (one-time fee): you buy the license once, and then use the ARES Commander desktop software forever on one computer at a time, whether Linux, MacOS, or Windows. When you want to upgrade the software, the cost is 50% of the list price. The license includes use of the full version of Touch and Kudo free for one year.
- Permanent License, with Support (one-time fee + annual payment): you buy the license once, and then pay annually for upgrades and support. The license includes the full version of Touch and Kudo free for each year in which you buy support.
- Subscription License (annual payment): you pay for the license every year, and then receive upgrades and support. The software stops working when you do not renew. This license includes the full version of Touch and Kudo free for a year and then for each year in which you renew the subscription.

The license operates differently, depending on the platform you use it with:

- Desktop license is portable, which means you can install ARES Commander on several desktop and laptop computers — Linux, MacOS, and Windows, it doesn't matter! — and then move the license to the one computer on which you need to use the program. You can operate just one copy of ARES Commander at a time.
- Android and iOS licenses operate the same as with other mobile apps: one license works for all portable devices registered in your name. If you have three Androids tablets, an iPad, and an iPhone, then you can run ARES Touch on all five at the same time.
- ARES Kudo license allows you to run the program on any supported Web browser. It can be used on an unlimited number of devices.

Volume license pricing begins at three users, and provides you with a lower cost. Flex network license pricing is 25% higher than the cost of a perpetual licenses.

See https://www.graebert.com/en/arescommander/price#pricing for current information.

Commands

At time of writing, Touch recognizes about 170 commands. The list is shown in the boxed text on another page in this chapter. (I keep saying "at time of writing," because Graebert updates the app every so often.)

Aliases

Touch does not support aliases (at time of writing). The installation includes a file with 749 aliases, and so support might be provided in a future release.

Aliases are important to making the app compatible with other CAD programs, such as AutoCAD and IntelliCAD.

Keyboard and Shortcuts

The on-screen keyboard and attached keyboards are limited to entering values in the prompt fields.

Touch does not support shortcut keystrokes, because shortcut keystrokes are not possible with touch-oriented mobile devices. They have no concept of function keys (such as F1) or Ctrl keys, such as Ctrl+A. Pressing **Enter** or spacebar has no effect. Pressing **Esc** exits the drawing. All other special keys, such as cursor keys and functions keys, have no effect on Touch.

Mouse Actions and Actions

Touch supports the touch actions listed earlier in this chapter. When you have a mouse attached to your device, then Touch respects the following button mouse actions:

Moving the mouse — moves the cursor

Clicking the left button — selects entities and command buttons

Dragging — moves entities that are selected; does not, however, stretch or perform other grip actions

Other buttons and roller wheel — no effect

For example, you cannot use the roller wheel to zoom, nor can you click the right button to display shortcut menus, or hold down the middle button to pan.

Portable devices do not support 3D mice, such as those from 3dconexxion, and so Touch does not work with them.

TIP You can connect and external keyboard and/or mouse though Bluetooth and USB connections. Android and iOS tablets come with the option of adding an external keyboard specific to the device. But these keyboards are not well designed for typing, and so there are alternatives:

Bluetooth. Because Bluetooth connects to just one device at a time, I recommend a Bluetooth keyboard that includes a touch pad, such as Samsung VG-KBD1500. A drawback to using Bluetooth with keyboards is that there is always a short lag between the time you press the key and the time the tablet reacts.

USB. When the USB port on a mobile device supports the OTG (on the go) standard, then you can attach a wired keyboard or mouse; use a USB hub to connect more than one device. A few tablets sport two or more full-size USB ports, allowing you to easily connect a wired keyboard and a mouse. Be careful, however, because external devices drain the battery faster.

System Variables

Touch supports system variables internally, and so with the **SetVariable** command you can enter sysvar names and change their values.

You cannot, however, list variables with the command's **?** option, and so I cannot easily list all variables names that are supported by Touch.

SUPPORT AND EXPORT FILES

No DWG is an island. It requires support files that define fonts, aliases, hatch patterns, and so on. At time of writing, Touch included the following support files:

Support Type	Support Files	Purpose
Command aliases	*Alias.xml* file	Maps Touch command names to ARES Commander names
Text fonts	*.shx* and *.ttf* files	Defines the fonts used by text in drawings
Font mapping	*Fonts.fmp* and *FontConfig.xml*	Maps Touch fonts to fonts used by other CAD programs
Hatch patterns	*.pat* files	Defines hatch patterns
Linetype patterns	*LTypeShp.shx* file	Defines complex linetype patterns
LISP routines	*.lsp* and *CoLispBase.xml* files	Defines LISP routines and maps LISP function names

TIP You can access these files with a file manager on Android devices; Apple does not allow users to do this. Sometimes tablets and smartphones come with a simple file manager installed by the manufacturer. If not, there are dozens available free on Google's online Play store. I use File Manager from Cheetah Mobile.

Touch reads and writes *.dwg* and *.dxf* files. The sole export format is PDF, at time of writing.

APIs

ARES Touch supports some of the APIs found in the desktop — C++, LISP, Tx, and DCL; no other CAD app does this.

API	Purpose
C++	Programming language used with Tx
DCL	Dialog control language for designing dialog boxes used with APIs
LISP	Interactive programming language; compatible with LISP routines written in AutoCAD, IntelliCAD
Tx	Teigha API from Open Design Alliance; compatible with ARX routines written with AutoCAD

To load *.lsp* functions into Touch, use the **LoadApplication** command. To load plug-ins, use the **Install-Plugin** command.

Graebert provides a utility program that converts dialog boxes from DCL format (used by ARES Commander) to the format required by Android and iOS. Contact support@graebert.com for more information.

Notes

CHAPTER 12

ARES Kudo for Web Browsers

In This Chapter

- Understanding how Web apps differ from desktop programs
- Touring the user interface of ARES Kudo
- Connecting Kudo with cloud storage services
- Learning about the differences between ARES Kudo and ARES Commander

ARES Kudo is the version of ARES that runs in Web browsers. It is a universal CAD program, because it operates on any device — desktop computers, laptops, tablets, and other kinds of hardware — that run supported Web browsers, such as recent versions of Google Chrome and Mozilla Firefox.

Graebert Gmbh designed Kudo to be part of a CAD-anywhere concept. With their ARES CAD software, it is possible to draw and edit DWG files in Web browsers (using Kudo), on tablets (using Touch), and with desktop computers (using Commander) on a single license. The three platforms also share the same API and a similar user interface.

Like ARES Touch from the last chapter, Kudo is more than a DWG viewer or markup app: a future release will also draw and edit DWG files without translation. Kudo runs in the cloud and stores your files in the cloud, automatically.

In this chapter, you learn how to operate Kudo in Web browsers, which is a somewhat different experience than running regular CAD on desktop computers. As Kudo is software still being developed, you will see the phrase “at time of writing” often, giving you hints about the additional functions future releases will provide.

For instance, Kudo is the technology used by Onshape Corporation for its drawing tab in its online mechanical CAD design program. See https://www.graebert.com/blog/product-news/ares-kudo-onshape-dwg-online-cad/ to learn more about how Kudo is used in Onshape.

KEY TERMS IN THIS CHAPTER

Cloud — multiple remote servers that store data and run programs

Hop — the route data takes between two switches on the Internet, where switches determine the best route

HTML — hypertext markup language used to define the look of Web pages

HTML5 — latest version of HTML that allows software like CAD to run in Web browsers

JavaScript — programming language used with Web pages

Latency — delay in reaction time as data moves through the Internet

One-truth — the most up-to-date version of a drawing, when there is more than one version

Panel — a palette that cannot be moved about the screen

Permissions — determines what you can do with a drawing, such as only view it, or edit it; also known as "role"

Tracemap — shows the route over which data travels between two point in the Internet

WelGL — graphics technology that allows 3D models to be displayed quickly in Web browsers (OpenGL for computers)

ABBREVIATION

F5	Key that refreshes the screen in a Web browser

Starting and Running Kudo

To run the ARES Kudo CAD program online, there is nothing to install. Starting it is like entering an address in a Web browser: https://kudo.graebert.com.

The drawing as presented in a tab of the Web browser. You gain access to Kudo under two conditions:

- When you receive a link from another Kudo user to view a drawing; you can view the drawing or else edit it, if you have editing rights
- When you pay a subscription fee for ARES Commander, which includes free access free to Kudo (as well as to ARES Touch for portable devices)

STARTING KUDO

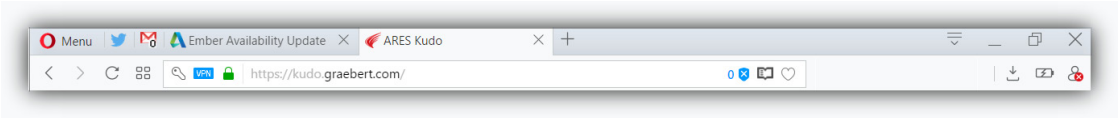
To run Kudo, you need the following items:

- A computer, laptop, or tablet running a compatible Web browser
- An internet connection
- A cloud storage service, such as Box, Dropbox, or Apple iCloud

That's all! To start the Kudo CAD program, follow these steps:

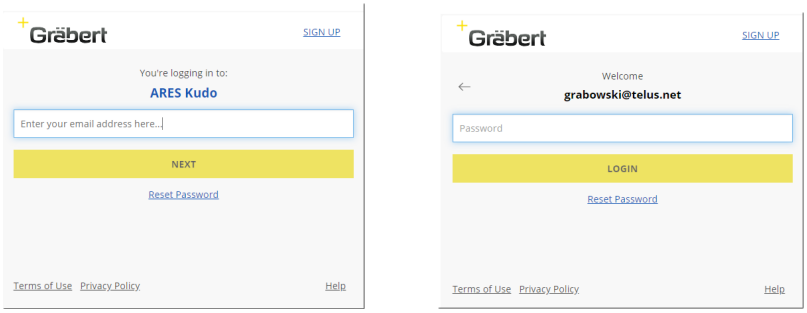
1. Open a Web browser, and then enter the following address:

   `kudo.graebert.com`



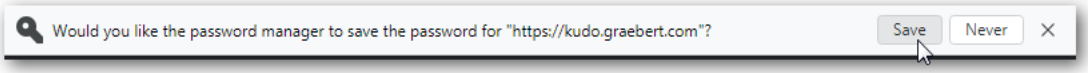
The 's' in https indicating that the Web connection is secure by encrypting communications between your computer and the distant server running Kudo.

2. Kudo loads into the browser page, and then asks you for your email address, and then your password. The login you use here is the same one you use for your Graebert account. By having that single Graebert login, you can run Kudo on any computer or tablet.



***Left:** Entering your email address; **right:** ...and your password*

The first time you enter this information, your Web browser may offer to save the password. I suggest that you allow it, as this makes getting into Kudo faster the next time.



Allowing the browser to save the password

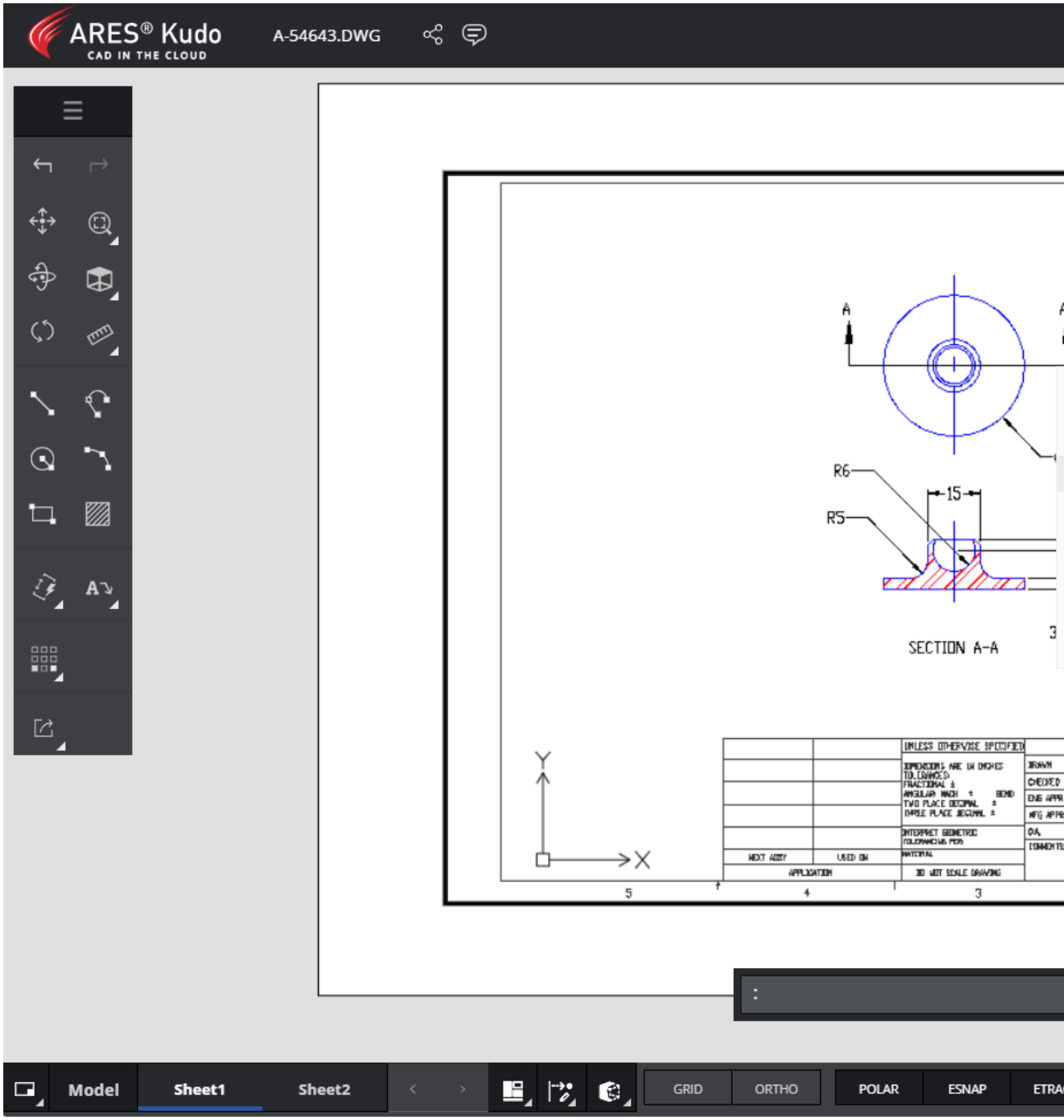
If, however, you are working on sensitive information or with someone else's computer, *do not* allow the browser to save your password data.

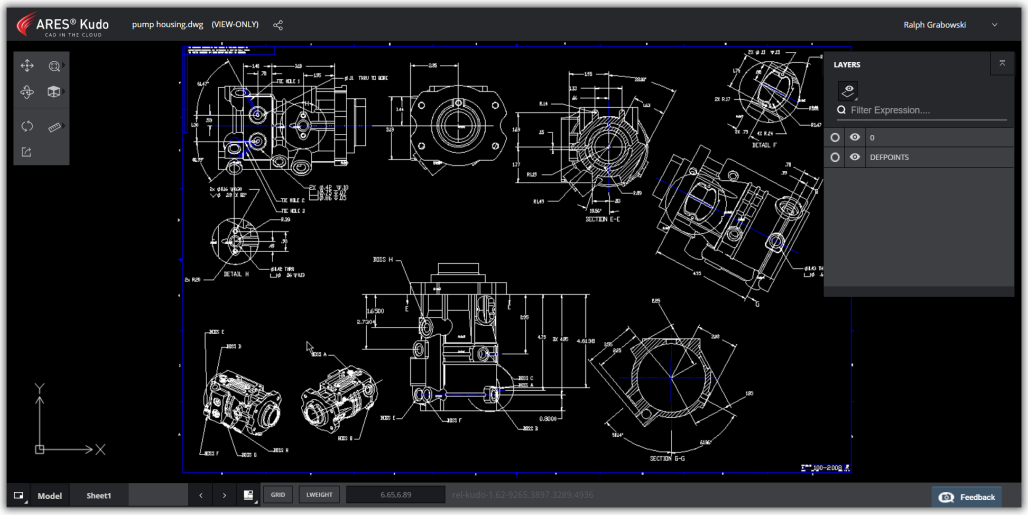
TWO VERSIONS OF KUDO

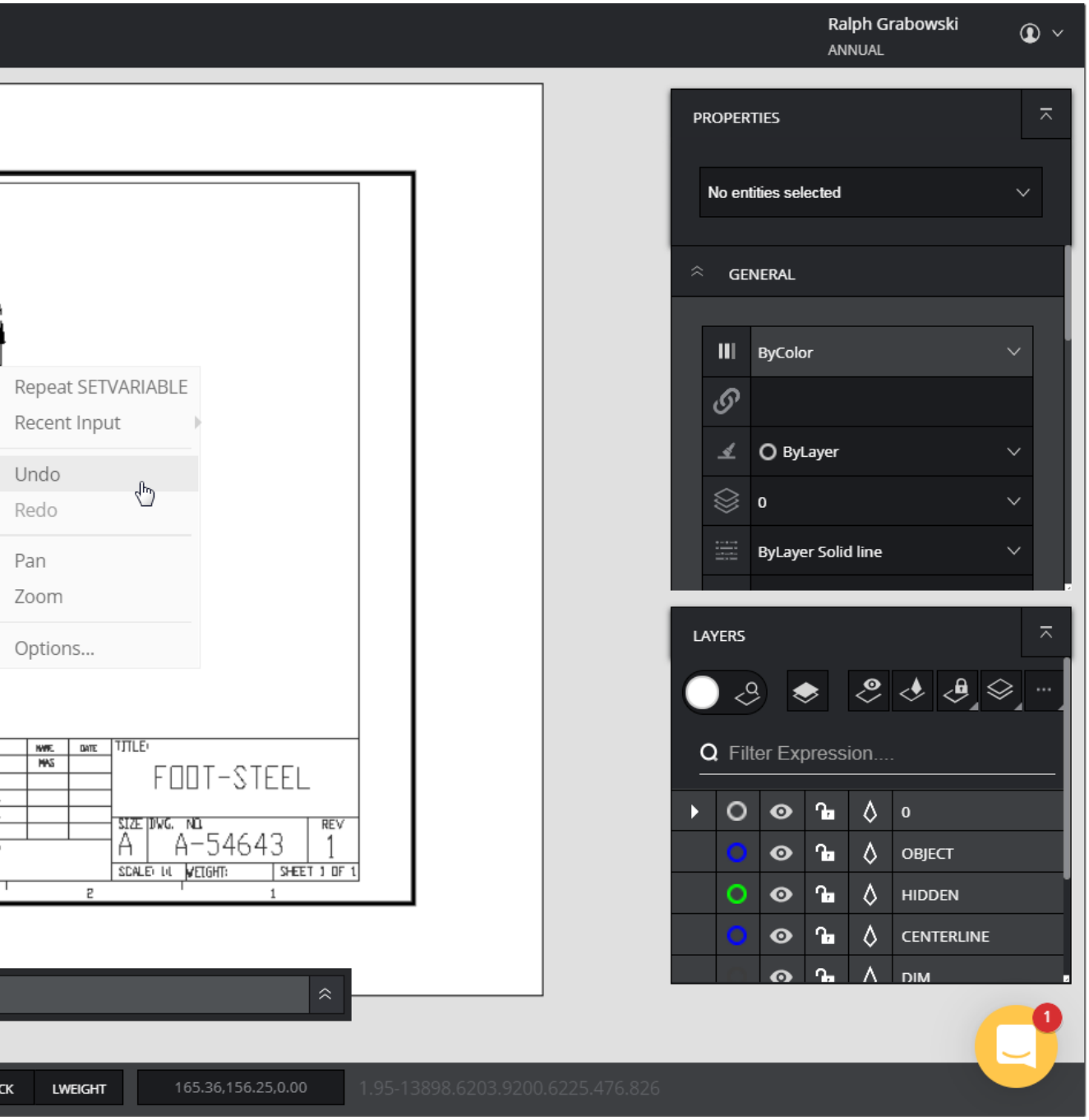
There are two versions of Kudo, as shown on this spread. One edits drawings, and requires payment through an annual license that also includes ARES Commander and Touch. (See figure below.)

The other version of Kudo only displays drawings, but can be used by anyone for free. It is identified by the "View-Only" tag in the upper left area of the screen. (See figure at right.)

***Below**: ARES Kudo in full editing mode; **right**: ARES Kudo displaying a 2D drawing in view-only mode*




ARES® Kudo
pump housing.dwg (VIEW-ONLY)
Ralph Grabowski
LAYERS
Filter Expression....
0
DEFPOINTS
Model
Sheet1
Feedback

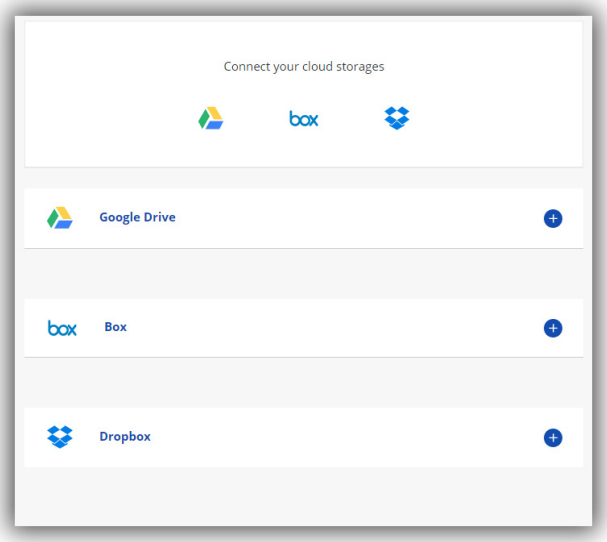

Ralph Grabowski
ANNUAL
PROPERTIES
No entities selected
GENERAL
ByColor
ByLayer
0
ByLayer Solid line
LAYERS
Filter Expression....
0
OBJECT
HIDDEN
CENTERLINE
DIM
Repeat SETVARIABLE
Recent Input
Undo
Redo
Pan
Zoom
Options...
TITLE:
FOOT-STEEL
SIZE
A
DWG. NO.
A-54643
REV
1
SCALE: 1/1
WEIGHT:
SHEET 1 OF 1
LWEIGHT
165.36,156.25,0.00
1.95-13898.6203.9200.6225.476.826

TIPS To easily restart Kudo, bookmark the "kudo.graebert.com" address in your Web browser.

If Kudo says that your Web browser is not supported, this is because it needs to understand HTML5, which older Web browsers do not.

Kudo sometimes cannot connect if your computer or Web browser is using VPN (virtual private network) software.

3. Click **Log In**. Notice that the "Connect your cloud storages" screen appears. If you have accounts with a service like Google Drive, Dropbox, and/or Apple iCloud (for iOS and MacOS only), then this is where you connect Kudo with online file storage.



Some of the cloud services supported by Kudo

Connecting Kudo with Cloud Storage

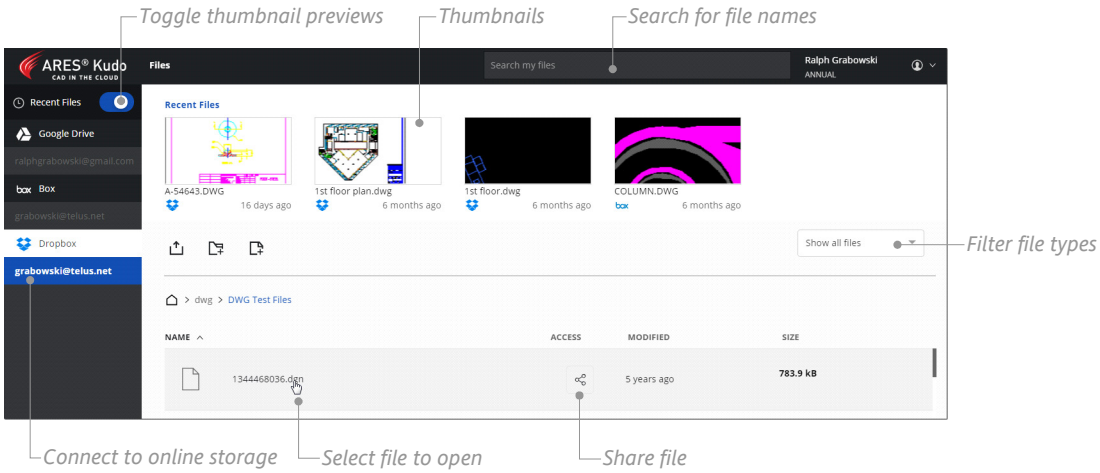
You do need to connect Kudo with at least one online storage service (a.k.a. cloud storage) the first time you use it. Kudo only works with files stored on the cloud (at time of writing). Kudo supports all the same services as ARES Touch:

- Apple iCloud (5GB free storage; for Apple devices only)
- Box
- Dropbox
- Google Drive (17GB free)
- Microsoft OneDrive (5-10GB free; built into Windows)
- Microsoft OneDrive for Business
- OnShape
- Trimble Connect

Graebert might support additional services in the future.

In the following tutorial, you connect Kudo with Google Drive cloud storage, because that's one of the ones that's free. (If you do not have a Google account, first create one before starting the tutorial at https://accounts.google.com/signup.)

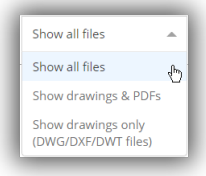
QUICK TOUR OF CLOUD STORAGE EXPLORER



Recent Files **Toggles thumbnails** — turns the display of the most recently-accessed file on and off

Search **Search for file names** — searches for file names across all cloud services

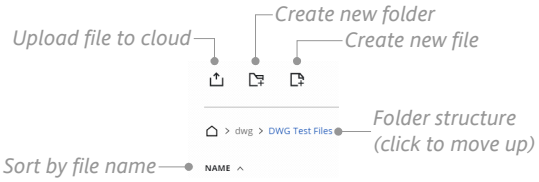
Show all files **Filter file types** — specifies the types of files to display: DWG, DXF, DWG, and PDF formats



Click to open — opens the file in Kudo; the view-only version can open only one drawing at a time

Click to share — opens dialog box for specifying how and with whom the file can be shared

FILE DETAILS



Upload file — uploads file from your computer to the current cloud service

Create new folder — creates a new folder in the current folder, on the current cloud service

Create new file — creates a new drawing by prompting for a file and a template to use (not in view-only)

Home — moves to the topmost folder

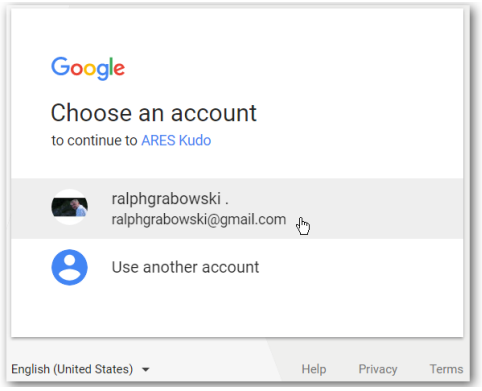
NAME **Click to sort** — sorts files by name, permissions, size, date when you click the headers; click again to reverse the order

Follow these steps to connect Kudo with a cloud storage service:

1. In the Connect Your Cloud Storage window, click the **Add** icon next to the name of a storage provider. For this tutorial, click the **Add** icon next to **Google Drive**.

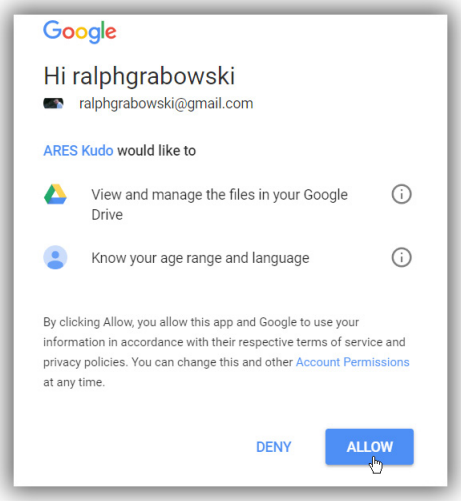
Choosing to connect to Google Drive

2. Notice that Google displays its account screen. This is the name of the account that will be linked to Kudo. Choose an account from the list provided.



Choosing a Google account with which to link Kudo

3. Wait a moment while Google learns from Kudo the kinds of access it will want from Google.

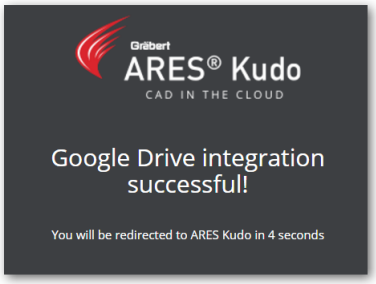


Allowing Kudo to access data in your Google account

Kudo will want from Google the following data:

- **View and manage files** — Kudo will be able to access all of the files stored on your Google Drive account, and not just *.dwg* files. "Manage" means to upload, update, search, download, delete, and change the permissions and meta data of files.
- **Know your age range and language** — the age range is needed in case some jurisdictions prevent online or software access to under-aged users, which could be under 17 and/or 18-20.

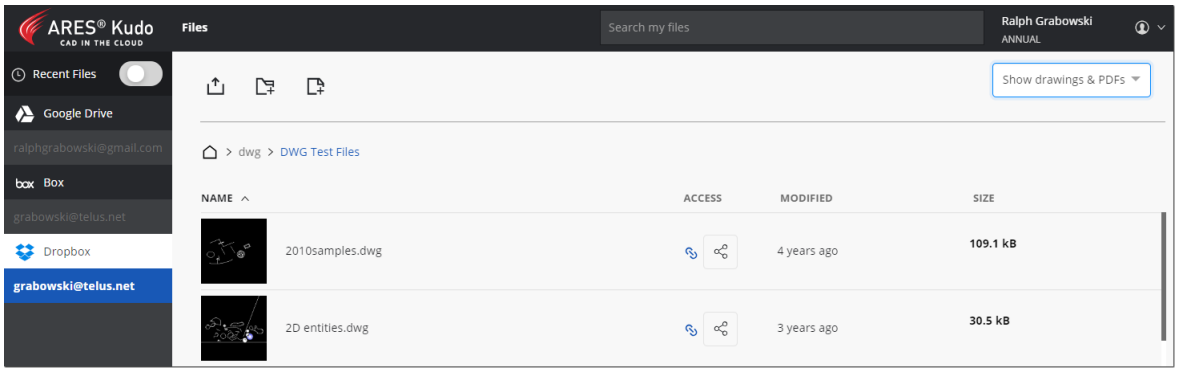
4. Click **Allow**. (If you click **Deny**, Kudo cannot access files from Google Drive.) Wait a moment while the connection between the two accounts is completed.



Success in linking Kudo with Google Drive

With these steps completed, Kudo can now open and save files that you store on Google Drive. You only ever have to go through theses steps this again to connect a different online storage provider, such as Dropbox or Box.

In the figure illustrated below, I show Kudo's cloud storage explorer with three cloud storage services connected; notice their names listed at the left side of the screen.

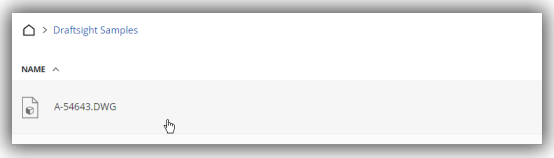


Kudo connected to Google Drive, Box, and Dropbox

Once two or more services are set up, you switch between them by clicking your *account name*, and not the same of the service! That's because you can have more than one account name associated with a service. In the figure above, I need to click on grabowski@telus.net and not Dropbox.

OPENING DRAWINGS

To open a file, click its name in the file manager.



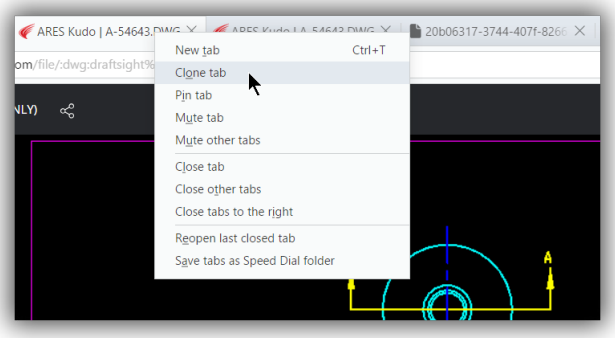
Selecting a file in the file manager to open it.

Kudo opens files in the following formats; more may be added later.

Extension	Format	Purpose
DWG	Drawing file	Used by many other CAD programs, such as ARES Commander
DWT	Template file	Used to start new drawings with preset properties and drawing elements
DXF	Drawing interchange file	Used by some CAD and CAM systems to exchange drawings
PDF	Portable document format	Used to view documents and drawings

View-Only Mode

The view-only version of Kudo opens drawings from the cloud; it cannot open them when they are stored on the computer. It cannot start new drawings, and it cannot open more than one drawing at a time. You can, however, open multiple instances of view-only Kudo to view and edit additional drawings — one drawing in each instance.



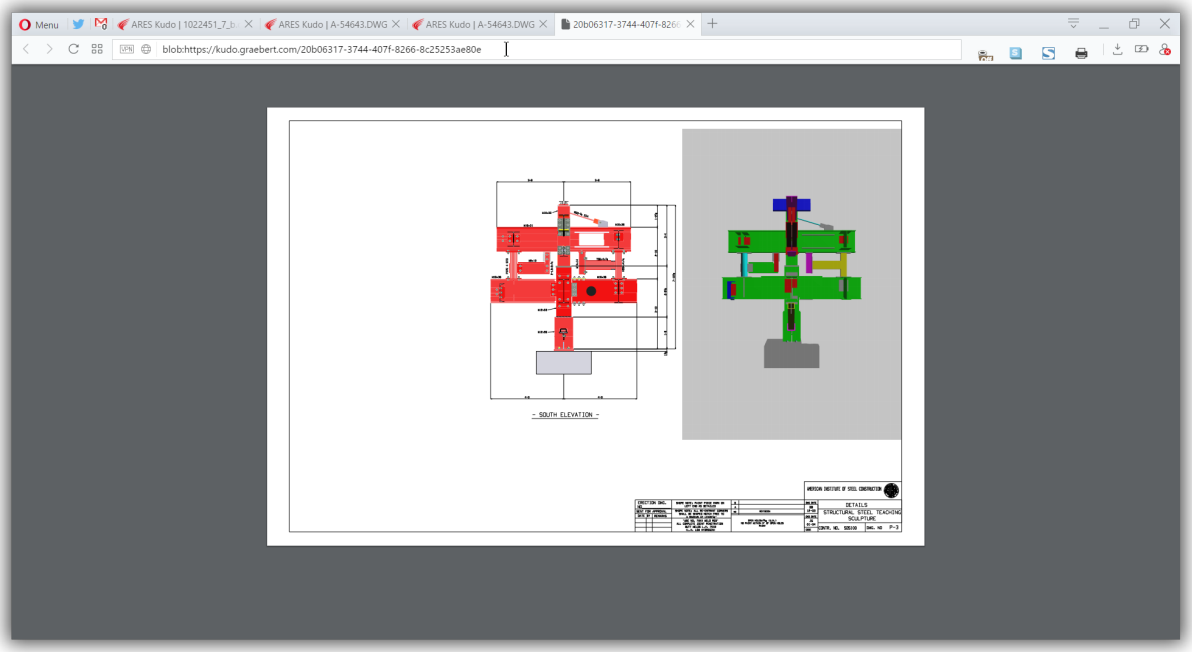
Cloning a tab to start another instance of Kudo

The easiest way to do start additional instances is to right-click the Kudo tab in the browser, and then choose **Clone Tab**. View-only Kudo cannot save drawings, but it can export drawings to PDF format.

PDF Files

When you open a PDF file, it opened in your computer's default PDF viewer — it is no opened in Kudo. The PDF file opens in a new tab of your Web browser or else in a stand-alone program, such as Adobe Acrobat, depending on how your computer is set up. You cannot edit PDF files, nor copy elements from them for pasting into Kudo; you can only view PDF files for reference.

The figure below shows Kudo running in three tabs of a Web browser, two with the same drawing. The topmost tab shows a PDF file opened from Kudo's file manager.



Viewing a PDF file with a browser's built-in PDF viewer

Opening Another Drawing

To open another drawing:

1. Click the **ARES Kudo** logo found in the upper left corner of the Kudo window.

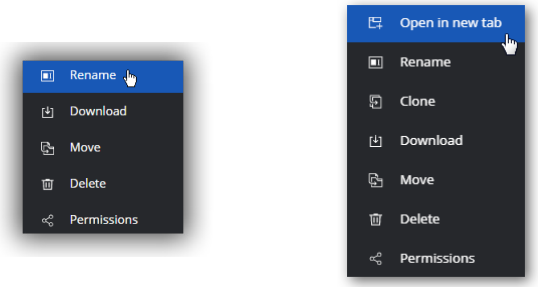
Clicking the logo takes you back to the file manager

2. Notice that you return to the cloud storage explorer. Choose a file, and then open it.

The newly-opened file open in a new tab. (If you are running view-only Kudo, then the new drawing replaces the former one.)

File Operations

Kudo provides a number of file operations in the cloud storage explorer. Right-click a file name to see a for shortcut menu of operations suitable for the selected fileL



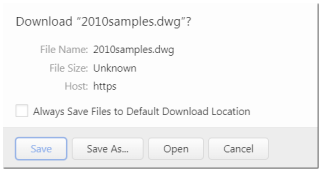
Left: *Shortcut menu in view-only version;* ***right:*** *...and in full version*

Open in a new tab — opens the file in Kudo running in a new tab of the Web browser

Rename — changes the name of the file

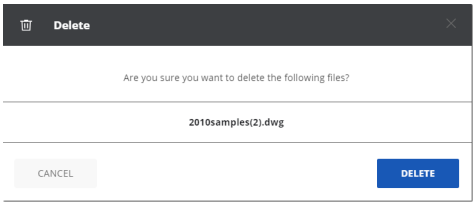
Clone — makes a copy of the file, giving it a suffix of (1)

Download — saves a copy of the file to your computer in a folder that you specify:



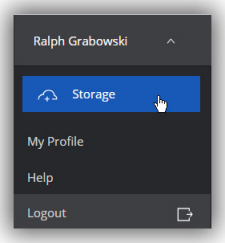
Move — moves the file to another folder in the same cloud service

Delete — erases the file, after you confirm the action:



Permissions — creates a link so that others can view or edit the file (more later in this chapter)

TIP To access the "Connect Your Cloud Storages" screen again, click your name (found in the upper right corner), and then choose **Storage**.



Walking Through Kudo's User Interface

After you select a drawing from the cloud explorer, Kudo displays its CAD interface.

In designing the user interface of Kudo for browsers, Graebert attempted as much as possible to mimic the look of its CAD app for tablets, Touch. (The user interface of ARES Commander for computers was also tweaked to look like Touch.)

The point to unifying the UI among the three programs is to make it easier for users to work with them. All three roughly look the same, wit the UI of all three programs offering toolbars, palettes, status bar, right-click menus, and so on. The primary difference in Kudo and Touch from ARES Commander is that they have no traditional menu bar, and that they are fully touch-aware.

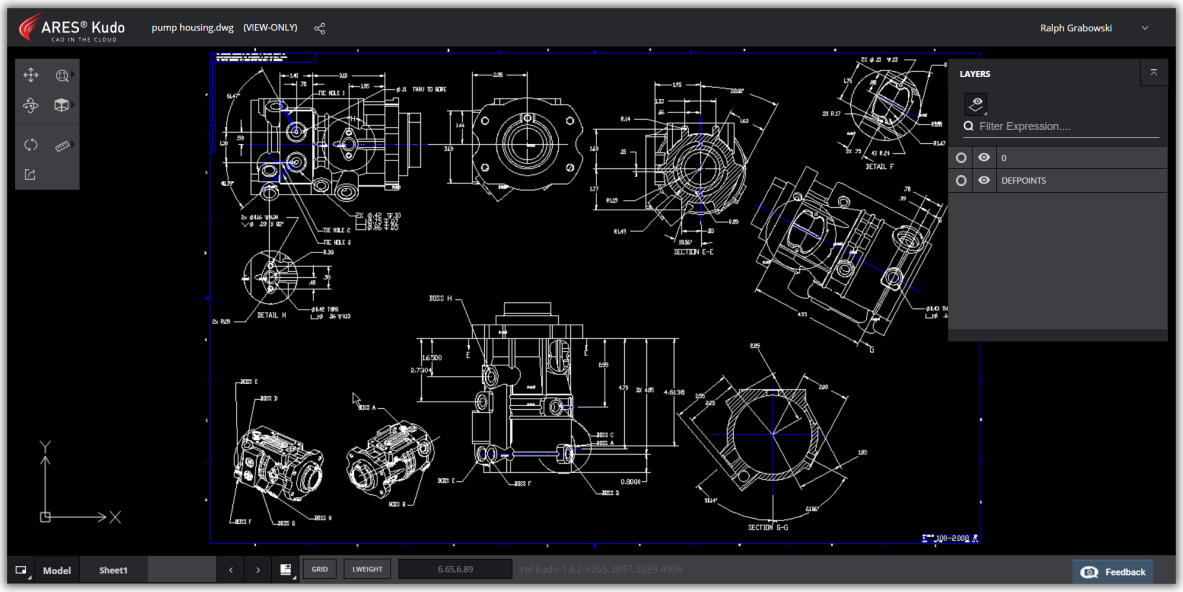
AFTER THE DRAWING IS OPEN

Kudo opens files in one of three interfaces:

View-only Interface — in which you view and print CAD drawings, but not edit them; a "View-only" tag accompanies the file name: architectural.dwg (VIEW-ONLY) (see figure below)

Full Editing Interface — in which you edit, view, and print CAD drawings

"PDF Interface" — in which you view and print PDF files, which are opened in a separate PDF viewer program



Kudo running with its view-only interface

ELEMENTS OF THE UI

The user interface of Kudo is a bit different from a desktop CAD package like ARES Commander. We'll look at the UI starting from the top and then working our way to the bottom of the window.

Title Bar

From left to right, the title bar offers the following functions:

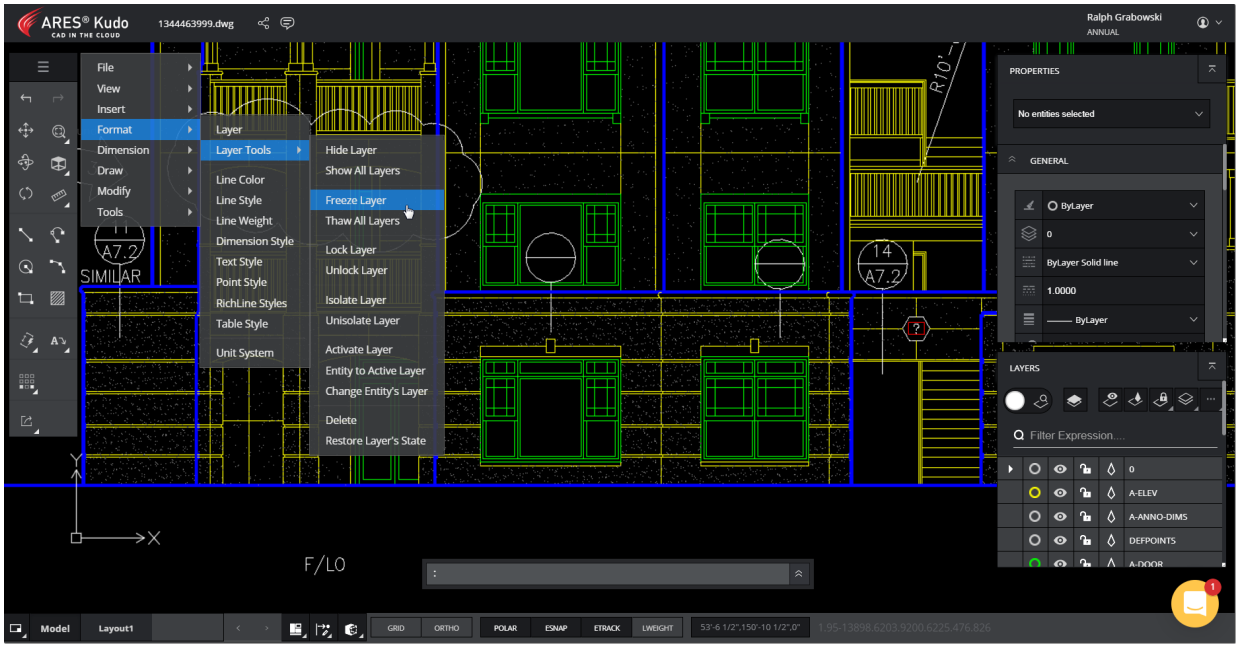
Title bar along the top of the Kudo window

UI Element	Purpose
ARES Kudo	Click to return to cloud storage explorer
File name	Reports the name of the current file
View-Only	Indicates the drawing cannot be edited
Share	Shares the drawing through an email'ed link or via cloud storage
Comments	**(NEW IN 2019)** Opens the Comments palette so that several people can leave notes about the drawing
User name	Displays menu to access storage settings, user profile, cloud storage explore, online help, and to log out

Drawing Area

Kudo's drawing area holds a toolbar and a slide-out menu on the left side. On the right are palettes, and along the bottom is the status bar and command prompt area. No ribbon, though, but there is a yellow dot for chatting with tech support.

Right-clicking entities displays shortcut menus. Overall, Kudo is closer to ARES Commander on the desktop than Touch on tablets.



Full version of Kudo with toolbar, sideout menus, palettes, status bar, and command prompt area (sample drawing sourced from Q-CAD)

(With the view-only version of Kudo, you view the drawing in the drawing area. You cannot select entities, nor does it offer right-click actions. You can, however, turn layers on and off, measure distances, and view model space and layouts.)

Kudo supports layouts and paper space, as shown below.

Layout being displayed in the drawing area of the Kudo window

On computers and devices with touch screens, you can use fingers to change the view of the drawing:

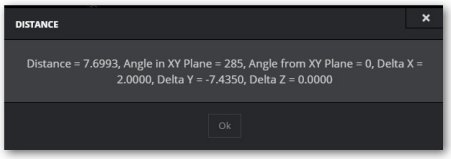
Action	Touch Action	Equivalent Mouse Action
Pan the drawing	Drag one finger	Drag middle button
Zoom in and out at the cursor point	Spread two fingers wider and narrower	Roll roller wheel
Zoom to extents	Double-tap	...

Toolbar

To interact with the drawing, Kudo offers a toolbar and palettes. The toolbar cannot be customized (at time of writing).

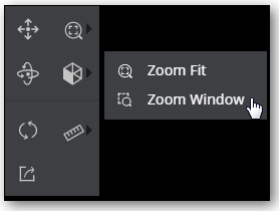
Left: *Toolbar at the left side of the Kudo window;* ***right:*** *...and shortened toolbar for view-only Kudo*

The two versions of the toolbar have the following commands in common:

Toolbar Element	Purpose
	**Pan**: moves the drawing
	**Zoom**: makes the drawing look larger and smaller; hold down the button to see the flyout: • Zoom Fit • Zoom Window
	**Orbit**: spins in the drawing in 3D
	**3D Views**: hold down the button to see the flyout: • Orthogonal top, bottom, left, right, front, and back views • Isometric SE (south east), SW, NE, and NW views
	**Rebuild**: refreshes the drawing
	**Inquiries**: returns information from the drawing; hold down the button to see a flyout: • X,y,z coordinates • Distance between two points 
	**Save as PDF**: saves the current drawing as a PDF file to your local computer

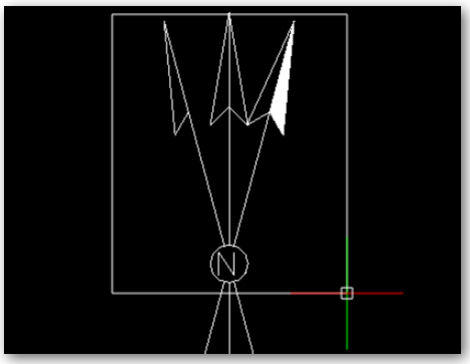
During the measurement commands, object snap mode kicks in automatically.

Flyouts. When you access a button from a flyout, the one you select becomes the topmost one, as in desktop CAD.



Accessing a button on a flyout

The figure above shows the **Zoom Fit** and **Zoom Window** options. To use the **Zoom Window** command, you click one point, move the cursor, and then pick a second point.

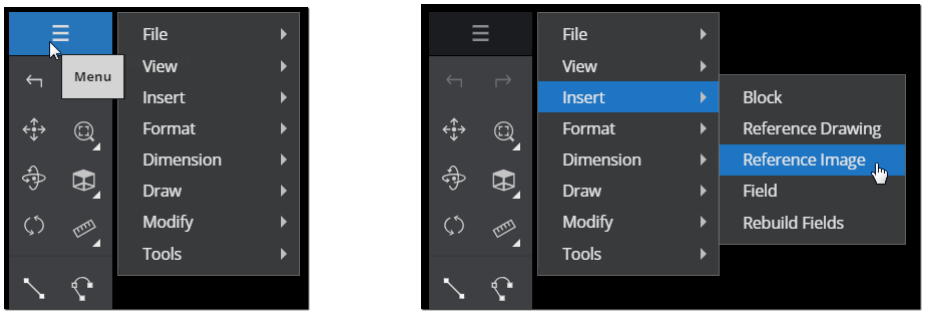


Clicking two points to define the zoom window

To return the drawing to its "home" view, click **Top** (from the 3D Views button), and then click **Zoom Fit**. The faster alternative is to double-tap the screen.

Slidout Menu

(Not available in view-only Kudo.) The hamburger icon ☰ atop the toolbar hides the menu. Clicking the icon slides out the menu, which in Kudo is arrayed vertically — as opposed to the horizontal layout in ARES Commander on the desktop. This change was made because the browser loads the top and bottom of the window with its own UI elements, squeezing the height available to the drawing.



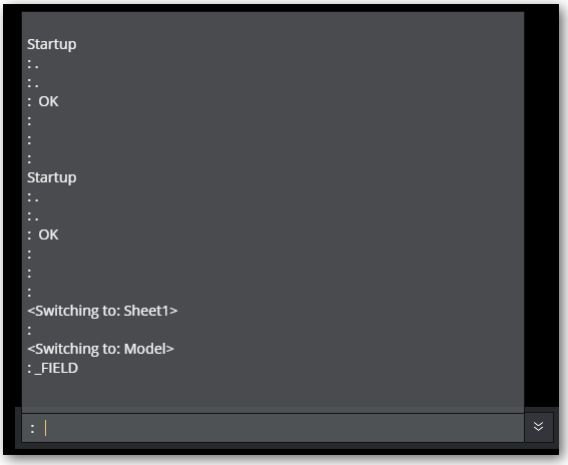
***Left**: clicking the hamburger menu; **right**: ...and selecting a menu item*

The Kudo version of the menu works just like on the desktop: select an item in the main menu, and a sub menu slides out. Click on a word to execute the command.

Some menu items execute commands; the others display a dialog box or a sub-sub-menu.

Command Bar

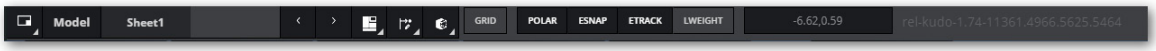
(Not available in view-only Kudo.) The command bar lets you enter ARES commands directly. Click the double-chevron ⌃⌃ icon to expand and contract the command bar.



Command bar fully extended

Status Bar

The status bar is along the bottom of the Kudo window. It is the same for the edit and view-only versions of Kudo. The content of this bar will change as Graebert add functions to Kudo over time.

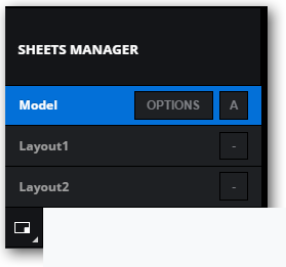


Status bar at the bottom of the Kudo window

From left to right, the controls on the status bar perform the following functions:

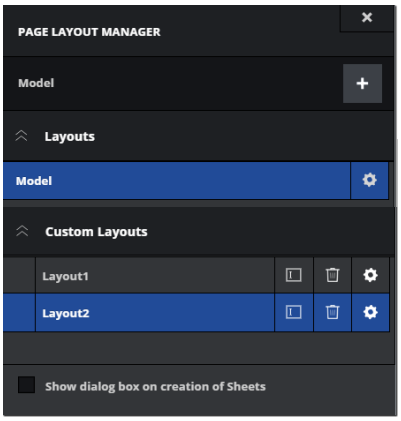
Status Bar Element	Purpose
	**Sheets Manager**: displays the sheet manager popup
Model, Sheet1	Switches the view between model space and layouts, if any.
< >	Reveals additional sheet names
	**Panels**: displays available palettes for properties, references, layers, and command window; view-only version offers only layer and then only layer visibility toggling
	**Esnap**: toggles individual entity snap modes
	**Shaded Views**: switches between shaded view modes
GRID	Turns the display of the grid (array of dots) on and off
POLAR	Turns polar assistance mode on and off
ESNAP	Turns entity snapping mode on and off
ETRACK	Turns entity tracking mode on and off
LWEIGHT	Turns the display of the lineweights (thicker lines) on and off
Coordinate display	Reports the x,y,z-coordinates of the cursor

Sheet Manager Popup. The Sheets Manager lets you switch the view between model space and layout sheets, if any. The view-only edition of Kudo cannot create or edit layouts.



Sheets manager accessing options for layouts

To access the Page Layout manager, click **Options**, and then **Settings**. (The letter "A" next to Options refers to the page size.)



Page layout manager in Kudo

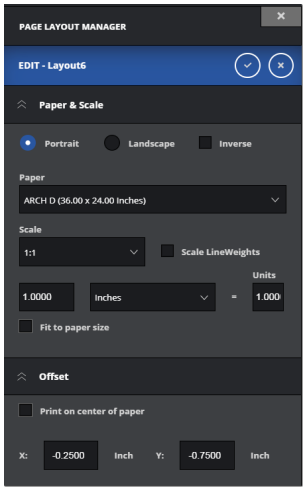
The Page Layout manager lists the names of layout sheets and Model space:

Here is the meaning of the controls in the Page Layout Manager palette. Notes that only

+ Add Layout	Adds a custom layout with the generic name "Layout2," etc
Settings	Displays the settings panel for layouts
Rename	Changes the name of the layout
Delete	Erases the layout from the drawing, with no warning
Show	Shows the Settings dialog box each time a new layout is created; see below

Layout Settings

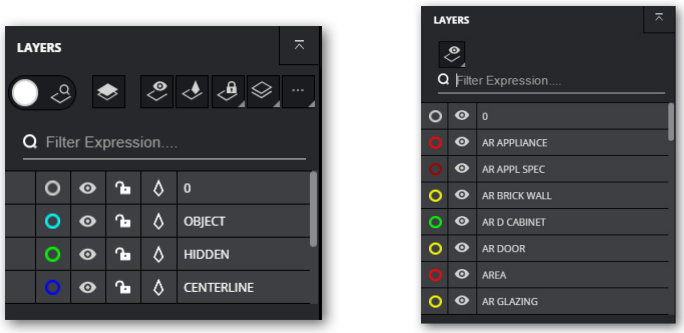
Click the **Settings** button for the page layout settings panel. In the view-only edition of Kudo, you only view settings, not change them. Click the **X** button to close the panel.



Page layout settings manager in Kudo

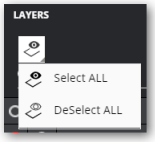
Layers Panel

The Layers panel lets you turn the visibility of layers on and off, freeze layers, and so on.



***Left:** Layers panel at the right side of the Kudo window; **right:** ...and visibility-only version in view-only Kudo*

The color dots indicate the color of the layer. (Layer colors cannot be changed in view-only Kudo.),. Use the **Filter Expression** field to search for layer names. For example, enter 'ar' to find all layers that begin with AR. Use the **Select/Deselect ALL** buttons to turn on all layers and off.



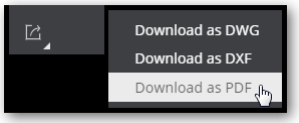
Turning on (or off) all layers

To make panels smaller, click the title bar; click a second time to expand them again.

Rolling up the Layers panel

Printing Drawings

Although Kudo supports page layouts, it cannot print them, because Web browsers have very weak support for printers. (The support is weak, because Web pages were never meant to be highly formatted documents, the way they are today.)

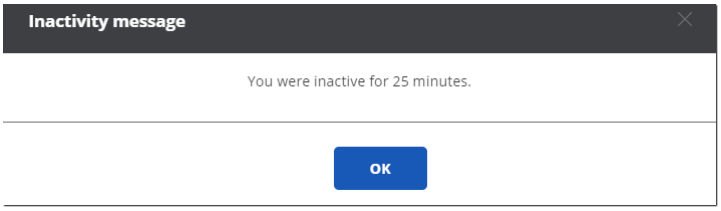


*Accessing the **Download as PDF** button on the Kudo toolbar*

The workaround is to download drawings as PDF files, and then print the PDFs. One of the strengths of the Portable Document Format is that it stores highly-accurate representations of pages, both text and graphics, including CAD drawings.

Inactivity Message

When you don't use Kudo for 25 minutes, the following Inactivity message appears in your Web browser:



You have been unproductive for nearly half an hour!

Click **OK** to continue working.

Here's the reason for the dialog box appearing in your Web browser: when you are running Kudo, it is communicating with a server computer located elsewhere. The communications link ties the server's resources to your Web browser. If you are not doing anything in Kudo, the resources on the server are still tied up.

When Kudo notices that you haven't done anything for 25 minutes, it severs its connection with the server, freeing up the resources on the server for other users. Running the server is not free, and so it must be shared with as many users as possible.

Clicking OK resumes the connection. The resumption is so smooth that you probably don't notice any delay or change in the screen.

How Kudo Works

The Kudo program does not run on your computer or tablet. Instead, it runs on *servers*, which are business-class computers designed to serve data quickly over the Internet. Room-fulls of server computers are run usually by companies like Amazon, Google, and Microsoft; software companies, such as, Graebert rent time on them.

Typical room full of server computers (image source Wood and Design)

WEB BROWSER TECHNOLOGY

There are two ways to run CAD in a Web browser, the old way and the new way.

The Old Way

The initial attempts to run a full CAD program on a server, which has graphics boards (usually from nVidia) that push raster images of what is happening on program's screen to your Web browser. Each time you make a change to the model or click on a button, the distant server re-sends the entire screen, at a rate of about 30 screenfulls per second.

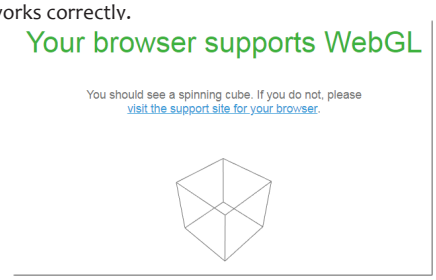
The advantage to this approach is that what you see in your Web browser is *identical* to the CAD program you run on your desktop. Using this method allows any program to operate "on the cloud." One company that provides this service is https://fra.me.

The drawback, however, is that this method requires a lot of server power to push the pixels to your screen, and so the number of people who can use the CAD program simultaneously is limited. In addition, I've found that opening and saving local files is awkward, as the system must set up a virtual drive on your computer.

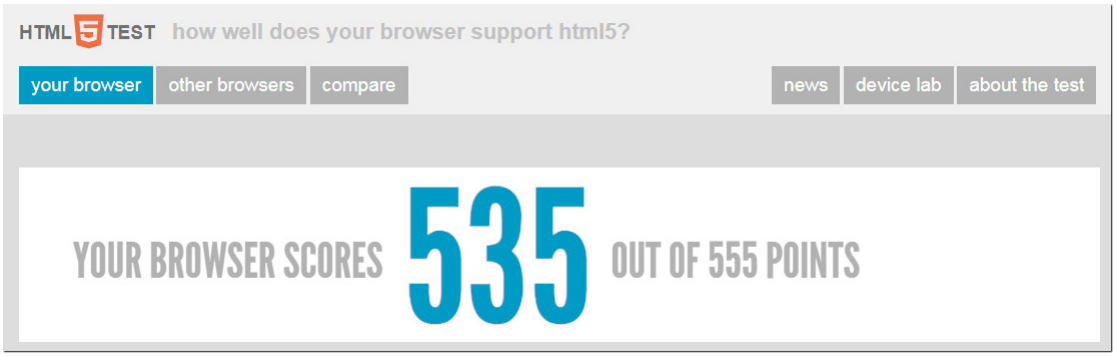
The New Way

Kudo uses a newer method that is more efficient, known as browser-based CAD. Here the Web browser does more of the work, which greatly reduces the amount of data that needs to come from the server. *Java* programming and *HTML5* technology running in Web browsers allows you to interact with the CAD drawing, picking points and clicking buttons — just as if it were running on Windows.

TIP Visit https://get.webgl.org to see if your Web browser supports WebGL; in some browsers, WebGL must be turned on manually before it works correctly.



Visit https://html5test.com to see how well your browser supports HTML5 (see figure below).



Testing the latest version of the Vivaldi Web browser for HTML5 compatibility
(In case you are wondering, Google Chrome rates 532)

WebGL technology allows Web browsers to display, zoom, and rotate 3D models fully rendered. (OpenGL is used on desktop computers to do the same thing.)

When you first start Kudo by entering kudo.graebert.com in your Web browser's address bar, the remote server uploads some of the programming code to the browser, such as the code needed to display the user interface.

When you open a drawing, some of the drawing is displayed in your Web browser. Just enough of the drawing is displayed so that you can see it "normally." This reduces the time it takes to display the drawing, reducing the lag time you might experience. When you zoom in or rotate the drawing, then the "hidden" parts are sent by the server to the browser.

When you make changes to the drawing — by adding a circle, moving a line, erasing a block — only the information regarding the change is sent by the browser to the server. This minimizes the amount of data that is sent back and forth.

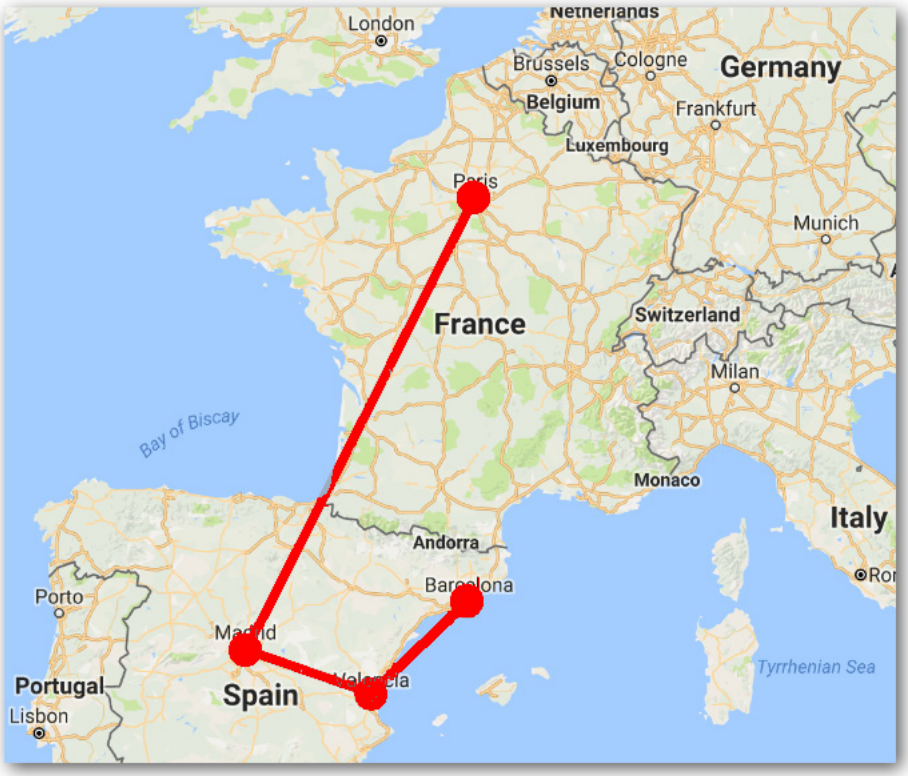
There is no "saving" of drawings, as each change is recorded by the server, and the *.dwg* file stored on the server is always up to date with what you see in the browser. An unlimited undo function lets you walk back to earlier versions of the drawing.

TIP Bandwidth is the speed of your Internet connection. A higher bandwidth allows more data to arrive at your Web browser in less time, but it does not solve the latency problem. You want high bandwidth and low latency. You can usually pay for higher bandwidth from a service provider, but you cannot control the latency.

Solving the Latency Problem

The primary disadvantage of server-based software is *latency*. This is a delay that occurs between the time the server sends the data and the time your browser receives it. Is measured in tens or hundreds of milliseconds. The delay is caused not so much by the transmission speed as by the route the data takes. Data is routed along the Internet via *switches*, with each switch determining the best subsequent route (called a "hop") for the data.

The figure below shows a *tracemap* that traces the hops and locates switches. Someone in Paris has accessed a Web site hosted in Barcelona. The shortest distance is not necessarily the fastest one. Indeed, the original design of the Internet (when it was called "Arpanet") was for a communications system that automatically bypasses blockages, such as in the event of a nuclear war.



Route taken by data via the Internet

To minimize latency, companies locate servers in many places around the world, and so Kudo should be using the server located nearest you. Amazon, for instance, has servers in Virginia, Ohio, Oregon, California, Canada, Brazil, Ireland, Frankfurt, London, Israel , South Africa, Dubai, Tokyo, Seoul, Singapore, Mumbai, Sydney, and Beijing. These locations were picked because they are closest to large population centers (northern Virginia, for example) or near cheap sources of electricity (Oregon, for instance).

Servers are often called "the cloud," because we don't necessarily know where the data is being stored; we just know it's somewhere "up there." Regulations in certain countries require that data be stored in their country, which can dictate the location where your Kudo data is stored.

For a technical discussion of how Kudo works, check out this blog which features a video interview with Graebert chief technical officer Robert Graebert: https://www.graebert.com/blog/featured/graebert-scales-a-stateful-cloud-based-cad-service-to-match-demand/

Benefits of Cloud CAD

Using a server-based CAD system like Kudo provides several benefits not available to desktop CAD.

DRAWINGS AVAILABLE EVERYWHERE

When you use Kudo, it does not store drawings on your computer; the *.dwg* files instead are stored on a remote server. Kudo lets you choose to store your drawings on online file storage services from Box, Dropbox, Google Drive, and iCloud.

The benefit is that you can access your drawings on *any* computer or tablet that runs a suitable Web browser and has a connection to the Internet. You may have already experienced this when you save a file to Dropbox and then access the file later on another computer. Kudo lets you search for your drawings on all of these online storage services.

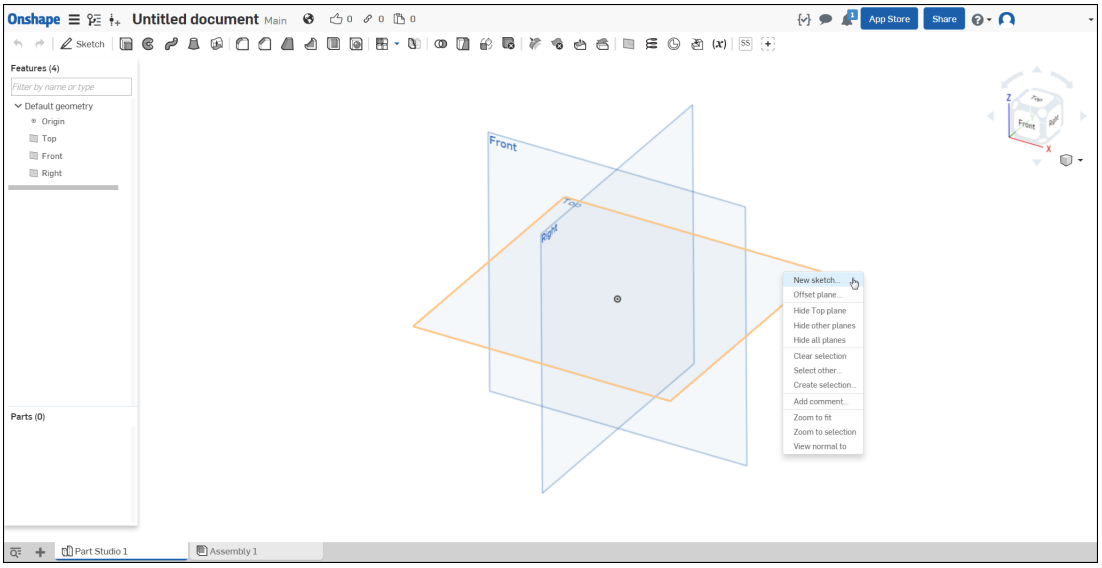
Drawings Available to Everyone

With *.dwg* files stored on servers, they can be made accessible to everyone. The files are not limited to being accessed by just you. In Kudo, you give other people permission to view or edit your drawings. This is useful when more than one person works on a project. For instance, you are designing traffic signals for an intersection, and your bosses want to show your progress to the client. You give them access to the drawings and they launch view-only (unlicensed) versions of Kudo.

As a side effect, everyone is viewing (or working on) the *latest* version of the drawing(s). Because they are stored on a server, there is only one copy — the original.

Specific to Graebert, the drawings stored on servers are available to all of its CAD platforms: ARES Commander for Windows, MacOS, or Linux on desktop; ARES Touch for Android and iOS on tablets and smartphones; and ARES Kudo for Web browsers. Recall from the previous chapter that a single license gives you access to all of these products.

Connection with Onshape. It is significant that the original version of "Kudo" appeared first in Onshape, an online MCAD system that also runs in Web browsers from remote servers. Graebert provided the drawing component, which generates 2D layouts from 3D models and then touches them up with annotations. Getting in with Onshape was a significant win, because it gave Graebert programmers working experience in optimizing Kudo with the Internet.



The opening screen of Onshape, a 3D solid modeler

Frequent Updates

Everyone running Kudo uses the same version; there is no version-lag, where some users employ an older and possibly incompatible version of CAD software. Bug fixes and feature updates are frequent, typically every three to four weeks.

Drawbacks to Cloud

Running CAD on remote servers has many advantages but not everyone uses it, because some advantages turn out to be disadvantages. Here are some of the benefits to continuing to run CAD on desktop computers:

Desktop computers keep working, even when there is no Internet connection. Kudo, by contrast, works only when there is an Internet connection. If the Internet is unavailable or temporarily broken down, then the software cannot launch, no drawings can be opened or edited. None of the sharing and collaboration features work.

The reaction speed of desktop computers is consistently fast, especially when outfitted with solid state drives, the maximum RAM, and an optimized graphics board. When the connection is slow or when you are located far enough away from a server, then Kudo's reaction speed lags.

The hardware on which the Web browser runs needs to be fast enough to operate the CAD software, even when the browser is an approved one. Older tablets and very old computers don't have the horsepower to meet the demands of WebGL and HTML5.

Desktop computers store drawings locally, such as on their hard drives or an in-office file server, and optionally at remote locations, such as off-site backup and the cloud. Some CAD firms are concerned that online storage services are insufficiently secure, and so prefer to keep files on-site or on a private cloud.

With desktop computers, you can buy a permanent license once, and then run the software for all time. Software that runs in Web browsers and on tablets tend to require annual payments, for a variety of reasons. When software runs in browsers, it runs continuously on servers, and both the software and hardware need 24/7 support to reduce downtimes; this has to be paid for. On tablets, the issue is that Apple drove down the prices we expect to pay — like $1 or free — and so CAD developers make too little income that way to write the advanced code that's needed; simple CAD programs are free, but advanced ones require annual payments to the developers.

Some corporations require stable software, software that is unchanged for the duration of the project, such as designing and maintaining aircraft or nuclear power plants. Running CAD on the desktop allows users to "freeze" the software to a known state, so that nothing changes. Cloud-based software changes at least once a month, and sometimes the changes create problems, such as introducing new bugs, or changing the way a command is carried out. It can be difficult to keep track of the changes occurring monthly.

The desktop version will almost always have the most functions, although there are cases where the browser and tablet environments have their advantage, function-wise — such as sharing and access with the browser, and touch interactions and integrated camera and GPS functions on tablets. The browser and tablet versions of CAD software have fewer functions, because not all aspects of desktop CAD translate well to the alternative platforms.

SO, WHICH IS BETTER?

After reading about the pros and cons of using cloud or desktop software, which should you choose? **Neither**, is the answer that might surprise you.

The best software to use is the one that is best for your situation:

- **ARES Commander for desktop and laptop computers** — when you want access to the most functions, keep drawings stored locally, pay a license fee once, and freeze the version of the software
- **ARES Touch for tablets and smartphones** — when you need a portable CAD system that also accesses the hardware integral to the devices
- **ARES Kudo for Web browsers** — when you need to access and share CAD drawings on just about any device

The single annual license from Graebert lets you pick and choose which version of ARES to run in each situation.

Notes

CHAPTER 13

ARES on the Cloud

In this Chapter

- Sharing drawings through Kudo file storage
- Community-wide commenting on drawings

Kudo's file manager is the center for coordinating drawings among in-office and out-of-office computers, even ones not running ARES.

It is useful to think of Kudo in two parts:

» **A drawing engine** that views, draws, and edits DWG files
» **The central file manager** that coordinates sharing of drawings between CAD systems and file storage services

In fact, it might have been better if the two parts had different names, so as to emphasize their distinctive roles.

In the previous chapter, we looked at Kudo, the drawing engine. In this chapter, we see how Kudo operates as a central file manager — a role that in our industry sometimes goes by the name of EDMS, short for "engineering data management system."

For background on using links for sharing files, you may want to watch the video at https://www.graebert.com/blog/tutorial/share-cad-drawings-in-dwg-as-an-online-link-for-free-viewing/.

Managing Drawings on Projects

People working on projects need access to drawings. One of the primary benefits to Kudo is that it shares project files easily — with ARES Commander in the office, with ARES Touch in the field, with other CAD systems, and even with anyone not using any CAD software. The files can be stored on a local drive, on the cloud, or on a private server internal to your company.

In summary, you and your project team access files, as follows:

Location	Hardware	Applicable Software
In the office	Desktops and laptops	ARES Commander through the Cloud Storage palette
	Tablets and phones	ARES Touch
	Web browsers	ARES Kudo
On the go	Tablets and phones	ARES Touch
Anywhere, anyone	Web browsers	ARES Kudo

The file management system in Kudo is available to everyone for free. It is not downloaded or installed, because it operates in a Web page.

Kudo does not need to be managed by your office IT staff, except initially. Kudo requires an account with Graebert, and you (or your firm) will have set up a corporate account. As well, some users may need assistance in linking Kudo to cloud storage service(s); see the previous chapter for the details on setting up Graebert accounts and cloud services in Kudo.

Once you have Kudo set up, you can begin making project files available to everyone who needs access. The most important aspect to sharing is setting *permissions*.

SETTING PERMISSIONS

Not everyone who has access to project files should be able to change the content of the files. Sometimes you want other people to edit your drawings, and sometimes you don't. This is akin to inviting friends over to visit (view only) or giving family members the key to the front door (edit).

View-only. When you want to limit people to only viewing drawings and other files, then this is called having "view-only" or "read-only" permission. The recipient can zoom, pan, and change layer settings — but that's all. None of these actions harm the content.

