



## Automating CAD Standards

Bill Fane – BCIT, CADALYST, etc

**CM211-2** Do you have trouble enforcing CAD standards? You don't need a published manual and a police force! Learn how most of your CAD standards can be built into your AutoCAD interface. You'll learn how to do simple menu and Toolbar customizing so that you can have picks for Part outline, Hidden line, Center line, etc. Whenever the user selects one, it will automatically be set to the correct layer. We'll also take a close look at – and dispel any fears about – Tool Palettes, and a brief look at the standards checker and where it fits in.

### About the Speaker:

Bill is a recovering doorknob designer, having been a product engineer and then product engineering manager for Weiser Lock in Vancouver, Canada, for 27 years. An AutoCAD user since 1986, Bill has taught AutoCAD and mechanical design at the British Columbia Institute of Technology for more than 10 years, and teaches Autodesk Inventor classes at the Institute's training center. He is an active member of the Vancouver AutoCAD Users Society, and an equally active writer: He has authored "The Learning Curve" column for "CADalyst" magazine since 1986; written about Autodesk Mechanical Desktop (now AutoCAD Mechanical) and Autodesk Inventor for Autodesk's "Point A Toplines," and writes for "Inside AutoCAD Journal," "Design Product News," and "Cutting Tool Engineering." He has been lecturing at AU since 1995.

bill\_fane@bcit.ca



Autodesk  
University  
2007



## NOTE

This course is intended for advanced beginners through to early intermediate users. It is based on AutoCAD 2008, noting that 2006 significantly changed how customizing is done in AutoCAD. On the other hand, the differences between the other 2000-series releases, including LT, are negligible and should be obvious from the content. The basic principles remain the same; the only difference is in how 2006 and later implements them. We will cover both implementations.

### **“Standardization through Customization”**

Boy, that title sounds like an oxymoron if I ever heard one. George Orwell would be proud of it. The bottom line, however, is that it is true.

As I see it, there are two common scenarios in a typical AutoCAD environment. In the first case there is a company “CAD Standards Manual” that is two inches thick. It spells out chapter and verse in excruciating detail exactly how drawings should be created. It may or may not be policed to any extent and each operator/designer/drafter usually ends up developing their own style.

In the other case, there is no formal standard at all. Everything just sort of grew from how the very first operator happened to start out. Everyone “knows” how it should be done and will gleefully point out other operator’s transgressions after the fact, but only if they happen to find them. Each operator/designer/drafter eventually develops their own style.

Okay, campers, pop quiz; which is the worst-case scenario?

Answer: They both are.

Now try to edit and/or plot a drawing that you did not create. Worse yet, others have already edited it each in their own unique style. Chaos! You want to freeze and thaw layers but have to guess and experiment to find out what does what. You want to plot and it takes three tries to get the pen widths correct. Text fonts and dimension styles are totally random.

What’s that? You say you don’t have to imagine it? You have “been there, done that, bought the t-shirt.”

Well, you have come to the right place because there is hope.

I managed a design office with a dozen people. After about ten years and seven AutoCAD releases we had accumulated over 5,000 drawings. Virtually every drawing used the same layer names, colours, and line types. All text styles, fonts, and sizes were consistent. Title blocks, sheet sizes, common notes, dimension styles, and scale factors all matched. Anyone could easily edit and/or plot anyone else’s drawing because we all knew how every drawing worked. All drawings plotted the same way.

We did not have a written procedure manual.

*“Hold it! You have just contradicted either or both of the preceding worst-case scenarios!”*



Yes, because there is a third way of maintaining CAD standards and that is through suitable customization. If you set up your CAD systems properly then you won't need a manual because it will all be done automatically. The users will comply with the standards simply because it is easier to do so than it is to deviate.

### **Just Checking...**

Current releases of AutoCAD include the Standards Checker functionality. In the earlier manifestations they can be used to inspect a drawing to make sure it has been created or edited with due regard for established standards. Current versions include "Big Brother" functions that monitor a drawing as it is being created or edited and will object when deviations are detected.

Powerful as these functions are, they are still "after the fact". The processes we will cover in this course offer a pre-emptive strike to help avoid non-compliance situations before they even arise.

What makes more sense; an ambulance at the bottom of the cliff, or a fence at the top?

### **In the Beginning...**

The best place to start is with the "template" drawing. This may be a little fundamental to many of you, but it can be crucial to our process.

The last five or six releases of AutoCAD all have one thing in common; how they look when they initially start up or when they start a new drawing looks completely different from release to release. This is compounded by the fact that there are usually several variants within each release. We have the "traditional" dialog, "AutoCAD Today", no dialogue, etc etc.

That having been said, if you do a bit of digging you will find that they all contain the following three methods for starting a new drawing:

"Start from scratch" means almost exactly that. You get to choose "English" or "Metric" to set up some rudimentary settings but other than that it is a completely blank drawing.

Unfortunately, this is the default mode when earlier AutoCAD releases were first installed.

By AutoCAD's own admission, "Use a wizard" is intended primarily for panicky first-time users. It leads you through a series of dialogue boxes that do a crude sort of setup. Real users ignore it.

In order to turn yourself into a power user, all you have to do is to find the "Use a template" choice.

And now some background on "templates". A template file is simply a prototype drawing. Whenever you start a new drawing from a template then AutoCAD actually starts your new drawing as a copy of the template. This template can already contain anything that can be included in a regular AutoCAD drawing:

- layers can be predefined, and the current layer set
- text and dimension styles can be predefined and the current ones set
- snap, ortho, and other settings can be preset and turned on or off
- title blocks, borders, and standard notes can be included
- anything else you want



When you click on the “Use a template” choice, then on “Browse”, it brings up a file dialogue box that lists a large number of pre-defined templates covering ANSI, ISO, DIN, and JIS standards as well as plotting setup alternates. These typically include a title box and border set up in paper space along with one big viewport into model space.

On the other hand, as you should be beginning to suspect, you can also create your own templates.

There are two ways of doing this.

