



Walt Disney World Swan and Dolphin Resort
Orlando, Florida

The Dos and Don'ts of Dynamic Blocks Revealed!

Matt Murphy - ACADventures

GD31-4 Blocks are the oldest form of reusable content known to AutoCAD users. Although the concept remains unchanged, there have been major developments that make Blocks more powerful. Dynamic Blocks have added a new level of power and complexity that is not easily mastered. Discover how the new grip and action features of Dynamic Blocks will increase your efficiency and productivity. You will not only learn how to build new Dynamic Blocks but also how to apply the new ideas to the existing Block libraries you use in AutoCAD everyday.

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

Introduction

Blocks have been around since the very early days of AutoCAD. They were the first reusable content we created and our ancestors of AutoCAD built thousands of them, stored them in read-only folders and we share them for everyone to use. The three basic reasons for using them hasn't changed:

- 1- They are easy to manipulate as they are unified
- 2- They provide consistency for standards parts and details
- 3- they reduce drawing file size as each block is a single data reference

Yet since AutoCAD 2004 there have been three major improvements in the use and creation of blocks:

- 1- The insert command is now obsolete with the introduction of tool palettes
- 2- Attribute data and fields can now be easily extracted back into the drawing and updated in tables
- 3- Blocks have become more flexible for manipulating and modifying, making the explode command obsolete

In this session we'll look at how Dynamic Blocks are more flexible to manipulate than your existing traditional blocks. Not only will we cover how to create Dynamic Blocks but also how to convert your existing traditional blocks into Dynamic Blocks and to manage them on Tool Palettes with very little effort.

I will also reveal many of the secrets that elude most users when creating and using Dynamic Blocks.

What are Dynamic Blocks?

They are blocks that can have a variable appearance and placement using grip constraints. Dynamic Blocks are flexible and intelligent. They can automatically align themselves with existing geometry. They can have multiple insertion points. They can change visibility, representing multiple versions in one block instead of inserting multiple variations, and you can edit the geometry without exploding. This enables you to modify the appearance of individual block references rather than searching for other block definitions to insert or redefining the existing ones.

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.

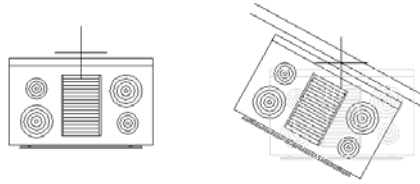
Traditional blocks:

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

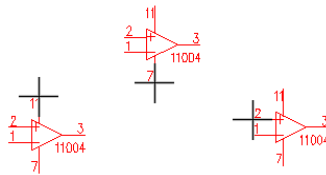
Dynamic Blocks:

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

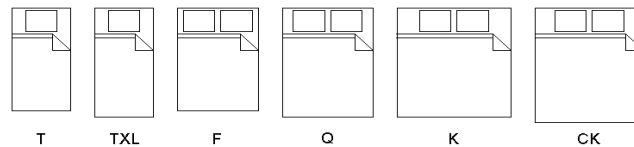
As you move your cursor near drawing geometry, blocks will automatically align themselves with other objects.



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.

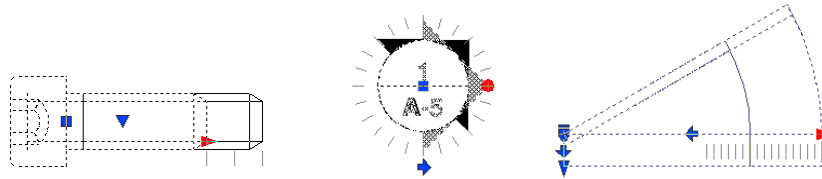


Block definitions can contain multiple representations of a particular symbol. Upon insertion, you can choose which representation to use. For example, a single block definition could store multiple representations of a bed, faucet, door or valve.



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 of 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

Dynamic Blocks will appear with a lightning bolt next to their preview before you insert them.

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 them.

