



Autodesk
University
2007

MNU, MNS, MNR, MNC, MNX, MNL ... or CUI

Speaker: Chris Lindner – WD Partners

GD105-2P This course guides anyone familiar the two-decade-old method of customizing AutoCAD's menus into the uncharted territory of the new CUIs. It outlines the procedures for creating, sharing and editing toolbars, menus, dashboards, shortcuts, mouse buttons, workspaces, and even touches on how to migrate legacy technologies (table, screen and image menus.. remember those?) safely and quickly.

After attending this class, you will be able to:

- Use previous menu knowledge to accelerate the CUI learning curve
- Create CUI files from existing MNU/MNS files
- Set up Enterprise and Main CUI files
- Work with workspaces and dashboards

About the Speaker:

Chris is a veteran AutoCAD user of over 20 years. He has been a CAD drafter, CAD manager, trainer, and consultant in a variety of industries. He is currently the Director of CAD technology for WD Partners, an international, multi-unit design and development firm. Chris is currently serving as the Vice President for the AUGI Board of Directors, and is Manager of AUGI's monthly HotNews publication where he also writes the popular TIPniques column.

chris.lindner@wdpartners.com



Autodesk
University
2007

1 Introduction

Since AutoCAD 2006, Autodesk has taken some innovative steps to modernizing how we tailor the AutoCAD environment. Rather than having to do customization with a variety of tools, practically all of AutoCAD's user interface (UI) elements (toolbars, menus, shortcuts, etc.) can be modified and managed from within AutoCAD itself using the new Customize User Interface (CUI) editor. This course will give you the fundamentals to fully leverage the new CUI

2 Out with the MNU, in with the New

The CUI is not perfect, but is a step in the right direction. For years, users have lived with and come to accept the nuances of MNU/MNS files.

2.1 What I Won't Miss

- Edit the MNU or MNS file? – I think we've all lost some customization by accidentally loading the MNU file, overwriting some modifications that had been save (when edited from within AutoCAD) to the MNS.
- Cascading problems - If you've ever incorrectly terminated or forgotten to terminate a cascade menu with the "<-" code, you know what this is about.
- Randomly named icons files – Pre-CUI releases of AutoCAD had a mind of their own when it came to naming and placing BMP files. No longer will it create icons with names like RCDA4295.bmp placed in hard-to-predict folders.
- Multiple editors – You could edit a few things from inside AutoCAD, but you always had to resort to an external text editor for some of the heavy lifting and tweaking.
- Edit-Load-Test cycle – Since MNU/MNS files had to be compiled, testing your customizations required you to exit AutoCAD or unload and reload the menu.
- Single definition of macros – If you placed a macro on a toolbar and a pull-down menu, when you needed to change it, you had to remember to do it in each place.
- Transferring customization between menus – Knowing what to copy from one MNU/MNS and where to paste it in another could be tricky.
- Sharing\protecting company menus – The only option with MNU/MNS files was to rely on the file's read-only attribute or placing them on a secure network location.

2.2 The More Things Change...

- Macros – Macro syntax is the same as before.
- Icon editor – The clunky icon editor is unchanged, and unimproved.
- Partial menus – Same as before.
- Menu-related AutoLISP (MNL) files – No change.

3 Comparisons

When learning something new, making a connection between what you knew and what is new can flatten the learning curve. This section will build in your knowledge of customization in pre-CUI releases. We will look at the CUI as it relates to MNU/MNS files.

3.1 Customization Files

<i>Old</i>	<i>New</i>	<i>Type</i>	<i>Notes</i>
MNU	CUI	Customization File	The MNU & MNS files are text-based, editable by Notepad or a similar ASCII editor. The CUI is an XML-based file and is edited by the CUI editor in AutoCAD.
MNS			
MNC			
MNR	MNR	Menu Resource	No change in concept. Contains the bitmaps that are used by the user interface elements.
MNL	MNL	AutoLISP Menu Source	No change in concept. Automatically loads lisp code with its accompanying MNS or CUI file.
BMP	BMP	BMP	No change in concept.