The first way is to start a new drawing and set everything up the way you want. When you are done, pick (File | Save As). This brings up a dialogue box that includes a “Save as type” scroll list which has a “Template file” option. All it does is save the drawing to a normal .DWG format but uses the extension .DWT.

The second way to create a template file is to simply rename any existing drawing file. That’s right; the only difference between a template and a drawing is that a template has the extension .DWT while a drawing has the extension .DWG. So why the difference? To match Microsoft standard terminology. How many Microsoft employees does it take to change a light bulb? They don’t; Bill declares “dark” to be the new standard.

*“I did that, but my template files don’t appear in the template list!”*

Not a problem. AutoCAD looks for template files in a specific folder, whose location depends on your release and /or other variant such as Mechanical Desktop, Architectural Desktop, and so on. To find the exact location for your installation, invoke the OPTIONS command (Tools | Options...)

Click on the “Files” tab in the Options dialog box. Expand the Drawing Template entry to see the full path, which varies with the release and variant. In earlier releases it will look something like

```
C: \Program Files\<<path>\Templ ate
```

while 2004 changed it to something like this:

```
C: \Documents and Settings\<<user name>\Local Settings\  
  Appl icati on Data\Autodesk\AutoCAD 200n\  
    R16. 0\enu\Templ ate
```

where <user name> is the user’s log-in name and n is the AutoCAD release number. The change was made to accommodate Microsoft standards, which now require that each user should be able to customize everything independently of other users on the same machine. So much for standardization...

If you save or move your templates to this folder then they will appear in the template list. Conversely, you can delete or move any of the supplied predefined templates that you will not be using in order reduce clutter, or “Browse” for the folder of your choice to make it the location for templates.

For our purposes, the best procedure is to point the templates to a new “custom” or “library” folder. If you are on a network, it should be a common network folder to which all authorized users have “read” rights so that every user is starting from the same templates.



While you are setting your options, take a look at the “Profiles” tab. You can also set up different profiles (“configurations”) with different template paths, and hence different templates, to suit different needs. You could create a profile for different disciplines or different clients or different projects. Later, simply setting a different profile to be current will reset the template folder path.

*“How do I set a specific template to be the default?”*

Easy. If you start a new drawing from a template, then next time you start another new drawing the last-used template becomes the default. If you have weeded out all the non-company templates from the template folder then the choice becomes limited to legal ones.

Earlier releases give the user a choice of starting from scratch, starting from a template, or using a wizard. If you use a template then “use a template” automatically becomes the default startup mode until you select another.

AutoCAD 2004 and later make it easier by defaulting to “use a template”, and you have to dig a bit to find the other options (hint: click the down arrow beside the “Open” button).

Now here is a little-known fact. When you start a “blank” drawing in releases earlier than 2004, AutoCAD does not really start a blank drawing. Instead, it actually uses the template *acad.dwt* for English or *acadiso.dwt* for metric drawings. These are two templates like any other, and are found in the .../templ ate folder. You can modify them yourself, like any other template. Thus, even if someone starts a “blank” drawing it will open up with everything pre-defined to suit your standards. No more “blank” drawings!

Here is another trick to help avoid blank drawings. When AutoCAD first starts, by default it opens a new drawing. It normally uses *acad.dwt* as the template, but there are two ways of solving this:

Edit *acad.dwt* to incorporate your standards, or

Select (*Tools | Options*). Select the *Files* tab, and then expand the *Template Settings* item. Expand the *Default Template File Name...* item, and then browse to your desired template. This then becomes the default whenever AutoCAD starts up.

Don’t forget, different profiles can have different template file name and location options. You can also have a series of desktop icons that launch AutoCAD under different profiles. At the end of the *Target* window simply add

```
/t "<profile name>"
```

where *<profile name>* is the name of the desired profile. You can easily create desktop icons that launch AutoCAD with suitable templates for specific projects, or clients, or disciplines, or whatever.

As you can easily see, a bit of work setting up suitable templates can accomplish two things:

1. It saves a lot of time on each subsequent new drawing. This has got to be the simplest and most cost-effective way to customize AutoCAD. It will typically save 20-30 minutes per new drawing.
2. It is the first step to automating standardization. If the correct layers and styles already exist whenever a new drawing is started then it rapidly becomes easier for the user to comply with the standard. In order to deviate they would have to deliberately go out of their way to create new, non-conforming items.



### **And on today's menu we have...**

Menu customizing takes a little more effort to learn and to do, but the benefits are enormous. No, you do not need to be a programmer. All you need is to be fairly conversant with AutoCAD commands. This may take a little research because most of us are so used to working from toolbars, pop-down menus, and dialogue boxes that we do not always realize the name of the underlying command that these items invoke.

Whenever you start AutoCAD have you ever noticed that little message flashing by that says, "Customization file loaded successfully" in AutoCAD 2006 and later, or "Menu loaded successfully" in earlier releases? Unlike almost any other program, AutoCAD's menu structures are not hard-coded into the program but instead are carried in an external file or files.

A full discussion of menu customizing procedures, syntax, and techniques is well beyond this course. What I will be showing you is how to use customizing effectively to help ensure compliance with CAD standards.

There are three kinds of menu customizing

1. The menu customizing files in AutoCAD 2005 and earlier releases are simple ASCII text files that can be edited with almost any text editor such as Notepad. This fact has two major benefits. First, foreign-language versions of AutoCAD become much easier to create, and second, it becomes possible for users to customize the menu structure to suit their needs.

As installed, the menu comes in one large file. It includes the pop-down menu bar, toolbars, mouse buttons, digitizer tablets, and image menus. You can edit this menu file, but it is usually better to create "partial menus" that load in and combine with the original.

Okay, I lied. There are actually as many as 6 menu files, but 3 of the 6 effectively hold the same thing.

