



Walt Disney World Swan and Dolphin Resort  
Orlando, Florida

## Secrets to Building a Better Block, Dynamically - Revealed!

Matt Murphy - ACADventures

**GD41-3** Blocks are the oldest form of reusable content known to AutoCAD users. They are one of the first things we learn, and we use them all the time. Although the concept remains unchanged, there have been major developments that make blocks more powerful. Learn the fundamental rules for building better block libraries, adding and extracting attributes, and then explore the new standard for blocks on palettes. Discover the Authoring Palette which adds grip and action features to dynamic blocks for greater efficiency and productivity. What's more, you will be able to apply the new ideas to the existing block libraries you use in AutoCAD. It's time to rethink the way you use blocks!

### **About the Speaker:**

Matt has been recognized by Autodesk as a leader in providing professional training since 1985. He has served as an advisor and consultant to the training industry as chair of the Autodesk Training Center Advisory Board (ATCAB) and executive committee (ATCEC), and to AUGI. Matt is also an Autodesk Certified Instructor and a Certified Technical Trainer. He has received numerous instructional awards for his efforts in raising the quality of the training industry. A widely acclaimed top trainer and featured columnist for AUGIWorld magazine, he has presented at Autodesk University for 11 years as one of the event's most popular presenters.

**[matt.murphy@ACADventures.com](mailto:matt.murphy@ACADventures.com)**



## Introduction

Shortly after the first version of AutoCAD became available, the desire to create reusable content was born with the Block Command. Blocks are the oldest form of reusable content known to AutoCAD users. They are one of the first things we learn, and we use them all the time.

The knowledge of using blocks has been passed down from our ancestors and generations of previous AutoCAD users. We have used them the same way our predecessors did thousands of projects ago. Although the concept remains unchanged, there have been developments that make blocks more powerful. But because our predecessors set the standards for using blocks, we tend not to question the old proven methods. We also may be less receptive to try these new techniques as we all dislike change.

In *Secrets to Building a Better Block, Dynamically — Revealed!*, I will show you some old and some new ways to look at and work with blocks. We'll look at how to build Dynamic Blocks, extract data into AutoCAD and manage your Blocks on Tool Palettes. What's more, you will be able to apply the new ideas to your existing block libraries to make AutoCAD work better for you.

## Follow the Rules

Blocks will behave exactly the way you want, provided you follow the rules for creating them. Generally, when you insert a block, the color, linetype and lineweight of objects in the block retain their original settings regardless of the current settings in the drawing. However, you can create blocks with objects that inherit the current color, linetype and lineweight settings. These objects have floating properties. You have three choices for how the properties of objects are treated when a block is inserted.

1. Objects in the Block do not inherit color, linetype and lineweight properties. The properties of the object in the Block do not change regardless of the current settings. For this choice, it is recommended that you set the color, linetype and lineweight properties individually for each object before you create the Block definition. Do not use BYBLOCK or BYLAYER for the properties of these objects.
2. Objects in the Block inherit color, linetype and lineweight properties from the color, linetype and lineweight assigned to the current layer. For this choice, before you create objects to be included in the Block definition, set the current layer to 0 and set the current color, linetype and lineweight to BYLAYER when you create the geometry for your Block.
3. Objects inherit color, linetype and lineweight from the current color, linetype and lineweight. This is like setting an override by not assigning the property from the current layer. For this choice, before you create objects to be included in the Block definition, set the current color, linetype or lineweight to BYBLOCK.

### Simply stated:

A Block will take on the properties of the current layer when inserted, provided it has either been created on Layer 0 or with the properties BYLAYER.

A Block will retain its original properties from the layer it was created on when the properties of objects have been set to BYBLOCK.

## Orientation and Insertion Point

What angle do you create a Block at? The angle of the Block and the insertion point are critical to accurate placement and orientation. Don't get fixed on having the insertion point on the object. Sometimes it makes more sense to track and use a reference location away from the object. You

need to think about how this Block is going to be placed and what angle it typically will be used at. That is how you determine what is best. Choose the best and most common use.

**SECRET REVEALED:** Don't know where the insertion point is on a Block? Use the POINT command to create a locator to identify the insertion point of the Block when you create the Block definition. This way you can turn the point visibility on with Point Style if you need to see it within the drawing.

### Don't Become a Blockhead

Blocks from the DesignCenter or another symbol library may be edited to meet your needs. Orient the geometry at the angle "most" commonly associated with its use. Define the insertion point that will optimize its efficient use later. Nothing wastes more time than inserting the Block and then having to move it because the insertion point is not where it should be. If a symbol can have more than one logical insertion point, choose one and select a standard.

### What's in a name?

Develop a naming scheme that makes sense to you and others. AutoCAD allows the use of up to 255 characters to name a Block. "Just because you can, doesn't mean you should." Again, keep it simple. Evaluate all the types of Blocks you use. Categorize them. Use hyphens and underscores. Keep it simple.

**A\_Door-90** for Architectural Door with 90° Swing

**P\_Tub-60** for Plumbing 60" Tub

**E\_Fix-CM** for Electrical Ceiling-Mounted Fixture

**TB\_Mech-A** for A-size Mechanical Title Block

### What is the standard?

This is a course in itself! Every company has developed standards for drawing creation. This is also true for the use of Blocks. In order for compliance to happen, good communication must take place. There is a variety of successful ways to communicate to your users. Once the Blocks have been created, print your Blocks in a hard copy 3-ring binder that is updated regularly. Also publish them in an electronic version. PDF format is the best as it protects the master document definitions. Each Block should have an image, the insertion point location, explanations, hyperlinks, bookmarks and thumbnails. Protect the Blocks in a read-only symbol library accessible through DesignCenter. You can also create custom Block palettes to organize your Blocks in more easily accessible ways within AutoCAD.

