



Walt Disney World Swan and Dolphin Resort
Orlando, Florida

Automated CAD Standards

Bill Fane - BCIT / CADALYST

CP12-2 Do you have trouble enforcing CAD standards? You don't need a published manual and a police force. Users are more likely to comply with standards if it is easier to comply than to deviate. Learn how most of your CAD standards can be built into your template drawings and menus. You will learn how to do simple menu customization so that instead of just having a Line command, you can have menu picks for Part Outline, Hidden Line, Center Line, and so on. Whenever users pick one, they will automatically be set to the correct layer. The new Custom User Interface functionality in AutoCAD 2006 makes it very easy to do this. We will also touch on the "old way" used in earlier releases, and on how the Standards Checker can assist in ensuring compliance.

About the Speaker:

An AutoCAD software user since 1986, Bill was a product engineer and manager for Weiser Lock in Vancouver, Canada for 27 years. Bill has taught AutoCAD and mechanical design at the British Columbia Institute of Technology since 1996 and teaches Autodesk Inventor at the Institute's Training Center. He has lectured on a wide range of subjects at Autodesk University since 1995. An active member of the Vancouver AutoCAD Users Society, he has written "The Learning Curve" column for CADalyst magazine since 1986, and writes about Autodesk Mechanical Desktop and Autodesk Inventor for Autodesk's Point A Toplines. He also writes for Inside AutoCAD Journal and Design Product News.

NOTE

This course is intended for advanced beginners through to early intermediate users. It is based on AutoCAD 2006, which 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 implements them. We will cover both versions.

“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.

Lack of adherence to standards is one of the biggest productivity-killers in AutoCAD. Imagine an office with several engineers, designers, and drafters. Over the years they have created several thousand drawings. Each one is set up differently, with different layer names and colours, different text styles, and so on. Staff turnover has compounded the problem as each user has developed their own style.

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.

In the Beginning...

The best place to start is with the “template” drawing.

The last four or five 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
- 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} then [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. You can type it in at the Command: prompt or you can pick {Tools} then {Options...} from the menu bar. Earlier releases use the PREFERENCES command, and current releases will accept either because Microsoft changed the "standard". How many Microsoft employees....

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>\Template

while 2004 changed it to something like this:

C:\Documents and Settings\<>user name>\Local Settings\

Application Data\Autodesk\AutoCAD 2004\R16.0\enu\Template

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.

Anyway, 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.

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).

You can also go back to your Windows Desktop. Right-click on your AutoCAD icon, then click on "Properties" then the "Shortcut" tab. At the end of the "Target" window, add

/t "<template>"

where <template> is the name of a template file. Now when you start AutoCAD it will automatically start a new drawing using the indicated template. You can easily create desktop icons that launch AutoCAD with suitable templates for specific projects, or clients, or disciplines, or whatever.

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 .../template 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!

The good news is that later releases default to “use a template” and it takes a bit of digging to get it to start a truly blank drawing.

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:

1. Edit acad.dwt to incorporate your standards, or
2. Select Tools > Options. Select the Files tab, then expand the Template Settings item. Expand the Default Template File Name... item, then browse to your desired template. This then becomes the default whenever AutoCAD starts up.

Don't forget, different profiles can have different file name and location options. You can also have a series of desktop icons launch AutoCAD under different profiles. As with templates, at the end of the Target window simply add

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

where <profile name> is the name of the desired profile.

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, 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.

This is where the biggest difference comes in between customizing AutoCAD 2006 versus the earlier releases. We'll start with the earlier releases first, for three reasons:

1. There are still a lot of people using earlier releases
2. The basic principles are the same for all releases
3. AutoCAD 2006 is able to import and migrate customizing files from earlier releases.

In the good old days...

The menu customizing files in earlier releases are simple ASCII text files that can be edited with almost any text editor or word processor. 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.

Let’s look at the general format of a 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.

Within a menu section many sections allow sub-menus. These each start with two asterisks followed by any name that you choose. You can thus create ***IMAGE sub-menus in order to have different image menus for **PARTS or **GEARS or **STAIRS or whatever.

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

```
***POP7
**DRAW
ID_MnDraw      [&Draw]
ID_Line        [&Line]^C^C_line
ID_Ray         [&Ray]^C^C_ray
ID_Xline       [Cons&truction Line]^C^C_xline
ID_Mline       [&Multiline]^C^C_mline
               [--]
ID_Pline       [&Polyline]^C^C_pline
ID_3dpoly      [&3D Polyline]^C^C_3dpoly
ID_Polygon     [Pol&ygon]^C^C_polygon
ID_Rectang     [Rectan&gle]^C^C_rectang
               [--]
```

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

**DRAW tells us that this is the “Draw” sub-section. Pop-down menus usually only have one section, so this is optional.

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 ampersand (&) indicates that the following letter, in this case a D, will appear underlined when you press the <Alt> key to indicate that this item can be activated by picking it with the mouse, or by hitting the

<Alt-D> key combination.