***Don't* use modify commands like Move, Rotate, Scale or Stretch to manipulate the geometry of a Dynamic Block.**

***Do* use the Dynamic Parameter Grip to manipulate the geometry of the Dynamic Block.**

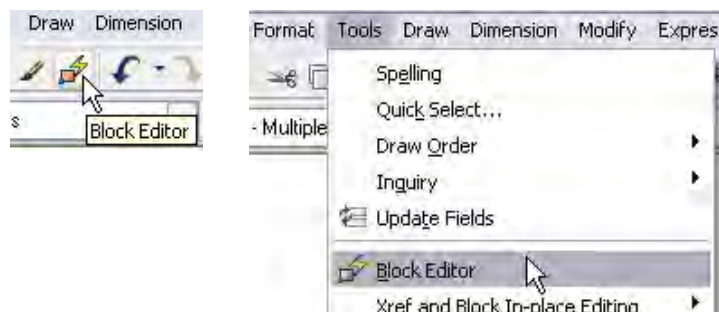
How do I create Dynamic Blocks?

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

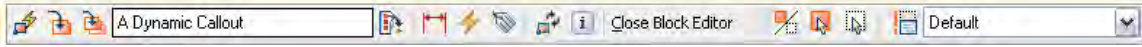
The block editing environment is specifically designed for defining blocks. Make sure that the BLOCKEDITLOCK system variable is set to 0. You can access the block editor using any of the following methods:

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

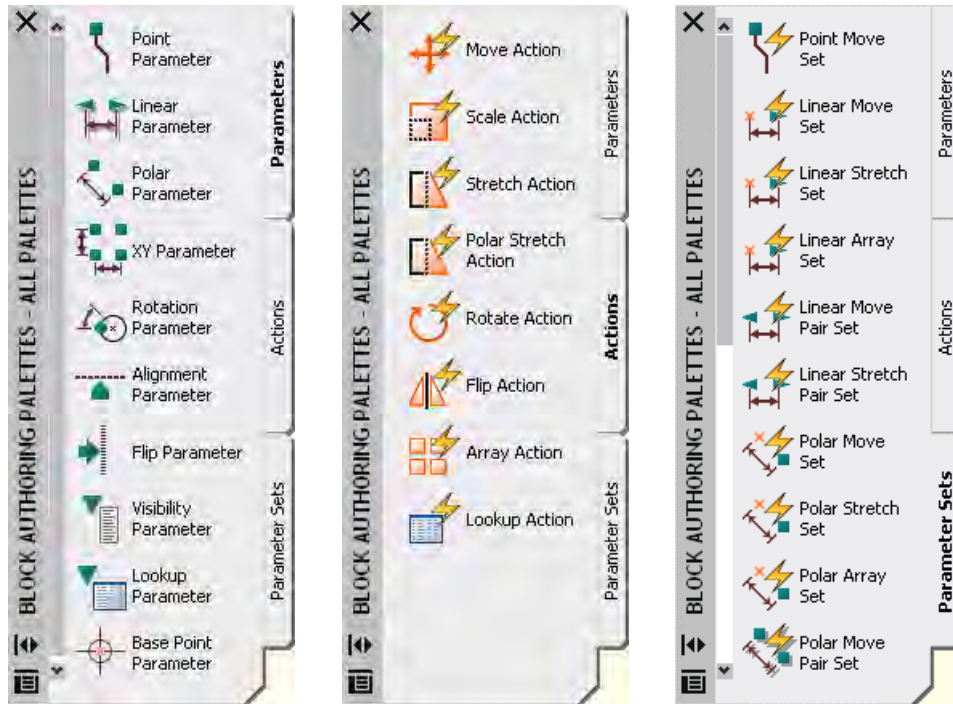


Add parameters and actions to block definitions

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 a 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.

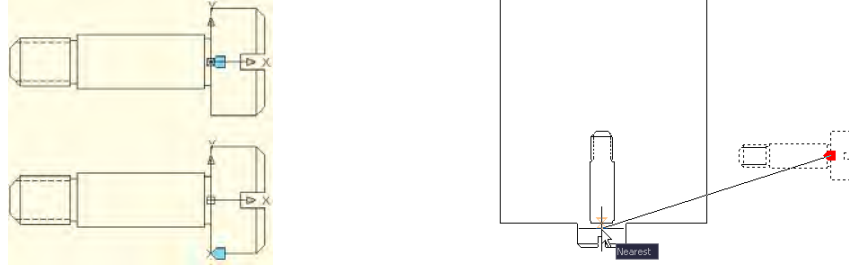


Parameters and actions work together to provide the editing capability of Dynamic Blocks. Parameters are dimensions that drive the block geometry. They are objects with their own relevant properties, which you can edit using the Properties window. For example, you can create a linear parameter to drive 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.