## New Rules and Special Instances for Block Busters

### Unit Block

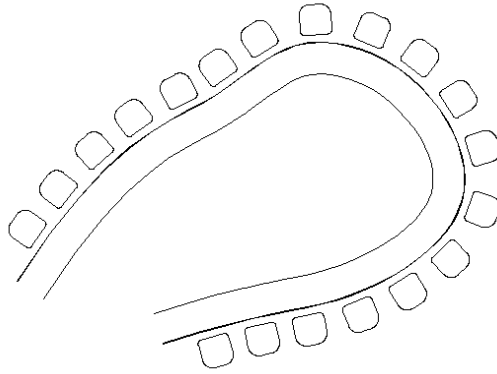
When you create the geometry, the general rule is to create it at full scale, just like your drawings. You should also have a Block for each unique type of object to be used. Although this makes sense for the majority of Blocks, you could wind up with thousands of extra Blocks. There are exceptions to this rule. Sometimes efficiency can be found in simplicity. Think about this: a 1" line and 1" radius arc can represent a door Block that may then be inserted at any opening size. Likewise, a 1" x 1" rectangle can represent a table Block that may then be inserted at any size. When you create a Block that is drawn to the lowest denominator size you can then insert it at multiple scales. This is called a Unit Block.

### Using Blocks with Divide and Measure

How about creating a multiple insert? Well, there is a multiple insert command that works like a rectangular array. But what about placing Blocks at specific distances along an line, arc or polyline?

## Secrets to Building a Better Block, Dynamically — Revealed!

There is a little-known feature of the DIVIDE and MEASURE Commands which allows you to use a Block instead of a Point. This technique also works with a polyline that is a spline curve as well.



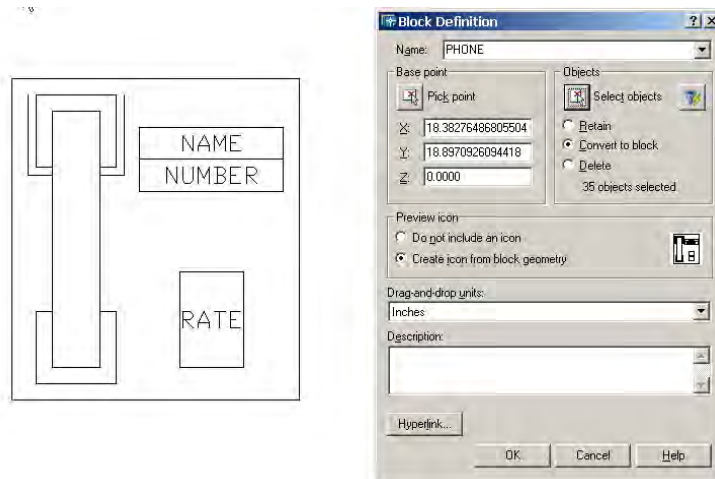
### Nested Blocks

Instead of inserting a standard suite of Blocks one at a time, you could create a nested Block. A nested Block is simply a Block with Blocks within it. Inserting a nested Block brings all the nested insertions with it. Let's say you are laying out a bathroom. You could have a Block for each fixture and insert them one at a time, or you could have a Block called "bath" that contains all the Block definitions for all the fixtures. Using this method, you insert one Block called "bath," explode it, and then move the individual bath fixtures into the appropriate locations.

### Blocks with Attributes

Attributes are essentially text that can be associated with each specific incident insertion. The text can be alpha or numerical values. Assigning attributes to Blocks adds power and quality to our drawings. Once the attributes have been assigned, they can then be exported into a database or spreadsheet program giving us material, parts or inventory lists from our drawings. Sometimes the attribute is used to give us control over size and location of text. In the past, creating, editing and managing attributes was the duty of the CAD Gods. Now, in a few short sessions, you too can become omnipotent. The process is quite simple:

1. Create the geometry as you would for a normal Block
2. Define the attributes
3. Create the Block by choosing the objects and attributes
4. Insert the Blocks into the drawing
5. Extract the data



### Defining the attribute

Here's where the data is defined. Think of an attribute as a tag that is attached to the Block. You can have as many tags as you need. Each tag has a label and placeholder to add your text information. If you forget what information goes on the tag you can add a prompt to remind you. Only one attribute can be defined at a time. The command is repeated to define additional attributes to one Block.

**Modes:** State of data assigned to Blocks that may be extracted into an external file.

*Invisible:* does not display on the drawing

*Constant:* static information that never changes

*Verify:* requires user verification that information is correct

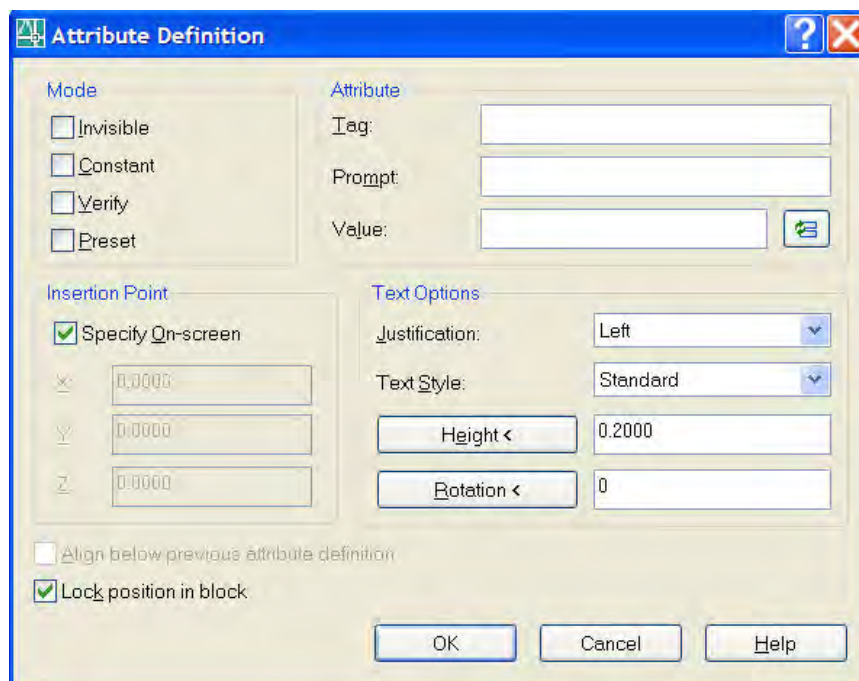
*Preset:* predetermined value that can be changed