- a. \*.MNU is a little-used "template" file that can be used to create a .MNS file.
  - b. \*.MNS is the "source" file. This is the one we usually edit.
  - c. \*.MNC and \*.MNR are compiled and resource files that AutoCAD creates automatically from the \*.MNS file.
  - d. \*.MNL is an optional AutoLISP file holding program routines used by menu macros.
  - e. \*.BMP files hold the toolbar button images.
2. AutoCAD 2006 revised the procedure completely. All interface customizing except tool palettes is now held in an Extensible Markup Language (XML) format. Although it is also a simple text file, you need to be a real programming guru (or geek) to edit it directly. Instead, AutoCAD provides an interface that lets you edit it directly from within AutoCAD. By an amazing coincidence, CUI is also the command that invokes it.
  3. AutoCAD 2004 introduced the little-used and little-understood tool palettes, which will figure centrally in our later discussions.

### **In the good old days...**

Let's look at the general format of an older-style menu file.

I mentioned a moment ago that the one file handles all of the menu sections. The first thing it needs is some way of telling the sections apart.



Easy. Each menu section starts with a name that begins with three asterisks. It must be one of the “legal” names, such as

```
***POPn where n = 1 to 500  
***TABLET  
***IMAGE  
***TOOLBARS
```

and so on.

Within each section the syntax is remarkably similar, so that once you understand the basic concepts you can easily apply them to other menu sections.

Consider the following menu excerpt that was cut and then abbreviated from the standard menu file:

```
***POP7  
[Draw]  
[Line]^C^C_line  
[Ray]^C^C_ray  
[Construction Line]^C^C_xline  
[Multiline]^C^C_mline  
[--]  
[Polyline]^C^C_pline  
[3D Polyline]^C^C_3dpoly  
[Polygon]^C^C_polygon  
[Rectangle]^C^C_rectang  
[--]
```

\*\*\*POP7 indicates that this will be the seventh pop-down menu, counting from the left end.

Anything enclosed in [square brackets] is a prompt. The first one, [Draw] will be the title that appears in the menu bar to show the name of this particular pop-down.

The lines following [Draw] all start with a suitable prompt followed by the desired AutoCAD commands. When you click on “Draw” in the menu bar it pops down a menu that displays the [prompt] portion of all the following lines until it hits the first blank line.

When you click on a menu item, AutoCAD picks up after the closing square bracket ]. It reads the rest of the line and feeds it into itself just as though you had typed it at the Command: prompt.

Hence, our first significant point in menu customizing. The macro following the [prompt] can contain anything that you can type in from the keyboard. This includes AutoCAD commands, responses to commands, and Lisp expressions (sorry, no Lisp in LT). It can be any length and can include more than one command. A single menu macro can therefore run a whole sequence of commands. The one thing you cannot do is to supply entries to dialogue boxes, so almost all dialogue-box commands also have a text-only command line equivalent. In some cases AutoCAD will recognize that the command is running from a menu macro and will automatically shift into command-line mode. In other cases, notably Layer and Insert, when used in a menu macro they must be preceded by a dash (i.e –layer, -insert).

Although AutoCAD 2006 and later uses a different mechanism for menu customizing, the syntax of the command macros launched by picking a menu item, a toolbar button, or tool palette icon remains identical to earlier releases.





Most macros start with four characters `^C^C`. AutoCAD interprets them as hitting `<Esc>` twice in order to cancel any command in progress before starting the desired one.

Now let's look at some of the customizing we can do to encourage compliance with CAD standards. For example, take a look at the fifth line. This obviously draws a "construction" line by invoking the `XLINE` command. What if you have decided that all construction lines must go on a separate layer named `CONST`? You could issue a printed edict to this effect and hope that everyone remembers to comply, or you can set it up to be automatic.

As mentioned earlier, the first step to automation is to have the layer already created in the prototype or template drawing.

The next step is to edit the menu file so the line in question looks exactly like this:

```
[Construction Line]^C^Cclayer const xline
```

Now let's dissect this and see what it does.

The prompt and `^C` have already been explained.

`CLAYER` is an AutoCAD command that sets the current layer to the supplied name. In our case it will set it to `CONST`.

The macro then goes on to start the `XLINE` command.

Hey, presto! With that one little bit of editing we have automated the standards compliance process. Every time a user picks (*Draw | Construction line*) they will automatically be flipped to the `CONST` layer before the `XLINE` command runs.

That covers the basic format and syntax of the menu files.

Now we are ready to do some actual menu work. The main menu file is quite large and has a lot of other things in it that we don't need to worry about right now so I would strongly suggest that you do not try to edit it at this time. Instead you should create a small menu file and use the "partial load" feature that I mentioned earlier to add its structure to the main menu that is already there. AutoCAD will remember this and next time you open AutoCAD it will automatically load the main menu plus your partial menu.

Once again, a reminder; this course is NOT intended to be a full-fledged course in menu/interface customizing, but instead is intended to show you some possibilities that will help with standards compliance.

Here is a sample menu addition that will create such a custom pop-down menu. It can be created as a simple text file using Notepad.



```
***MENUGROUP=MyCustom
***POP1
[Lines]
[Part outline]^C^Cclayer part line
[Hidden]^C^Cclayer hidden line
[Centre]^C^Cclayer centre line
[Section]^C^Cclayer section line
[Construction]^C^Cclayer const xline
// end of menu file
```

### Let's Get Loaded!

Now that you have created your partial menu it is time to load it.

#### NOTE

The following section is based on **AutoCAD 2005** and earlier. The **MENULOAD** command still works in 2006 and later, but it must be entered at the Command: prompt. In addition, the dialog box is a little different.

From the menu bar pick (*Tools | Customize | Menus...*). This invokes the **MENULOAD** command that pops up a dialogue box with the "Menu Groups" tab active. Click on the "File name" box and type in the name of your menu, such as TEST.MNS. Click on "Load". If you put it in the current working folder (or a folder that is in AutoCAD's support file search path) then it will be loaded. If not, you can always "Browse" and go looking for it.

Once it has loaded, note that the **\*\*\*MENUGROUP** name that you gave it in the file (in our example, "MyCustom") appears in the left-hand "Menu Groups" window.