Edit. You can, however, allow others on the team to change project files and drawings. This is called "edit" permission. The owner, which might be you, always retains all rights to the files, including viewing and editing them.

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- **Comments** command opens the Comments palette so that you and other users comment on drawings, using the desktop, tablet, and browser versions of ARES.
- **CommentsHide** command closes the palette.
- Generate a view-only links, view file previews, and do right-click options (added to ARES Commander 2018 service pack 1):

SHARING DRAWINGS VIA KUDO FILE MANAGER

To share drawings and other files among all project members, you take these steps:

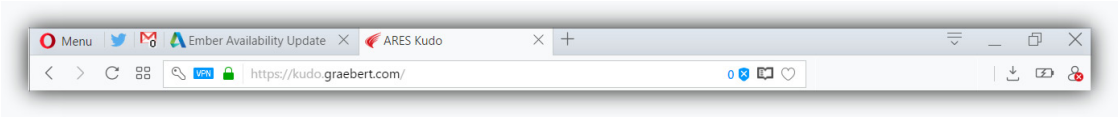
i. Launch Kudo in your Web browser (after registering your account with Graebert, if necessary)

ii. Add cloud storage services to Kudo (and add project files to them, if necessary)

iii. Assign permissions to project folders and files (which notifies users by email)

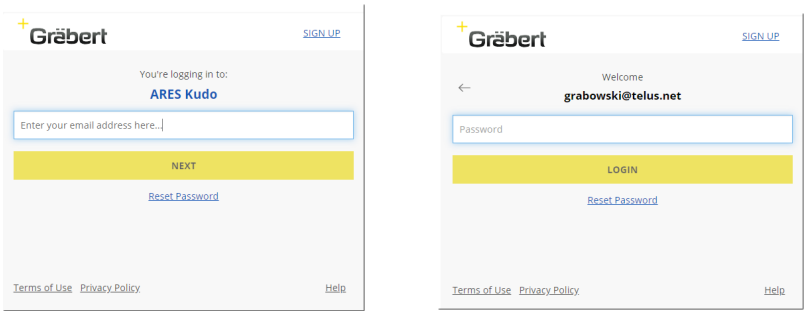
Step 1. Launch Kudo in Browser

Open your Web browser, and then enter **kudo.graebert.com** in the address bar.



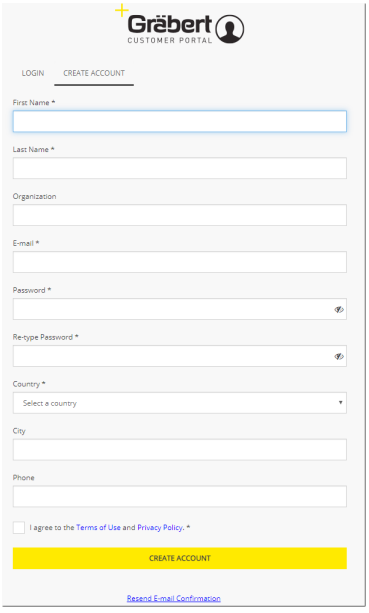
The 's' in https indicating that the Web connection is secure by encrypting communications between your computer and the distant server running Kudo.

Enter your login details — email address and password.



***Left:** Entering your email address; **right:** ...and your password*

If you do not have an account with Graebert, then click Sign Up, and then fill out the information requested. As a minimum, enter something into the fields marked with the asterisk (*).

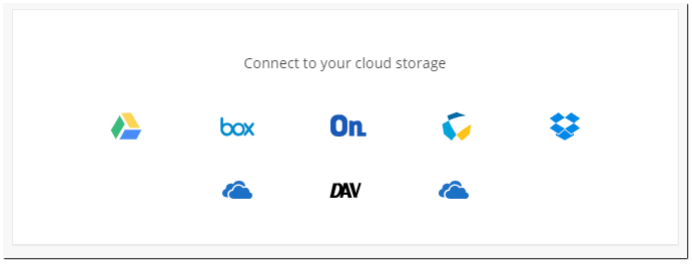


Signing up for a new Graebert account

Graebert needs you to have the account so that it can figure out who you are when you access Kudo from a variety of browsers.

Step 2. Add Cloud Services to Kudo

Once inside Kudo, click the Ralph Grabowski ANNUAL button, and then choose **Storage**. Kudo displays the cloud storage services that it supports. (This list may change should Graebert add other services.)

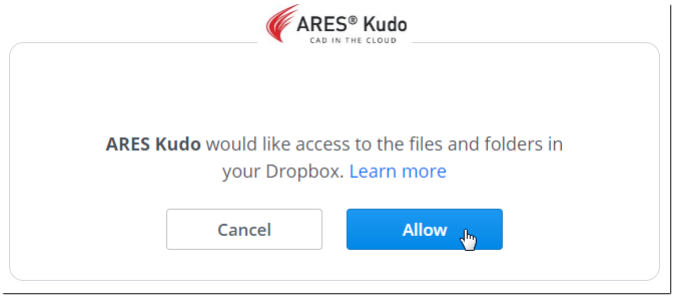


Cloud storage services supported by Kudo (at time of writing)

Click the **Add New Account** button next to the name of a cloud service. You can have more than one account per service (under different logins), and you can attach more than one service to Kudo, such as Dropbox and OneDrive.

Drive service with Add button at right

(If necessary, first sign up with the cloud service.) Log in to the cloud service with your user name and password. Kudo asks you for permission so that it can access your files in the cloud service. Click **Allow**. (If you click Cancel, then you cannot access your files through Kudo.)



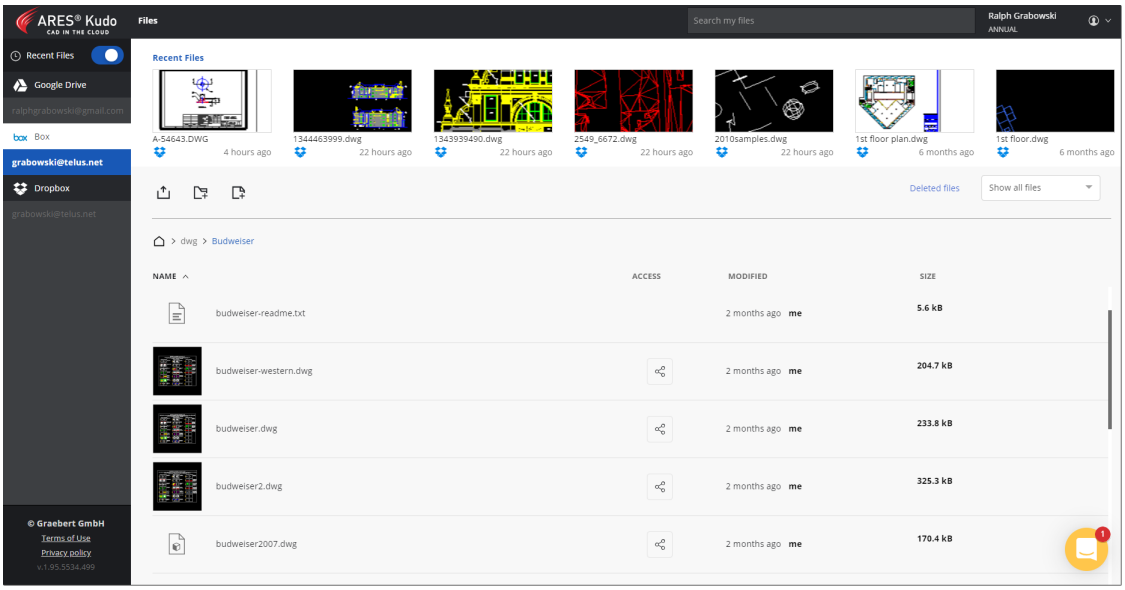
Giving Kudo access to your online files

In a few seconds, Kudo reports it has successfully made the connection, and then returns you to its file manager.

Successful connection

Repeat the process to connect to other cloud services.

The names of connected file services appear on the lefthand side in Kudo's file manager, while the folders and files from the service appear in the middle of the file manager.



Cloud services on the left, files and folders in the middle

A thumbnail-size previews image appears next to each file; when the folder has never been opened in Kudo, it takes a minute for the preview images to generated by the server, and then displayed in your Web browser.

You can view the files and folder from only one cloud service at a time. However, Kudo's search function searches across all connected services.

Step 3. Assign Permissions to Project Files

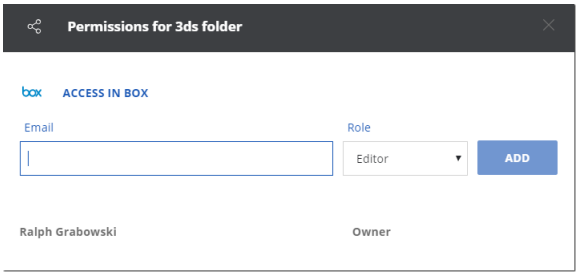
For every member of a team who needs access to project files, you can set the level of permission for individual files or entire folders. (When I say "member of a team," I mean positions like clients, sub-contractors, and bosses.)

Clearly, it is faster to set the permission level of an entire folder than for each file individually. Only one person can be the *owner* of a file or folder. If you are not the owner, then you cannot change the permission level.

As you set the permission levels, you also send team members email messages at the same time. The messages alert the recipients that they now have access to the project file(s). The email contains a unique link to the file(s).

To send a link by email, follow these steps:

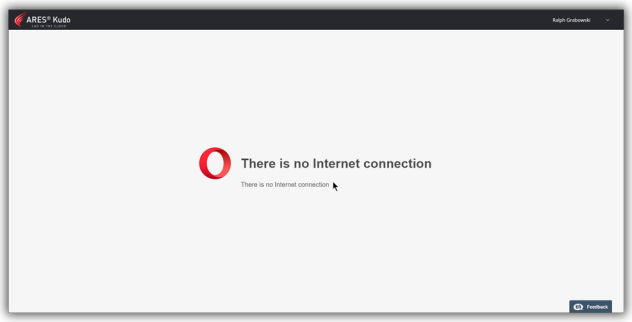
1. In the Kudo file manager, click the **Access** button next to a folder or file name. Notice the Permissions dialog box.



Permissions dialog box

OFFLINE DRAWING ACCESS

The #1 issue in using browser-based CAD is this: what do you do when there's no Internet connection? Cloud programs like Kudo depend on a distant server for their operations, and this requires a connection through the Internet. No Internet, no Kudo. This is no different from any other Web page.

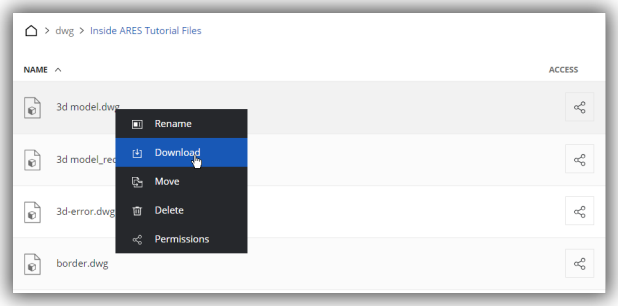


When there is no Internet connection, Kudo does not work

PREPARE TO BE OFFLINE

When you know that your computer or tablet will be offline, such as at a remote site or on an airplane, download drawings to your computer ahead of time. Follow these steps in Kudo:

1. In cloud explorer, right-click the drawing you want to download from the cloud, and choose **Download**.



Selecting a file to download from the cloud to the local computer

2. Notice the Save As dialog box. Choose a folder in which to save the file, and then click **Save**.
3. As Kudo downloads only one file at a time, repeat the process for the other drawings you need. Kudo does not download entire folders or groups of files.
4. Kudo cannot view or edit downloaded files until the computer is again online, so you need to use a different ARES program while off-line:

» When working on a laptop, use ARES Commander

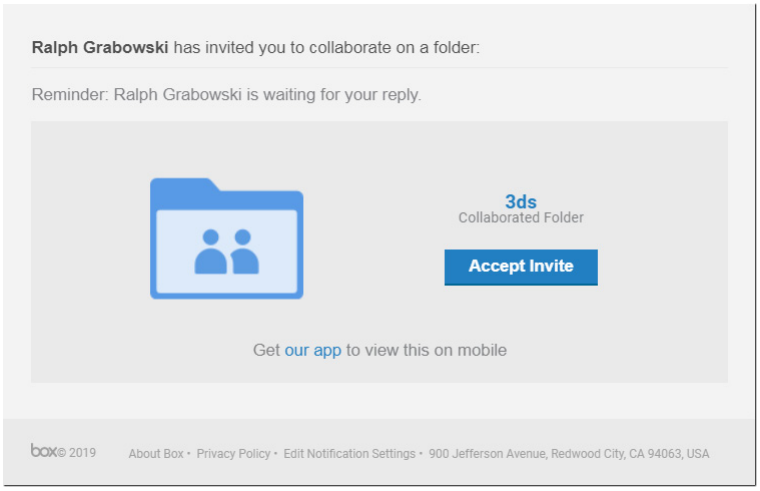
» When working with a tablet, use ARES Touch

All three program — Kudo, Commander, and Touch — are provided to you under one license, and so there is no additional cost for you.

BACK ONLINE

When the connection to the Internet is restored, click the Web browser's **Refresh** button or press function key **F5** to restart Kudo.

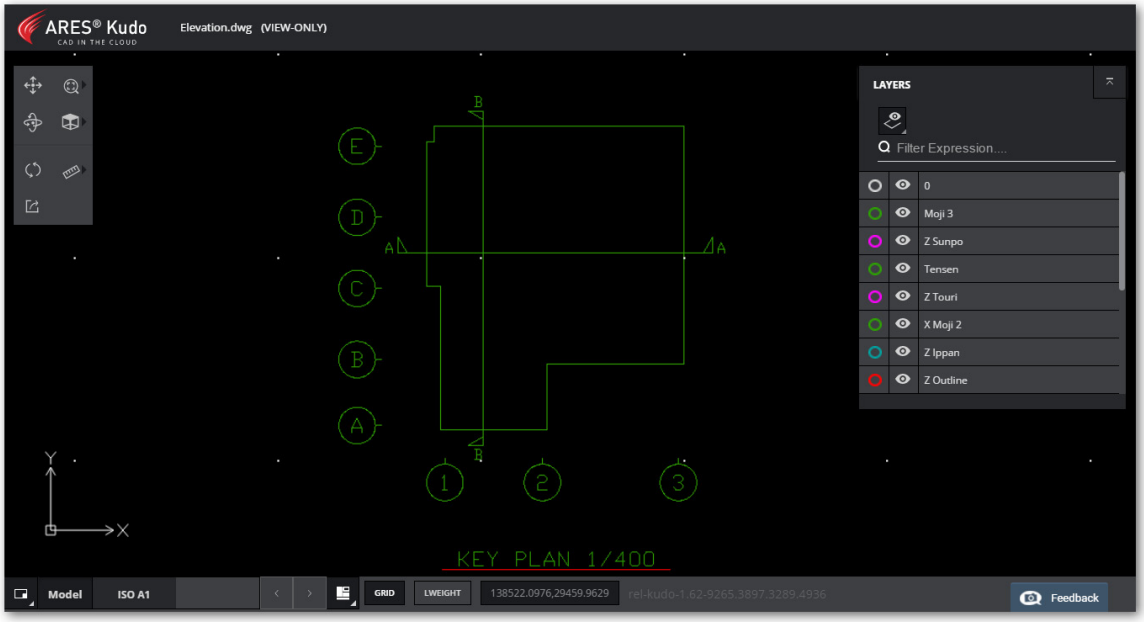
2. Fill in the information requested:
   a. Enter the **Email** address of a team member. You cannot, unfortunately, enter multiple email addresses at once. Should you make a syntax error in the address, the dialog box warns you, "Your email didn't pass validation."
   b. Select a **Role**, either Editor (can edit the file) or Viewer (can only view the file)
   c. Click **Add**. If the email address has already been given a sharing role, a dialog box alerts you, "Couldn't share folder: User already is a collaborator."
3. The emailed invitations received by recipients look like this:



Email'ed invite

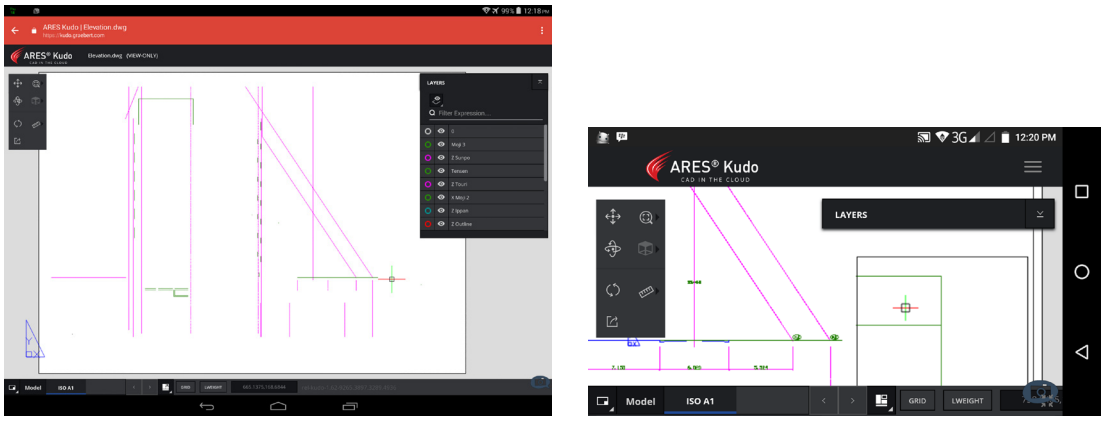
4. When recipients click the link, their Web browser opens Kudo in a new browser tab in view-only mode, and then loads the drawing. Kudo looks exactly the same as in your copy, except that there is no user information (missing from top-right in the figure below), and the recipient has no access to the file manager.

The link that Graebert generates is unique to the drawing and to the recipient. Here is the shared drawing in a Web browser running on a desktop computer:



Viewing the received drawing in Kudo

The link can be opened in tablets and even smartphones, as illustrated below:

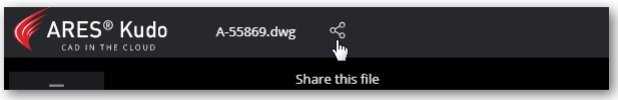


Left: *Drawing opened on an Android tablet;* ***right****: ...and opened on an Android smartphone*

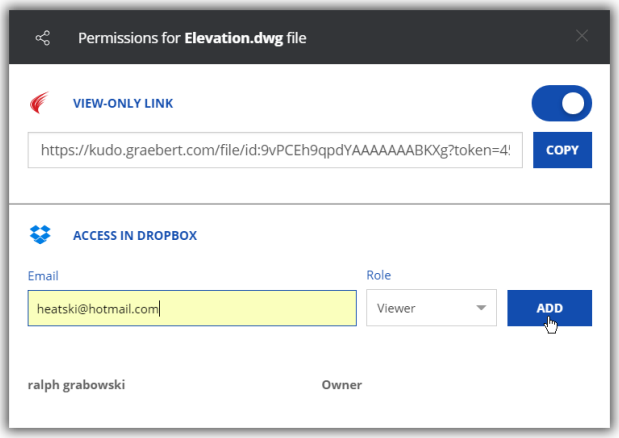
SHARING DRAWINGS FROM INSIDE KUDO

To share the current drawing from inside Kudo, follow these steps, which are a bit different from employing sharing in the file manager:

1. On the Kudo title bar, click the **Share this File** button.



2. Notice the Permissions dialog box.



Fill out either the upper *or* the lower half of the dialog box, because they support two different ways to share the drawing:

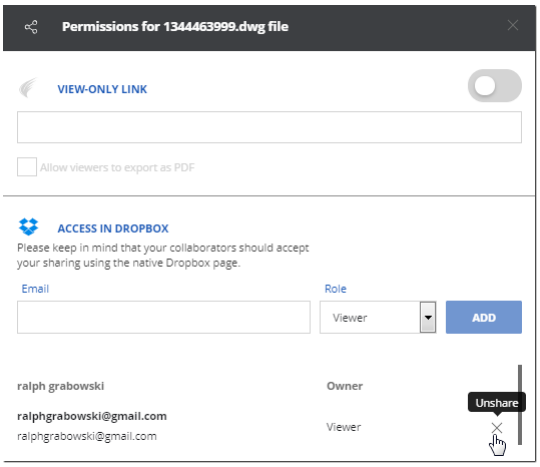
» **View-only Link** (upper half) generates a link that you send to recipients by email or text message; they click the link to opens the drawing in their Web browser and immediately run Kudo; see "Sharing via Email Link" below.

» **Access in** (lower half) also generates a link, but to a cloud storage service such as Dropbox, from which recipients access the file; see "Sharing via Cloud Storage" elsewhere in this chapter.

Halting Access to Files

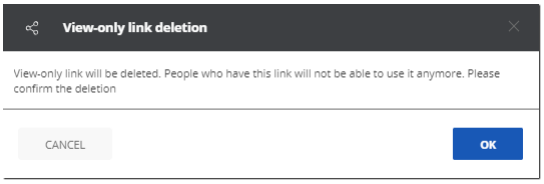
At some point, staff leave, projects end, and so you need to block access to shared files. To do so, follow these steps:

1. Click the **Access** button next to a folder or file name. Notice that the bottom of the Permissions dialog box lists the names of people who have access.



Removing share status

2. Notice the **Unshare** button next to the recipient's name. Click the **X**.



Removing the link

3. Click **OK** to remove the link. Removing the link removes the recipient's ability to view the file.

SYNCING KUDO WITH ARES COMMANDER

When you want to use the same set of drawing in more than one location, then you follow this workflow:

i. Use ARES Commander on desktop computers to do most of the drawing creation and editing work.

ii. When you are out in the field, use Touch on tablets to view and markup drawings.

iii. When you want to share drawings with non-ARES users, send them links from ARES Commander or Touch to view them with Kudo on their computer's Web browser.

One Truth. When drawings have the same name, we can get mixed up as to which one is the most recent version. With many drawings and revisions of drawings flying around projects, it is crucial that there be "one truth," as one industry phrase calls it. *One truth* refers to the drawing that is most up to date.

Three drawings that look similar but are different versions. Which one is most-current?

One way to enforce this is to ensure that only view-only copies are derived from the original. This role is well-played by PDF files. The original is the only one that is allowed to be edited.

Date and Time. As PDF files cannot be edited, a different solution is needed when two different people need to edit the same drawing, or one person need to edit it on different platforms — say with ARES Commander on a desktop computer and then on a laptop.

It is possible to use the file's date-and-time stamp as a reference, but this can become a tedious way to compare files. Worse, the date-and-time stamp can be wrong, such as when the clocks on computers are in different time zones, or if the battery that saves the date and time has died. In some cases, operating system stamps copied files with the current date and time, overwriting the Date Last Saved information.

Drawing Comparison. One way to control the problem is to use *drawing comparison*, in which two versions of the same drawing are compared to locate differences — additions, removals, and entity moves. The job of comparing more than two or three drawings is also tedious; it's better to just not have the two versions in the first place.

So the alternative is to maintain a central location for storing drawings.

Central File Storage. Centralized file storage is not new; in fact, it was the default method of storing all files in the earliest days of computing, 1970s to 1980s. If you used a computer at a company or school back then, it was at a terminal attached to a single, shared central computer (mainframe or minicomputer) with a single, shared storage system. Then came along personal computers, which arranged things exactly in the opposite manner: users worked at their own computers, each with its own storage as if they were on an isolated island; networking between personal computers was rare, expensive, and slow. Today, we

live with a mix of the two extremes: personal computers that still store files locally yet are connected to each other through fast ethernet and pervasive Internet.

Once a design project involves more than one person and more than one computer, we need a way to keep track of drawings as they are created, and then edited, approved, and updated. One solution is to use a service like Dropbox as the central location for storing all files — together with Kudo to monitor them. As files are edited and saved (by more than one drafter), Kudo compares variations of the same drawing and determines which one is the most recent one, whether edited by Touch or ARES Commander.

CAD software from Graebert accesses files from the following locations:

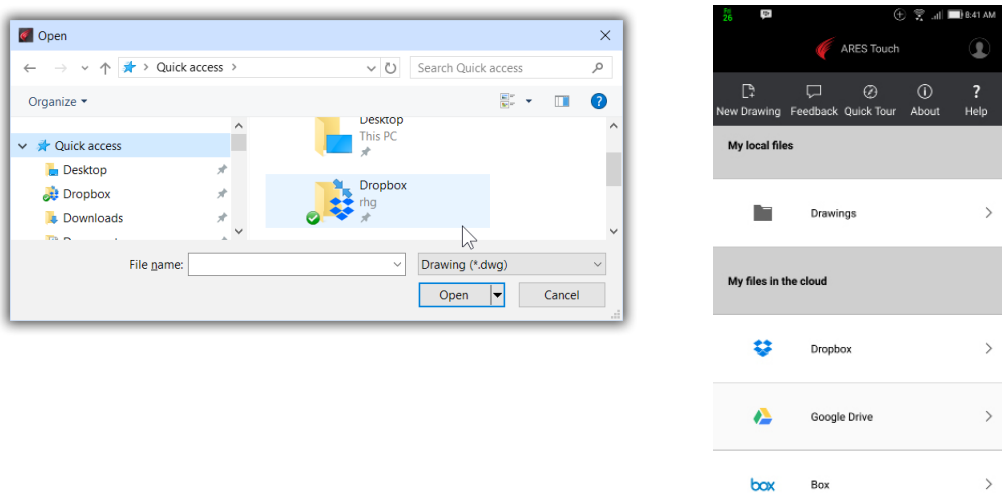
CAD Program	Local Storage	Network	Cloud (Dropbox, etc)
ARES Commander (Linux, MacOS, Windows)	Read/write	Read/write	Read/write
ARES Touch (Android, iOS)	Read/write	...	Read/write
ARES Kudo (Web browser)	...	...	Read/write
ARES Kudo view-only	...	...	Read-only

There is nothing special about working with the cloud on a desktop computer or a tablet. Once a cloud service, like Dropbox, is installed on your computer, you open and save drawing files from and to the cloud with the CAD program's regular **Open** and **Save** commands — you navigate to the cloud's folder on your computer's drive. To open a file from the cloud, follow these steps:

1. In ARES Commander, enter the **Open** command.

   (In Touch, launch the app.)

2. In ARES Commander's File Open dialog box, navigate to the Dropbox folder or other cloud service. See figure below, at left.



***Left**: Accessing the Dropbox folder in ARES Commander; **right**: ...and in Touch*

   (In Touch, tap the name of the cloud service, as shown above at right.)

3. Choose the *.dwg* file to open. As you work on the drawing, the ARES automatically saves and backs up the file on Dropbox.

4. When you close the drawing or use the Save command, ARES saves the *.dwg* on Dropbox, and closes the backup and lock files.

When everyone on a design team accesses drawing files only through Dropbox or another centralized file server, then two problems are solved:

- Only one person can work on a drawing at a time, because the file is locked for the duration. When someone else opens the drawing, ARES reports that it is in use and opens it in read-only mode, so that it cannot be saved.
- The most-recent save is the most recent version of the drawing.

It is possible, however, to circumvent Dropbox by copying drawings from Dropbox to the computer's internal hard drive, and then edit them independently of the other designers. And so these workarounds do not completely solve the problem of who last edited the drawing and which one is most up to date.

A mediator is needed!

Coordinating Files through Kudo

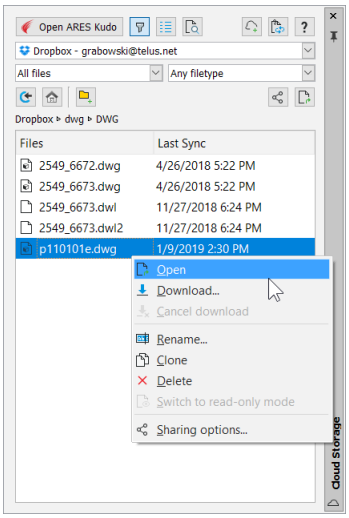
Graebert has designed Kudo to be a drawing mediator. It regulates drawing access, no matter where the drawing is edited — online, offline, on a tablet, or on a desktop computer. The drawback is that file coordination requires planning and discipline. The *.dwg* files you and your design team works with must be in a single folder.

The technology is built into Kudo and Touch as they were developed during the sharing era of the 2010s. (Graebert is still developing it, so you can expect more capabilities to come.) ARES Commander was developed earlier and so the capability was added through these commands:

- **CloudStorage** command displays the new Cloud Storage palette, from which you open drawings stored in connected cloud services, such as Dropbox. Because you connected Kudo with cloud storage service(s), you do not need to make the connection with ARES Commander.
- **HideCloudStorage** command hides the Cloud Storage palette.
- **SaveToCloudStorage** command uploads local drawings to a connected cloud storage service, if the files are not already there. Once there, you use the regular **Save** command and the files are synchronized with the cloud storage.

To use the palette, follow these steps:

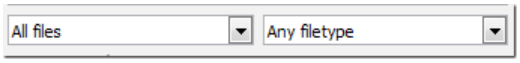
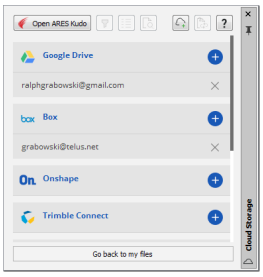
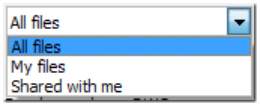
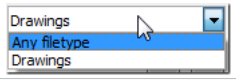
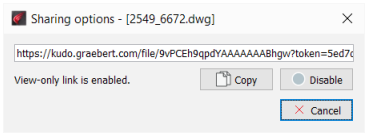
1. Open the palette with the **CloudStorage** command.



Cloud Storage palette

2. Click on the Dropbox - grabowski@telus.net droplist, and then choose a cloud storage account. Because you connected Kudo with cloud storage service(s) earlier, you do not need to make any connections here inside ARES Commander.
3. Navigate through the folders to find the file you want to open.
4. Click the **Open** button to view the drawing in ARES Commander.

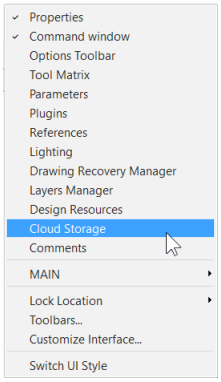
The Cloud Storage palette offers you the following controls:

Icon	Palette Control	Purpose
Open ARES Kudo	Open ARES Kudo	Opens Kudo in your computer's default Web browser at the cloud storage explorer page
(NEW IN 2019)	Filter	Toggles the file filter droplists: 
(NEW IN 2019)	Additional Info	Toggles the information panel
(NEW IN 2019)	Preview Image	Toggles the preview panel
(NEW IN 2019)	Manage Storages	Displays the available cloud storage services  To return to the list of files, click **Go Back to My Files**
	Refresh	Refreshes the list of files
(NEW IN 2019)	Help	Display help for this palette
Dropbox - grabowski@telus.net	Cloud Storage Control	Selects a cloud storage account from which to access files
All files	Filter by Ownership	Filters the list of files by who owns them: 
Any filetype	Files by Type	Filters the list of files by their type: 
	Back	Go up one level in the folder structure
	Home	Go to the home folder of the cloud service
(NEW IN 2019)	Create Folder	Makes a new folder in the current folder
(NEW IN 2019)	Sharing Options	Displays the Sharing Options dialog box: 
	Open	Opens the selected drawing in ARES Commander
Dropbox ▹ dwg ▹ DWG	Path	Displays the path of folders to the current one

Working with Two Drawings at Once

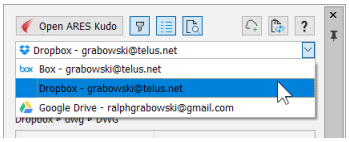
For instance, here is how the system works with ARES Commander and Kudo opening the same drawing:

1. First, start ARES Commander. Do not open a drawing yet.
2. Open the Cloud Storage panel:
   a. Right-click the ribbon or a toolbar.
   b. From the list of panels, choose **Cloud Storage**:



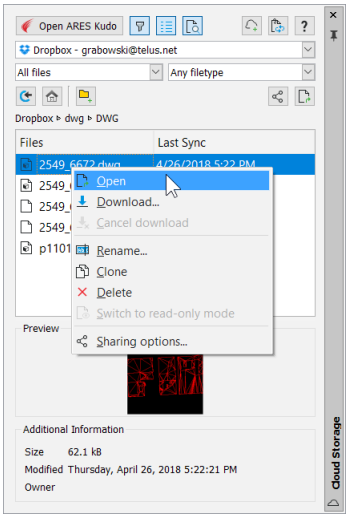
Opening the Cloud Storage panel

3. Near the top of the Cloud Services panel, choose a cloud service from the droplist, such as **Dropbox**.



Choosing a cloud service

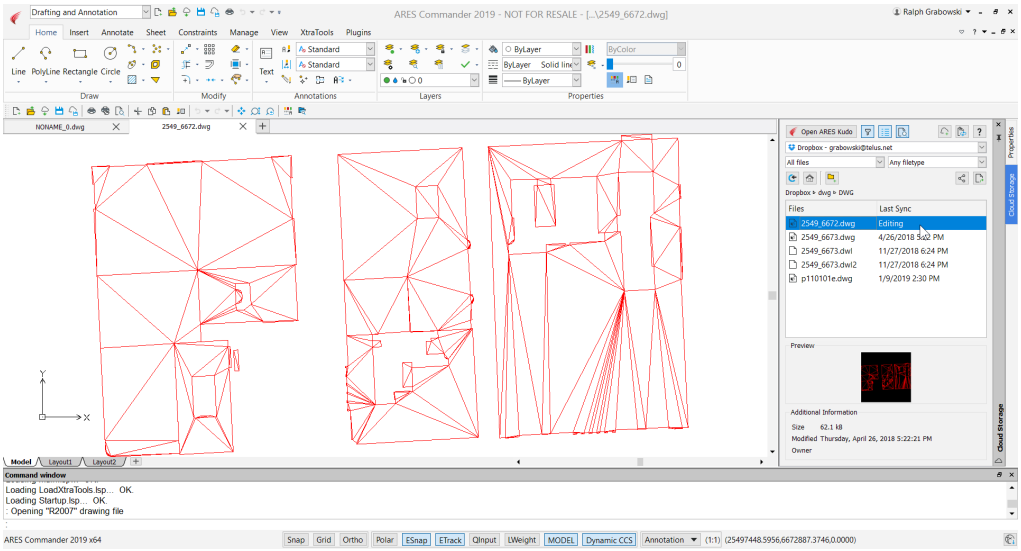
4. Navigate through Dropbox's folders to find the *.dwg* file to open. To open the file:
   » Double-click its name
   » Or, right-click and then choose **Open** from the shortcut menu
   » Or, click the **Open** button



Selecting the .dwg file to open

5. Notice that the drawing is opened in the program, and so can be edited.

   More critically, notice that the Cloud Storage palette changes the status of the drawing to "Editing."



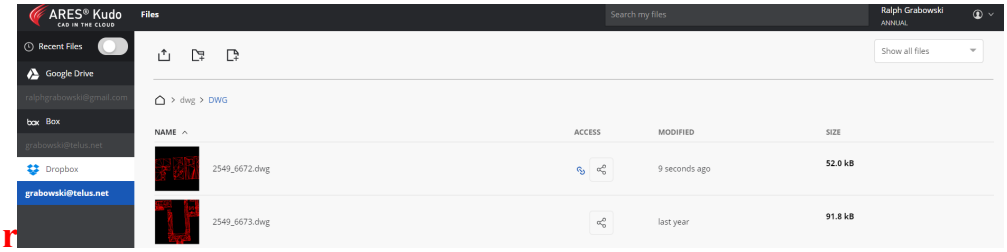
Drawing being editing from Dropbox through the Cloud Storage panel

6. To open the same drawing in another copy of ARES, start Kudo using one of these methods:

   » In the Cloud Storage panel, click the **Open ARES Kudo**  button. This method is useful when you want to run Kudo on the same computer, because it saves you needing to log in to Kudo.

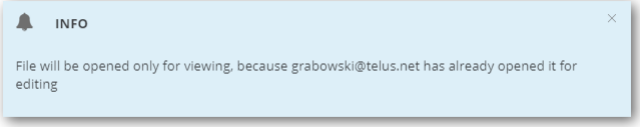
   » Or, in a Web browser's address bar, enter kudo.graebert.com. You need to use this method when running Kudo on a different computer.

7. In the Kudo file manager, go to the same cloud service (Dropbox), and then navigate to the same folder. Notice that the **Access** column for this opened file shows a chain link icon to indicated it is open elsewhere; as well, the **Modified** column shows it was changed very recently.



Kudo opening to the file manager

8. Open the same *.dwg* file. After the Kudo drawing editor opens, notice the info box telling you that the drawing is opened in view-only mode. The drawing is locked against editing, because it is open in another ARES program.



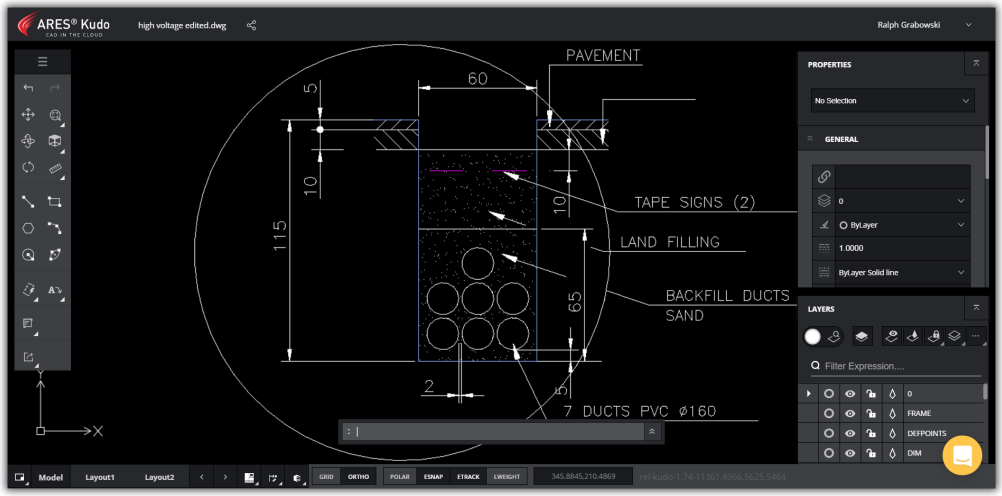
Information dialog box in Kudo

9. Now switch back to ARES Commander:
   a. Make a change to the drawing, such as erasing a large amount of it — so that the change is really obvious!
   b. Commit the change(s) with the **Save** command.
10. Back to the Kudo side of things, see the revised drawing by updating it: press **F5** or using the Web browser's **Refresh** command

Releasing a Locked Drawing

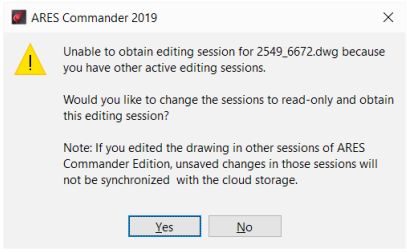
For this tutorial, close the drawing you opened in ARES Command and Kudo, and then follow these steps:

1. In Kudo, open a drawing. Notice that you can edit it.



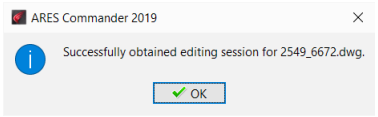
Drawing open for editing in Kudo

2. Now switch to ARES Commander, and then use the Cloud Storage palette to open the same drawing.
3. Notice the dialog box. It offers to let you take over control of the drawing. This means that Kudo will revert to view-only mode and that ARES Commander will be able to edit it. Click **Yes** in the dialog box.



Dialog box for regaining control of the drawing in ARES Commander

4. Notice that ARES Command can now edit the drawing, while Kudo only can view it.



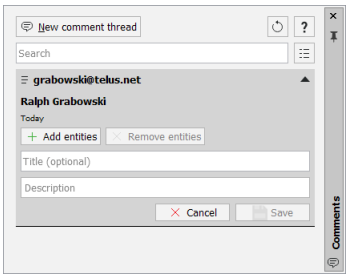
Control regained

Commenting on Drawings

(NEW IN 2019) Comments can be attached to drawing through the Comments palette. This allows one or more people to make suggestions, ask questions, or talk about last night's game scores (just kidding). The system works only with drawings that are opened from the cloud or when local drawings are saved to the cloud.

> **Tip** The Comments command works after (1) when you are signed into your Graebert account and (2) the drawing is opened from the cloud, or else stored online in cloud storage with the **SaveToCloudStorage** command.

The palette is opened with the **Comments** command, closed with the **HideComments** command. It offers the following controls:



Comments palette

Icon	Palette Control	Purpose
	New Comment Thread	
	Refresh	Refreshes the conversation threads
?	Help	Opens the help information about this palette
	Search	Search for comments
	Filter	Filters the conversation threads by type: Show entity comments Show drawing comments Show resolved comments Show unresolved comments Sort by oldest first Sort by newest first
	Collapse	Collapses the thread
+ Add entities	Add Entities	Prompts you to select entities in the command line; press Enter when done selecting entities to comment on
Remove entities	Remove Entities	Removes entities from the selection set
	Resolve	Marks the comment thread as resolved, meaning the issue has been dealt with satisfactorily
	Zoom	Zooms to the entities referred to in the comment thread
...	More Options	Edits the comment or deletes it: Edit... Delete
	Title	(Optional) Gives a heading to the comment thread
	Description	Describes the problem or makes a comment
	Cancel / Save	Cancels the comments and so does not save the changes; saves the comments for others to read and comment on

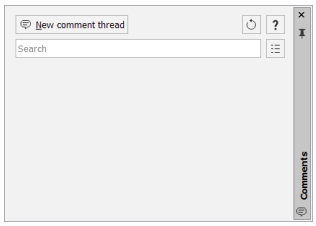
STARTING A COMMENT THREAD

To start or continue a comment session, the drawing must be accessed from the cloud.

1. Open the Comments palette with the **Comments** command. Notice that it is grayed out.

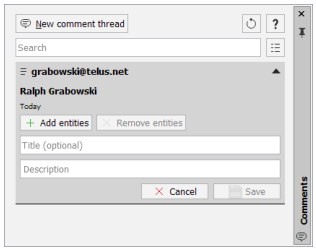
Comments palette unavailable for use

2. From the Cloud Storage palette, open a drawing. Notice that the Comments palette is activated.



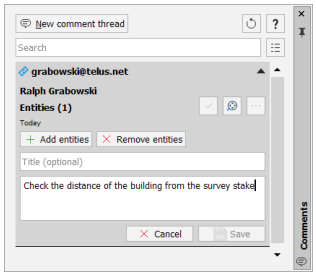
Comments palette activated

3. To write your first comment, click **New Comment Thread**. Notice that the palette fills out with controls.



Comments palette with controls

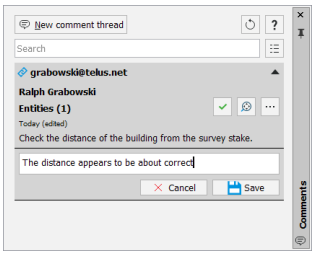
4. Enter an optional title, which would describe the comment thread.
5. Enter your comments:
   a. Type your comment in the **Description** field
   b. If you want the comments to relate to specific entities, click **Add Entities** and then choose one or more of them from the drawing.



Adding a comment to selected entities

   c. Click **Save**.

6. You and other team members can then add their comments in the next Description field that opens up.



Adding more comments to the thread

TIP Click the **Zoom** button to see which entities are being discussed.

Summary

In this chapter, you learned about coordinating drawings with other users through the Cloud Storage palette, and then how to comment on drawings and entities in drawings with the Comments palette.

Notes

PART IV

2D & 3D Mechanical Design

Notes

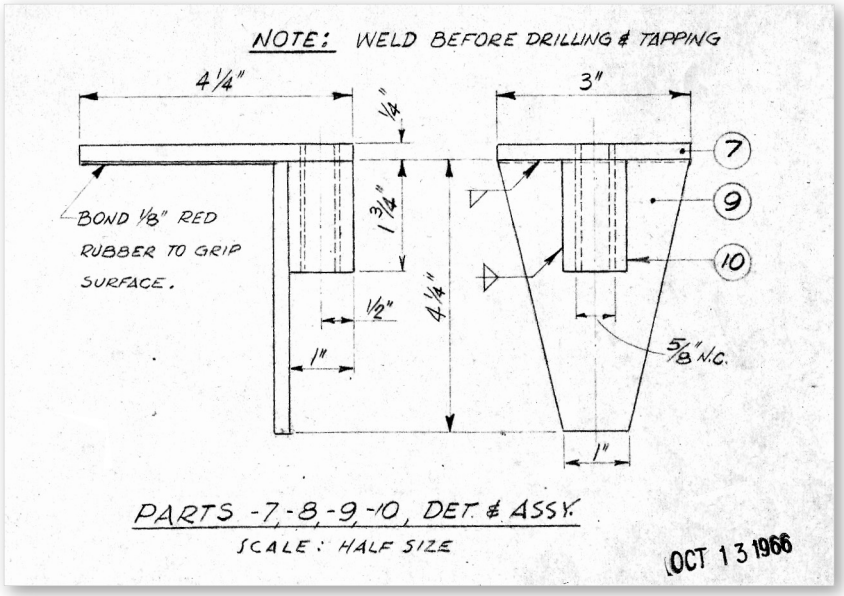
CHAPTER 14

Scan and Trace

In This Chapter

- Scanning paper drawings
- Importing scanned image files
- Correcting images with editing commands
- Tracing over scanned drawings with drawing commands

Companies have rooms filled with paper drawings created in the days before CAD. When these drawings need to be edited, the necessary manual drafting tools may no longer be available. In cases like these, paper drawings are scanned, imported into the program, and then traced over. In this chapter, you learn about inserting raster images, correcting them for rotation and scale, and then tracing over them. The image file (462KB) needed for this tutorial can be downloaded from www.graebert.com/insideares/sample/scan.tif.



Hand-drafted drawing courtesy Herbert Grabowski

Scan of hand-made drawing to be converted to a CAD drawing

KEY TERMS IN THIS CHAPTER

Chamfers replace two intersecting lines with a single line.

Dual dimensions report dimensions in metric and imperial units.

Displacement specifies the relative distance to move entities.

Entity linestyle scale specifies the scale factor for individual entities in a drawing

Frames refers to the rectangles around images.

Full paths specifies the drive and full path to files.

Global linestyle scale specifies the scale factor for all linestyles in a drawing.

Linestyles defines repeating pattern of dashes, dots, and gaps for entities.

Move moves entities by a specified distance.

Reference images refers to the raster image files attached to drawings.

Relative paths describes the path from one file relative to a second file.

Rotate turns entities by a specified angle.

Scale resizes entities larger or smaller.

Scanners convert paper drawings into raster files.

ABBREVIATIONS

A	Imperial size of standard letter paper, 8.5" x 11"
A0	Metric size of largest standard paper sheets, 841 × 1189mm
A4	Metric size of standard letter paper, 210 × 297mm
DIN	Deutsche Industrie Norm (German industrial standards)
E	Imperial size of largest standard paper sheets, 36" x 48"
ISO	International Organization of Standards
TIF	File extension of raster images stored in TIFF format (tagged image file format)

COMMANDS

Command	Shortcuts	Menu Selection
AttachImage	imageattach	Insert | Reference Image
Chamfer	cha	Modify | Chamfer
DisplayImageFrame	...	...[1]
Linestyle	lt, linetype	Format | Line Style
LtScale	...	...
Move	m	Modify | Move
Rotate	ro	Modify | Rotate
Scale	sc	Modify | Scale

[1] No menu selection.

Paper drawings come in a variety of sizes, ranging from the small A4 (210 × 297mm) to the large A0 (841 × 1189mm); these are standard metric sizes that follow the international ISO and Geran DIN standards. In imperial units used in USA and Canada, the standard paper sizes range from A (8.5 x 11") to E (36 x 48").

Low cost, everyday office scanners, such as those built into desktop inkjet printers, are suitable for scanning A4- and A-size drawings. The Canon all-in-one printer-scanner shown at left is a typical example.

Scanners for larger drawings tend to be floor-standing models, such as the image shown courtesy of from Image Access illustrated below. These are much more expensive than desktop models, costing as much as $10,000 for monochrome models and even more for color ones.

Desktop (top) and large-format scanners for digitizing paper drawings

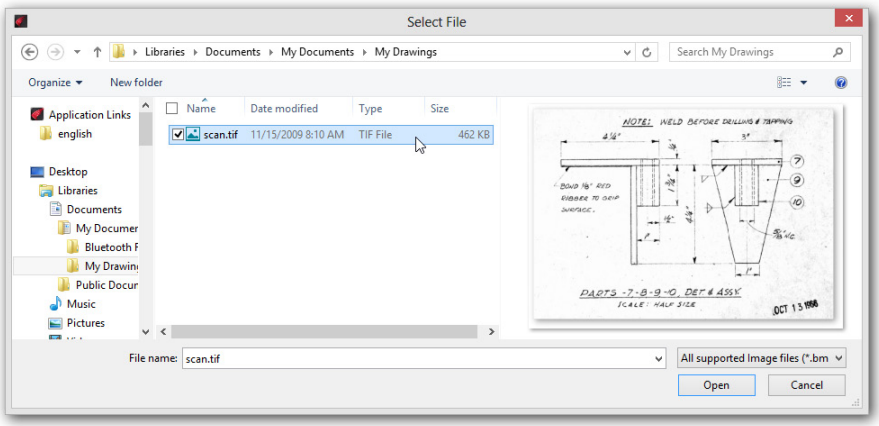
The alternative to owning a scanner is to use a local scanning service, which typically charges by the square foot or meter.

AttachImage: Importing Scanned Images

Command: AttachImage
Aliases: atimg, attachimg, iat, imageattach
Menu bar: Insert | Reference Image
Ribbon: Insert | Attach | Attach Image (Drafting and Annotation)
Toolbar: --

The old drawing is dimensioned in inches, so let's start a new drawing also with Imperial units. Later in this chapter, you see how to make the program display dimensions with dual measurements: Imperial and metric.

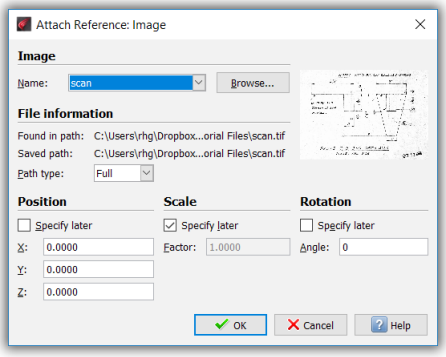
1. Launch the program, and then start a new drawing by entering the **New** command.
2. In the file dialog box, choose the *standard.dwt* template file (which uses Imperial units), and then click **Open**.
3. The next step is to load the scanned image into the drawing. From the **Insert** menu, choose **Reference Image**.
4. In the Select File dialog box, choose the *scan.tif* file, and then click **Open**.



Selecting the scanned file prior to attaching it to the drawing

TIP This command cannot be used with drawings that are opened from the cloud , such as from Dropbox.

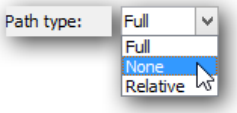
5. Notice the Attach Reference Image dialog box. It looks almost like the one for inserting blocks.



Dialog for specifying how the scanned image will be placed in the drawing

Make the following changes to the dialog box:

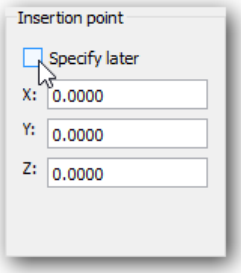
a. Change **Path Type** to **None**. This option disconnects the image from its folder location, which is useful when you need to send the drawing to other users.



Disconnecting the raster file from its location on the computer's hard drive

b. In the **Position** section, turn off the **Specify Later** option. This lets you specify the insertion point now, which is the lower left corner of the image.

c. Set the **Position** coordinates to **0,0,0**. This places the lower-left corner of the image precisely at the origin of the drawing.



Specifying the insertion point for the scanned drawing

d. In the **Scale** section, turn off the **Specify Later** option.

e. Ensure that the **Factor** is set to **1**. This places the image at its full size, with no scaling.

Left: *Setting the scale factor to 1.0;* ***right:*** *Setting the rotation angle to 0 degrees*

d. In the **Rotation** section, leave the **Angle** at 0 degrees.

WHICH IMAGE FILE TYPE TO USE?

the program supports a number of image file types; given the choice, you should always work with images in TIFF format (.tif files); it is unfortunate that most images today are in the less desirable JPEG format.

Extension	Name	Benefits; Drawbacks
BMP	Bitmap	Output by some programs; creates very large files
GIF	Graphics Interchange Format	Popular last decade; limited to 256 colors
TIF	Tagged Image Format	Used by desktop publishing; no limitations
JPG	Joint Photographic Group	Extremely small file sizes through high compression, which also creates distortions
PNG	Portable Network Graphics	Designed to be a distortion-free alternative to JPG,but not in common use

SUMMARY OF ATTACHIMAGE DIALOG BOX OPTIONS

```
: attachimage
(Displays Select File dialog box. Choose an image file; displays Attach
Reference Image dialog box.)
```

IMAGE

Name displays the file name of the selected image; click the droplist to choose from among previously-attached images.

Browse selects image file names from the Select File dialog box. Choose from among *.bmp*, *.gif*, *.tif*, *.jpg*, and *.png* file types.

FILE INFORMATION

Path Type specifies how the program stores the path to the image file:

- **Full** stores the complete drive and path name.
- **Relative** stores the relative path to the image file.
- **None** stores the file name only, no path.

INSERTION POINT

Position (formerly Insertion Point) specifies the location of the lower left corner of the image:

- **Specify Later** prompts you for the insertion point in the Command window after the dialog box is closed, and so it allows interactive placement.
- **X, Y, Z** specify the x, y, and x coordinates for the image.

SCALE

Scale specifies the size of the image:

- **Specify Later** prompts you for the scale factor in the Command window after the dialog box is closed, and so it allows interactive sizing.
- **Scale factor** specifies the size of the image; values larger than 1.0 increase the size; those between 0.0 and 1.0 shrink the size.

ROTATION

Rotation specifies the angle of the image:

- **Specify Later** prompts you for the angle in the Command window after the dialog box is closed, and so it allows interactive rotation.
- **Angle** specifies the angle of rotation; positive angles rotate the image counterclockwise, starting from the positive x axis (0 degrees).

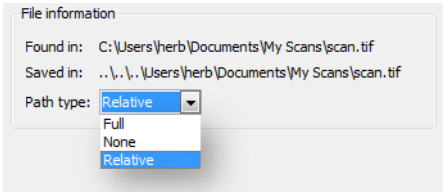
6. Click **OK** to exit the Attach Reference Image dialog box. Notice that the image appears in the drawing, but that it looks quite small.

Scanned image placed in the drawing at its origin

ABOUT PATH TYPES

The program remembers the path to the source image files in these ways:

Full Path remembers the entire path, from drive to folders to file name. This option is best when the drawing and image files are on different computers, because a drive name can include the name of the network. An example of an absolute path is shown next to **Found In** in the figure below.



Choosing path types for attached images

The full path option fails, however, when the DWG file is moved to a different computer or even to another folder, because the path becomes invalid instantly.

Relative Path remembers the relative path between the DWG file and the image file. Use this path type if the drawing and image are moved (together) among computers; keep the image files in a folder inside the folder holding the DWG file. An example of a relative path is shown next to **Saved In** in the figure above. Each use of the "..\" metacharacter means the program should look "one folder higher" from the current one.

This option does not work, however, when the DWG and image files are on different computers. Also, this option does not work when the drawing is not yet saved.

No Path remembers only the image's file name.

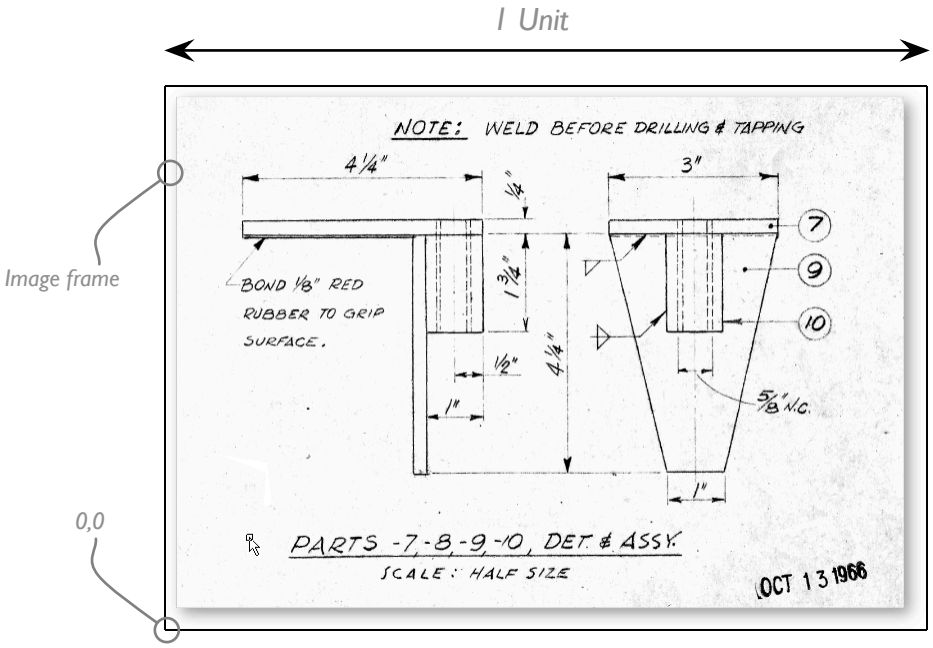
7. To enlarge it so that you can see it better, enter the **Zoom Fit** command.
8. Save the drawing with the **SaveAs** command, naming it *scan.dwg*

When images are placed in drawings, they are known as *attached* or *referenced* images. This is because the image files are not a part of the drawing; instead, the drawing contains a reference to the image files in the form of a path such as *C:\Users\Documents\My Scans*. The program does this to keep the size of the drawing file smaller, because image files can be very large in size.

Each time the program opens a drawing with an image attached, it refers to the path to find the image, and then loads it into the drawing. See the boxed text for more on paths.

EXAMINING THE ATTACHED IMAGE

Now that the scanned image is in the drawing, let's take a look at some of its characteristics:



Elements of the imported scan

- The lower left corner is at 0,0 — the insertion point).
- The width is 1 unit — 1 inch, in this case.
- The image is surrounded by a black rectangle — called the "frame."

To select the image, you must click the *frame*; you cannot select the image by clicking the image itself!

TIP You can turn image frames off and on with the **DisplayImageFrame** command. It is normal to turn off the frames prior to plotting so that excess black rectangles do not appear on the paper. When frames are turned off, however, you can no longer select the image; for this reason the frame is usually left on at all other times.

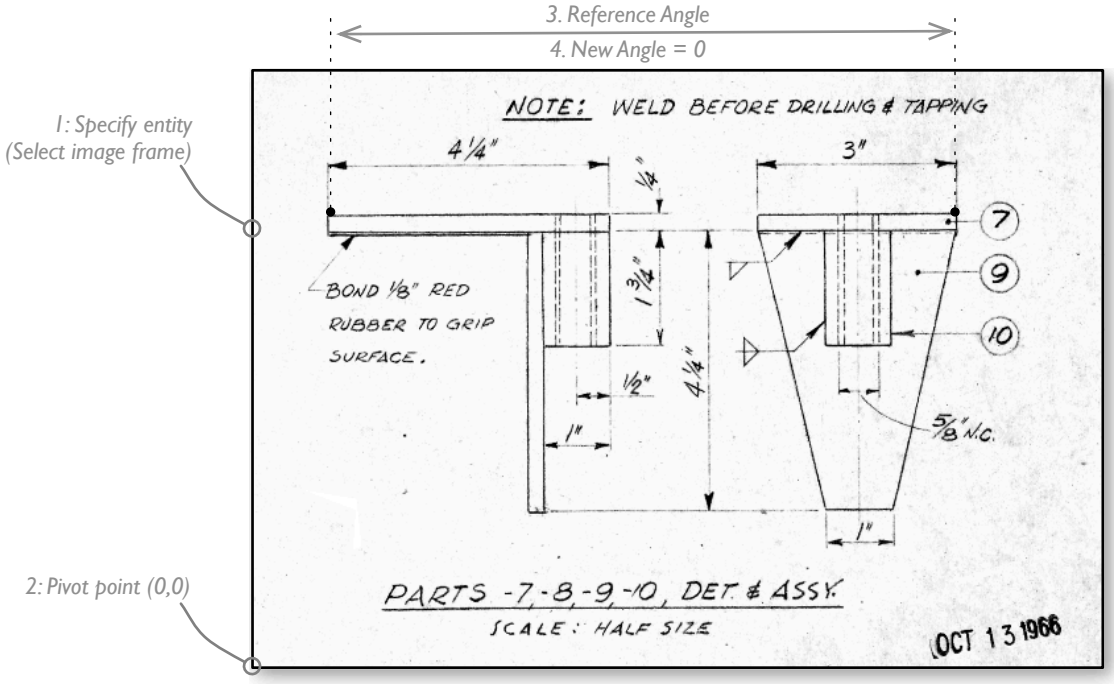
The image is not perfectly straight; it is tilted horizontally and vertically. As well, it is the wrong size (scale factor). You fix these problems in the next section.

ROTATE: ROTATING ENTITIES

Command: Rotate
Alias: ro
Menu bar: Modify | Rotate
Ribbon: Home | Modify | Copy | Rotate (Drafting and Annotation)
Toolbar:

Paper drawings are often not perfectly aligned when they are scanned. So, you use the Rotate command to straighten them out. Rotate is one of many editing commands that work with images; others include Scale, Copy, Move, Array, and Erase.

To rotate the image, follow these steps:



Strategy for straightening out the skewed scan

1. Start the **Rotate** command using one of the methods listed above:

```
: rotate
Active positive angle in CCS: DIRECTION=counterclockwise BASE=0
```

2. Select the image by clicking on its frame, or by entering **L** (short for "last" object drawn), and then press **Enter** to end entity selection:

```
Specify entities» L
1 found, 1 total
Specify entities» (Press Enter to continue)
```

3. The *pivot point* indicates the spot about which the image is rotated. For this tutorial, it doesn't matter much where it is located, but 0,0 is a convenient spot.

```
Specify pivot point» 0,0
```

4. Use the **Reference** method to specify the rotation angle. The word "reference" in this command is unrelated to the image being a reference image. (You use this same Reference system later to scale the image to the correct size.)

   This method of using a reference angle is a bit tricky. I find I end up trying it a couple of times before I get it right. In summary, this is how the it works:

   Step 1. Specify a Reference Angle — pick two points in the drawing that indicate the current angle.

   For this tutorial: pick the endpoints of a line that *should* be horizontal, but isn't.

   Step 2. Specify the New Angle — enter the angle you want the line to become.

   For this tutorial, the angle is 0, because we want lines that should be horizontal to become actually horizontal.

   Follow the above steps to straighten out the image:

   a. Enter **r** to specify the Reference option:

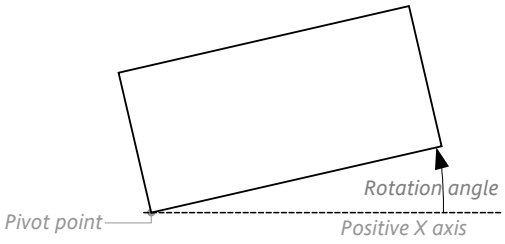
   ```
   Options: Reference or
   Specify rotation angle» r
   ```

   b. To indicate the current angle, pick two points: in this case, pick a point at either end of the two horizontal lines illustrated above.

   ```
   Default: 0
   Specify reference angle» (Pick one end of the 4-1/4" line)
   Specify second point» (Pick the far end of the 3" line)
   ```

SUMMARY OF ROTATE COMMAND OPTIONS

```
: rotate
Active positive angle in CCS: DIRECTION=counterclockwise BASE=0
Specify entities» (Choose one or more entities.)
Options: Reference or
Specify pivot point» (Pick a point.)
Specify rotation angle» (Specify an angle.)
```



Pivot Point specifies the point about which the selected entities rotate.

Rotation Angle specifies the angle by which to rotate the entities; measures angle from the positive x axis (0 degrees).

Reference changes the starting angle (overrides 0 as the starting angle):

```
Specify reference angle» (Pick two points, or enter an angle.)
```

Use the **Units** command to change the default direction (counterclockwise) and base point (0,0).

c. Now enter the angle that this horizontal line should be — 0 degrees:

```
Default: 0
Specify new angle» 0
```

5. Notice that the image rotates slightly as the hand-drafted content of the image becomes orthogonal.

The image frame, however, looks crooked. You solve this by turning off the display of the frame with the **DisplayImageFrame** command, and then entering the **n** (No image frame) option:

```
: displayimageframe
Default: Above image
Confirm: Display image frames?
Specify No, Above image, or Below image» n
```

TIP You cannot use entity snaps with images, because images are made up of small squares (called "pixels," short for picture elements) of varying shades of gray. (They are not vector entities, and so there is nothing to which to snap.) The figure below is a zoomed-in view of the image; lines, text, and arrowheads are clumps of pixels that fool the eye into seeing lines and text.

SCALE: RESIZEING ENTITIES

Command: Scale
Alias: sc
Menu bar: Modify | Scale
Ribbon: Home | Modify | Copy | Scale (Drafting and Annotation)
Toolbar:

The scanned image is just one inch wide; you need to change it to its actual size with the Scale command. The best approach is to use the longest known distance as a reference; in this case, it is 4-1/4"-long line. When you tell the program that you want this image to be a different size, you tell it to make the 4-1/4"-long line to be 4-1/4" long.

Even though the drawing was originally drafted at half-scale, you can ignore this, because you will use actual dimensions from the drawing.

To correct the size of the reference image, follow these steps:

1. Start the **Scale** command using one of the methods listed above, and then select the image again with the **L** option.

   ```
   : scale
   Specify entities» L
   1 found, 1 total
   Specify entities» (Press Enter to continue)
   ```

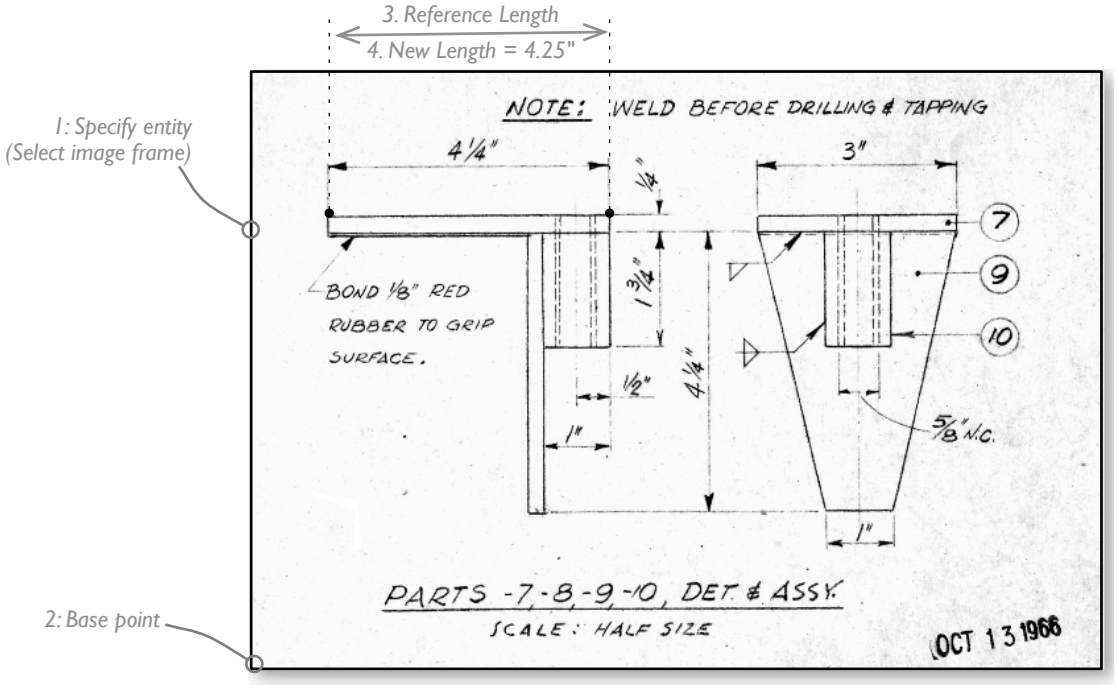
   (If you turned off the frame, then you have to use the Last option to select the image, since you can no longer pick the frame.)

2. The *Base Point* is the same as the Rotate command's Pivot Point, which is **0,0**:

   ```
   Specify base point» 0,0
   ```

3. Again, you use the Reference option. This time, you pick two points that represent the future (or true) length, and then enter the length you want it to be:

   ```
   Options: Reference or
   Specify scale factor» r
   Default: 1.0000
   Specify reference length» (Pick one end of the 4-1/4" line)
   Specify second point» (Pick other end of the line)
   Default: 1.0000
   Options: Points or
   Specify new length» 4.25
   ```



Strategy for making the scan the correct size

TIP To make more accurate picks in the drawing, scroll the mouse's wheel to zoom into the line, making it larger on the screen.

4. Notice that the image becomes much larger. Use the **Zoom Fit** command to refit the image to the drawing window.
5. To prove to yourself that the image is now at the correct angle and size, use the **GetDistance** command:

```
: getdistance
Specify start point» (Pick one end of the 4-1/4" line)
Specify end point» (Pick other end of the line)
```

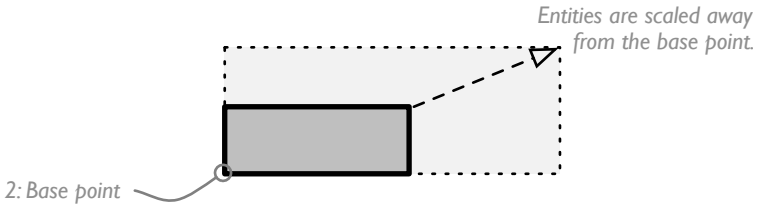
The program reports that the distance is (almost) 4.25 units long, and that the angle is 0 degrees:

```
Distance = 4.2837, Angle in XY Plane = 0, Angle from XY Plane = 0
Delta X = 4.2837, Delta Y = 0.0000, Delta Z = 0.0000
```

6. Press **Ctrl+s** to save your work (Cmd+s on Macs).

SUMMARY OF SCALE COMMAND OPTIONS

```
: scale
Specify entities» (Select one or more entities)
Specify base point» (Pick a point about which to scale the entities)
Options: Reference or
Specify scale factor» (Specify the scale factor)
```



Base Point is the point relative to which the selected entities are scaled.

Scale Factor specifies the amount by which to size the entities:

- Values larger than 1.0 increase the size; e.g. 2.0 doubles the size.
- Values between 0.0 and 1.0 decrease the size; e.g. 0.5 halves the size.

Reference scales without you needing to know the scale factor:

```
Specify reference line» (Pick two points.)
```

When the base point is not on the entity, the program determines the scale factor from the distance between the base point to all points on the entity.

Tracing Over Scanned Images

To "convert" the paper drawing to vectors, you trace over the image using the drawing commands found in the program.

The difficulty is this: paper drawings created by hand have inherent errors. As I noted at the start of Chapter 14, the "DO NOT SCALE" warning placed by drafters on their hand-made drawings meant that no one should not measure lengths of items with a ruler. This is why dimensions were invented.

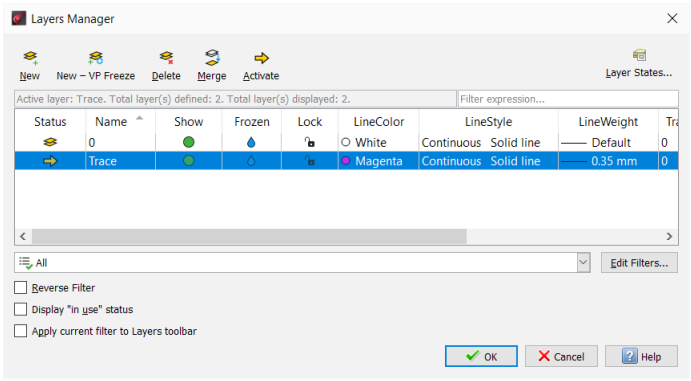
Because the scanned paper drawing is inexact, you'll begin tracing it from one corner as the reference point. Then, you use the dimensions in the drawing to determine the lengths of CAD entities.

LAYERS AND RECTANGLES

In this tutorial, you create a new layer for tracing over the scan, and then use several drawing and editing commands to replicate the paper drawing. Continue with the drawing from the previous tutorials in this chapter:

1. Create a new layer on which you will do the tracing:
   a. Enter the **LayersDialog** command to open the Layers Manager dialog box.
   b. To create a new layer, click the New **New** button, and then name the new layer "Trace."
   c. To make the CAD entities really stand out, change the **LineColor** of the Trace layer to **magenta** (bright pink!).
   d. As well, change the **LineWeight** to a heavier value, such as **0.35mm**.
   e. To make the layer active, click the **Activate** button.

   When you are done, the dialog box should look like this.



Creating the "Trace" layer in the Layers Manager dialog box

   f. Click **OK** to close the dialog box.

2. On the status bar, click **LWT** to turn on lineweights. Together with the magenta color, this makes the new CAD drawing easier to see over top of the scanned image.

Drawing the Side View

This scanned drawing has two views, the side view on the left hand side and the front view on the right. Each view is constructed of many rectangles, and so you use the **Rectangle** command many times.

1. The rectangles connect with each other at intersections, and so it makes sense to turn on Intersection entity snap, as follows:

   a. On the status bar, right-click the **ESnap** button.

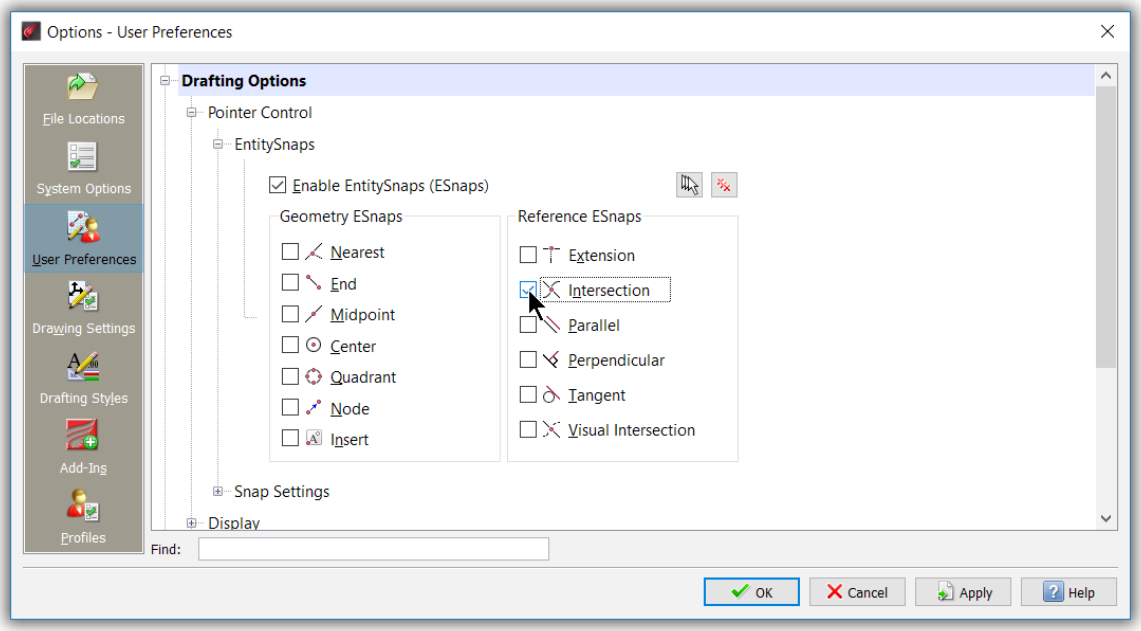
   b. In the shortcut menu, choose **Settings**. Notice that the Options dialog box opens to the Entity Snap settings, conveniently.

   c. In the Options dialog box, turn on the following options:

      » **Enable EntitySnaps (Esnap)**

      » **Intersection**

TIP Here is the fast way to set just one entity snap: first, turn them all off with the **Clear All** button. Then turn on **Intersection**.