**Attribute:** Parts that comprise the definition.

*Tag:* Attribute name (256 characters, no spaces)

*Prompt:* Upon insertion, prompts user to assign value

*Value:* Default value (omit \$, %, etc., symbols for future extraction purposes)



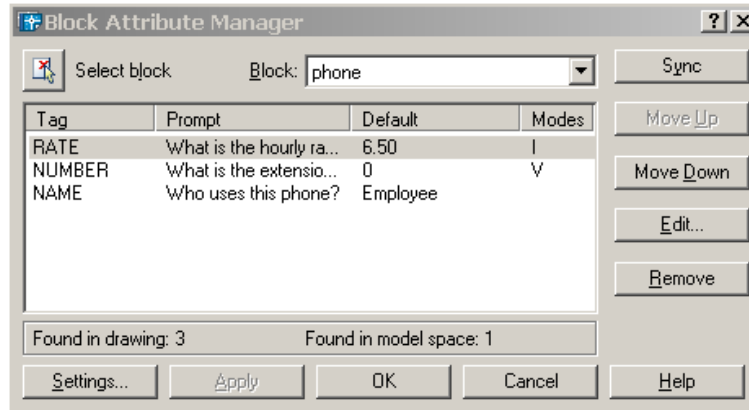
**Insertion Point & Text Options:** Defines where and how the attribute text will appear in the drawing. Remember that attribute text can be justified just like Single-line Text or Dtext. Choosing an accurate location and justification will help maintain the location of the attribute. This will make revisions easier.

**Align below previous attribute definition:** Once first attribute is assigned, if checked, the next attribute will be automatically aligned for you. You won't be prompted for the insertion or text options.

**SECRET REVEALED:** When you create the Block definition, select the attribute tags in the order you would like the prompts to be asked. Otherwise the last definition that is created is the first tag to be prompted.

## Managing Block Attributes

BATTMAN is the command used to open the Block Attribute Manager. Here you can edit various properties of attributes, remove attributes and synchronize redefined Blocks. BATTMAN is a “global editor.”



**Block:** Lists all Blocks with attributes in the drawing.

**Sync:** Allows the user to apply attribute definition changes to all instances of the Block.

**Move Up/Move Down:** Allows user to reorder the attributes.

**Edit:** Allows user to change the mode, tag, prompt, default value, text options and properties.

**Remove:** Allows user to remove attribute from Block definition.

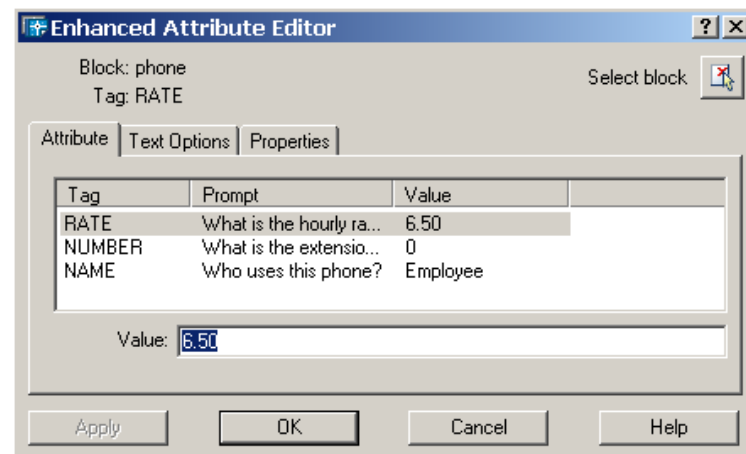
**Settings:** Allows the user to change what is displayed in the attribute list.

**Emphasize duplicate tags:** If checked will display duplicate attributes in color.

**Apply changes to existing references:** If checked will update all existing Blocks.

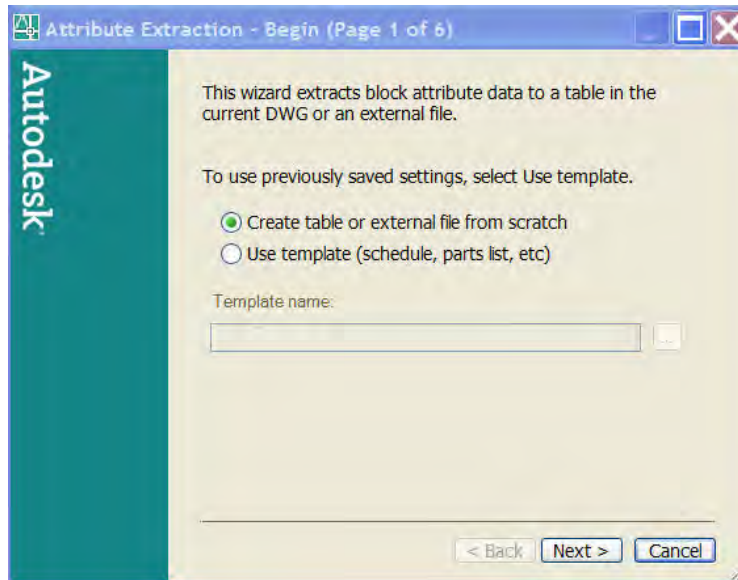
## Editing Attributes

EATTEDIT is the command used to open the Enhanced Attribute Editor. Here you can edit the value, text options and properties of an attribute. Notice the same three tabs (Attribute, Text Options and Properties) as seen in BATTMAN. The primary difference is that EATTEDIT is a “single-instance editor”; therefore, the tags and prompts cannot be edited here.



## Extracting the Data

EATTEXT is the command used to launch the Attribute Extraction Wizard. No more need for cryptic template files. Here even the novice user can create an extraction file to export attribute data into an Excel spreadsheet or Access database.