3.2 Customization Contents

<i>MNU/MNS Sections</i>	<i>CUI Elements</i>	<i>Notes</i>
***MENUGROUP	"Customization group"	No change in concept.
***BUTTONS(1-4)	"Mouse Buttons"	1-Click, 2- Shift+Click, 3- Ctrl+Click, 4- Ctrl+Shift+Click
***AUX(1-4)	Legacy – "Tablet Buttons"	1-Click, 2- Shift+Click, 3- Ctrl+Click, 4- Ctrl+Shift+Click
***POP0	Shortcut Menu – "Object Snap Cursor Menu"	Referenced by Ctrl+ Right-Click or Shift+Right-Click
***POP(1-n)	"Menus"	Pull-down menus & sub-menus (cascade menus)
***POP(500-n)	Shortcut Menu – "<name> Object(s) Menu"	Context-sensitive, object specific, right-click menus
***TOOLBARS	"Toolbars"	Includes "Flyouts" (i.e. "nested" toolbars)
***IMAGE	Legacy – "Image Tile Menus"	
***SCREEN	Legacy – "Screen Menus"	
***TABLET	Legacy – "Tablet Menus"	
***HELPSTRINGS	"Description"	
***ACCELERATORS	Keyboard Shortcuts – "Shortcut Keys"	
n/a	Keyboard Shortcuts – "Temporary Override Keys"	



n/a	"Dashboard Panels"	
***POP(500-n)	"Double Click Actions"	

3.3 Customization Commands

<i>MNU/MNS Commands</i>	<i>CUI Commands</i>	<i>Notes</i>
MENULOAD	MENULOAD or CUILOAD	Loads a MNU, MNS, or CUI file. Note: dialog box doesn't indicate Main, Enterprise, or Partial CUIs
MENUUNLOAD	MENUUNLOAD or CUIUNLOAD	Unloads loaded CUIs.
MENU	MENU	Loads a CUI as the Main customization file.
TOOLBAR	TOOLBAR	
n/a	CUIIMPORT	
n/a	CUIEXPORT	
n/a	CUI	Launches the full CUI editor
n/a	QUICKCUI	Displays a "collapsed" CUI editor

3.4 AutoLISP Functions

There are no changes to menu-/cui-related AutoLISP functions.

(menugroup <i>groupname</i>)	Verifies that a menugroup is loaded. Example: (menugroup "acad")
(menucmd <i>string</i>)	Issues menu commands, or sets and retrieves menu item status.

4 Migrating MNU/MNS Content

Now let's look at what's involved in taking an MNU/MNS file from a previous release of AutoCAD and bringing it into AutoCAD 2008 as a CUI file. You've put a lot of work into those old MNU/MNS files that you don't want to repeat.

4.1 To Migrate or Not To Migrate

Menus notoriously grow as we add more and more functions, but they don't often shrink. We rarely "clean house" because we "just might need it again someday". Upgrading to the CUI format is a good time to be "selective" about what you carry forward into your new CUI(s). Here are a couple things to consider at this time:

- Unused content – Those macros, toolbars, or menus that you created for a specific project and which are no longer needed. Perhaps they were created for a specific purpose or project which has long been passed. This is the time to weed out.
- Obsolete content – With each release of AutoCAD, it seems Autodesk will add a new feature that you carefully crafted as an elaborate macro. It may be time to "kiss it goodbye". Become familiar with the new features added in each release. As you bring customizations forward, evaluate our custom tools that may have been rendered obsolete by the new release.
- Legacy content – The CUI now classifies a few of its sections as "legacy"; tablet (auxiliary) buttons, table menus, screen menus, & image tile menus. If you are still using any of these menu sections, they are still supported, but the term "legacy" doesn't exactly instill confidence in their future. You may also want to consider moving the legacy contents onto a new CUI section. For example, you may want to convert your Image tile menus into a Tool Palette.
- Organizing the contents – It may also be a good time to reconsider how our customizations are incorporated into your CAD environment. You want an environment that allows customization to enhance production, but will not be painful to maintain or challenging to upgrade with future releases.

Comment [cs1]: Examples "Glue" (Join) and "VPLock"

Comment [cs2]: Delete Tablet & Screen menu sections. May improve CUI editor speed.