Turning on Intersection entity snap mode

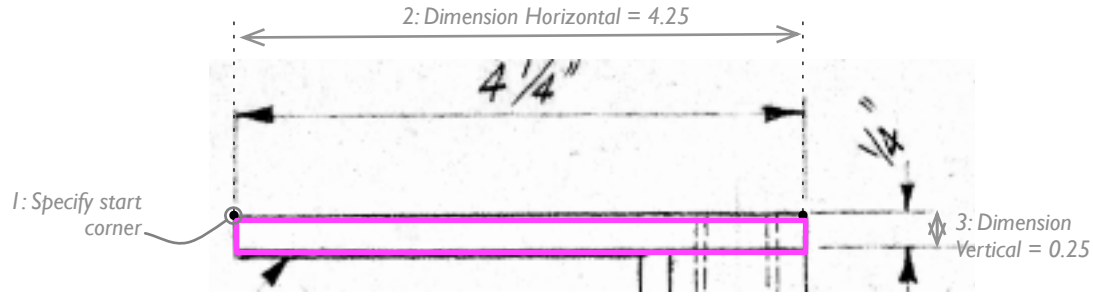
   d. Click **OK** to close the dialog box.

      On the status bar, the **ESnap** button should be glowing blue.

2. To create an accurate CAD drawing from the image, you need to choose a point on the scanned image from which to start. You draw starting from this point, employing the actual dimensions as reported by the scan.

a. Start the **Rectangle** command, and then pick the point shown in the figure below:

```
: rectangle
Options: Chamfer, Elevation, Fillet, Thickness, line Width or
Specify start corner» (Pick a point)
```



Tracing the scan element with the rectangle command

b. Select the **Dimensions** option. Specify the dimensions shown in the figure: **4.25"** by **-0.25"**.

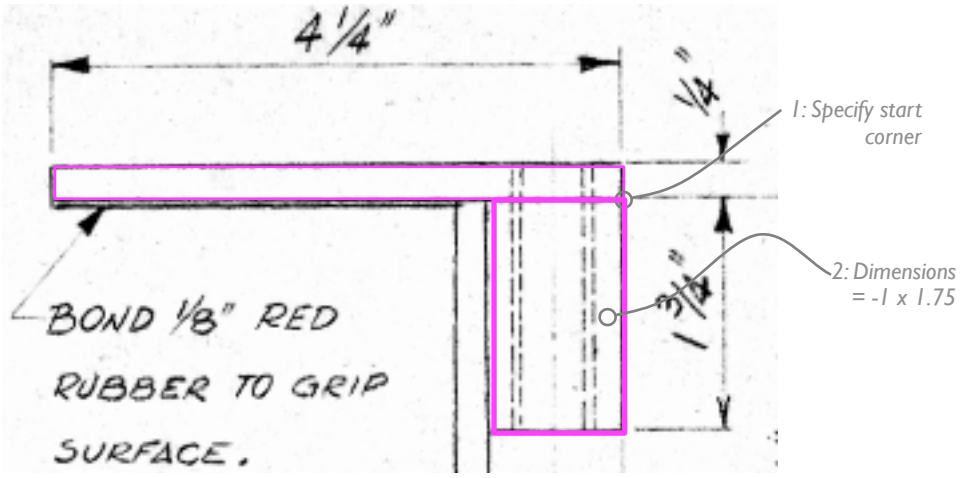
```
Options: Area, Dimensions, Rotation or
Specify opposite corner» d
```

c. Enter the values shown by the scale: 4.25 by 0.25. When you enter negative values for the dimensions of a rectangle, the program draws the rectangle downwards from the starting corner.

```
Default: 10.0000
Specify horizontal dimension» 4.25
Default: 10.0000
Specify vertical dimension» -.25
```

Notice that the perfectly drawn CAD rectangle does not match the hand-drawn one; this is to be expected.

3. Attach the next rectangle to the first one, ignoring the inaccuracies in the handmade drawing.



Tracing the second rectangular shape

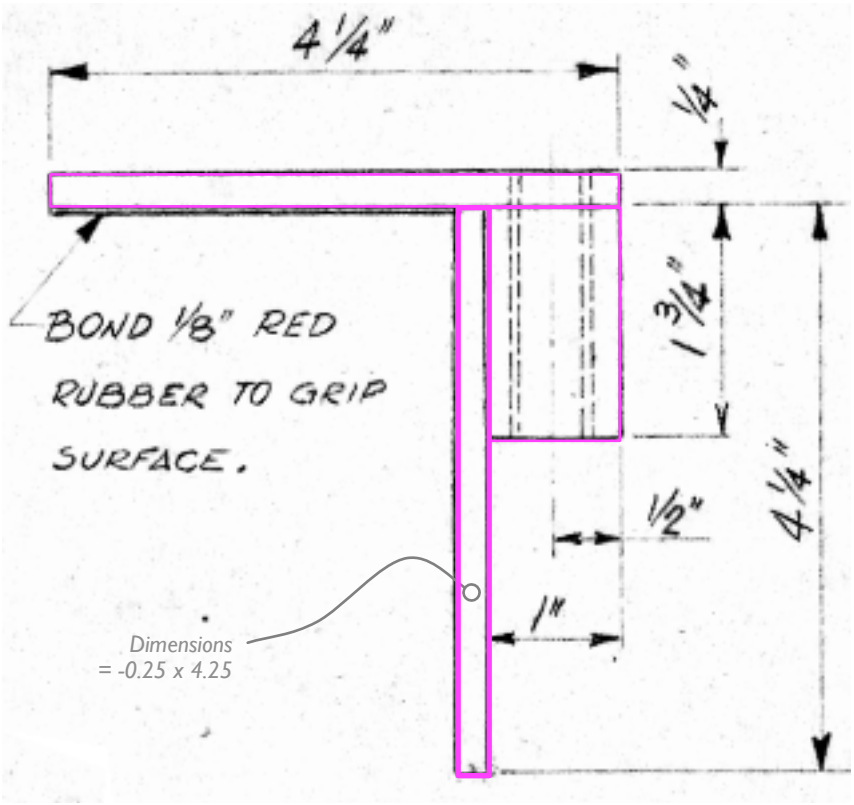
```
: rectangle
Specify start corner» (Pick the point shown by the figure above)
```

a. The dimensions indicate the rectangle is 1.75" tall by 1" wide; enter negative values (**-1.75** and **-1**) so that the program will draw the rectangle downwards and to the left from the start point.

```
Specify opposite corner» d
Specify horizontal dimension» -1
Default: 10.0000
Specify vertical dimension» -1.75
```

Entity snap ensures the new rectangle begins precisely at an intersection of the first one:

4. Press the spacebar to draw the next rectangle, using dimensions of **-0.25** and **-4.25**.



Tracing the final rectangle

5. The red rubber strip is the final rectangle, whose dimensions can be worked out as **-3** and **-0.125**.

6. Press **Ctrl+s** to save your valuable work (**Cmd+s** on Macs).

(The dashed lines are drawn later in this chapter.)

Drawing the Front View

Command: Move
Alias: m
Menu bar: Modify | Move
Ribbon: Home | Modify | Copy | Move (Drafting and Annotation)
Toolbar:

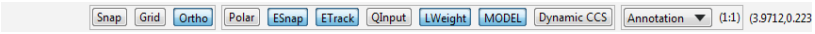
The front view is similar to the side view, because it has two rectangles. But it differs from the side view in that the rectangles are centered on each other. And there are those angled lines to deal with.

Drawing one rectangle centered on another is tricky. There are several approaches you could take, but I feel the easiest one is draw-and-move: draw a rectangle in a convenient location, and then move it into place.

Another issue is that the front view should be aligned precisely with the side view. To draw the initial rectangle, you use *entity tracking* to line up the initial rectangle with one in the side view.

Both of these techniques are illustrated in the following steps:

1. Turn on ortho by clicking **Ortho** on the status bar.
2. Turn on entity tracking by clicking **ETrack**. Both buttons should look blue.



Ortho and ETrack (entity tracking) buttons turned on

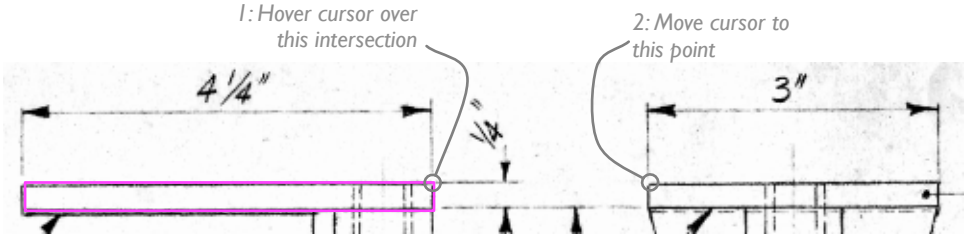
3. Use entity tracking to position the front view's first rectangle relative to those in the side view.
   a. Start the **Rectangle** command.

      `: rectangle`

   b. At the 'Specify start corner' prompt, move the cursor so that it hovers over the upper right corner of the rectangle illustrated by point #1 in the figure below. *Do not click!*

      `Specify start corner» (Move the cursor over point 1 in the figure below, but do not click!)`

   `Just pause the cursor at the intersection; the program knows you want to acquire this point.

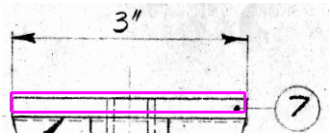


Using entity tracking to start another rectangle in line with the first

   c. After a moment, move the cursor horizontally to point #2, and then click. The program places markers to indicate the entity tracking snap points. They are hard to see, a green-colored plus sign (+) over point #1, and a red-colored cross (x) over point #2.

Using entity track to start drawing a distance from another point

d. Use the **Dimensions** option to size the rectangle: **3** horizontal and **-.25** vertical.

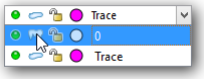


Sizing the rectangle

3. Turn off **Ortho** mode.

TIP When you find the entities in the scanned image getting too busy, freeze its layer, as follows:

a. From the Layers toolbar (or ribbon panel), click the layer droplist.



b. Click the puddle (a melted icicle!) icon next to layer 0.

Notice that the scanned drawing disappears, and that the cloud icon looks like icicles; this indicates the layer is frozen (can be neither seen nor edited).

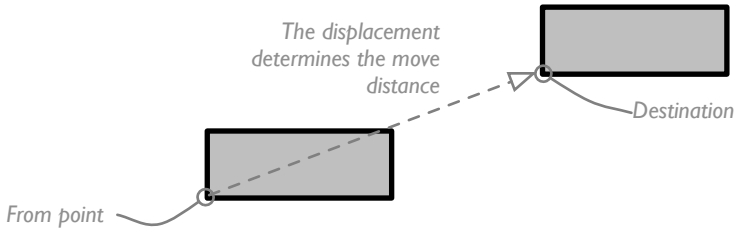
MOVE: MOVING ENTITIES

1. The second rectangle is centered below the first one. In this case, you draw the rectangle "anywhere," and then move it into place:

a. Draw a rectangle with dimensions **1** and **-1.75**. Its location in the drawing does not matter.

SUMMARY OF MOVE COMMAND OPTIONS

```
: move
Specify entities» (Choose one or more entities)
Options: Displacement or
Specify from point» (Specify a starting point)
Options: Enter to use from point as displacement or
Specify destination: (Specify the ending point)
```



Displacement specifies the relative distance to move the selected entities.

From Point specifies the starting point to determine the move distance; can use x,y or x,y,z coordinates.

Enter uses the distance between the original (0,0) and the 'From point' coordinates to determine the distance to move entities.

The 'From Point' does not need to be on the entities being moved. Use entity snaps for precise moves.

b. Start the **Move** command with one of the methods noted above:

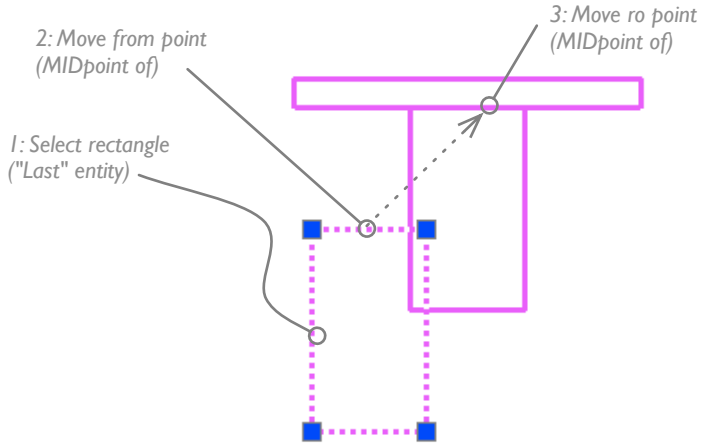
```
: move
```

c. Use the **Last** entity selection mode to select the rectangle you just drew:

```
Specify entities» L
Specify entities» (Press Enter to continue)
```

d. Use **Midpoint** entity snap to grab the rectangle at point #2, illustrated below:

```
Options: Displacement or
Specify from point» mid
of (Pick the center of the rectangle's top line)
```

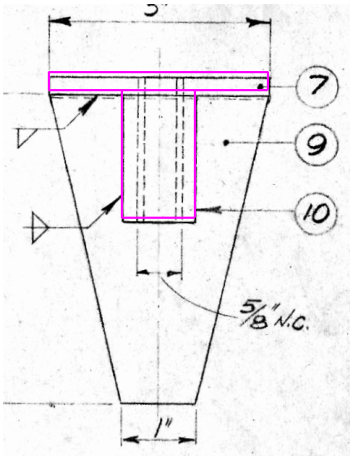


Moving a rectangle

e. Repeat **Midpoint** entity snap to grab the other rectangle at point #3:

```
Options: Enter to use from point as displacement or
Specify destination» mid
of (Pick the center of the rectangle's bottom line)
```

Notice that the moved rectangle snaps into place.



Rectangle moved into place

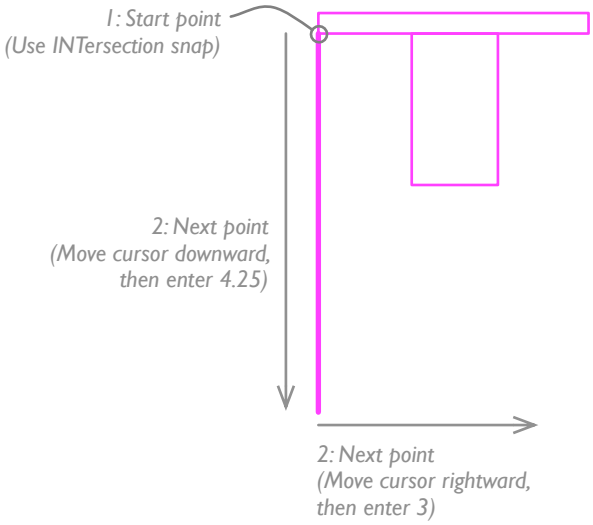
2. Press **Ctrl+s** to save the drawing (**Cmd+s** on Macs).

TIP As an alternative to the procedure described above, you could have used the **Copy** command to copy the existing 1x1.75 rectangle from the side view to the front view. This illustrates that there are many different ways to accomplish the same result in CAD.

Chamfer: Chamfering Entities

Command: Chamfer
Alias: cha
Menu bar: Modify | Chamfer
Ribbon: Home | Modify | Trim | Chamfer (Drafting and Annotation)
Toolbar:

The trapezoid shaped part can be drawn by several methods. I feel the easiest is to draw three lines with direct distance entry, and then use the Chamfer command to create the angles.



Using direct distance entry to draw a line

1. Turn on **Ortho** mode.
2. Start the **Line** command. Draw three lines using direct distance entry:
   a. Move the mouse downwards. Notice that ortho mode keeps the line perfectly vertical.
   b. On the keyboard, enter the length of the line, **4.25**, and then press **Enter**. Notice that the program draws the line.
   c. Move the mouse to the right.
   d. Enter **3**, and then press **Enter**.
   e. Move the mouse upwards, and then enter **4.25** again.
   f. Press **Enter** twice to end the Line command. The three lines are in place.

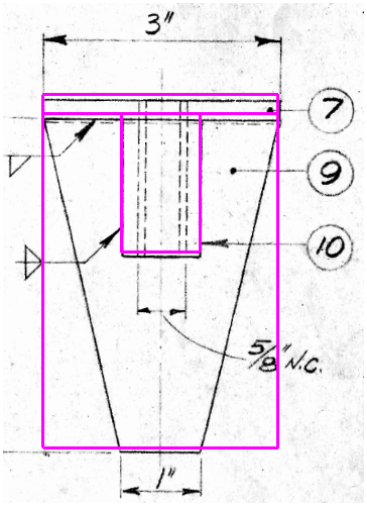
Here is the command sequence:

```
: line
Options: Segments or Specify start point» (Pick at point 1, as shown in the figure above)
Options: Segments, Undo, Enter to exit or Specify next point» (Move the cursor downwards, and then enter:) 4.25
Options: Segments, Undo, Enter to exit or Specify next point» (Move the cursor to the right, and then enter:) 3
Options: Segments, Undo, Close, Enter to exit or Specify next point» (Move the cursor upwards, and then enter:) 4.25
Options: Segments, Undo, Close, Enter to exit or Specify next point» (Press Enter to end the command)
```

The **Chamfer** command works like Fillet, except that it connects two lines with a straight line instead of an arc. While arc fillets are defined only by a radius, the chamfer line can be defined by distances or angles:

How chamfering works

In this case, chamfer by distance is the easier approach. By examining the drawing's dimensions, you can determine that the chamfer distances are 4.25 x 1.



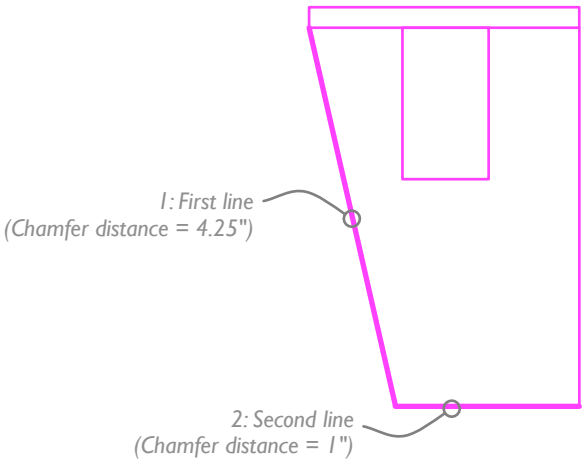
Lines before being chamfered

3. Start the **Chamfer** command with one of the methods listed above.

```
: chamfer
```

   a. At the 'Specify first line' prompt, enter the **Multiple** option, because you have two chamfers to perform.

```
(Trim mode) Active chamfer Dist1 = 0.0000, Dist2 = 0.0000
Options: Angle, Distance, mEthod, Multiple, Polyline, Trim mode, Undo or
Specify first line» m
```



Chamfer distances

b. To specify the chamfer distances, enter the **Distance** option:

```
Specify first line» d
Default: 0.0000
Specify first distance» 4.25
Default: 4.2500
Specify second distance» 1
```

c. Now you chamfer the first pair of lines. Remember that the first chamfer distance (4.25) applies to the first line you pick (the vertical line):

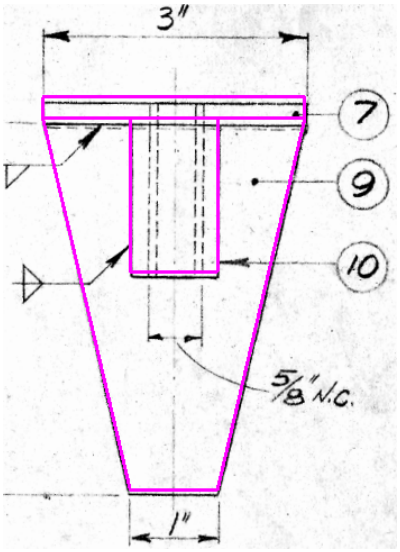
```
Specify first line» (Pick 4.25"-long line)
Options: Shift + select to apply corner or
Specify second line» (Pick 3"-long line)
```

d. Repeat for the other side:

```
Specify first line» (Pick 4.25"-long line)
Specify second line» (Pick 3"-long line)
```

e. Press **Enter** to end the Chamfer command:

```
Specify first line» (Press Enter)
```



Rectangle chamfered

4. Press **Ctrl+s** to save the drawing (**Cmd+s** on Macs).

SUMMARY OF CHAMFER COMMAND OPTIONS

```
: chamfer
(Trim mode) Active chamfer Dist1 = 0.0000, Dist2 = 0.0000
Options: Angle, Distance, mEthod, Multiple, Polyline, Trim mode, Undo or
Specify first line» (Enter an option, or pick a line.)
Options: Shift + select to apply corner or
Specify second line» (Pick the other line.)
```

Angle specifies a length and an angle of the chamfer.

Distance specifies two lengths for the chamfer; enter 0 for both to clean up intersecting lines.

mEthod toggles the default chamfer method: length and angle, or two lengths.

Multiple repeats the command to chamfer more than two entities.

Polyline chamfers all vertices of polyline-based entities, such as polylines, polygons, and rings.

Trim mode toggles trimming of entities. When on, portions of entities beyond the chamfer are cut off; when off, the entities are preserved.

Undo undoes the previous chamfer when Multiple mode is active.

Shift+select applies chamfer distances of 0, overriding the default chamfer distances.

SUMMARY OF LINESTYLE DIALOG BOX OPTIONS

```
: linestyle
(Displays the Line Style dialog box)
```

Show determines which linestyle names to display in the dialog box:

- **All LineStyles** lists all linestyles loaded into the current drawing.
- **LineStyles in Use** lists those linestyles used by entities and layers.
- **References LineStyles** listed linestyles used by referenced drawings.

Load opens the Load Linestyles dialog box; choose one or more to load into the drawing.

Delete removes the selected linestyle from the drawing; in-use linestyles cannot be erased.

Activate makes the selected linestyle active.

Show Parameters expands the dialog box to show rarely-used options:

Name edits the linestyle's name.

Description edits the linestyle's description.

Global Scale specifies the scale factor for all linestyles in the drawing.

Entities Scale specifies the scale factor for new entities relative to the global scale factor.

Scale Based on Sheet's Units applies the Viewport scale factor to linestyles; when off, linestyles have the same scale factor in model and sheet modes.

Reverse Filter reverses the list of names displayed by the Show droplist.

Linestyle: Drawing Patterned Lines

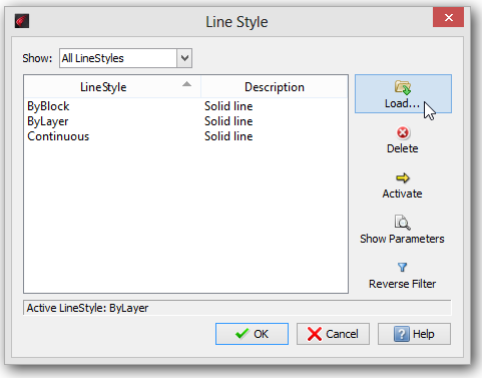
Command: Linestyle
Aliases: style, linetyle, ltype, lt
Menu bar: Format | Line Style
Ribbon: Home | Properties | Linestyle (Drafting and Annotation)
Toolbar: --

Some of the lines in the handmade drawing are dashed. Traditionally, dashes represent hidden lines. In this case, the dashed lines represent holes through the steel parts.

In the program, dashed lines are drawn with the usual commands, such as Line and Offset. But two initial steps are required: (1) the dashed linestyle must be loaded into the drawing with the **LineStyle** command, and (2) the active linestyle (linetype) must be set to Dashed.

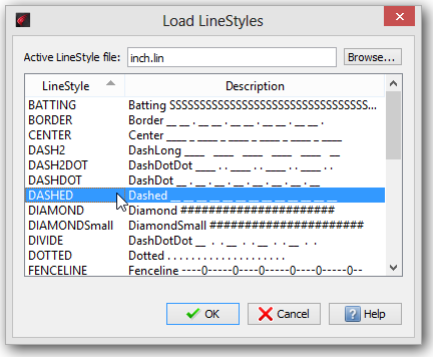
Follow these steps to load the Dashed linestyle into the drawing:

1. Start the **Linestyle** command using one of the methods listed above. Notice the Line Style dialog box. It lists the names of linestyles currently stored in the drawing — all of them are solid lines.



Dialog box listing line styles (linetypes) currently loaded in the drawing

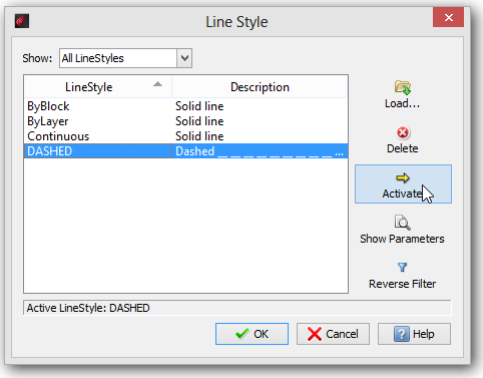
2. Click the Load... **Load** button. Notice the Load LineStyles dialog box with its list of linestyle names and pattern samples:



All available line styles

3. Choose DASHED, and then click **OK**. This action loads the linestyle definition for dashed lines to the drawing.

4. Back in the Line Style dialog box, choose **DASHED**, and then click **Activate**.



Dashed line style added to drawing

5. Click **OK** to close the dialog box.

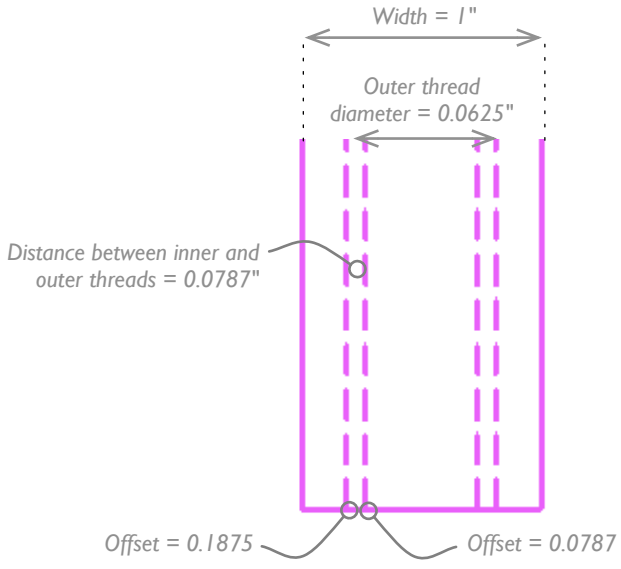
With the needed linestyle loaded, you can now determine how to draw the needed lines. I suggest using the **Offset** command.

From the drawing, you can determine the offset distances. The *5/8" N.C.* notation means that the hole is 5/8" in diameter. N.C. is the abbreviation used by the American National standard indicating a "coarse thread."

The 5/8" diameter is factional; in decimal notation, it is 0.625". This represents the outer diameter of the hole; the inner diameter is determined by applying a 60-degree line to the thread's pitch distance. I won't force you to go through the mathematics; the distances are illustrated below:

Thus, the offset distances you need to draw the dashed lines are as follows:

- 0.1875"
- 0.0787"



Determining the offset distances needed to simulate bolt threads

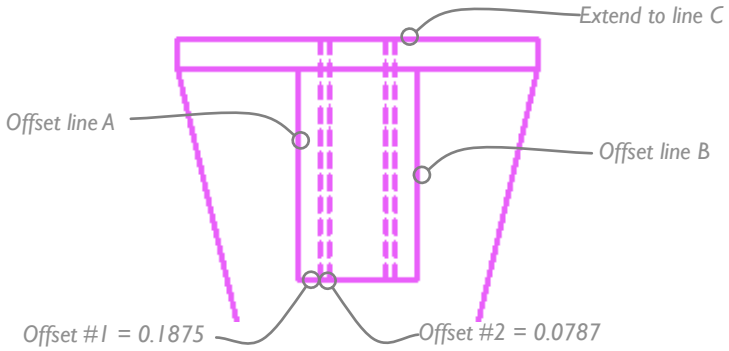
1. Use the **Explode** command to explode the rectangle, otherwise the Offset command will not work correctly.

```
: explode
Specify entities» (Select the rectangle)
Specify entities» (Press Enter)
```

2. Start the **Offset** command, and then specify the offset distances. First, work with the two lines that are offset by 0.1875".

```
: offset
Specify distance» .1875

Specify source entity» (Pick line A)
Specify side for destination» (Pick inside the rectangle)

Specify source entity» (Pick line B)
Specify side for destination» (Pick inside the rectangle)

Specify source entity» (Press Enter to exit)
```

Repeat the Offset command to make the two inner lines that are offset by 0.0787.

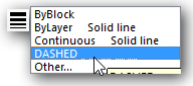


Offset locations and distances

3. The lines need to extend to the topmost line, which is accomplished with the **Extend** command. Refer to the figure above.

```
: extend
Specify boundary edges» (Pick line B)
Specify boundary edges» (Press Enter to end entity selection)

Specify segments to extend» (Pick the dashed lines)
Specify segments to extend» (Press Enter to exit the command)
```

4. To change the linestyle to Dashed, follow these steps:
   a. Select the four lines.
   b. In the Properties toolbar (or ribbon panel), click the **LineStyles** droplist, and then choose DASHED.

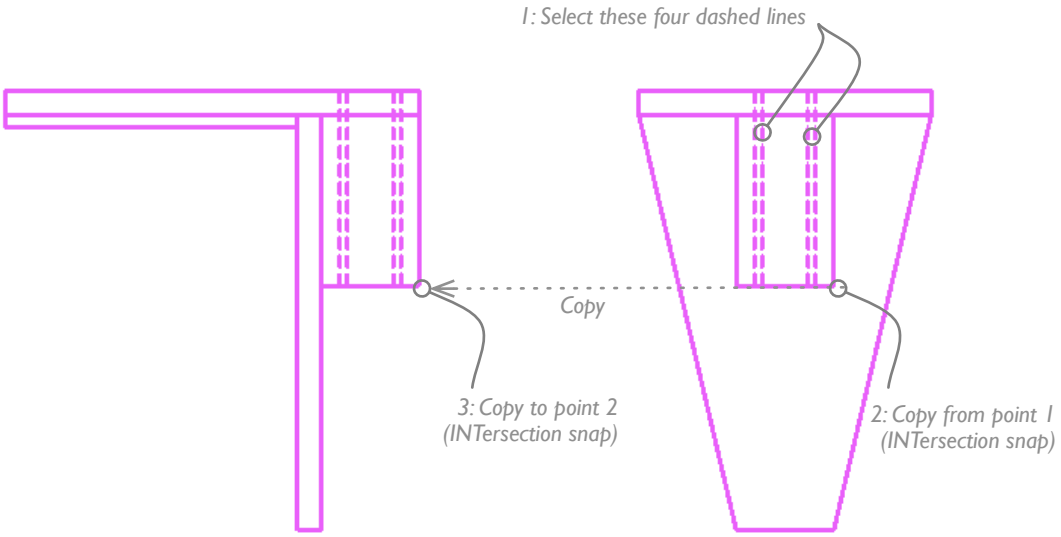


Choosing Dashed linestyle from the toolbar

Notice that the lines change from solid to dashed, as illustrated above.

5. Use the **Copy** command to copy the four dashed lines from the front view to the side view. *Hint*: Select them with **Window** selection mode, and then move them with **Intersection** entity snap.

```
: copy
Specify entities» w
Specify first corner» (Pick a point)
Specify opposite corner» (Pick another point to encompass the four lines)
Specify entities» (Press Enter to continue)

Specify from point» int
of (Pick point #1)
Specify second point» int
of (Pick point #2)
Specify second point» (Press Enter to end the command)
```



Copying lines

6. If the dashed lines are too close or too far apart, use **LtScale** to change this:

```
: ltscale
Default: 1.0000
Enter new value for LTSCALE» 0.5
```

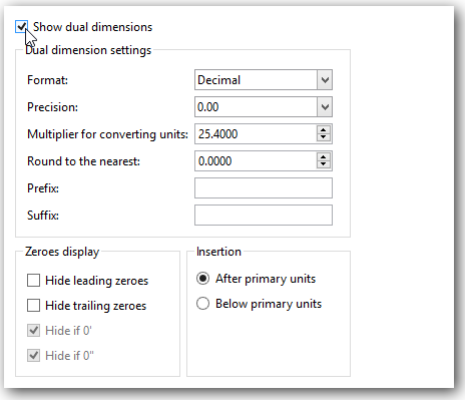
Values over 1.0 make linestyles larger, while those between 0.0 and 1.0 make them smaller.

7. Save the drawing.

Dual Dimensioning

The final step is to dimension the drawing. The hand drawn one was drafted in 1966 using inches. You redrew it in inches as well; most countries, however, use metric units. Through the **DimensionStyle** command, the program can show dimensions in both Imperial and metric units. Here's how:

1. Enter the **DimensionStyle** command.
2. In the Options dialog box, open the **Dual Dimension** node.



Turning on the Dual Dimension option

3. Turn on **Show Dual Dimensions**.

   The other settings are already appropriate for adding metric measurements to the dimensions through the value of 25.4 used for the **Multiplier for Converting Units**.

4. Click **OK**.
5. It's a good idea to place dimensions on a layer of their own. Create a new layer with these parameters:

Layer Property	Value
Name	**Dims**
LineColor	**8 (gray)**
LineWeight	**0.00mm**

   Make the Dims layer active.

6. Start the **LinearDimension** command.

```
: lineardimension
```

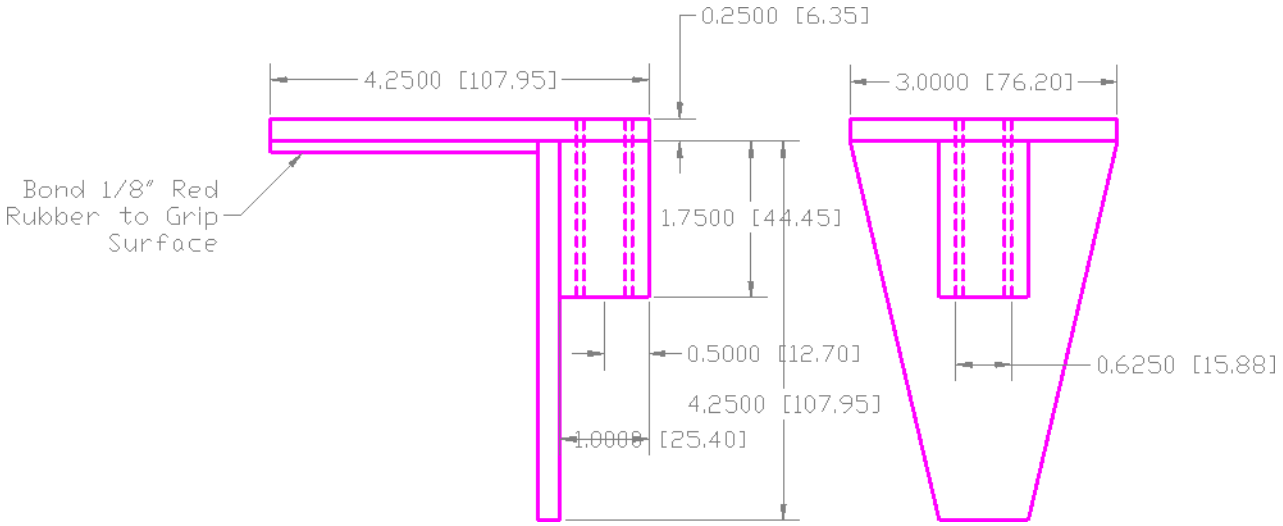
   a. Pick two points to locate the extension lines:

```
Specify first extension line position» (Pick one end of a line)
Specify second extension line position» (Pick other end of the line)
```

   b. ...and then locate the dimension line.

```
Specify dimension line position» (Pick a point)
Dimension Text : 3.0000
```

7. Apply other dimensions. See the figure on the next page as the example.
8. Use the **Leader** command to place notes.



Drawing complete with Imperial-metric dimensions

9. Save the drawing.
10. If you wish, print the completed drawing.

TIPS The **MakeLineStyle** command defines new line styles. This command can be found in the XtraTools menu and ribbon tab.

Just as Quick Input lets you enter commands, options, and coordinates at the cursor, the Heads-up Display lets you modify entities after you selected them.

It provides the following functions (from left to right) zoom to the specified entity(ies); change the entity's line style and width interactively; change the entity's layer; moves the entity to the active layer; moves the entity to the layer associated with another entity; dimensions the entity using Smart Dimensions; turns the entity(ies) into a block.
You turn this option on and off through the **Options** command: choose **User Preferences > Drafting Options > Heads-Up Display**.

Summary

In this chapter you learned how to bring old paper drawings into the program, and then trace over them accurately to create a precise CAD drawing.

In the following chapters, you build on this drawing, first by creating a 3D-like isometric version, and then making a true 3D model.

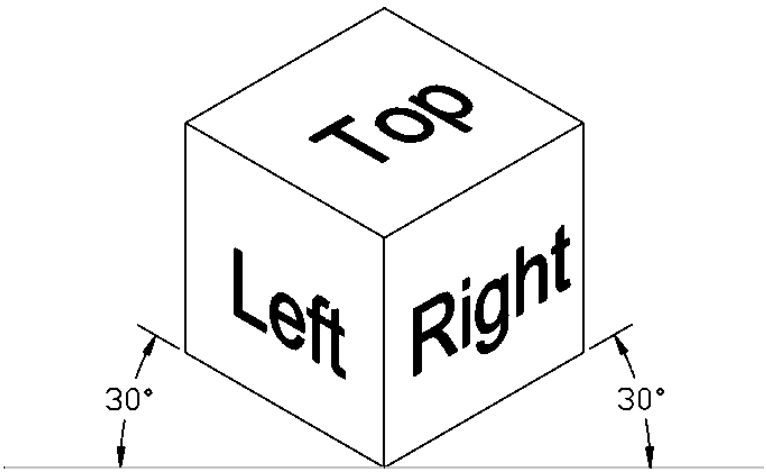
CHAPTER 15

Isometric Drafting

In This Chapter

- Customizing template files.
- Preparing the program for isometric drawings.
- Defining text and dimension styles.
- Drawing in isometric mode.

Isometric drawings look like 3D drawings, but are faked. They clearly illustrate three sides of 3D models — the left, top, and right — but are drawn using 2D drafting techniques. For instance, the cube below is drawn in an isometric view. I have labeled the three sides that are usually shown in these kinds of drawings. To achieve the "3D" look, the sides are slanted by 30 degrees.



Isometric drawings draw the left and right sides at 30 degrees

Isometric drawings are most commonly used for piping schematics, but can be used to represent architectural and mechanical designs.

KEY TERMS IN THIS CHAPTER

Isometric drawings refers to the 2D drawings that show the left, front (right), and top view simultaneously.

Isometric planes refers to the three drawing planes of isometric drawings, also known as "isoplanes."

Oblique angle refers to the angle by which text is slanted forwards or backwards.

Rotate angle refers to the angle by which text is rotated.

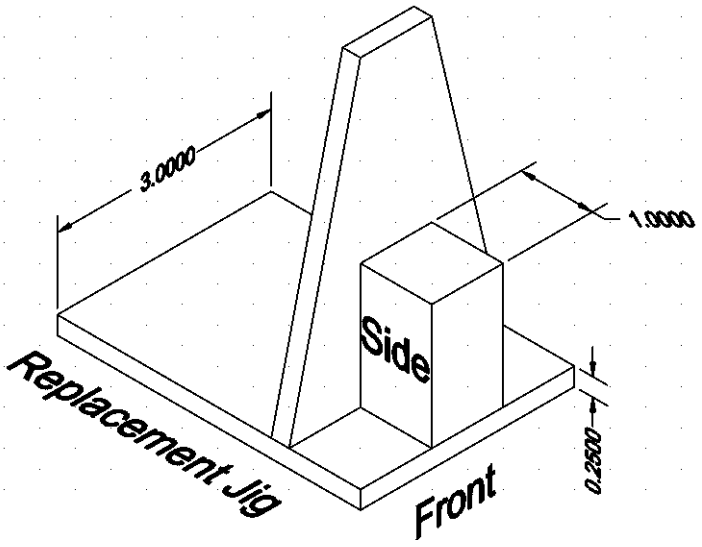
ABBREVIATION

F5 Shortcut keystroke for switching between isometric planes.

COMMAND

Command	Shortcut	Menu Selection
EditDimension	...	Dimension | Oblique

CHAPTER DRAWING:



Lines and text are drawn in three isometric planes at 30-degree angles. The curious thing is that for text and dimensions to look correct, they must also be slanted by 30 degrees! In this chapter, you learn how to create isometric text and dimension styles to help you place them quickly.

Isometric planes are often called "isoplanes" for short. The program has an isometric mode to help with these kinds of drawings, but "hides" it inside the Snap command. When isometric mode is turned on, the program displays the cursor's arm isometrically, and changes the display of the grid to match the current isometric plane. It also adjusts snap and ortho modes to work correctly at 30-degree angles.

There are no other commands that adapt to isometric mode. For instance, there is no built-in way to draw text and dimensions at isometric slants, and so the first tutorial sets up a DWT template file that contains all the settings needed for drawing isometrically.

(For what it is worth, isometric drawings have nothing to do with ISO template drawings and other similarly-named files, except that both are based on the Greek word for "same": while "I.S.O." is short for the International Organization of Standards, "isometric" means same-sided.)

Setting Up an Isometric Template

In this tutorial, you set up a template drawing that can be reused for isometric drawings you create in the future.

1. Start a new drawing with the *standard.dwt* template drawing.
2. To turn on isometric drafting mode, follow these steps:
   a. Start the **Snap** command and then enter the **Grid style** option:

```
: snap
Default: 0.5000
Options: Grid style, OFf, ON, ORientation, SPacing or
Specify horizontal and vertical spacing» g
```

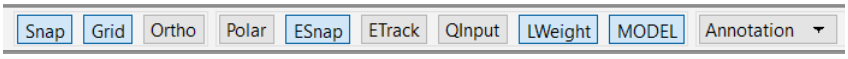
   b. At the prompt, enter 'i' to specify isometric mode:

```
Default: Rectangular
Options: Isometric, or Rectangular
Specify option» i
```

   c. Change the spacing to **0.25** units:

```
Default: 0.5000
Specify isometric snap spacing» .25
```

3. On the status bar, click the following buttons to turn on the related modes: **Snap**, **Grid**, **ESnap**, and **LWeight**.

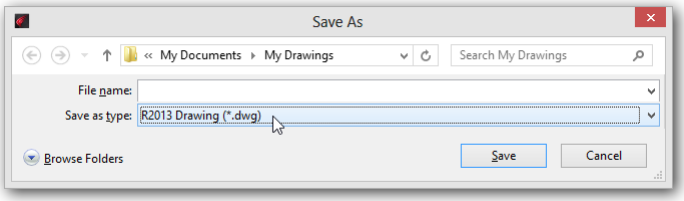


Settings on the status bar for this tutorial

4. The UCS icon is not useful for isometric drafting, so turn it off with the **UcsIcon** system variable. Set the value to 0:

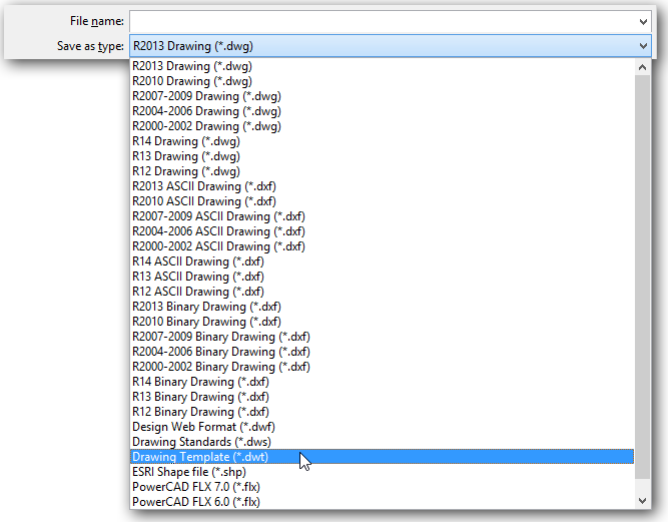
```
: ucsicon
Default: 3
Enter new value for UCSICON» 0
```

5. Save the drawing as a template file, as follows:
   a. Enter the **SaveAs** command.
   b. Look for the **Save As Type** droplist. It is near the bottom of the dialog box.



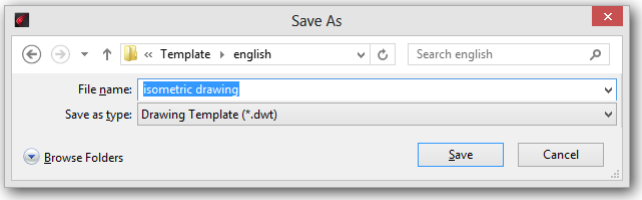
Saving the drawing as a template

   c. Click the droplist, and then choose "Drawing Template (*.dwt)." You'll find it near the end of the list.



Choosing the .dwg template file type

   d. Notice that the program automatically changes the folder to the one in which it stores its template files. Change the **File Name** to "Isometric Drawings."

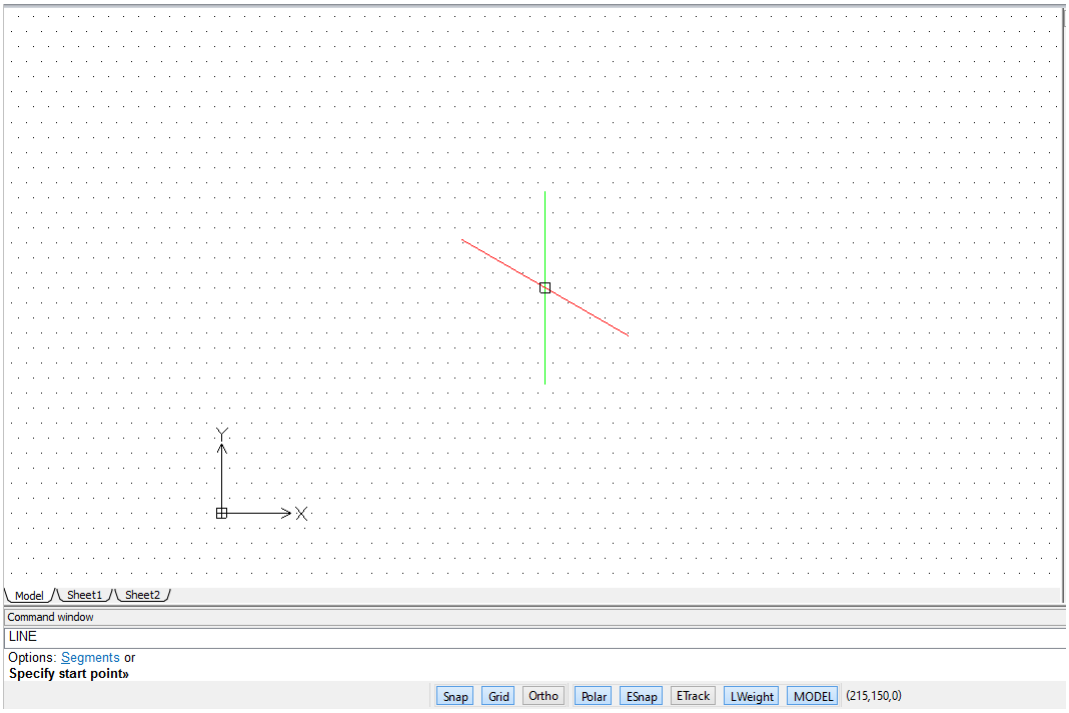


Naming the template drawing

   e. Click **Save**.

At this point, the drawing should look similar to the one illustrated below:

- Isometric grid is visible
- UCS icon is not visible
- Snap, Grid, ESnap and LWeight buttons are turned on at the status bar

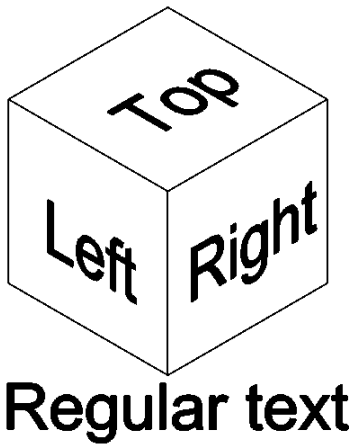


The program is ready to being isometric drafting

In the next parts of this tutorial, you create the text and dimension styles needed for isometric drawing.

ISOMETRIC TEXT STYLES

The figure below illustrates the problem when text is placed normally in an isometric drawing, it looks wrong. The solution is to slant text by 30 degrees, both obliquely and rotated, as shown by the words Top, Left, and Right. But each isoplane needs a different combination of obliquing angle and rotation.



Text on the three isometric planes must be slanted to look correct

Each isometric plane requires a different set of 30-degree angles — some positive, some negative — as shown by the table below:

Isoplane Name	TextStyle Command's Obliquing Angle	SimpleNote Command's Rotation Angle
Left	–30°	–30°
Top	30°	–30°
Right	30°	30°

The *obliquing* angle is set by the **TextStyle** command. Two styles are needed, one for the -30° of the left isoplane, and another that handles the +30° in both the top and right isoplanes. After you create the styles, they are saved to the template file.

The *rotation* angle is set by the **SimpleNote** command, -30° and +30° again. These angles cannot, however, be saved in a template drawing, and so you have to apply them as you place text in isometric drawings.

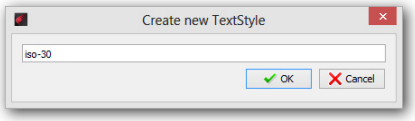
Let's see how to create the text styles.

1. Continue with the *Isometric drawing.dwt* template drawing to create the text styles for isometric text. The parameters you need to enter as as follows:

Text Style Name	Oblique Angle	Used for...
iso-30	-30	Left isoplane
iso+30	30	Top isoplane and right isoplane

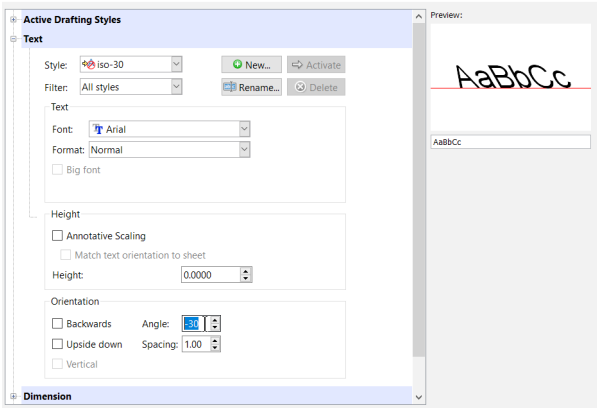
2. Enter the **TextStyle** command. Notice that the Options dialog box opens to the Drafting Styles | Text section. First, create the text style for the left isoplane (-30 degrees), as follows:

   a. To create a new text style, click the **New** button. In the Create New Text Style dialog box, enter **Iso-30**, and then click **OK**.



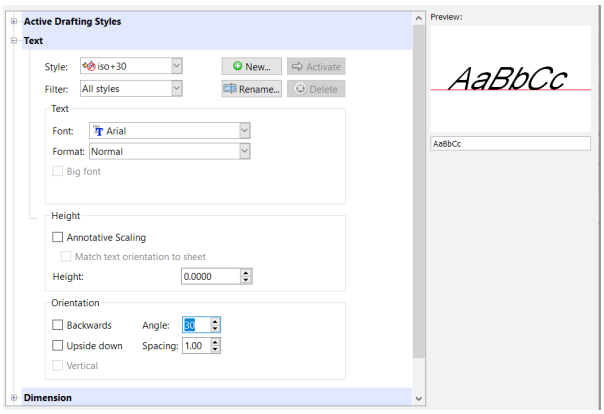
Starting the isometric text style

   b. In the **Orientation** section of the Options dialog box, change **Angle** to **-30**. Notice that the Preview thumbnail updates to show the obliqued text.



Settings for the first isometric text style

3. Create the second isometric text style, using these steps (and shown by the figure below):
   a. First, click **New** to create another new text style, and then name it "iso+30".
   b. Click **OK** to exit the Create New TextStyle dialog box.
   c. In the Options dialog box, change **Angle** to **30**.

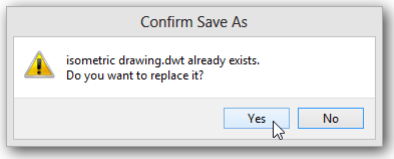


Settings for the second isometric text style

Style Property	Value	Comment
Name	iso+30	*(Click New to create the new style)*
Angle	30	*(Continue using Arial as the font)*

4. Click **OK** to exit the dialog box.
5. The drawing looks no different, but now contains the two new text styles. Save the drawing again as a template; *you cannot, however, use the Save command or press Ctrl+S*, because the program would save the file as a regular DWG file.

   Instead, save the drawing as a DWT template file: repeat the same steps of the previous tutorial:
   a. Enter the **SaveAs** command.
   b. From the **Save As Type** droplist, again choose "Drawing Template (*.dwt)."
   c. Click **Save** to save the template file.
   d. Select **Yes** when the program asks if you want to replace *isometric drawings.dwt*:



Replacing the template file with the changed drawing

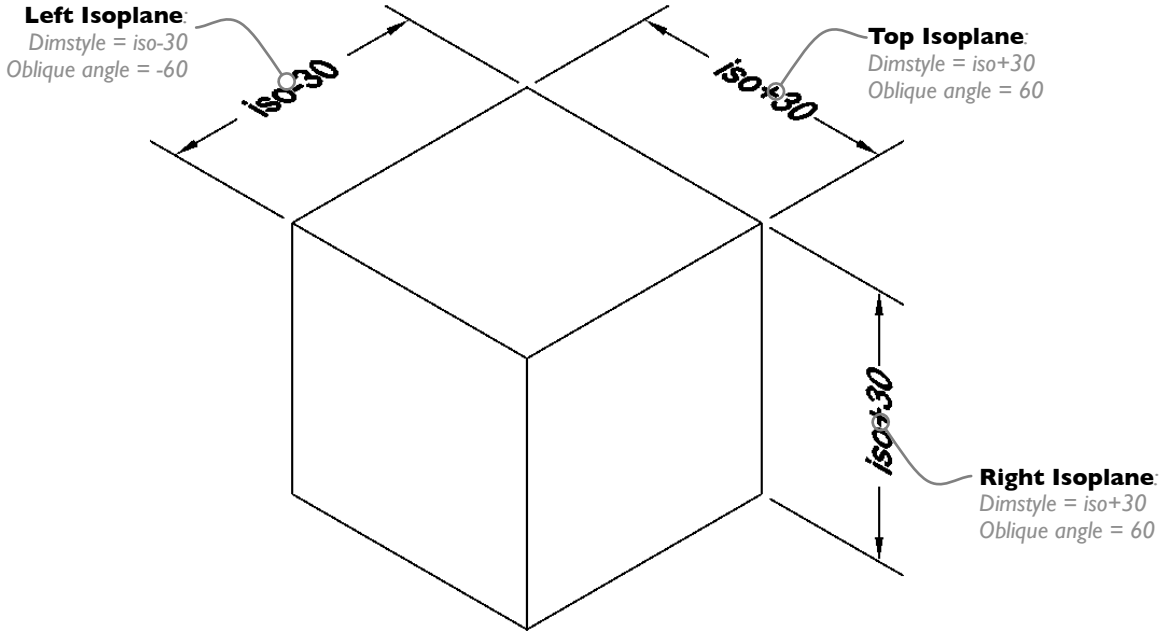
ISOMETRIC DIMENSION STYLES

With the isometric-friendly text styles in place, you can now go ahead and create the two isometric dimension styles with the TextStyle command. In these styles, the dimension text needs to be slanted like isometric text; in fact, the dimension styles make use of the two text styles you created in the previous tutorial.

To draw dimensions isometrically, you use the DimAligned command, because it automatically aligns linear dimensions with slanted entities at 30 degrees. Dimensions in the top isoplane need their extension lines slanted by 60 degrees. The slanting is done during the EditDimension command's Oblique Angle option.

Here are all of the parameters you will be using:

Isoplane	DimStyle Command Text Style	EditDimension Command Extension Lines Obliquing Angle
Left	iso–30	-60°
Top	iso+30	60°
Right	iso+30	60°



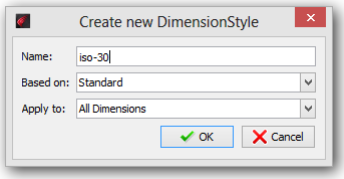
Parameters for dimension styles of isometric dimensions

Continue with the *Isometric drawing.dwt* template to create the two styles for isometric dimensions, as follows:

Dimension Style Name	Text Style Name	Used for...
iso-30	iso-30	Left isoplane
iso+30	iso+30	Top and right isoplanes

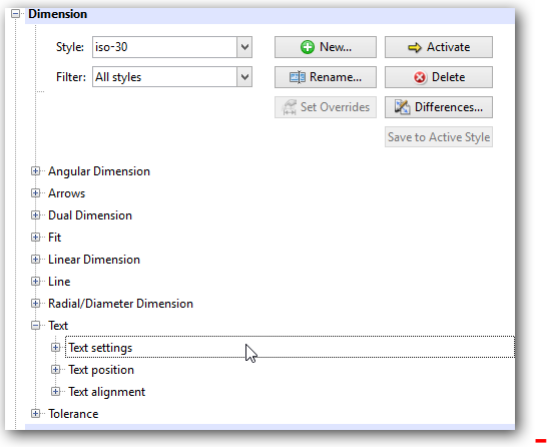
1. Start the **DimensionStyle** command.
2. Click **New** to create a new dimension style. It is the style for the left isoplane, -30 degrees.

3. Enter “iso-30” for **Name**, and then click **OK**.



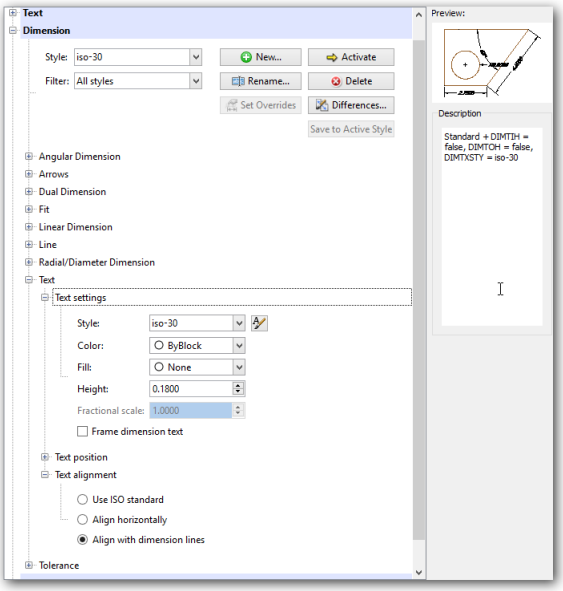
Creating the first isometric dimension style

4. Back in the Options dialog box, open the **Text** node, and then open the **Text Settings** node.



Settings for the first isometric dimensions style

5. From the **Style** droplist, select **iso-30**.
6. Open the **Text Alignment** node.
7. Turn on the **Align with dimension line** option. After the two changes are made, the dialog box should look like the figure below.



All of the settings required for an isometric dimension style

8. Repeat the process to create the second isometric dimension style named "iso+30." Use these settings:

Style Property	Value	Comment
Name	iso+30	*(Click New to create the new style.)*
Style	iso+30	

9. Click **OK** to exit the dialog box.
10. The template drawing is complete. Save it one more time with the **SaveAs** command. Remember to choose "Drawing Template (*.dwt)" from the **Save As Type** droplist.

The Isometric drawings.dwt template file now contains all settings needed for drafting in isometric mode.

Drawing in Isometric Mode

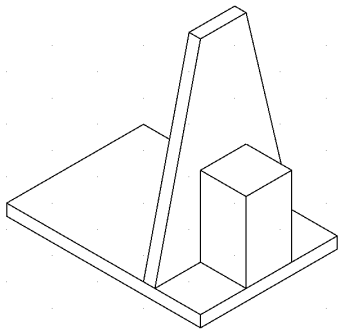
In this tutorial, you recreate the 2D drawing from the last chapter as a 3D-looking isometric drawing. (I assume that the program is still running from the last tutorial.)

1. Enter the **New** command.
2. In the Specify Template dialog box, choose *Isometric drawings.dwt.*
3. Click **Open**. Notice that the program creates a new drawing with the generic name *NONAME_1.dwg* (or similar).
4. With the **Save** command, immediately save the drawing by the name of *Isometric part.dwg*.

You are now ready to start drawing in isometric mode.

DRAWING ISOMETRIC BOXES

You will draw the part upside-down to show the details more clearly, as illustrated below.

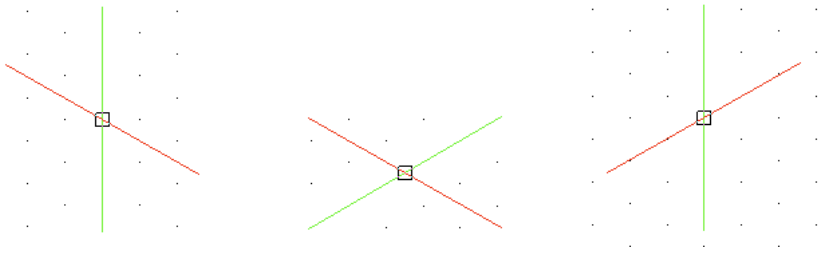


What the completed drawing will look like

1. Enter the **line** command:

```
: line
```

Notice the isometric cursor with its slanted cross hairs. It tells you which isometric drawing plane is current. There are three planes: left, top, and right (or front).

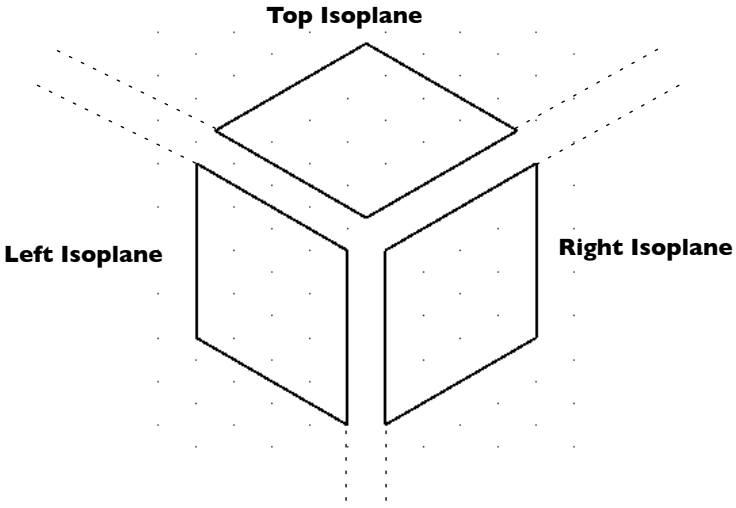


Left to right: *Cross hair cursors for left, top, and right isoplanes.*

You switch between isoplanes by pressing the **F5** function key. Each time you do, the program reports the name of the current plane:

```
F5 <IsometricPlane Top>

F5 <IsometricPlane Right>

F5 <IsometricPlane Left>
```

In each plane, you can draw in two directions only, as summarized by the figure below:



Visualizing how squares look in each isometric plane

This means you will be pressing **F5** frequently throughout the following tutorials!

2. Press **F5** until the program reports the top isoplane:

```
F5 <IsometricPlane Top>
```

3. The first entity to draw is the top plate, which is 3" wide by 4.25" long. Continuing with the Line command, begin drawing at the center of the bottom portion of the drawing area. (The exact location is not important.)

```
Specify start point» (Pick a point in the lower center of the drawing)
```

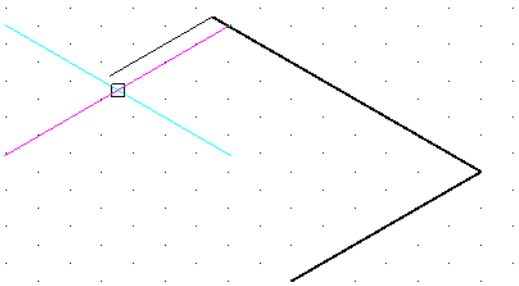
4. Move the cursor to the right. Notice that the program's ortho mode forces the line to travel in the isoplane, 30 degrees up from the x axis.

5. Enter the length, **3**, and then press **Enter**.

```
Specify next point» (Move cursor to the right, and then enter:) 3
```

6. Move the cursor up, enter the next length, **4.5**, and then press **Enter**:

```
Specify next point» (Move cursor up, and then enter:) 4.25
```



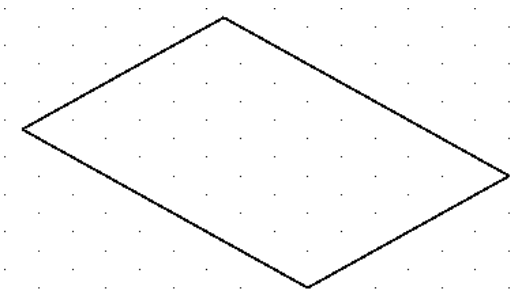
Starting to draw a rectangle in the top isoplane

7. Move cursor left, and then enter **3**:

```
Specify next point» 3
```

8. Close the "rectangle" by typing 'c':

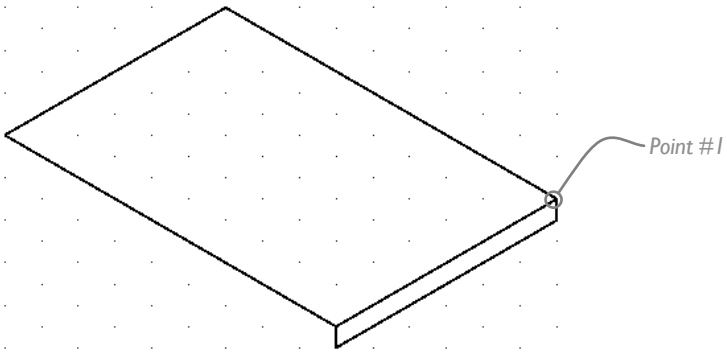
```
Specify next point» c
```



Completed rectangle in top isoplane

9. To draw the right side of the plate, first switch to the next isoplane, right:

```
F5 <IsometricPlane Right>
```



Adding a rectangle to the right isoplane

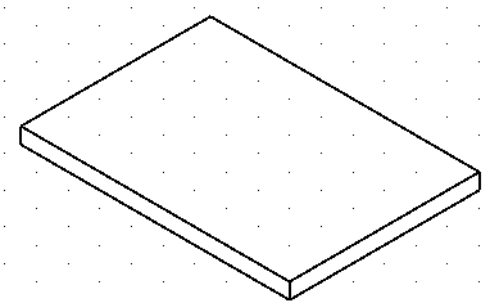
10. Press the spacebar to restart the Line command, and then draw the 0.25" thick sides on the right isoplane, as follows:

```
: (Press spacebar to repeat command)
LINE Specify start point» (Pick point #1)
Specify next point» 0.25
Specify next point» 3
Specify next point» 0.25
Specify next point» (Press Enter to end the command)
```

11. To draw the left side of the plate, switch to the left isoplane:

```
F5 <IsometricPlane Left>
```

12. Press the spacebar to restart the Line command, and then draw the 0.25" thick sides on the left isoplane. You now have a rectangular plate that should look like the one in the figure below.



Adding a rectangle to the left isoplane

13. Press **Ctrl+s** to save your work (**Cmd+s** on Macs).

DRAWING IRREGULAR BOXES

Drawing rectangles and squares isometrically is fairly straightforward, as the tutorial above showed. The difficulty comes with drawing irregularly-shaped objects; the workaround is to draw a rectangular outline, and then trim it.

In this tutorial, you draw the trapezoid-shape. It is 3" wide by 4.25 tall.

1. The trapezoid is located one inch from the front edge of the base plate. You will use the **From** entity snap to begin the line:

```
: line
```

Press **F5** until the drawing is in the right isoplane.

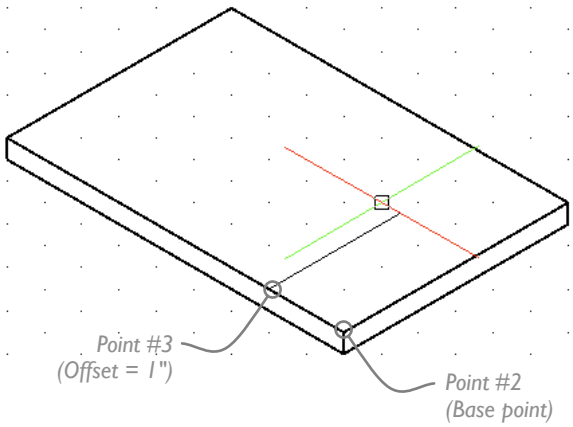
```
Specify next point» F5 <IsometricPlane Right>
```

2. Enter **From** to indicate the offset distance, and then pick point #2:

```
Specify start point» from
Base point (Pick point #2)
```

3. Move the cursor to the left-top, and then specify the offset distance, 1:

```
(Offset) (Move cursor towards point #3, and then enter:) 1
```

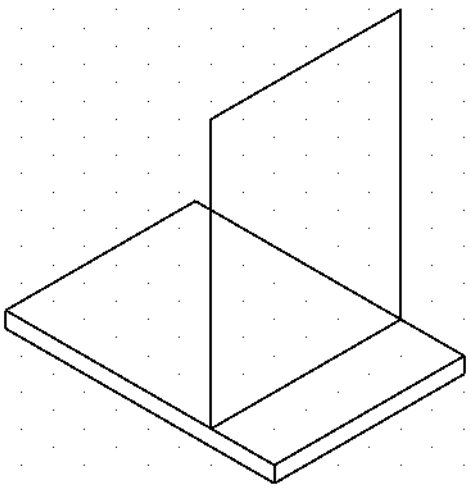


Determining the start point for the next rectangle

4. You can now draw the rest of the rectangle as per the earlier tutorial.

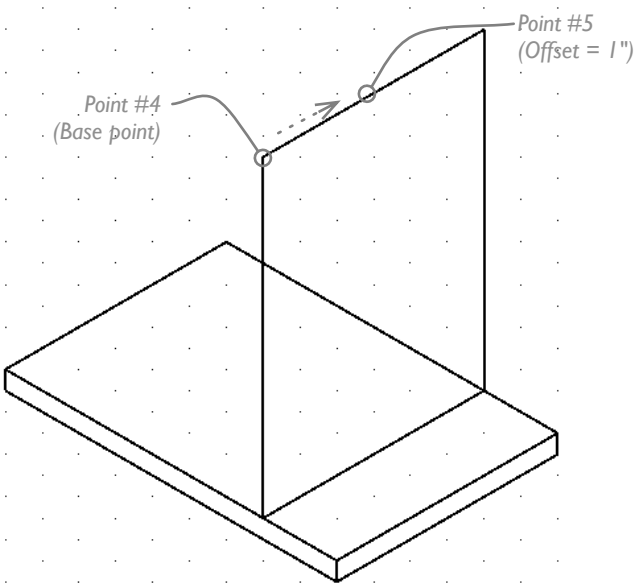
```
Specify next point» 3
Specify next point» 4.25
Specify next point» 3
Specify next point» (Press Enter)
```

The result should look like the figure below.



Completed rectangle

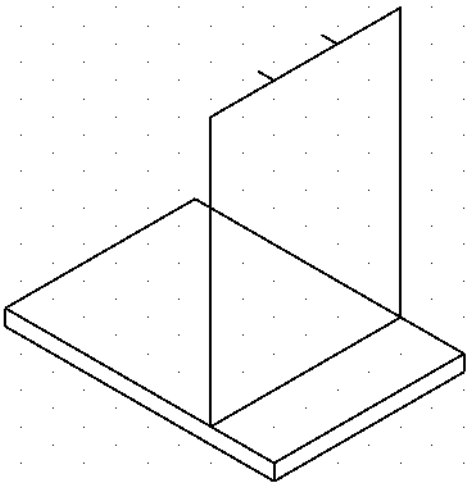
5. Now you need to trim the rectangle to represent the trapezoid. The top is 1" wide. Use **From** again to position the next line 0.25" long in the top isoplane:



Determining the start points of the trapezoid

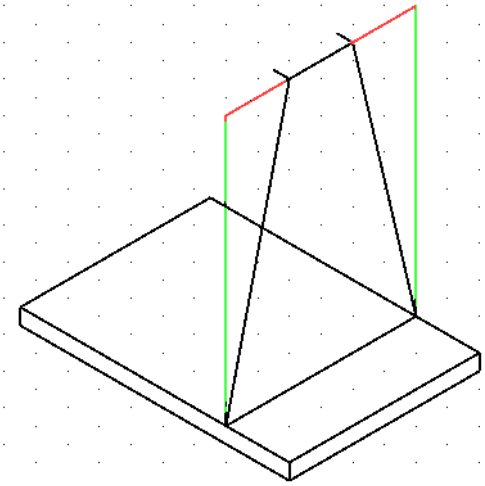
```
: (Press Enter to repeat the Line command)
LINE F5 <IsometricPlane Top>

Specify start point» from
Base point (Pick point #4)
(Offset) (Move cursor to point #5) 1

Specify next point» .25
Specify next point» (Press Enter)
```

6. Repeat to draw the other 0.25"-long line.



Top of the trapezoid

7. With the two stubby lines in place, you can add lines to form the trapezoid, as follows:
   a. Turn off ortho mode by clicking the **Ortho** button on the status bar.
   b. With the **Line** command, draw the two diagonal lines shown in the figure below.