The menus may also have some optional stuff to the left of the square brackets. These are pointers to a ***HELPSTRINGS section that generates those little “help” notes at the bottom edge of the screen.

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).

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 sixth 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} then {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.

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
```

Now let's take a quick look at each line and see what it does.

```
***MENUGROUP=MyCustom
```

Each partial menu needs a unique name. You will see this later when you load it in.

```
***POP1
```

If this menu file were loaded by itself then it would be the first item on the left end of the menu bar. When you do a partial load into an existing menu you will be able to specify the exact position. The only requirement is that each ***POPn section in a given menu group must have a unique number from 1 to 499.

```
[Lines]
```

Remember? Anything within [square brackets] is a prompt. The first one appears in the menu bar itself.

From the previous discussion I think you can easily see what the next five lines do.

Finally, at the end of the ***POP1 menu section note that earlier releases require one blank line before the start of the next menu section. Conversely, there cannot be any blank lines within the pop-down section or it will stop popping down at that point. If you do want a blank line in a pop-down, use a blank prompt []. The [--] combination produces a separator bar.

In menu files one or more forward slashes (//) indicate a remark that AutoCAD will ignore. I stuck one in here just to make it obvious that there is a blank line at the end of the menu section.

When you Save the menu file in Notepad you must give it a file name that includes the extension *.MNS such as TEST.MNS. Note that some versions of Notepad will also tack a .TXT extension on the end unless you save as the file type "All files". For now you should save it into AutoCAD's current working folder (Hint: in AutoCAD, select {File} then {Save As...} to see the current folder).

Let's Get Loaded!

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

From the menu bar pick {Tools} then {Customize} then {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. Click on it 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, 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. To find it, click on "Help" in the menu bar then "Developer help topics" in the pop-down. When the dialogue box appears, click on the "Contents" tab, then "Customization guide" then "Custom menus". This expands out to tell you all there is to know about menu customizing.

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, the space bar, and the right mouse button are all 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.

There are two complications to this, however. One is that not all text editors will allow spaces at the end of a line, and the other is that multiple spaces can be confusing to humans. To solve both of these problems AutoCAD also allows a semicolon (;) to represent an <Enter> in menu macros. My menu macro would run exactly the same way if I were to replace any or all of the semicolons

with spaces and vice-versa.

On the other hand, note that every space after the closing prompt bracket] is significant. Spaces and tabs ahead of the prompt opening are ignored.

\ (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.

+ (plus sign)

Each menu macro can be as long as you want and can contain as many commands and responses as you need. When they get very long, however, there can be two problems; first, they get very difficult for a human to read, and earlier versions under DOS only allow 255 characters per line. Normally AutoCAD reads each line of text as a separate macro. A plus sign at the end of a line tells AutoCAD to carry on and add the next line to the end of the current one. This can be repeated as often as necessary, so a single macro could be several lines long.

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.

*****AUX1 and ***BUTTONS1**

These sections define what the mouse buttons and digitizer puck buttons do. The left button is always <pick>.

NOTE: The first line after the menu section label ***AUX1 or ***BUTTONS1 is used only when the SHORTCUTMENU system variable is set to 0. If SHORTCUTMENU is set to a value other than 0, the built-in menu is used. Similarly, the second line after the ***AUX1 or ***BUTTONS1 label is used only when the MBUTTONPAN system variable is set to 0.

Now here are two items that apply specifically to pop-down menus.

*****POPO**

This is the menu section that defines what happens in the mouse pop-up menu that appears when you press <shift+right button>.

Cascading Menus

If a prompt begins with the two characters -> then AutoCAD will not display the next item right away. In fact, it will not display any more choices until it gets to the line following a prompt that begins with <-. The lines in between are a “cascading menu” that only pop out when the first one is picked. For example,