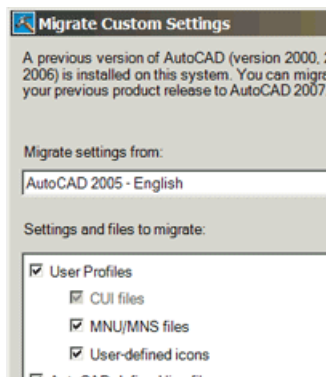
Comment [cs3]: Balancing control & flexibility
Example: Single menu (i.e. Express) or integrated. Replacement toolbars or supplemental.

4.2 Carrying It Forward

Once you've decided which customizations "makes the cut", it's time to start the process of bringing it into a new CUI. There are a few options available for bringing previous-release customizations forward into 2008.

1. Migrate – AutoCAD's "Migrate Custom Settings" application will migrate settings from a previously installed release of AutoCAD, including converting MNU/MNS files to CUI files. Generally, you will be given an option to migrate content when you first start 2008 after an upgrade from a previous version. If you choose not to at that time, you can still initiate it later by going to Start > All Programs > Autodesk > AutoCAD 2008 > Migrate Custom Settings.

Caution: There have been reports that migrating a CUI from a previous release to a newer release may corrupt it, preventing it from being used in the previous release. Solution: make a copy of your CUI & MNR files just to be safe.



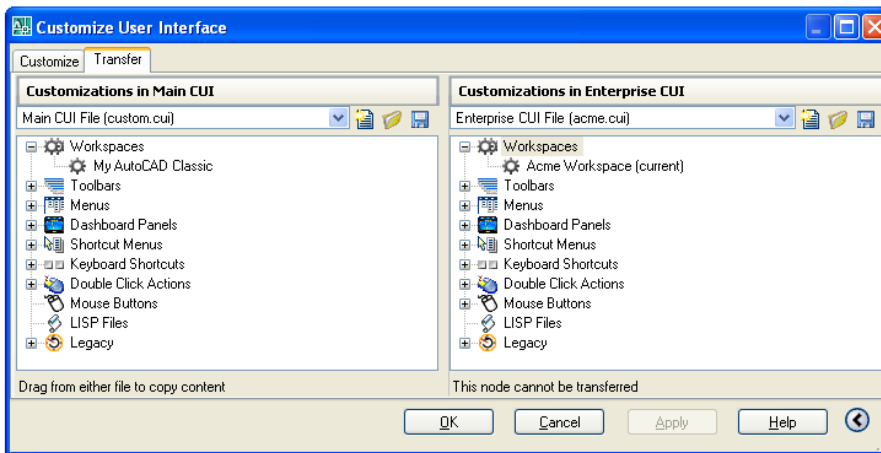
- Convert – There are various ways that you can “open” or load an MNU/MNS file. You can use the “Open” icon in the CUI editor, or use the CUILOAD command. However you choose to do it, AutoCAD will convert it *in its entirety* to a CUI file automatically.

If the MNU/MNS file you are converting has partials under it, then they will be converted to CUIs as well.

Caution: When converting a MNU/MNS in this fashion, the related MNR file gets rebuilt, rendering it unusable in previous versions of AutoCAD.

- Transfer – If you prefer to be selective about what you bring forward, then the CUI editor’s Transfer tab will be your best bet. The Transfer tab essentially lets you have two CUI files open simultaneously, letting you drag-and-drop from one to another. You can use the Transfer tab to “save” content that you don’t want to delete by putting it in a “archive” cui. You can also use the CUIIMPORT command (Tools->Customize) which opens the CUI editor and defaults to the Transfer tab.

Comment [cs14]: Use the Ctrl-select and Shift-select to select multiple elements for transferring.



Tip: In addition to converting MNU/MNS files, the Transfer tab can be used to “purge” existing CUI files. Just transfer the customization that you want to keep into a new CUI and save it as a new CUI file.