Adding diagonal lines in isometric mode

8. Now you trim and erase some of the lines to complete the trapezoid:
   a. Start the **Trim** command.

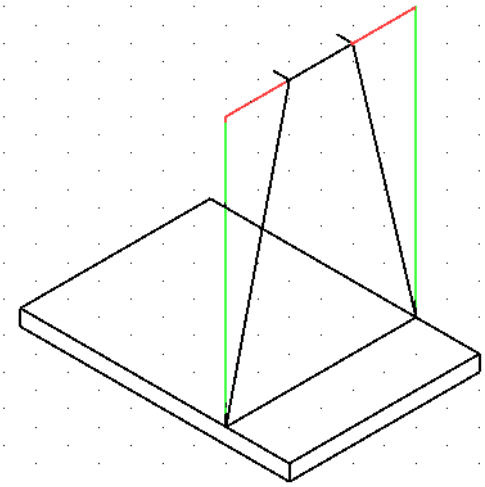
```
: trim
```

   b. Press **Enter** to select all entities as cutting edges.

```
Specify cutting edges ...
Options: Enter to specify all entities or
Specify entities» (Press Enter to select all entities in the drawing)
```

   c. Remove the line segments shown in red by the figure below:

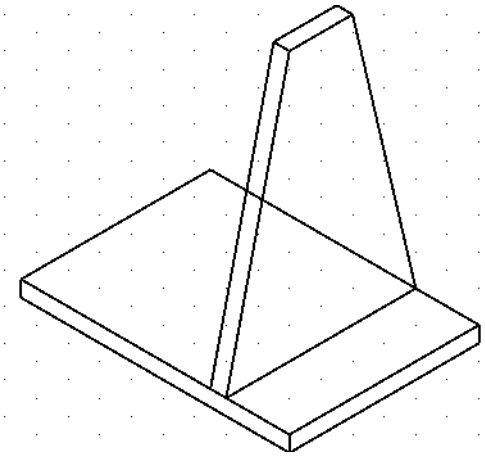
```
Specify segments to remove» (Select one segment shown in red)
Specify segments to remove» (Select the other red segment)
Specify segments to remove» (Press Enter to end the command)
```



Lines to be trimmed to create the isometric trapezoid

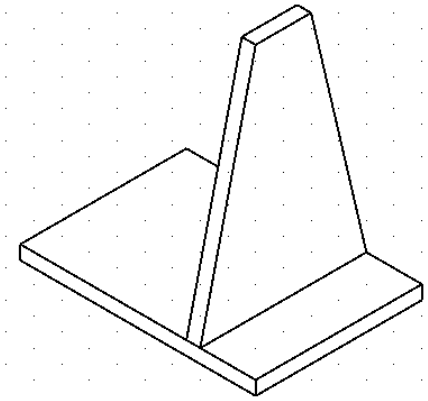
   d. Use the **Erase** command to remove the segments shown in green. (These cannot be removed by the Trim command.)

9. Draw the remainder of the trapezoid with the **Line** command.



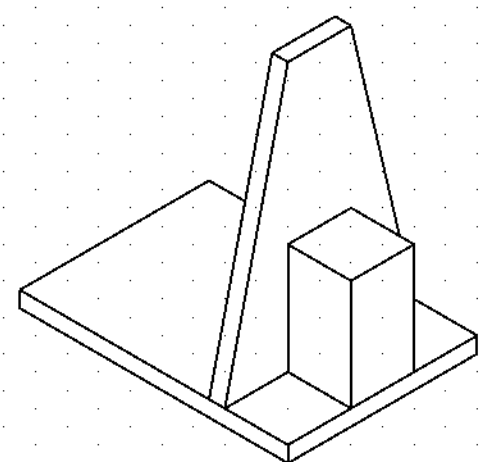
Completing the trapezoid

10. Use the **Trim** command to remove the hidden lines.



Hidden lines removed manually

11. To complete the drawing, use the techniques you learned above to draw the one remaining box. It has a 1"-square base and is 1.75" tall. The result should look like this:



Completed isometric drawing

12. Press **Ctrl+s** to save your work to disc (**Cmd+s** on Macs).

Placing Isometric Text

Adding text to isometric drawings takes these two steps:

i. Select an isoplane in which to work.

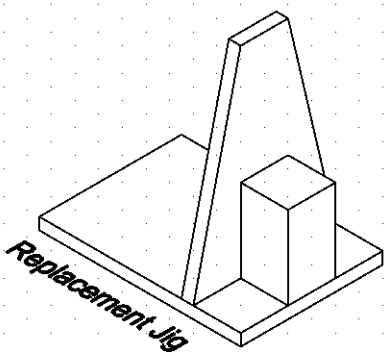
ii. Start the **-SimpleNote** or **Note** command, and then specify the parameters appropriate to the isoplane:

» Style name
» Rotation angle

Use the following table to remind you which style names and rotation angles to use for each isoplane:

Isoplane	Text Style Name	Rotation Angle
Left	iso-30	–30°
Top	iso+30	–30°
Right	iso+30	30°

In the following tutorial, you add text to the top isoplane. The result should look similar to the figure illustrated below.



Text drawn using isometric text style

1. Press **F5** until you reach the top isoplane:

```
: F5 <IsometricPlane Top>
```

2. Start the **-SimpleNote** command, and then enter 'st' to specify the text style:

```
: -simplenote
Options: Justify, textSTyle or
Specify start position» st
```

3. Specify the **iso+30** style name.

```
Default: Standard
Options: ? to list or
Specify name» iso+30
```

4. Pick a point in the drawing to start the text, and then specify a suitable height, such as 0.35":

```
Specify start position» (Pick a point at which to start the text)
Specify height» 0.35
```

5. From the table above, you know that the angle needs to be -30 degrees:

```
Specify text angle» -30
```

6. Finally, enter some text, and then press **Enter** twice.

```
Specify text» Replacement Jig
Specify text» (Press Enter to end the command)
```

PLACING ISOMETRIC DIMENSIONS

Adding dimensions to isometric drawings is much more complex than adding text, unfortunately. It takes these steps:

i. Select an isoplane in which to work.

ii. Set the appropriate dimension style.

iii. Use the **ParallelDimension** command for linear dimensions.

iv. After all dimensions are placed, use the **Dimension | Oblique** menu item to adjust the angle of extension lines.

Use the following table to remind you which dimension styles to use for each isoplane:

Isoplane	Use this DimStyle	Apply this obliquing angle with EditDimension
Left	iso–30	-60°
Top	iso+30	60°
Right	iso+30	60°

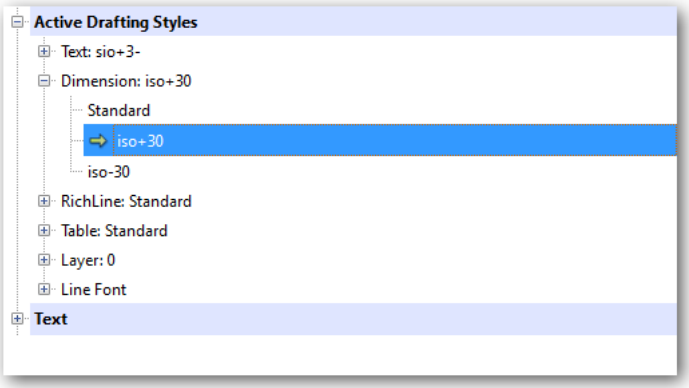
In this tutorial, you place a dimension in the top isoplane.

1. Before starting the DimAligned command for placing isometric dimensions, you need to set the dimension style name. This is done through the Styles command:

   a. Enter the **Styles** command.

   b. Open the **Dimension** node.

   c. Double-click **iso+30** to make the active dimension style.



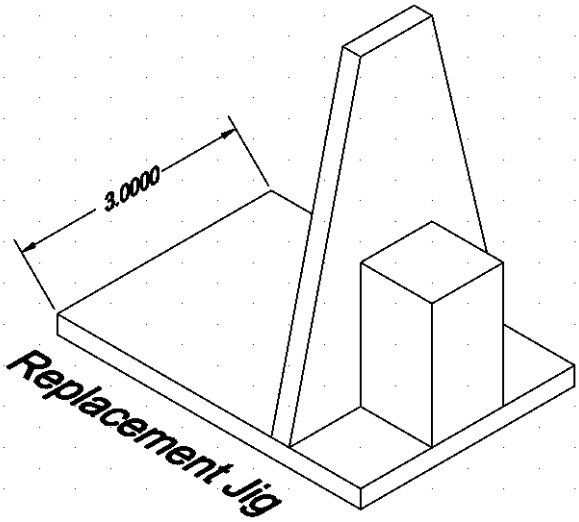
Activating the "iso+30" dimension style

   d. Click **OK**.

2. Ensure the active isoplane is set to top.

3. Start the **ParallelDimension** command:

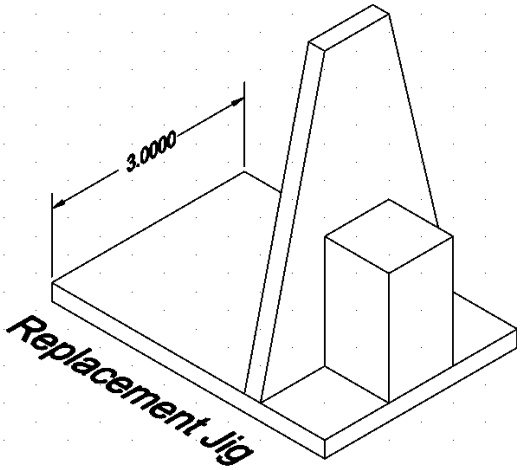
```
: paralleldimension
Default: Entity
Options: Entity or
Specify first extension line position» e
Specify entity» (Pick a line to dimension)

Options: Angle, Note, Text or
Specify dimension line position» (Position the dimension line)
Dimension Text : 3.0000
```



Initial placement of the first dimension

4. After all dimensions are placed, use the **Dimension | Oblique** menu to adjust the angle of extension lines. (You could enter the EditDimension command and its Oblique Angle option, but the menu pick takes you directly to the obliquing option, which is more convenient.)

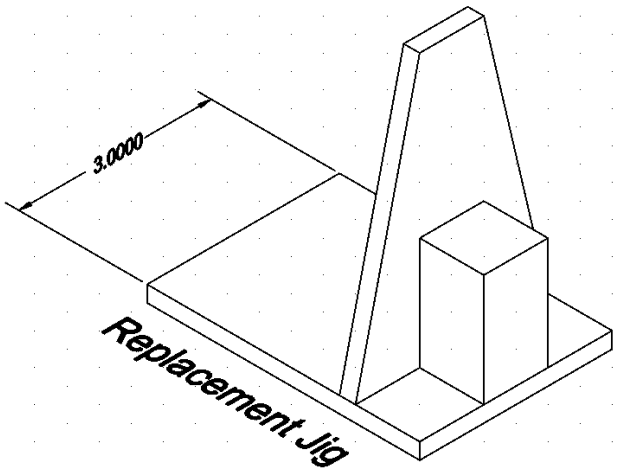
   For this dimension, adjust the extension lines by 60 degrees, as follows:



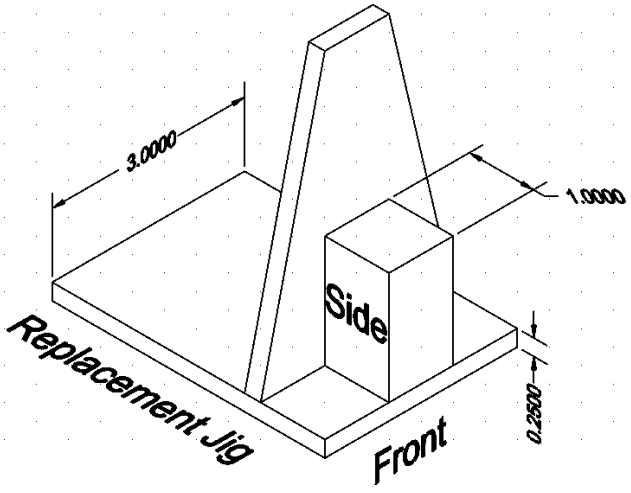
Dimension correctly angled for the top isometric plane

```
: _EDITDIMENSION
Default: Home
Options: Angle, Home, Move, New, Oblique or
Specify option» _O
Options: Enter to exit or
Specify oblique angle» -60

Specify entities» (Choose the dimension)
1 found, 1 total
Specify entities» (Press Enter to end the command)
```

TIP You get a different — but equally as valid — result when you enter -60 degrees, as illustrated below.



5. If you wish, add more isometric text and dimensions to the drawing.



Remainder of text and dimensions placed on drawing

6. Press **Ctrl+s** to save your work, and then exit the program (**Cmd+s** on Macs).

Summary

In this chapter you learned how to simulate the look of a three-dimensional drawing through the use of isometric drafting. You learned how to oblique the angles of text and dimensions to make them look "correct." In the next chapter, you turn this design into a real 3D drawing.

CHAPTER 16

Solid Modeling

In This Chapter

- Understanding solid modeling
- Drawing in 3D with solid primitives
- Applying Boolean operations
- Analyzing solid models

Solid modeling is sometimes considered the ultimate in CAD design, for it creates the most precise 3D models possible, ones that can be analyzed as if they were physical prototypes. The program provides two primary approaches to creating 3D solid models:

» Draw 3D primitives, and then combine and edit them into complex shapes.
» Draw 2D entities, and then revolve and extrude them into 3D models.

This chapter works with the first approach, using 3D primitives such as boxes, cylinders, and wedges. The program's capabilities in solid modeling include the following functions:

Draw solid primitives as a box, wedge, cone, sphere, cylinder, torus, pyramid, or planar surface.

Make solid construction with extrudes, sweeps, revolves, and lofts.

Perform Boolean operations that union (join together), subtract (remove one from the other), and intersect (common volumes) solids.

Edit solid models to fillet, chamfer, copy, and color edges; move, copy, and offset faces; shell, imprint, and separate solids; and more.

In this chapter, you work with some of these functions. For a head start, you may want to watch the video at https://www.graebert.com/blog/videos/how-to-use-3d-solid-editing-with-grips-and-dynamic-ccs-in-ares-commander-2018/

KEY TERMS IN THIS CHAPTER

Arcball refers to the visual aid for rotating 3D viewpoints with the RollView command.

Boolean refers to the editing operations that join or remove 3D solid entities.

Cube refers to a 3D solid box with equally sized sides.

Extrusion refers to the closed 2D entity thickened into a 3D solid.

Primitives refers to the basic 3D solid entities, such as box, sphere, and cylinder.

Solid models refers to the 3D entities with solid interiors.

Subtract removes one or more 3D solid entities from one or more others.

Swivel moves 3D viewpoint left and right.

Taper angle specifies the angle at which an extrusion thickens.

Tilt moves 3D viewpoint up and down.

Union joins two or more 3D solid entities into a single entity.

COMMANDS

Command	Shortcuts	Menu Selection
Box	...	Solids | Draw | Box
Cylinder	cy	Solids | Draw | Cylinder
Extrude	ext	Solids | Draw | Extrude
MassProp	...	Solids | Draw | Region/Mass Properties
PlanView	plan, pview	...[1]
Polyline3D	3dpoly, 3p	Draw | 3D Polyline
RollView	3do, orbit	View | Constrained Orbit
Subtract	su	Solids | Solids Editing | Subtract
Union	uni	Solids | Solids Editing | Union

[1] No menu selection.

Drawing 3D Solid Primitives

Command: Box
Alias: --
Menu bar: Solids | Draw | Box
Ribbon: Home | Modeling | Box (3D Modeling)
Toolbar: 

In the following tutorial, you create a 3D solid model from the 2D drawings you drafted in the previous chapters.

1. Start the program with a new drawing based on the *standard.dwt* template file.
2. Enter the **Box** command, and then notice that the program prompts you in the Command window.

   ```
   : box
   ```

3. Place the first corner of the box at the drawing's origin, as follows:

   ```
   Options: Center or First corner of box» 0,0,0
   ```

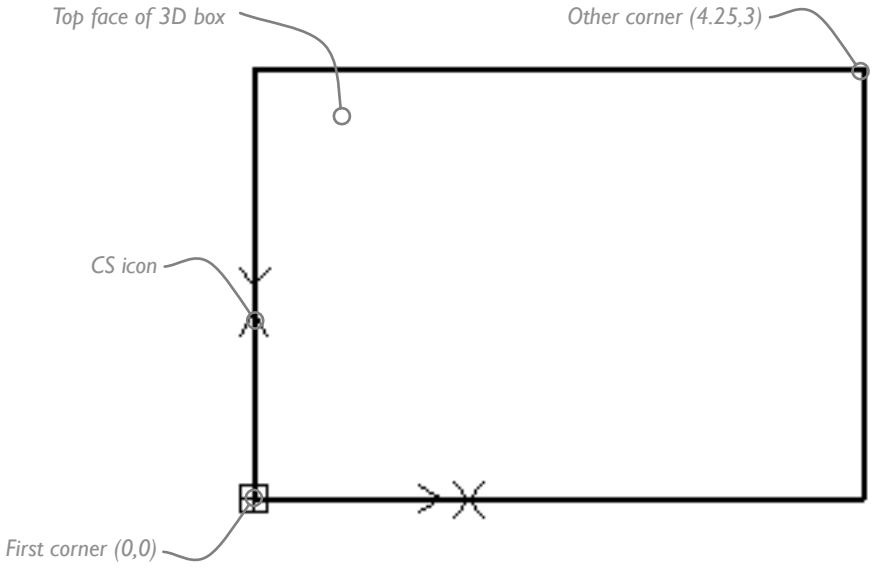
4. The box is 4.25" long by 3" wide, and so enter these dimensions as coordinates:

   ```
   Options: Cube, Length or Other corner» 4.25,3
   ```

5. ... and it is 0.25" high. Specify this height, as follows:

   ```
   Options: 2Point or Height» .25
   ```

   The figure below shows the top of the box, along with the CCS icon in the lower left corner. (The CCS icon is useful when working in 3D space.)



Elements of drawing a box in three dimensions

6. Use the **Save** command to save the drawing by the name "3d model.dwg."

VIEWING IN 3D

Command: RollView
Aliases: 3dorbit, 3do, orbit
Menu bar: View | Constrained Orbit
Ribbon: Home | Modify | 3D Rotate (3D Modeling)
Shortcut: Shift+*middle button*
Toolbar: --

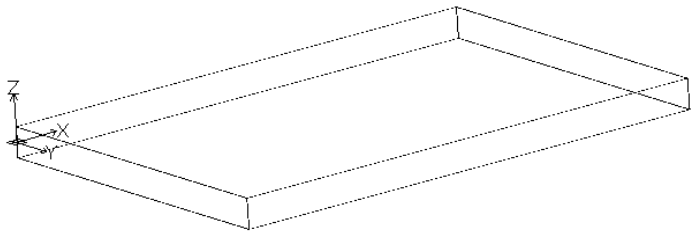
The box that you drew was drawn in 2D, just like all earlier drawings. To see it in 3D, rotate the viewpoint.

1. To view the box in three dimensions, enter the **RollView** command.

   ```
   : rollview
   ```

   Notice the rollview cursor ⊙.

2. Hold down the left mouse button, and then drag the cursor. As you do, the box moves from plan view to a 3D viewpoint. The viewpoint varies depending on how you move the mouse.



Box appearing in three dimensional space

SUMMARY OF BOX COMMAND OPTIONS

```
: box
Options: Center or
First corner of box» (Enter coordinates or pick a point)

Options: Cube, Length or
Other corner» (Enter coordinates or pick a point)

Options: 2Point or
Height» (Enter coordinates or pick a point)
```

Center specifies the location of the cube's center.

Cube creates a cube of equal sides; prompts you for the length to apply to all sides:

```
Length» (Enter a length or pick two points)
```

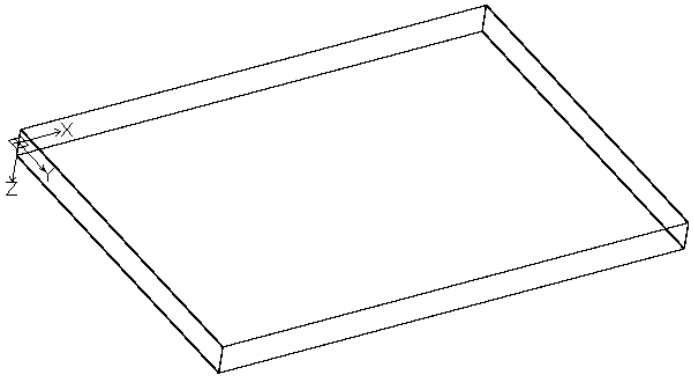
Length specifies the length and width of the base:

```
Length» (Enter a length or pick two points)
Width» (Enter another length or pick two points)
```

2Point specifies the height by picking two points anywhere in the drawing:

```
Specify first point» (Enter coordinates or pick a point)
Specify second point» (Enter coordinates or pick a point)
```

3. Keep moving the mouse until the box is flipped over. You know you are successful when the Z of the CCS icon faces downwards.



Box is flipped over when Z coordinate points downward

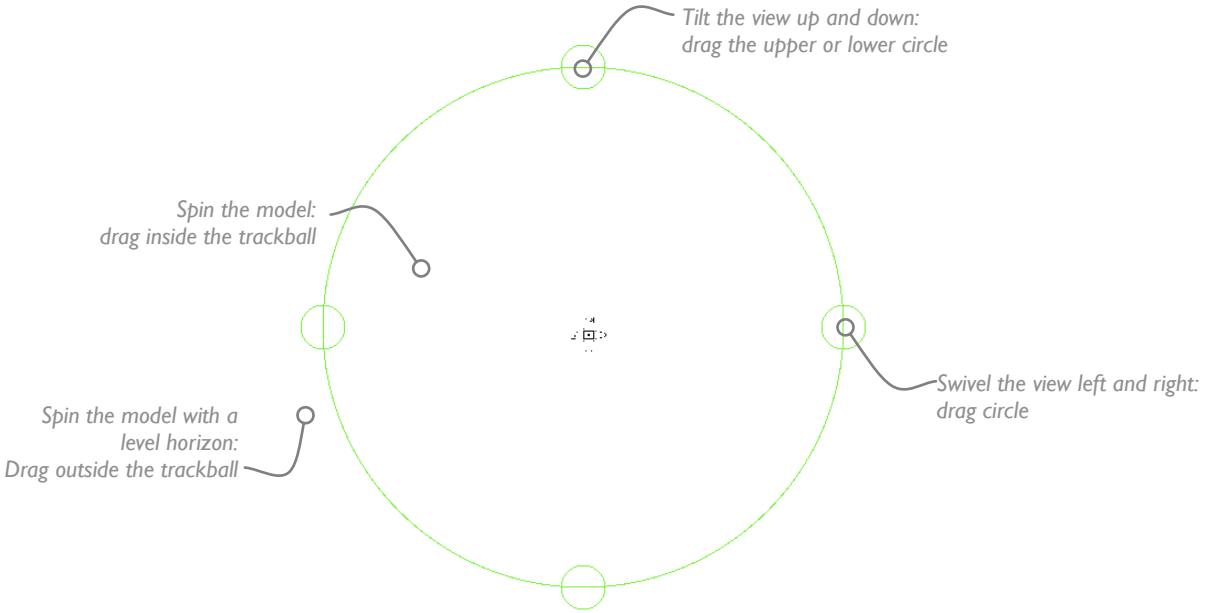
4. When you are satisfied with the 3D view, press **Esc** or **Enter** to exit the command.

```
Press ESC or ENTER to exit. (Press Enter.)
```

SUMMARY OF ROLLVIEW'S ARC BALL OPTIONS

```
: rollview
```

Hold down the **Shift** key to display the arcball:



Action	How
Spin model around the viewpoint...	...by dragging **inside** the arcball with left mouse button.
Rotate viewpoint with level horizon...	...by dragging **outside** the arcball.
Tilt viewpoint up and down...	...by dragging circle **at top or bottom** of the arcball.
Swivel viewpoint left and right...	...by dragging circle **at left or right** of the arcball.
Zoom in and out....	...by rolling the mouse's roller wheel forwards and back.

While the RollView command is active, you can change the viewpoint in a number of ways:

- **Rotate** the view in 3D — hold down the left mouse button, and then drag the mouse
- **Display** the arcball — hold down the Shift key
- **Zoom** in and out — roll the scroll wheel forwards and backwards
- **Pan** around the drawing — hold down the scroll wheel button, and then drag the mouse
- **Exit** the command — right click

TIP To return to the 2D viewpoint, enter the **Planview** command, and then press **Enter** twice, like this:

```
: planview (Press Enter)
Default: Active
Options: Active ccs, CCs, or World
Specify option» (Press Enter)
```

ADDING MORE SOLIDS

To continue modeling the 3D object, you draw three more solids using the following commands:

i. **Box** command creates the red rubber grip as a 3x3" square by 0.125" thick.

ii. **Extrude** command creates the trapezoid as 3" (top) x 1" (bottom) x 4.25" (tall) by 0.25" thick.

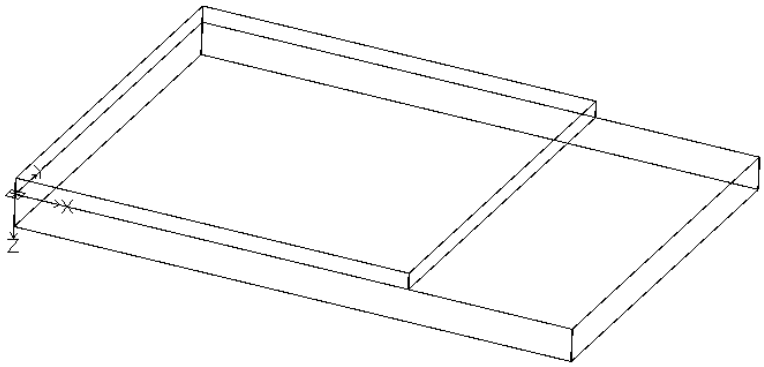
iii. **Box** command creates the square post as 1x1" square by 1.75" tall.

iv. The hole is "drilled" later, made from a 3D cylinder that is subtracted from the other parts.

Here is how to add the three solids to the drawing:

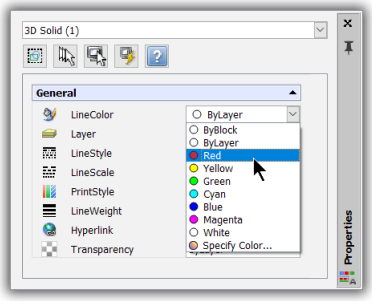
1. The red rubber grip is drawn just like the first box, except that the height is negative to draw it downwards — in the negative z-direction:

```
: box
Options: Center or First corner of box» 0,0
Options: Cube, Length or Other corner» 3.3
Options: 2Point or Height» -0.125
```



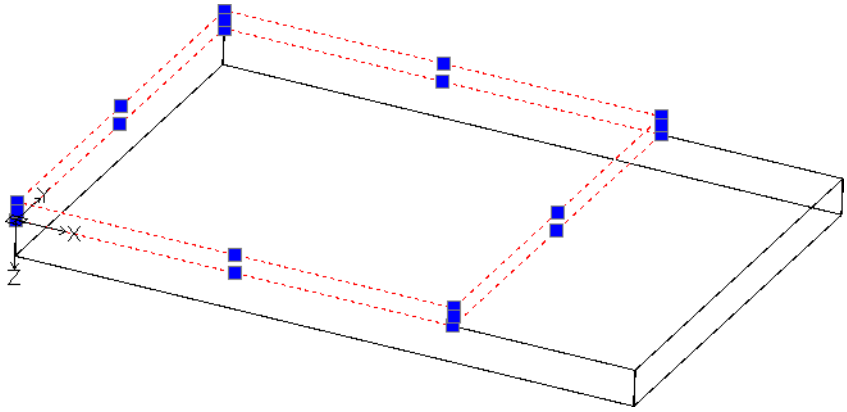
Second box laid on top of the first

2. Change the color of this new box to red, like this:
   a. Select the "rubber mat" box.
   b. From the Properties toolbar, click the **Color** droplist.
   c. Select **Red**.



Using the Properties palette to change the color of the box

Notice that the rectangular solid turns red.



Selected box turns red

   d. Press **Esc** to unselect the box.

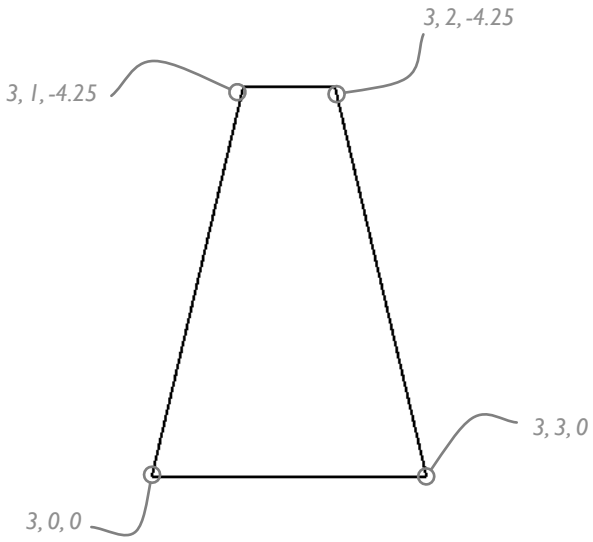
EXTRUDING 2D OBJECTS TO CREATE 3D SOLIDS

Command: Extrude
Alias: ext
Menu bar: Solids | Draw | Extrude
Ribbon: Home | Modeling | Extrude (3D Modeling)
Toolbar:

There are often several ways to solve a design problem in 3D modeling, and that is the situation you face with the trapezoid-shaped part. One approach is to draw a box, and then subtract the triangular portions.

Another is to draw a 2D outline of the shape, and then extrude (thicken) it into a 3D solid — which is the approach you take in this tutorial.

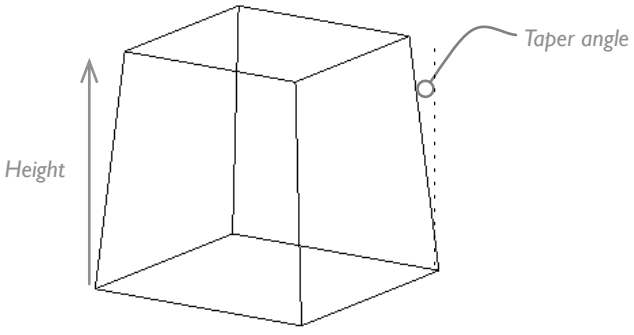
1. The corners of the trapezoid have the following coordinates:



Dimensions of the trapezoid

SUMMARY OF EXTRUDE COMMAND OPTIONS

```
: extrude
Active wire frame density: ISOLINES=4

Specify entities» (Pick one or more closed 2D entities)
Specify entities» (Press Enter to end entity selection)

Options: Path, Taper angle or
Specify extrusion height» (Enter a value, or pick two points)
```



Path selects an open 2D entity that defines the extrusion path:

```
Specify extrusion path» (Pick an entity)
```

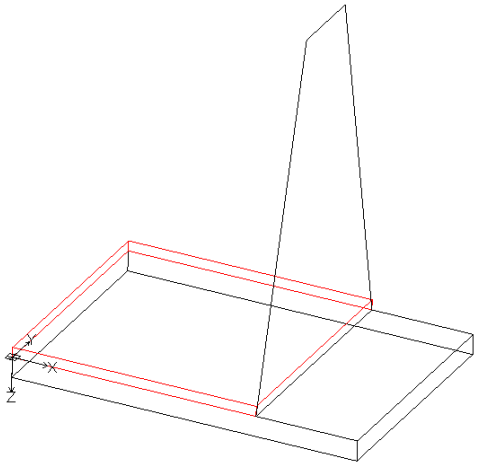
Taper angle specifies the angle at which the extrusion narrows or expands.

```
Specify angle of taper for extrusion» (Enter an angle)
```

Extrusion height specifies the extrusion distance.

2. Use the **Polyline3D** commands to draw the outline. (You cannot use the regular Polyline command, because it works with 2D coordinates only.)

```
: polyline3d
Specify start point» 3,0,0

Options: Undo, Enter to go back or Specify vertex position» 3,3,0
Options: Undo, Enter to exit or Specify next vertex position» 3,2,-4.25
Options: Close, Undo, Enter to exit or Specify next vertex position» 3,1,-4.25
Options: Close, Undo, Enter to exit or Specify next vertex position» c
```



Trapezoid drawn vertically

3. "Fatten" the trapezoid by applying the **Extrude** command, as follows:

   a. Start the **Extrude** command by one of the methods listed above.

```
: extrude
```

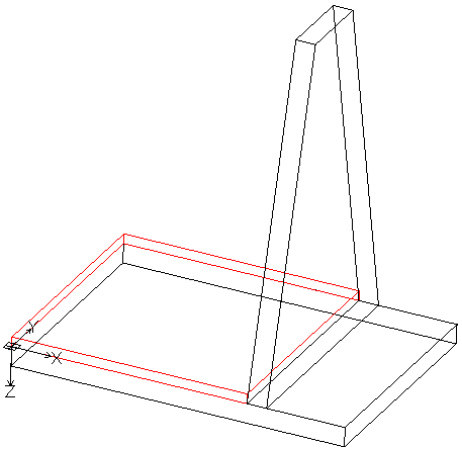
   b. Select the trapezoidal polygon.

```
Specify entities» (Pick the trapezoid)
Specify entities» (Press Enter to end entity selection)
```

   c. Specify the extrusion distance, 0.25":

```
Options: Path, Taper angle or
Specify extrusion height» .25
```

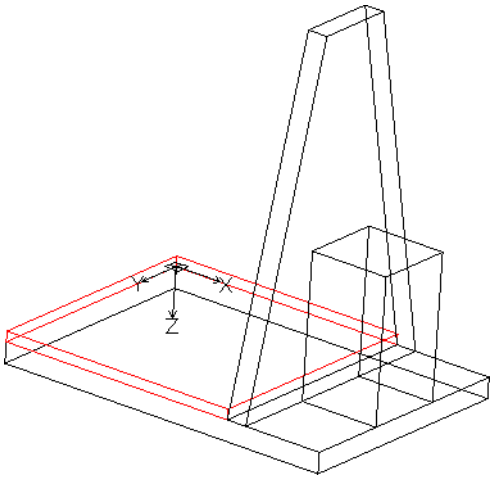
Notice that the program makes the 2D trapezoid into one that is 3D.



2D trapezoid outline turned into 3D object with thickness

4. Add the square box, which has a 1x1" base and is 1.75" tall. It is located at the center of the end of the main plate. To place it, use the coordinates shown below with the Length option:

```
: box
First corner of box» 3.25,2
Options: Cube, Length or Other corner» L
Length» -1
Width» -1
Height» -1.75
```



Box added to model

5. Press **Ctrl+s** to save your work (**Cmd+s** on Macs).

DRAWING CYLINDERS (HOLES)

Command: Cylinder
Alias: cy
Menu bar: Solids | Draw | Cylinder
Ribbon: Home | Modeling | Cylinder (3D Modeling)
Toolbar:

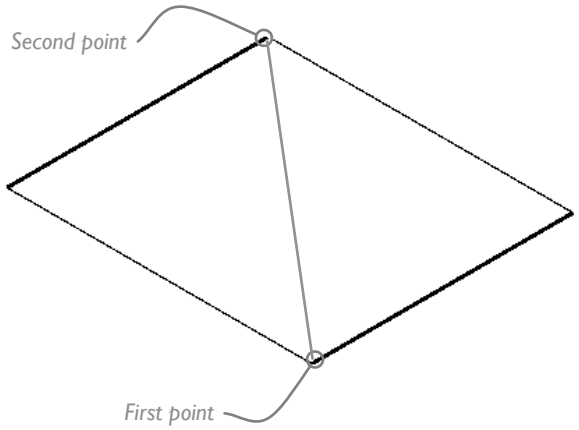
The square box has a hole drilled through it. Making holes in solid models takes two steps:

i. Draw a cylinder the same size as the hole.

ii. Subtract the cylinder from the rest of the model.

The tricky part is centering the cylinder in the square box. The easiest way to accomplish this is to draw a line that spans the diagonal of the box top, and then start the cylinder at the midpoint of the line.

1. On the status bar, ensure **ESnap** is turned on: you need the ENDPoint and MIDpoint entity snaps.
2. Start the **Line** command, and then draw a diagonal, as illustrated in the figure below.



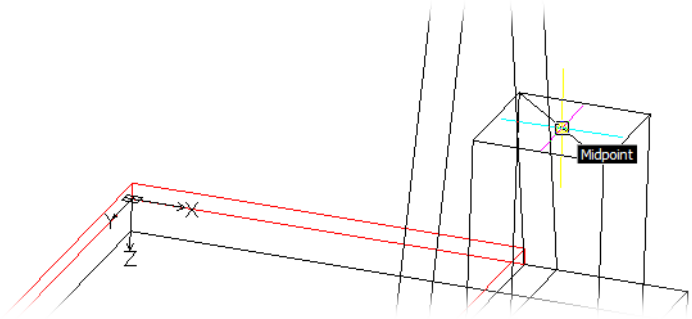
Drawing a construction line to find the rectangle's center

3. Start the **Cylinder** command using one of the methods listed above.

```
: cylinder
```

   a. Pick the midpoint of the line:

```
Options: 3P, 2P, TTR, TTT, Elliptical or
Center point for base of cylinder» (Pick the midpoint of the line)
```



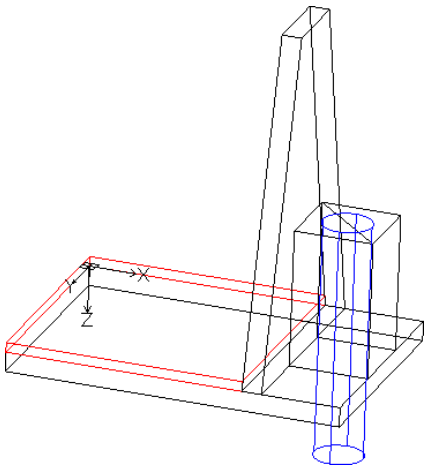
Starting the cylinder at the midpoint of the construction line

b. Specify the 0.625" diameter of the cylinder using the **Diameter** option:

```
Options: Diameter or  Radius for base of 1» d
Specify diameter for base of cylinder» 0.625
```

c. The length does not matter, because it will be erased in the next step. It only matters that the cylinder is longer than the combined length of the boxes, or 2".

```
Options: Center of other end or
Height of 1» (Drag the cursor so that the cylinder extends below
the boxes, as illustrated below)
```



The cylinder added to the model

4. Erase the line, as it is no longer needed.

The cylinder is in place, and so the next step is to cut out its shape with a Subtract command and leave behind the hole behind.

Applying Boolean Operations

Parts of solid models can be added and subtracted from each other, creating parts that look different. The operations are known as "booleans," named after George Boole who invented this logic. The operation to join two parts is known as "union"; removing one part from another is "subtract."

In the following tutorials, you use both operations, subtraction to create the hole, and then union to "weld" together all the parts of the model .

SUBTRACTING SOLIDS

Command: Subtract
Alias: su
Menu bar: Solids | Solid Editing | Subtract
Ribbon: Home | Modeling | Subtract (3D Modeling)
Toolbar:

To create holes in solid models, you subtract cylinders from the rest of the model. It can, however, be a bit tricky to understand how the Subtract command works — which entities are being subtracted and which subtracted from. Here is how to approach it:

i. First, select the entities from which the cylinder will be subtracted, In this tutorial, that's the boxes.

ii. Then, select the entities that will be subtracted. In this case, it is the cylinder.

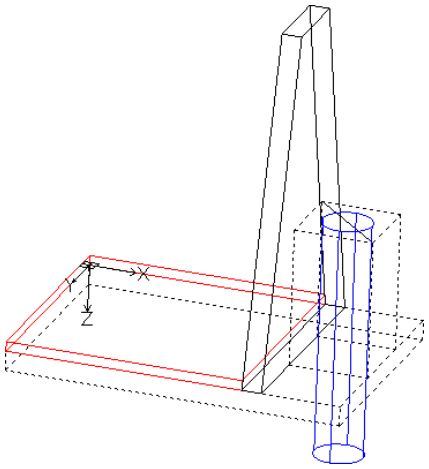
Follow these steps:

1. Start the **Subtract** command using one of the methods listed above.

```
: subtract
```

2. Select the two boxes through which the cylinder passes. The boxes are shown with dashed lines in the figure below, while the cylinder is in blue to distinguish it from the rest of the model.

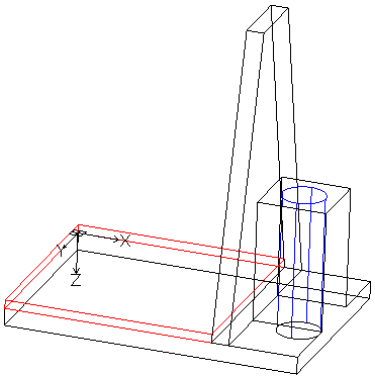
```
Specify solids and regions to subtract from...
Specify entities» (Select square box)
1 found, 1 total
Specify entities» (Select base plate)
1 found, 2 total
Specify entities» (Press Enter to continue)
```



Selecting the elements that will be subtracted from

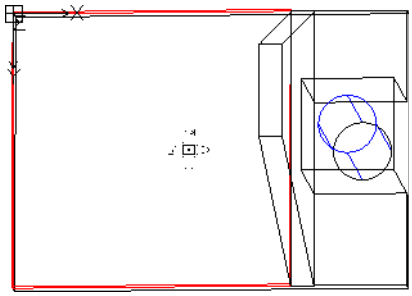
3. Select the cylinder:

```
Specify solids and regions to subtract...
Specify entities» (Select the cylinder)
1 found, 1 total
Specify entities» (Press Enter to end the command)
```



Selecting the cylinder for subtraction

4. To show that the hole now exists in the model, use the **RollView** command to twist the model around.



Twisting the view to see the newly formed hole

JOINING SOLIDS

Command: Union
Alias: uni
Menu bar: Solids | Solid Editing | Union
Ribbon: Home | Modeling | Union (3D Modeling)
Toolbar:

All of the parts should now be "welded" together, except for the red rubber mat. The welding is done with the Union command: it joins two or more solid parts into one. (Unbeknownst to you, the Subtract command unioned the two solids through which the hole passes.)

To join two or more solid entities, follow these steps:

1. Enter the **Union** command.

```
: union
```

2. Select the solid entities to join:

```
Specify entities» (Select all entities in the drawing -- except the red rubber mat)
Specify entities» (Press Enter to end the command)
```

3. Press **Ctrl+s** to save the drawing (**Cmd+s** on Macs).

The parts are now one solid model. Being a single object means it can now be analyzed for its physical properties — something not possible with a 2D drawing.

TIP Use the **Intersect** command to remove all solids except for the volume in common.

Analyzing Solid Models

Command: MassProp
Menu bar: Solids | Draw | Region/Mass Properties
Ribbon: ...
Toolbar: 

Solid models are unique in CAD, because some of their physical properties can be determined and then used for further analysis. This is done with the **MassProp** command, which is short for "mass properties." It reports on the volume, moments of inertia, and so on.

Using the command is simplicity itself; understanding the results, less so; see the boxed text for an explanation of the terms.

1. Enter the **MassProp** command.

```
: massprop
```

2. Select all the entities in the *3d model* drawing:

```
Specify entities» (Select all solid model parts)
3 found, 3 total
Specify entities» (Press Enter)
```

3. Press function key **F2** to switch to the Command Window, and then review the report:

```
---------------- SOLIDS ----------------
Mass: 7.5739
Volume: 7.5739
Bounding box X: 0.0000 -- 4.2500
Y: 0.0000 -- 3.0000
Z: -4.2500 -- 0.2500
Centroid: X=2.5566 Y=1.5000 Z=-0.5949
Moments of inertia: X=32.1967 Y=70.3458 Z=80.7330
Products of inertia:
XY: 29.0446
YZ: -6.7589
ZX: -15.0347
Radii of gyration: X=2.0618 Y=3.0476 Z=3.2649
Principal moments and X-Y-Z directions about centroid:
I: 9.7138 along [0.7864,0.0000,-0.6177]
J: 18.1623 along [0.0000,1.0000,0.0000]
K: 16.9498 along [0.6177,0.0000,0.7864]
```

The sole problem is that you cannot assign a density to represent the materials, whether steel or wood. This means that the mass is always equal to the volume, because density is set to 1.

This concludes the tutorial. You can save the drawing, and then exit the program.

SUMMARY OF MASSPROP REPORT

Mass equals the volume, because density = 1.

Volume is the 3D space occupied by the 3D solid.

Bounding Box reports the x,y,z coordinate triple of a 3D box enclosing 3D solids or bodies.

Centroid reports the x,y,z coordinates of the center of mass of 3D solids and bodies.

Moments of Inertia are *Mass* * $Radius^2$ about each axis.

Products of Inertia are *Mass* * *Distance* (of centroid to y,z axis) * *Distance* (of centroid to x,z axis).

Radii of Gyration are $(MomentOfInertia / Mass)^{1/2}$

Principal Moments are the axes along which the mass is symmetrical.

Summary

In this chapter, you learned how to create 3D models with a variety of solid modeling commands. In the next chapter, you learn another approach to 3D modeling using the PushPull command.

CHAPTER 17

Direct 3D Modeling

In this Chapter

- Understanding direct modeling
- Modeling in 3D interactively
- Applying editing operations

Direct modeling is a revolution in solid modeling that makes it possible to design as if you were working with Plasticine, directly pushing and pulling the modeling clay to interactively form 3D shapes. Indeed, the primary direct modeling command is named "PushPull."

Recall from the last chapter that I outlined two basic ways to construct 3D models:

» Draw 3D primitives, and then combine and edit them into complex shapes.

» Draw 2D entities, and then revolve and extrude them into 3D models.

This chapter works with the second approach, using 2D primitives such as circles and rectangles to make 3D versions of them. You create the same model as in the last chapter, but by using different commands. This once again shows that CAD can create the same result using different approaches. You get to decide which approach you like better!

KEY TERMS IN THIS CHAPTER

Direct modeling is modeling and editing models without needing to first specify what to edit.

Push-pull is a form of direct modeling and editing that works with entities, boundaries, and faces.

COMMANDS

Command	Shortcuts	Menu Selection
ChamferEdges	...	Solids | Solid Editing | Chamfer Edges
PushPull		Solids | Draw | Push and Pull

WHAT'S NEW IN 2019 FOR THIS CHAPTER

- **ChamferEdges** command adds chamfers to 3D models.
- **PolySolid** command draws wall-like entities (not described in this chapter)
- **PushPull** command extrudes 2D entities and faces interactively.
- **Rectangle** command gains new options:
  - » **3Corner** - constructs a rectangle by three corner points:
  - » **3Point center** constructs a rectangle by its center point, a midpoint of one size, and a corner point.
  - » **CEnter** constructs the rectangle by its center point and one corner point
  - » **Parallelogram** constructs a rectangle by two points, with the third point specifying the angle of the slanted sides

Sketching in 2D

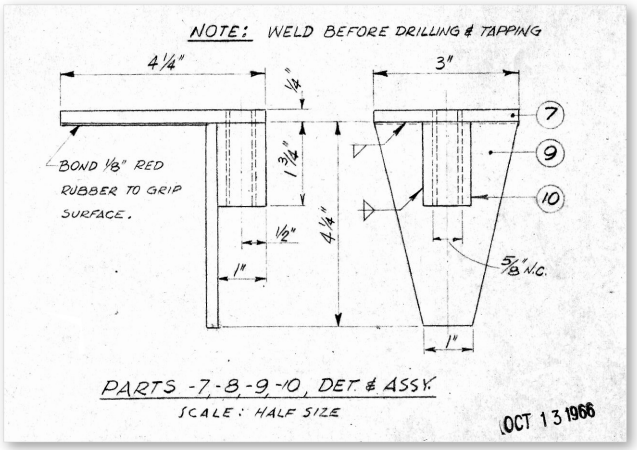
In this tutorial, you create the same 3D solid model as in the last chapter, but you use a very different approach: you use direct modeling on 2D entities. The tactic is thus:

i. Sketch everything in 2D

ii. Extrude the sketches into 3D with PushPull command

iii. Cut off some sides with ChamferEdges command

As a reminder, here are the dimensions of the model:



SKETCH EVERYTHING IN 2D

Begin this modeling tutorial by sketching the bases of all the parts as a 2D drawing. "Sketching" is the term used in 3D modeling to refer to 2D drafting, which can include constraints, although not in this tutorial.

1. Start the program with a new drawing based on the *standard.dwt* template file.

Sketching the Base Rectangle

2. To draw the base, enter the **Rectangle** command:

Rectangle drawn with the Rectangle command

```
: rectangle
```

a. Place the first corner of the rectangle at the drawing's origin, 0,0 as follows:

```
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Par-
allelogram, Thickness, line Width or Specify start corner» 0,0
```

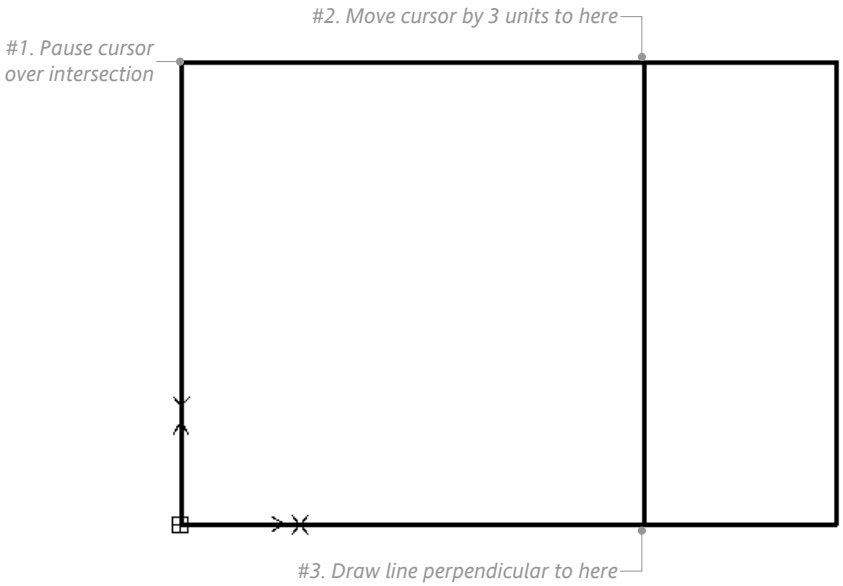
b. The rectangle is 4.25" long by 3" wide, so specify the **Dimensions** option and then enter those numbers as the input:

```
Options: Area, Dimensions, Rotation or Specify opposite corner» d
Default: 10.0000
Specify horizontal dimension» 4.25
Default: 10.0000
Specify vertical dimension» 3
```

The result is the rectangle shown above.

Sketching with Tracking

3. Draw a line across the rectangle to delineate the 3" x 3" area. This area later becomes the 1/8"-thick rubber grip.



Using tracking to start drawing a line

Because the line needs to start 3" away from a corner, use tracking to find the starting point. The on-screen movements involved with entity tracking can be tricky, so turn on ortho, entity snap, and entity tracking modes to help you out.

a. Start the **Line** command:

```
: line
```

b. At the prompt, enter "tk" to start tracking mode:

```
Options: Segments, or Specify start point» tk
```

c. Pick the corner at #1 in the figure above:

```
First temporary point (Pick at the intersection at #1)
```

d. Move the cursor rightward along to #2, and then enter **3** as the distance to move:

```
Next point (Press ENTER to end inferencing) (Move the cursor in the direction of
#2, and then enter:) 3
```

e. Tell the program to stop tracking by pressing **Enter**:

```
Next point (Press ENTER to end inferencing) (Press Enter to end tracking)
```

f. With tracking ended, the program starts drawing the line. Move the cursor downwards to the opposite line at #3, and then enter **per** to activate PERpendicular entity snap:

```
Options: Segments, Undo, Enter to exit or Specify next point» per
```

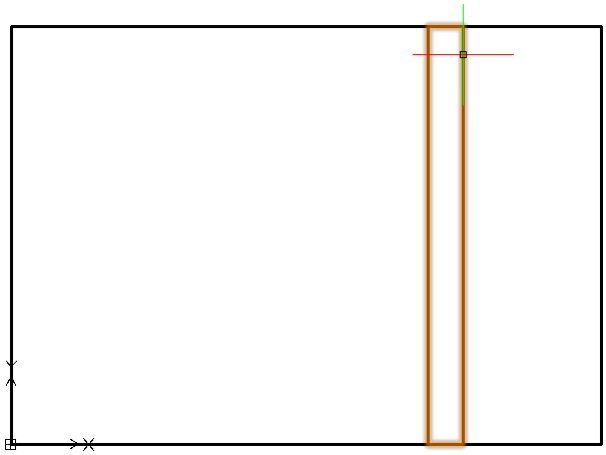
g. When the program asks "to" what, click on the line at #3:

```
to (Click on the line at #3)
```

h. Press **Enter** to end the command:

```
Options: Segments, Undo, Close, Enter to exit or Specify next point» (Press Enter)
```

4. Use the **Save** command to save the drawing by the name "3d direct model.dwg."
5. The vertical piece is sketched as another rectangle:
   a. Draw a rectangle sized 0.25" by 3", as shown by the figure below. Use INTersection entity snap to make sure it starts in the corner indicated by the cursor.



Adding another rectangle, shown highlighted here

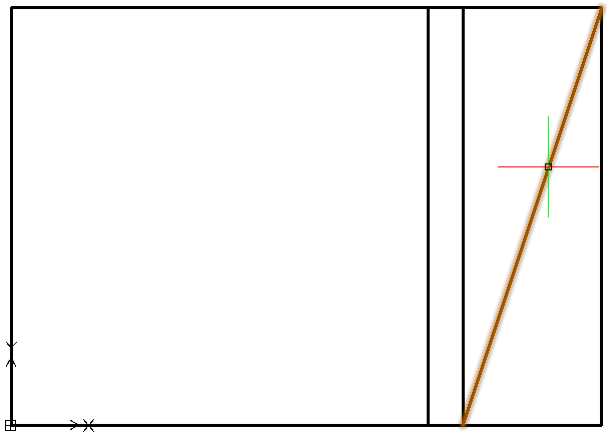
```
: rectangle
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Par-
allelogram, Thickness, line Width or Specify start corner» (Pick at corner indi-
cated by the crosshair cursor)

Options: Area, Dimensions, Rotation or Specify opposite corner» d
Specify horizontal dimension» .25
Specify vertical dimension» 3
```

Sketching With Construction Lines

6. The final elements are the tall square with the hole. You draw a square with the Rectangle command, and the hole is made from a circle. Recall from the previous chapter that the easiest way to draw the square and circle centered is to first draw a construction line:
   a. As the construction line is drawn diagonally, turn off **Ortho** mode on the status bar,

```
<Ortho Off>
```

b. Use the **Line** command to draw the line from one corner to the other, using INTersection entity snap to make the line's placement accurate:

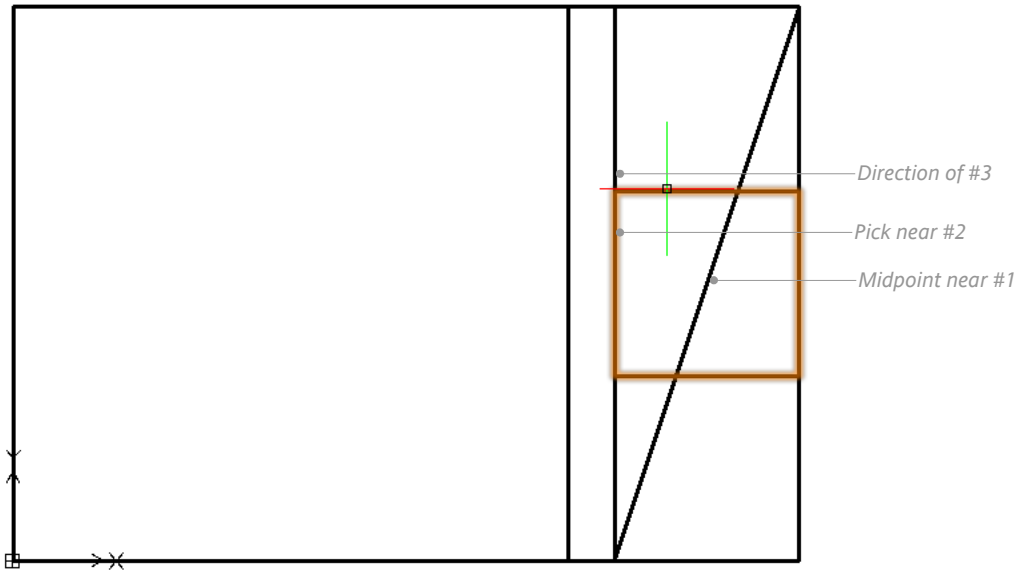


Diagonal construction line, shown highlighted

```
: line
Options: Segments, Enter to continue from last point or Specify start point» int
of (Pick one corner)
Options: Segments, Undo, Enter to exit or Specify next point» int
of (Pick the other corner)
Options: Segments, Undo, Enter to exit or Specify next point» (Press Enter to exit
the command)
```

Rectangle by 3Point-center

7. **(NEW IN 2019)** The square can be drawn with the **Rectangle** command's 3Point-center option, which is one of several new options for this command in ARES 2019:



Drawing a rectangle with three points

a. Start the **Rectangle** command, then specify the **3Point center** option by entering "3p":

```
: rectangle
Options: 3Corner, 3Point center, CEnter, COrner, Chamfer, Elevation, Fillet, Par-
allelogram, Thickness, line Width or Specify start corner» 3p
```

b. The options prompt you for a midpoint, a side, and a corner. The center of the rectangle is on the diagonal line; capture it with the MIDpoint entity snap, as follows:

```
Specify center point» mid
of (Pick the diagonal line, near #1)
```

c. When the command asks for the "mid-point of side," it's asking for the width of the rectangle. Indicate this by picking the adjacent rectangle, at #2 in the figure:

```
Specify mid-point of side» (Move the cursor to the rectangle, by #2)
```

d. When the command asks for the "corner point," it's asking for the height of the rectangle. Indicate this by moving the cursor by three units, to #3 in the figure:

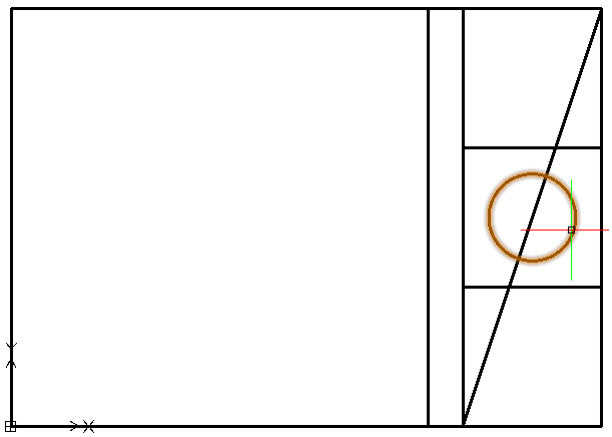
```
Specify corner point» (Move the cursor in the direction of #3, then enter:) .5
```

Sketching Circles

The final 2D sketch is the circle, which is 5/8" in diameter — 0.625". Draw it using MIDpoint entity snap to capture the middle of the diagonal line:

8. Enter the **Circle** command, and then respond to the prompts, as follows:

```
: circle
Options: 3Point, 2Point, Ttr, TTT or Specify center point» mid
of (Pick middle of diagonal line)
Default: 1.0
Options: Diameter or Specify radius» d
Default: 2.0
Specify diameter» .625
```



Circle added to center of square, shown highlighted

9. Remove the diagonal line with the **Erase** command, as we no longer need it.
10. Save the drawing by pressing **Ctrl+S**.

PULLING 2D TO 3D

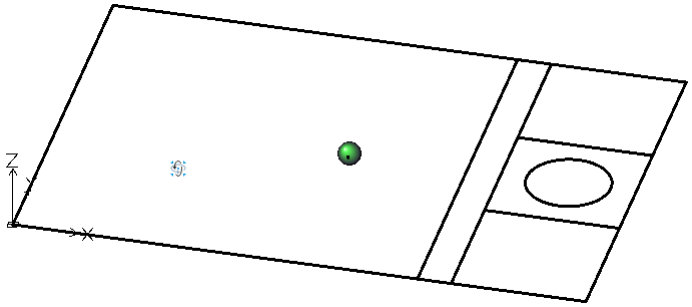
Command: PushPull
Alias: --
Menu bar: Solids | Draw | Push and Pull
Ribbon: Home | Modeling | Push and Pull (3D Modeling)
Toolbar: Modeling | PushPull

(NEW IN 2019) With the 2D entities drawn, you now use the PushPull command to drag them thicker. Follow these steps:

1. To better see what is going on, use the RollView command to change your viewpoint:
   a. Enter the **RollView** command:

      `: rollview`

   b. Hold down the left mouse button, and then drag the rollview cursor to change the view from plan to 3D. (Recall that the green ball reports the center of rotation.) The viewpoint you end up with depends on how you move the mouse.



3D view of the 2D entities

   c. Once you have the 3D viewpoint as you want it, press **Esc** to exit the command:

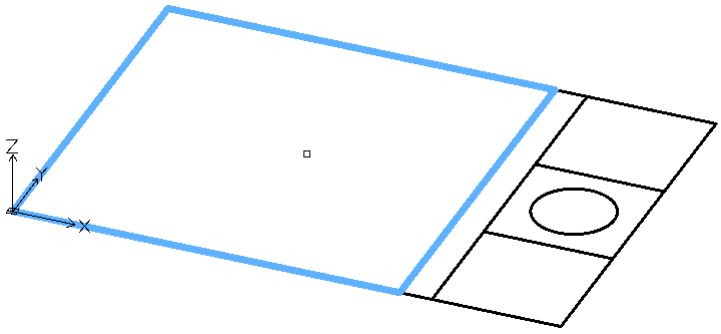
      `Press ESC or ENTER to exit. (Press Esc)`

2. You use the PushPull command to extrude the 2D entities in 3D solids:
   a. Start the command with one of the methods listed above:

      `: pushpull`

   b. Pick a point inside the rectangle shown in the figure. This is the red rubber grip that is 1/8" tall — 0.125".

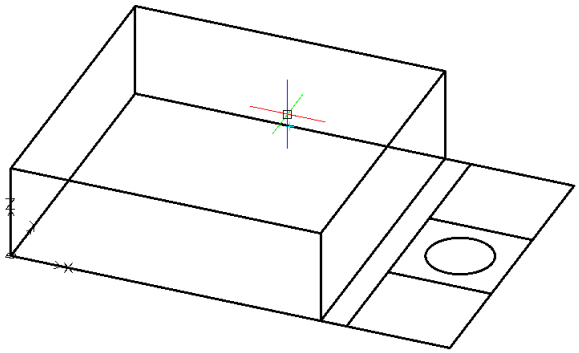
      `Specify entity or internal point» (Pick a point inside the rectangle)`



Choosing a rectangular area to extrude

Notice that the program recognizes the boundary (shown in blue) created by the line and three sides of the rectangle.

c. As you move the cursor, notice that the program previews the height of the extrusion:

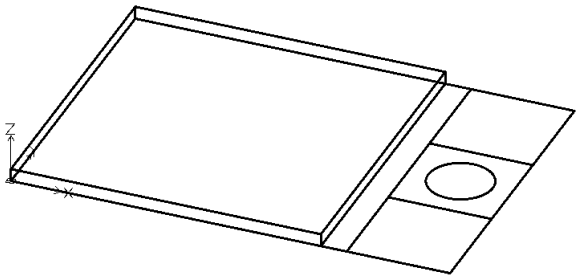


Interactively showing the height of the extrusion

Specify the height of 0.125":

```
Options: Multiple or Specify height» .125
```

And the first solid is created using the push-pull method:



First solid of the 3D model

Push-pulling Entities

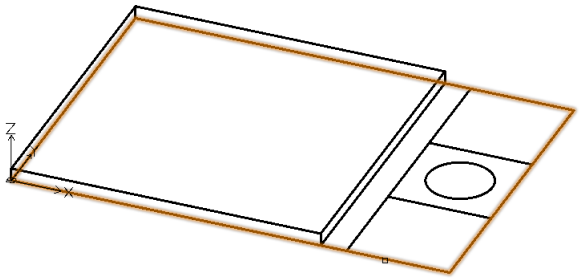
You did the first extrusion with the PushPull command by picking inside an area. This time, you use the Entity option, which extrudes an entity.

1. Restart the command by pressing the spacebar:

```
: (press spacebar)
pushpull
```

2. Pick the large rectangle entity, as shown in the figure:

```
Specify entity or internal point» (Pick the large rectangle)
1 found
```

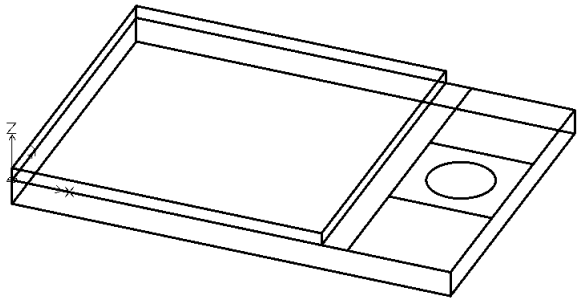


Choosing the rectangle entity to extrude

3. Specify a height of 0.25" but this time make it negative, so that the extrusion is drawn downwards, as in **-0.25"**:

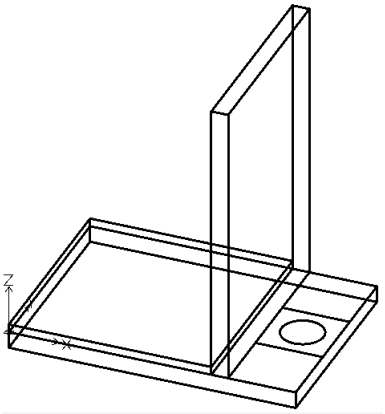
```
Options: Multiple or Specify height» -0.25
1 extrusion created
```

And the second solid is created using the push-pull method:



Two 3D solids created from two 2D entities

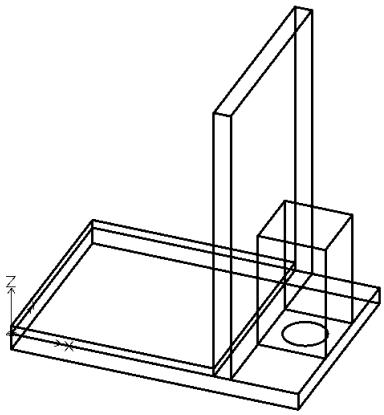
4. Continue using the PushPull command on the remaining 2D entities. Extrude the narrow rectangle to 4.25". You can select the entity or pick inside of it.



Rectangle pulled to 4.25" high

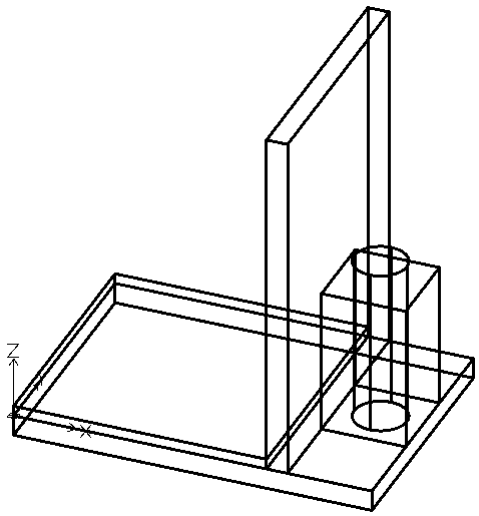
Later, you use the ChamferEdges command to make it look pyramidal.

5. Extrude the square to 1.75":



Square pulled to 1.75" high

6. Extrude the circle into a cylinder. As the height of the cylinder does not matter, extrude it to 2" or more to make it easier to select later, when it comes time to turn it into a hole.



All 2D sketches extruded into 3D

7. Save the drawing by pressing **Ctrl+S**.

SUMMARY OF PUSHPULL COMMAND OPTIONS

```
: pushpull
Specify entity or internal point»
1 found
Options: Multiple or Specify height» .25
```

Specify entity selects an open or closed entity to extrude.

Internal point picks a point inside a closed entity.

- » Closed entities like rectangles and circles extrude into 3D solids.
- » Open entities like lines and arcs extrude into 3D surfaces.

Multiple selects more than one 2D entity to extrude.

Specify height specifies the height of extrusion; you can enter a value or drag the cursor:

- » Positive values extrude upwards
- » Negative values extrude downwards
- » If the extrusion goes through an existing solid, then it forms a hole.

Switching to Hidden Line Mode

At this point, the drawing is getting complex enough that it is probably helpful to change the viewing mode to Hidden. This is done with the ShadeMode command, as follows:

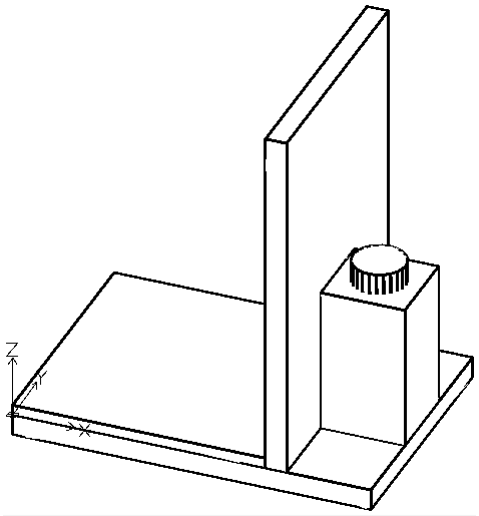
1. Enter the ShadeMode command:

```
: shademode
Active mode: 2D
Default: 2D
```

2. Specify Hidden mode:

```
Options: 2D, 3D wireframe, Hidden, Flat, Gouraud, fLat with edges or gOuraud with
edges
Specify option» h
```

Ah, that's better!



Hidden lines removed

Chamfering in 3D

Command: ChamferEdges
Alias: --
Menu bar: Solids | Solid Editing | Chamfer Edges
Ribbon: Home | Solid Editing | Chamfer Edges (3D Modeling)
Toolbar: Solid Editing | Chamfer Edges

(NEW IN 2019) You use the ChamferEdges command to slant the sizes of the tall box. This command works on the edges or faces of 3D entities, because the regular Chamfer command does not.

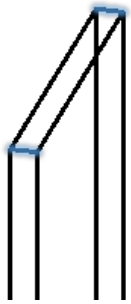
1. Enter the **ChamferEdges** command:

```
: chamferedges
Active chamfer Dist1 = 1.0000, Dist2 = 1.0000
```

2. Specify the distances by which the program should cut back the edges. The chamfer distances are 1" wide and 4.5" tall. Use the **Distance** option to specify the two distances:

```
Options: Loop, Face, Distance or Specify edge» d
Active chamfer Dist1 = 1.0000, Dist2 = 1.0000

Default: 1.0000
Specify first distance» 4.25

Default: 1.0000
Specify second distance» 1
```

3. Now pick the two short edges at the top (highlighted in blue in the figure below) as the ones to be chamfered:



Selecting edges to chamfer

```
Options: Loop, Face, Distance or Specify edge» (Pick one edge)
1 found
Active chamfer Dist1 = 1.0000, Dist2 = 1.0000
Options: Loop, Face, Distance or Specify edge» (Pick the other edge)
1 found
```

SUMMARY OF CHAMFEREDGES COMMAND OPTIONS

```
: chamferedges
Active chamfer Dist1 = 1.0000, Dist2 = 1.0000
Options: Loop, Face, Distance or Specify edge»
```

Specify edge picks an edge to chamfer

Loop prompts you to pick an edge, and the program selects all edges of the face adjacent to the edge; as there are two faces adjacent to every edge, you also need to specify which face to chamfer

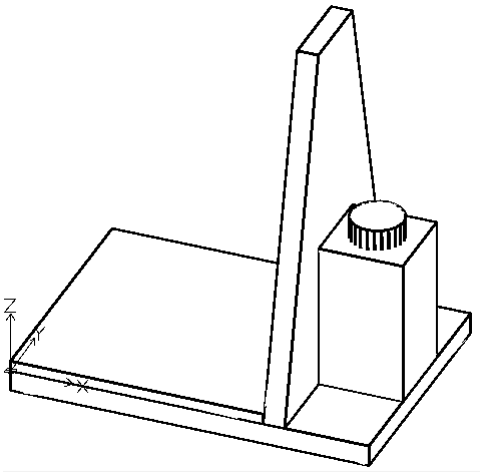
Face prompts you to pick a face, and then the program selects all edges of the face

Distance specifies the first and second chamfer distances.

Tip This command does not work on round faces, such as on a sphere or the curved part of a cone.

4. With the chamfers specified, press **Enter** to exit the command:

```
Options: Loop, Face, Distance or Specify edge» (Press Enter to end the command)
```



Chamfered edges

5. Press **Ctrl+S** to save your valuable work.

When chamfer distances are not equal, as in this case, they can be tricky to apply. You may at times need to try one or the other set to see which one works correctly.

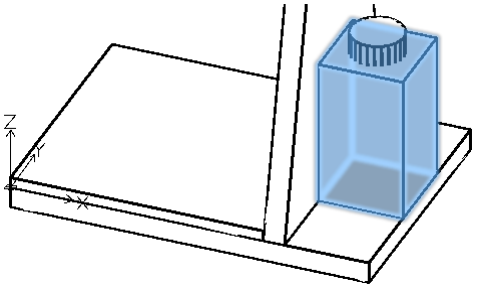
DRILLING HOLES

The final element of the 3D model is to add hole. At present, it is a cylinder. You turn it into a hole by removing the cylinder from the square solid with the Subtract command.

1. Enter the subtract command:

```
: subtract
```

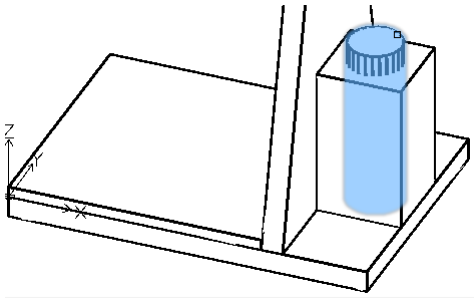
2. First you select the entity that remains after the subtraction process. This is the square solid, as highlighted in blue:



Selecting the remaining solid

```
Specify solids and regions to subtract from...
Specify entities» (Pick the square solid)
1 found
Specify entities» (Press Enter to continue)
```

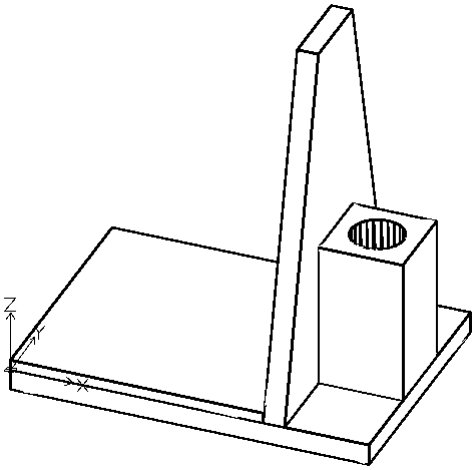
2. Now you select the solid that is to be removed, as shown in blue in the figure below:



Selecting the subtracting solid

```
Specify solids and regions to subtract...
Specify entities» (Pick the cylinder)
1 found
Specify entities» (Press Enter to end the command)
```

3. Success!



Completed 3D model

Press **Ctrl+S** to save your valuable work. If you wish, print it on paper with the Print command.

Summary

In this chapter, you learned how to work with direct 3D modeling using the PushPull command. In the next chapter, you learn how to set up multiple views of this 3D drawing through the use of viewports.

Notes

CHAPTER 18

Multiple Viewports, Multiple Drawings

In This Chapter

- Viewing 3D models from multiple viewpoints in view tiles.
- Creating overlapping viewports in sheet mode.
- Attaching referenced drawings.

Viewing a 2D drawing is easy, because the entire drawing is always visible to us from plan view. Viewing 3D models is tougher, because we to need to work on and see six sides — front and back, left and right, top and bottom — yet CAD programs let us see just one or two at a time; 3D displays are not yet generally available.