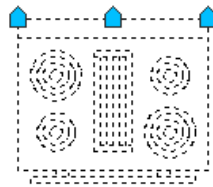
Each parameter only works with specific types of actions and a few parameters require no actions. We'll begin with the most basic parameters, the ones that don't require any actions: Alignment, Visibility and Base Point. These parameters are easy to create and they can dramatically increase the efficiency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

The Dos and Don'ts of Dynamic Blocks Revealed!

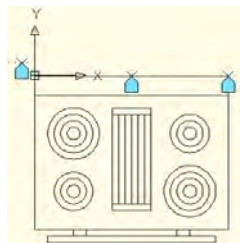
Alignment parameters require no actions although they can be included in the selection set of an action. You can add an alignment parameter to enable a block to align automatically to nearby geometry. If you create the alignment parameter at the origin point of the block, the alignment functionality 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.

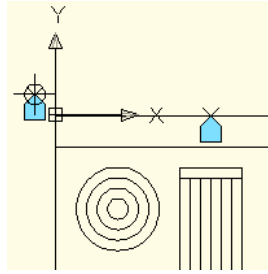


The parameter grips are actual objects in the block editor. If you don't want a grip to be included in the cycling options, you can select the grip in the block editor and then use the Properties window to turn off the Cycling option. If you want to change the location of the alignment grips, you can 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.



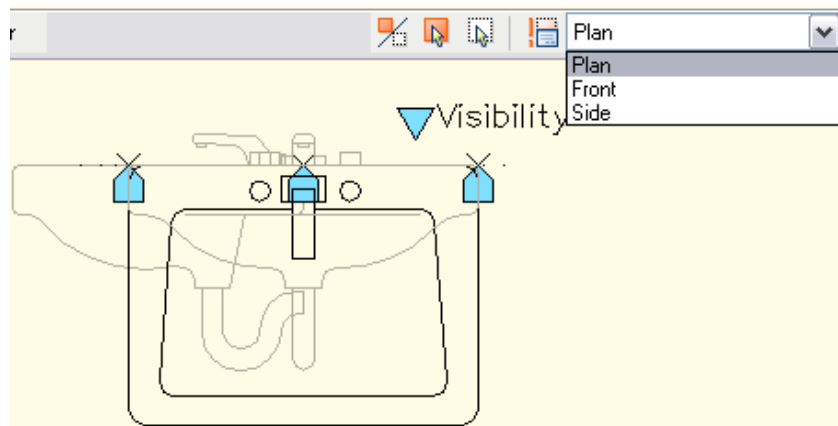
The base point parameter defines a base point for the Dynamic Block reference relative to the geometry in the block, overriding the default origin point of the block. Like the alignment parameter, the base point parameter does not require any actions, but can belong to an action's selection set. The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

The Dos and Don'ts of Dynamic Blocks 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.



Don't attempt to create or edit a Dynamic Block with Block, BMake, Refedit or with the Explode command.

Do 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.

The Dos and Don'ts of Dynamic Blocks Revealed!

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 parameters 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.

Don't create new Dynamic Blocks from scratch.

Do modify your existing Block libraries to give them greater functionality.

Add parameters to your existing blocks

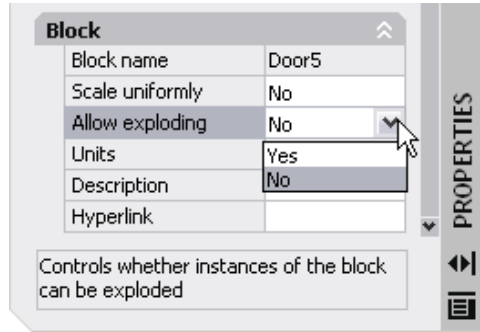
Dynamic Blocks are very powerful. But who has time to create all new blocks from scratch? The easiest way to gain the power of Dynamic Blocks is to streamline your current Block library and add parameters for greater productivity. With a few simple clicks you can add power to your existing blocks with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

Secret Revealed: Add parameters that require no actions. Add point, flip, alignment and visibility parameters to your existing blocks.