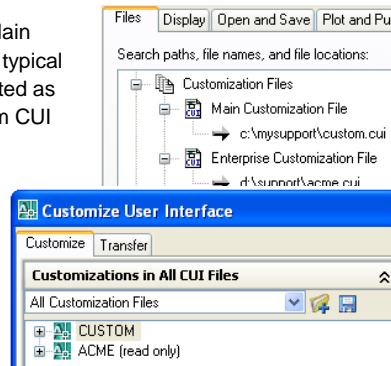
5 Managing CUI files

The “CUI” acronym can mean many things. Not only is it an AutoCAD command, it represents a file type, as well as a type of Customization file. There are two types of Customization file assignments, “Main” and “Enterprise”. A third term, “partial”, refers to how a CUI file is loaded.

5.1 Enterprise and Main Customization Files

Prior to the CUI, protecting a company’s menu customization was either accomplished by difficult-to-enforce policies, or the ingenuity of a company’s CAD Manager or IT staff. The CUI offers a solution. There are now two new designations for CUI files, “main” and “enterprise”. The concept is this: any CUI file can be designated as either the “Main” or the “Enterprise” customization file. In the CUI editor, you make changes to the Main customization file, but not to the Enterprise customization file. A typical scenario would be that the corporate, shared CUI file is designated as the Enterprise customization file, while a user’s personal, custom CUI file is designated as the Main customization file.

Specifying the Main and Enterprise customization files is made on the Files tab in the Options dialog box on the “Customization Files” node. Simply “Browse” to the desired CUI file for each branch. The CUI designated as the Enterprise will then be shown as “read-only” in the CUI editor. In the example (right), the Main customization file, CUSTOM, is fully editable. The company’s cui, “ACME”, is protected as the Enterprise customization file.




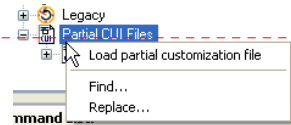
Since these CUI files are assigned in the Options dialog, there’s nothing preventing a savvy user from reassigning the CUI files themselves. So, while the Enterprise designation helps, it is not bullet-proof. Combining it with a secure network location may seem redundant, but is the only way to make a CUI file truly secure.

CAD Managers who also must do production may want to set up different profiles to easily switch between a production environment and a working environment. A “production” profile would have the company’s primary CUI file designated as the Enterprise (read-only) customization file to prevent any accidental changes. When it comes time make changes to the CUI, a “maintenance” profile would be used which has the company’s primary CUI file designated as the Main customization file.

Comment [cs15]: Switch between “AU Manager” and “AU User” profiles to show class difference.

5.2 Partial CUIs

The basic concept of a “partial” CUI is the same as prior releases. A partial CUI essentially “piggy-backs” onto a Main or Enterprise customization file. There is no fundamental difference between a CUI file that is partial-loaded and any other CUI file. CUI files that are intended to be used a partial CUI usually will not contain all of the customizations that a Main or Enterprise CUI does. The Express tools CUI file (acetmain.cui) is a perfect example. It is not intended to be a full-blown CUI file but rather a supplement to an already loaded CUI. Partial menus can be loaded by right-clicking on the Partial CUI Files node, by the CUILOAD command or by the  button in the “Customizations In” pane.



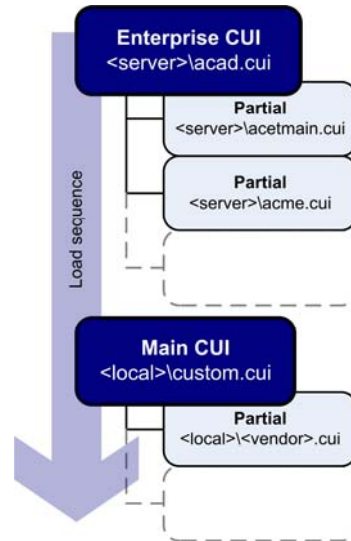
Comment [cs16]: Load the “AU Samples.cui”.

5.3 What Goes Where

With all these possibilities (Main, Enterprise, & Partial), determining where to put customizations becomes an important consideration. For example, if a keyboard shortcut is defined in both the Main and Enterprise customization files, what order are they loaded in and which definition will take precedence? The load sequence of CUIs is illustrated by the graphic at the right. The load priority is show in the table below.