In this chapter, you learn how to view 3D drawing from multiple angles at the same time using viewports.

KEY TERMS IN THIS CHAPTER

Sheet mode displays drawings as if they were on a sheet of paper; also known as paper space and as layouts.

Reference files refers to additional drawing files attached to the active drawing.

View tiles refers to tiled windows in model mode.

Viewports refers to overlapping windows in sheet mode.

ABBREVIATIONS

Ctrl+R Moves the focus from one view tile or viewport to the next (press Cmd+R on Mac).

Ctrl+Tab Switches focus between open drawings (press Cmd+Tab on Mac).

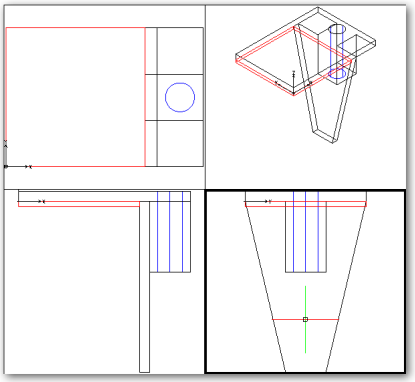
COMMANDS

Command	Shortcuts	Menu Selection
References	xr, xref	Tools | References
Viewport	mv, mview	...[1]
-Viewport	-vports	...
Viewtiles	viewports, vports	View | View Tiles | View Tiles Manager

[1] No menu selection.

To quickly move the viewpoint around the model, we can use last chapter's RollView command, but still we see just one side at a time on our 2D monitors. To view a 3D model from multiple viewpoints at the same time, we need an alternative.

The **Viewtiles** command splits the screen into two or more windows. We can see a different side of the 3D model in each window. The windows are called "view tiles," and each one can display a different viewpoint in model space, allowing us to see two, three, even all six sides of a 3D model at once. (A separate command does a similar thing in paper space, **Viewports**.)

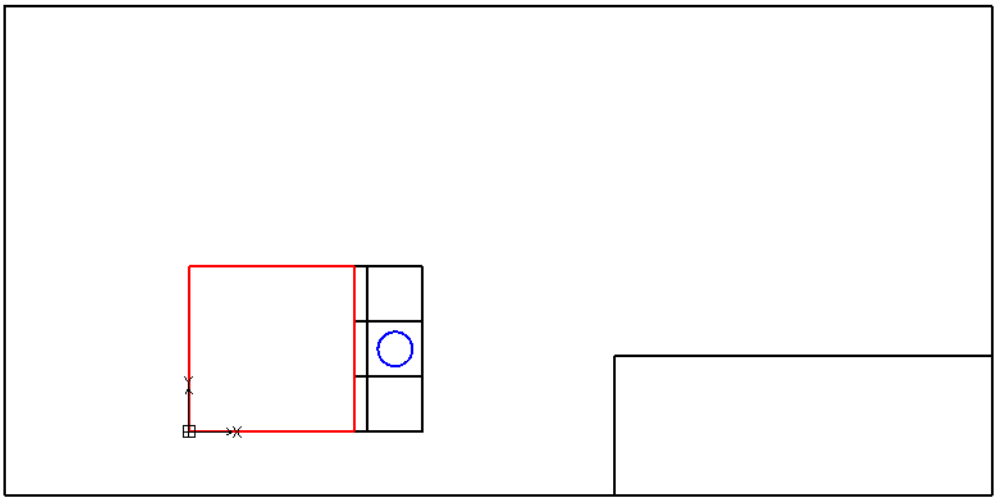


Model space split into four viewtiles and showing a 3D model's top, isometric, front, and side views simultaneously

A nice touch is that each view tile can have settings independent of the others. For example, each view tile can have different zoom levels, and a different view orientation; the snap and grid can be on or off, and have different spacings in every view tile.

(As we split the screen into view tiles, however, the problem becomes that each one provides a smaller view than before. One way to overcome this is to make some view tiles larger than others: larger ones for working in, smaller ones for monitoring with.)

In addition to showing different viewpoints of one model, a drawing can display one ore more other drawings at the same time. This is done by attaching additional *.dwg* files with the **References** command, with each one looking like you had drawn them in place. These attached drawings are known as "reference files," and the process is not too different from attaching image files, as in Chapter 14. Reference files can be somewhat independent of one other.



Two drawings displayed simultaneously; the border drawing is referenced

The program can open multiple drawings separately through repeated use of the **Open** command. To switch between them quickly, press the **Alt+Tab** key or Cmd+~ on Mac. (These shortcuts works with most other programs, too.)

And multiple copies of the program can run at the same time, each with multiple drawings, each drawing with multiple references and multiple viewports/viewtiles!

In this chapter, you learn how to create multiple viewtiles, and then to attach multiple drawings. You see the effect these functions have on your work flow.

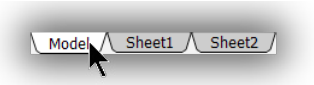
Creating Multiple Viewports

Command: Viewtiles
Aliases: viewports, vports, vtiles
Menu bar: View | View Tiles | View Tiles Manager
Shortcut: Ctrl+R (Cmd+R on Macs)
Toolbar: --
Ribbon: Sheet | View Tiles (Drafting and Annotation)

In the following tutorial, you split the area of a drawing into four *view tiles*, and then set up each tile to display a different view of a 3D model. The model you use is the one that you built in Chapter 16; the views will be the four standard views used by engineers, even in the days of paper drawings: front, right, top, and isometric view.

If necessary, start the program and then open the *3d model.dwg* file.

1. To create view tiles, ensure the program is in model space: click the **Model** tab to make sure of this.

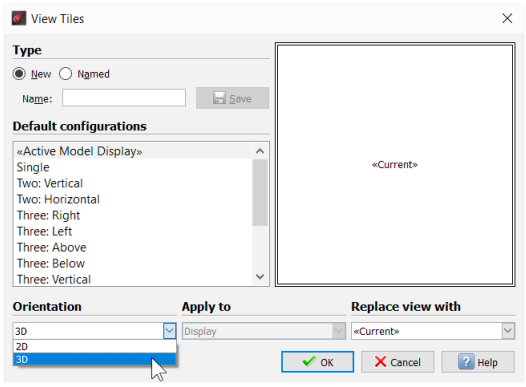


Clicking the Model tab to ensure the program is in model space

2. Enter the **Viewtiles** command using one of the methods listed above:

```
: viewtiles
```

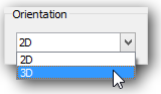
Notice that the View Tiles dialog box has a long list of default view tile arrangements listed on the left side, and a preview area on the right side.



Dialog box for creating view tiles

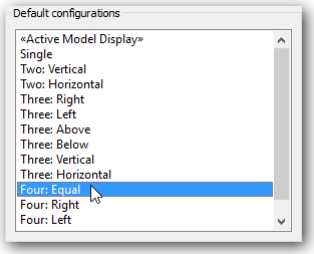
3. The job of the View Tiles dialog box is to split the drawing area into two or more view tiles. To do so, follow these steps:

   a. From the **Orientation** droplist, choose **3D**. This option changes the kinds of views to three-dimensional ones so that the 3D model is seen from multiple viewpoints.



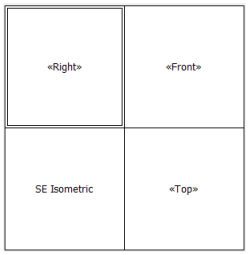
Choosing "3D" from the Orientation droplist

   b. In the Default Configurations list, choose **Four: Equal**. This option splits the drawing area into four equal-sized view tiles.



Choosing the number of view tiles to create

Notice that the Preview area shows names of the standard engineering views: Right, Front, SE Isometric, and Top. Three of them show you flat 2D-like views of the 3D model, while the isometric one shows you a 3D view.



Default arrangement of view tiles

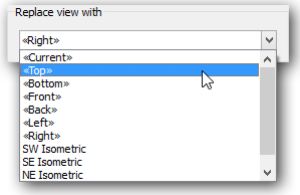
4. There is a problem, however: the views are not in correct for designers working in North America. Engineering drawings in Canada and USA use this arrangement of views:

Top	SW Isometric
Front	Right

You change the view arrangement with the **Replace View With** droplist, like this:

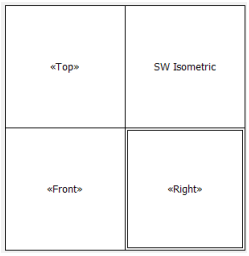
   a. In the Preview area, click the **<Right>** view tile.

   b. From the **Replace View With** droplist, choose a different viewpoint name. Replace <Right> with **<Top>**.



Replacing view tile orientations

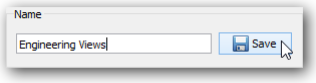
c. Repeat for the other view tiles — <Front>, <Right>, and <SW Isometric>. You end up with the correct arrangement. The Preview window should look like this now:



View tiles with corrected orientations (views)

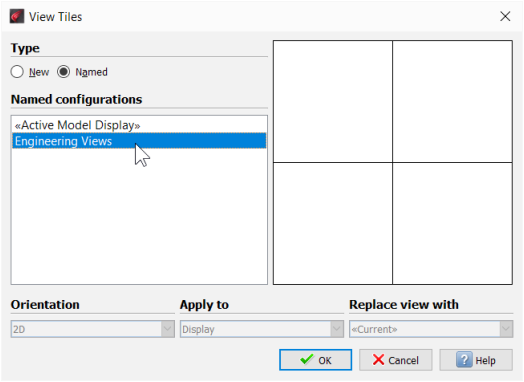
5. The program won't remember this arrangement of view tiles unless you save it with a name:

a. In the **Name** field, enter "Engineering Views," and then click **Save**.



Saving the view tile orientation

b. The next time you open the Viewtiles command, click the **Named** radio button, and "Engineering Views" will be listed under Named Configurations. (See figure below.)



Selecting a view tile arrangement that you previously saved by name

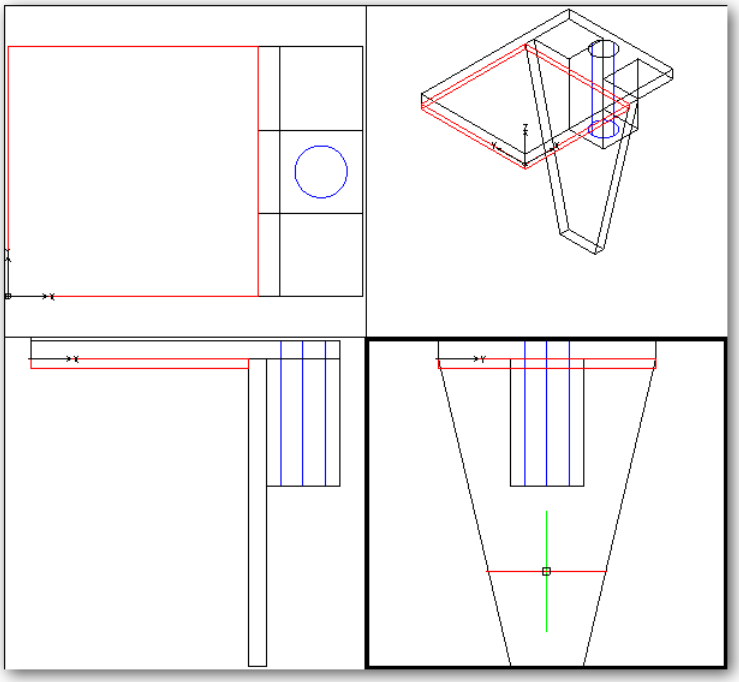
---

TIPS The catch to using saved view tile arrangements is that they are saved only with this drawing, and not to other drawings. This means "Engineering Views" won't be available in other drawings. My suggestion is to save these custom arrangements in a DWT template file so that they are always available.

The **<Right>** view tile has a double border. This means that it will be the active view tile when you exit this dialog box.

---

6. Click **OK**. Notice that the drawing area splits into four, and that each view tile shows the 3D model from a different viewpoint. The heavy border surrounds the active view tile, the one in which you can work.



The 3D model appearing in the four view tiles

Tps for Working with View Tiles

You can work on the model in any view tile, just as if it were full-screen. However, **you must click on a view tile before you can work in it**, otherwise the program doesn't know which one you are focussed on. The active view tile is the one with the heavy border.

A nice touch is that any thing you do in the active view tile is shown in real time in the other view tiles.

You can draw and edit objects between view tiles. For example, start a line in one view tile, click on another view tile, and then continue drawing the line segment.

The same coordinate system is displayed in all view tiles, which is inconvenient for drawing and editing. To change x,y-plane of the coordinate system to match the view point, use the **CCS** command with the **View** option, as follows:

1. Click on the view tile you want to change.
2. Enter the **CCS** command (a.k.a. UCS command):

   ```
   : ccs
   ```

3. Specify the **View** option:

   ```
   Options: align to Entity, NAmed, Previous, View, World, X, Y, Z, ZAxis or Specify origin: v
   ```

   Notice that the CCS coordinates rotate to match the view tile's viewpoint

If you see the model only partially in a view tile (or not at all), click the view tile to activate it, and then use the **Zoom Fit** command.

MORE TIPS As an alternative to clicking on a view tile to make it active is to press **Ctrl+R** (Cmd+R on Mac) to move the focus to the view tile in which you want to work. Keep pressing Ctrl+R until you reach the view tile you want.

You can set drawing aids — such as snap, grid, and even hidden-line removal (**HideView** command) — independently for each view tile.

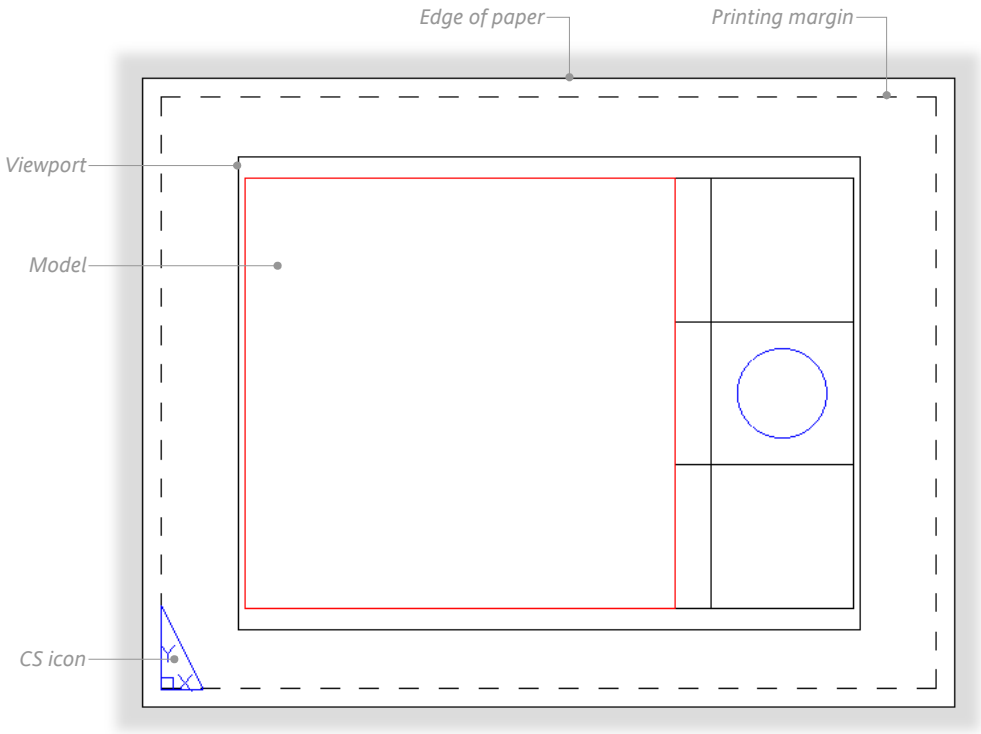
View tiles have their limitations, however:

- View tiles can only be tiled, which means you cannot overlap them, unlike programs in Windows or viewports in paper space (layout mode).
- You cannot resize vile tiles easily; to resize, you have to go back into the ViewTiles command then choose a different configuration from the Default Configurations list.
- You can only print the active (current) view tile — and not all four at once.

I don't know why model space has these limitations for viewports. But there is a workaround: create viewports in layouts (sheets), as described next.

SHEETS AND VIEWPORTS

A second way to take advantage of multiple viewpoints is through *sheets*, which are designed for preparing drawings for printing. Plus, they prove to be more flexible than view tiles. In other CAD programs, sheets are known as "layouts." Indeed, a sheet looks like a piece of paper onto which one or more viewports are opened, as illustrated below.



Elements of a sheet

Edge of Paper shows the sheet of paper used by the currently-assigned printer. In most cases, the white rectangle represents a standard A- or A4-size sheet.

Printable Margin uses the dashed line to indicate the printer's margin. Beyond this, the printer cannot print, and so it makes sense to keep the model within the dashed rectangle.

Background is shown in gray; it too is an unprintable area.

Viewport is the black rectangle that shows the 3D model or 2D drawing. Paper mode can have multiple overlapping viewports.

CCS Icon looks like the triangle used for hand drafting, and reminds you that the drawing is currently is sheet mode.

Like view tiles, sheets can show any part of a 2D or 3D drawing. However, whereas a drawing can have just one Model tab with one set of view tiles, the drawings have as many as 256 Sheets tab. Each tab can be customized for a specific printer or print job — in fact, that's the point to Sheet tabs: laying out the drawing in the way you want it printed.

For instance, one layout might be set up for a small-format color printer, while another is for printing to a large-format monochrome one.

A sheet can have one or more *viewports*, which are like view tiles but far more flexible. Each viewport presents a view of the drawing, such as a 2D plan view, a 3D viewpoint, or a zoomed in view. You can overlap viewports, use them to clip portions away from the drawing (to hide areas you don't want to show), and even make them from non-rectangular boundaries.

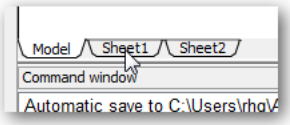
They are created and edited with the **Viewports** command, or directly using grips. In fact, viewports are considered entities in their own right. This is the primary difference between viewports and view tiles: viewports are flexible entities by view tiles are just static rectangles. With viewports, you can...

- Make the viewport of any closed shape, such as circle or spline (provided the shape doesn't intersect itself)
- Edit the boundary of viewports
- Change the color of the viewport's boundary
- Resize, move, and copy viewports
- Lock viewport boundaries so that they cannot be moved
- Freeze the viewport layer so that the viewport is invisible

Tutorial: Switching to Sheet Tab

In the following tutorial, you switch the drawing from model tab to sheet tab, and then create several overlapping viewports.

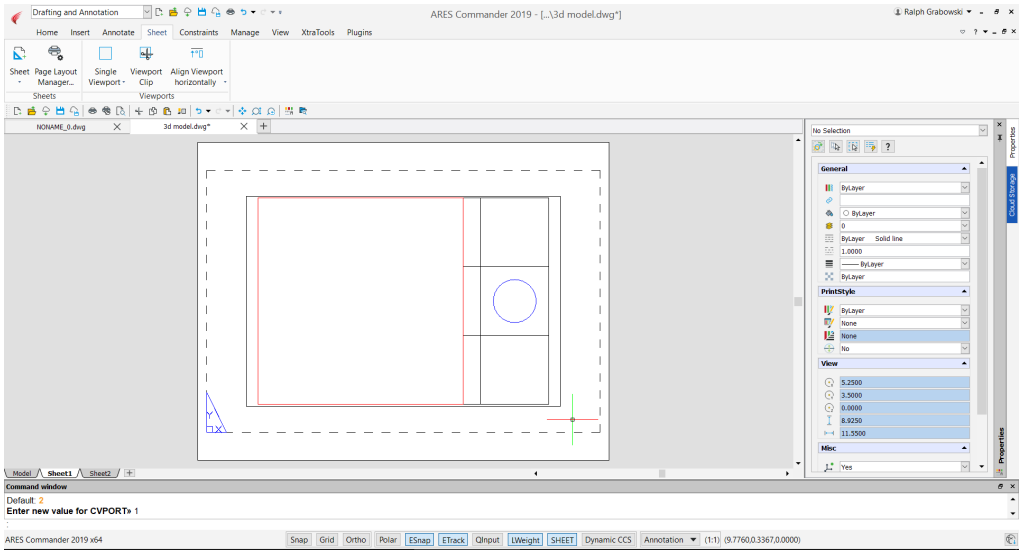
1. In the program, look below the drawing area and spot the three tabs labelled Model, Sheet1, and Sheet 2.
2. Click on **Sheet1**.



Switching to model to sheet layout

Notice that the drawing area changes its look, and that the plan view of the 3D model appears. This is known as sheet mode. On the status bar, notice the **SHEET** button. It reminds you that the drawing is currently in sheet mode.

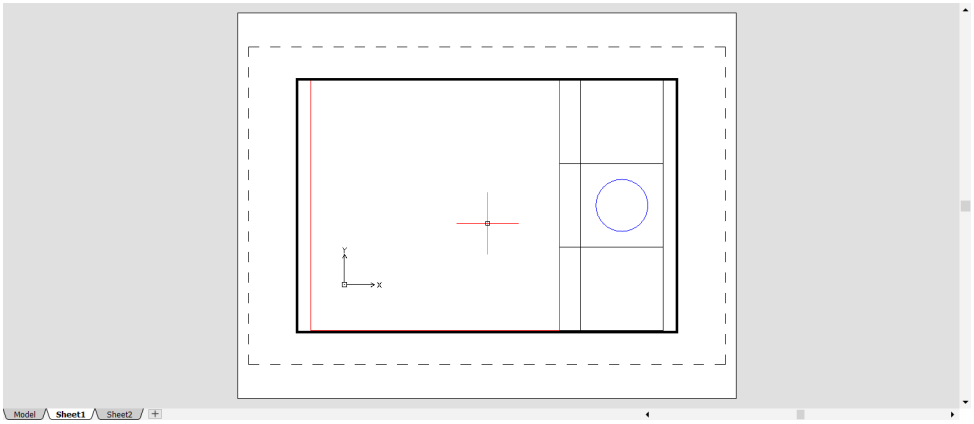
Status bar reports drawing is in sheet mode



The program in sheet mode

3. Just as you can edit in view tiles, you can edit in sheet mode's viewports:

   a. Move the cursor inside the viewport, and then double-click.

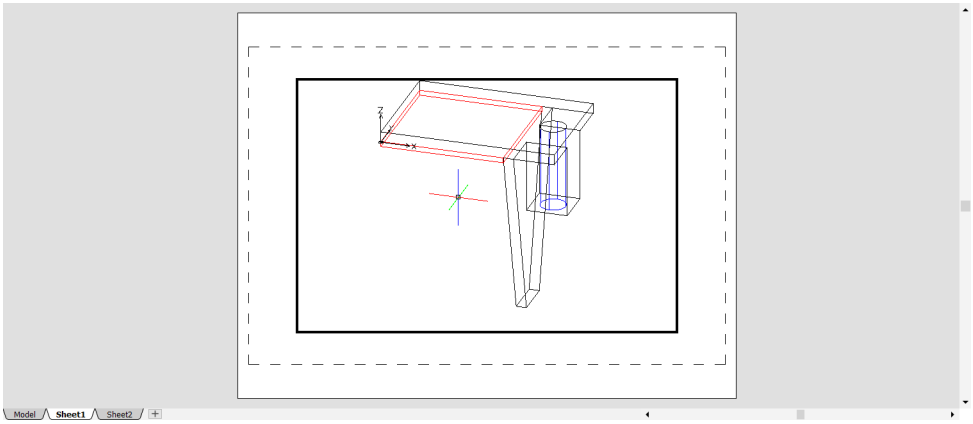


Working in model space

Notice that these changes take place:

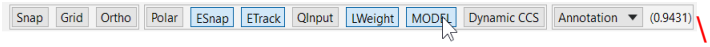
- The viewport is surrounded by a heavy black rectangle.
- The CCS icon moves to inside the viewport, and reverts to its familiar look.
- The button on the status bar changes to **MODEL**.

b. If you wish, use the **RollView** command to change the viewpoint of the model. The screen should now look something like the figure below.



3D model appearing in paper space

c. To switch back to sheet mode, double-click outside the viewport (or click the **MODEL** button).



Click the MODEL - SHEET button

PLACING TEXT IN VIEWPORTS

Text and other annotative entities, like dimensions and linetypes, are dependent on the plotting size, and so they need to be given special treatment. The program can calculate the correct size of annotative entities semi-automatically, if you let it. (From now on, I'll just refer to text.)

Here is how it works: you tell the program the scale factor at which you plan to print the drawing, and it figures out how big to make the text so that it is plotted at the right size. The scale factor at which you print the drawing depends on the size of the paper (such as A-size, A0-size, and so on) and the extents of the drawing.

```
Annotative scale = Sheet scale = Plot scale
```

All this applies especially when working with viewports, because the whole point to sheet mode is that it sets up the drawing for plotting. Remember that you can have as many sheets as you want for a drawing, one for each printing situation to handle various sizes of paper, whether the plot is in color or shades of gray, and so on.

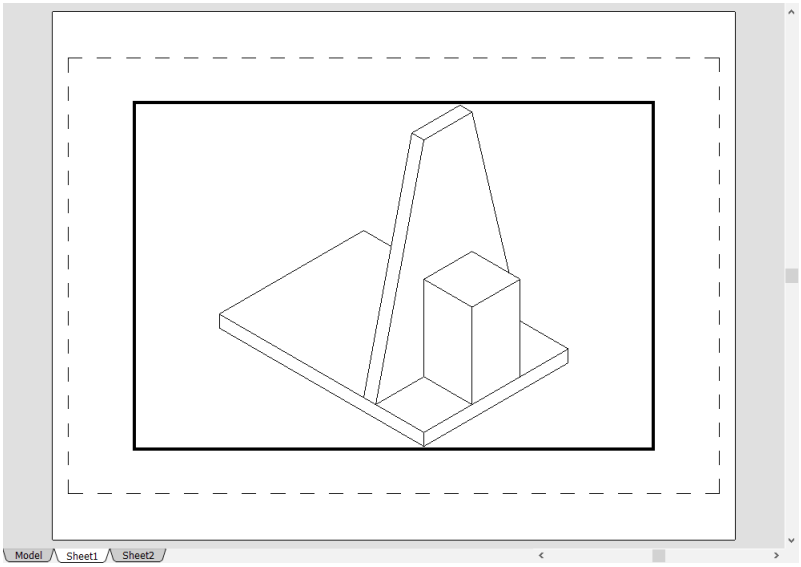
TIP *Scale* is the inverse of *scale factor*. A scale of 0.125 is a scale factor of 1:8.

Annotative Text Scale Factors

Chapter 9 provided a tutorial on creating text with annotative text scale factors. I repeat it here in a condense manner to apply it to dimensions in this chapter's tutorial.

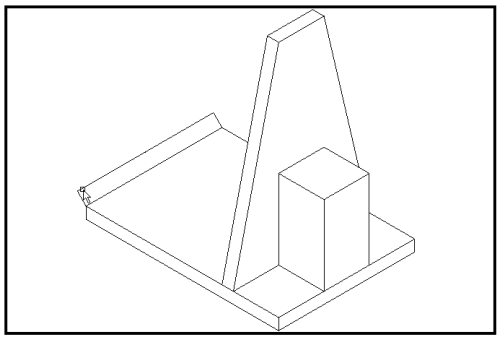
1. Open the *chapter 13.dwg* file. I've scaled the drawing from the previous chapters 40x larger to drive home the benefit of annotative scaling.

2. Make sure you are in the **Sheet1** tab, and then enter model space by double-clicking inside the viewport's border. You know that you are successful when the viewport border turns into thick lines, as illustrated below.



Inside the model space of Sheet1

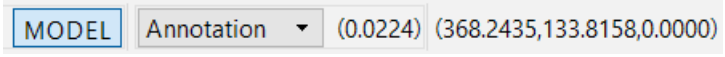
3. Place a dimension without the benefit of annotative scaling. Use the **DimAligned** command, with the following result.



The barely visible dimension being pointed out by the cursor

Notice that the elements of the dimension are nearly invisible. Now, you could use the DimScale variable to change the size of the dimension. But the better solution is to use annotative scaling.

4. At the right end of the status bar, look for the current scale. For this drawing, you should see (0.0224) just before the coordinates.



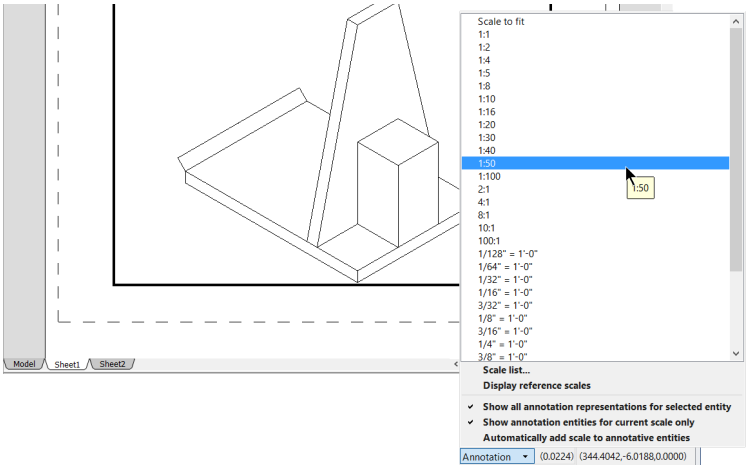
Annotative scale reported on the status bar

The program worked out the scale of 0.0224 from the extents of the drawing fitting the viewport, placed on an A-size sheet of paper.

5. Because the scale of 0.0224 is the same as a scale factor of 1:44.6, we need to need to change the size of the drawing inside of the viewport to a standard factor. (How to convert scale to scale factor: with your calculator, divide 1 by the scale of 0.0224 — the answer is 44.6.)

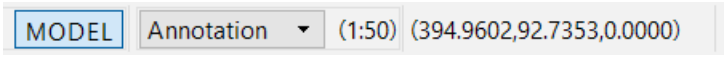
   For the scale factor, I picked the next largest whole number, 50, because (1) scale factors are usually whole numbers, and (2) the next largest number makes the drawing slightly smaller in the viewport. We can't have some of the drawing hidden by the edges of the viewport!

6. Click the annotative scale list, and then select the factor of the nearest value that results in a smaller looking drawing. In my drawing, the nearest standard scale factor to 44.6 is 1:50.



Picking the closest scale factor

You know that you have picked the correct scale when the drawing becomes slightly smaller, but not too much smaller. Notice on the status bar that the reported scale factor matches the annotation factor you selected.



Annotation scale changed to annotation scale factor of 1:50

7. With the annotative scale factor set up, you can now place dimensions — in Model tab.

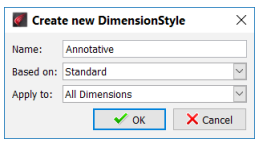
   (Here's why, as explained by Bill Fane: "If you zoom and pan while entering text in a layout tab, you mess up the viewport's scale; when you go back into model space, the annotations are missing, because their scale no longer matches the scale of the viewport. For this reason, it is best to enter annotative text in Model tab.")

   Therefore, switch to **Model** tab.

8. The drawing probably does not include a annotative dimension style, so quickly create one from the existing Standard style:

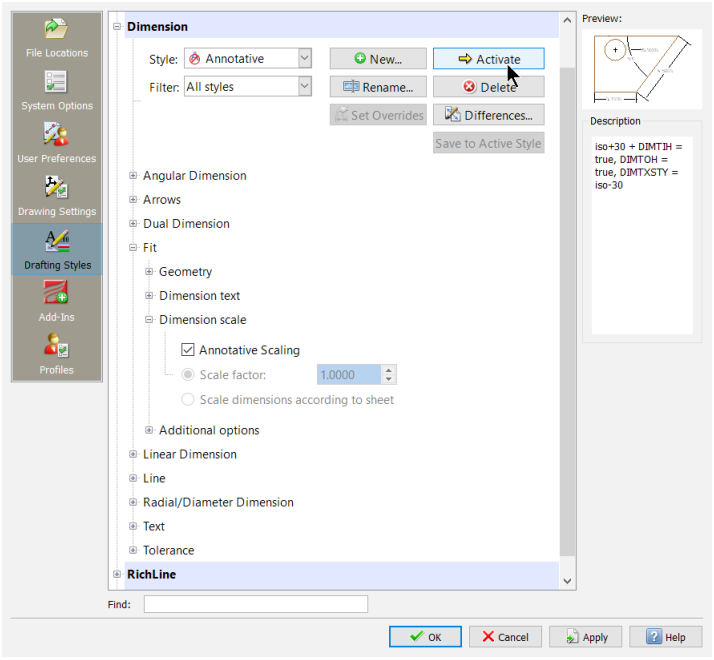
   a. To modify a dimension style, enter the **DimensionStyle** command.

   b. Make a copy of the existing "Standard" style by clicking the **New** button and giving it name of "Annotative."



Naming the annotative dimension styel

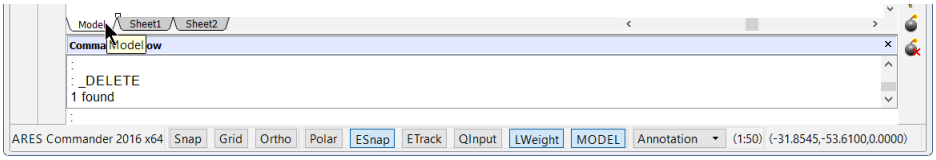
c. Open the **Fit** node, and then open the **Dimension Scale** node. Check the **Annotative Scaling** option, and then click "Activate."



Making a dimension scale annotatively-scaled

d. Click **OK** to close the dialog box.

9. Make sure the drawing is in **Model** tab, not in model space of Sheet1 tab. Change the annotative scale factor to 1:50.



In Model tab, setting the annotative scale factor to 1:50

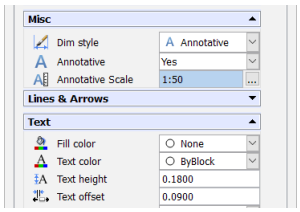
10. Run the **ParallelDimension** command, and then dimension a part of the drawing. Notice that the text and arrowheads are legible!

11. When you select the dimension for the Properties palette, notice how the program handles the scaling for you:

» Under Misc, the **Annotative Scale** factor is 1:50

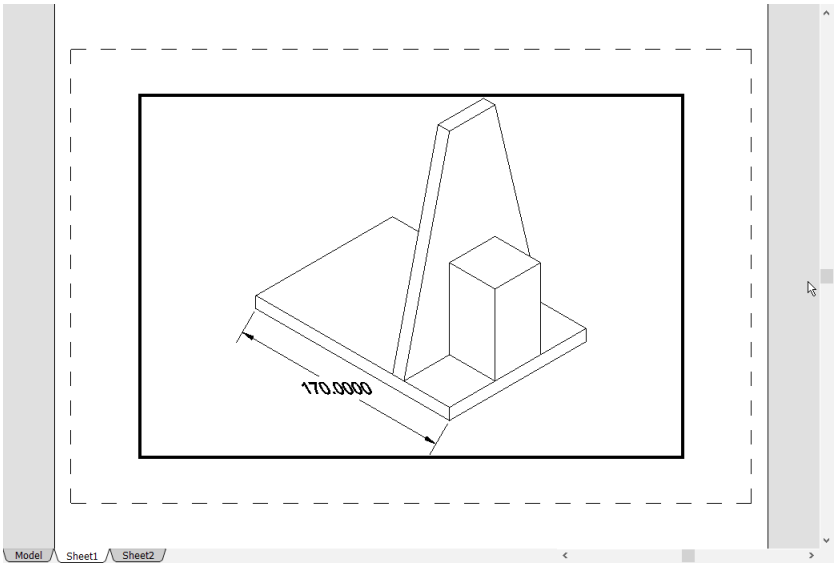
» Under Text, the **Text Height** is 0.18 units.

This means that the program has multiplied 0.18 by 50 to make the text a legible 9 units tall!



Properties palette reporting the model height of annotatively-scaled text

12. The dimension is now scaled appropriately for the viewport. To see this, switch to **Sheet1** to see that the dimension text and arrowheads appear at the correct size in the viewport.



Dimension correctly scaled using annotative scaling

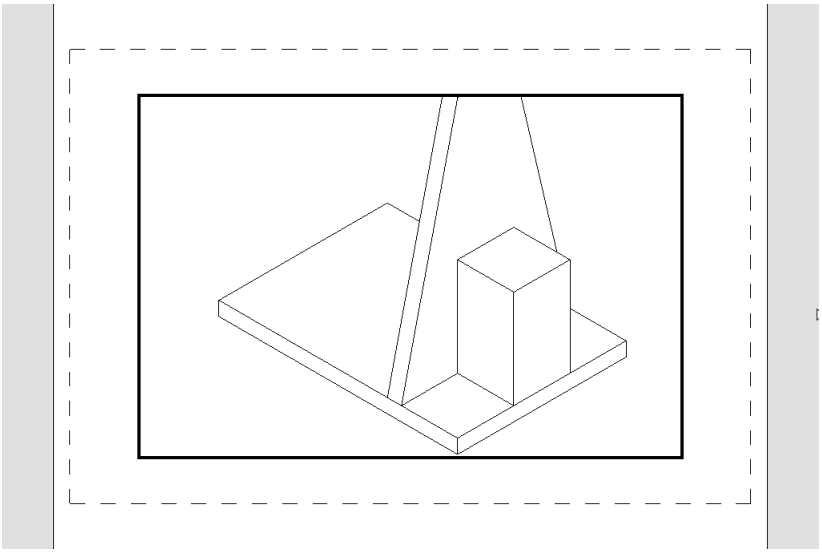
13. Pass the cursor over the dimension. Notice the **A** alerting it is an annotatively scaled entity.

Second Scale, Second Sheet

Should you need to plot the drawing at another scale, then I recommend you set up a second sheet at the new annotation scale. But you can have multiple annotative scale factors in a single sheet.

Recall that we only see annotative entities when their scale factor matches that of the viewport. Let's see an example of a disappearing dimension.

1. To see the effect of non-matching annotation scales, change the scale of the viewport to something like **1:40**. (On the status bar, click **Annotation**, and then choose 1:40.) Notice that the dimension disappears. You are seeing the automation of annotative scaling.

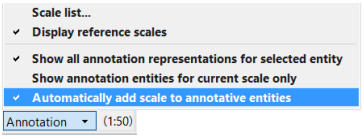


Annotatively-scaled text disappearing after the viewport scale changes

2. Let's make the dimension visible at the 1:40 scale. Follow these steps:

   a. Switch to **Model** tab.

   b. On the status bar, click the **Annotation** button, and then turn on the **Automatically Add Scale to Annotative Entities**. This causes the program to add annotative scales to annotative entities as you choose them from the list of scale factors.



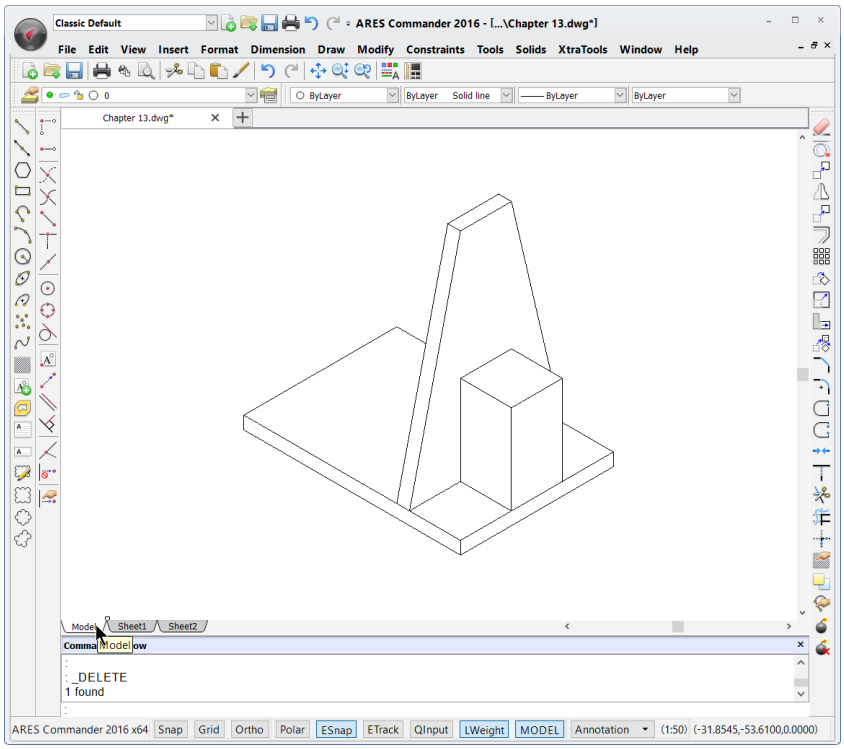
Turning on automatically adding annotation scale factors

   c. Click **Annotation** again, and this time choose **1:40**. Notice that the dimension text becomes smaller as the program adjusts it automatically.

3. While you are in Model tab, pass the cursor over the dimension. Notice the double-**A** icon alerting it is an entity with two or more annotation scales.

Double-A icon indicating the entity has more than one annotative scale applied

4. Click **Sheet1** tab and notice that the dimension has reappeared!



Dimension reappearing in Sheet1 tab

Now, technically you should apply 1:40 to the model in the Sheet2 tab, but by setting it to 1:50 I show you how the program shows and hides annotative entities automatically, as the scales change.

ADDING MORE VIEWPORTS

Command: Viewport
Aliases: mv, mview
Menubar: --
Toolbar: --
Ribbon: Sheet | View Tiles

When you enter sheet mode for the first time, the program creates one viewport automatically. You create additional ones with the **Viewport** command, as follows:

1. Enter the **Viewport** command using one of the methods listed above.

   ```
   : viewport
   ```

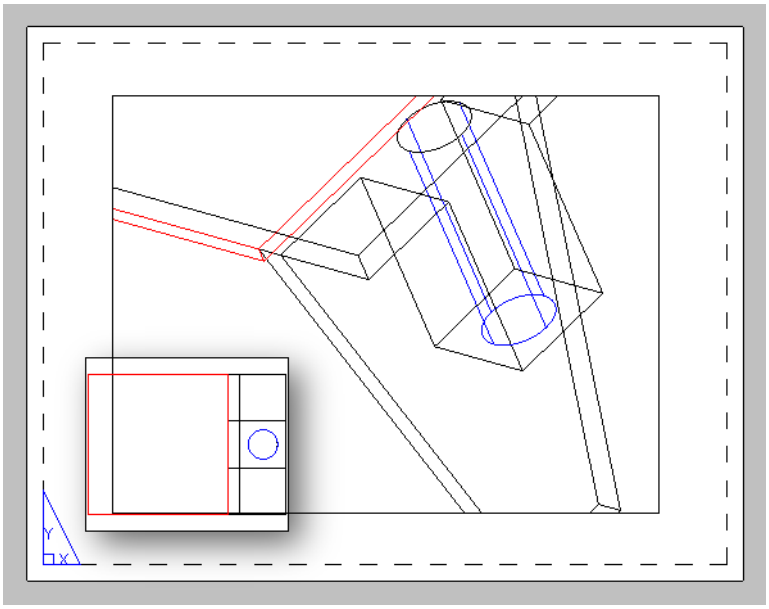
2. Pick a point on the sheet:

   ```
   Default: Fit
   Options: 2, 3, 4, Align Entity, Fit, Lock, OFf, ON, Polygonal, Restore, SHaded
   view or Specify start corner» (Pick a point)
   ```

3. Pick a second point.

   ```
   Specify second corner» (Pick another point)
   ```

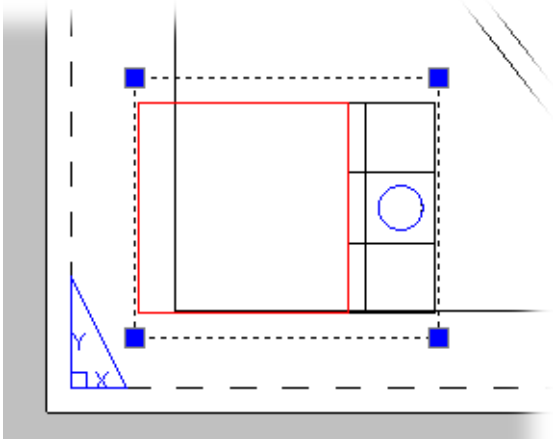
   Notice that the model appears inside the rectangle formed by the two pick points (shown highlighted by the figure below).



Second viewport added, showing a different view of the 3D model

4. Viewports are entities, and so you can edit their boundaries, such as resizing, copying, and deleting them. Here is how to do these modifications:

a. Select a viewport. Notice that it reacts like other entities: the rectangular board is highlighted (dashed) and grips appear.



Selecting a viewport in order to edit it

b. Grab the boundary with the cursor, and then drag it. Notice that it moves around the sheet.

c. You can resize the rectangular boundary by dragging a grip.

d. Here are some editing commands to try out on the viewport:

- To make a copy of a viewport, use the **Copy** command.
- To delete the viewport, use the **Erase** command.
- To bring back an erased viewport, use the **U** command.

SUMMARY OF VIEWPORT COMMAND OPTIONS

```
: viewport
Default: Fit
Options: 2, 3, 4, Align, Entity, Fit, Lock, OFf, ON, Polygonal, Restore,
SHaded view or
Specify start corner» (Enter an option, or pick a point)
Specify second corner» (Pick another point)
```

2, 3, 4 specifies the number of tiled viewports; after they are created, you can move and resize them.

Align aligns entities in separate viewports.

Entity converts the selected entity into a clipping boundary.

Fit creates a single viewport that fits inside the margins.

Lock locks the viewport to prevent editing, moving, and so on.

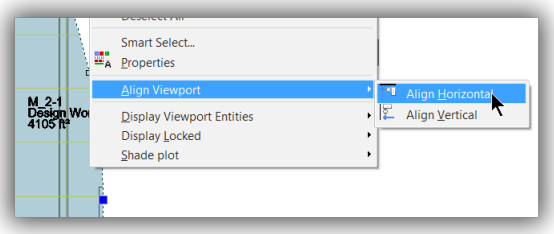
OFf, ON toggles the visibility of the contents of the selected viewport.

Polygonal specifies the vertices of a polygonal (non-rectangular) viewport.

Restore returns the viewport to its default settings.

SHaded view changes the display mode of the viewport, from wireframe to hidden-line removal or shaded.

5. You apply properties specific to the viewport through a shortcut menu. Select a viewport, and then right-click. Notice the shortcut menu, and that near the bottom there are options specific to viewports:



Viewport-related commands found in the shortcut menu

Align Viewport aligns the content of viewports horizontally or vertically:

Display Viewport Entities toggles the display of the content of the viewport:

» When off, the viewport is blank and you see only its border
» When on, you see the contents of the viewport

Display Locked prevents and enables editing of the viewport:

» When on, you can edit the
» When off, the contents of the viewport can be edited

Shade Plot displays 3D content of the viewport in one of several shaded modes:

» **As Displayed** displays the contents of the viewport in their default mode. For instance, if the model was in hidden-line mode before you created the viewport, it displays now in hidden-line mode
» **Wireframe** displays the content of the viewport in wireframe mode (no hidden-line removal or rendering)
» **Hidden** displays the contents in hidden-line mode. Entities hidden by other entities are ot displayed
» **Rendered** displays the contents as a rendering

Attaching Multiple Drawings

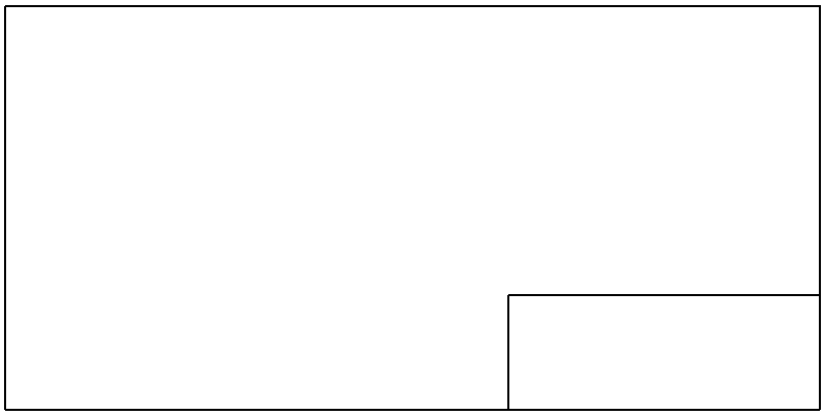
Command: References
Aliases: er, externalreferences, refs, xlink, xr, xref, image, im
Menu bar: Tools | References
Ribbon: Insert | References Manager
Toolbar:

In the previous tutorials of this chapter, you learned how to create multiple views of the same drawing. But the program also lets you view more than one drawing at the same time. Through the **References** command, you can attach one or more drawings to be displayed at the same time.

To see how this is accomplished, you first create a simple drawing border in the following tutorial, and then attach it to the 3d model drawing:

1. Start a new drawing with the **New** command.

2. Use the **Rectangle** command to draw a simple drawing border and title block, as illustrated below. (The size does not matter, since the drawing will be scaled later.)



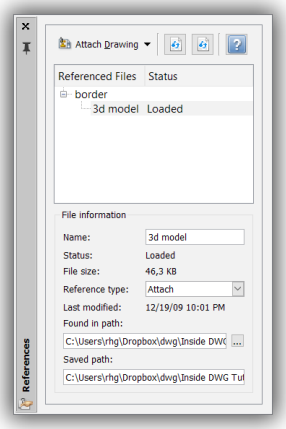
Simple title block and drawing border

3. With the **Save** command, name the drawing as *border.dwg*, and then exit the drawing with the **Close** command.
4. Back in the *3d model.dwg* drawing, collapse the four viewports into one with the **-Viewport** command, as follows:

```
: -viewport
Default: 3
Options: ? to list, 2, 3, 4, Delete, Join, Restore, Save, or SIngle
Specify option» si
```

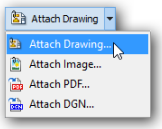
(You can also restore a single view tile in model space with the ViewTiles command.)

5. Start the **References** command using one of the methods listed above. Notice that the Reference palette appears.



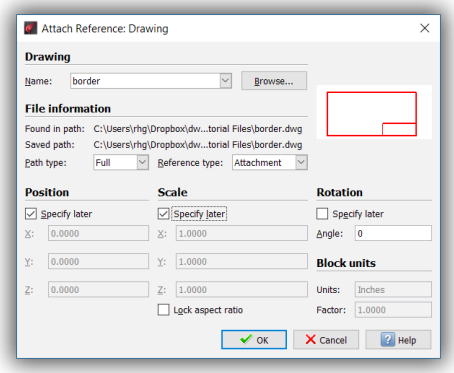
The References palette for controlling attached drawings

6. To attach an external drawing to this one, click **Attach Drawing**. (Notice that the same palette is used to attach raster images, PDF files, and DGN drawings.)



Choosing the file type to attach

7. When the Select File dialog box appears, choose *border.dwg* and then click **Open**.
8. Notice the Attach Reference Drawing dialog box. (It operates much the same way as the Attach Reference Image dialog box from Chapter 14.)



Dialog box for determining how files are attached to the drawing

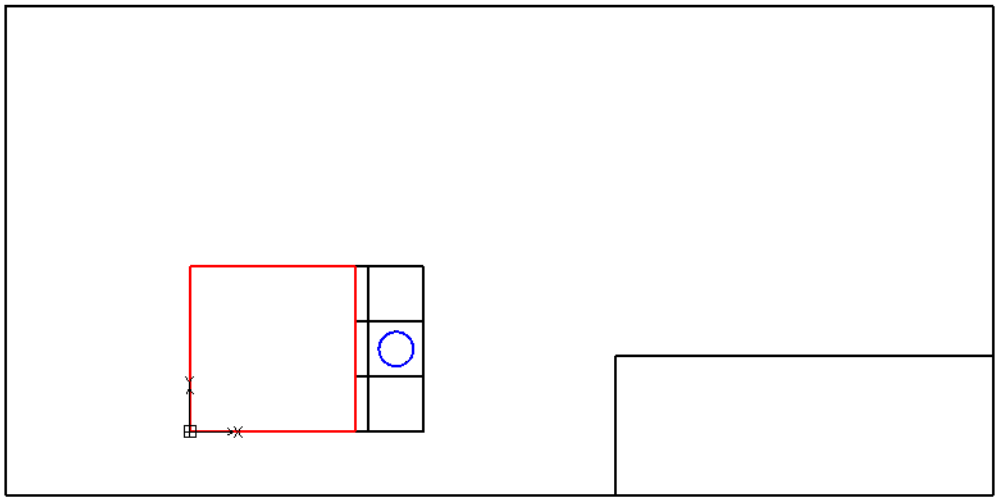
For this tutorial, make the following changes to the settings in the dialog box:

Dialog Box Option	Value
Insertion point	Specify Later
Scale	Specify Later
Rotation	0

9. Click **OK**, and then follow the prompts in the Command window:

```
Specify position» (Pick a point in the lower left corner of the drawing)
X scale or specify opposite corner» (Pick a second point in the upper right corner
of the drawing)
Default: 1.0000
Specify Y scale» (Press Enter)
```

Notice that the border drawing appears with the 3D model, as if the two were one drawing:



Border drawing attached to the 3D model drawing

Updating Referenced Drawings

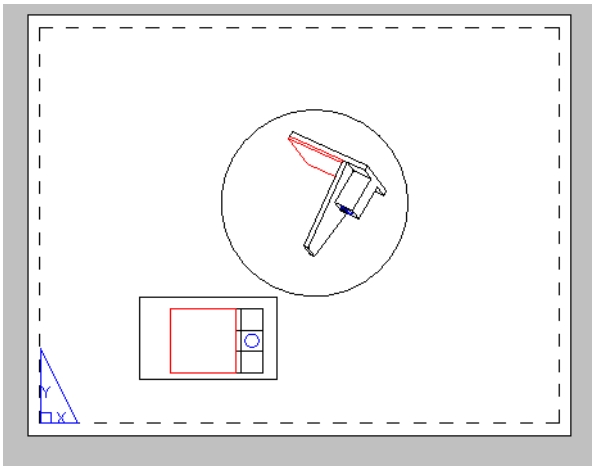
When it is attached, the *border.dwg* drawing acts like a block. Yet it remains independent of the *3d model* drawing. Another drafter can continue working on *border.dwg*, changing it. When you update the *border.dwg* in *3d model*, you will see the changes. Do this now, as follows:

1. Open *border.dwg*, and then change the color of its rectangles to red.

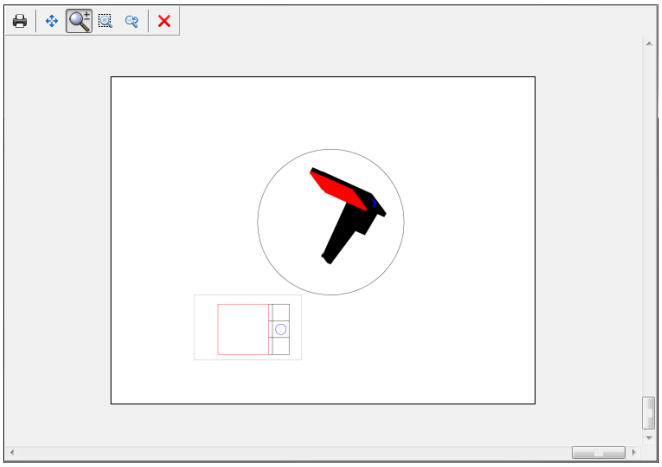
Changing the border drawing to red

TIP Viewport boundaries do not need to be rectangular; they can be any closed shape. To create a circular viewport, follow these steps:

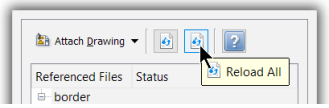
1. Draw a circle with the **Circle** command.
2. Start the **Viewport** command, and then choose the **Entity** option.
3. Choose the circle at the 'Select clipping entities' prompt.



When you print the drawing in sheet mode, you get what you see, as shown by the Print Preview window below.

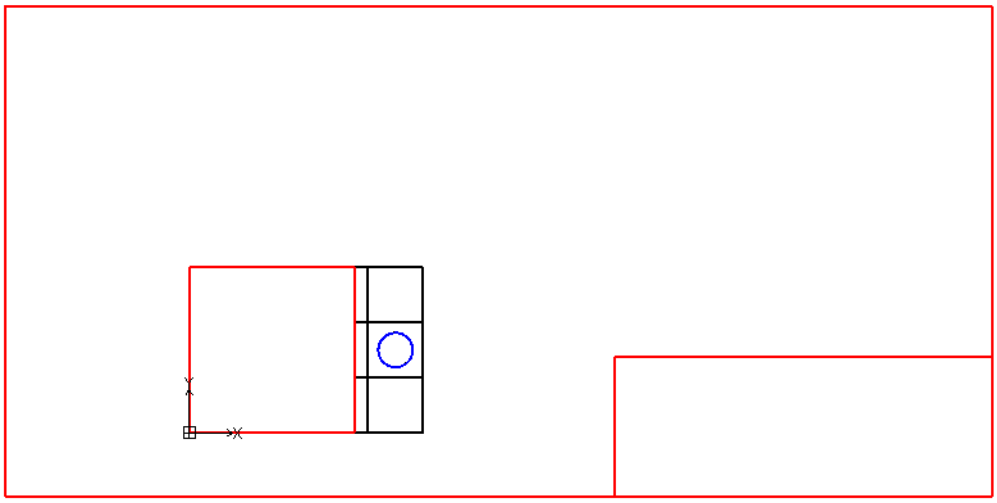


2. Save the drawing.
3. Press **Ctrl+Tab** (Cmd+Tab on Mac) to return to the *3d model* drawing.
4. In the References palette, select **border**, and then click **Reload All**.



Reloading the boarder drawing into the 3D model drawing

Notice that the drawing border turns red.



Border drawing appearing red as an attachment

5. You can attach drawings in sheet mode as well. Here's how to do this:
   a. Click **Sheet1**.
   b. Run the **References** command, and then place *border.dwg* on the sheet.

TIP You can apply certain editing commands to attached reference files, such as Move and Rotate.

Summary

You learned how to display 3D models in layouts in preparation for printing, and how to attach one drawing to another to created combined views.

The next chapter describes how to create parametric drawings using dimensional and geometric constraints.

Notes

CHAPTER 19

Dimensional & Geometric Constraints, and Parameters

In This Chapter

- Dimensional constraints
- Geometric constraints
- Parameters and equations

We use constraints and parameters for these kinds of drawing activities:

Turning sketches into drawings by changing badly drafted drawings into tidy ones

Relating parts one to one another by attaching entities to each other with permanent esnap-type connections

Generating configurations by making multiple versions of the same drawing, just with sizes changes

You learn about the two types of constraints — dimensional and geometric —, how to use them, and how to apply equations (parameters).

KEY TERMS IN THIS CHAPTER

Coincident constraint fixes two points on entities, or fixes a point on two entities

Collinear constraint fixes two segments (lines, etc) so that they lay on a straight line

Concentric constraint fixes two curves (circles, arcs, etc) so that they have the same center point

Constraint Bars are icons that identify which geometric constraints are applied to the selected entity

Constraints hold entities in place by geometric or dimensional means

Derived constraints use geometry to determine the value of dimensional constraints

Dimensional constraints determine the size of entities and the distances between them

Equal constraint fixes two entities (lines, arcs, circles, etc) so that they have the same length or radius

Expressions are formulas used by dimensional constraints

Fixed constraint locks geometry to a location in the drawing

Forms describe the look or style of dimensional constraints

Geometric constraints determine the position of entities and their relationships with other entities

Names identify dimensional constraints, such as d1 for distance #1 and rad2 for radius #2

Parameters combine names, expressions, and values to control dimensional constraints

Smooth constraint fixes splines to other spline, arcs, lines, or polyline fluidly

Symmetric constraint fixes two entities so in a mirror-like position

Values are numerical constants used by dimensional constraints and parameters

ABBREVIATORS

dc Dimensional constraint

gc Geometric constraint

d Distance

dia Diameter

rad Radius

COMMANDS

Command	Shortcuts	Menu Selection
DcSmart	...	...
DeleteConstraint	...	Constraints | Delete Constraints
DimensionalConstraint ...		Constraints | Dimensional Constraints
GeometricConstraint	...	Constraints | Geometric Constraints
Parameters ...		Constraints | Parameters Manager
ParametersHide	...	...

Types of Constraints

The program uses two types of constraints for parametric drawings:

- **Dimensional** constraints determine the size of entities and the distances between them
- **2D geometric** constraints determine the position of entities and their relationships with other entities

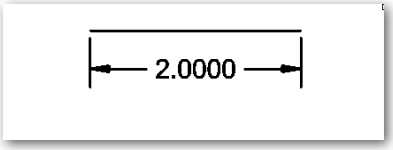
In summary, when you want to fix the **size** of an entity, use a *dimensional* constraint. When you want to fix the **position** of an entity, use a *geometric* constraint.

ABOUT DIMENSIONAL CONSTRAINTS

Command:	DimensionalConstraint
Menu bar:	Constraints | Dimensional Constraints
Ribbon:	Constraints | Dimensional
Shortcut:	...
Toolbar:	Dimensional Constraints

Dimensional constraints are very much like regular, associative dimensions, but work the other way around. Let me describe the difference to you:

Regular dimensions report the size of the entity to which they are attached. For example, dimension a 2" line and the linear dimension reports 2".



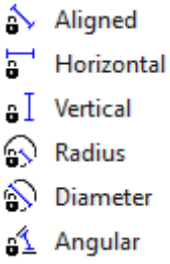
Dimensioning a 2-unit line

Stretch the line to 3", and the linear dimension updates and reports the new length of 3". You can edit the value of the linear dimension, but doing so does not change the length of the line.

Dimensional constraints work the opposite way. They do determine the length of the line. Change the value of the dimensional constraint, and the line changes its length. You cannot manually change the length of the line, because it is *constrained*! Dimensional constraints look like this:

Constraining a line to 2 units

The program comes with the following dimensional constraints, which closely mimic regular dimensions. Later on I'll describe them in detail.

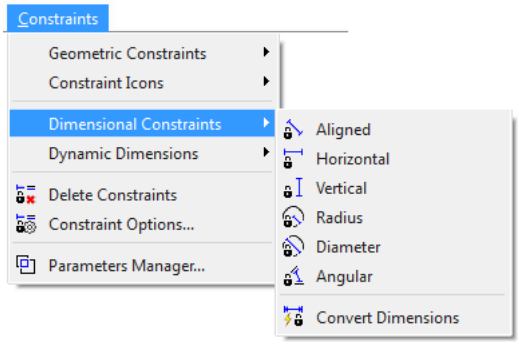


Dimensional constraints included with the program

SUMMARY OF DIMENSIONAL CONSTRAINTS

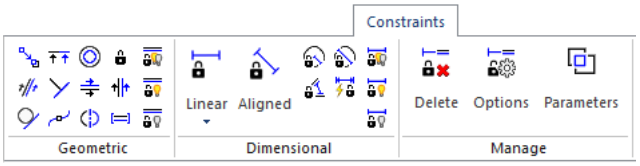
MENU BAR

Dimensional constraints can be accessed from **Constraints** on the menu bar:



RIBBON

From the **Constraints** tab on the ribbon:



COMMAND PROMPT

Entered as commands at the prompt:

DcAligned constrains the distance between two entities

DcAngular constrains:

- Angle between three points on one entity
- Angle between two lines or polyline segments
- Total angle of an arc or curved polyline segment

DcDiameter constrains the diameter of a circle, arc, or curved polyline segment

DcHorizontal constrains the horizontal distance between two points on an entitt or between entities

DcLinear constrains the horizontal or vertical distance between two defining points on or between entities

DcRadial constrains the radius of a circle, arc, or curved polyline segment

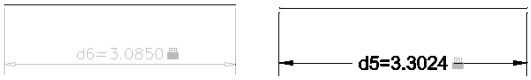
DcVertical constrains the vertical distance between two defining points on entities

Or entered as options to the **DimensionalConstraint** command:

```
: dimensionalconstraint
Default: Aligned
Options: Aligned, ANgular, Diameter, Horizontal, LInear, Radial, Vertical,
Convert or Type
```

CONTROLLING DIMENSIONAL CONSTRAINTS

CConstraintForm system variable toggles the current look ("form") of dimensional constraints. The look can be switched between dynamic and annotational forms. Set this system variable to 1 just before plotting drawings.



***Left**: Dynamic constraint; **right**: annotational constraint*

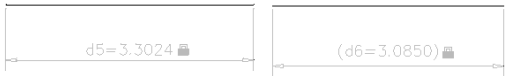
Value	CConstraintForm
0	**Dynamic** (default): dimensional constraints ignore dimension style and view scale
1	**Annotational**: constraints look like regular dimensions, and take on the current dimension style and view scale

ConstraintNameFormat system variable determines the format of dimensional constraint text:

***Left**: Value; **center**: name; **right**: expression (name and value)*

Value	ConstraintNameFormat	Example
0	Name only	d1
1	Value only	4.321
2	Expression (default)	d1 = 4.321

DcDerived (undocumented) command determines if the constraint or the geometry controls the other. Normally the dimensional constraint controls the geometry, but this command lets you reverse is so that the geometry controls the value of the constraint, like with regular dimensions. Derived constraints show brackets around the text. This command is applied to selected entities.



***Left**: Not-derived constraint; **right**: derived constraint showing parentheses*

DcDerived	Meaning
Not derived (Default)	Dimensional constraint controls the size of the geometry
Derived	Geometry controls the dimensional constraint and the constraint cannot be edited
Inverse	Reverses the not-derived/derived setting of the selected entity(ies)

TIP To erase a dimensional constraint, select it and then press the **Del** key. To erase geometric constraints, you can use only the **DelConstraint** command.

Continued on next page...

DcDisplay command toggles the display of dynamic dimensional constraints; this command has no effect on annotational constraints or regular dimensions.

DcDisplay	Meaning
Show	Shows the dimensional constraints of the selected entities.
Showall	*(Hidden option)* Displays all dimensional constraints
Hide	Hides the dimensional constraints icons of the selected entities
Hideall	*(Hidden option)* Hides all dimensional constraints

DimConstraintIcon system variable determines when the padlock icon is displayed:

***Left**: Padlock icon being shown; **right**: off*

Value	DimConstraintIcon
0	**Off**: Padlock icon not displayed
1	**Dynamic**: Icon displayed with dynamic constraints
2	**Annotational**: Icon displayed with annotational constraints
3	**On** (default): Icon displayed always

DynConstraintMode system variable toggles the display of hidden dimensional constraints when related entities are selected.

Value	DynConstraintMode
0	**Off**: Dimensional constraints are kept hidden
1	**On**: Hidden constraints are displayed

ParameterCopyMode system variable determines how dimensional constraints and their variables are copied between drawings, block definitions, and model and layout sheets:

Values	ParameterCopyMode
0	None: Dimensional constraints and variables are not copied to target entities
1	(Default) Copies dimensional constraints and variables to target entities; expressions are replaced by numerical values and naming conflicts are resolved
2	Copies dimensional constraints, variables, and user-defined variables to target entities; user-defined variables are retained when possible, otherwise expressions are replaced with numerical values
3	Like 2, but changes missing referenced dimensional constraints into user variables
4	Copies dimensional constraints, variables, and expressions to target entities; renames variables automatically when conflicts occur

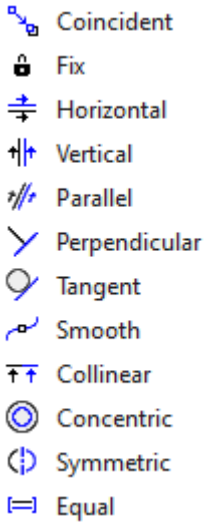
ConstraintRelax and ConstraintSolveMode system variables are not implemented in this release; they are provided only for compatibility with add-on programs.

TIP The program lets you convert regular dimensions into dimensional constraints. This lets you modernize old drawings!

ABOUT GEOMETRIC CONSTRAINTS

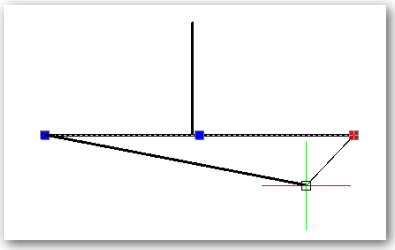
Command: GeometricConstraint
Menu bar: Constraints | Geometric Constraints
Ribbon: Constraints | Geometric
Shortcut: ...
Toolbar: Geometric Constraints

The program comes with the following geometric constraints, which in some cases mimic entity snaps. I'll describe all of them in detail later on. (There also are 3D constraints, which a few other CAD programs support. These are used especially for complex 3D models by combining several parts into a single assembly.)



Dimensional constraints provided by the program

When you use an entity snap, such as perpendicular, to draw one line perfectly perpendicular to another, the relationship ends once you finish placing the second line. The entity snap is **temporary**; you can then make the two lines oblique or even parallel to each other, as shown below.



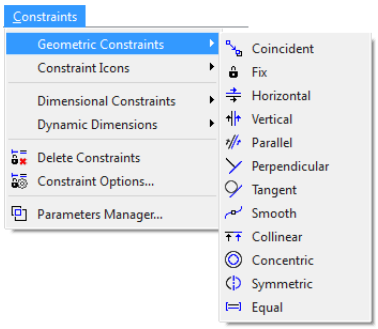
Using entity snap to make one line perpendicular to another

Geometric constraints are like **permanent** entity snaps. When you apply the perpendicular geometric constraint to the two lines, they remain perpendicular for all time. (Well, by "for all time" I mean for as

SUMMARY OF GEOMETRIC CONSTRAINTS

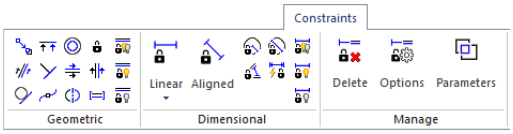
MENU BAR

Geometric constraints can be accessed from **Constraints** on the menu bar:



RIBBON

From the **Constraints** tab on the ribbon:



CIOMMAND PROMPT

Entered as commands at the prompt:

Coincident constrains two points on entities (or constrains a point on an entity to an entity)

Fix constrains entities to fixed positions

Horizontal constrains lines and linear polyline segments (or pairs of points on entities) to be horizontal (parallel to the x-axis of the current coordinate system)

Vertical constrains lines and linear polyline segments (or pairs of points on entities) to be vertical (parallel to the y-axis of the current coordinate system)

Parallel constrains two lines or linear polyline segments to be parallel to each other

Perpendicular constrains two lines or linear polyline segments to be perpendicular to each other

Tangent constrains one entity to be tangent to another

Smooth constrains splines to other spline, arcs, lines, or polyline fluidly

Collinear constrains lines collinearly

Concentric constrains center points of arcs, circles, ellipses, or elliptical arcs concentrically

Symmetric constrains two entities (or points on entities) symmetrically

Equal constrains lines to the same length, or arcs and circles to the same radius

Or entered as options to the **GeometricConstraint** command:

```
: geometricconstraint
Default: Coincident
Options: Coincident, COLlinear, CONcentric, Equal, Fix, Horizontal, PAral-
lel, Perpendicular, SMooth, Symmetric, Tangent or Vertical
```

CONTROLLING GEOMETRIC CONSTRAINTS

ConstraintBarDisplay system variable determines how geometric constraint icons ("bars") appear:

Value	ConstraintBarDisplay
0	**Off**: Keeps geometric constraint icons hidden but is overridden by the ConstraintIcon command's ShowAll option
1	**During applying**: Displays icons when you apply geometrical constraints to entities
2	**During editing**: Icons not displayed as constraint applied; appear when entity is edited
3	**Full time** (default): Displays the icon after you apply geometrical constraints to specified entities and when you select constrained entities

ConstraintBarMode system variable determines which geometric constraints display icons. This allows you to suppress the display of ones unimportant to you. The determination is made by adding up the bitcode values listed below for the ones you want displayed; a value of 4095 displays all of them, and is the default.

Bitcode	Geometric Constraint Icon
1	Horizontal
2	Vertical
4	Perpendicular
8	Parallel
16	Tangent
32	Smooth
64	Coincident
128	Concentric
256	Collinear
512	Symmetric
1024	Equal
2048	Fixed

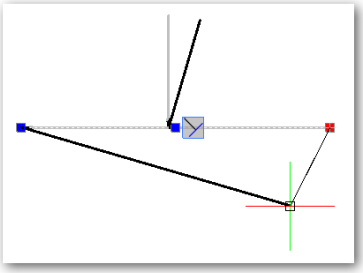
ConstraintIcon command toggles the display of geometric constraint icons:

ConstraintIcon	Meaning
Show	Shows the constraint icons of the selected entity(ies)
Showall	*(Hidden option)* displays all icons
Hide	Hides the constraint icons of the selected entity(ies)
Hideall	*(Hidden option)* hides all icons
Reset	Relocates the icons of the selected entity(ies) to their default position
Resetall	*(Hidden option)* returns all icons to their default location

DelConstraint command removes all dimensional and geometric constraints from the entities you select. To remove individual geometric constraints from an entity, right-click its icon and then choose **Delete**.

TIP The regular **Delete** command removes dimensional constraints from the drawing, but not geometric ones. When you delete an entity, however, then all attached constraints are also erased.

long the constraint is attached; constraints can be erased.) When you move or rotate one line, the other one moves with it, as illustrated below.

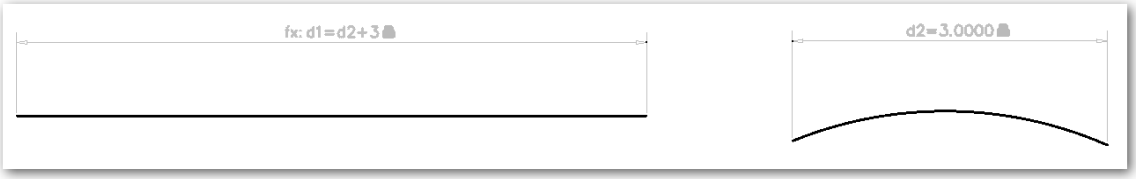


Applying a perpendicular geomtric constraint

ABOUT PARAMETERS

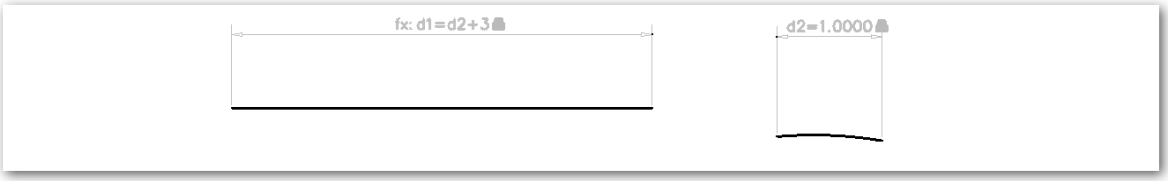
There is a third aspect to constraints known as *parameters* (or "parametrics" or "expressions"), which apply only to dimensional constraints. Instead of just distances (such as the 2" line), we can use a formula, such as **d1 = d2 + 3**, to determine the length of the line. This formula means that the length of the line (represented by d1) is the length of another entity, like a arc (d2), plus 3".

The "fx" indicates that the length of the line is determined by a formula. The arc (d2) controls the line (d1); the line does not control the arc.