**NOTE:** The following four paragraphs do not apply to AutoCAD 2006 and later. These releases automatically load all your pop-downs at the right-hand end of the menu bar.

Click on your menugroup in this window to highlight it, and then click on the "Menu Bar" tab.

The left window now shows a list of all the **\*\*\*POPn** pop-down sections that you defined in the file ("Lines") while the right window shows the ones that are currently displayed in the menu bar (the standard AutoCAD ones).

Click on one of your menus in the left window to highlight it, and then click on one of the existing menus in the right window. Click on "Insert" and your menu will be added to the right-hand list immediately above the one that you highlighted and hey, presto, your new menu items appear instantly in the menu bar! If you change your mind you can always highlight any menu and "Remove" it.

Click on "Close".

Click on your menu item and it pops down, exactly as defined in your menu file. Click on an item from the pop-down and the exact string of commands and responses that you put in the menu will be invoked.

Assuming your drawing already contains the layers as specified in the menu (In our example these would be PART, HIDDEN, CENTRE, SECTION and CONST) then picking a menu item will switch you to its layer and start the LINE (or XLINE) command.

Magic! No more lines on the wrong layer!



I think you can easily see where this is going. By repeating the sample macro with suitable changes you can easily create separate menu picks for things like text, dimensioning, crosshatching, and so on.

Standardization through customization!

AutoCAD will remember things and next time you open AutoCAD it will automatically load the main menu plus your partial menu exactly as you left it.

### **A Gnarly Bit**

If you edit your partial menu file and want to reload it during the same AutoCAD editing session you need to run MENULOAD again, “Unload” your menu group, “Load” the file again, and reload your menus back into the menu bar. The other method is to create two profiles, and then use the OPTIONS command to switch to the other and back.

### **More Syntax**

The documentation is all found in the “Help” system.

AutoCAD 2005 & earlier: To find it, click on (*Help | Developer help topics | Contents | Customization guide | Custom menus*). This expands out to tell you all there is to know about menu customizing.

AutoCAD 2006 & later: (*Help | Additional resources | Developer help | Customization guide | Customize the user interface*).

Here are a few significant points that will get you going.

#### **<Enter>**

You may recall that AutoCAD allows you to hit the space bar at any time that it needs an <Enter>. The <Enter> key, and the space bar are exactly equivalent. One of the reasons for this is to allow menu macro lines to include <Enter> within them. In a text editor or word processor, when you hit <Enter> it drops down to the next line but we do not want to do that within a menu macro and hence we can use a space instead.

Multiple spaces can be confusing to humans, so AutoCAD also allows a semicolon (;) to represent an <Enter> in macros. A menu macro runs exactly the same way if any or all of the semicolons were replaced with spaces and vice-versa.

On the other hand, note that every space in a macro is significant.

#### **\ (backslash)**

The backslash character in a macro means, “stop and wait for user input”. For example, a macro to insert a block might look like this:

```
[lamp]-insert lamp \;;
```

This will issue the -INSERT command, feed it the block name “lamp”, and then wait for the user to select an insertion point. The two semicolons tell it to take the default scaling factors.

#### **Control Codes**

AutoCAD has provision for invoking “control codes” within a macro. The most common usage is to start each macro with the six characters ^C^C^P. This tells AutoCAD to issue two <Esc> codes to cancel any command in process, and a <Ctrl-P> to stop the following commands from echoing to the Command: prompt area. This gives a faster-running, cleaner looking menu



operation. It is often handy to leave the ^P out initially while you are debugging and then put it in when you are finished.

## TOOLBARS

Toolbars can be created and edited directly from within AutoCAD. Once again, **2005** and earlier are different from **2006** and later. Let's start with one of the many ways in 2005.

### NOTE

Toolbar changes and creations instantly get written back to the appropriate \*.MNS file automatically. Make sure you keep a safe copy of the original before making any changes.

### Edit a button:

1. Right-click on any toolbar button.
2. Click on "Customize" from the menu that pops up. This brings up the Customize dialog box.
3. Click on any button in any toolbar. This switches the Customize dialog to the Button Properties tab. Its windows show the following, all of which can be edited:
  - a. *Name* is the tool tip that appears when the cursor pauses on a button.
  - b. *Description* is the help string that appears at the bottom of the screen.
  - c. *Macro...* is the command macro that will be fed to AutoCAD when this button is selected. It can be edited following exactly the same procedures and rules that we have previously discussed for other menu sections.
  - d. *Button Image* can be edited, or replaced with another. You may need to edit two different images if you check the Large buttons box under the Toolbars tab.

### Add/delete/move a toolbar button:

1. Right-click on any toolbar button.
2. Click on "Customize" from the menu that pops up. This brings up the Customize dialog box.
3. While the Customize dialog box is active you can:
  1. Delete a button from any toolbar by dragging it from the toolbar and dropping it in the open drawing space.
  2. Change the sequence of buttons in any bar by dragging and dropping along it.
  3. Move a button from any toolbar to any other bar by dragging and dropping.
  4. To add a button, activate the Commands tab. Select a command category, then drag and drop a suitable button from the Commands window onto the desired toolbar.

### Create a new toolbar:

1. Right-click on any toolbar button.
2. Click on "Customize" from the menu that pops up. This brings up the Customize dialog box.
3. Select the Toolbars tab.
4. You will be asked for a name, and you can select the desired menu group name.



#### NOTE

As previously noted, Toolbar customizing automatically (and instantly) gets written back to the .MNS menu file. If you have previously loaded a partial menu, then your Menu Group name will appear in the window. If you select your group, the new toolbar will be added to your \*.MNS file. By doing this, you can easily copy it to other computers or to a network server, and you will avoid messing up the standard menu files.

5. Select the New button to create a new toolbar.
6. New buttons can be created, or existing ones moved, as previously described.

#### Tips, Trips, Traps, & Tripe

As suggested earlier it is usually easier to start with a “partial menu”. When you get a section working properly you can always use the MENULOAD command to unload any of the standard pop-downs and replace them with yours.