**Select Drawing:** Allows the user to select objects in the current drawing, all Blocks in the current drawing, or external drawing files from which to extract attributes.

**Settings:** Allows the user to choose whether to include xrefs and nested Blocks in the extraction.

**Use Template:** Allows the user to select Block attribute settings from previously saved template file.

**Select Attributes:** Allows the user to select the specific Blocks and specific attributes per Block to include in the extraction.

**View Output:** Allows the user to preview the output format of the extracted information.

**Save Template:** Allows the user to save the attribute extraction settings to a template file with the *.blk* file extension.

**Export:** Allows the user to specify the attribute extraction file name and information, and then export the attribute information to the specified file.

## System Variables and Commands to Know

**ATTDIA:** when set to 0 will disable the Enter Attributes dialog box and send the prompt to the command line.

**ATTDISP:** has three settings: NORMAL (default) will display attributes as defined in BATTMAN; ON will display all attributes, including those set to “invisible”; and OFF will disable the display of all attributes.

**DDATTE:** may be used to edit attribute values, but not properties.

**ATTREDEF:** may be used to redefine a Block and attributes.

**ATTREQ:** determines whether the Insert command uses the default attribute setting during insertion of Blocks. Set to 1, turns on prompts or dialog box for attribute values. Set to 0, assumes defaults for the values of all attributes.

**SECRET REVEALED:** Since most of the variables and commands that control Blocks are NOT in menu or Toolbar, I would create a Toolbar that contains a button for each of these. A simple menu

macro can be written for the command string. Now I can have them in an easily accessible place to pick instead of typing.

## Dynamic Blocks

### Why should I use dynamic blocks?

Blocks, an essential part of nearly any drawing, are used to represent real-world objects. Different variations of real-world objects can require you to define just as many variations of blocks. Dynamic Blocks will reduce your block library size. They can be locked and protected from being exploded. They can be placed on a Tool Palette for greater organization and they will make you more productive by maintaining standards and reducing your clicks and picks.

### Block references are easier to use

Introduced in AutoCAD 2006, dynamic block functionality enables you to edit the appearance of block instances without having to explode them. You can even manipulate a block instance during and after inserting it into a drawing.

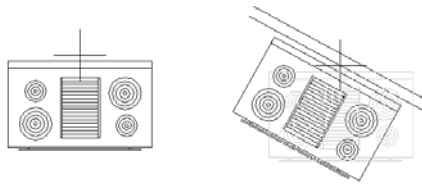
### Traditional blocks:

- Multiple steps to place and align
- Design changes require erasing and reinserting or exploding, editing and redefining

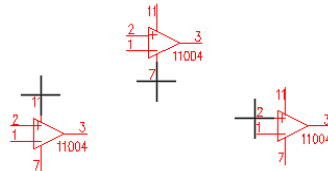
### Dynamic blocks:

- Automatically align to nearby geometry
- Cycle between multiple insertion points
- Change visibility of geometry to streamline design changes
- Edit geometry within a block without exploding

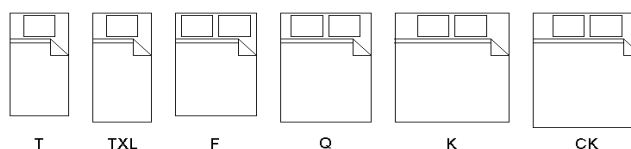
As you move your cursor near drawing geometry, blocks will automatically align themselves with other objects.



As you insert a dynamic block, you can cycle between key insertion points to find the one that makes the most sense for your current situation. The ability to cycle through insertion points can eliminate the need for you to move the block after it is inserted.

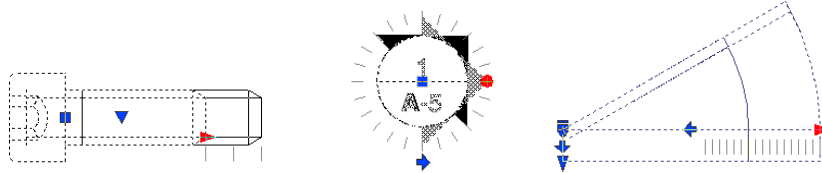


Block definitions can contain multiple representations of a particular symbol. Upon insertion, you can choose which representation to use. For example, a single block definition could store multiple representations of a bed, faucet, door or valve.



## Understanding Dynamic Blocks

Parameters control the actions of Dynamic Blocks. Parameters are special grips within dynamic blocks that enable you to move, scale, stretch, and rotate, array, and flip individual block geometry. Parameters define the feature of the block that you can change. For example, you might have a bolt block, which you can stretch to a total length between 1 and 4 units. As you stretch the bolt, the length is constrained to .5-unit increments and threads are automatically added or removed as you stretch the bolt. A second example might be a callout block that includes a circle, text, and a leader line. You can rotate the leader around the circle while the text and circle remain static. A third example might be a door block. You can stretch the door width and flip the direction of the door swing.



## Working with Dynamic Blocks

A dynamic block will appear with a lightning bolt next to its preview before you insert it.

When you insert a dynamic block, you can toggle through or cycle through the multiple insertion points with the CTRL key before you place it.

Once you insert a dynamic block, you can edit it using the Parameter control grip. Now you can grip edit blocks the way we have grip edited other types of AutoCAD geometry. There is no need to explode it.

**SECRET REVEALED:** use the Dynamic Parameter Grip to manipulate the geometry of the Dynamic Block.

### How do I create dynamic blocks?

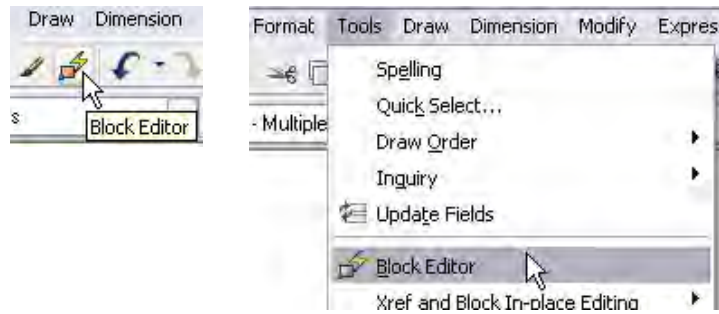
The block editor enables you to create new block definitions or update your existing blocks. You can access the block editor from several locations and then use the block authoring tools to add parameters and actions to your block definition.