A formulat (fx) defining the lenth of the line (d1) based on the length of the arc (d2)

When we change the length of the arc, the line is forced to change its length in synch. The line will always 3" longer than the polyline. Below, the 1"-long arc forces the line to shorten to 4" long.



Short line resulting from shorter arc

The program uses an automatic naming system for variables. You can, however, replace the names with own ones. Here are the names that are given automatically:

Constraint Command	Default Variable Names	Applies To
DcHorizontal, DcVertical, DcAligned	d1, d2, *etc*	d = Linear distances
DcDiameter	dia1, dia2, *etc*	dia = Diameters
DcRadial	rad1, rad2, *etc*	rad = Radii
DcAngular	ang1, ang2, *etc*	ang = Angles
Parameter	user1, user2, *etc*	user = User-defined expressions

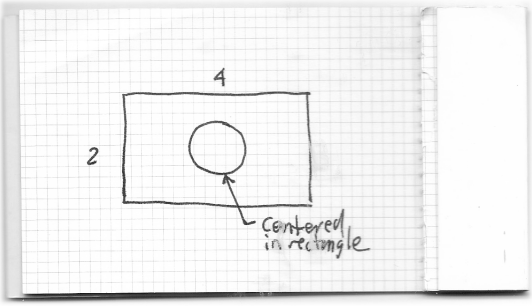
Parameters apply only to dimensional constraints; geometric constraints have nothing like parameters. Here is the pecking order:

Geometric constraints control parameters, dimensional constraints, and entity *positions*

Parameters control dimensional constraints, entity *sizes*, and distances between entities

Dimensional constraints control entity *sizes* and distances between entities

This chapter's tutorial uses all three in designing a spacer plate made of a 2x4 rectangle with a centered circle.

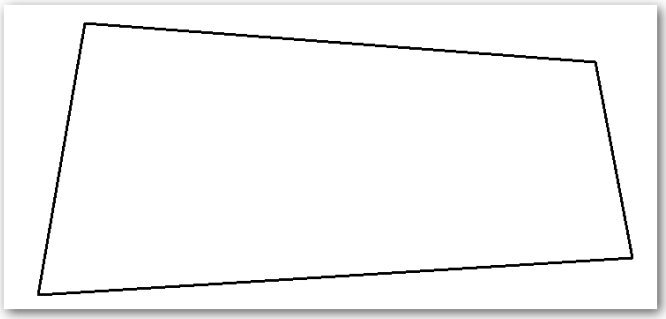


Sketch of tutorial drawing for this chapter

Sketches to Drawings

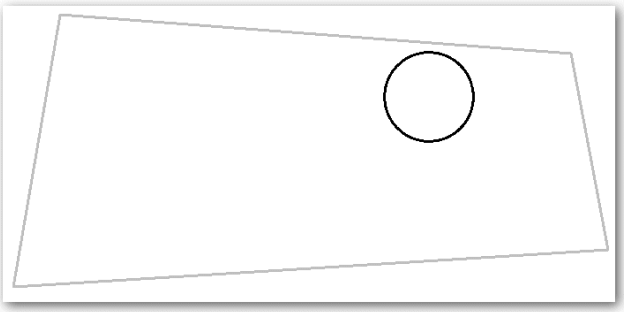
One way to use constraints is to straighten out sloppy sketches. This is how it works:

1. With the **Polyline** command, draw the rectangle without regard to neatness. The result can look something as badly drawn as this.



Irregular polygon

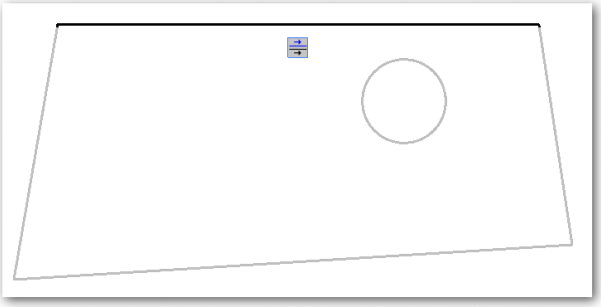
2. Inside the quadrilateral, draw a circle of any size at no location in particular. The tutorial will be easier for you when you ensure the circle is completely inside the polygon.



...and a circle

3. Now you learn how to use geometric constraints to straighten out the quadrilateral (no parallel sides) into a proper rectangle (all corners are at 90 degrees). The order in which you apply the constraints doesn't really matter. I'll start by forcing the top segment to be perfectly horizontal with the **GcHorizontal** command, like this:

```
: gchorizontal
Options: 2Points or
Specify entity» (Pick the top segment of the rectangle)
```

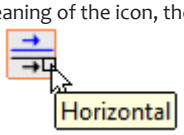


Applying a horizontal geometric constraint

Notice that the top line becomes perfectly horizontal. The two connected segments (left and right) follow along, because they are part of the polyline. This is why I had you construct the rectangle from a polyline; if you had used lines, you would have to add constraints to keep the lines connected.

Notice also the  icon near the straightened-out segment. It alerts you that this segment has the horizontal constraint applied.

TIP If you cannot remember the meaning of the icon, then pause the cursor over it to reveal the tooltip:

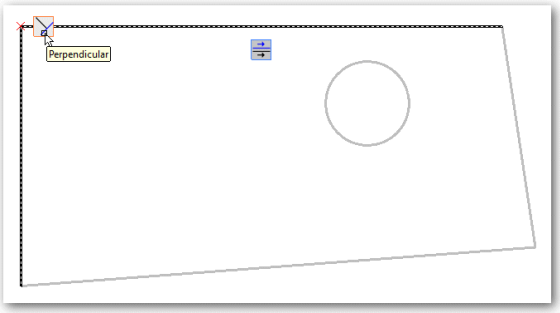


---

4. The polygon is supposed to be a rectangle with perpendicular corners. Use the **Perpendicular** constraint to straighten out the two sides by making them perpendicular to the top line.

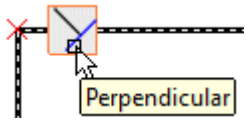
```
: gcperpendicular
Specify first entity» (Pick the top segment)
Specify second entity» (Pick a side segment)
```

The pick order is important! Although you don't know it from the command's prompts, the first entity you pick is the *governing* entity: it controls the second entity you pick. So, the second entity will be forced to be perpendicular to the first. The result looks like this:



Adding a perpendicular geometric constraint

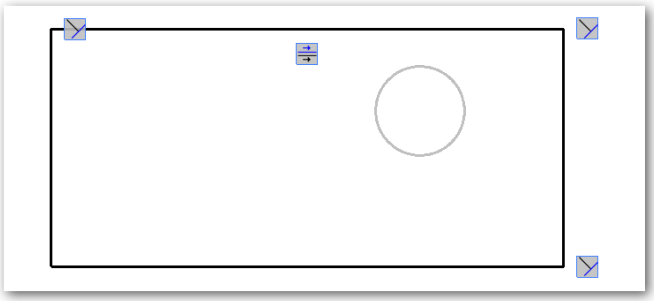
5. This time when you pause the cursor over the Perpendicular icon, notice the changes that occur to the polyline:
   - » The two segments are highlighted to indicate that the perpendicular constraint applies to them
   - » A red X appears at the exact location of the constraint



X marks the spot

(There is no similar X on the horizontal constraint, because there is no single place at which the segment is fixed to be horizontal.)

6. Repeat twice more, applying the Perpendicular constraint to two more corners. The polygon is now a proper rectangle.



Additional perpendicular constraints

TIP You need to constrain only three corners perpendicularly, not four. This due to the law of rectangles: the sum of the interior angles is always 360 degrees. When three corners are fixed at 90 degrees, the fourth one is forced to take on the remaining number of degrees: 360 - 90 - 90 - 90 = 90.

PART RELATIONSHIPS

The next step is to size the rectangle to the 2x4 dimensions. Here you switch from using geometric constraints to dimensional ones. Dimensional constraints set the sizes of entities, such as the lengths of polyline segments.

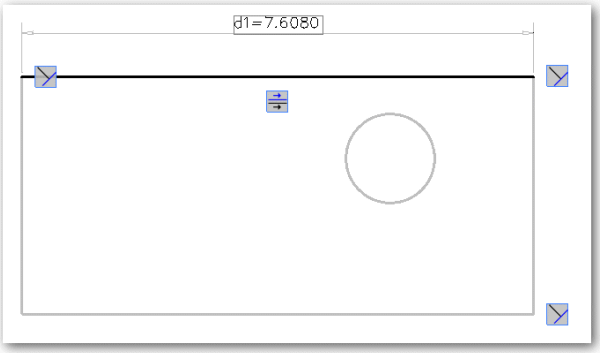
1. To specify the length of the horizontal segment at the top, start the **DcHorizontal** command:

```
: dchorizontal
Default: Entity
Options: Entity or
```

2. Use the **Entity** option to dimension the segment:

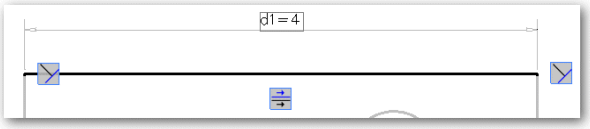
```
Specify first extension line position» e
Specify entity» (Pick the top segment)
Specify dimension line position» (Move the dimension line away from the segment)
```

3. Now you specify the length of the line. Notice that the dimension text reads something like **d1=7.6080** (or however it appears in your drawing). This means that the program determined the length of the segment is 7.6080 and assigned this value to parameter d1:



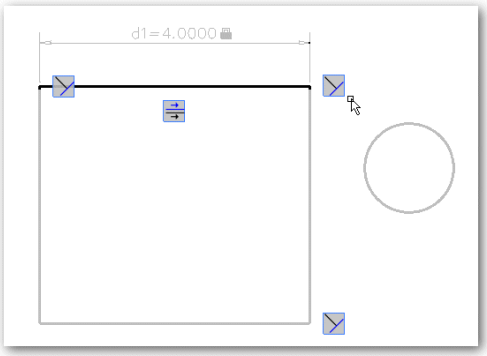
Specifying the length of d1

   a. Edit the "7.6080" value to read **4**:



Parameter d1 equalling 4

   b. And then press **Enter**. Notice that the rectangle shortens itself so that the top and bottom segments are 4 units long.



Segment shortened to 4 units

   As well, notice the padlock icon. It means that the length of the segment is locked to the value specified by the constraint.

   This is what the parts of the dimension text mean:

   d — the parameter is a distance (length)

   1 — this is the first distance parameter in the drawing

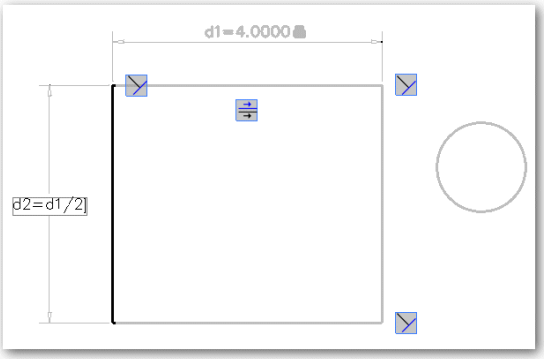
   4.0000 — value of the distance

   Padlock icon — the length of line is locked to the value of the constraint

4. The height of the rectangle is 2, half that of the width. Here you use a formula together with the Vertical constraint to ensure that this is always the case. Start the **DcVertical** command, and then specify the entity option:

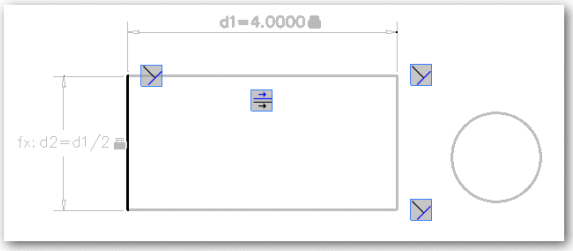
```
: dcvertical
Default: Entity
Options: Entity or
Specify first extension line position» e
Specify entity» (Pick the vertical segment)
Specify dimension line position» (Position the dimension)
```

This time, edit the dimension text to read **d2 = d1 / 2**. This forces the height always to be half of the width: distance d1 divided by 2.



Adding another formula

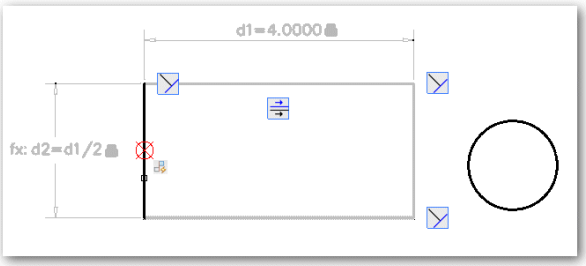
After you press **Enter**, the constraint looks like this, where the "fx:" prefix reminds you that the value of this constraint is a formula.



Formula fx in place

5. The final step is to center the circle inside the rectangle. There is no constraint for centering a circle in a rectangle, so you have to use dimensional constraints with formulas. Start the **DcHorizontal** command, but instead of selecting an entity, you pick two points. By picking a point on each entity, you specify the distance between them.

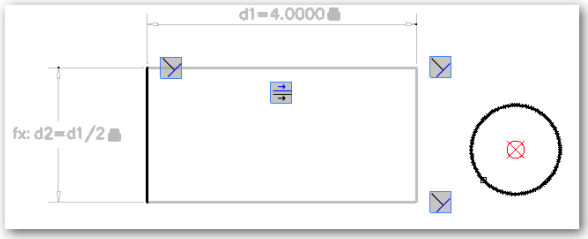
```
: dchorizontal
Specify first extension line position» (Pick the vertical segment, shown with the
red icon)
```



Red icon showing the constraint attachment location

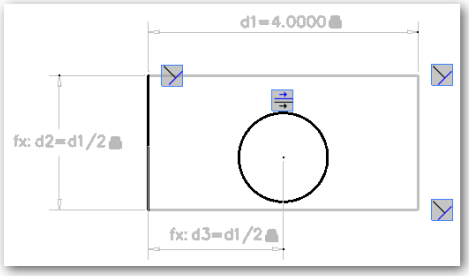
Notice the red ⊗ circle-x icon. It alerts you to where one end point of the horizontal constraint will be located. When you next pick the circle, another circle-x icon will appear:

```
Specify second extension line position» (Pick the circle)
```



Picking the circle

```
Specify dimension line position» (Place the dimension line)
Dimension Text: 5.4767
```

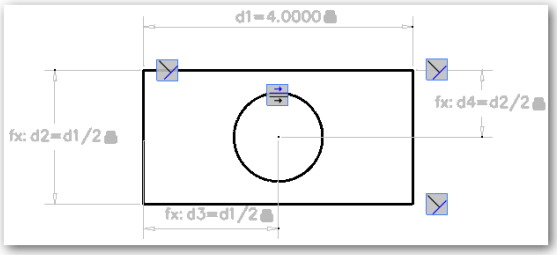


Placing the horizontal dimensional constraint

6. Edit the dimension text to read **d3=d1 / 2**, and then press **Enter**. This formula forces the circle to always be half way along the line — which is like centering it.
7. To position the circle vertically in the rectangle, use the **DcVertical** command and use the same formula:

```
: dcvertical
Default: Entity
Options: Entity or
Specify first extension line position» (Pick the horizontal segment)
Specify second extension line position» (Pick the circle)
Specify dimension line position» (Place the dimension line)
Dimension Text: 1.2185
```

Edit the vertical dimension's text to read **d4=d2 / 2**, and then press **Enter**. The result forces the circle to the center of the rectangle, like this:



Placing another formula

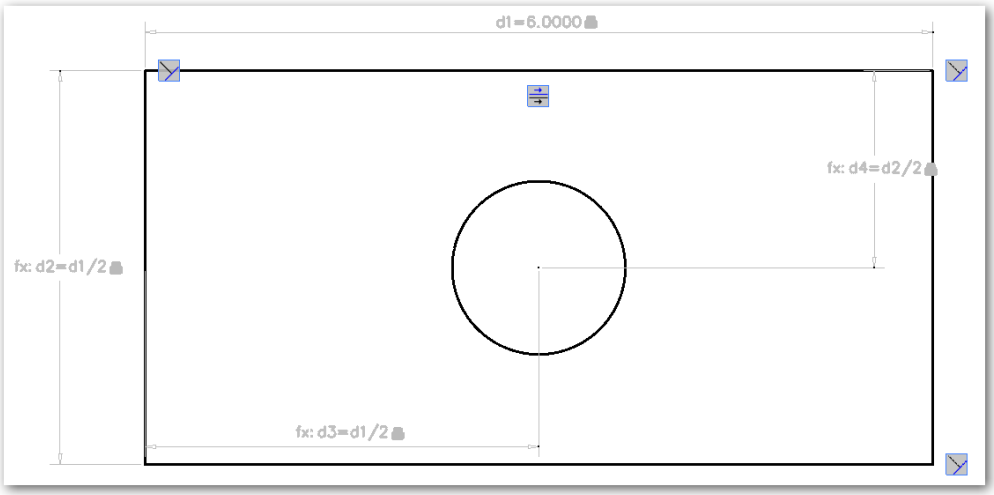
GENERATE CONFIGURATIONS

The design of the spacer plate is complete. It is the correct shape and size, with constraints fixing the pieces in place. What makes constraints exciting, however, is when they are used to change the *configuration* of the drawing.

Configurations allow you to generate variations on a single drawing. That 2x4 plates does not have to stay that size; when a customer asks for 3x6 plates, the drawing is quickly changed by editing the values of parameters.

This is how configurations work:

1. Double-click dimension **d1**, and then replace "4.0000" with **6**. Notice that the entire rectangle becomes larger, and that the circle stays centered.



Editing the dimensional constraint

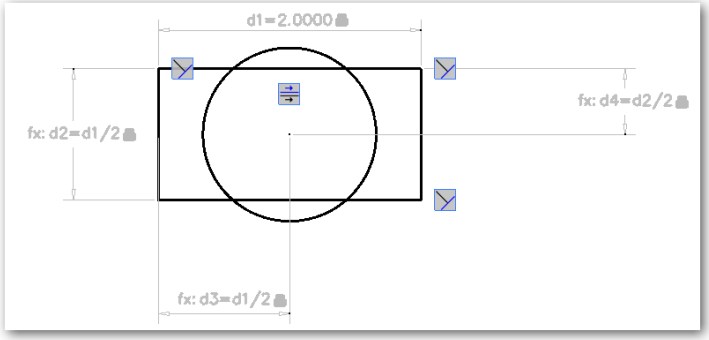
This change in size occurs uniformly because of the constraints you added to the drawing:

» Dimensional constraints are linked to dimension **d1**, and so the height of the rectangle and position of the circle also change in response to it

» Geometric constraints keep the corners perpendicular

The program performed all the calculations needed to resize the dependent entities automatically. There is no need for you to employ the Scale or Stretch commands!

2. Now make the rectangle smaller: change the "6.0000" to **2**. Oops! The circle is too large. The circle is supposed to define a hole in the spacer plate, but now the drawing is screwed up.

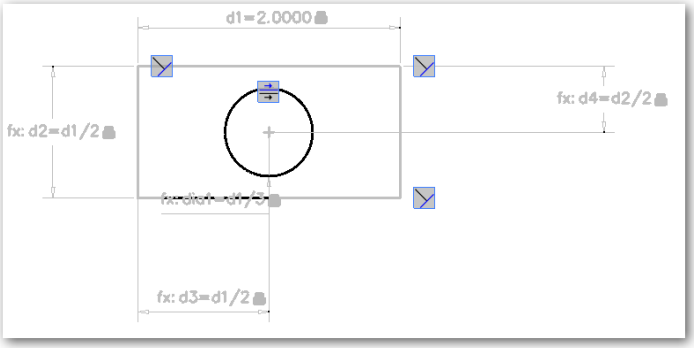


Oops!

3. No problem: you need to add a diameter constraint that resizes the circle smaller and bigger as the plate changes size. Use the **DcDiameter** command to add the diameter constraint, like this:

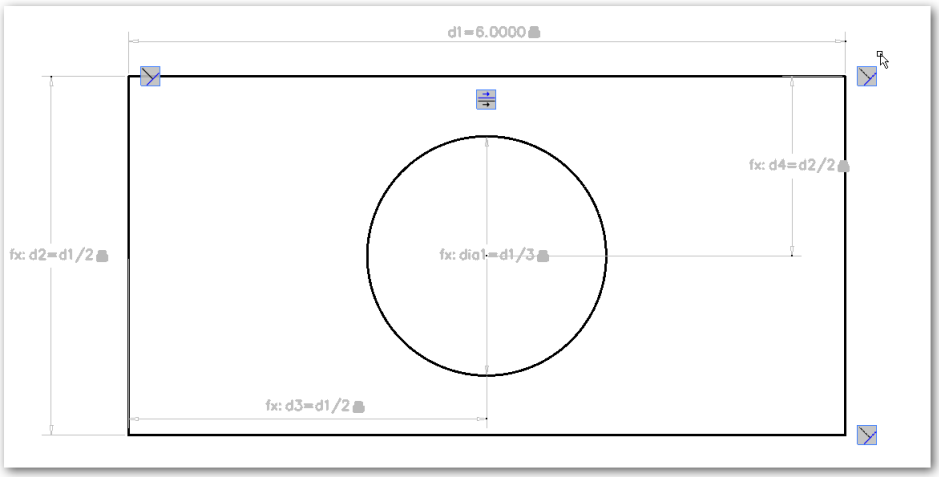
```
: dcdiameter
Specify curved entity» (Pick the circle)
Specify dimension position» (Position the constraint)
Dimension Text : 1.3188
```

Edit the dimension text to read **dia1=d1 / 3**. This ensures that the circle's diameter is always 1/3 the length of d1, the top segment.



Correcting the placement of the circle, part i

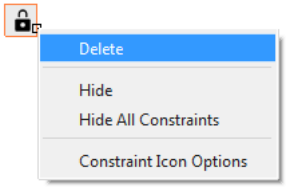
4. With the new constraint in place, make a change to see how the circle will resize itself, along with the polyline. Change **d1** to **6**, as follows:



Correcting the placement of the circle, part i

TIPS You can move geometric constraint icons by dragging them around the drawing. This lets you move them out of crowded drawing areas. Use the **ConstraintIcon** command's Resetall option to return icons to their original positions.

Right-click a geometric icon to see this shortcut menu:



- **Delete** erases the geometric constraint, removing it from the entity; you cannot use the program's regular Delete command on geometric constraints, although you can on dimensional constraints.
- **Hide** makes the selected icon invisible; to make icons visible again, use the **ConstraintIcon** command's Showall option.
- **Hide All Constraints** makes all icons invisible
- **Constraint Icon Options** displays Options dialog at Display Settings > Constraints.

APPLYING CONSTRAINTS SEMI-AUTOMATICALLY

The **DcSmart** command takes a best guess at which dimensional constraint to apply to individual entities. The command repeats itself so that you can reuse the command for many entities. There is no similar command for geometric constraints. The command applies certain dimensional constraints separately to linear and curved objects:

Entities	Dimensional Constraints Applied
Lines, Polyline segments	Angular, Aligned, Linear, Lock
Arc, Circle, Polyline arcs, Rings	Angular, Linear, Diametrial, Radial

Here is how to use the command

```
: dcsmart
Specify entity: (Select one entity only.)
```

If a line is selected, then the following prompt appears:

```
Options: Angular, Lock or specify dimension line location: (Enter an option or pick a point.)
Options: Undo or specify entity: (Pick another entity, or press Enter to exit.)
```

If a curve is selected, then the following prompt appears

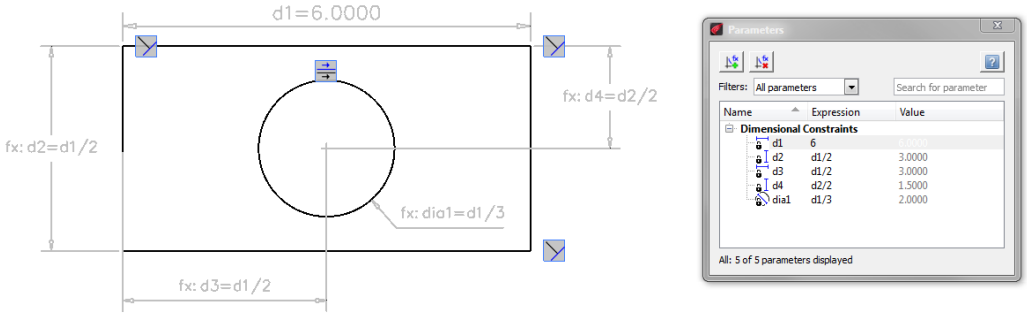
```
Options: Diameter, Radius, LInear, ANgular, or specify dimension line location: (Enter an option or pick a point.)
Options: Undo or specify entity: (Pick another entity, or press Enter to exit.)
```

USING THE PARAMETRICS PALETTE

Command: Parameters
Menu bar: Constraints | Parameters Manager
Ribbon: Constraints | Manage | Parameters
Shortcut: ...
Toolbar:

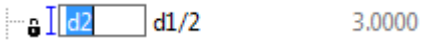
When you have a lot of expressions controlling entities in drawings, it can get confusing about where they are and to which entities they apply. To solve this, the program has the Parameters palette. You open it with the **Parameters** command

When drawings contain dimensional constraints and formulas, then you see all of them listed in the palette, as illustrated below. (This palette has no effect on geometric constraints.)



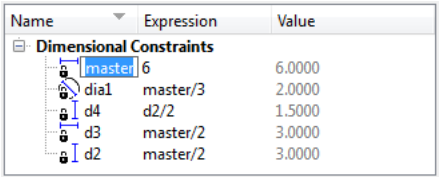
***Left**: Drawing with parametric equations; **right**: Parameters palette*

The primary purpose of the palette is to let you easily edit the names (like d2) and expressions (like d1/2) found in the drawing. To edit them, double-click a name or expression twice, so that it looks like this:



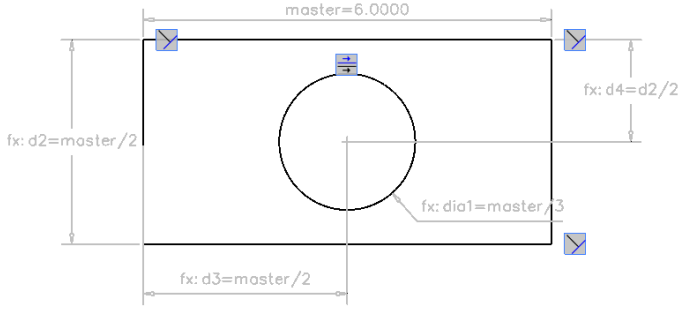
Editing a parameter

When you change the name of a constraint, the program updates all references to it elsewhere. For example, here I changed the name of d1 to master, and all references to d1 also changed:



Linked parameters updating

In the drawing, the parameter names are also changed to match:



Parametric drawing udpated

SUMMARY OF PARAMETRIC EXPRESSIONS

The following expressions can be used with parameters:

+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation
%	Floating point modulo
cos (expr)	Cosine
sin (expr)	Sine
tan (expr)	Tangent
acos (expr)	Arc cosine
asin (expr)	Arc sine
atan (expr)	Arc tangent
cosh (expr)	Hyperbolic cosine
sinh (expr)	Hyperbolic sine
tanh (expr)	Hyperbolic tangent
acosh (expr)	Arc hyperbolic cosine
asinh (expr)	Arc hyperbolic sine
atanh (expr)	Arc hyperbolic tangent
sqrt (expr)	Square root
sign (expr)	Signum function (-1,0,1)
abs (expr)	Absolute value
trunc (expr)	Truncate decimal
round (expr)	Round to nearest integer
floor (expr)	Round down
ceil (expr)	Round up
max (expr1;expr2)	Largest element in array
min (expr1;expr2)	Smallest element in array
d2r (expr)	Degrees to radians
r2d (expr)	Radians to degrees
ln (expr)	Logarithm, base e
log (expr)	Logarithm, base 10
exp (expr)	Exponent, base e
exp10 (expr)	Exponent, base 10
pow (expr1;expr2)	Power function

The following constants are available:

pi	3.14159
e	2.71828
random	Random decimal between 0 and 1

When you change an expression, the program modifies the constrained drawing to match. You can use all the expressions listed in the boxed text on the next page, "Summary of Parametric Expressions."

OTHER PARAMETRIC MANIPULATIONS

You have learned how to edit dimensional constraints directly in the drawing. (Geometric constraints cannot be manipulated.) There are a few other ways to adjust parameters. Let's take a look at them.

Grips Editing

When you click dimension d1, three grips show up:

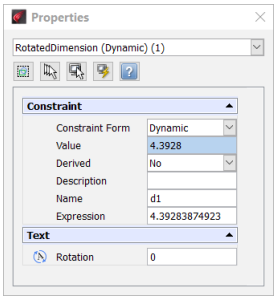
Grips on dimensional constraints

These grips let you modify the dimensional constraint (and associated geometry) interactively:

Grip	Dimensional Constraint	Associated Geometry
	Moves the constraint	Is unaffected
	Changes the length	Adjusts its size and position to the new length

Properties Bar

When you select a dimensional constraint, the Properties bar reports its properties:



Properties of dimensional constraints

You can change nearly all of the fields listed in it:

Property	Options	Meaning
Constraint Form	Dynamic Annotational	Displays constraint like a dimension constraint; ignores dimstyles Displays constraint like a regular dimension, including the current dimstyle
Value	*(read-only)* Value of the dimensional constraint; result of the expression	
Derived	Yes No	The geometry controls the dimensional constraint The dimensional constraint controls the geometry
Description		User-definable text
Name	d1 dia1 rad1 ang1	Distance Diameter Radius Angle
Expression		Expression or value expressed by the constraint
Text Rotation		Rotates the angle of the dimension text

Other Parameter Controls

These commands and system variable may prove useful when working with parameters.

ParametersHide command closes the Parameters palette.

ParametersStatus system variable reports the status of the Parameters palette: 0 = hidden, 1 = displayed

-Parameters creates and edits user-defined parameters at the command line. You can only enter the names of existing variables. Here is an example of how it is used:

```
: -parameters
Options: ? to list, Delete, Edit, New or Rename
Specify option» n
Specify name for new user parameter» asdf
Specify expression for new user parameter» d2*4
```

Summary

You learned how to use powerful tool that lets you set up relationships between entities and to specify sizes of entities. Geometric constraints fix parts of entities in place like permanent entity snaps. Dimensional constraints control sizes of entities and the distances between them. Parameters let you programmatically control the sizes and distances.

This concludes the *Inside ARES Commander 2019* book. The appendices that follow contain summaries of information that you may find helpful for working with the program.

Notes

PART V

Appendices

Notes

APPENDIX A

Alphabetical Command Summary

In This Appendix

- Brief descriptions of commands
- Command names listed in alphabetical order

The program works with 580 commands, as listed on the following pages in alphabetical order.

Some commands have a - (dash) prefix in front of their names: this indicates that the command displays its prompts at the command line; equivalent command names without the dash display instead a dialog box.

Commands new since the last edition of this book are shown in red.

The program works on several operating systems, and so not all commands are available on Linux or Mac OS X. Those commands not found on all operating systems shown in gray.

A

About displays information about the program.

-ActionManager manages macros.

ActionRecord records keystrokes as macros.

ActionStop stops recording keystrokes.

ActivateLayer activates the layer of a specified entity.

ActivateViewport activates the specified viewport in model or sheet mode.

Align aligns entities with other entities in 2D and 3D space.

Align3D aligns entities with other entities in 2D and 3D space by specifying a source and a destination plane.

AngleDimension draws angle dimensions.

AnnEntityScale and **-AnnEntityScale** add and remove annotation scales

AnnEntityScaleAdd updates annotatively scaled entities to the drawing's active scale.

AnnEntityScaleRemove updates annotatively scaled entities to remove the drawing's active annotation scale.

AnnReset resets all scale representations to the entity's original positions

AnnUpdate updates annotative scale factors to match updates to TextSyle and DimensionStyle.

Arc draws arcs.

ArcLengthDimension draws arc length dimensions.

AreaBoundary and **-AreaBoundary** draws polyline or region entities of enclosed areas.

ARender displays the current drawing rendered in a separate window.

AttachDgn attaches Microstation DGN design files as underlays.

AttachDrawing attaches referenced drawings to the current drawing.

AttachImage attaches images to the current drawing.

AttachLink attaches hyperlinks to entities or areas.

AttachPDF attaches Adobe PDF document files as underlays (Windows and Mac OS X only).

AutoCompleteOptions and **-AutoCompleteOptions** sets options for auto-completion of command names.

AutoRebuild controls automatic drawing rebuilds.

B

BaselineDimension continues linear, angular, or ordinate dimensions from the baseline of the previous or selected dimension.

BatchPrint sends a set of drawings and/or sheets to printers as a batch print job.

BlipMode enables and disables display of marker blips.

BlockAttributeManager manages properties and settings of block attributes.

BlockAttributeOutput writes block attribute values of selected blocks to text files.

Box draws three-dimensional solid boxes.

BoxText turns boxed text display and print mode on and off.

BrowseDesignResources specifies the default path for the Design Resources palette.

C

CBActivity adds activities to custom blocks.

CBActivityIcon toggles visibility of activity icons.

CBActivityTool creates and applies activities to custom blocks.

CBClose asks if you want to save the changes, and then exits the Custom Block Editor back to the drawing editor

CBElement creates a new Custom Block element.

CBElementSet creates a new Custom Block element and then associates and activity.

CBHide sets the visibility state to hidden.

CBOptions displays the Options dialog box at the for the Custom Block Editor section.

CBSave saves the Custom Block definition.

CBSaveAs saves the Custom Block definition by another name.

CBShow sets the visibility state to visible.

CBValueTable opens the Property Value Table dialog.

CBVisibility displays the Visibility States dialog box.

CCS sets, saves, and restores custom coordinate systems.

Centerline places center lines between pairs of lines, arcs, and polyline segments.

CenterMark draws center marks or center lines of circles and arcs.

Chamfer bevels entities.

ChamferEdges cuts edges from 3D solids.

ChangeElements adds and removes entities from blocks or references edited as components.

ChangeSpace transfers entities from Model workspace to Sheet workspace and vice versa.

Check evaluates the integrity of drawings and corrects errors.

Circle draws circles.

CircularPattern places copies of entities in circular patterns (Pattern command's Circular option).

Clean or **-Clean** remove unused named references from drawings, such as blocks and layers.

ClipboardCopy copies entities to the Clipboard.

ClipDgn clips DGN underlays using a boundary.

ClipImage defines clipping boundaries to suppress the display of portions of referenced images.

ClipPdf clips PDF underlays using a boundary (Windows and Mac OS X only).

ClipReference defines clipping boundaries to suppress display of portions of blocks or references.

ClipViewport clips the display of tiled views on sheets with non-rectangular shapes.

Close exits the current drawing, but not the program.

CloseAll exits all open drawings, but not the program.

CloseComponent closes a block or reference being edited as a component.

Cloud draws revision clouds.

CommandHistory opens the command history in a window.

CommandWindow displays the hidden command prompt area.

Comment opens the Comments palette, when you are signed into your Graebert account.

CommentsHide closes the palette.

Cone draws three-dimensional solid cones.

ConstraintIcon shows, hides, and resets geometric constraint icons.

ConstraintOptions displays a dialog box for choosing the constraints to inferred and geometry to seek.

ConstraintType toggles dimensional constraints between dynamic or annotation type; applies to subsequent placements of constraints.

ContinueDimension continues linear, angular, or ordinate dimensions from the second extension line of the previous or a selected dimension.

ConvertConstraint converts associative dimensions to dimensional constraints.

ConvertOle converts OLE objects into other formats (WINDOWS ONLY).

ConvertPrintStyles converts print styles between color-dependent and named.

Copy duplicates entities.

Copy@ copies entities with a specified reference point to the Clipboard.

CopyHistory copies the command history to the Clipboard.

CSIcon sets the visibility and position of the coordinate symbol (CS icon).

CSStyle manages custom coordinate systems (CCS).

Customize customizes user interface elements, such as menus, toolbars, and shortcuts.

CustomPaperSize adds and removes custom paper sizes for the program's "printer" drivers that output drawings in PDF, PNG, JPG, SVG, or DWF formats.

Cut copies entities to the Clipboard and deletes the entities.

Cylinder draws three-dimensional solid cylinders.

D

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.

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.

Delete removes entities from drawings.

DeleteConstraints removes all constraints from the selected entities.

DiscardDuplicates and **-DiscardDuplicates** remove duplicate and overlapping entities from the drawing.

DeleteLayer deletes entities and the layer of selected entities.

DesignResources displays the Design Resources palette for dragging blocks, referenced drawings, images, layers, linestyles, dimension styles, text styles, table styles, and sheets in the current drawing.

DetachDgn detaches DGN underlays from the current drawing.

DetachDrawing detaches referenced drawings.

DetachImage detaches referenced images attached to drawings.

DetachLink detaches hyperlinks that are attached to entities or areas.

DetachPdf detaches PDF underlays from the current drawing (WINDOWS AND MAC ONLY).

DgnUnderlayOptions specifies esnap and frame options for DGN underlays.

DiameterDimension draws diameter dimensions for circles and arcs.

Dim switches from command mode to dimension mode.

Dim1 executes one dimensioning command and returns to the command prompt.

DimensionalConstraint attaches dimensional constraints to entities, or between constraint points on entities.

DimensionStyle or **-DimensionStyle** creates and modifies dimension styles.

DirectionalLight inserts directional distant lights for renderings.

DisplayBlockAttributes defines block attribute visibility.

DisplayCoords toggles the display of coordinates on the status bar.

DisplayDialogs switches between dialog boxes and command prompts for commands asking for file names.

DisplayImageFrame determines whether frames are displayed at the borders of referenced images.

DisplayOrder changes the display order of entities and inserted images in the drawings.

DisplayQuality controls the resolution of circular and curved entities.

DraftingOptions sets drafting options, such as snap, grid, ortho, entity snaps, and entity tracking.

DraftingStyles manages text styles, dimension styles, richline styles, and table styles; sets styles for use with subsequent commands.

DrawingBounds sets and controls the limits of the drawing.

DrawingProperties displays a dialog box reporting the properties of the current drawing file; creates custom properties.

DrawingRecovery displays the Drawing Recovery Manager palette to let you recover drawings after unexpected system failures.

DesignResources displays the Design Resources palette for dragging blocks, referenced drawings, images, layers, linestyles, dimension styles, text styles, table styles, and sheets in the current drawing.

DrawingSettings sets the behavior of drawing commands, the appearance of certain drawing entities, the unit system of the current drawing, and manages Custom Coordinate Systems (CCSs).

DrawingStandards associates a *.dws* standards file to the current drawing.

E

EdgeMesh draws 3D polygon meshes based on four adjoining edges.

EditAnnotation edits items, such as single-line text, dimension text, and attribute definitions.

EditBasepoint edits entities of blocks or references opened as components.

EditBlockAttribute or **-EditBlockAttribute** modifies block attributes.

EditBlockAttributeDefinition modifies existing block attribute definitions.

EditComponent edits entities within blocks and references, in-place.

EditCustomBlock opens the Custom Block Editor environment.

EditDimension edits dimension text and extension lines.

EditDimensionText moves and rotates dimension text.

EditEntityGroup adds entities to (and removes them from) entity groups; also renames them.

EditHatch or **-EditHatch** modifies hatches and fills.

EditImage attaches and detaches images to and from the drawing, and manages image references.

EditIpBlockAttribute edit attributes of blocks in-place.

EditLength lengthens or shortens entities.

EditMultiLeader adds and removes leader lines.

EditNote edits multiline text.

EditPolyLine edits polylines, polyline vertices, and 3D polygon meshes.

EditRichLine or **-EditRichLine** edits richline (multiline) segments and intersections.

EditScaleList or -**EditScaleList** customize the list of scale factors used for printing drawings, managing page layouts, and scaling viewports in layout sheets.

EditSolid edits bodies, faces and edges of three-dimensional solid objects.

EditTable changes text in a table cell.

EditTableCell inserts, deletes, merges, and resizes cells, rows, and columns of tables.

EditTolerance modifies tolerances (datum indicators and basic dimension notation).

EditVertex edits 2D polyline vertices.

EditXBlockAttribute modifies attribute values and properties of inserted blocks.

Ellipse draws ellipses and elliptical arcs.

-EmbedDrawing embeds dependent objects and styles of referenced drawings in the drawing.

EnterPoint specifies the look and position of points.

EntityGrips sets preferences for editing with grips.

EntityGroup or **-EntityGroup** creates user-defined selection sets of entities.

EntitySnap or **-EntitySnap** sets running entity snap modes.

Exit exits the software.

Explode breaks complex objects into their component entities.

ExplodeBlockX explodes blocks and converts attributes to SimpleNotes; also explodes tables.

ExplodeText explodes text into simple entities such as polylines.

ExplodeX explodes ellipses and splines into simple entities.

Export saves entities in other file formats.

ExportBMP saves the drawing as a raster BMP file (device-independent Bitmap).

ExportDrawing or **-ExportDrawing** writes entities, blocks, or the entire drawing to a new drawing file.

ExportDXF exports the drawing as a vector DXF file (Drawing Interchange Format).

ExportEMF saves selected entities to an vector EMF file (Enhanced MetaFile; WINDOWS ONLY).

ExportEPS exports the drawing in Adobe EPS format (embedded PostScript).

ExportJPG saves selected entities to a raster JPEG file.

ExportPDF saves the current view of the graphics display to a PDF file (Portable Document Format).

ExportPNG saves selected entities to a raster PNG file (Portable Network Graphics).

ExportSAT exports solid modeling objects from the drawing to an ACIS *.sat* file (Save As Text).

ExportSLD exports the current view to a vector SLD file (slide).

ExportSLT exports solid modeling objects to an SLT file (Stereo LiThography) for 3D printing.

ExportSVG exports the current view to a vector SVG file (Scalable Vector Graphics).

ExportTable exports the content of tables in CSV (comma-separated format) files.

ExportWMF exports the current view to a vector WMF file (Windows metafile) (WINDOWS ONLY).

Extend extends entities to bounding edges defined by other entities.

ExtractBlockAttribute or -**ExtractBlockAttribute** extracts block attribute values of selected blocks to text files based on a template file defining the data to write.

Extrude extrudes entities as 3D solids or surfaces.

F

Face draws 3D faces.

Field inserts text that is updated automatically when system variables change.

FileLocations displays, sets, and edits default paths and/or file names.

FileManager opens the operating system's file manager, such as Windows Explorer or Finder.

FillArea fills areas with a solid color or color gradient

Fillet rounds entities.

FilletEdges fillets the edges of 3D solids.

Find finds and replaces text in notes, annotations, and dimension text.

Flip mirrors and deletes the original entity.

FlipArrows flips the direction of arrows in dimensions.

FreezeLayer freezes the layers of selected entities.

FullScreen displays the graphics area as a full screen (without toolbars and palettes).

G

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.

GcSymmetric constrains two entities, or two points on entities, to be symmetric about a line of symmetry.

GcSmooth constrains a spline to be fluidly continuous to another spline, or arc, or line, or polyline.

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.

Gesture customizes mouse gestures.

GetArea calculates the area and perimeter of entities or of defined areas.

GetDistance measures the distance and angle between two points.

GetMassProperties calculates and reports the mass properties of 3D solids and regions, such as their area and centroid.

GetProperties displays information about selected entities.

GetStatus displays status information about settings and memory usage of the drawing.

GetTime displays time-related statistics about the drawing and lets you toggle (on or off), or reset a timer.

GetXY displays the coordinate values of a location.

Gravity determines the size of the entity snap target box.

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

H

Hatch or **-Hatch** fills a selected boundary with a pattern.

Helix creates 2D spirals and 3D helixes.

Help displays online help.

HideCommandHistory closes the command history window.

HideCommandWindow hides the command prompt area.

HideDesignResources hides the Design Resources palette.

HideDrawingRecovery hides the Drawing Recovery Manager palette.

HideDrawingTabs hides drawing tabs.

HideEntities hides selected entities temporarily.

HideFullScreen returns from full screen display mode to normal display mode.

HideLayer turns off the layer of the selected entity.

HideLightlist hides the lighting palette.

HideParameters hides the Parameters palette.

HidePlugins hides the Plugins palette.

HideProperties hides the properties palette.

HideReferences hides the references palette.

HideToolMatrix hides the Tool Matrix palette.

HideView generates a hidden line view.

HorizontalDimension places horizontal dimensions.

Hyperlink or **-Hyperlink** adds hyperlinks to entities or modifies existing hyperlinks.

I

Import displays a dialog box for importing files into the drawing.

ImageMenu displays image menus.

ImageQuality determines the display quality of images attached to the drawing.

Import imports SAT files.

ImportDGN places Microstation V7 and V8 files in drawings.

ImportDWF inserts DWF files.

ImportDXF opens the selected DXF file in a separate drawing window.

ImportPDF inserts content from PDF files as blocks in the drawing; all or selected pages can be imported from multi-page PDF documents (Windows and Mac only).

ImportSAT reads an ACIS *.sat* file.

ImportSTL imports stereolithography files (*.stl*) and then inserts them into the current drawing as a 3D solid.

InfiniteLine draws infinite construction lines.

InsertBlock or **-InsertBlock** inserts blocks or another drawing into the current drawing.

InsertBlockN inserts blocks as an array.

InsertObject displays data from other programs in drawings, such as text documents, spreadsheets, and images (WINDOWS ONLY).

InsertShape inserts shape from *.shx* shape files into the drawing.

Interfere checks interferences between solid models.

Intersect creates regions or 3D solids from the intersection of regions or 3D solids.

IsolateLayer isolates the layers of selected entities.

IsolateEntities hides all other entities from view.

IsometricGrid controls the isometric plane (left, right, or top) when isometric snap is used.

J

JoggedDimension draws jogged radius dimensions for circles and arcs.

L

Language sets the language to communicate with the software.

Layer opens the Layers Manager palette.

-Layer controls layers and layer properties.

LayerDialog opens the legacy Layers Manager dialog box.

LayerPreview displays the content of individual layers, one by one.

LayersDgn toggles the visibility of layers in DGN underlays.

LayersPdf toggles the visibility of layers in PDF underlays (Windows and Mac OS X only).

LayerState saves and restores the properties of layers.

Leader draws leader lines that connect annotations to drawing entities.

Light places lights in drawings.

Lightlist displays the lighting palette.

Line draws straight line segments.

LinearDimension draws linear dimensions.

LinearPattern places copies of entities in a linear pattern (Pattern command, Linear option).

LineColor or **-LineColor** sets the color for new entities.

LineScale sets the global linestyle scale factor.

LineStyle or **-LineStyle** creates, loads, and sets line styles.

LineWeight or **-LineWeight** sets lineweight options.

LoadApplication loads applications for use in the program.

LoadCustomization adds and removes partial customization (CUI, CUIX, XML, MNU, MNS, and MNI) files to and from the main customization file.

LoadMenu adds menu or customization files to customization sources.

LoadScript executes a sequence of commands from scripts.

LoadShape loads shape files for use by the Shape command.

LockLayer locks the layers of selected entities.

Loft generates lofts between path entities that act as cross-sections.

M

MakeBlock or **-MakeBlock** makes blocks from selected entities.

MakeBlockAttribute or **-MakeBlockAttribute** makes block attribute definitions.

MakeFlatSnapshot flattens 3D models to 2D drawings.

MakeSlideLibrary reads a *.txt* text file containing the names of *.sld* slide files, and makes an *.slb* slide library file for use by image menus.

MarkDivisions places evenly spaced point entities or blocks along the length of entities.

MarkLengths places point entities or blocks at measured intervals on entities.

Mask covers existing entities with a blank area.

MatchLayer changes the layer of selected entities to match a destination layer.

MergeLayer and -MergeLayer merge one or more layers with a destination Layer via dialog box and command line.

Mesh draws 3D polygon meshes.

Migration migrates profiles, preferences, file location settings, and UI customizations from the previous version of the software. (This is a stand-alone program that runs automatically after you exit the program and then start it again.)

Mirror draws mirror image copies of entities.

Mirror3D draws mirror images of entities about a plane in 3D space.

MirrorHatch determines whether hatches and gradient fills are reflected or retain their pattern direction when you apply Mirror and Flip commands:

Model toggles from a sheet tab to the model tab.

ModelMode toggles from sheet workspace to a model workspace on sheets.

Modify changes the properties of drawing entities.

ModifyProperties changes the layer, color, linestyle, linescale, and thickness of entities.

Move displaces entities a specified distance in a specified direction.

MoveDimensionText moves dimension text.

MultiLeader creates multileader entities based on the current style and scale.

MultiLeaderStyle edits multiline leader styles.

N

New starts new drawing files.

NontargetLight places free lights in drawings (lights without a targets).

Note or **-Note** draws text entities.

NoteOptions determines whether text-related commands use dialog box, toolbar, or command line as their interface.

O

ObliqueDimension modifies the orientation of existing linear dimensions.

Offset offsets linear entities in parallel orientation.

Open opens an existing drawing file.

OpenOle opens OLE objects for modification (WINDOWS ONLY).

OpenReference opens referenced drawings in a new drawing window.

OpenWebpage displays the specified Web site with the computer's default browser.

Options configures program operating parameters.

OrdinateDimension draws ordinate point dimensions.

Ortho constrains the pointer so it moves parallel to the axes of the current coordinate system.

OsCalc displays the default calculator provided by the operating system.

OverrideDimensionStyle overrides dimension style variables.

P

PackAndGo packages together the current drawing file plus dependent files (all references, font and mapping files, and print style and configuration files).

PageLayout sets print and plot defaults.

Pan or **-Pan** moves the drawing display in the active view tile. (Pan performs dynamic pan.)

PanDown pans down.

PanLeft pans left.

PanRight pans right.

PanUp pans up.

ParallelDimension draws aligned linear dimensions.

Parameters and **-Parameters** create and edit constraint expressions and values.

Paste inserts data from the Clipboard.

PasteAsBlock inserts data from the Clipboard as block.

Paste@SourcePosition pastes entities from the clipboard at the coordinates from the source drawing.

PasteSelected pastes entities from the clipboard, after the user specifies the format.

PathPattern places copies of entities along a path (Pattern command's Path option).

Pattern or **-Pattern** draws multiple copies of entities in a pattern.

Pattern3D draws linear or circular patterns of entities in 3D space.

PauseScript pauses scripts.

PlaneSurf draws rectangular planar surfaces.

PlanView sets plan view to construction plane.

Plugins loads plugins (small programs) provided by third-party developers.

Point draws point entities.

PointFormat defines the type and size of point entities.

PointLight places point lights in drawings.

PolyFace draws 3D polyface meshes, vertex by vertex.

Polygon draws equilateral closed polylines.

Polyline draws 2D polylines.

Polyline3D draws 3D polylines of straight line segments in 3D space.

Polysolid draws wall-like 3D solids from points picked in the drawing, or else converted from 2D entities, such as lines, polylines, arcs, circles, ellipses, and splines.

PowerTrim trims multiple entities by dragging the pointer across them, or through selection sets.

Preview shows a preview before printing the drawing.

Print or **-Print** plots the drawing to a plotter, printer, or file.

PrintStamp prints or plots a stamp on the plot.

PrintStyle or **-PrintStyle** manages named print styles.

Profiles stores and manages individual profiles based on specified settings.

Properties displays the Properties palette to change drawing entity properties.

PropertyPainter applies properties from one entity to selected entities.

Pyramid draws three-dimensional solid pyramids.

Q

QuickGroup creates unnamed groups, which can be named later; does not replace the Group command.

QuickModify copies, moves, rotates, or scales one or more entities during one command.

QuickPrint prints the current view of a drawing to the printer set as the default output device.

QuickUngroup ungroups selected groups.

R

RadiusDimension draws radial dimensions for circles and arcs.

Ray draws semi-infinite construction lines

Rebuild rebuilds the drawing and refreshes the active view tile.

RebuildAll rebuilds the drawing and refreshes all view tiles.

RebuildDimension updates dimensions with the active dimension style.

RebuildField updates the values of field text.

RecordMacro records keyboard input and mouse actions.

Recover repairs damaged drawings.

Rectangle draws a rectangular polyline.

Redefine restores built-in commands that have been undefined using the Undefine command.

RedefineBasePoint sets new base points for blocks.

Redo reverses the effects of a previous U or UndoN command.

RedoN reverses last or all previous U and UndoN commands.

References or **-References** displays the references palette.

Refresh refreshes the display of the active view tile.

RefreshAliases reloads the modified alias commands map file.

RefreshAll refreshes the display of all currently-open view tiles.

Region converts an entity enclosing an area into a region.

RelateDimension reestablishes the relation of specified dimensions to entities.

RemoveSignature removes an existing digital signature from the current drawing or from file names that you specify.

Rename or **-Rename** changes the names of objects.

Repeat repeats the specified command.

ReplaceDimensionText modifies the dimension text value.

ReplaceNew creates a new drawing, replacing the current one.

ReplaceOpen opens a drawing, replacing the one currently open.

ReportBug opens the default Web browser to the support.graebert.com Web site for reporting bugs and other support issues.

ResetDimensionText undoes the movement of dimension text.

ResetOle resets the OLE object to its original state (WINDOWS ONLY).

ResumeScript resumes an interrupted script.

Revolve draws 3D solids or surfaces by revolving 2D objects about an axis.

RevolvedMesh draws rotated surfaces about a specified axis.

RichLine draws rich lines.

RichLineStyle edits rich line styles.

Ring draws filled circles or rings.

RollView controls 3D view points.

RollViewAutoTarget determines how the target point is detected for RollView commands.

RollViewCenter sets the center of rotation, and then goes into constrained orbit viewing.

RollViewContinuous rotates the view in 3D space with continuous motion about a specified pivot point.

RollViewFree views models interactively in 3D space without constraining the roll.

RollViewHorizontal swivels the model view left and right about an imaginary vertical axis.

RollViewVertical tilts model view of the model up and down about an imaginary horizontal axis.

Rotate rotates entities about base points.

Rotate3D moves entities about 3D axes.

RotatedDimension places linear dimensions that are rotated by an angle.

RotateDimensionText rotates dimension text.

RuledMesh draws ruled surfaces between two curves.

S

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.

SaveComponent saves changes made to blocks or references after being edited as components.

Scale enlarges or reduces specified entities equally in the X, Y, and Z directions.

ScriptN repeats script execution.

Section creates a cross section based on the intersection of a plane and 3D solids.

Select places selected entities in the 'Previous' selection set.

SelectAll selects all drawing entities including those not visible.

SelectMatching selects entities of the same type and properties.

SelectionFilter determines a list of requirements that drawing entities must meet to be included in a selection set.

SetByLayer changes property overrides of specified entities to ByLayer.

SetFieldDisplay toggles the gray background to field text.

SetVariable lists or sets the values of system variables.

Shade shades the drawing mode.

ShadeView determines the shade mode.

Sheet creates, copies, renames, and deletes sheet tabs (layout modes).

SheetMode toggles from a model workspace to sheet workspace on sheets.

Shell prompts to start a program external to the program, such as Notepad or an image editor.

ShowLayers turns on all layers.

ShowLicense displays the license agreement.

SignFile adds a digital signature to the current drawing file.

SimpleNote or **-SimpleNote** displays text on screen as you enter it.

SimplePolygon draws closed equilateral polygons with a minimum of prompts.

Sketch draws freehand lines.

Slice slices 3D solids with a plane or surface.

SmartBMP exports the image of the current drawing to a BMP bitmap file.

SmartCalculator performs mathematical and trigonometric computations and unit conversions.

SmartDimension dimensions entities appropriately, such as linear dimensions for lines and radial dimensions for circles.

SmartLeader draws leader lines and annotations.

SmartNew starts a new drawing with the default template drawing.

SmartOpen opens a drawing specified at the command prompt.

SmartSelect draws entity selections based on filtering criteria.

Snap restricts pointer movements and pointing in the drawing to specified intervals.

Solid draws solid-filled 2D faces.

SpellCheck checks the spelling of text in the drawing (Windows and Mac OS X only).

Sphere draws three-dimensional solid spheres.

Spline draws quadratic or cubic non-uniform rational Bezier spline (NURBS) curves.

Split splits entities into two, or removes a portion of an entity.

SplitDimension splits dimension and extension lines at intersections of other entities; also joins split dimension and extension lines.

SpotLight inserts spot lights into drawings.

StackOptions sets the alignment options and size of stacked (fraction) text.

StopMacro stores macros recorded with RecordMacro in a *.vcprog* file.

Stretch moves or stretches entities.

Subtract creates a composite region or a 3D solid by subtraction.

Sweep creates solid primitives or surfaces by sweeping two dimensional entities along a path.

SwitchArea switches between model and sheet tabs.

SystemOptions determines general operating options.

T

Table or **-Table** draws tables in drawings.

TableStyle draws and manages table styles.

TabulatedMesh draws tabulated surfaces from a path curve and a direction vector.

TargetpointLight places targeted point lights.

TextStyle or **-TextStyle** creates or modifies text styles, and sets the active text style.

ThawLayers thaws all layers.

Thicken creates 3D solids by thickening surfaces.

ToActiveLayer changes the layers of specified entities to the active layer.

Tolerance draws tolerances (datum indicators and basic dimension notation).

Toolbars or **-Toolbars** displays and hides toolbars.

ToolMatrix displays the Tool Matrix palette.

Torus draws three-dimensional toroid solids.

Trace draws traces.

Trapezoid draws trapezoid 2D polylines.

Trim trims entities at a cutting edge defined by other entities.

U

U reverses the most recent command.

UIStyle switches the user interface between dark and light themes.

UiProfile manages user interface profiles.

Undefine disables built-in commands.

Undelete restores deleted entities.

UndoLayer restores previous layer settings.

UndoN reverses the effect of multiple commands.

Union creates composite regions or solids by addition.

UnisolateEntities makes entities visible again following the IsolateEntitites and HideEntities commands

UnisolateLayer undoes layer isolation.

UnitSystem or **-UnitSystem** sets coordinate and angle display formats and precision.

UnlockLayer unlocks locked layers.

UnrelateDimension dissolves the relation of specified dimensions to entities.

UpdateBlockAttributes updates all instances of a block with new or modified attributes.

UserPreferences defines drafting options, mouse options, and alias commands.

V

VerifyDimensions verifies all measurements of related dimensions in the drawing.

VerifySignature checks the digital signature of a drawing for authentication.

VerifyStandards verifies that drawings conform to standards specified by the DrawingStandards command.

VerticalDimensions places vertical dimensions.

ViewDirection sets the viewing direction for a 3D view of the drawing.

Viewport or **-Viewport** draws and controls viewports on sheet tabs.

ViewportLayer controls layer visibility in viewports on sheets.

Views or **-Views** saves, restores, and manages user-defined model and sheet views, and presets views.

ViewSlide displays captured slides in the drawing area.

ViewTiles draws and controls single or multiple model view tiles or sheet viewports.

VoiceNote records audio, and then places it in the drawing.

VSTA records, runs, and edits macros at the command line, using Microsoft's Visual Studio Tools for Applications programming language.

VSTAManager records, runs, and edits VSTA macros in a dialog box.

W

Wedge draws three-dimensional solids with a sloped face tapering along the X axis.

Weld joins two entities into a single entity.

Windows cascades and tiles windows, and arranges icons.

XTra Tools

BreakLine places break symbol in lines.

CopyMultiple creates multiple copies of the selected entities. **ConvertSimpleNotes** converts regular simple notes (text) into notes (mtext).

CurvedText places text along curves.

EditTextCase modifies the case of selected text, attributes, notes, or dimensions.

EditTextJustification modifies the justification point of selected text, paragraph, or attribute entities.

EditTextOrientation rotates selected text, paragraph, or attribute entities keeping their current position.

ExchangeDimensionStyle exports the dimension styles from the current drawing to an external ASCII file.

ExportSettings exports the current settings to an external file.

ExportVariables exports the values of system variable to a script file (.scr file).

FitText resizes the width of a text entity to fit between two points.

GetNestedProperties displays the properties of an entity in a block or a reference.

GetBlockInfo reports information about blocks in drawings.

ImportBlockAttributes imports attribute values in blocks.

ImportDimensionStyle imports dimension styles and their settings from an ASCII file (with .dim extension).

LispExplorer lists LISP routines by function name.

MakeLineStyle defines new line styles.

MakeShape defines a shape based on selected entities.

MergeSheets merges several sheets (layouts) into a single sheet.

NumberText inserts sequential numbering to selected text entities.

OffsetX creates new entities by applying an offset to the selected entities.

PatternHatch fills areas with images, xrefs, masks, and blocks .

RedefineBlockAttribute redefines attributes in blocks.

RedefineBlockAttributeValues globally modifies the values of block attributes in all selected blocks.

ReplaceBlock replaces one block with another one.

ReplaceBlockByReference replaces blocks with xrefs.

ScaleBlock scales blocks relative to their insertion point.

TextFrame draws a frame around the specified text entities.

TextMask places a mask (blank area) behind specified text entities.

TextUnMask removes the masks assigned to selected text entities.

VariableToScript appends selected variables to an existing script file (.scr file).

Z

Zoom increases or decreases the visible part of the drawing.

ZoomBack zooms to previous view.

ZoomDynamic zooms the drawing in real-time (dynamic zoom).

ZoomFactor zooms by a factor.

ZoomFit zooms to the drawing extents.

ZoomIn zooms in to the drawing.

ZoomOut zooms out from the drawing.

ZoomWindow zooms to a region of the drawing specified by a box.

ZPlane sets elevation and thickness of new drawing entities.

#

3D draws 3D polygon mesh objects: boxes, cones, cylinders, dishes, domes, pyramids, spheres, tori, wedges, or meshes.

3PointAngleDimension draws angular dimensions based on three points.

4PointAngleDimension draws angular dimensions based on four points.

Removed Commands

~~3DMouseButtons~~, ~~3DMouseOptions~~, and ~~-3DMouseOptions~~ are replaced by the utility program provided by 3dConnexion.

Notes

APPENDIX B

Alphabetical System Variable Summary

In This Appendix

- Brief description of system variables
- System variables listed in alphabetical order

The program uses system variables to control the state of drawings, options in commands, and aspects of the user interface. On the following pages, over 700 system variables are listed in alphabetical order. You can view and modify the values of system variables with the **SetVarariable** command.

Some system variables only report values, and so you cannot change their value. These ones are known as "read only" variables. The **(r/o)** marker indicates that they are read-only.

A number of system variables are included for compatibility with other CAD programs, such as AutoCAD, and some of them have no effect in the program. In these cases, then the note *(The program does not support...)* is attached to them. When a group of system variables all start with the same few letters and are included with ARES for compatibility, they are grouped together and shown with an ellipsis (...), such as **Camera...** .

A few system variables are present in the program for legacy reasons, which means that they provide compatibility for Graebert's older software, such as PowerCAD.

System variables new since the last edition of this book are shown in red.

Commands related to dimensional and geometric constraints are available in the Windows version only.

A

AcadLspAsDoc (*not yet supported)*

AcadPrefix (r/o) specifies paths to support files used by the program.

AcadVer (r/o) for compatibility with apps that also run on AutoCAD; reports "17.0".

AcisOutVer controls the ACIS version number; valid values are 15, 16, 17, 18, 20, 21, 30, 40, or 70.

AcisProxyMode toggles how unsupported ACIS solids displayed, as meshes or edges only.

AcisSaveAsMode controls how ACIS entities (3D solids, bodies, regions) are exploded for export.

ActDb (r/o) returns ID number of active drawing database.

ActPath reports the path to where Action macros are stored.

ActRecorderState (r/o) reports whether the Action recorder is active.

AdcState (r/o) reports whether the Design Resources palette is open.

AFlags controls default attribute display mode.

AngBase controls direction of zero degrees relative to CCS.

AngDir controls rotation of positive angles.

AnnoAllVisible shows or hides annotative entities that do not support the current annotation scale.

AnnoAutoScale controls if setting a new annotation scale are added automatically to annotative entities.

AnnotativeDwg determines whether the current drawing will act as annotative Block when inserted into another drawing.

ApBox toggles display of aperture box cursor.

Aperture controls entity snap aperture in pixels.

ApiLanguage is reserved for API (application programming interface) settings.

ApiLevel is used for application development API settings.

ApiPath is used for application development API settings.

ApState reports state of ApBox system variable.

Area (r/o) reports area measured by Area command.

ArLineDisp determines whether edges of faces and bodies are displayed by the real-time render window (ARender command).

AttDia controls user interface for entering attributes.

AttIpe determines the type of pop-up toolbar to display when editing multiline BlockAttributes in place.

AttMode controls display of attributes.

AttMulti controls whether you can create multiline BlockAttributes with the commands to define BlockAttribute.

AttReq toggles whether attribute values are prompted for during insertion.

AuditCtl toggles creation of *.adt* audit log files.

AUnits specifies type of angular units.

AUPrec specifies number of decimal places displayed by angles.

AutoNew determines whether the program starts new drawings based on a default template drawing, or asks the user to specify the template file.

AutoNewName specifies the name for new drawings; default is "noname_*n*," where *n* is incremented.

B

BackgroundPlot toggles background printing.

BackZ controls location of back clipping plane offset from target plane.

BActionColor through **BGripObjSize** *(The program does not have a dynamic block editor.)*

BigFontAlt specifies the replacement Bigfont file to substitute for Bigfont used in the current drawing but is not available.

BindType specifies how reference (xref) names are converted when bound or edited; while the program supports the binding of references, it ignores the setting of this system variable.

BlipMode toggles display of blip marks.

BlockEdit... and **BParameter...** *(The program does not have a dynamic block editor.)*

BtMarkDisplay toggles the display of value set increment markers in Dynamic Blocks.

BuildVersion (r/o) reports the build version of the program.

BvMode *(The program does not have a dynamic block editor.)*

C

CalcInput *(The program does not support results exported from the Calculator.)*

Camera... *(The program does not have a Camera command.)*

CAnnoScale specifies the name of the current annotation scale for the active workspace (Model, or Viewports on layout Sheets).

CAnnoScaleValue specifies the value of the current annotation scale for the active workspace (Model, or Viewports on layout Sheets).

CConstraintForm toggles between creating dynamic and annotational constraints with dimensional constraints commands.

CDate (r/o) specifies current date and time in format yyyymmdd.hhmmssdd.

CDefaultColor sets color for the display of default values at the command line.

CeColor specifies current color.

CLExtension determines the default center line extension distance for the Centerline command.

CeLtScale specifies current linetype scaling factor.

CeLType specifies current linetype.

CeLWeight specifies current lineweight in millimeters.

CenterMt determines how grips resize the Note bounding box.

CeTransparency specifies the current level of transparency for entities.

ChamferA specifies current value of first chamfer distance.

ChamferB specifies current value of second chamfer distance.

ChamferC specifies current value of chamfer length.

ChamferD specifies current value of chamfer angle.

ChamMode toggles chamfer input mode.

CircleRad specifies most-recent circle radius.

CLayer specifies name of current layer.

CleanScreenState (r/o) reports whether full screen (cleanscreen) mode is active.

CliState (r/o) reports whether command line palette is active.

CMaterial *(The program does not support materials.)*

CmdActive (r/o) reports type of command currently active.

CmdDia toggles between dialog box and command-line prompts for the SmartLeader command; has no effect in the program.

CmdEcho toggles LISP command display.

CmdInputHistoryMax specifies the maximum number of lines of command-prompt history to record in the text window; has no effect in the program.

CmdLnText specifies command prompt; default is ':'.

CmdNames (r/o) reports name of command currently active.

CMLeaderStyle specifies the current multileader style for new multileader entities.

CMLJust specifies the current richline (multiline) justification mode.

CMLScale specifies the current scale factor for the overall richline width.

CMLStyle specifies current richline style.

ColorTheme toggles the color of the user interface between light and dark themes.

Compass toggles display of 3D compass.

ConstraintBarDisplay toggles the display of constraint icons.

ConstraintBarMode toggles the display of geometrical constraint icons in constraint bars.

ConstraintNameFormat determines the format of text in dimensional constraints: value, formula, or name.

ConstraintRelax toggles whether how constraints are relaxes or enforced when objects are edited.

ConstraintResolveMode *(The program does not support resolve mode.)*

Coords specifies coordinate display style.

COptionsColor determines the color of the command window.

CopyMode determines if the Copy command repeats, or makes one copy and exits.

CPlotStyle specifies current print style.

CProfile (r/o) specifies name of current profile (*.arg*) file.

CrossingAreaColor *(The program does not support selection area fills.)*

CShadow *(The program does not support shadow casting.)*

CTab specifies name of current tab.

CTableStyle specifies name of current table style name.

CursorMode specifies look of crosshair cursor.

CursorSize specifies cursor size as a percent of viewport size.

CView... (The program does not support detail and section styles.)

CVPort specifies current viewport number.

D

DashboardState *(The program does not have a Dashboard palette.)*

Date (r/o) reports current date in Julian format, such as 2448860.54043252.

DbcState *(The program does not support database links)*

DblClkEdit toggles editing by double-clicking objects.

DBMod (r/o) reports how drawing has been modified.

DctCust *(The program does not have a custom dictionary.)*

DctMain controls the language code for spelling dictionary.

DebugMode controls graphic card optimization; used for support reasons.

DebugMode2 specifies a debugging mode for developers.

DefaultLighting toggles distant lighting.

DefaultLightingType toggles between new (1) and old (0) types of lighting.

DefLPlStyle reports default print style for layer 0.

DefPlStyle reports default print style for new objects.

DefPlStyleTable specifies the default print style for new entities and layers in drawing files that use the R14 (or earlier) DWG or DXF format.

DelObj toggles deletion of source objects used to create solids and regions.

DemandLoad *(The program does not support proxy entities.)*

DevOptions specifies options reserved for third-party developers.

DgnFrame toggles the rectangular frame around referenced DGN files.

DgnImportMax specifies the maximum number of DGN elements to reference; 0 = no limit.

DgnOsnap toggles entity snapping to DGN elements.

DiaStat (r/o) reports whether user exited dialog box by clicking OK or Cancel.

DimConstraintIcon toggles the display of lock icons near dimensional constraints.

DispSilh toggles silhouette display of 3D solids.

Distance (r/o) reports distance last measured by GetDistance command.

DonutId specifies inside diameter of donuts.

DonutOd specifies outside diameter of donuts.

DragMode controls drag mode.

DragP1 (hardware acceleration) specifies maximum number of entities to drag before checking for input from the mouse.

DragP2 (software acceleration) specifies maximum number of entities to drag before checking for input from the mouse.

DragVs *(The program does not support visual styles.)*

DrawOrderCtrl specifies behavior of draw order.

DrState (This program does not support the Drawing Recovery panel.)

DTextEd toggles the user interface of the SimpleNote command between dialog box and command-line.

DtTool... (Legacy FelixCAD/PowerCAD variable for control panel settings; replaced in ARES with ToolMatrix)

DwfFrame toggles the frame around DWF overlays.

DwfOsnap toggles entity snapping of DWF frames.

DwgCodePage (r/o) reports the same value as SysCodePage.

DwgName (r/o) reports current drawing file name, such as "drawing1.dwg".

DwgPrefix (r/o) reports drawing's drive and folder, such as "C:\Users\rhg\Documents\".

DwgTitled (r/o) reports whether the drawing has been given a file name.

DxEval determines which actions cause the program to check referenced files.

DxfPrec specifies the default precision when saving drawings as DXF files.

DxfTextAdjustAlignment adjusts the alignment of text in drawings exported to DXF files.

DynaSnap specifies visibility of entity snap cues and tooltips.

DynaSnapPolarMode toggles polar snap mode.

DynaSnapSize specifies the size of entity snap icons displayed when moving the cursor over geometric features of objects.

DynConstraintMode determines whether hidden dimensional constraints are displayed when constrained objects are selected.

DynDiGrip determines which input boxes are displayed as you move an EntityGrip.

DynDiVis controls the Quick Input dimension input method when you stretch an entity using an EntityGrip.

DynMode controls the use of Quick Input pointer input and dimension input methods.

DynPiCoords controls whether the Quick Input pointer input method uses relative or absolute coordinates.

DynPiFormat controls whether the Quick Input pointer input method uses polar or Cartesian coordinates.

DynPiVis controls the Quick Input pointer input method.

DynPrompt turns on or off the use of prompt tooltips and command input boxes display for Quick Input.

DynToolTips determines which kind of tooltips are affected by appearance settings.

Dimensions

DimADec specifies angular dimension precision.

DimAlt toggles alternate units.

DimAltD specifies alternate unit decimal places.

DimAltF specifies alternate unit scale factor.

DimAltMzf specifies the alternate sub-units size.

DimAltMzs specifies the alternate sub-units suffix.

DimAltRnd specifies rounding factor of alternate units.

DimAltTD specifies decimal places of tolerance alternate units.

DimAltTZ specifies display of zeros in alternate tolerance units.

DimAltU specifies display of alternate units.

DimAltZ specifies display of zeros in alternate units.

DimAnno returns whether the current dimension style is an annotative style.

DimAPost specifies prefix and suffix for alternate text.

DimArcSym specifies location of arc symbol.

DimAso toggles between associative and non-associative dimensions.

DimAssoc specifies how dimensions are created.

DimASz specifies default arrowhead length.

DimAtFit specifies how text and arrows are fitted when there is insufficient space between extension lines.

DimAUnit specifies format of angular dimensions.

DimAZin specifies display of zeros in angular dimensions.

DimBlk specifies name of arrowhead block.

DimBlk1 specifies name of first arrowhead's block.

DimBlk2 specifies name of second arrowhead block.

DimCen specifies how center marks are drawn.

DimClrD specifies color of dimension lines.

DimClrE specifies color of extension lines and leaders.

DimClrT specifies color of dimension text.

DimDec specifies number of decimal places for primary tolerance.

DimDLE specifies length of dimension line extension.

DimDLI specifies increment distance of continued dimension lines.

DimDSep specifies decimal separator (a single character).

DimExe specifies distance of extension line above dimension line.

DimExO specifies extension line origin offset.

DimFit specifies how dimension text is fitted around extension lines.

DimFrac specifies fraction format.

DimFXL specifies default length of fixed extension lines.

DimFxlOn toggles fixed-length extension lines.

DimGap specifies gap between text and dimension line.

DimJogAng specifies default angle for jogged dimension lines.

DimJust specifies positioning of horizontal text.

DimLdrBlk specifies name of block used for leader arrowheads.

DimLFac specifies linear unit scale factor.

DimLim toggles display of dimension limits.

DimLtEx1 specifies linetype for first extension line.

DimLtEx2 specifies linetype for second extension line.

DimLtype specifies linetype name for dimension line.

DimLUnit specifies dimension units (except angular).

DimLwD controls dimension line lineweight.

DimLwE specifies extension lines lineweight.

DimMzf specifies the sub-zero factor for metric dimensions.

DimMzs specifies sub-zero suffix for metric dimensions.

DimPost specifies default prefix or suffix for dimension text.

DimRailAngle specifies the radial/diameter snap angle for dimension offset snap; default = 0.2618

DimRailFirstOffset specifies the offset of the first dimension line from the measured linear entity; default = 0.4000

DimRailOffset specifies the offset distance between successive dimension lines for dimension offset snap; default = 0.2500

DimRailsEnabled toggles dimension offset snap; default = 1

DimRnd specifies rounding value for dimension distances.

DimSAh toggles whether separate arrowhead blocks are used.

DimScale specifies overall dimension scale factor.

DimSD1 toggles display of first dimension line.

DimSD2 toggles display of second dimension line.