If your macro runs out before the command finishes, that is if you do not supply enough input in the macro to have the command finish, then it goes as far as it can and then drops back to letting the human provide any remaining input. Thus the simple command LINE with no further input from the macro will drop through to asking “From point:” and then repeatedly “to point” until you are finished.

- The CLAYER command is used to change the current layer.
- The TEXTSTYLE command sets the current text style.
- The TEXTSIZE command sets the default height for text. This is overridden for any text style definition that has a height greater than zero.
- The DIMSTYLE command sets the current dimension style.
- Make sure you know how each command works and what replies it needs before putting it in a macro. I usually step through it manually at the keyboard making notes as I go.
- A menu macro can contain ANYTHING you can type in at the Command: prompt. This can include Lisp and Diesel programming expressions (except in LT).
- A helpful addition to the TEXT commands is to include macro replies for text height and rotation angle. This will save the operator many keystrokes and will eliminate deviations from the standard
- If you are going to edit any existing menu file make sure to keep a safe backup copy first in case when you mess up and need to go back.
- If you are going to edit any existing menu file make sure to keep a safe backup copy first in case when you mess up and need to go back. This is important enough that it was worth repeating.

In releases prior to 2004, the standard “Object Properties” toolbar includes the “color”, “linetype”, “line weight, and “plot style” controls. They are a CAD manager’s worst nightmare



and a veritable creation of the Devil. To get rid of them, edit the standard ACAD.MNS file. Find the

\*\*TB\_OBJECT\_PROPERTIES section and delete the following:

```
[_Control(_Color)]
  [--]
                                     [_Control(_Linetype)]
  [--]
  [_Control(_Lineweight)]
  [--]
  [_Control(_PlotStyle)]
```

Thankfully, AutoCAD 2004 made this a separate toolbar called PROPERTIES. It can easily be deleted using the toolbar editing facilities described earlier.

The standard menu/CUI files for regular AutoCAD are called ACAD.MNS or ACAD.CUI, depending on your release, but variants such as LT, Mechanical/Architectural Desktop, and so on use others. The command MENAME will show you the full path and file name of the current base menu.

The standard AutoCAD menu files also have matching \*.MNL files. These contain a bunch of Lisp routines that are called from within the menu. If you copy or rename an .MNS file be sure you get the matching .MNL file as well.

In earlier releases, when moving or copying a custom menu you need at least the .MNS and possibly .BMP and .MNL files, and/or you need the .MNC, .MNR, and .MNL files. AutoCAD 2006 and later use .CUI files.

How does AutoCAD find a menu file? You need to understand where AutoCAD goes searching for "support" files. This includes menus, fonts, AutoLISP programs, drawings to –INSERT, hatch patterns, and so on.

1. it starts by looking in the "current" folder, as defined in the "Start In" window in the desktop icon that launched AutoCAD
2. next, it looks in the folder containing the current drawing file,
3. then, it searches through any folders specified by the "Support files" setting under "Options", working from the top down,
4. Finally, it looks in the folder where ACAD.EXE lives.

For our purposes, the third item is the most significant. It is usually not good practice to put support files in the current folder or in the drawing folder, because if you later work from a different folder (say for a different project) then AutoCAD will be unable to find them. The best place to put them is in a special custom folder, such as C:\CUSTOM for example. If you are on a network, it can be a common network folder to which all authorized users have "read" rights.

Now all you have to do is tell AutoCAD where to find it.

Once again, invoke the OPTIONS command. When the dialogue box comes up click on the *{Files}* tab then double-click on *{Support file search path}*. This will display the current search path that AutoCAD is using.



Click on the *{Add}* button to insert a new item, and then *Browse* for the path to your custom folder.

Whenever AutoCAD is looking for any support file it checks the folders in the order in which they are listed here. You should select your new path and click on *{Move up}* enough times to put it at the head of the list so AutoCAD will look there first. That way if you happen to use a file name that is the same as a standard AutoCAD support file then it will find your custom one first and will ignore the standard one. Click on *{OK}*.

From now on, when you use the *MENULOAD* command you only need to enter the menu file name. AutoCAD will automatically find it in your search path.

AutoCAD LT menus can be customized in exactly the same manner, but they do not support Lisp or Diesel.

### AutoCAD 2006; CUI is for yui...

As mentioned earlier, AutoCAD 2006 introduced a whole new customizing environment, called the *Customize User Interface*, or *CUI*.

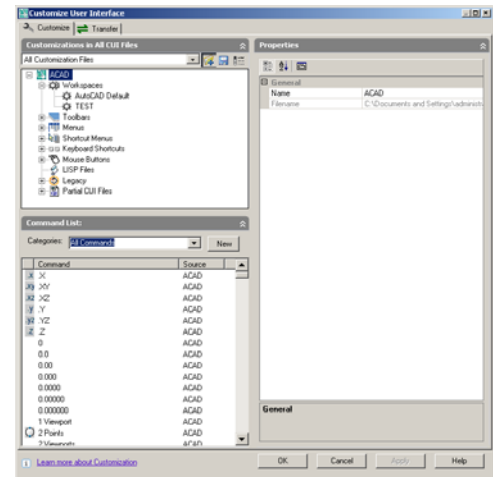
All interface customizing except tool palettes is now held in a single file. No more .MNS, .MNU, etc etc. The *ACAD.CUI* file is in *Extensible Markup Language (XML)* format. Although it is also a simple text file, you need to be a real programming guru to edit it directly. Instead, AutoCAD provides an interface that lets you edit it from within AutoCAD. By an amazing coincidence, *CUI* is also the command that invokes it.

Like earlier menu customizing, *CUI* is a major topic in its own right and the subject of several other classes. Once again, we will just hit the high spots to wet your appetite.

Before you do anything else, make a safe copy of the standard *ACAD.cui* file.

Start the *CUI* command, either by entering it in at the *Command: prompt* or by clicking (*Tools | Customize | Interface*). You can also right-click within any toolbar and then click on *Customize...* In any case, up comes the dialog box of *Figure 1*.