```
***POP2
[Hello]
[->Lines]
[Part]
```

```

[hidden]
[centre]
[<-section]
[Circles]
:

```

When you click on "Hello" in the menu bar the pop-down will only show "Lines" and "Circles" until you click on "Lines" at which time the four in between will appear. You can also nest cascades, so that picking a cascaded item produces a deeper level of cascading.

***IMAGE

You can even create your own image or icon menus. The syntax is virtually identical to everything you have seen so far except that the [prompt] contains the name of an image file to display in the menu tile (hey, it's still a "prompt!"). On the other hand, tool palettes have become a faster and easier method of producing much the same result.

***TOOLBARS

Toolbars can be created and edited by using a text editor and working on the .MNS file, but there is a much easier way directly from within AutoCAD. In fact, there are even several different ways of doing this from within AutoCAD so I'll just walk you through one of them.

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:
 - Name is the tool tip that appears when the cursor pauses on a button.
 - Description is the help string that appears at the bottom of the screen.
 - 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.
 - 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:

4. Right-click on any toolbar button.
5. Click on "Customize" from the menu that pops up. This brings up the Customize dialog box.
6. While the Customize dialog box is active you can:
 - Delete a button from any toolbar by dragging it from the toolbar and dropping it in the open drawing space.
 - Change the sequence of buttons in any bar by dragging and dropping along it.

- Move a button from any toolbar to any other bar by dragging and dropping.
- 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.
- You can create a new toolbar by dragging a button from the Commands window into open drawing space, but there is a better way.

Create a new toolbar:

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

NOTE

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.

11. Select the New button to create a new toolbar.

Tool Palettes

The new tool palettes in 2004 make it very simple to set up custom block insertion and hatch pattern applicators. A 10-second demo can show how easy it is to use them, and to add or delete items from them. For our purposes, there is one more significant operation. Right-click on any tool palette item, then click on Properties. You can adjust the layer, colour, scale, rotation angle, and so on for any tool palette item. Once again, we can customize to ensure that objects will be inserted on the correct standard layer. In 2005, tool palettes can also include Lisp expressions.

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.
- Notepad is usually easiest and quickest for editing menu files. You can use any word processor, just make sure that you "Save as" and use the file type "text only (TXT)". Do not use the word processor's native format. You will give it a name such as CUSTOM.MNS but many word processors will then add another .TXT on to the end of that. You will have to rename it before AutoCAD will load it.
- 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 for regular AutoCAD is called ACAD.MNS, but variants such as LT, Mechanical/Architectural Desktop, and so on use others. The command MENUENAME 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.
- Other file names you will encounter:
 - *.MNS is the "source" text file that you create with your text editor, and to which AutoCAD writes current toolbar settings.
 - AutoCAD then automatically compiles this into *.MNC and *.MNR files that it needs. If you delete these they will be reconstructed next time the menu loads.
 - *.BMP files are the bitmaps for toolbar icons. They must be present if the *.MNR file is being re-created.
 - *.MNU is the menu "template" file. In earlier DOS versions it was the one we edited. If you re-load it then any toolbar customizing will be lost.
- 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.

- How does AutoCAD find a menu file? You need to understand where AutoCAD goes searching for "support" files. This includes menus, fonts, AutoLISP programs, hatch patterns, and so on as well as drawings to -INSERT.
 - it starts by looking in the "current" folder, as defined in the "Start In" window in the desktop icon that launched AutoCAD
 - next, it looks in the folder containing the current drawing file,
 - then, it searches through any folders specified by the "Support files" setting under "Options",
 - 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, 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 directly 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.

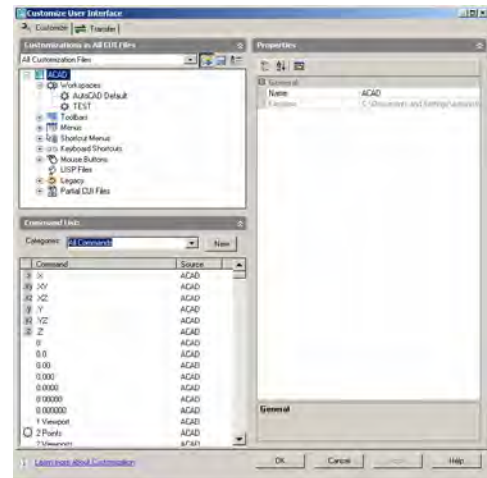


Figure 1: The basic CUI dialogue box

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 Figure 2. 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.

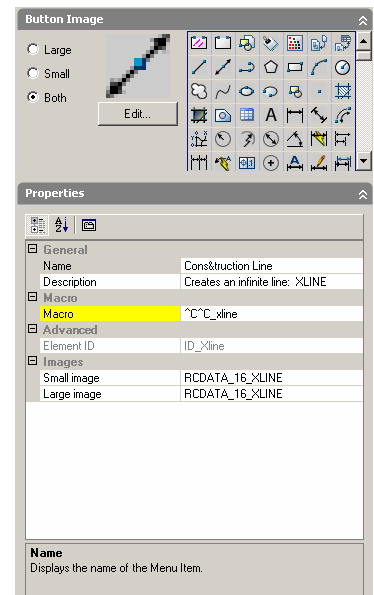


Figure 2: The Construction Line menu item

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 soon.

1. Click in the command window in the lower left corner of the dialog box.
2. Click the down arrow at the end of the Categories window, then click on Draw from the scroll list. This filters our command list down to a manageable size.
3. 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.
4. 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.
5. 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 we MUST have AutoCAD 2006 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 imply import into the CUI.

For more information on the CUI, check out Lynn Allen's "Circles and Lines" articles at www.cadalyst.com.

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 have covered in this course offer a pre-emptive strike to help avoid non-compliance situations before they even arise.

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.