Don't leave your Dynamic Blocks open for modification when sharing.

Do disable Block editing and prevent exploding for organization control and to provide protection from renegade designers.

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.

Don't use the insert command to place Blocks into a drawing.

Do place your Blocks on a Tool Palette and access them from there.

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 intend to create a Tool Palette from, because 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.



Don't create individual Blocks as files using Wblock as your ancestors did.

Do 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 file types that can be dragged to the Tool Palette from Windows Explorer are drawing files and image files.

Don't create dozens of Tool Palettes that make it harder to find the Blocks you use.

Do 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 the Tool Palette Extension in AutoCAD 2004 or AutoCAD 2005. By default, All Palettes is active.

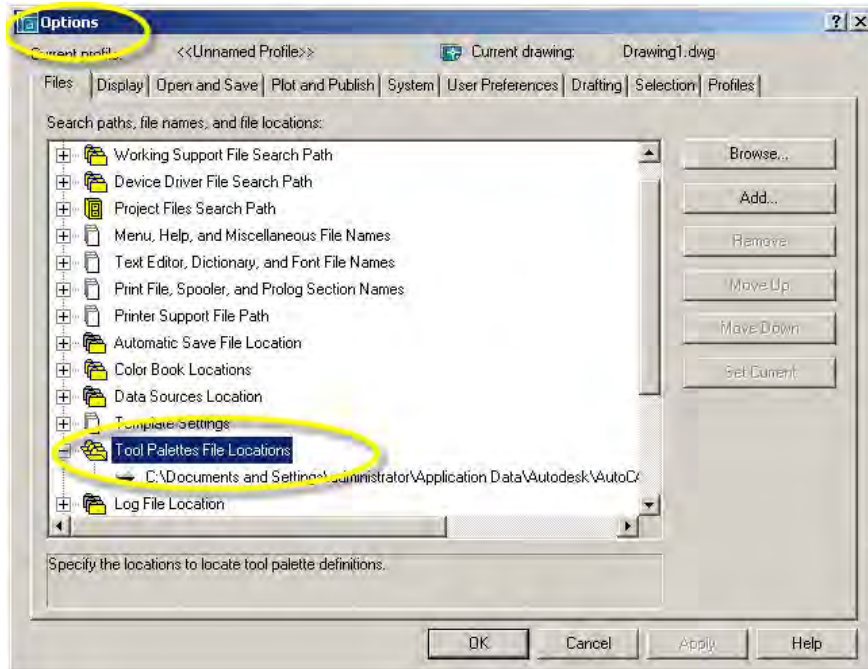
There are some problems with the group feature. Groups are not easily shareable, as they are stored with each user's AutoCAD profile. But I have a solution for you!

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 200XR16. 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*_TOOLPALETTEPATH "H: /Tool Pal ettes/El ectri cal 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 i cation Data/Autodesk/AutoCAD 200XR16. X/enu/support/Tool Pal ette, H: /Tool Pal ettes/Si te PI 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.

The Dos and Don'ts of Dynamic Blocks Revealed!

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.)

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.

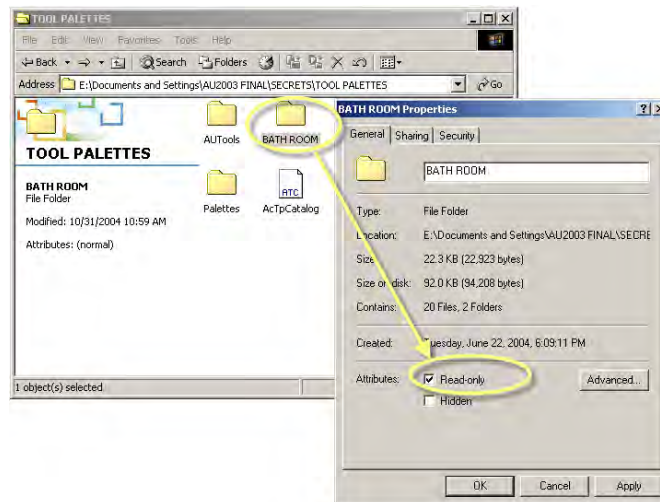
It's all relative

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.



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 to 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.

The Dos and Don'ts of Dynamic Blocks Revealed!

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 into or attached to a drawing.

Summary

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!