### Use the Block Editor

The block editing environment is specifically designed for defining blocks. Make sure that the BLOCKEDITLOCK system variable is set to 0. You can access the block editor using any of the following methods:

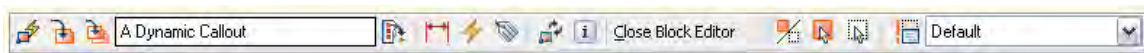
- BEDIT command
- BE command alias
- Standard toolbar
- Tools menu
- Right-click menu with a block selected
- Block Definition dialog box

## Secrets to Building a Better Block, Dynamically — Revealed!

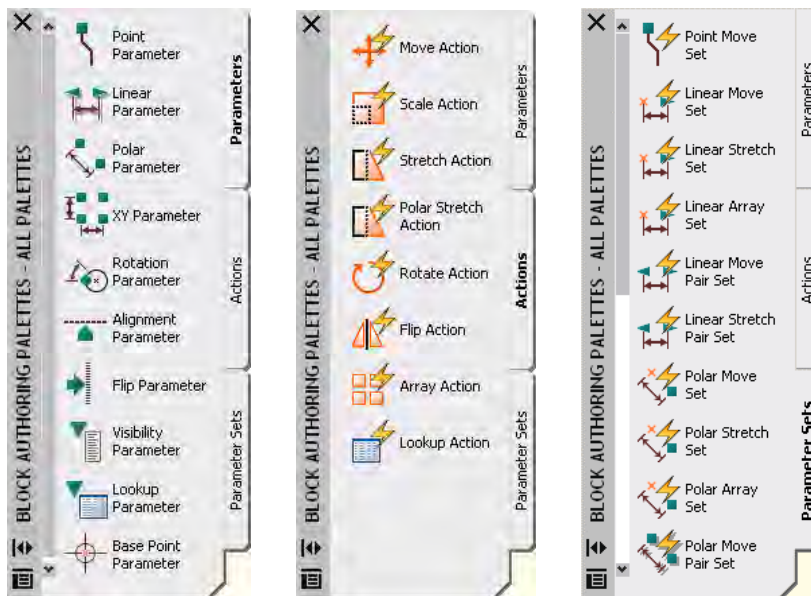


### Add parameters and actions to block definitions

In the block editor, you can use typical AutoCAD drawing and editing functionality to create and modify the geometry for your block definition. In addition, the block editor includes a toolbar and block authoring palette, which enable you to apply parameters and actions to your block geometry.



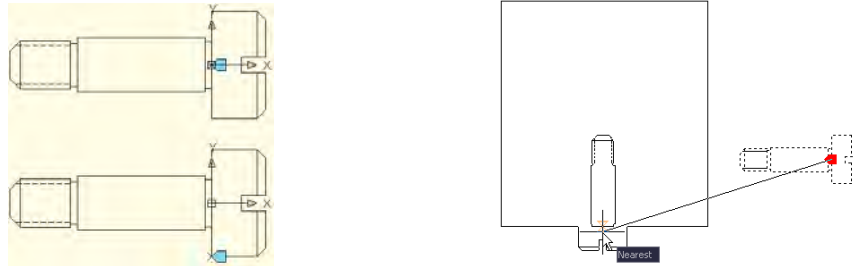
The block authoring palette includes three tabs. The first tab contains all of the available parameters. The second tab contains all of the available actions. And the third tab contains sets of the most commonly used combinations of parameters and actions.



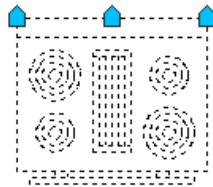
Parameters and actions work together to provide the editing capability of dynamic blocks. Parameters are dimensions that drive the block geometry. They are objects with their own relevant properties, which you can edit using the Properties window. For example, you can create a linear parameter to drive the width of a door and then apply properties that constrain the door width to 2-inch increments between the values of 18 and 36 inches. Actions are what change the geometry as you edit a block instance. For example, if you want to change the width of an inserted door block, you must apply a stretch action to the linear parameter that defines the door width.

Each parameter only works with specific types of actions and a few parameters require no actions. We'll begin with the most basic parameters, the ones that don't require any actions: Alignment, Visibility and Base Point. These parameters are easy to create and they can dramatically increase the efficiency of your existing blocks with minimal effort.

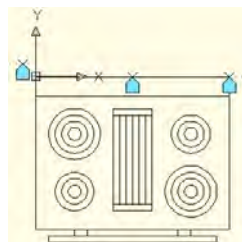
Alignment parameters require no actions although they can be included in the selection set of an action. You can add an alignment parameter to enable a block to align automatically to nearby geometry. If you create the alignment parameter at the origin point of the block, the alignment functionality will be available upon insertion. If you do not add an alignment parameter at the origin point, the alignment capability is only available when you select the alignment grip on an inserted block.



For some blocks, you might find it useful to include multiple insertion points. For example, when inserting a stove block, you might want to insert it using the left corner, the right corner or the center, depending on the other geometry in the drawing. You can define your blocks with multiple alignment parameters and then use the CTRL key to cycle between the alignment grips upon insertion.

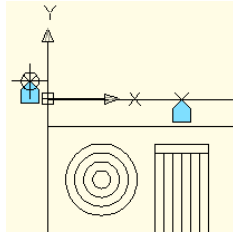


The parameter grips are actual objects in the block editor. If you don't want a grip to be included in the cycling options, you can select the grip in the block editor and then use the Properties window to turn off the Cycling option. If you want to change the location of the alignment grips, you can move the grip in the block editor. For example, if you want the stove to insert a slight distance away from the wall, you would create or move the alignment grip so that it is away from the stove geometry. In this example, changing the location of the left alignment grip means that it is no longer located at the origin point of the block. Of course, you could move all of the geometry so that it is in the correct relation to the origin. However, the simpler option is to insert a basepoint parameter.