The upper left section shows the types of menus, the lower left lists AutoCAD's commands, and the right side is where actual customizing takes place.



**Figure 1: The basic CUI dialogue box**

Paralleling our earlier *XLINE* example, we need to go to the upper left section and:

1. Click on the + sign beside *Menus* to expand it;
2. Click on the + beside *Draw* to expand it;
3. Click on *Construction Line*.

The right-hand window will now look like

Scroll down the command list and find the *Line* command. Drag it into the upper left window and drop it in the *Draw* menu section just below the existing line command entry. You will now have two *Line* commands in this menu.

**Figure 2: The Construction Line menu item**

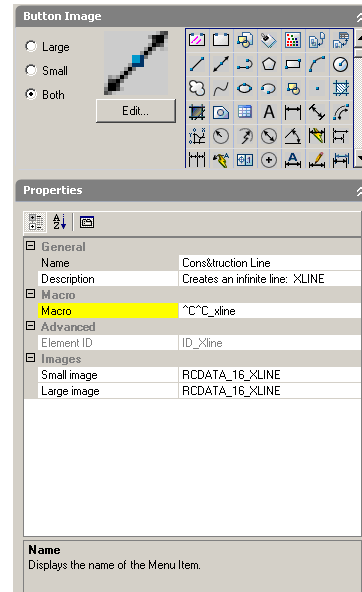
1. Click on the new entry to bring up its edit window in the right section of the dialog box. Edit it as follows:

. In particular, note the Macro line. Hey, it looks almost exactly like the example from the earlier releases! Right! We can edit the macro in exactly the same way as previously. We can add the CLayer command and layer name, the backslash will pause for user input, and so on.

Now we'll go back to Figure 1 and click on the Draw menu name.

We want to add new items for our hidden lines, object lines, and so on.

2. Click in the command window in the lower left corner of the dialog box.
3. Click the down arrow at the end of the Categories window, and then click on Draw from the scroll list. This filters our command list down to a manageable size.
4. Scroll down the command list and find the Line command. Drag it into the upper left window and drop it in the Draw menu section just below the existing line command entry. You will now have two Line commands in this menu.



**Figure 2: The Construction Line menu item**

5. Click on the new entry to bring up its edit window in the right section of the dialog box. Edit it as follows:
  - a. Change its name to *Hidden Line*
  - b. Edit it's macro to add CLayer Const, just as we did in the earlier release.
  - c. Edit its description.
  - d. If you want to, edit the image bitmap to show the line in the appropriate colour and line type.
6. Repeat for the other line types and layers.

Obviously, we can continue doing this for text, dimensioning, and so on. A significant point to notice is that nothing actually happens to AutoCAD until you click on Apply. If you Cancel before that then your edits are ignored.

Toolbars can also be edited and created in much the same fashion. The CUI command will also allow us to create partial CUI files, just as we did with .mns files. We can also bring custom files forward from earlier releases, but if you are using AutoCAD 2006 then you MUST have Service





Pack 1 installed for this to work properly. This latter facility allows us to cheat a bit; if we want to we can continue working the old way and then simply import into the CUI.

A significant point to notice here is that I lied a bit when I said all interface definitions are held in a single CUI file. As with the earlier releases, we can also have partial CUI files that add to the main one. To help with this, AutoCAD thoughtfully provides a blank CUSTOM file for our nefarious purposes.

For more information on the CUI, check out Lynn Allen’s “Circles and Lines” articles at [www.cadalyst.com](http://www.cadalyst.com).

### Pleasing To The Palette

Tool palettes are probably one of the most powerful and yet least-used weapons in our fight for standards compliance. I’ll outline the reasons for this in a moment.

If your tool palette window is not visible, invoke the *ToolPalettes* command (*Tools | Tool Palette Window*) in earlier releases, or (*Tools | Palettes | Tool Palette Window*) in AutoCAD 2007.

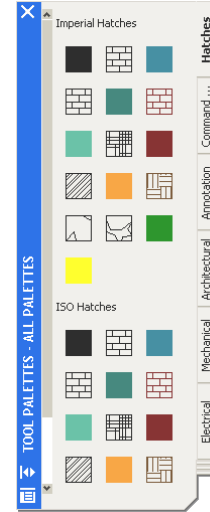
Pressing the Ctrl+3 key combination, entering the keyboard shortcut TP, or



clicking the *Tool Palettes* button on the standard toolbar will also launch it.

Yours may not look exactly like Figure 3 because of the aforementioned differences between releases. On the other hand, if you click on the Hatches tab you should get pretty close to Figure 3, no matter which release you are using.

Let’s sort out a bit of basic terminology. First, the entire lump is called the *tool palette window*, where individual tabs each bring up a separate *tool palette*. Each palette contains one or more *tools* (icons) icons.



**Figure 3: The Hatches tool palette**

Okay, draw a simple figure such as a circle, a rectangle, or three lines that enclose an area. Now we want to apply a hatch pattern to it. We could do it the hard way, which is to select (*Draw | Hatch*) and then work our way through the dialog box to select a pattern, then close the dialog box so we can select a definition point for the hatch, and finally return to the dialog box to accept everything.

On the other hand, we can do it the easy way by using the tool palette. Simply click on a hatch pattern in the palette, and then click within a bounded area. The hatching is applied to the selected area. Two clicks and we are done!

### Now You See It...

The good news is that we can leave the tool palette window open all the time. I know, it would seem that it is going to take up a lot of real estate in your drawing screen, but here is a trick we can use.

Notice the two icons at the bottom of the vertical blue bar along the edge of the window. The second one up from the bottom toggles Auto-Hide. When it is on, the window reduces to showing just the blue bar. If we hover the cursor over the bar then the window scrolls out and



we can select from it. When we move the cursor into the drawing area then the window retreats back into the blue bar, whether we have made a selection or not.