	First loaded	Last loaded	No effect
Workspaces	● ¹		
Toolbars			●
Menus			●
Dashboards			●
Keyboard Shortcuts		●	
Double Click Actions		●	
Mouse Buttons	●		
LISP files	●		
Legacy	●		
MNL files	●		

¹ Workspaces defined in partial CUIs are ignored; only those define in a Main or an Enterprise CUI file are brought in when the CUI file is loaded.

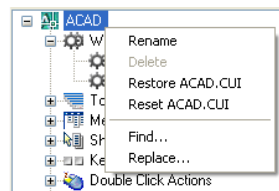


Since the Main customization file (and its partials) is loaded last, this makes it possible for users to create and manage their own “Toolbars”, “Menus”, “Dashboards”, “Keyboard Shortcuts”, “Double-click Actions”, without impacting the company’s customization (which is protected in the Enterprise CUI). The Enterprise CUI will control Workspaces All other customizations are protected in the company’s Enterprise CUI and are either unaffected by load order or are defined by the “first loaded” CUI.

Caution: Shortcut menus (not listed in the chart above) have been to be problematic with the CUI. It seems that the shortcut menus would only load from one CUI, ignoring any shortcut menus defined in other loaded CUI’s. Refer to AUGI threads <http://forums.augi.com/showthread.php?t=32551> and <http://forums.augi.com/showthread.php?t=55923> for further explanations and workarounds.

5.4 Backing up CUIs

When changes are saved, a backup is made of the previous CUI, named <filename>.bak.cui. If you mess up a CUI, you have two options. On the right-click menu, the “Restore” option will restore the current CUI with the most recent backup. The “Reset” option will replace a standard CUI with the originally installed cui file from the “C:\Program Files\AutoCAD 2008\UserDataCache\Support” folder. Note: the Reset option will not work for custom CUI files. It will only work for CUI files that shipped with the Autodesk product.



6 Working with Commands

A “command” is the basic building block in the CUI. It defines how the command looks and behaves in the CUI. This section explores what a CUI command, what it isn’t, how to create one and how to use them.

6.1 When a Command is not a Command

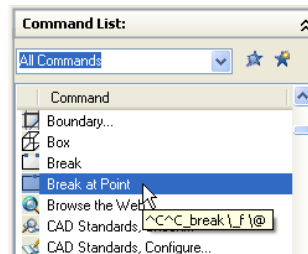
The term “command” can refer to different things in AutoCAD. Here’s how I differentiate between them.

1. An “AutoCAD” command: This is a command that ships with AutoCAD. This is what we mean when we mention the “LINE” *command* or the “RENDER” *command*. It is what you would type in at the command line.
2. A “custom” command: This would be a user-defined function, such as an alias, AutoLISP function or VBa routine.
3. A “CUI” command: A CUI command is basically a macro that calls an “AutoCAD” command or custom command. A CUI command contains properties to define what it does (in the form of a macro), and how it will appear on the interface (name, description, icon).

6.2 The Command List

CUI commands are defined and stored within a CUI file. The CUI commands for all loaded CUIs are listed in the Commands List pane. The command list can be filtered by selecting a specific CUI in the drop-down list, to show only the commands defined in that CUI.

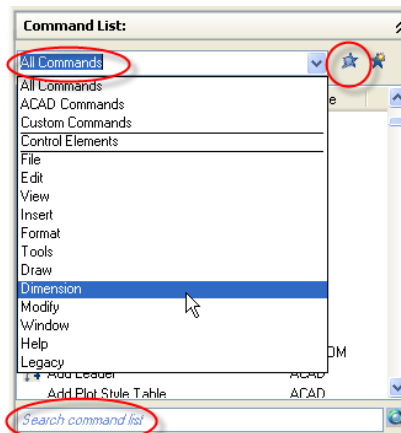
CUI commands are listed by their command name (as assigned in the Properties pane). The CUI command’s name may not be the AutoCAD command that you are familiar with, but rather the name that you would see when it is placed on a toolbar or menu. For example, the in the Command List at the right, the “Break at Point” *CUI command* is highlighted. This is the name that would appear on the pull-down menu, but this *CUI command* will actually execute the “BREAK” *AutoCAD command* as part of the macro.