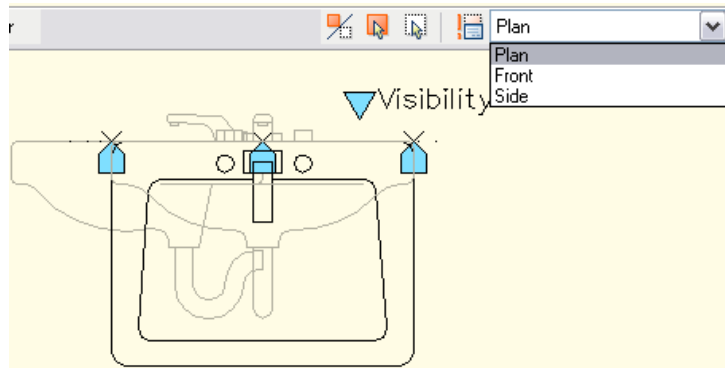
The base point parameter defines a base point for the dynamic block reference relative to the geometry in the block, overriding the default origin point of the block. Like the alignment parameter, the base point parameter does not require any actions, but can belong to an action's selection set. The base point parameter displays in the block editor, as a circle with crosshairs.

## Secrets to Building a Better Block, Dynamically — Revealed!



You can use the visibility parameter to define a block that turns geometry on or off depending on the state. A visibility parameter must include at least two states and you can use the visibility tools in the upper right corner of the block editor to switch between states and to show or hide the geometry for each state. For example, a valve block might include five visibility states. The block definition contains the geometry for all of the states but the visibility of the geometry varies between each of the states.

You can combine multiple types of parameters to create a more powerful block definition. For example, you might create a sink block with a visibility parameter to turn on different nested blocks for the plan, front and side views. In addition, you could include alignment parameters at the appropriate locations in each of the three visibility states.



**SECRET REVEALED:** create and edit your existing Blocks and Dynamic Blocks using the new Block Editing tool.

**SECRET REVEALED:** Although you can create individual parameters for each action, that would require you to grip edit three different times (one for each action). Instead, create one parameter and let it drive all three actions. In this example, you know you need a parameter that supports the Move, Stretch and Scale actions. In the following table, your possible parameter choices are Linear, Polar and XY.

Parameters

Alignment								
Visibility								
Base Point								
Point								
Linear								
Polar								
XY								
Rotation								
Flip								
Lookup								
	Move	Stretch	Scale	Array	Polar Stretch	Rotate	Flip	Lookup

Actions

After you determine which combination of parameters and actions will enable you to edit each individual object to meet your needs, you can begin adding the parameters. The process for adding parameters varies depending on the type of parameter. After you add the parameter, you may see an exclamation symbol indicating that the parameter requires an action. You can either select an appropriate action from the block authoring tools, or you can double-click on the parameter and select from the list of actions that are available for that parameter. For each action that you add, you will select the specific object(s) that you want that action to edit.

You can continue to add parameter and actions to make your blocks more powerful. In the door example, you might want to add a base point parameter and a flip parameter and action. Using the Properties window, you can apply various properties to the actions, parameters and parameter grips. For example, you might want to change the names of the parameters and actions to something more meaningful.

**SECRET REVEALED:** It is best to add all of the parameters before adding the actions because you might want to include some parameters in the selection sets of other actions. However, as you are learning to create dynamic blocks, you might find it helpful to get one piece working at a time. You can always add to an action's selection set by double-clicking on the action or clicking on the number of selection set objects in the Properties window.

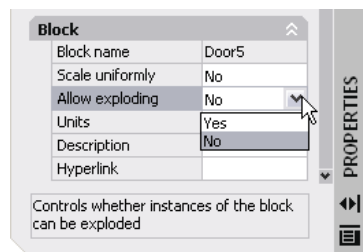
**SECRET REVEALED:** Modify your existing Block libraries to give them greater functionality.

### Add Parameters to Your Existing Blocks

Dynamic Blocks are very powerful, but who has time to create all new blocks from scratch? The easiest way to gain the power of Dynamic Blocks is to streamline your current Block library and add parameters for greater productivity. With a few simple clicks you can add power to your existing blocks with little time and effort.

**SECRET REVEALED:** Add parameters that require no actions. Add point, flip, alignment and visibility parameters to your existing blocks.

**SUPER SECRET REVEALED:** Lock your block! A new option in the Properties window enables you to prevent someone from exploding the block references. To access this option, open your block definition in the block editor. Use the Properties window without any objects selected.



**SUPER SUPER SECRET REVEALED:** Lock the block editor! Prevent your renegade designers from editing your dynamic blocks with the system variable BLOCKEDITLOCK set to 1. Then undefine the command.

**SECRET REVEALED:** Place your Blocks on a Tool Palette and access them from there.

## Tool Palettes

### Now put your blocks on a Palette!

#### Creating a Block Tool

The power of the Tool Palettes' Block tool allows you to assign properties such as layer, color, etc. to the blocks on a palette. There are even more powerful properties of the Block tool that allow you to set rotation and scale and even whether the block should be exploded or not when it is inserted into a drawing.

The procedure for creating Block tools from an entire drawing is easy. Right-click on any drawing file found in DesignCenter and choose Create Tool Palette. A new palette will be made from all the Blocks within the drawing with the name of the drawing as the name of the palette.

**SECRET REVEALED:** Make sure you use the purge command three times on any drawing you intent to create a Tool Palette from, as anonymous blocks will be added to the palette with this technique.

**SECRET REVEALED:** You can also use this exact same technique for your ancestral block libraries that exist on your network drive. Simply navigate to the folder in DesignCenter, right-click on the folder and choose Create Tool Palette.



**SECRET REVEALED:** Create one drawing file for each previous named Block folder.

**SUPER SECRET REVEALED:** You can also change the scale and rotation of a block tool without exploding or redefining the Block. Right-click on the Block Tool and choose Properties.