DimSE1 toggles display of first extension line.

DimSE2 toggles display of second extension line.

DimSho toggles dimension updates while dragging.

DimSOXD toggles display of dimension lines outside of extension lines.

DimStyle (r/o) specifies name of current dimension style.

DimTAD specifies vertical position of text.

DimTDec specifies number of decimal places for primary tolerances.

DimTxtDirection specifies the reading direction by which text is displayed in dimensions.

DimTFac specifies scale factor for tolerance text height.

DimTFill toggles background fill color for dimension text.

DimTFillClr specifies background color for dimension text.

DimTIH toggles alignment of text placed inside extension lines.

DimTIX toggles placement of text inside extension lines.

DimTM specifies value of minus tolerance.

DimTMove specifies how dimension text is moved.

DimTOFL toggles placement of dimension lines.

DimTOH toggles text alignment when outside of extension lines.

DimTol toggles generation of dimension tolerances.

DimTolJ specifies vertical justification of tolerances.

DimTP specifies value of plus tolerance.

DimTSz specifies size of oblique tick strokes.

DimTVP controls vertical position of text when DimTad = 0.

DimTxSty specifies name of text style.

DimTxt specifies text height.

DimTZin specifies display of zeros in tolerances.

DimUnit specifies units used by dimensions.

DimUPT specifies user-positioned text.

DimZIN specifies display of zero in feet-inches units.

E

EdgeMode toggles edge mode for Trim and Extend commands.

Elevation specifies current elevation relative to current CCS.

EnterpriseMenu (r/o) points the the folder holding common menu files.

ErrNo (r/o) reports error numbers from Lisp and other APIs.

ErState (r/o) reports display of external references palette.

Expert specifies display of prompts.

ExplMode toggles whether Explode command explodes non-uniformly scaled blocks.

Export... *(The program does not support this export function.)*

ExtMax (r/o) specifies upper-right coordinates of drawing extents.

ExtMin (r/o) specifies lower-left coordinates of drawing extents.

ExtNames specifies format of named objects; a legacy system variable.

F

FaceTRatio specifies aspect ratio of facets on rounded 3D bodies.

FaceTRres specifies smoothness of shaded and hidden-line objects.

FcMultiSel is like LISP's *ssget* flag, specifies whether one or multiple entities are added to selection sets chosen from the graphics area. (FC is short for FelixCAD.)

FcTemplate specifies default template drawing used by New command.

FcVersion specifies the version number based on legacy Graebert CAD product numbering.

FieldDisplay toggles the gray background behind field text.

FieldEval determines when field test is reevaluated.

FileDia toggles user interface for file-access commands.

FileTabPreview determines the preview type when you hover over a drawing tab.

FileTabState controls the display of drawing tabs at the top of the drawing windows area.

FilletRad specifies current fillet radius.

FilletRad3d specifies the default radius for 3D fillets.

FillMode toggles fill of 2D solids, wide polylines, traces, fills, and hatches.

FontAlt specifies font used for missing fonts.

FontMap specifies name of font mapping file.

Frame determines whether frames are displayed and/or plotted.

FrontZ (r/o) reports front clipping plane offset.

FullOpen *(The program does not open drawings partially.)*

FullPlotPath toggles whether the file name or the full path is sent to the spooler (printer buffer).

G

Geo... *(The program does not support geographic coordinates.)*

Gf... *(The program has gradient fill but does not support this system variable.)*

GraebertStorage toggles the display of the Cloud Storage palette.

GraebertStorageServer specifies the URL for the default cloud storage service.

GridDisplay determines grid display (sum of bit codes).

GridMajor *(The program does not support lined grids.)*

GridMode toggles display of grid.

GridUnit specifies x, y spacing of grid.

GripBlock toggles display of grips in blocks.

GripColor specifies color of unselected grips.

GripDynColor controls the fill color of user-defined grips of dynamic blocks.

GripHot specifies the color of hot grips.

GripHover specifies grip color when cursor hovers.

GripMultifunctional toggles the use of multifunction grips with entities.

GripObjLimit *(The program does not limit the number of grips displayed.)*

Grips toggles display of grips.

GripSize specifies size of grips.

GripTips *(The program does not have grip tooltips.)*

GroupDisplayMode determines how groups are selected: all entities, as a group, or by a bounding box.

Gt... *(The program does not support these variables for grip tools.)*

H

HaloGap *(The program does not support visual styles.)*

Handles (r/o) reports whether entity handles (not grips) can be accessed by applications.

HidePrecision determines the precision of hidden-line removal.

HideText *(The program always hides text.)*

Highlight toggles entity selection highlighting.

HighlightColor specifies the color of selected entities.

HPAng specifies current hatch pattern angle.

HpAnnotative controls whether new hatches or filled areas are annotative entities.

HpAssoc toggles associativity of hatches.

HpBound specifies whether polylines or regions are created by Hatch and Boundary commands.

HpDouble toggles double hatching.

HpDrawOrder specifies draw order of hatches and fills.

HpGapTol specifies largest gap allowed in hatch boundaries.

HpInherit toggles how the PropertyPainter command copies hatch origin from source entity to destination entities.

HpLayer specifies the default layer name for hatch patterns.

HpLinetype toggles whether linetypes used for hatch lines are displayed as continuous or as linetypes.

HpName specifies default hatch name.

HpObjWarning specifies maximum number of hatch boundaries that can be selected before program warns users.

HpOrigin specifies default origin for hatch objects.

HpOriginMode specifies default hatch origin point.

HpScale specifies current hatch scale factor.

HpSeparate specifies whether the Hatch command creates a single hatch, or separate hatches when applied to multiple boundaries.

HpSpace specifies default spacing of user-defined hatches.

HpTransparency specifies the current level of transparency for hatch patterns.

HyperlinkBase *(The program does not support hyperlink base addresses.)*

I

ImageFrame toggles the visibility of the rectangular frame around referenced images.

ImageHlt *(The program does not toggle image frame highlighting.)*

ImpliedFace *(The program does not detect implied faces.)*

IndexCtl *(The program does not support spatial indices.)*

InetLocation specifies the default Web address, such as http://www.graebert.com.

IniFilename stores the name of an application initialization file used for development support.

InitgetFlag is like LISP's initget flag, maintains compatibility with legacy Graebert CAD products.

InputHistoryMode *(The program does not support command history modes.)*

InputSearchDelay specifies the delay in milliseconds before commands are suggested.

InsBase specifies default insertion base point relative to current CCS for Insert and Xref commands.

InsName specifies default block name.

InsUnits specifies drawing units of blocks dragged into drawings.

InsUnitsDefSource specifies source drawing units value.

InsUnitsDefTarget specifies target drawing units.

IntelligentUpdate specifies the frame rate for display updates.

Interfere... *(The program does not support interference display.)*

Intersection... *(The program does not support intersection display.)*

ISaveBak toggles creation of *.bak* backup files.

ISavePercent specifies when the program performs a full save; partial saves are faster than full saves.

IsoLines specifies number of contour lines on 3D solids.

L

Language specifies the language used by the user interface: 1 = German, 2 = English, 6 = Japanese, 7 = French, 11 = Spanish, and so on.

LastAngle (r/o) reports end angle of last-drawn arc.

LastAppLoadFolder (r/o) reports the folder path from which the last application was loaded.

LastAttachFolder (r/o) reports the path to the folder from which the last externally referenced drawing file was attached.

LastInsertFolder (r/o) reports the folder path from which the last block was inserted.

LastLispFolder (r/o) reports the folder path from which the last LISP routine was loaded.

LastOpenFolder (r/o) reports the path to the folder from which the last drawing file was opened.

LastPoint reports x,y,z coordinates of last-entered point.

LastPrompt (r/o) reports last string on command line.

LastScriptFolder (r/o) reports the folder path from which the last SCR script file was loaded.

LastShapeFolder (r/o) reports the path to the folder from which the last SHX shape file was loaded.

Latitude *(The program does not support geographic data*.)

LayerDlgMode variable toggles the interface opened by the Layer command.

LayerEval *(The program does not support layer reconciliation.)*

LayerFilterAlert *(The program does not limit the number of filtered layers.)*

LayerNotify *(The program does not support layer reconciliation.)*

LayLockFadeCtl specifies the percentage of fading applied to entities on locked layer.

LayoutRegenCtl determines if display lists are saved for layouts.

LensLength *(The program does not support cameras.)*

LightGlyphDisplay toggles display of light glyph.

LightingUnits specifies type of lighting used.

LightListState (r/o) reports whether the Lighting palette is open.

LightsInBlocks toggles the use of lights in blocks.

LimCheck toggles drawing limits checking.

LimMax specifies upper right drawing limits.

LimMin specifies lower left drawing limits.

LineSmoothing toggles anti-aliasing to smooth the display of entities.

LispInit determines if LISP variables are saved between drawing sessions.

Locale (r/o) reports ISO language code used by the operating system.

LocalRootPrefix (r/o) specifies path to local root folder.

LockUi specifies position and size of toolbars and palettes.

LoftAng1 specifies first loft angle.

LoftAng2 specifies second loft angle.

LoftMag1 specifies magnitude of loft at first cross section.

LoftMag2 specifies magnitude of loft at second cross section.

LoftNormals determines the type of lofting.

LoftParam determines the shape of the loft.

LogFileMode toggles writing command prompts to *.log* file.

LogFileName (r/o) specifies name of *.log* logging file.

LogFilePath specifies path to *.log* logging file.

LogInName (r/o) reports user's login name.

Longitude *(The program does not support geometric location.)*

LspALoad *(obsolete)* names the LISP file to load when opening FLX files from PowerCAD and FelixCAD.

LTScale specifies current linetype scale factor.

LUnits specifies current linear units display.

LUPrec specifies current decimal places (or inverse of smallest fraction) of linear units.

LwDefault specifies default lineweight, in millimeters.

LwDisplay toggles whether lineweights are displayed.

LwDispScale specifies the factor by which lineweights are scaled on screen.

LwUnits toggles units used for lineweights.

M

MatState *(The program does not have a materials browser.)*

MaxActVp specifies the maximum number of active viewports.

MaxHatchDensity specifies maximum density of hatching.

MaxSort specifies the maximum number of items that will be sorted in lists.

MButtonPan toggles behavior of wheel mouse.

MeasureInit toggles the default system of measurement, inches or millimeters.

Measurement toggles current drawing units.

MemorySize specifies the memory size of the application; not measured in bytes.

MenuCtl *(The program does not have side menus.)*

MenuEcho specifies menu and prompt echoing.

MenuName specifies the name of the .cui or .cui file.

MeshType specifies polyface meshes, whether set to 0 or 1.

MirrHatch toggles whether hatches and gradient fills are mirrored during the Mirror and Flip commands.

MirrText toggles text handling by Mirror command.

MLeaderScale specifies the overall scale factor for multileader entities.

ModeMacro displays text strings on the status bar.

MsLtScale determines whether LineStyles on the Model tab or in Viewports of layout Sheets are scaled by annotation scale settings.

MsmState *(The program does not have a Markup Set Manager palette.)*

MsOleScale determines the size of text of OLE objects placed in paper space.

MStopOnErr toggles whether macros halt on error.

MTextEd specifies the name of the text editor for editing multiline notes, including external ones.

MTextFixed determines whether the Note editor's position is fixed.

MtMode toggles multi-threaded mode.

MtJigString specifies the dummy text to display in the Note bounding box.

MyDocumentsPrefix (r/o) reports the path to the *Documents* folder of the currently logged-in user.

N

NavVCubeDisplay (This program does not support navigation cubes)

Nomutt toggles the display of command-line text during programs.

NorthDirection (The program does not support geographic data)

O

ObjectIsolationMode determines whether objects isolated or hidden with the IsolateEntities and HideEntities command remain hidden between editing sessions.

Obscured... *(The program does not support the display of obscured entities.)*

OffsetDist specifies current offset distance.

OffsetGapType specifies how polylines reconnect when segments are offset.

OleFrame specifies visibility of OLE frames.

OleHide toggles the display of OLE entities in drawings and on plots.

OleQuality determines the quality of plotted OLE entities.

OleStartup toggles the launch of OLE-related apps.

OpenFilterIndex presets the default drawing file format for the Open command by setting the index number for the Files of Type droplist of the Open dialog box.

OpenFormatVersion presets the default drawing file format for the Open command by setting the index number for the Files of Type droplist of the Open dialog box.

OpenPartial (The program does not open drawings partially)

OpmState toggles whether Properties palette is active (OPM = Object Properties Manager).

OrbitAutoTarget specifies how the target point is detected for RollView commands.

OrthoMode toggles orthographic mode.

OsMode specifies current entity snap modes (OS = Object Snap).

OsnapCoord determines if the keyboard overrides entity snap modes.

OsnapHatch determines if entity snap works on hatch patterns.

OsnapNodeLegacy toggles whether entity snap works on Note bounding boxes.

OsnapOverride determines how entity snaps are overridden.

OsnapZ determines how entity snaps work on 3D entities.

OsOptions determines when entity with negative z values are entity-snapped.

P

PageSetupManager determines whether the Page Layout dialog box is displayed with the creation of new sheets.

PaletteOpaque *(The program does not support transparent palettes.)*

PanScale specifies scale factor when panning left, right, up, and down.

PaperUpdate toggles how program plots layouts when paper size differs from plotter's default size.

ParameterCopyMode determines how constrained entities are copied.

ParameterStatus reports whether the Parameters Manager palette is displayed.

PdfFrame toggles the display and plotting of frames around referenced PDF files.

PdfOsnap toggles entity snapping to referenced PDF files.

PdMode specifies point display style.

PdSize specifies point display size.

PEditAccept toggles display of the EditPolyline command's 'Entity selected is not a polyline' prompt.

PEllipse toggles entity used for creating ellipses.

Perimeter (r/o) reports perimeter calculated by the last GetArea command.

Perspective toggles perspective viewing in 3D viewpoints.

PFaceVMax reports the maximum number of faces.

PickAdd toggles meaning of Shift key on selection sets.

PickAuto toggles selection set mode.

PickBox specifies size of selection pickbox.

PickDrag toggles selection window mode.

PickFirst toggles command-selection mode.

PickStyle specifies how groups and associative hatches are selected.

Platform (r/o) reports name of operating system.

PlineCache controls creation of the Oddb2dpolyline vertex cache when database file is opened.

PLineGen toggles polyline linetype generation.

PLineType specifies automatic conversion and creation of 2D polylines by PLine command.

PLineWid specifies current polyline width.

PlotOffset specifies the offset distance of drawings on the paper.

PlotRotMode specifies whether the drawing is rotated on the paper.

Plotter specifies plotter name; no longer used.

PlotTransparencyOverride determines when transparent objects are plotted (not yet supported).

PlQuiet toggles display during batch plotting and scripts.

PointCloud... (This program does not support point clouds.)

PolarAddAng specifies user-defined polar angles.

PolarAng specifies increment of polar angle.

PolarDist specifies polar snap increment when SnapStyl is set to 1 (isometric).

PolarMode specifies polar and entity snap tracking.

PolySides specifies default number of polygon sides.

Popups (r/o) determines if the display can handle dialog boxes and menus.

PrevCmd (r/o) reports the name of the command currently executing (for use by application development).

Preview_height specifies the height of preview images.

Preview_width specifies the width of preview image

PreviewCreationTransparency specifies the transparency of previews during the creation of certain 3D objects.

PreviewEffect specifies visual effect when previewing selected entities.

PreviewFilter specifies exclusion of objects from selection previewing.

PreviewType toggles the type of drawing view used for thumbnails.

Product (r/o) reports name of software.

Program (r/o) reports name of software.

ProjectName determines the current project name for xrefs.

ProjMode specifies projection mode for Trim and Extend commands.

ProxyGraphics *(System variable has no effect on proxy graphics.)*

ProxyNotice *(System variable has no effect on proxy graphics.)*

ProxyShow determines the display of proxy objects the next time the drawing is opened: not displayed, displayed, or represented by a bounding box.

ProxyWebSearch *(System variable has no effect on proxy graphics.)*

PsLtScale toggles paper space linetype scaling.

PSolHeight sets the default height for 3D solids constructed by the PolySolid command.

PSolWidth sets the default width for the 3D solids constructed by the PolySolid command.

PsProlog specifies the PostScript prolog file name.

PsQuality specifies the quality of the PostScript resolution.

PStyleMode toggles plot color matching mode of drawing.

PStylePolicy (r/o) reports whether entity color is associated with its plot style.

PsVpScale controls the paper space view factor.

Publish... *(The program does not publish drawings and sheets.)*

PUcsBase (r/o) reports name of defining origin and orientation of orthographic CCS settings; works in paper space only.

Q

QcState *(The program does not have the Quick Calc palette.)*

QTextMode toggles quick text mode.

R

R12SaveAccuracy specifies number of segments between spline control segments, or on 90° elliptical arcs, when saving ellipses and splines to R12 DWG or DXF

R12SaveDeviation specifies deviation when saving ellipses and splines to R12 DWG or DXF.

RasterDpi specifies the percentage of RAM to allocate for printing raster images.

RasterPreview (r/o) toggles creation of preview images.

RasterThreshold specifies the amount of RAM to allocate for printing raster images.

RClickContextMenuTime specifies the time between user right-clicking and displaying a context menu.

RealWorldScale controls the rendering of materials with units set to real-world scale.

RecoveryMode specifies recording of drawing recovery information after program failure.

RefEditName specifies reference file name when in reference-editing mode.

RegenMode toggles regeneration mode.

Re-init reloads the *.pgp* file.

RememberFolders toggles whether all file dialog boxes go to folders specified by the program's properties, or if each dialog box remembers its own folder.

RenderEngine specifies the engine to use for renderings.

RenderPrefsState (r/o) reports on the status of the Rendering Preferences.

ReportError toggles sending error reports to the programmers.

RevCloudCreateMode (The program does not support presetting default mode for revision clouds.)

RevCloudGrips (The program uses only one grip for each revision cloud.)

RibbonState (r/o) reports whether the ribbon is displayed.

RoamableRootPrefix (r/o) reports the path to the Roaming folder.

RtDisplay determines if some or all of raster images are displayed during realtime pan and zoom.

S

SavedDwgCheckSum specifies whether drawing files created by Teigha API are recognized as trusted drawings; not implemented.

SaveFidelity *(The program does not support annotative scaling.)*

SaveFile (r/o) reports automatic save file name.

SaveFilePath specifies path for automatic save files.

SaveFilterIndex presets the default drawing file format for the SaveAs command by setting its index number for the Files of Type droplist of the SaveFile dialog box.

SaveName (r/o) reports drawing's save-as file name.

SaveRoundtrip determines whether entities unknown to DWG R14 are stored in R14 DWG drawing files.

SaveTime specifies automatic save interval, in minutes.

Screen... *(The program does not support the side screen menu.)*

ScreenSize (r/o) reports current viewport size, in pixels.

Sdi *(The program does not support single document mode.)*

SelectBox specifies the size of the square pickbox cursor when selecting entities; same as the PickBox variable.

SelectionAnnoDisplay specifies whether annotative entities, if selected, show all representations of the annotation scales assigned to them, or only the original scale.

Selection... *(The program does not support colored selection areas.)*

SelectionPreview specifies selection preview.

SelectSimilarMode determines which properties must match for an entity of the same type to be selected with the SelectMatching command.

SerNumber (r/o)reports the serial number of the user's licence.

SetByLayerMode specifies the properties used by the SetByLayer command: color, linetypes, lineweights, materials, plot styles, and transparency.

ShadEdge determines the color of edges and faces displayed by ShadeMode command.

ShadeDif specifies the ambient lighting of ShadeMode command.

ShadowPlaneLocation *(The program does not support shadow planes.)*

ShapeAlt specifies the SHX shape file to substitute for references to unavailable shapes found in the current drawing.

ShortcutMenu specifies display of shortcut menus.

ShowHist *(The program does not have solid history.)*

ShowHyperlinkCursor toggles the display of the hyperlink cursor and tooltip, which appear when the cursor moves over entities with hyperlinks.

ShowLayerUsage toggles layer-usage icons in Layers dialog box.

ShpName specifies default shape name.

SigWarn controls whether the Digital Signature Verification dialog box appears each time you attach external drawings with valid digital signatures.

SketchInc specifies Sketch command's recording increment.

SkPoly toggles sketch line mode.

SkyStatus *(The program does not display skies.)*

SnapAng specifies rotation angle for snap and grid.

SnapBase specifies current origin for snap and grid.

SnapIsoPair specifies current isometric drawing plane.

SnapMode toggles snap mode.

SnapStyl toggles snap style.

SnapType toggles snap for current viewport.

SnapUnit specifies x,y spacing of snap distances.

SolidCheck toggles solids validation.

SolidHist *(The program does not have solid history.)*

SortEnts determines entity display order.

SpaceSwitch toggles between allowing and preventing switching between model and paper space inside viewports with a double-click.

SplFrame toggles display of frames on polylines and meshes.

SplineSegs specifies number of line segments that define splined polylines.

SplineType specifies type of spline curve.

SplPeriodic specifies the method for closing splines.

Ss... *(The program does not support sheet sets.)*

StandardsViolation toggles the use of standards checking in drawings.

Startup *(The program does not have a Startup dialog box.)*

Step... *(The program does not support animations.)*

Sun... *(The program does not have a Sun light.)*

SurfTab1 specifies density of m-direction surfaces and meshes.

SurfTab2 specifies density of n-direction surfaces and meshes.

SurfType specifies smoothing of surfaces by EditPolyline command.

SurfU specifies surface density in m-direction.

SurfV specifies surface density in n-direction.

SysCodePage (r/o) reports system code page; set by operating system.

T

TabDeviceMode toggles tablet between tablet and digitizer modes (for OEMs only)

TableIndicator toggles the display of the display of row numbers and column letters when editing Table cells in-place.

TableToolbar controls the display of the Edit Table toolbar that assists editing table cells in-place.

TabMenuName (r/o) reports the name of the current tablet menu (for OEMs only).

TabMode toggles tablet mode (for OEMs only).

Target (r/o) reports the x,y,z coordinates of the target of the current viewpoint.

Taskbar toggles the display of each drawing as a button on the Windows taskbar.

TbCustomize *(The program does not prevent toolbars from being customized.)*

TDCreate (r/o) reports date and time that drawing was created.

TDInDwg (r/o) reports duration since drawing was loaded.

TDuCreate (r/o) reports universal date and time when drawing was created.

TDUpdate (r/o) reports date and time of last update.

TDUsrTimer (r/o) reports decimal time elapsed by user-timer.

TDuUpdate (r/o) reports universal date and time of last save.

TempOverrides toggles temporary overrides, such as snap modes.

TempPrefix (r/o) specifies the path to the folder set by the Temp variable.

TextAlignMode specifies the default alignment for aligned text.

TextEd reports the status of the text editor.

TextEval toggles how LISP is evaluated during the -SimpleNote command.

Text... *(The program does not support text plotting parameters.)*

TextSize specifies default height of text.

TextStyle specifies default name of text style.

Thickness specifies default entity thickness.

ThumbSize toggles inclusion of preview images in drawing files.

TileMode toggles view mode.

TileModeLightSwitch controls the synchronization of lighting in all model space viewports.

TimeZone *(The program does not support times zones for sun lights.)*

ToolTipMerge determines whether to merge EntitySnap (ESnap) tooltips and Quick Input tooltips into a single tooltip.

ToolTips toggles display of tooltips.

TouchScreenMode determines the stylus behavior when using a touchscreen with third-party software.

TpState *(The program does not have a Tools palette.)*

TraceWid specifies current width of traces.

TrackPath controls display of polar and entity snap tracking alignment paths.

TransparencyDisplay toggles the use of transparency in layers and entities.

Tray... *(The program does not have a tray.)*

Tree... *(The program does not use tree-style object optimization.)*

TrimMode toggles use of trims during Chamfer and Fillet commands.

TSpaceFac sets the line spacing distance specified as a text height factor for notes.

TSpaceType sets the type of line spacing used within notes:

TStackAlign sets the vertical alignment of stacked text.

TStackSize determines the height of stacked text (fractions) relative to the specified text height.

U

UcsAxisAng specifies default angle for rotating CCS around an axis (CCS = custom coordinate system).

UcsBase specifies name of CCS that defines origin and orientation of orthographic CCS settings.

UcsDetect determines whether the custom coordinate system aligns dynamically to 3D solid objects.

UcsFollow toggles views displayed with new CCSs.

UcsIcon specifies display of CCS icon.

UcsName (r/o) reports name of current CCS view.

UcsOrg (r/o) reports origin of current CCS.

UcsOrtho specifies whether related orthographic CCS settings are restored automatically.

UcsView toggles whether current CCS is saved with a named view.

UcsVp toggles whether CCS in active viewports remains fixed (locked) or changes (unlocked) to match CCS of current viewport.

UcsXDir (r/o) reports x-direction of current CCS.

UcsYDir (r/o) reports y-direction of current CCS.

UndoCtl (r/o) reports settings that control Undo actions.

UndoMarks (r/o) reports number of undo marks.

UndoZoomPanGrouping determines whether zoom and pan operations are grouped as a single operation.

UnitMode toggles type of units display.

UOsnap toggles entity snapping for referenced files.

UpdateThumbnail *(The program does not support sheet sets.)*

UserI1 thru **UserI5** store five user-definable integer variables.

UserR1 thru **UserR5** store five user-definable real variables.

UserS1 thru **UserS5** store five user-definable string variables; values are not saved.

V

ViewCtr (r/o) reports x, y, z coordinates of center of current view.

ViewDir (r/o) reports current view direction relative to CCS.

ViewMode (r/o) report viewing mode.

ViewSize (r/o) reports height of current view in drawing units.

ViewTwist (r/o) reports twist angle of current view.

VisRetain specifies xref drawing's layer settings.

VpMaximizedState *(The program will support maximized viewports in a future release.)*

Vs*... (The program does not support visual styles.)*

VsMax *(The program will support maximized viewports in a future release.)*

VsMin *(The program will support maximized viewports in a future release.)*

Vs... *(The program does not support visual styles.)*

VstaState defines the state of VSTA integration; 0 = VSTA successfully loaded.

VstaVisible toggles the visibility of the VSTA toolbar.

Vt... *(The program does not support variable speed zooming.)*

W

Whip... *(The program does not use the WHIP display driver.)*

WindowAreaColor *(The program does not fill windowed selection areas.)*

WipeoutFrame toggles the display and plotting of frames around wipeout entities.

WmfBkGnd specifies the background color of drawings exported to WMF files.

WmfForeGnd specifies the foreground color of drawings exported to WMF files.

WorldUcs (r/o) toggles matching of world coordinates with CCS.

WorldView toggles view during the RollView and ViewDirection commands.

WriteStat (r/o) reports whether *.dwg* file is read-only.

WsCurrent specifies the name of the current workspace.

X

XClipFrame toggles visibility of external reference clipping boundaries.

XEdit specifies whether the current drawing can be edited in-place when being referenced by another drawing.

XFadeCtl *(The program does not fade xrefs.)*

XLoadCtl *(The program does demand-load files.)*

XLoadPath specifies the path to referenced drawings.

XRefCtl *(The program does not create xref log files.)*

XRefNotify *(The program does not have notification balloons.)*

XrefOverride toggles whether layer properties on xrefs are treated as ByLayer or not.

XrefType specifies whether externally referenced drawings are attached as Attachments or Overlays.

Z

ZInScale specifies zoom factor for ZoomIn command.

ZoomFactor specifies zoom level via mouse wheel.

ZoomWheel switches zoom direction when mouse wheel is rotated forward.

ZOutScale specifies zoom factor for ZoomOut command.

#

3dDwfPrec specifies the precision of drawings exported in 3D DWF format.

3dMOptions sets options for the 3dconnexion 3D mouse.

3dMSensitivity sets the sensitivity for the 3dconnexion 3D mouse.

3dMVelocity sets the speed of the 3dconnexion 3D mouse.

_PkSer (r/o) reports the package serial number.

APPENDIX C

Command Aliases

In This Appendix

- Aliases sorted by command name
- Aliases sorted by alias name

The program allows you to enter abbreviations for command names, such as L for Line. The abbreviations are known as "aliases." Aliases let you to enter commands quickly, but also to enter the names of command employed by AutoCAD and other CAD systems. For example, you can enter **dimstyle** (from AutoCAD) to execute the program's DimensionStyle command, or enter the program's own alias for this command, **D**.

You customize aliases through the User Preferences section of the **Options** command's dialog box. The program stores aliases in the *alias.xml* file. It reads alias definition files from AutoCAD (*.pgp*) and IntelliCAD (*.ica*).

On the following pages, 750 aliases are listed alphabetically in two ways: first, by command name and then by alias name.

Aliases native to the program are shown in **boldface**.

Aliases Sorted by Command Names

Command Name	Aliases
A	
-actionmanager	-actmanager
actionrecord	actrecord
actionstop	actstop
activatelayer	actlay, laymcur
activatetablestyle	atbs, ct, ctablestyle
align	**al**
align3d	3dalign
angledimension	aldim, angledim, dan, dimang, dimangular
arc	**a**
arclengthdimension	dar, dimarc
areaboundary	**ab**, -ab, bo, boundary, bpoly
-areaboundary	**-bo**, -boundary
arender	render
attachdgn	dgnattach
-attachdgn	-dgnattach
attachdrawing	atdwg, attachdwg, xa, xattach
attachimage	atimg, attachimg, iat, imageattach
attachlink	atlnk, attachurl
attachpdf	pdfattach, pdfattach
-attachpdf	-pdfattach
autorebuild	regenauto
B	
baselinedimension	basedim, bldim, dba, dimbase, dimbaseline
blockattributeoutput	attout, battout
boxtext	btext, qtext
borrowlicense	licensetogo
browsedesignresources	adcnavigate
C	
ccs	ucs
centermark	cm, dce, dimcenter
chamfer	**cha**
changeelements	refset
check	**chk**, audit
circle	**c**
clean	**cl**, pu, purge
-clean	-pu, -purge
clipboardcopy	copyclip
clipdgn	dgnclip
clipimage	imageclip
clippdf	pdfclip
clipreference	**clip**, xc, xclip
clipviewport	vpclip
closecomponent	refclose
closetoolbar	palclose
cloud	revcloud
commandhistory	cmdhist, textscr
commandwindow	cli, cmdwin, commandline
ConstraintIcon	constraintbar
ConstraintOptions	constraintsettings
ConstraintType	dcform, dctype
continuedimension	cdim, dco, dimcont, dimcontinue
ConvertConstraint	dcconvert
convertprintstyles	convertpstyles
copy	**co,** cp
copy@	**cp@**, copybase
copyhistory	copyhist
csicon	ucsicon
csstyle	css, uc, ucsman
customize	**cui**, cust, to, toolbar
cut	cutclip
cylinder	cyl
D	
DcAligned	dcparallel
DcRadial	dcradius
delete	**del**, e, erase
DeleteConstraint	delconstraint
deletelayer	dellay, laydel
deletereferencepoints	delrps
designresources	adcenter, drs
detachimage	dimage, imagedetach
detachlink	detachurl
DimensionalConstraint	dimconstraint
diameterdimension	ddi, dimdia, dimdiameter
dimensionstyle	**d**, ddim, dimsty, dimstyle, dst
-dimensionstyle	-dimstyle
directionallight	distantlight
displayblockattributes	attdisp, dispblkatt, dsatt
displayimageframe	iframe, imageframe
displayorder	dr, draworder
displayquality	viewres
draftingoptions	**dop**, ds, dsettings, se
+draftingoptions	+dsettings

Command Name	Aliases
drawingbounds	bounds, limits
drawingproperties	dwgprops
drawingstandards	standards

E

edgemesh	edgesurf
editannotation	**ed**, ddedit, edanno, edittext, edt, textedit
editblock	be, bedit, eb, editcustomblock
-editblock	-be, -bedit
editblockattribute	ate, attedit, attxedit, ddatte, edblkatt
-editblockattribute	-ate, atte, -attedit, -edblkatt, qattedit
editblockattributedefinition	attdedit, edblattdef, edblkattdef
-editblockattributedefinition	-attdedit
editcomponent	refedit
editdimension	ded, dimed, dimedit, editdim
editdimensiontext	dimted, dimtedit, editdimtxt
EditEntityGroup	groupedit
edithatch	hatchedit, he
-edithatch	-hatchedit
editimage	-im, -image
editipblockattribute	attipedit
editlength	edlen, len, lengthen
EditMultiLeader	mle
editnote	edn, mtedit
editpolyline	edpl, edpline, pe, pedit, polyedit
editrichline	editrline, edrl, mledit
-editrichline	-mledit
EditScaleList	scalelistedit
-EditScaleList	-scalelistedit
editsolid	solidedit
edittable	edtbl, tabledit, tableedit
edittolerance	edittol, edtol, toledit
editvertex	editvtx, vtxedit
ellipse	**el**
-embeddrawing	-embeddwg, -xbind
enterpoint	entpt
entitygrips	**gr**, ddgrips, egrips,
entitygroup	**g**, group, egroup
-entitygroup	**-g**, -group
entitysnap	**es**, esnap, os, osnap
-entitysnap	**-es**, -os, -osnap
exit	quit
explode	x
ExplodeBlockX	burst
explodetext	txtexp
export	**exp**
exportbmp	bmpout
exportdrawing	acadwblockdialog, dwgout, exportdwg, w, wblock
-exportdrawing	-exportdwg, -w, -wblock
exportdxf	dxfout
exportemf	emfout, qwmfout
exporteps	psout
exportjpg	jpgout
exportpdf	pdfout
exportpng	pngout
exportsat	acisout, satout
exportsld	mslide, qsldout, sldout
exportsvg	svgout
extend	**ex**
extractblockattribute	attext
-extractblockattribute	-attext, -attout
extrude	**ext**

F

face	3dface, 3f
filemanager	explorer
fillarea	gradient
fillet	f
flip	**fl**
freezelayer	frzlay, layfrz
fullscreen	cleanscreenon, fscreen

G

GeometricContraint	geomconstraint
getarea	**ga**, aa, area
getdistance	di, dist, gd, getdist
GetMassProperties	massprop
getproperties	**gp**, getprops, li, list, ls
getstatus	gs, status
gettime	gt, time
getxy	**gxy,** id
gravity	aperture

H

hatch	**h**, bh, bhatch,
-hatch	**-h**,-bhatch, qhatch
hidecommandhistory	graphscr, hidecmdhist
hidecommandwindow	commandlinehide, hidecmdwin
hidedesignresources	adcclose
hidefullscreen	cleanscreenoff, hfscreen
hidelayer	hidelay, layoff
hidelightlist	hlights, lightlistclose

Command Name	Aliases
HideParameters	parametersclose
hideproperties	hideprops, prclose, propertiesclose
hidereferences	hiderefs, xrefclose
hideview	**hi**, hide, hview, qhide
horizontaldimension	dimhor, hdim

I

imagequality	iquality
import	imp
ImportDgn	dgnimport
importdwf	dwfattach, dwfin
importdxf	dxfin
importsat	acisin, satin
infiniteline	**il**, iline, xl, xline
insertblock	**i**, insert, symbol
-insertblock	**-i**, -insert, qinsert, -symbol
insertblockn	insblock, minsert
insertobject	insertobj
insertshape	inshape, insshape, shape
interfere	inf
intersect	in
isolatelayer	isolay, layiso
isometricgrid	isogrid, isoplane

J

joggeddimension	**jog**, dimjogged, djo, jogdim

L

layer	**la**
-layer	**-la**, qlayer
LayerPreview	laywalk
layersdgn	dgnlayers
layerspdf	pdflayers
leader	**lead**
line	l
lineardimension	dimlin, dimlinear, dli, ldim
linecolor	**col**, color, colour, lc, lcolor
-linecolor	-color, -lcolor
linescale	lscale, lts, ltscale, linetype, lstyle, lt, ltype, qlinetype
-linestyle	-linetype, loadltype, -lstyle, -lt, -ltype
lineweight	**lw**, lweight
-lineweight	-lweight
loadapplication	ap, appload
loadcustomization	cuiload, menuload
loadmenu	lmenu, menu, pulldown
loadscript	**scr**, lscript, script
loadshape	load
loadtoolbar	palette
locklayer	laylck, lcklay, llay

M

makeblock	**b**, block, bmake, mblck, partdef
-makeblock	**-b**, -block
makeblockattribute	att, attdef, mblkatt
-makeblockattribute	-att, -attdef, qattdef
makeflatsnapshot	flatshot
makeslidelibrary	makeslidelib
markdivisions	**div**, divide, mdiv
marklengths	**me**, measure, mlen
mask	wipeout
matchlayer	laymch, mlay
mesh	3dmesh
mirror	**mi**, qmirror
mirror3d	3dmirror, mi3d
modelmode	**mm**, ms, mspace
modify	-ch, change, mod, qpropedit
modifyproperties	chprop, modprops
move	**m**
movedimensiontext	dimtmove, movedimtxt
MultiLeader	mleader

N

nontargetlight	freespot
note	**n**, mt, mtext, nm t
-note	-mtext, qmtext, -t

O

obliquedimension	dimobl, obliquedim, odim
offset	**o**, qoffset
openreference	xopen
opentoolbar	palopen
openwebpage	browser, oweb
options	**op**, config
ordinatedimension	dimord, dimordinate, dor, orddim, ordinatedim
oscalc	cal, calc, wcalc
overridedimensionstyle	dimover, dimoverride, dov, overdims

P

packandgo	etransmit
-packandgo	-etransmit

Command Name	Aliases
pagelayout	pagesetup
pan	**p**, dpan, pandynamic
-pan	**-p**
pandown	pd
paralleldimension	dal, dimali, dimaligned, paralleldim, pdim
Parameters	constraintmanager
paste	pasteclip
paste@sourceposition	pasteorig
pasteasblock	pasteblock
pasteselected	pastespec
pattern	ar, array, pat
-pattern	-ar, -array, qarray
pattern3d	3a, 3darray, pat3d
pausescript	delay
planview	plan, pview
point	**po**, pt
pointformat	ddptype
polyface	pface
polygon	**pol**, pgon
polyline	**pl**, pline
polyline3d	**pl3**, 3dpoly, 3p, pline3d
preview	**pre**
print	dwfout, exportdwf, plot
-print	-plot
printstamp	-plotstamp, pstamp
printstyle	plotstyle, pstyle
-printstyle	-plotstyle
properties	**pr**, ch, ddmodify, mo, props
propertypainter	ma, matchprop, paint

Q

QuickGroup	g, group
quickprint	qprint
~~quickmodify~~	~~mocoro~~
QuickUngroup	ungroup

R

radiusdimension	dimrad, dimradius, dra, radiusdim, rdim
rebuild	**re**, regen
rebuildall	**rea**, regenall
rebuildfield	updatefield
rectangle	**rec**, rect, rectang
redon	mredo
references	er, externalreferences, im, image, refs, xlink, xr, xref
-references	qxlink, qxref, -xr, - xref
refresh	**r**, redraw
refreshaliases	realias, reinit
refreshall	**ra**, redrawall
region	**reg**
relatedimension	dimreassociate, dre, redim
rename	**ren**, rn
-rename	-ren
repeat	multiple
replacedimensiontext	dimtnew, replacedimtxt
replacenew	rnew
replaceopen	ropen
replacerecover	rrecover
resetdimensiontext	dimthome, resetdimtext
resumescript	resume
revolve	rev
revolvedmesh	revsurf
richline	**rl**, ml, mline
richlinestyle	**rls**, mlstyle, rlinestyle, rlstyle
ring	do, donut
rollview	3do, 3dorbit, orbit
rotate	**ro**
rotate3d	ro3d
rotateddimension	dimrot
rotatedimensiontext	dimtrot, rodimtext
ruledmesh	revmesh, rulesurf

S

save	qsave
savecomponent	refsave
scale	**sc**
scriptn	rscript
section	sec
selectionfilter	**sf**, fi, filter
setvariable	set, setvar
shadeview	**sha**, shademode, -shademode, sview
sheet	layout, lo
-sheet	-layout
sheetmode	**sm**, ps, pspace,
showlayers	layon
simplenote	dt, ddtext, snote, text
-simplenote	dtext, -snote, text, -text
simplepolygon	ngon, spoly
slice	sl
smartbmp	qbmpout
smartcalculator	qc, quickcalc, smartcalc
smartleader	**le**, qleader
smartnew	qnew, snew

Command Name	Aliases
smartopen	qopen, sop
smartselect	sleader, ss, qselect
snap	**sn**
solid	**so**
spellcheck	**sp**, spell
spline	**spl**
split	br, break
stretch	s
subtract	su
switcharea	switch, tlmode

T

table	**tb**
tablestyle	**ts**
tablet	ta
tabulatedmesh	tabmesh, tabsurf
targetpointlight	targetpoint
textstyle	**st**, style, txs
-textstyle	-style
thawlayers	laythw, thawlay
tilemode	ti, tm
toactivelayer	2alay, laycur
tolerance	**tol**
toolbars	tbconfig
torus	tor
trim	**tr**

U

undelete	oops, undel
undolayer	layerp
undon	undo
union	uni
unisolatelayer	layuniso
unitsystem	**un**, units, unitsys, us
-unitsystem	**-un**, -units
unloadcustomization	cuiunload, menuunload
unlocklayer	layulk, unla
unrelatedimension	dda, dimdisassociate, undim, unrelatedim
updateblockattributes	attsync

Command Name	Aliases

V

verifydimensions	dimregen
verifystandards	checkstandards
verticaldimension	dimver, vdim
viewdirection	vdirect, -vp, vpoint
viewport	mv, mview
-viewport	qviewport, -vports
viewportlayer	vplayer
views	**v**, view
-views	**-v**, qview, -view
viewslide	qvslide, vslide
viewtiles	viewports, vports, vtiles
+viewtiles	+vports
voicenote	vnote

W

wedge	we
weld	j, join
windows	syswindows

Z

zoom	**z**
zoomback	**zb**
zoomdynamic	**zd**, dzoom, zoomdyn
zoomfactor	**zfa**
zoomfit	**zf**
zoomin	**zi**
zoomout	**zo**
zoomwindow	**zw**, zoomarea, zoomwin
zplane	elev

#

3dmove	3m
3pointangledimension	3padim, dim3ap, dima3p
4pointangledimension	4padim, dim4ap, dima4p

Aliases Sorted by Alias Names

Alias	Command Name
A	
a	arc
aa	getarea
ab	areaboundary
-ab	areaboundary
acadwblockdialog	exportdrawing
acisin	importsat
acisout	exportsat
actlay	activatelayer
-actmanager	-actionmanager
actrecord	actionrecord
actstop	actionstop
adcclose	hidedesignresources
adcenter	designresources
adcnavigate	browsedesignresources
al	align
aldim	angledimension
angledim	angledimension
ap	loadapplication
aperture	gravity
appload	loadapplication
ar	pattern
-ar	-pattern
area	getarea
array	pattern
-array	-pattern
atbs	activatetablestyle
atdwg	attachdrawing
ate	editblockattribute
-ate	-editblockattribute
atimg	attachimage
atlnk	attachlink
att	makeblockattribute
-att	-makeblockattribute
attachdwg	attachdrawing
attachimg	attachimage
attachurl	attachlink
attdedit	editblockattributedefinition
-attdedit	-editblockattributedefinition
attdef	makeblockattribute
-attdef	-makeblockattribute
attdisp	displayblockattributes
atte	-editblockattribute
attedit	editblockattribute
-attedit	-editblockattribute
-attext	-extractblockattribute
attext	extractblockattribute
attipedit	editipblockattribute
attout	blockattributeoutput
-attout	-extractblockattribute
attsync	updateblockattributes
attxedit	editblockattribute
audit	check
B	
b	makeblock
-b	-makeblock
basedim	baselinedimension
battout	blockattributeoutput
be	editblock
-be	-editblock
bedit	editblock
-bedit	editblock
bh	hatch
bhatch	hatch
-bhatch	-hatch
bldim	baselinedimension
block	makeblock
-block	-makeblock
bmake	makeblock
bmpout	exportbmp
bo	areaboundary
-bo	-areaboundary
boundary	areaboundary
-boundary	-areaboundary
bounds	drawingbounds
bpoly	areaboundary
br	split
break	split
browser	openwebpage
btext	boxtext
burst	ExplodeBlockX
C	
c	circle
cal	oscalc
calc	oscalc
cdim	continuedimension
-ch	modify
ch	properties

Alias	Command Name
cha	chamfer
change	modify
checkstandards	verifystandards
chk	check
chprop	modifyproperties
cl	clean
cleanscreenoff	hidefullscreen
cleanscreenon	fullscreen
cli	commandwindow
clip	clipreference
cm	centermark
cmdhist	commandhistory
cmdwin	commandwindow
co	copy
col	linecolor
color	linecolor
-color	-linecolor
colour	linecolor
commandline	commandwindow
commandlinehide	hidecommandwindow
config	options
constraintbar	ConstraintIcon
constraintmanager	Parameters
constraintsettings	ConstraintOptions
convertpstyles	convertprintstyles
copybase	copy@
copyclip	clipboardcopy
copyhist	copyhistory
cp	copy
cp@	copy@
css	csstyle
ct	activatetablestyle
ctablestyle	activatetablestyle
cui	customize
cuiload	loadcustomization
cuiunload	unloadcustomization
cust	customize
cutclip	cut
cyl	cylinder

D

Alias	Command Name
d	dimensionstyle
dal	paralleldimension
dan	angledimension
dar	arclengthdimension
dba	baselinedimension
dcconvert	ConvertConstraint
dce	centermark
dcform	ConstraintType
dco	continuedimension
dcparallel	DcAligned
dcradius	DcRadial
dctype	ConstraintType
dda	unrelatedimension
ddatte	editblockattribute
ddedit	editannotation
ddgrips	entitygrips
ddi	diameterdimension
ddim	dimensionstyle
ddmodify	properties
ddptype	pointformat
ddtext	simplenote
ded	editdimension
del	delete
delay	pausescript
delconstraint	DeleteConstraint
dellay	deletelayer
delrps	deletereferencepoints
detachurl	detachlink
dgnattach	attachdgn
-dgnattach	-attachdgn
dgnclip	clipdgn
dgnimport	ImportDgn
dgnlayers	layersdgn
di	getdistance
dim3ap	3pointangledimension
dim4ap	4pointangledimension
dima3p	3pointangledimension
dima4p	4pointangledimension
dimage	detachimage
dimali	paralleldimension
dimaligned	paralleldimension
dimang	angledimension
dimangular	angledimension
dimarc	arclengthdimension
dimbase	baselinedimension
dimbaseline	baselinedimension
dimcenter	centermark
dimconstraint	DimensionalConstraint
dimcont	continuedimension
dimcontinue	continuedimension
dimdia	diameterdimension
dimdiameter	diameterdimension
dimdisassociate	unrelatedimension
dimed	editdimension

Alias	Command Name
dimedit	editdimension
dimhor	horizontaldimension
dimjogged	joggeddimension
dimlin	lineardimension
dimlinear	lineardimension
dimobl	obliquedimension
dimord	ordinatedimension
dimordinate	ordinatedimension
dimover	overridedimensionstyle
dimoverride	overridedimensionstyle
dimrad	radiusdimension
dimradius	radiusdimension
dimreassociate	relatedimension
dimregen	verifydimensions
dimrot	rotateddimension
dimsty	dimensionstyle
dimstyle	dimensionstyle
-dimstyle	-dimensionstyle
dimted	editdimensiontext
dimtedit	editdimensiontext
dimthome	resetdimensiontext
dimtmove	movedimensiontext
dimtnew	replacedimensiontext
dimtrot	rotatedimensiontext
dimver	verticaldimension
dispblkatt	displayblockattributes
dist	getdistance
distantlight	directionallight
div	markdivisions
divide	markdivisions
djo	joggeddimension
dli	lineardimension
do	ring
donut	ring
dop	draftingoptions
dor	ordinatedimension
dov	overridedimensionstyle
dpan	pan
dr	displayorder
dra	radiusdimension
draworder	displayorder
dre	relatedimension
drs	designresources
ds	draftingoptions
dsatt	displayblockattributes
dsettings	draftingoptions
+dsettings	+draftingoptions
dst	dimensionstyle
dt	simplenote
dtext	-simplenote
dwfattach	importdwf
dwfin	importdwf
dwfout	print
dwgout	exportdrawing
dwgprops	drawingproperties
dxfin	importdxf
dxfout	exportdxf
dzoom	zoomdynamic

E

e	delete
eb	editblock
ed	editannotation
edanno	editannotation
edblattdef	editblockattributedefinition
-edblkatt	-editblockattribute
edblkatt	editblockattribute
edblkattdef	editblockattributedefinition
editcustomblock	editblock
edgesurf	edgemesh
editdim	editdimension
editdimtxt	editdimensiontext
editrline	editrichline
edittext	editannotation
edittol	edittolerance
editvtx	editvertex
edlen	editlength
edn	editnote
edpl	editpolyline
edpline	editpolyline
edrl	editrichline
edt	editannotation
edtbl	edittable
edtol	edittolerance
egrips	entitygrips
egroup	entitygroup
el	ellipse
elev	zplane
-embeddwg	-embeddrawing
emfout	exportemf
entpt	enterpoint
er	references
erase	delete
es	entitysnap
-es	-entitysnap
esnap	entitysnap
etransmit	packandgo
-etransmit	-packandgo

Alias	Command Name
ex	extend
exp	export
explorer	filemanager
exportdwf	print
exportdwg	exportdrawing
-exportdwg	-exportdrawing
ext	extrude
externalreferences	references

F

f	fillet
fi	selectionfilter
filter	selectionfilter
fl	flip
flatshot	makeflatsnapshot
freespot	nontargetlight
frzlay	freezelayer
fscreen	fullscreen

G

g	entitygroup
-g	-entitygroup
g	QuickGroup
ga	getarea
gd	getdistance
geomconstraint	GeometricContraint
getdist	getdistance
getprops	getproperties
gp	getproperties
gr	entitygrips
gradient	fillarea
graphscr	hidecommandhistory
-group	-entitygroup
group	QuickGroup
group	entitygroup
groupedit	EditEntityGroup
gs	getstatus
gt	gettime
gxy	getxy

H

-h	-hatch
h	hatch
hatchedit	edithatch
-hatchedit	-edithatch
hdim	horizontaldimension
he	edithatch
hfscreen	hidefullscreen
hi	hideview
hide	hideview
hidecmdhist	hidecommandhistory
hidecmdwin	hidecommandwindow
hidelay	hidelayer
hideprops	hideproperties
hiderefs	hidereferences
hlights	hidelightlist
hview	hideview

I

i	insertblock
-i	-insertblock
iat	attachimage
id	getxy
iframe	displayimageframe
il	infiniteline
iline	infiniteline
-im	editimage
im	references
-image	editimage
image	references
imageattach	attachimage
imageclip	clipimage
imagedetach	detachimage
imageframe	displayimageframe
imp	import
in	intersect
inf	interfere
insblock	insertblockn
insert	insertblock
-insert	-insertblock
insertobj	insertobject
inshape	insertshape
insshape	insertshape
iquality	imagequality
isogrid	isometricgrid
isolay	isolatelayer
isoplane	isometricgrid

J

j	weld
jog	joggeddimension
jogdim	joggeddimension
join	weld
jpgout	exportjpg

Alias	Command Name

L

l	line
-la	-layer
la	layer
laycur	toactivelayer
laydel	deletelayer
layerp	undolayer
layfrz	freezelayer
layiso	isolatelayer
laylck	locklayer
laymch	matchlayer
laymcur	activatelayer
layoff	hidelayer
layon	showlayers
layout	sheet
-layout	-sheet
laythw	thawlayers
layulk	unlocklayer
layuniso	unisolatelayer
laywalk	LayerPreview
lc	linecolor
lcklay	locklayer
lcolor	linecolor
-lcolor	-linecolor
ldim	lineardimension
le	smartleader
lead	leader
len	editlength
lengthen	editlength
li	getproperties
licensetogo	borrowlicense
lightlistclose	hidelightlist
limits	drawingbounds
linetype	linescale
-linetype	-linestyle
list	getproperties
llay	locklayer
lmenu	loadmenu
lo	sheet
load	loadshape
loadltype	-linestyle
ls	getproperties
lscale	linescale
lscript	loadscript
lstyle	linescale
-lstyle	-linestyle
lt	linescale
-lt	-linestyle
lts	linescale
ltscale	linescale
ltype	linescale
-ltype	-linestyle
lw	lineweight
lweight	lineweight
-lweight	-lineweight

M

m	move
ma	propertypainter
makeslidelib	makeslidelibrary
massprop	GetMassProperties
matchprop	propertypainter
mblck	makeblock
mblkatt	makeblockattribute
mdiv	markdivisions
me	marklengths
measure	marklengths
menu	loadmenu
menuload	loadcustomization
menuunload	unloadcustomization
mi	mirror
mi3d	mirror3d
minsert	insertblockn
ml	richline
mlay	matchlayer
mle	EditMultiLeader
mleader	MultiLeader
mledit	editrichline
-mledit	-editrichline
mlen	marklengths
mline	richline
mlstyle	richlinestyle
mm	modelmode
mo	properties
mocoro	quickmodify
mod	modify
modprops	modifyproperties
movedimtxt	movedimensiontext
mredo	redon
ms	modelmode
mslide	exportsld
mspace	modelmode
mt	note
mtedit	editnote
mtext	note
-mtext	-note

Alias	Command Name
multiple	repeat
mv	viewport
mview	viewport

N

n	note
ngon	simplepolygon
nm	note

O

o	offset
obliquedim	obliquedimension
odim	obliquedimension
oops	undelete
op	options
orbit	rollview
orddim	ordinatedimension
ordinatedim	ordinatedimension
os	entitysnap
-os	-entitysnap
osnap	entitysnap
-osnap	-entitysnap
overdims	overridedimensionstyle
oweb	openwebpage

P

p	pan
-p	-pan
pagesetup	pagelayout
paint	propertypainter
palclose	closetoolbar
palette	loadtoolbar
palopen	opentoolbar
pandynamic	pan
paralleldim	paralleldimension
parametersclose	HideParameters
partdef	makeblock
pasteblock	pasteasblock
pasteclip	paste
pasteorig	paste@sourceposition
pastespec	pasteselected
pat	pattern
pat3d	pattern3d
pd	pandown
pdfattach	attachpdf
pdfattach	attachpdf
-pdfattach	-attachpdf
pdfclip	clippdf
pdflayers	layerspdf
pdfout	exportpdf
pdim	paralleldimension
pe	editpolyline
pedit	editpolyline
pface	polyface
pgon	polygon
pl	polyline
pl3	polyline3d
plan	planview
pline	polyline
pline3d	polyline3d
plot	print
-plot	-print
-plotstamp	printstamp
plotstyle	printstyle
-plotstyle	-printstyle
pngout	exportpng
po	point
pol	polygon
polyedit	editpolyline
pr	properties
prclose	hideproperties
pre	preview
propertiesclose	hideproperties
props	properties
ps	sheetmode
psout	exporteps
pspace	sheetmode
pstamp	printstamp
pstyle	printstyle
pt	point
pu	clean
-pu	-clean
pulldown	loadmenu
purge	clean
-purge	-clean
pview	planview
Q	
qarray	-pattern
qattdef	-makeblockattribute
qattedit	-editblockattribute
qbmpout	smartbmp
qc	smartcalculator
qhatch	-hatch
qhide	hideview
qinsert	-insertblock
qlayer	-layer

Alias	Command Name
qleader	smartleader
qlinetype	linescale
qmirror	mirror
qmtext	-note
qnew	smartnew
qoffset	offset
qopen	smartopen
qprint	quickprint
qpropedit	modify
qsave	save
qselect	smartselect
qsldout	exportsld
qtext	boxtext
quickcalc	smartcalculator
quit	exit
qview	-views
qviewport	-viewport
qvslide	viewslide
qwmfout	exportemf
qxlink	-references
qxref	-references

R

r	refresh
ra	refreshall
radiusdim	radiusdimension
rdim	radiusdimension
re	rebuild
rea	rebuildall
realias	refreshaliases
rec	rectangle
rect	rectangle
rectang	rectangle
redim	relatedimension
redraw	refresh
redrawall	refreshall
refclose	closecomponent
refedit	editcomponent
refs	references
refsave	savecomponent
refset	changeelements
reg	region
regen	rebuild
regenall	rebuildall
regenauto	autorebuild
reinit	refreshaliases
ren	rename
-ren	-rename
render	arender
replacedimtxt	replacedimensiontext
resetdimtext	resetdimensiontext
resume	resumescript
rev	revolve
revcloud	cloud
revmesh	ruledmesh
revsurf	revolvedmesh
rl	richline
rlinestyle	richlinestyle
rls	richlinestyle
rlstyle	richlinestyle
rn	rename
rnew	replacenew
ro	rotate
ro3d	rotate3d
rodimtext	rotatedimensiontext
ropen	replaceopen
rrecover	replacerecover
rscript	scriptn
rulesurf	ruledmesh

S

s	stretch
satin	importsat
satout	exportsat
sc	scale
scalelistedit	EditScaleList
-scalelistedit	-EditScaleList
scr	loadscript
script	loadscript
se	draftingoptions
sec	section
set	setvariable
setvar	setvariable
sf	selectionfilter
sha	shadeview
shademode	shadeview
-shademode	shadeview
shape	insertshape
sl	slice
sldout	exportsld
sleader	smartselect
sm	sheetmode
smartcalc	smartcalculator
sn	snap
snew	smartnew
snote	simplenote

Alias	Command Name
-snote	-simplenote
so	solid
solidedit	editsolid
sop	smartopen
sp	spellcheck
spell	spellcheck
spl	spline
spoly	simplepolygon
ss	smartselect
st	textstyle
standards	drawingstandards
status	getstatus
style	textstyle
-style	-textstyle
su	subtract
svgout	exportsvg
sview	shadeview
switch	switcharea
symbol	insertblock
-symbol	-insertblock
syswindows	windows

T

t	note
-t	-note
ta	tablet
tabledit	edittable
tableedit	edittable
tabmesh	tabulatedmesh
tabsurf	tabulatedmesh
targetpoint	targetpointlight
tb	table
tbconfig	toolbars
text	-simplenote
text	simplenote
-text	-simplenote
textedit	editannotation
textscr	commandhistory
thawlay	thawlayers
ti	tilemode
time	gettime
tlmode	switcharea
tm	tilemode
to	customize
tol	tolerance
toledit	edittolerance
toolbar	customize
tor	torus
tr	trim
ts	tablestyle
txs	textstyle
txtexp	explodetext

U

uc	csstyle
ucs	ccs
ucsicon	csicon
ucsman	csstyle
un	unitsystem
-un	-unitsystem
undel	undelete
undim	unrelatedimension
undo	undon
ungroup	QuickUngroup
uni	union
units	unitsystem
-units	-unitsystem
unitsys	unitsystem
unla	unlocklayer
unrelatedim	unrelatedimension
updatefield	rebuildfield
us	unitsystem

V

v	views
-v	-views
vdim	verticaldimension
vdirect	viewdirection
view	views
-view	-views
viewports	viewtiles
viewres	displayquality
vnote	voicenote
-vp	viewdirection
vpclip	clipviewport
vplayer	viewportlayer
vpoint	viewdirection
+vports	+viewtiles
-vports	-viewport
vports	viewtiles
vslide	viewslide
vtiles	viewtiles
vtxedit	editvertex

Alias	Command Name

W

w	exportdrawing
-w	-exportdrawing
wblock	exportdrawing
-wblock	-exportdrawing
wcalc	oscalc
we	wedge
wipeout	mask

X

x	explode
xa	attachdrawing
xattach	attachdrawing
-xbind	-embeddrawing
xc	clipreference
xclip	clipreference
xl	infiniteline
xline	infiniteline
xlink	references
xopen	openreference
xr	references
-xr	-references
xref	references
-xref	-references
xrefclose	hidereferences

Z

z	zoom
zb	zoomback
zd	zoomdynamic
zf	zoomfit
zfa	zoomfactor
zi	zoomin
zo	zoomout
zoomarea	zoomwindow
zoomdyn	zoomdynamic
zoomwin	zoomwindow
zw	zoomwindow

#

2alay	toactivelayer
3a	pattern3d
3dalign	align3d
3darray	pattern3d
3dface	face
3dmesh	mesh
3dmirror	mirror3d
3do	rollview
3dorbit	rollview
3dpoly	polyline3d
3f	face
3m	3dmove
3p	polyline3d
3padim	3pointangledimension
^padim	4pointangledimension

Notes

APPENDIX D

Keyboard Shortcuts

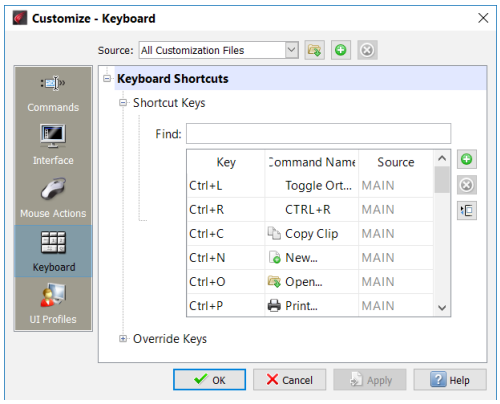
In This Appendix

- Keyboard shortcuts
- Temporary overrides

You can execute common commands through keyboard shortcuts. Once memorized, shortcuts are the fastest ways to enter commands. For example, to quickly save the drawing, press **Ctrl+S**. You can temporarily override entity snaps and drawing modes with a key. For instance, to turn on ortho mode temporarily, hold down the **Shift** key.

In this appendix, the shortcut keystrokes are listed twice, once for the program running on Linux and Windows, and then again for the program on MacOSX. New keyboard shortcuts are shown in red.

To customize keyboard shortcuts in the program, enter the **Customize** command, and then choose **Keyboard**.



Changing the meaning of shortcut keystrokes in the Customize dialog box

Keyboard Shortcuts for Linux and Windows

The following tables list keyboard shortcuts for the Windows and Linux versions:

Function	Keyboard Shortcut	Command Executed
Cleanscreen toggle	**Ctrl+0**	Fullscreen, HideFullscreen
Properties palette	**Ctrl+1**	Properties, HideProperties
Drawing Resources palette	**Ctrl+2**	DrawingResources, HideDrawingResources
Cloud Storage palette	**Ctrl+3**	CloudStorage
Command bar	**Ctrl+9**	CommandWindow, HideCommandWindow
Select all entities	**Ctrl+A**	SelectAll
Toggle group selection mode	**Ctrl+Shift+A**	TogglePickStyle
Snap toggle	**Ctrl+B**	Snap
Copy to Clipboard	**Ctrl+C**	ClipboardCopy
Copy with reference point	**Ctrl+Shift+C**	Copy@
Toggle dynamic CS mode	Ctrl+D	UcsDetect
Find and replace text	**Ctrl+F**	Find
Grid toggle	**Ctrl+G**	Grid
Toggle group selection mode	**Ctrl+H**	TogglePickStyle
Toggle coordinate display mode	Ctrl+I	Coords
Hyperlinks dialog box	**Ctrl+K**	Hyperlink
Ortho mode toggle	**Ctrl+L**	Ortho
New drawing	**Ctrl+N**	New
Open drawing	**Ctrl+O**	Open
Print dialog box	**Ctrl+P**	Print
Quit	**Ctrl+Q**	Exit
Switch viewports	**Ctrl+R**	^R
Save drawing	**Ctrl+S**	Save
Save drawing as	**Ctrl+Shift+S**	SaveAs
Paste from Clipboard	**Ctrl+V**	Paste
Paste as block	**Ctrl+Shift+V**	PasteAsBlock
Cut to Clipboard	**Ctrl+X**	Cut
Redo	**Ctrl+Y**	Redo
Undo	**Ctrl+Z**	U
Erase entities	**Del**	Delete
Cancel current command	**Esc**	^C
Enforce ortho mode	**Shift**	Shift

Function Keys for Linux and Windows

The following keystrokes cannot be customized by ARES. Those marked with an asterisk (*) are defined by the operating system.

Function	Keyboard Shortcut	Command Executed
* Help	**F1**	Help
Text screen toggle	**F2**	CommandHistory, HideCommandHistory
Entity snap toggle	**F3**	-ESnap
Close program	**Alt+F4**	Close
* Close drawing	**Ctrl+F4**	Close
Isometric toggle	**F5**	IsometricGrid
Dynamic CCS toggle	**F6**	UcsDetect
* Switch to next drawing	**Ctrl+F6**	...
Grid toggle	**F7**	^G
Ortho toggle	**F8**	^L
Snap toggle	**F9**	^B
Polar mode toggle	**F10**	...
Entity tracking toggle	**F11**	...
Quick input toggle	**F12**	...

Temporary Overrides

Although temporary override keystrokes appear in the Customize dialog box, they do not work in ARES. These keystrokes are used to modify entity snaps and other drawing modes temporarily during drawing and editing commands.

Keyboard Shortcuts for Mac OS X

The following tables list keyboard shortcuts for the Mac OS X version.

Mac OS X Function	Keyboard Shortcut	Command Executed
Cleanscreen toggle	**Cmd+0**	Fullscreen, HideFullscreen
Properties palette	**Cmd+1**	Properties, HideProperties
Drawing Resources palette	**Cmd+2**	DrawingResources, HideDrawingResources
Cloud Storage palette	**Cmd+3**	CloudStorage
Command window	**Cmd+9**	CommandWindow, HideCommandWindow
Select all entities	**Cmd+A**	SelectAll
Toggle group selection mode	**Cmd+Shift+A**	TogglePickStyle
Snap toggle	**Cmd+B**	Snap
Copy to Clipboard	**Cmd+C**	ClipboardCopy
Copy with reference point	**Cmd+Shift+C**	Copy@
Find and replace text	**Cmd+F**	Find
Grid toggle	**Cmd+G**	Grid
Toggle group selection mode	**Cmd+H**	TogglePickStyle
Toggle coordinate display mode	**Cmd+I**	Coords
Hyperlinks dialog box	**Cmd+K**	Hyperlink
Ortho toggle	**Cmd+L**	Ortho
New drawing	**Cmd+N**	New
Open drawing	**Cmd+O**	Open
Print dialog box	**Cmd+P**	Print
Quit	**Cmd+Q**	Exit
Switch viewports	**Cmd+R**	^R
Save drawing	**Cmd+S**	Save
Save drawing as	**Cmd+Shift+S**	SaveAs
Paste from Clipboard	**Cmd+V**	Paste
Paste as block	**Cmd+Shift+V**	PasteAsBlock
Cut to Clipboard	**Cmd+X**	Cut
Redo	**Cmd+Y**	Redo
Undo	**Cmd+Z**	U
Erase entities	**fn+Delete**	Delete
Cancel current command	**Esc**	^C
Toggle ortho mode	**Shift**	Ortho
Display online help	**Cmd+?**	Help

Function Keys for Mac OS X

The following keystrokes cannot be customized by ARES. Those marked with an asterisk (*) are defined by the operating system.

Mac OS X Function	Keyboard Shortcut	Command Executed
* Help	**F1**	Help
Toggle command history	**F2**	CommandWindow, HideCommandWindow
Toggle entity snaps	**F3**	Esnap
Switch to next isometric grid	**F5**	IsometericGrid
Dynamic CCS	**F6**	UcsDetect
Toggle grid display	**F7**	Grid
Toggle ortho mode	**F8**	Ortho
Toggle snap mode	**F9**	Snap
Toggle polar mode	**F10**	...
* Hide/show ARES	**F11**	...
Toggle quick input	**F12**	...

Notes

APPENDIX E

Button Clicks & Double Clicks

In This Appendix

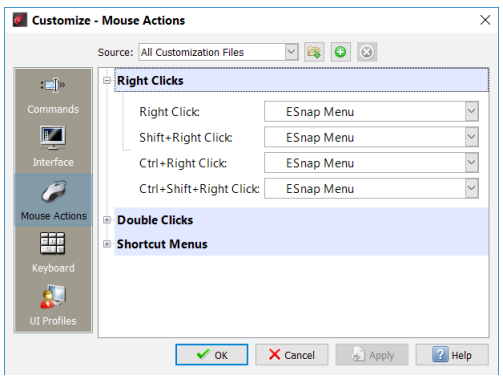
- Mouse buttons
- Double-click actions
- 3D mouse buttons
- Mouse gestures
- Touch gestures

When you click a mouse button, the program executes a command or displays a shortcut menu. The table on the fol-lowing page applies to buttons on mice.

The program also supports the following input methods:

- » Double-clicking entities to access the Properties command to modify the properties of the entity; in some cases, a command related o the entity is executed
- » Three-D mice from 3Dconnexion, which are used to navigate 3D space in conjunction with regular mice
- » Mouse gestures, which execute commands based on how the mouse is move with the right button held down
- » Some OEM versions of the program support digitizing tablets and pucks
- » Some touchscreen gestures are recognized

To customize mouse buttons and double-click actions in the program, you enter the **Customize** command, and then choose **Mouse Actions**. (Button #1 always picks and so can never be changed.) The buttons of 3D mice are customized by the utility software provided by 3dConnexion.



Customizing mouse button actions with the Customize dialog box

Mouse Buttons

Most mouse buttons activate context menus. Button #1 (right mouse button) cannot be customized. The program does not support digitizing tablets.

Primary Buttons

Button	Action
1 (left)	Picks entities (cannot be changed)
2 (right)	Displays the context menu
3 (middle)	Displays the context menu

Shift+Buttons

Button	Action
2 (right)	Displays entity snap menu
3 (middle)	Orbits drawing transparently

Ctrl+Button

Button	Action
2 (right)	Displays entity snap menu

Shift+Ctrl+Buttons

Button	Action
2 (right)	Displays entity snap menu
3 (middle)	Orbits drawing transparently

Double-click

Button	Action
1 (left)	*(See the Double-click Section)*
2 (right)	Zooms drawing to fit viewport

Roller Wheel

Wheel Movement	Action
Roll Up	Zooms in
Roll Down	Zooms out
Hold down	Pans the drawing

Double Click Actions

Most double-clicked entities display the Properties palette. Those that do not are shown in the list below.

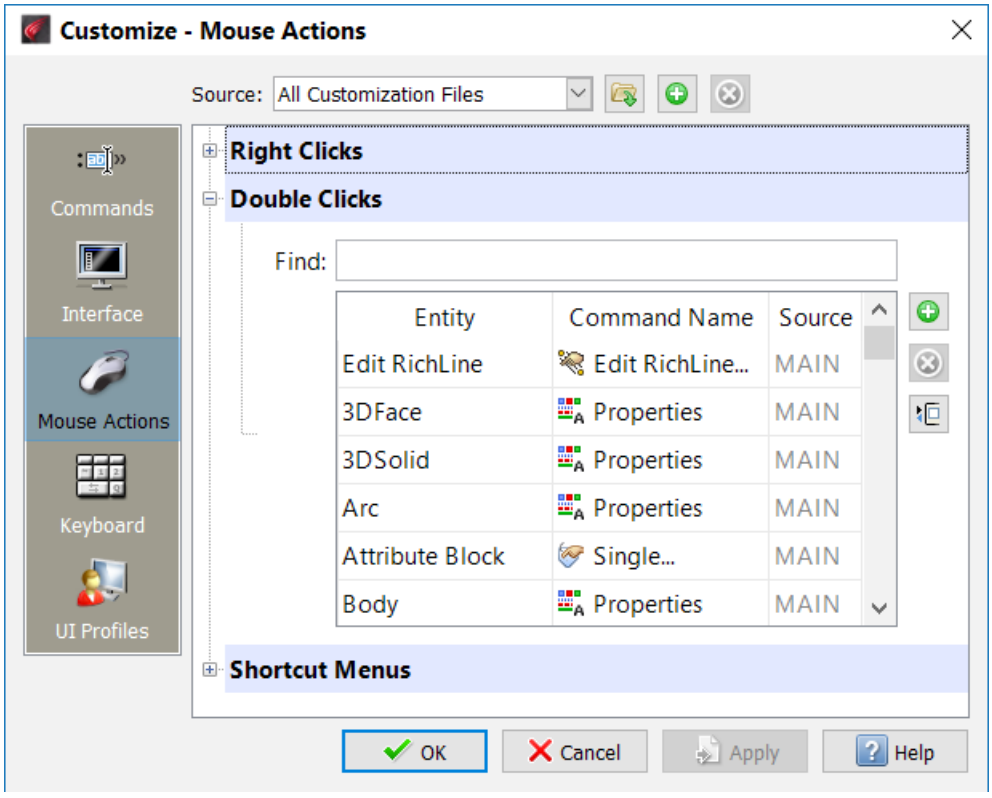
Entity Double-clicked	Command Executed
Attribute definition (AttDef)	EditAnnotation
Attribute instance (AttBlockRef)	EditBlockAttribute
Block instance	EditComponent
Block reference	EditCustomBlock
Constraint	EditAnnotation
Curved Text	CurvedText
External Reference	EditComponent
Hatch	EditHatch
LwPolyline	EditPolyline
Note (Mtext)	EditAnnotation
Ole2Frame	OpenOle
Polyline	EditPolyline
RichLine	EditRichLine
Text	EditAnnotation
Tolerance	EditTolerance
SimpleNote (Text)	EditAnnotation
Vnote (Voice note)	VoiceNote
Arc	Properties
Body	Properties
Centerlines	Properties
Circle	Properties
Cloud	Properties
DgnReference	Properties
Dimension	Properties
Ellipse	Properties
ExtrudedSurface	Properties
Helix	Properties
ILine	Properties
Leader	Properties
Light	Properties
Line	Properties
LoftedSurface	Properties
Mask	Properties
Point	Properties

Entity Double-clicked	Command Executed
Ray	Properties
Region	Properties
Shape	Properties
Solid	Properties
Table	Properties
Trace	Properties
3dFace	Properties
3dSolid	Properties

Middle button

Entity Double-clicked	Command Executed
...	Zooms drawing to fit viewport.

To customize mouse buttons and double-click actions in the program, enter the **Customize** command, and then choose **Mouse Actions**. Button #1 (right mouse button) cannot be customized.

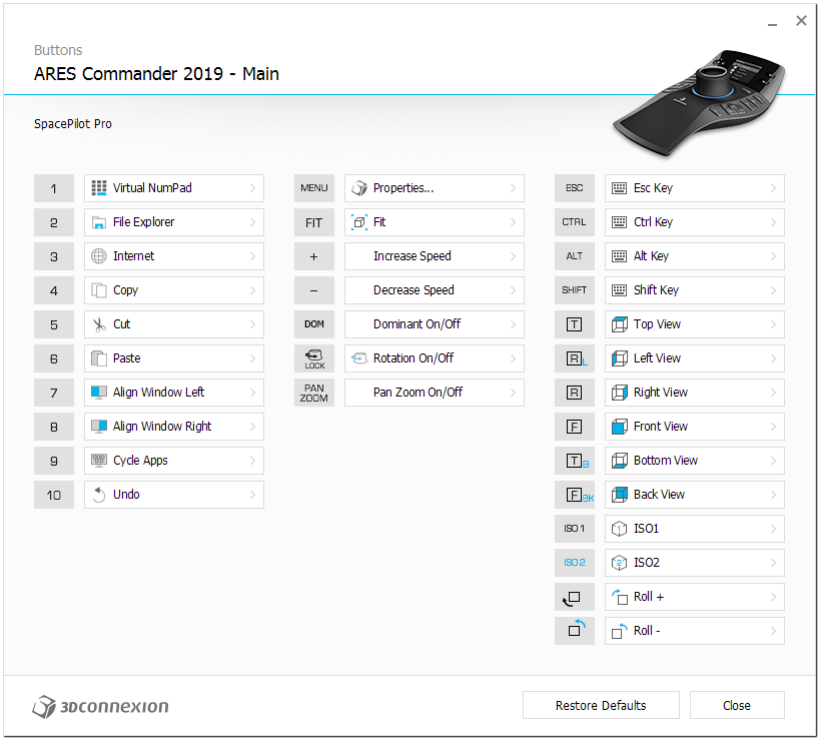


Customizing mouse button actions with the Customize dialog box

3D Mouse Actions

To use a 3Dconnexion-brand mouse with the program, follow these steps:

1. If necessary, exit the program.
2. Attach the 3D mouse to the computer.
3. Restart the program.
4. Use the 3DxWare utility program to customize the functions of the buttons and of the puck.