Since the CUI commands’ names often do not reflect the actual AutoCAD command in the Command List, there are a few tools to help you locate a desired command. The drop-down list that says “All Commands” can be used to select pre-defined groups of commands (as shown, left). Selecting a specific group from the list will filter the list accordingly.

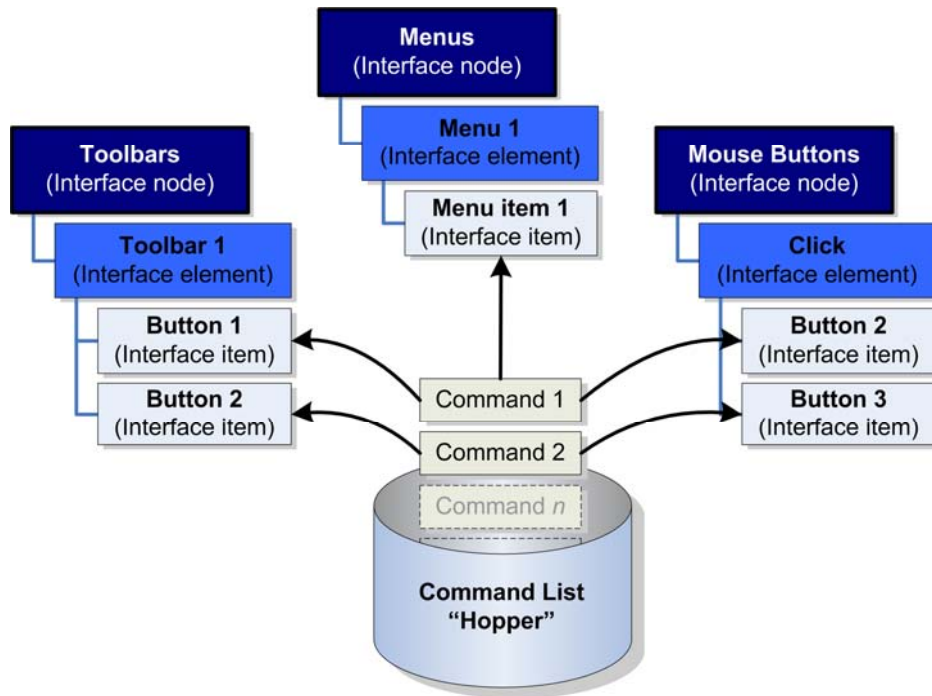
Also, the Find icon (upper right corner) will search through all loaded CUIs and show where a specified text string is found anywhere in the loaded CUIs.

The “Search command list” will quickly filter the command list to display only those commands that contain the specified text string.



Each CUI command is assigned a unique "Element ID". By giving each CUI command an identity (its element ID), commands can be defined once, and referenced (by their ID) multiple times throughout the interface. The Command List pane serves as a sort of "central source" for command definitions. This allows for CUI commands to be used in various places throughout the CUI, with each instance referring back to a common definition.

Comment [csl7]: Mention as being similar to a block definition or an xref. Mention that CUI command definitions can't be shared between CUI (they are unique to a single CUI) The CUI is actual "hopper".

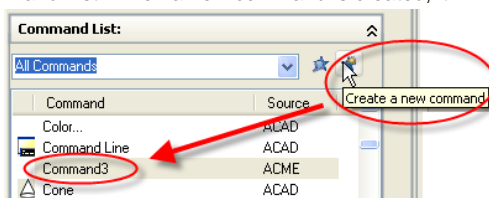


6.3 Custom CUI Commands

Creating your own CUI commands is a simple process. It involves basically two steps.

1. Create a new CUI command by clicking the "New" button in the "Command List" pane, or by selecting "New Command" on the right-click menu in the Command List. When a new command is created, it appears in the command list named "Command#".