#### Windows and Internet Explorer

Autodesk enabled the use of dragging and dropping files directly from Windows Explorer onto a Tool Palette. This works in a similar way to dragging content from the DesignCenter to a Tool Palette. Some of the files types that can be dragged to the Tool Palette from Windows Explorer are drawing files and image files.

**SECRET REVEALED:** Create small simple Tool Palettes, lock them and demand-load those smaller palettes as you need them.

## Managing Tool Palettes

### Tool Palette Groups

Tool Palette groups are a way to simultaneously control the display of certain Tool Palettes. You might have a Tool Palette for electrical and another for plumbing when creating building plans. Maintaining organization used to be difficult as all palettes were open and active at once. But this organization problem has been resolved with the Group feature when customizing, if you're using Tool Palette Extension in AutoCAD 2004 or AutoCAD 2005. By default, All Palettes is active.

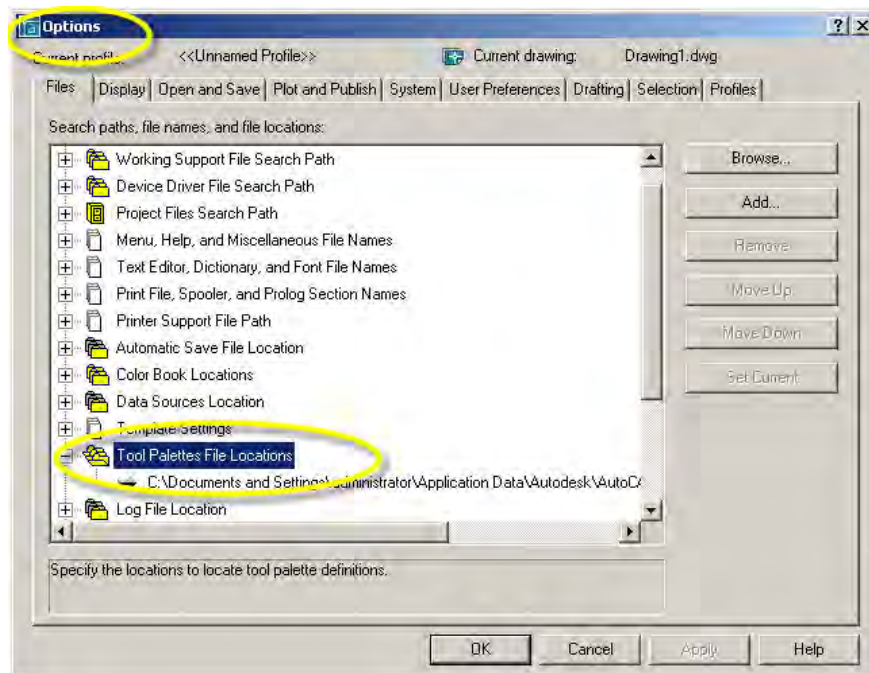
There are some problems with the group feature. Groups are not easily shareable, as they are stored with each user's AutoCAD profile. But I have a solution for you!

### Managing Tool Palettes with Paths

Just like managing your support path statements in AutoCAD, you can set a Tool Palette path location and not use the group feature.

To demand-load palettes, you can set the Tool Palette path using a system variable. Having multiple paths set with a small toolbar macro will allow you to switch between palettes. This technique uses a system variable called `*_TOOLPALETTEPATH` and allows you to change the location AutoCAD is looking in for Tool Palettes. By default the location is: "C:\Documents and Settings\Appl i cation Data\Autodesk\AutoCAD 200X\R16. X\enu\support\Tool Pal ette".

If you copy the Tool Palette directory to a location on the network and give it a different name, you can use a macro like the one below with a Toolbar button to switch to that location. This gives the illusion of groups, but in more of a manageable way: `^C^C*_TOOLPALETTEPATH "H: /Tool Palettes/Electrical Tool s"`. You could also write this macro into a custom command and access it from a shortcut key or menu.



**SECRET REVEALED:** You could also create a Command Tool which could be used to switch between different directories of Tool Palettes paths. You can use this method to share Tool Palette content while allowing users to add and manage their own content by having multiple Tool Palette paths, like this macro:

^^C\*\_TOOLPALETTEPATH "C: /Documents and Settings//Appl icati on Data/Autodesk/AutoCAD 200XR16. X/enu/support/Tool Pal ette, H: /Tool Pal ettes/Si te Pl an Tool s".

**SUPER SECRET REVEALED:** You can also create a blank palette window with no tools by setting an empty path or a path that does not exist. AutoCAD will automatically build the necessary XML content as an ATP file with support folder locations for images.

### Importing and Exporting Tool Palettes

A Tool Palette is exported and imported through the Customize dialog box. This is the same dialog box that is used for adding and modifying both Toolbars and Accelerator Keys.

You can specify both a name and location for the exporting of the Tool Palette. AutoCAD creates an XTP (eXported Tool Palette file) and a new folder containing the images for the tools. (I always wished toolbars did that.) If some images do not appear they were probably created from toolbar buttons. Darn! Also, since these tools use absolute paths in the tool, AutoCAD does not pull this information together when exporting a Tool Palette. This is why I mentioned earlier that Tool Palettes could be shared as long as the drive path in the tool property is the same.

To import a Tool Palette is pretty much the same as exporting a Tool Palette. Select Import instead of Export and the Import Tool Palette dialog box will come up. Browse to the location in which the XTP file and its associated images are stored and select it.

Once the XTP file has been selected, click the open button and the Tool Palette will be added. This will not overwrite an existing Tool Palette, if one with the same name already exists.

**Note:** The technique listed here is for "vanilla" AutoCAD only. Architectural Desktop (ADT) and Autodesk Building Systems (ABS) do not have an Import or Export option like the ones found in AutoCAD 2004 and 2005.