### Pick me, pick me!

We have seen how quick and easy it is to use tool palettes, and how to set up their visibility. The reason they are so seldom used, however, is because of the limited selection they contain. Worse yet, they change so radically from release to release. The answer is that Autodesk never intended that they would come pre-configured as a standard feature. Instead, their intention has always been that individual users are expected to customize them, and the shipping version is just a sample that tends to highlight other features in the new release.

Take a close look at the Hatches palette again. Three of them all contain the same icon, and have the same Brick label, but the third one is a dark reddish-brown color whereas the other two are black.

If you use each icon to hatch a different area, one will produce a normal Brick hatch pattern, one will produce it rotated 90 degrees, and the third will produce a dark reddish-brown pattern.

Yes! A variety of properties can be pre-defined for our hatch patterns, and each combination can be assigned its own palette tool. This can go a long way towards improving productivity and maintaining CAD standards.

### Counting Your Hatches...

Right-click in any open area of a palette, and then select New Palette from the context menu. It will default to being named New Palette, so let's rename it to My Hatches.

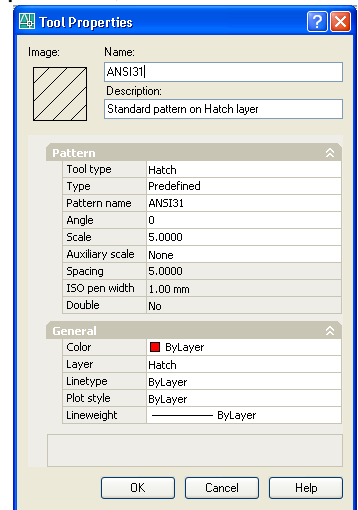
Now draw several circles, and create a layer called Hatch with a suitable color. Make the Hatch layer current.

Hatch each circle using the standard Hatch command to produce a variety of patterns, scale factors, and rotation angles.

Now comes the magic. Click on an existing hatching, pause, then click and hold to drag it onto the tool palette. Note that you need to pause between clicks; if you are too fast, then AutoCAD interprets it as a double click and invokes hatch editing.

A new tool icon is created for each hatching that you drag onto the palette, and each tool has the layer, colour, pattern, scale, and rotation angle of its prototype.

If you change your mind, you can always edit things. Simply right-click on any tool palette icon and then select Properties. This brings up a dialog box that suits the type of object you are changing. Figure 4 shows the Tool Properties dialog for a Hatch tool. As you can see, you can edit virtually everything about the tool.



**Figure 4: This dialog box edits the properties of a Hatch tool**

Ah, but it gets even better! Start a new blank drawing and observe how the tool palette includes your new tool icons.



Create a rectangle, and then hatch it using one of your new hatch tools. A quick check will reveal that the new hatching is on the correct layer, and the correct layer has the correct color and other specifications, even though the layer did not already exist in your new drawing! AutoCAD created it on the fly to match the palette tool specifications.

If you think that was good, then consider this. AutoCAD 2005 added a significant new functionality to tool palettes, called “tools by example”. Let’s see how they work.

Create a new layer called Hidden. Apply the Hidden line type, a color of red, and a line weight of 0.004”. Draw a circle on this layer.

Now, click on the circle so it highlights, pause, then click on it again and drag it onto a tool palette. Remember, don’t do this too quickly or you will launch the Properties editor instead. The Circle command icon will magically appear on your tool palette.

Clicking this new icon will start the Circle command, and will place the circle on the Hidden layer regardless of the current layer setting. Once again, it will create the layer to our specifications if it does not already exist in the drawing.

### **Flyout By The Seat Of Our Tools...**

Take a close look at the Circle icon on the tool palette. Notice a small “right arrow” triangle next to the Circle icon? Click on it, and a flyout will reveal icons for the eight fundamental “draw” commands! Better yet, if you click on one of them, such as Ellipse, then the ellipse that you create will automatically be on the Hidden layer.

The last-use tool becomes the current one shown on the palette

It doesn’t matter which of the “draw” commands you use initially to create the Draw flyout tool. The flyout will contain all the draw tools, and they will all have the same properties. Editing the properties of any icon in the flyout will apply the changes to all icons.

Do you want to automate compliance with CAD standards? Simply create flyout icons for drawing on the Object layer, the Hidden layer, the Center layer, the Construction layer, and so on. Just click on the icon from the correct group and everything lands where it should, and layers will be created on the fly if they don’t exist.

Tools by Example also works for dimensioning and text. The tools will have the layer and style as the prototype. A set of tools that replicate your standard text and dimension styles can easily be created. Once again, these styles and layers will be created in the target drawing if they don’t already exist.

On the other hand, commands such as Boundary, Polygon, and Donut actually just create polylines, so any tool palette icons you create from them will simply create the Pline icon and the standard Draw flyout.

### **What’s In a Name?**

Now here comes one of the crunchy bits. Having created tool flyouts for your various layers, you realize they all have the same name. How can you tell which is which?

Simply right-click on an icon, then pick Rename from the context menu. You can now rename the Circle icon to be the Hidden icon.

All the icons in this flyout are now called Hidden even if they create lines, arcs, splines, and so on. Each set of tools can be renamed to reflect the layer on which it resides.



Ah, but what if you have dozens or perhaps even hundreds of layers, such as in the typical architectural drawing?

No problem. Having created a tool set you can right-click it and then Copy-Paste to make one or more duplicates of it. Now right-click a tool in each set and then adjust its properties to the appropriate layer. As the number of tool sets grows, AutoCAD 2006 and later allows you to add text items and separator lines. Simply right-click in a suitable open area of the tool palette, then select Text or Separator from the context menu. You can thus group tool sets by floor, or by discipline, or whatever. If things get even further out of hand you can always right-click and select New Palette to create a new palette for the various layer groups.

You can also drag-and-drop to move tools or tool sets up and down within a palette, and you can copy from one palette and paste it into a different one.

### **A Tool Off The Old Block...**

Block insertion tools are also extremely easy to create. Simply click once on an existing block insertion to highlight it, pause, and then click and drag it onto a tool palette. Once again, don't forget to pause; if you click too soon, AutoCAD interprets this as a double-click and opens the Block Editor instead.