Buttons customized through 3Dconnexion's utility program

(NEW IN 2019) As of 2019, you no longer customize the actions of the 3D mouse with the 3dMouseOptions command, nor change the functions of it buttons through the 3dMouse-Buttons command. As well, the 3D Mouse menu and tab no longer appear.

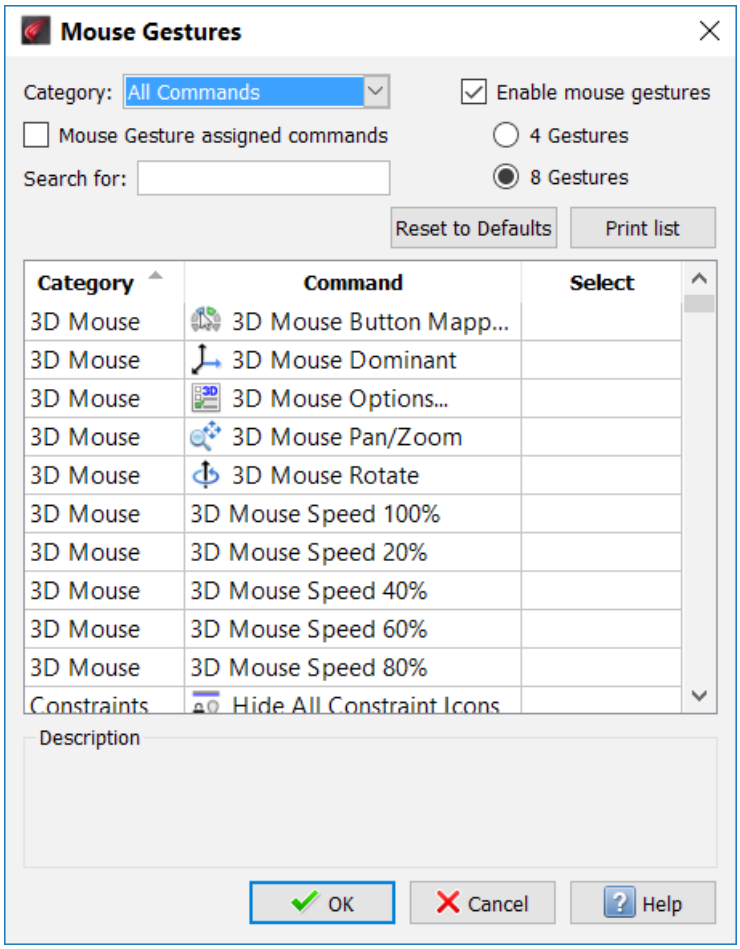
To customize the functions of a 3D mouse, use the 3DxWare software provided by the manufacturer. You download the 200MB utility program from the vendor's Web site at https://www.3dconnexion.com/service/drivers.html.

Mouse Gestures

When mouse gestures are turned on, you can start commands by dragging the mouse in one of eight directions. To *drag*, make sure no command is active, hold down the right mouse button, and then drag the mouse in one of directions listed below.

Mouse Direction	Command Executed
→	Line
↘	SimpleNote
↓	Save
↙	Copy
←	Open
↖	Offset
↗	Trim
↑	Zoom;Bounds

Mouse gestures are customized through the **Gesture** command.



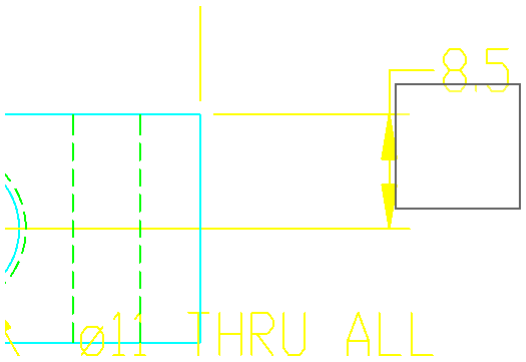
Using the Gestures dialog box to customize the commands executed by mouse gestures

Touchscreen Gestures

The program does not specifically support touch screens, but the following touch motions work with it:

Touch Action	Reaction	Similar Mouse Action
Tap	Selects UI elements Picks entities	Click left mouse button
Hold	Displays context menu	Click right mouse button
Drag and tap	Windowed selection of entities	Click left button and drag the mouse
Double-tap	Selects files Opens folder	Double-clicking left mouse button

TIP Holding the finger down on the screen acts like pressing the right mouse button. In this case, Windows displays a square to show that it has registered the action. (See figure below.) When you left your finger, the context menu appears.



Windows displaying a square to indicate the context menu will be displayed when the finger is lifted from the touch screen

Other touch motions do not work, such as pinching two fingers to zoom in. Touches cannot be customized.

Notes

APPENDIX F

ARES Commander for MacOS and Linux

In This Appendix

- Installing ARES Commander for MacOS
- Installing ARES Commander for Linux
- How the MacOS and Linux versions differ from the Windows version

The Linux, MacOS, and Windows versions of ARES Commander look and operate very similarly to each other. Fortunately, Graebert had the foresight to create a uniform user interface for all three, so that when you use one of them, you know all of them.

Here you also learn about the small differences in capabilities between the Windows version on the one hand, and of the Linux and MacOS versions on the other hand. In general, the latter two are missing a few features specific to the Windows operating system, such as OLE functions (object linking and embedding). Once you are in the program, however, there is almost nothing different in the user interface between the three, except for the ribbon (available only on Windows) and printing (different interface on Mac).

This appendix is particularly useful for those not fully familiar with Linux, for it describes step by step how to download and install the Linux version on your PC.

The MacOS Version of ARES

The MacOS version of the program operates very similarly to the Windows version.

If you are familiar with the program on Windows, there are, however, a few differences between the two of which you should be aware, such as how the software is installed and which functions are missing.

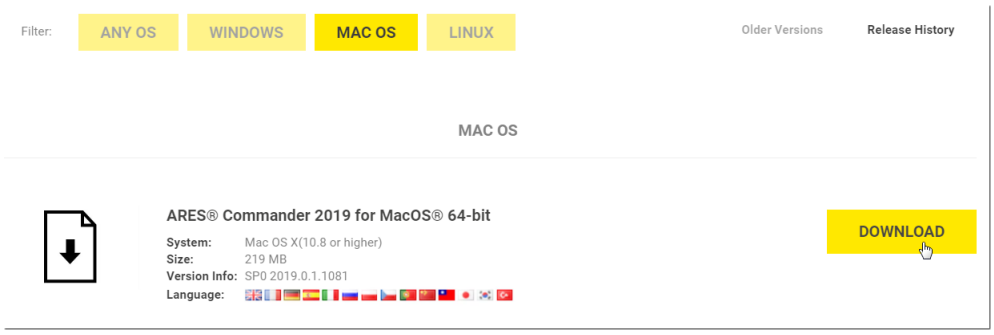
INSTALLING THE PROGRAM ON MacOS

Before installing the program, ensure that your Mac desktop or laptop is running a recent release of MacOS. These are the MacOS versions supported officially by Graebert:

» MacOS v10.8 (Mountain Lion) or higher

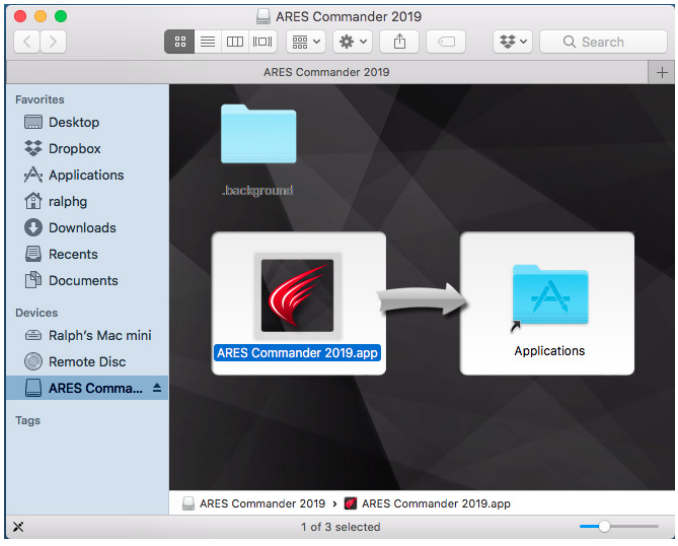
Installing the program on a Mac works like this:

1. Go to https://www.graebert.com/cad-software/download/ares-commander/ with your Web browser, and then click **Mac OS**.



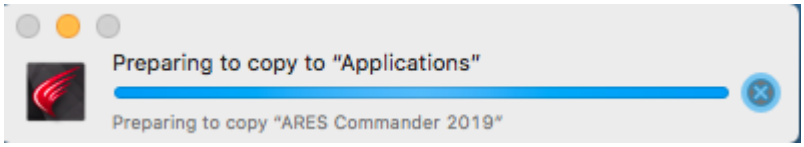
Locating MacOS version of ARES to download

2. Click **Download** and wait for the Mac *.dmg* (Disk iMaGe) file to arrive on the Mac.
3. When the Web browser finishes downloading the DMG file, locate it in the **Downloads** folder of your Mac computer.
4. Open it by double-clicking the DMG file, or else open the file from inside the Web browser.
5. Notice the dialog box. In the dialog box, drag the icon onto the Application folder.



Installing the program by dragging its DMG file into the Applications folder

5. After a minute or so, the program is "installed."



MacOS copying the program to the Applications folder

Starting the Program (MacOS)

With the program installed on your Mac computer, you can now start it. Follow these steps:

1. On the Mac desktop, open the **Application** folder with a single click — not a double-click!
2. Find the ARES icon, and then click on it.

Locating the icon in the Application folder

2. Notice that the icon appears on the dock, and that the program starts up.

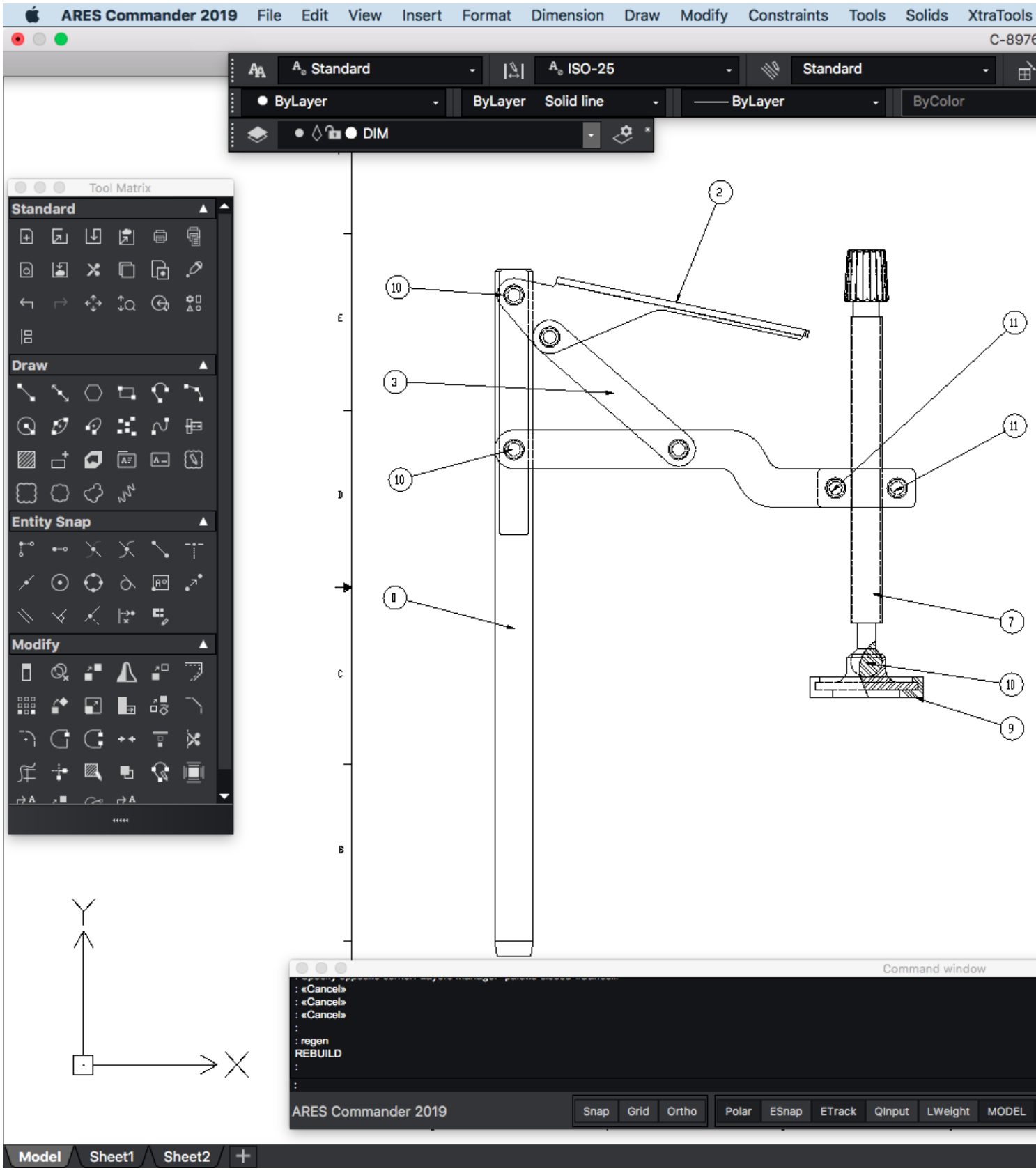
Program icon appearing on the dock

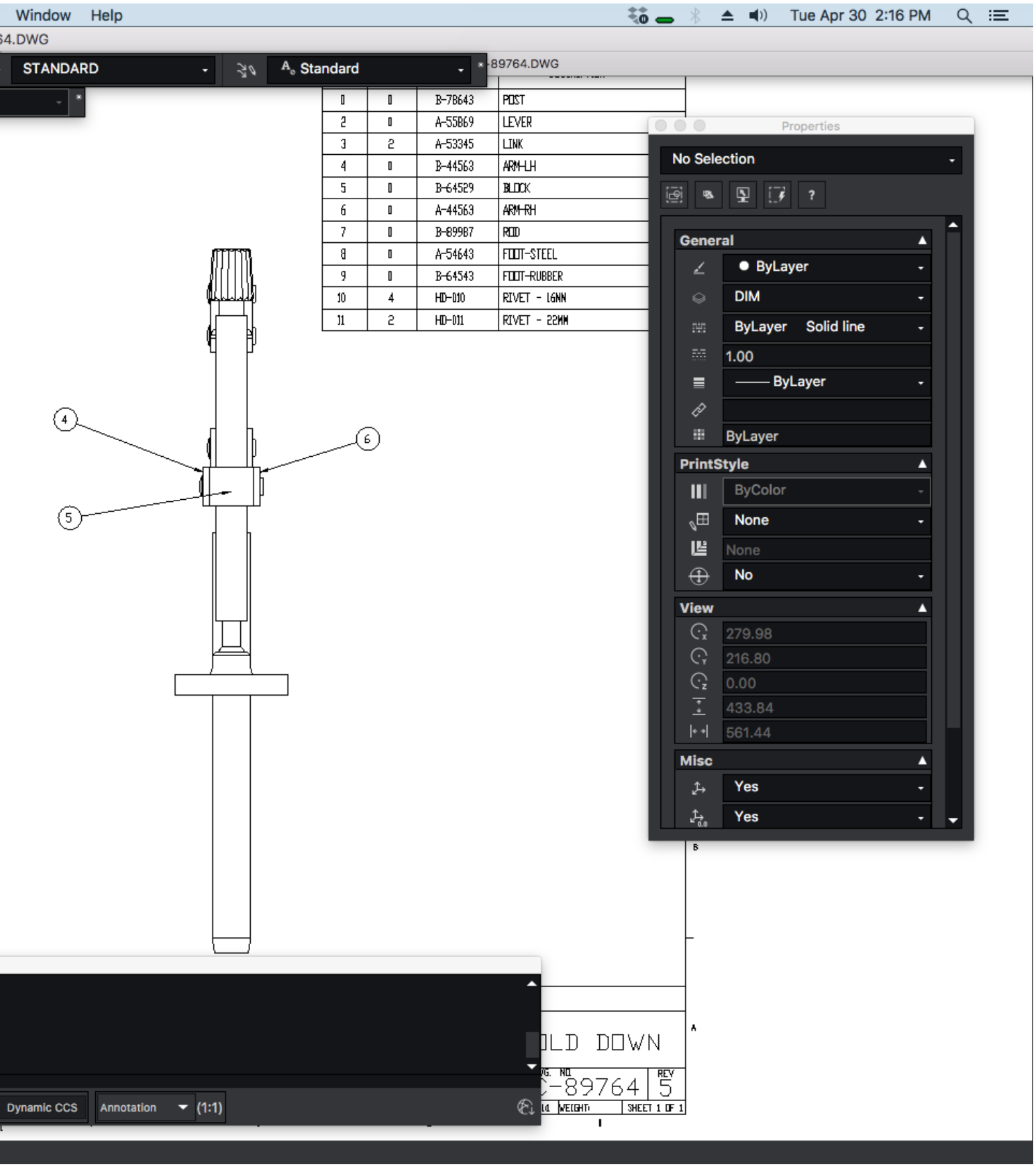
TIP: You can keep the icon in the dock to start the program easily. To do so, right-click the icon, choose **Options**, and then choose **Keep in Dock**.

---

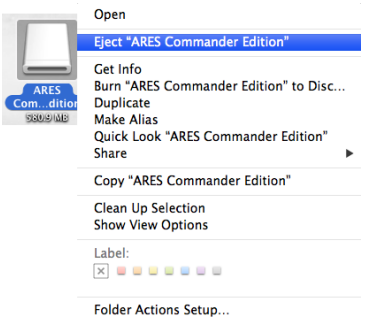
3. The first time the program runs, it asks you to register the software with Graebert:
   - » If you purchased the software, then you must register it.
   - » If you installed an evaluation version, then you don't have to register it, but then the program looses the ability to print and save drawings. When you register the evaluation version, the program is unlocked and performs these useful functions.

   To take the steps required for registration, see "Starting and Registering the Program" in the Linux section later in this chapter.


ARES Commander 2019
File
Edit
View
Insert
Format
Dimension
Draw
Modify
Constraints
Tools
Solids
XtraTools
Standard
ISO-25
Standard
ByLayer
ByLayer
Solid line
ByLayer
ByColor
DIM
Tool Matrix
Standard
Draw
Entity Snap
Modify
Command window
: «Cancel»
: «Cancel»
: «Cancel»
:
: regen
REBUILD
:
:
ARES Commander 2019
Snap
Grid
Ortho
Polar
ESnap
ETrack
QInput
LWeight
MODEL
Model
Sheet1
Sheet2


Window Help
Tue Apr 30 2:16 PM
STANDARD
Standard
89764.DWG
Properties
No Selection
General
ByLayer
DIM
ByLayer Solid line
1.00
ByLayer
ByLayer
PrintStyle
ByColor
None
None
No
View
279.98
216.80
0.00
433.84
561.44
Misc
Yes
Yes
4
5
6
Dynamic CCS
Annotation
(1:1)
SHEET 1 OF 1

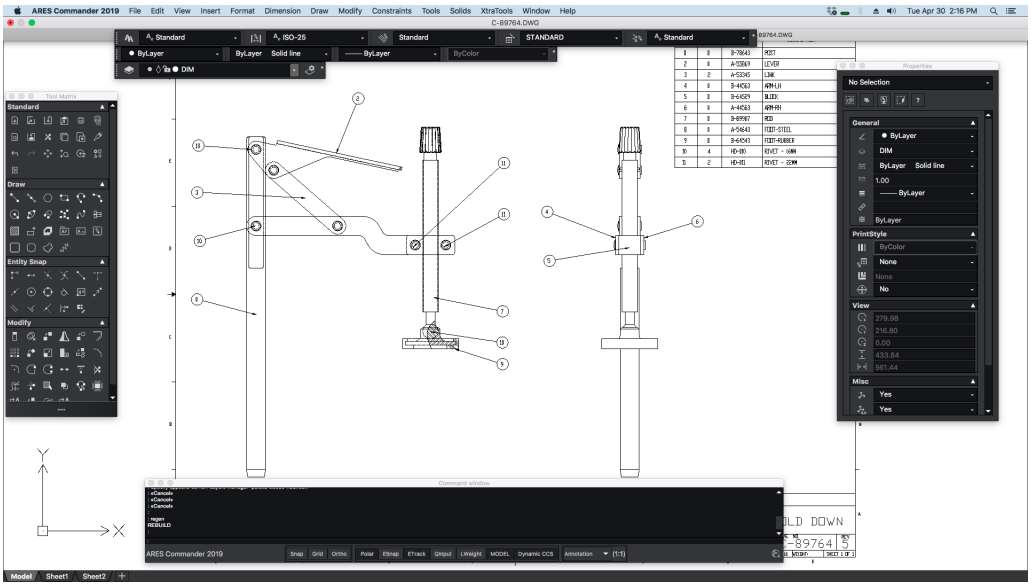
4. As the final step, clean up the desktop by ejecting the installed program's "drive," like this:
   a. Right-click the the program's disk icon.
   b. Choose **Eject** from the shortcut menu.



Ejecting the installation "disk"

CHANGES IN THE USER INTERFACE (MacOS)

The user interface of the MacOS version looks somewhat different from Windows, because it mimics the separated interface of professional Mac application (see figure below). Whereas in Windows most UI (user interface) elements reside inside a master MDI (short for "multiple document interface") window, Mac applications tend to float UI elements independently of others.



If you don't like lots of windows, then the **UiMode** system variable switches the program's user interface. In the default no mode, it is Mac-like; in the yes mode, like Windows, which the program calls "fixed-style toolbar mode":

```
: uimode
Confirm: Do you want to change to fixed-style toolbar mode?
Specify Yes or No» (Enter Y or N)
```

» **Yes** — MacOS interface is more like that of Windows
» **No** — MacOS interface follows the Apple UI guidelines

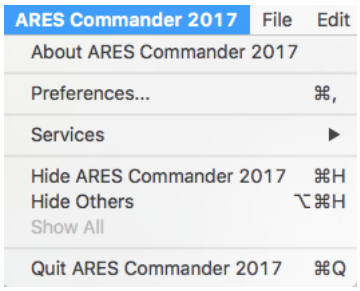
To effect the change in user interface, however, you have to exit the program, and then restart it.

About the Mac Application Menu (MacOS Only)

The ribbon user interface is not available on the Mac version of the program, because the Mac version follows Apple's rules for user interface design. This means that the Application button is replaced by the "ARES Commander" menu (near the far left end of the menu bar). The Quick Access toolbar and ribbon are unavailable.

"ARES Commander" replacing the Application button of Windows

For Windows users who are new to Mac computers, it is useful to know that the **ARES Commander** menu houses the "options" command under the name of **Preferences**.



Preferences menu item on Mac replacing the Options one from Windows

What's Missing from The Mac Version

All commands related to OLE (object linking and embedding) are missing from the Mac version, because MacOS does not support them. The missing commands are as follows:

- ConvertOLE
- InsertObject
- OpenOLE
- ResetOLE

Workspaces are not supported, and so the Workspace droplist does not appear in the MacOS version:

The following commands are not yet supported, but may appear in a future release of the software:

- ExportEMF
- ExportWMF

Application Programming Interfaces

The following APIs are found in the Windows version but not yet supported on Mac, and in some cases never will be in MacOS:

- LISP encryption is not supported
- DRX or ARX-like runtime extension is not supported
- ActiveX, COM, and VSTA never will be supported

Like OLE, ActiveX, COM (common object model), VSTA, and .Net are unique to Windows, and so do not operate on MacOS.

The Linux Version of ARES

The Linux version of the program operates very similarly to the Windows version.

If you are familiar with the Windows version, then you should be aware that there are a few differences between the two, including differences in how the software is installed and which functions are not available in the Linux version.

INSTALLING THE PROGRAM ON LINUX

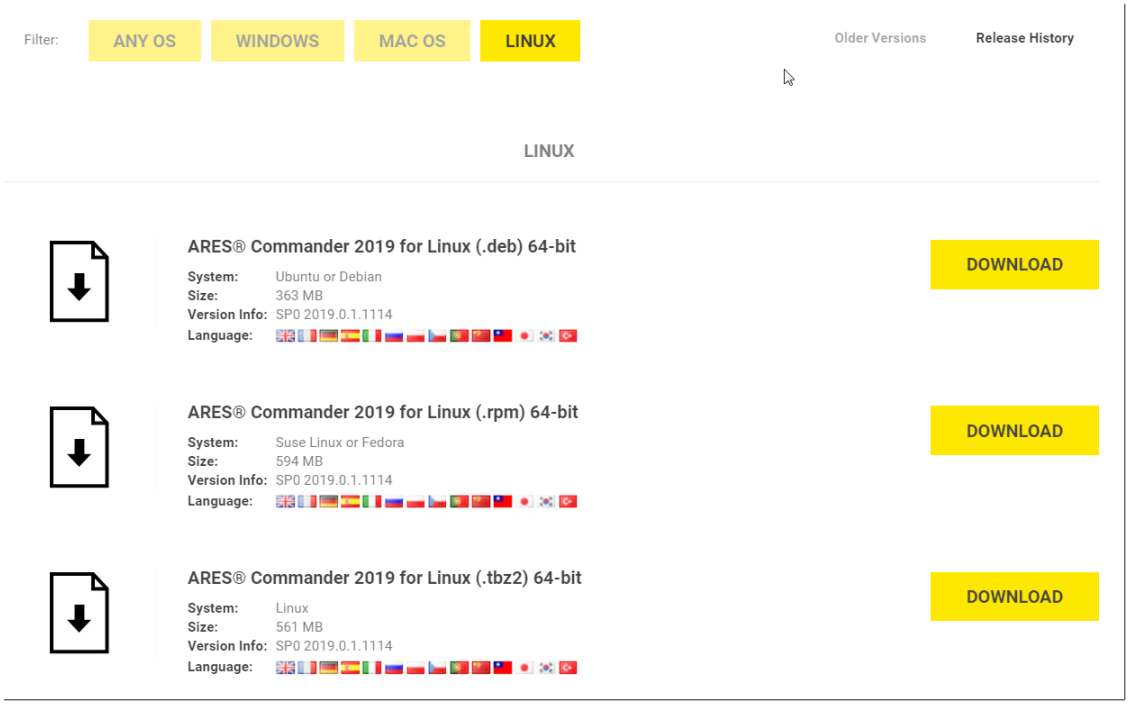
You can install the Linux version of the program on your 64-bit PC running recent versions of Linux:

- Ubuntu (*.deb* installation files)
- Debian (*.deb* installation files)
- Suse (*.rpm* installation files)
- Fedora (*.rpm* installation files)
- Generic Linux (*.tbz2* installation files)

ARES for Linux rund on other dialects as well. For instance, I use Mint Linux, which is based on Ubuntu Linux. Ubuntu had been the most popular dialect of Linux for desktop computers; it is based on Debian Linux. Mint keeps the traditional desktop look and includes proprietary software, such as Adobe Flash and MP3 support, which Ubuntu leaves out. You can download Mint free from https://www.linuxmint.com/download.php.

To install the program on Linux, follow these steps:

1. Go to the https://www.graebert.com/cad-software/download/ares-commander/ Web page, and then click on **Linux**.



Choosing the installer for Linux

2. Choose the download file that most closely matches the distribution of Linux running on your computer:

   RPM — Suse, Mandiva, and Fedora distributions use RPM files for installing software. which are installed with the RPM package installer.

   RPM is short for "RPM package manager."

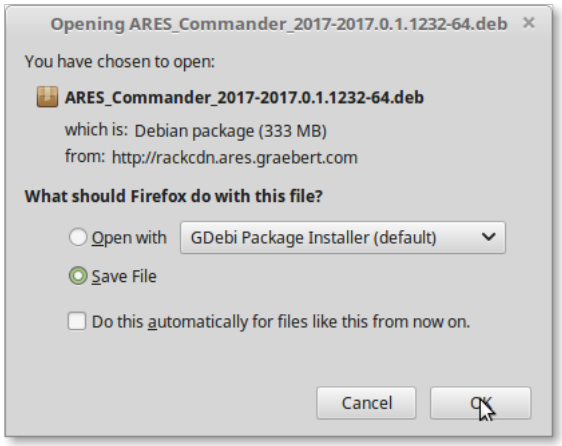
   DEB — Ubuntu and Debian distributions use DEB package files for installing software; this is the download needed for Mint Linux. DEB package files are installed using the GDebi package installer.

   Deb is short for "Debian."

   tbz2 — Many distributions of Linux can use Tbz2 package files, which are installed with the BZip2 installer.

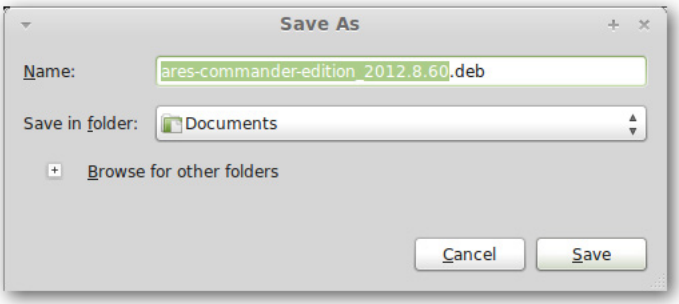
   Tbz is short for "tape archive," and bundles together all needed files with Tar (like DMG on Macs), and then compresses them with BZip2.

3. When the Web browser asks to open or save the file, I suggest saving the file. Choose **Save**. (The steps are illustrated for the Firefox Web browser running in Linux Mint . The steps taken and the dialog boxes shown may be different with other Web browsers and distributions of Linux.)



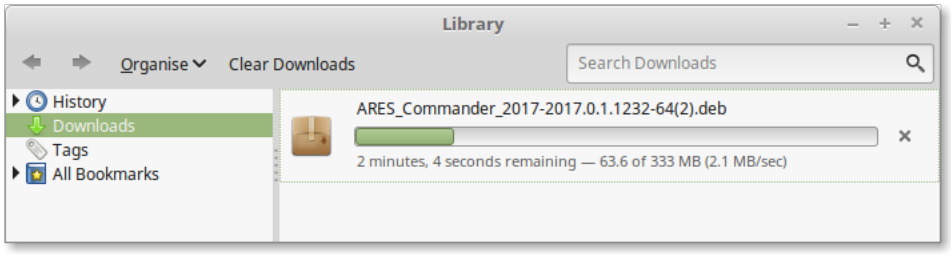
Saving the download file

4. If necessary, choose the folder in which to save the file.



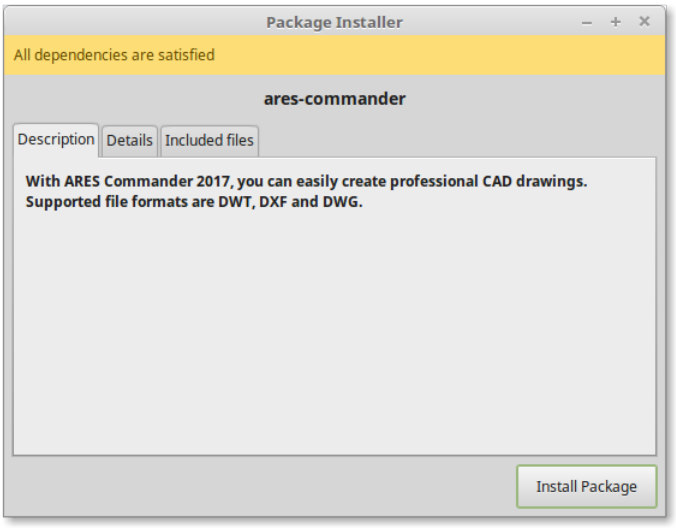
Choosing the folder in which to save the downloaded file

Wait a few minutes for the file to download.



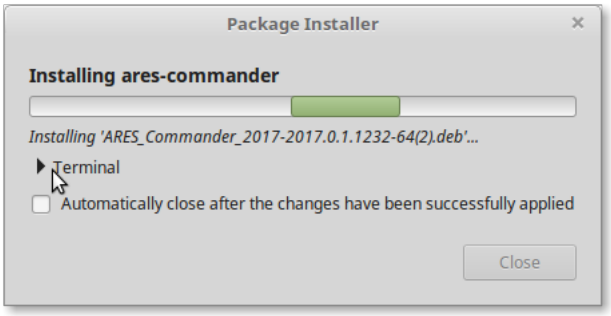
File downloading

5. After the file is fully downloaded, double-click its name. (Alternatively, right-click the name of the file in the Web browser's Library | Downloads window, and then chose **Open**.)
6. When the Package Installer program appears, wait for it to examine the contents of the package, during which the Install Package button looks gray. When the **Install Package** button becomes available, click it to install the program.



Installing the program with the Package Installer program

7. Notice that a dialog box appears asks for your Linux password to protect your computer against malicious software that could install itself automatically. Enter the same password with which you use to start Linux, and then click **OK**.
8. Wait while the installer installs the program and its support files. When it is done,click **Close**

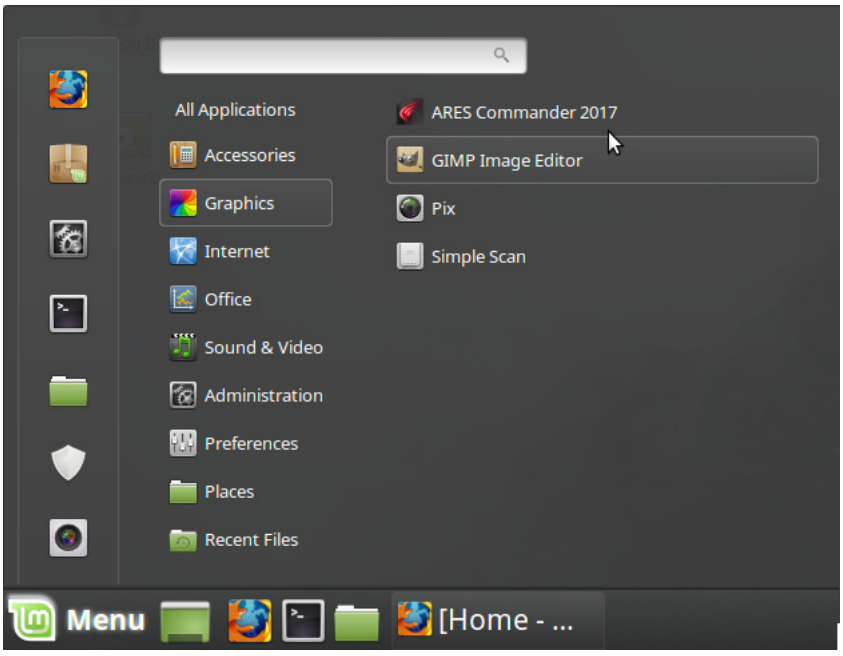


Waiting for installation to finish

Starting the Program (Linux)

The Linux version of the program is now installed on your computer. To start the program on Linux, follow these steps:

1. On the Linux panel (a.k.a. taskbar), click the Menu **Main Menu** icon. This icon looks different in every variant of Linux, but is often found at the left end of the task bar. (It is just like the Start button in Windows.)
2. From the popup menu, choose the **Graphics** section.

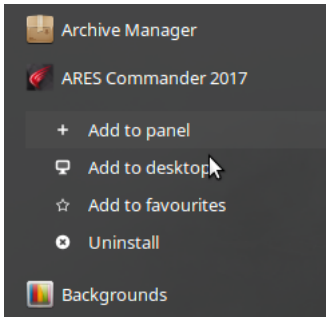


Starting the program from the Linux main menu

3. Click **ARES Commander**. Notice that the program opens.

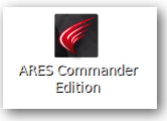
TIP If you wish to place the icon on the Linux desktop, follow these steps:

1. On the Linux taskbar, click the **Main Menu** icon.
2. Right-click **ARES Commander**.



3. Choose **Add to Desktop**.

Notice that the icon appears on the desktop. You can now click this icon to start the software.



Registering the Program (Linux and MacOS)

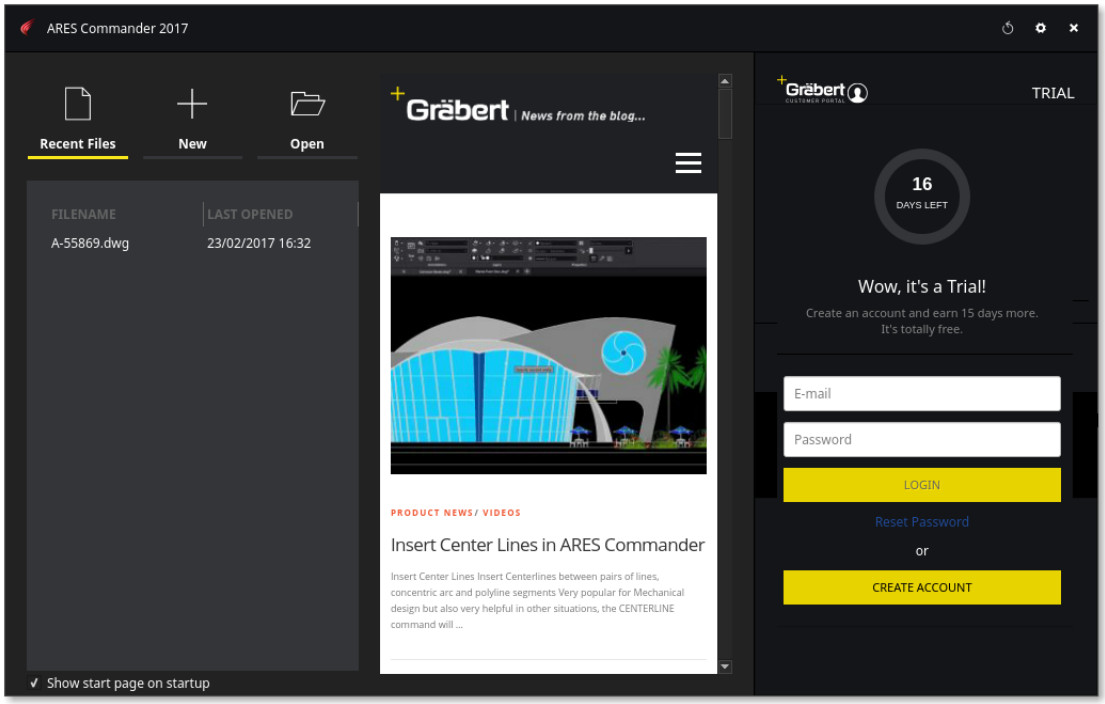
The first time you start the program, you need to enter some registration information. You don't have to do this with the evaluation version, but then you won't be able to save or print drawings; so, registering the 16-day version lets the program perform these additional functions. The steps are the same for all versions of the program, whether running Linux, MacOS, or Windows.

1. When the program opens, it presents a weclome screen. It is divided into these sections:

   Left — accesses recent, new, and existing (Open) files

   Center — displays news from the Graebert corporate blog

   Right — activates the licence

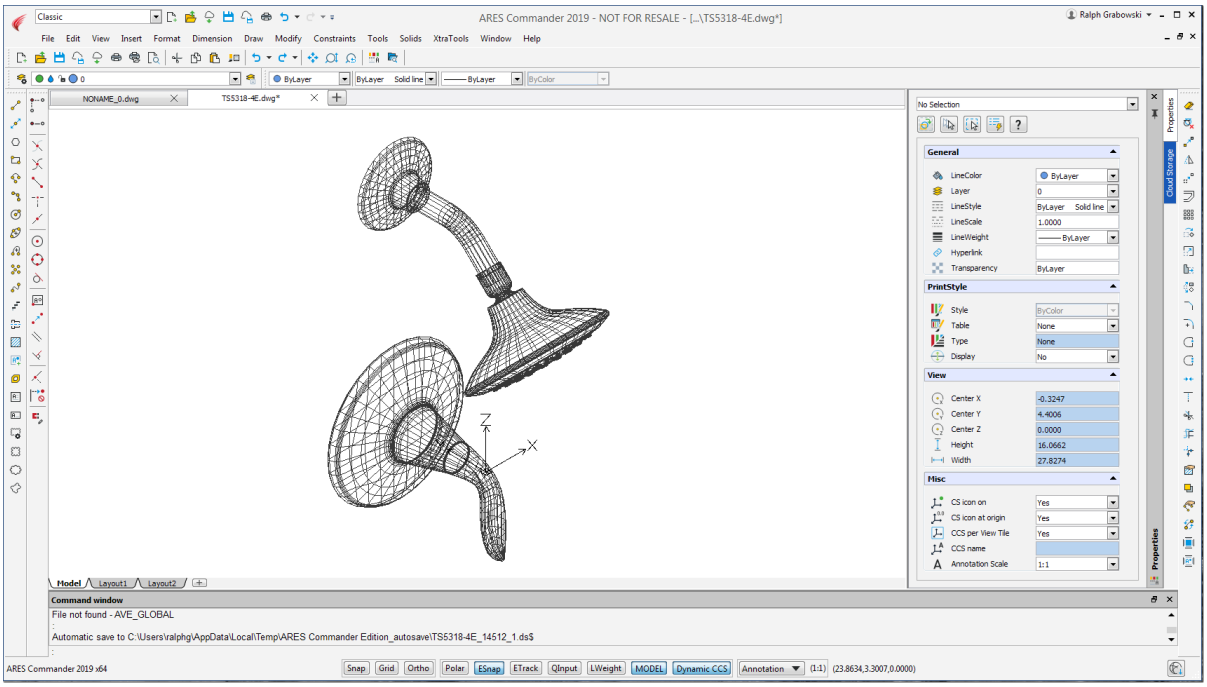


Options to consider when starting the program

2. To active the software, take either of these steps:
   - » **If you have a Graebert account** — enter your email address and password, then click **Login**
   - » **If you don't have the account** — click **Create Account**
3. Follow the instructions you see. When done, you can starting using ARES Commander.

CHANGES IN THE USER INTERFACE (LINUX)

The user interface of the Linux version looks very similar to the Windows version running the Classic workspace (see figure below) with toolbars and menus instead of the ribbon. You may notice minor differences due to the way each operating system's UI engine displays buttons and so on.



Windows version of the program looking like the Linux version

Functions Not Available in The Linux Version

All commands related to OLE are missing, because Linux does not support object linking and embedding. Specifically, the missing commands are these:

- ConvertOLE
- InsertObject
- OpenOLE
- ResetOLE

Workspaces are not supported, and so the Workspace droplist does not appear in the Linux version:

The following commands are not yet supported, but may appear in a future release of the software:

- AttachPDF, ClipPDF, LayersPDF, and related system variables
- ExportEMF and ExportWMF
- SpellCheck

Notes

APPENDIX G

Registering ARES

In This Appendix

- Understanding how registration works
- Registering ARES Commander, ARES Kudo, and ARES Touch
- Unbinding and binding licenses to hardware
- Controlling access in organizations

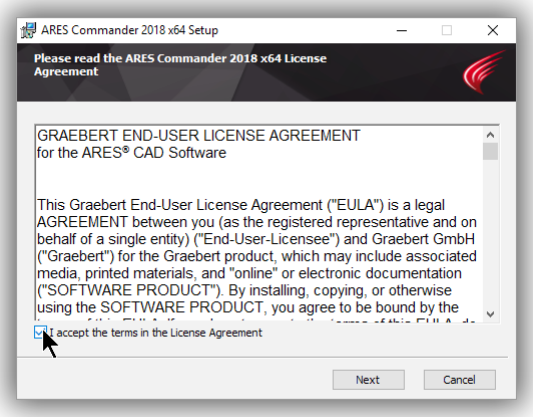
The first time you ever start an ARES product, you need to create an account with Graebert Gmbh. You register by specifying a login name (which is your email address) and a password. The account keeps track of the licenses you have paid for, and the computers on which ARES is running.

In this appendix, you learn about registering your newly-installed software.

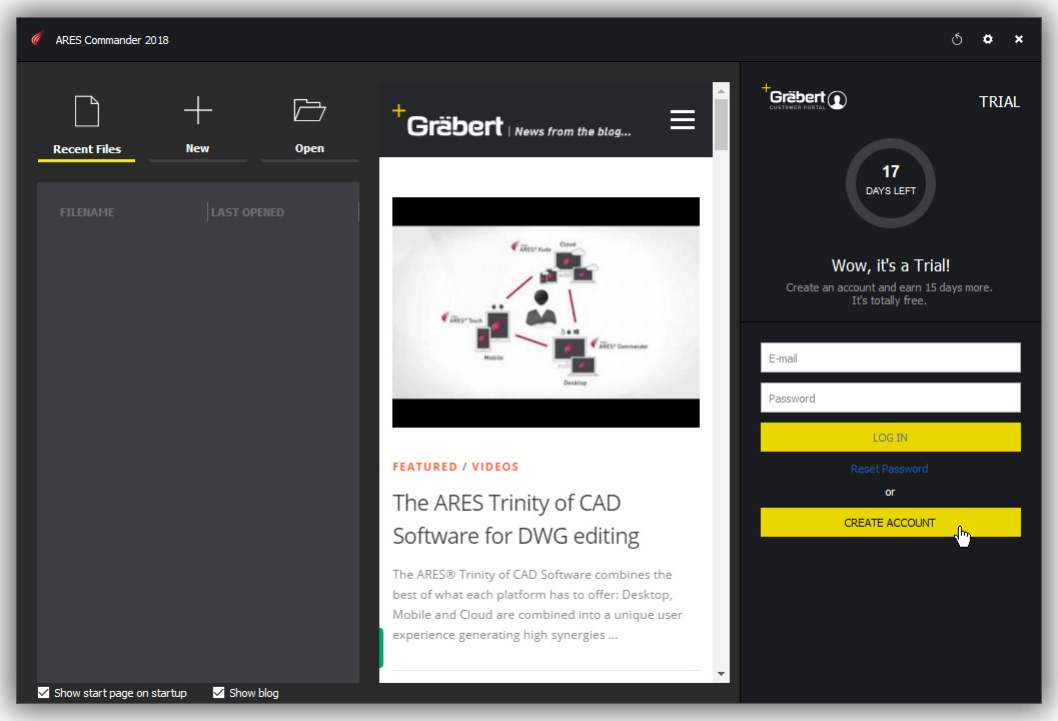
Registering an Account

You create your account with Graebert Gmbh the first time you use one of the their software programs. While you do not need to create an account for the trial version, doing so extends the free time from 15 days to 30. After downloading the software and installing it, follow these steps:

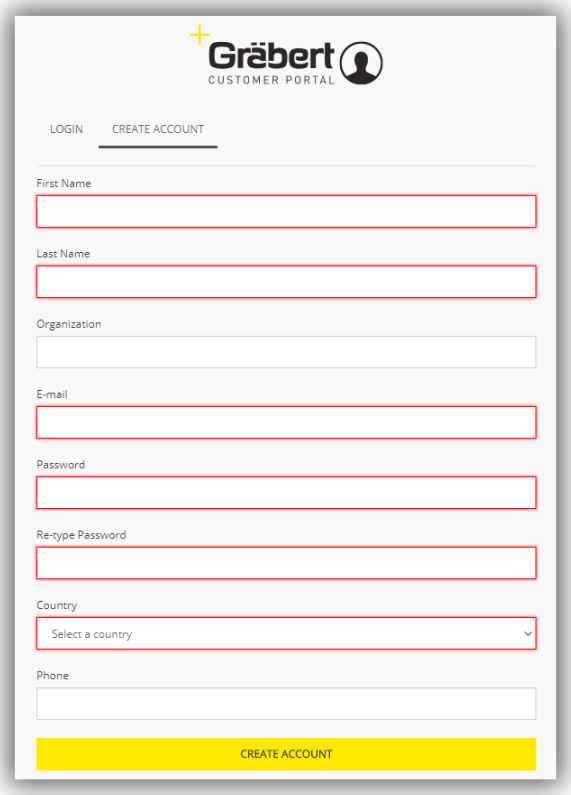
1. Start ARES Commander, ARES Touch, or ARES Kudo.
2. If you are asked to agree with licensing terms, do so, and then click **Next**.



3. Notice that you are prompted to enter a user name and password. (The dialog box for entering these looks different, depending on the type of ARES; the one below is for ARES Commander.) If you have not specified a user name and password yet, click **Create Account**.

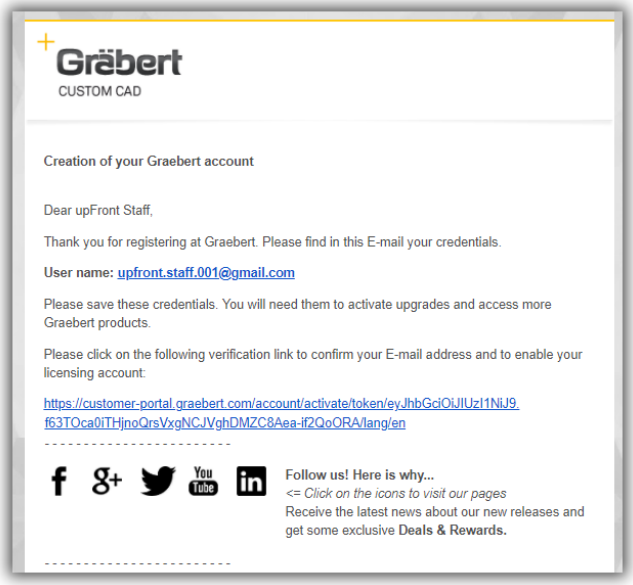


4. Notice that your computer's Web browser opens and that you are prompted to enter information about yourself. Fill out all the fields that are outlined in red in the figure below.



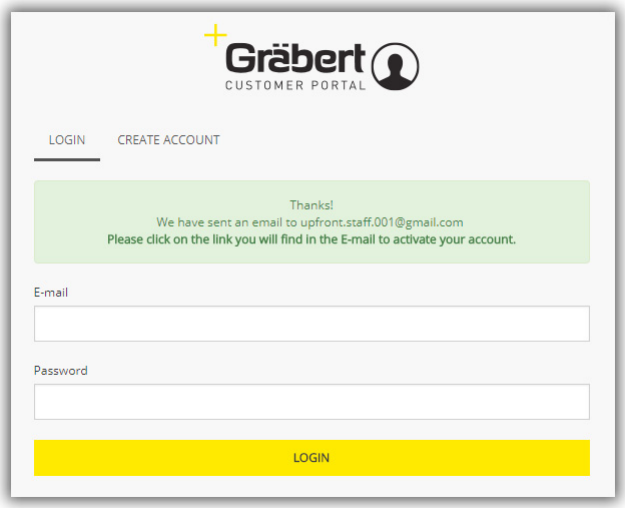
5. Click **Create Account**.

6. Now you need to wait for Graebert to send you an email message that confirms the account creation. This usually happens pretty quickly.



When the email arrives, click the blue link you find in the message. This tells Graebert that the account you created is legitimate.

7. The link opens the Graebert Customer Portal page in your Web browser. Enter your email address and password, and you're in!

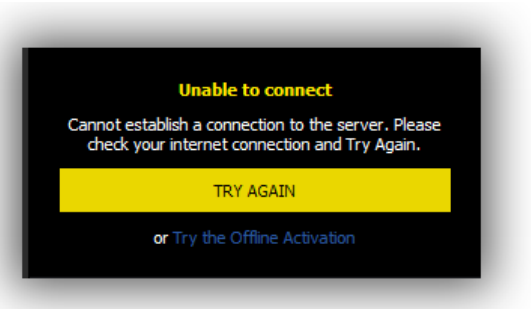


OFFLINE REGISTRATION

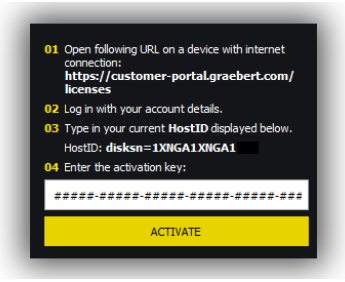
Some design firms have a policy that their computers cannot be connected to the Internet. This is to protect them from attacks from outside, as well as prevent their intellectual property from being stolen. In other cases, laptops might not have access to the Internet due to the location of the job site.

Graebert offers an offline registration method for any computer that is disconnected from the Internet for more than 30 days at a time. It does require another device to be connected to the Internet, but this is a one-time procedure.

1. Start ARES Commander.
2. When it notices no Internet connection, the Start Page asks you to "Try Again" to connect. Ignore this by clicking the blue **Try Offline Activation** link.

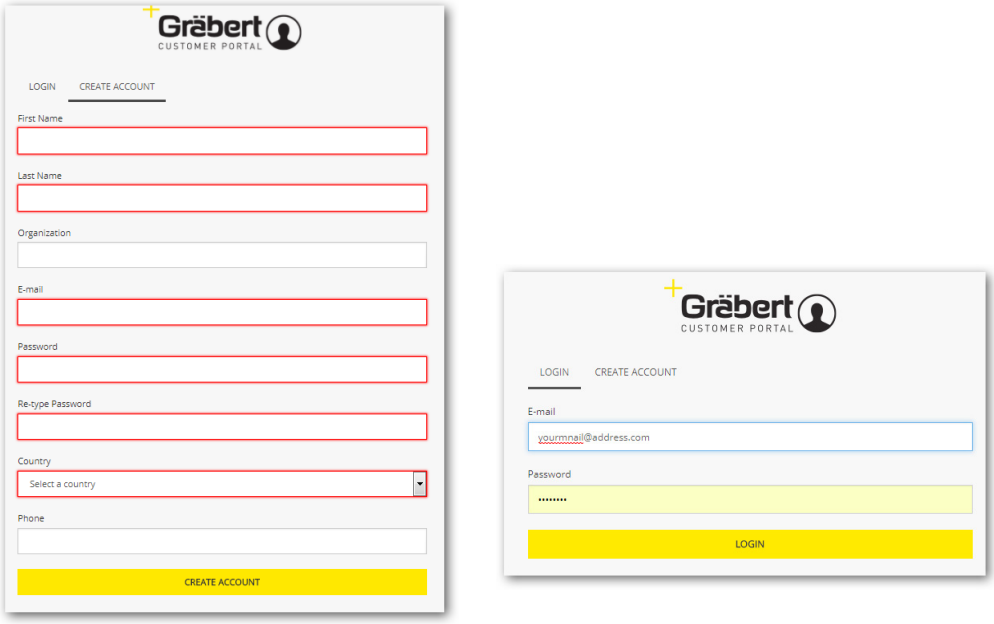


Notice that the Start Page displays the four steps needed to register offline.



3. With a Web browser on another computer or a mobile device connected to the Internet, go to the https://customer-portal.graebert.com site.

4. In the Graebert Customer Portal, either (a) create a new account by entering the information shown by the red-outlined boxes in the figure, or (b) login with your existing account.

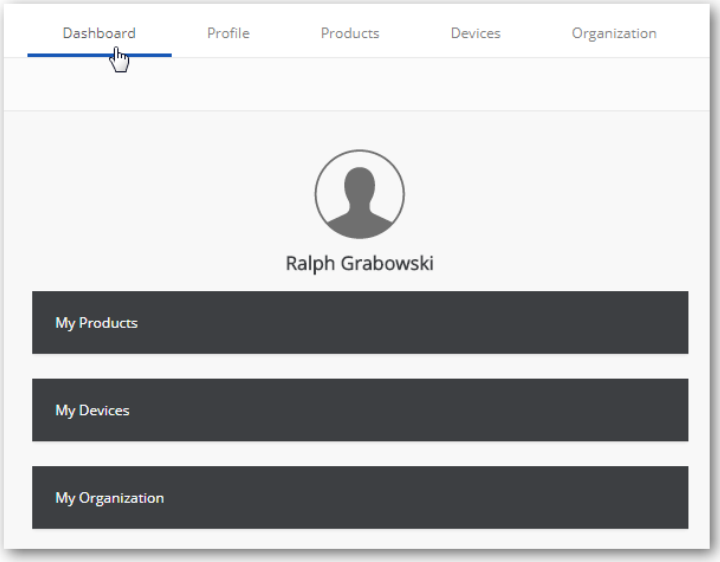


Left: *Creating a new account with Graebert;* ***right:*** *Logging in with an existing account*

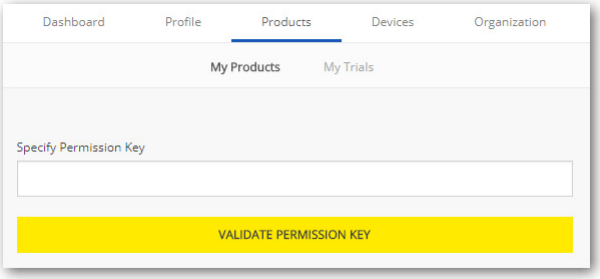
(If this is a new account, wait for the email confirmation, and then continue with the following steps.)

5. Once logged in, notice the Dashboard tab:

   a. Click **My Products**.

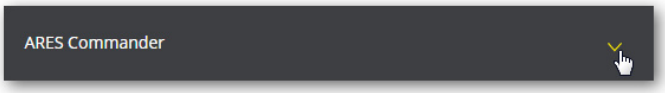


b. Copy with **Ctrl+C** the permission key you received with the purchase of the software, and then paste it with **Ctrl+V** into the **Specify Permission Key box**.



c. Click **Validate Permission Key**. Notice that the name of the software appears under My Products.

6. Click the arrow next to the product name to expand product information:



a. Click **Activate Offline**.

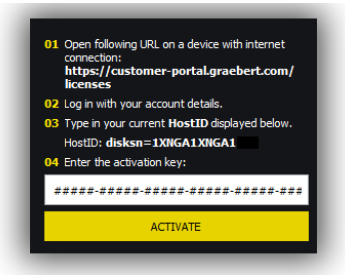
b. However, click **Cancel** if the following criteria do not apply to you:

» The offline option locks the license to the specified computer
» The offline license cannot be unlocked from the computer with this customer portal

c. Enter the HostID number displayed by the software (in Step 2, earlier).

d. Click **Activate**.

7. Wait for an email to arrive with the activation key for the Internet-less computer. Enter the number into the "####" field shown in step 04, and then click **Activate**:



TIP Offline activation locks the license for 30 days, during which you cannot unbind the computer in the Graebert Customer Portal.

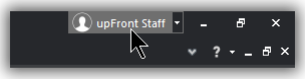
CHECKING YOUR LICENSES

You can control the access your computers have to the software. Graebert provides this function through its customer portal at https://customer-portal.graebert.com.

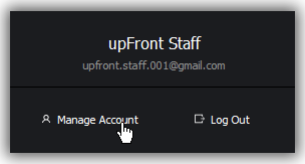
You can go to the customer-portal.graebert.com Web site directly, or else indirectly from the following places in each ARES program:

ARES Commander

a. In the upper right corner of the program, click your name.



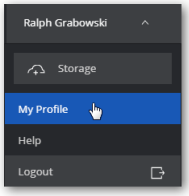
b. Notice that the Start Page dialog box appears. In the right-hand pane, choose **My Account**.



c. When the Web browser appears, log in.

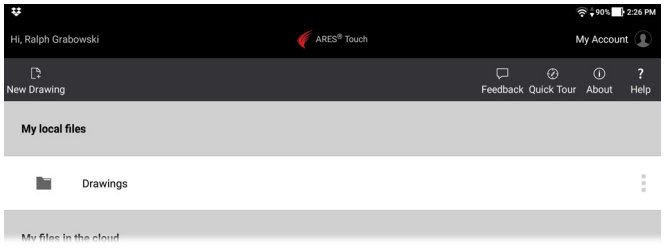
ARES Kudo

a. In the upper right corner of Kudo, click your name, and then choose **My Profile**. Notice that the Web browser appears; you do not need to log in to the portal.

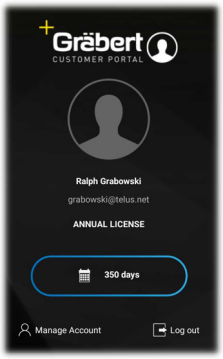


ARES Touch

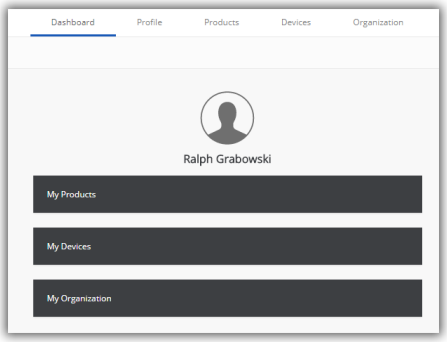
a. In the app, return to the file manager, and then in the upper right corner tap **My Account**.



b. In the dialog box that appears, tap **Manage Account**. The Web browser opens: log in.



Once in the Dashboard, you can check on the ARES software you are licensed to use and the computers on which they run.



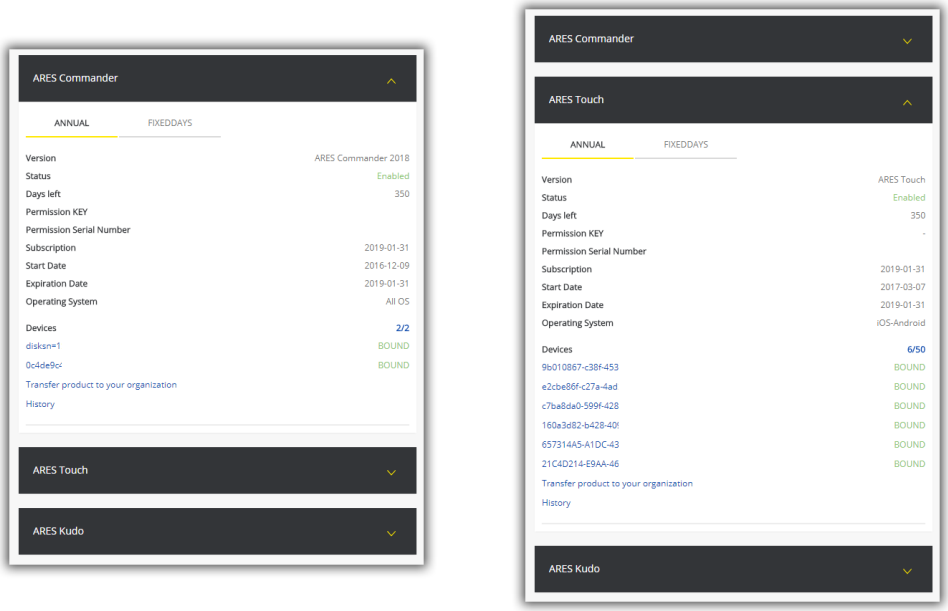
Notice that you can access the various categories in two places: along the top of the screen (Dashboard, Profile, Products, and so on) or else in the center of the screen (My Products, My Devices, and so on). They are the same.

Software: My Products

Click **Products** (or **My Products**) to see a list of the software that you are authorized to run.

Then click each of the software titles to see details on it. Below I show the info provided for ARES Commander and ARES Touch. Notice that Touch has a long list of devices on which it is authorized to run. This is because I can install it on any Android and/or iOS device with which I am registered with Google

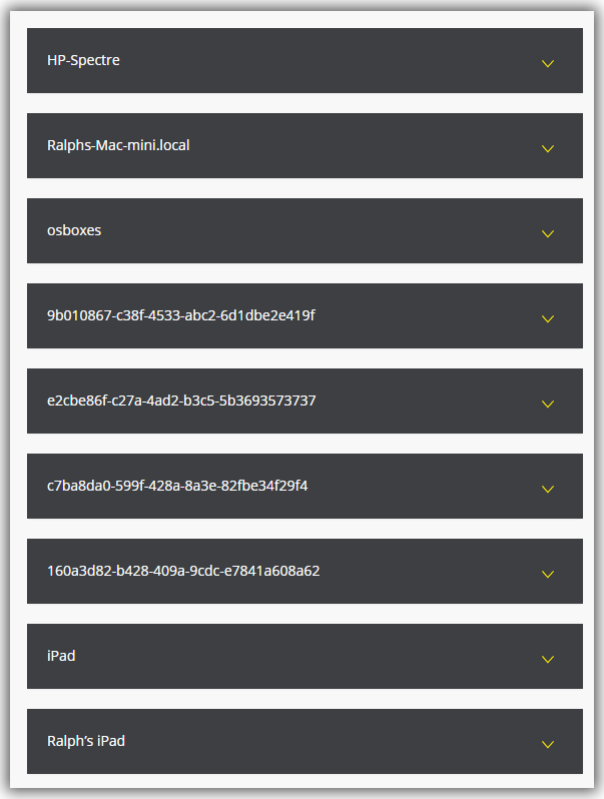
or Apple. I have a lot of tablets and smartphones, and so Touch runs on many of them. By contract, ARES Commander runs at most on two of my computers for a single license.



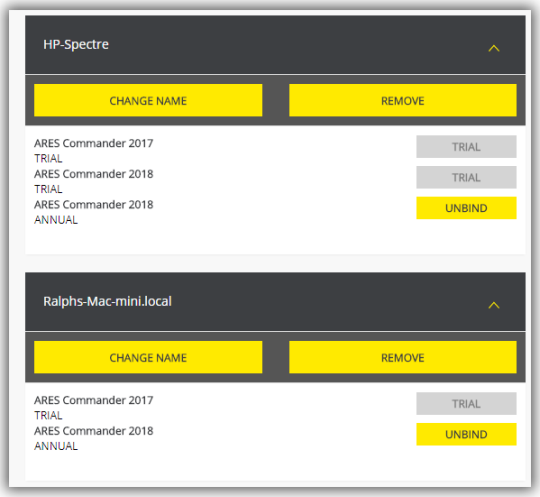
Left: *Status information for ARES Commander;* ***right:*** *...and for ARES Touch*

Devices

Click **Devices** (or **My Devices**) to see a list of the hardware on which you are authorized to run Graebert software. Your list might be longer or shorter than mine shown below.



Click the name of a device to see which software it is authorized to run. Below I show the result for my Windows laptop and my MacOs Mini computer.



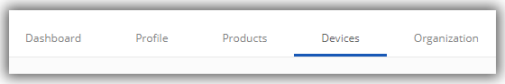
(The Organization tab is described later in this appendix.)

MOVING LICENSES BETWEEN COMPUTERS

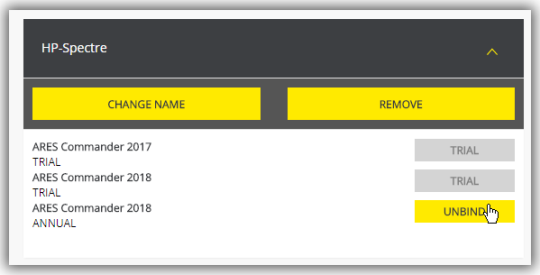
Graebert is generous in the number of devices on which you can run its software. So, for the ARES Commander, you can install and run a copy each on your desktop and laptop computer, for instance, or else on a Windows and a MacOS computer. For ARES Touch, you can install it on all your tablets and smartphones with your Google or Apple login. And for ARES Kudo, you can run it in any Web browser.

Sometimes, however, you may want to move a license from one desktop computer to another. Follow these steps:

1. In the customer portal, click **Devices**.



2. Choose the computer from which you want to remove the license authorization.
3. Click **Unbind**.



4. Now choose the computer that should get the license. Click **Bind**.

If a computer still runs ARES Commander in full mode (saves and prints), then it might still have days left over from a trial version, or was given a 30-day token for offline use. Moving licenses with Touch and Kudo is not necessary, as they can run on as many devices as necessary.

ARES for Organizations

When organizations deploy ARES among employees, an IT manager or CAD manager controls the access that employees have to the software. Graebert provides this function through its customer portal at https://customer-portal.graebert.com.

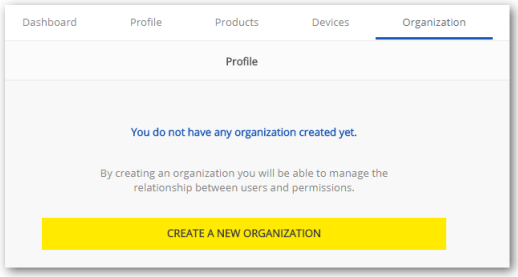
1. You can go to the Web site directly, or else indirectly from the following places in each ARES program:

   ARES Commander — in the upper right corner, click your name, and then choose **My Account**. When the Web browser appears, log in, and then choose **Organization**.

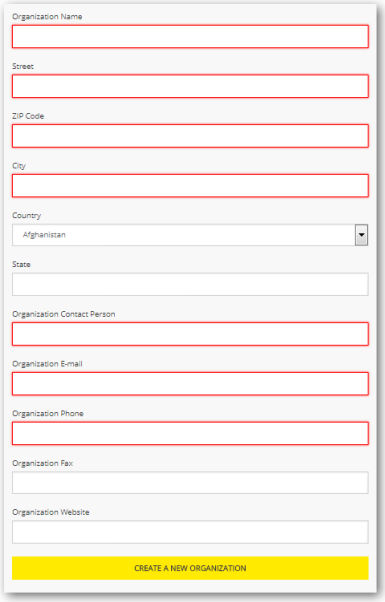
   ARES Kudo — in the upper right corner, click your name, and then choose **My Profile**. In the new browser tab, click **Edit My Profile**, and then choose **Organization**.

   ARES Touch — return to the file manager, and then tap **My Account**. When you tap **Manage Account**, the Web browser opens: log in and choose the **Organization** tab.

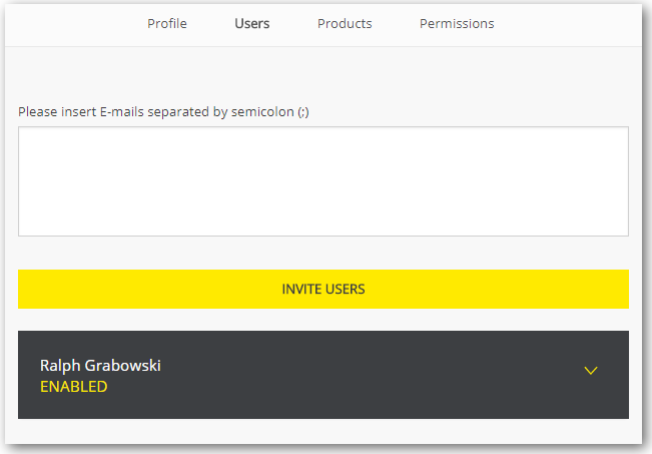
2. In the Organization tab, click **Create a New Organization**.



3. Fill out the fields outlined in red as shown below, and then click **Create a New Organization**.



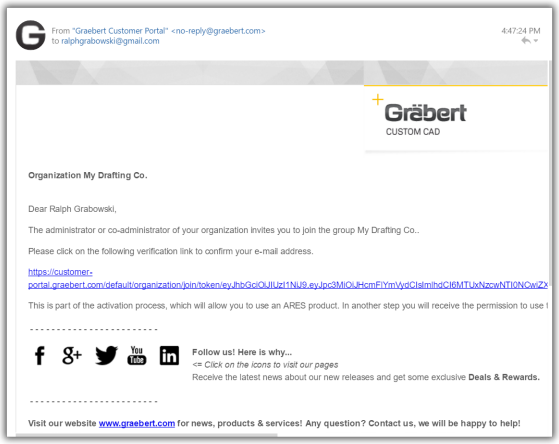
4. With the organization's details established, choose the **Users** tab, and then enter the email address of each person to be invited to access ARES Kudo.



When you have more then one email address, separate each of them with a semi-colon (;).

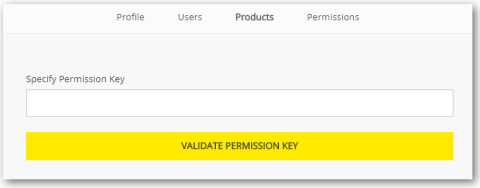
5. You can include the following information for each user: email address, first and last name, and title. As you are the one setting up the account, you automatically are the administrator ("Admin").

6. When done, click **Invite Users**. Each receives an invitation email generated by Graebert Gmbh. They must click a link to confirm their participation.

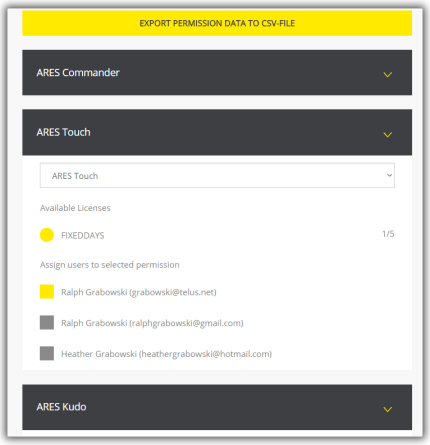


7. Upon clicking the link, the Create Account screen appears in their Web browser. The new user has to fill out the same information as shown earlier in Step 4 of "Registering an Account", including setting a password.

8. Now enter the permission key. This gives your firm access to the software you licensed from Graebert. Click **Validate Permission Key**.



9. With the users and software established, you now assign permissions to each user.



For detailed information about the topic of licensing, refer to the following Web page online: https://customer-portal.graebert.com/help/#t=html%2Fchapter_Licensing_ARES_Applications.htm

Gräbert

The Trinity of CAD

Desktop - Mobile - Cloud

LEARN MORE ABOUT THE ARES CAD SOFTWARE ON

WWW.GRAEBERT.COM

Don’t miss also all the great contents we published for you on:

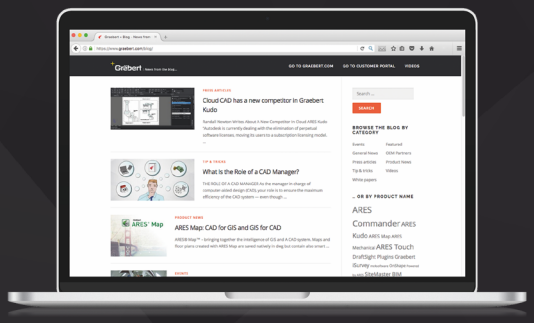
YouTube

youtube.com/graebertTV

Tens of videos
to get started, discover features,
and grow in expertise.

THE GRAEBERT BLOG

www.graebert.com/blog



Hundreds of blog posts with
news, white papers, tip & tricks,
press articles and more.

Follow us on your favorite social networks:

facebook
facebook.com/arescadsoftware

linkedin
linkedin.com/company/graebert-gmbh

twitter
twitter.com/graebertcad

Gräbert
The Trinity of CAD
Desktop - Mobile - Cloud

Website: www.graebert.com
Hotline: support.graebert.com

Nestorstrasse 36a | 10709 Berlin | Germany
Order Desk: licensing@graebert.com