### Sharing Your Toys – I mean Tools!

Tool Palettes can be shared with others by exporting and importing, from one machine to another. The initial process is very easy, but can take some additional steps in getting them to work on a different machine. You must be careful with drive mapping and paths. If the mapping is not consistent from your machine to another the tools will not work. Palettes are designed for personal productivity, yet there are solutions to making and managing tools with multiple users.

### It's All Relative

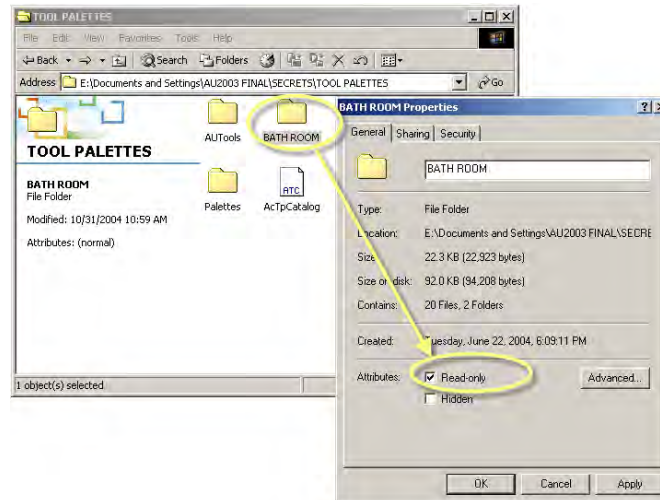
If you move the block from its path location there is no way to make a relative path. Block references from Tool Palettes search the current drawing first, mapped path second and then support file locations last.

**Note:** It might seem obvious but it's worth saying: Make sure the path locations for the source files exist in your support path locations under Options... Files.

### Protecting Your Tools

Once you have created your content and set your palettes in a shared folder, you'll want to protect them from being modified. Network drives can be set to read only and local drives can be set by right-clicking on the file in Windows Explorer and setting the properties of the folder to read only.

## Secrets to Building a Better Block, Dynamically – Revealed!



The next time you restart AutoCAD and open the Tool Palette Windows you'll see a small padlock in the lower corner.

**SECRET REVEALED:** Locking the Tool Palette folder only prevents the content from being changed. Individuals who have access to the folder that contains the definition will still have the ability to rearrange the tools on the palette as well as control the appearance of the palette.

**SECRET REVEALED:** When moving ATC files and folders to new locations, tools that appear in the palette will appear in the order they were created.

**SUPER SECRET REVEALED:** If you want to maintain the order of tools on shared palettes, you'll need to move the original profile (.AWS) file to the local machine. As long as the profile exists locally with the same name as the original profile, the .AWS file will maintain the order of the tools based on their most recent locations.

### Dynamic Blocks System Variables

**FINAL SECRETS REVEALED:** This class wouldn't be complete without a list of the common system variables that affect Dynamic Blocks.

**BAUTHORPALETTE** - Opens the Block Authoring Palettes window in the Block Editor.

**BAUTHORPALETTECLOSE** - Closes the Block Authoring Palettes window in the Block Editor.

**BCLOSE** - Closes the Block Editor.

**BCYCLEORDER** - Changes the cycling order of grips for a dynamic block reference.

**BEDIT** - Opens the Edit Block Definition dialog box and then the Block Editor.

**-BEDIT** - Opens the Edit Block Definition dialog box and then the Block Editor (command line).

**BGRIPSET** - Creates, deletes or resets grips associated with a parameter.

**BSAVE** - Saves the current block definition.

**BSAVEAS** - Saves a copy of the current block definition with a new name.

**RESETBLOCK** - Resets one or more dynamic block references to the default values of the block definition.

**BACTIONCOLOR** - Sets the text color of actions in the Block Editor.

**BGRIPOBJCOLOR** - Sets the color of grips in the Block Editor.

**BGRIPOBJSIZE** - Sets the display size of custom grips in the Block Editor relative to the screen display.

**BLOCKEDITLOCK** - Prevents opening of the Block Editor and editing of dynamic blocks definitions. Set it to 1. By doing this, when you double click on a block it will open the REFEDIT feature rather than the Block Editor. By default in AutoCAD 2006 the BLOCKEDITLOCK is set to 0. This prevents users from editing Dynamic Blocks.

**BLOCKEDITOR** - Reflects whether or not the Block Editor is open.

**BPARAMETERCOLOR** - Sets the color of parameters in the Block Editor.

**BPARAMETERFONT** - Sets the font used for parameters and actions in the Block Editor.

**BPARAMETERSIZE** - Sets the size of parameter text and features in the Block Editor relative to the screen display.

**BTMARKDISPLAY** - Controls whether or not value set markers are displayed.

**GRIPDYNCOLOR** - Controls the color of custom grips for dynamic blocks.

**INSUNITS** - Specifies a drawing-units value for automatic scaling of block, images or xrefs inserted in or attached to a drawing.

## Summary

Blocks were the first source and remain the staple for standardizing reusable content. Although they have fundamentally remained unchanged, they now have greater power, flexibility and usability than the Blocks and Wblocks used by our design ancestors. The incorporation of DesignCenter and Tool Palettes allow us to manage this content better. You know the Block is useless if you can't find it. Tool Palettes not only keep Blocks easily accessible but they maintain the path and location of the Block Tool. This helps us maintain our standards. Tool Palettes can also be locked to prevent users from adding and deleting content as well. Good reusable Block content is the key to productivity and consistency.

I believe that the creation and incorporation of Tool Palettes into AutoCAD is the greatest productivity tool ever! Adding parameters (that require no actions) to your existing block library allows you to increase power and functionality with very little effort. And for those of you that have plenty of spare time, you can rebuild your block libraries or build new dynamic block libraries with actions associated with parameters.

It's all about the clicks and picks. If you learn to apply tool palettes for blocks, you'll never use the insert command again. If you add dynamic parameters to your existing blocks, you'll never need to use the explode command again. In both cases you will be more productive because you will be able to do your design work with fewer clicks and picks. Reducing repetitive steps and reducing clicks and picks is how you become more productive working with AutoCAD, period!