The exact same process works with external references (xrefs) and raster images. Simply select them, pause, then select and drag onto a palette.

You can also access block definitions in other drawings by using the Design Center, and then dragging and dropping them into your tool palette. In fact, you can automatically create tools for all the blocks in a drawing in one hit, or for all the drawings in a folder and its sub-folder.

As with previous tool types, you can right-click the icon for a block, an xref or a raster image and adjust its properties. These properties include rotation angle, scale factor, layer, color and more. AutoCAD will honor these properties when you insert one of these objects from its tool. In particular, note that you can create several tools from the same object and then adjust their properties individually, just as you did for hatch patterns.

### **The Crunchy Bit**

There is one slightly crunchy bit with the tools for these three object types, however. Unlike the layer definitions associated with hatches, lines, circles, etc., the block, xref and raster images are not saved within the tool palette file. All it contains are reference links to the original source file, which must exist whenever you want to use one of these tools. This can cause problems when you want to share your palettes with other people, or if you move the source files to different locations.

Fortunately, there is a quick workaround. If a tool cannot find its source file in the original location, it will start searching down through AutoCAD's support file search path. Start the Options command (Tools | Options) and select Files. Click on the Support Files Search Path, click on Add and then Browse to the new file location.

### **But Wait! There's More!**

A palette tool is not limited to simply executing a single command. If you modify the properties of a palette tool by right-clicking on it, you can enter a macro string of several commands, or even an AutoLISP expression, to be evaluated when the tool is selected. For example, you might want a block insertion tool to execute a Zoom All before it invokes the Insert command. The syntax of command macros has already been discussed.



## Kindergarten 101

What is one of the first things you learned in kindergarten? Always share your toys! Having built the ultimate tool palette set, it is nice to share it with your coworkers. And it's not just nice a nice thing to do, sharing can go a long way towards ensuring compliance with your CAD standards.

Let's start with the case where your office is on a network; later, you'll see how to send a copy to someone who is not on your network.

The first step in sharing is to find the toys you want to share. Following Microsoft-recommended practice, user-customizable files are found in the folder

```
C: \Documents and Settings\\Local Settings\  
  Application Data\Autodesk\AutoCAD200n\  
    R1n.0\enu\Support\Tool Palette
```

(where *<user name>* is your login name and *n* depends on your AutoCAD release). The really bizarre thing is that by default, this folder is hidden. If it doesn't show in Windows Explorer, then you need to click on Tools / Folder Options in Explorer, then click the View tab, then turn on Show Hidden Files and Folders.

Anyway, a file and a subfolder are in the Tool Palette folder. This is where AutoCAD stores all the tool palette information.

Sharing is a simple two-step process. First, copy the Tool Palette folder to a common location on the network.

Next, start the Options command (*Tools | Options*), and then select the Files tab. Expand *Tool Palettes File Locations*, then Browse to your new location on the network.

Okay, I lied -- you will probably need a third step. As mentioned earlier, if your tool palettes include block insertions, then you need to make the support drawings available. Once again, you can set the support file search path to a network location for them. The cunning bit here is that it does not have to be a unique location. For simplicity, you can use the same location as the tool palette files.

Once this is set up, all you need to do is to have each co-worker set their options the same way, and now everyone can use the same palettes. Any changes made to the network copies will update each machine the next time AutoCAD is started.

As things stand so far, any changes that anyone makes to the tool palettes will reflect back to everyone. You will probably want to go to your network files and use the Properties option of Windows Explorer to set them as *Read Only*. Obviously this is not bullet-proof, because anyone can set the properties back, but at least it will minimize accidental changes. You can also set network file rights to Read Only for the entire folder for general users.

## One on One

As threatened, I will now cover the case of sharing a palette with someone who is not on your network. All you need to do is to click on (*Tools | Customize | Tool Palettes*). Now right-click on the desired palette, and then click on *Export* to create a suitable export file.

When you send the palette file to someone else, make sure you also include any required block definition files.



At the other end, the recipient simply follows the same process, except they click on *Import*, then browse to the file. They also need to copy the block definition files to a suitable location.

### **The Clever Bits**

First, note that the tool palette and support file search paths are unique to each AutoCAD profile. Thus you can set up different tool palettes for different clients, different disciplines, etc.

Next, note the fine print. When using Options to set the tool palette file location, the prompt is plural. Thus you can have multiple paths to multiple tool palettes. If you do this, then AutoCAD builds one large tool palette window incorporating all your tool palettes, in the order in which the search paths are listed.

### **The Slightly Clever Bits**

The Help facility states that tool palette files are not compatible between releases. This is partly true, in that I have had no problem so far in using earlier files with a later release. The converse, however, is an interesting case. The palettes appear to open properly, but some chunks are missing, and other sections don't function as they should.

Transferring a tool palette to AutoCAD LT can also have a few minor issues due to some of the differences in capabilities of LT.

All in all, tool palettes are a simple, powerful customizing tool.

### **And now for something completely different**

You should also learn how to attach "attributes" to blocks. These allow you to create "fill in the blank" text objects that attach to block definitions. When your menu macro -INSERTs your title block then the operator will be prompted for things like name, drawing title, date, etc which will automatically be filled in at the correct locations and with the correct text styles and sizes.

The Field objects, added in AutoCAD 2005, can also be very useful in assuring standards compliance. Fields can be used to do such things as have a title block that automatically fills in the date, file name, creator's name, and so on in specified text styles and sizes.

### **Have fun!**

A bit of customizing will make your management of AutoCAD far easier. My staff used the layer, text style, and dimension style commands so rarely that I removed them from the menus and they forgot how to use them. Everything they needed was preset and automatic. Compliance with standards becomes 100% when compliance is easier than non-compliance. Resistance is futile...

### **And Don't Forget...**

*[www.autodesk.com/auconnect](http://www.autodesk.com/auconnect)*

will connect you to Autodesk University content files. This includes course handouts, sample files, datasets, and the a/v files for the presentations that were recorded live.