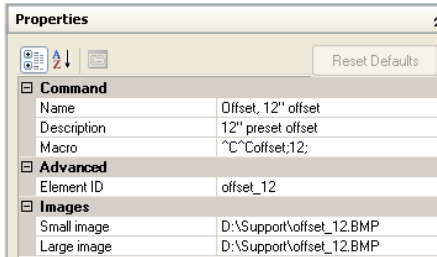
All new commands are created in the Main CUI. Note that the Source column (at right) says "ACME". This means that the new command is being defined in the ACME.cui file, which in this case has been designated as the Main CUI.



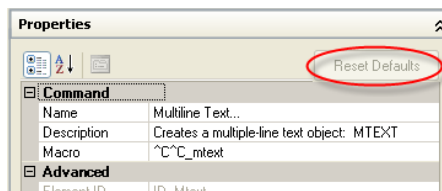
Tip: The Transfer tab can be used to "move" a command to a different CUI.

2. Assign properties in the Properties pane.

- a. Name: The command's name is what it will be listed as in the Command List, and also the label that will be shown if it is used in a pull-down menu.
- b. Description: This will be displayed on the status bar or as a tooltip if it is used on a toolbar or dashboard.
- c. Macro: This is the part of the command that actually does the work. The macro format is the same as in pre-CUI releases of AutoCAD.
- d. Element ID: This is the command's unique identification label. On certain commands, the ID can be user-defined to allow programs to call it by its ID.
- e. Image: These are optional icons that visually represent the command. Images in the CUI can be pathed so a specific location reducing the likelihood that an icon can't be found.



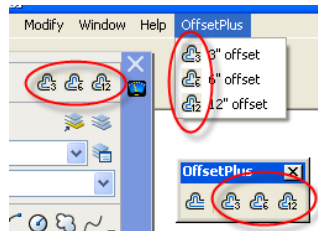
Tip: If you modify the properties of one of the CUI commands that are defined in the ACAD.cui, you can restore it back to its original properties by using the "Reset Defaults" button.



6.4 Command Images

Historically, images were associated with toolbar buttons only. Now, every CUI command can be assigned an image which appears on toolbars, menus and dashboards.

For the most part, the same rules still apply for working with images. Images can come from a MNR (menu resource) file or from individual BMP files if the path to the BMP files is included as one of AutoCAD's search paths. AutoCAD 2007 introduced a new search path option "Custom Icon Location". This path specifies a default location for AutoCAD to place any custom icons that you create. It is also possible to assign a full path to an image file in the Image properties of a CUI command, similar to assigning a hard-path to an xref.



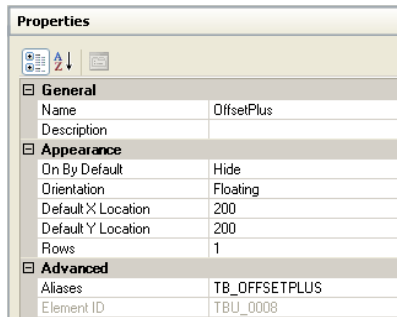
7 Interface Elements

In order for a CUI command to actually be used and visible in AutoCAD, it must be placed on an “interface element”. Elements may be visible on the interface (like pull-down menus, toolbars, etc.) or just assignments (like keyboard shortcuts, overrides, etc.). Populating most interface elements is simply a process of “dragging-and-dropping” commands from the Command List pane (remember the “hopper”?). This section looks at the various elements and their unique properties.

7.1 Toolbars, Menus, and Dashboards

Toolbars, menus and dashboards are created very similar within the CUI editor. In this exercise, we will create a new toolbar for some custom OFFSET commands.

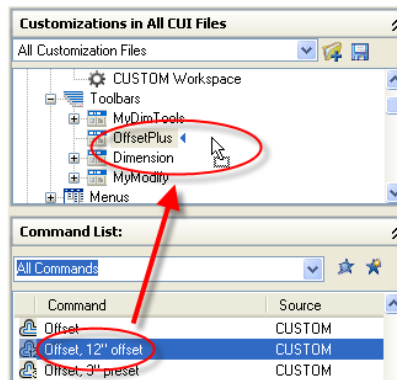
1. Create a new toolbar by right-clicking on the Toolbars node, select New, and changing its default name (Toolbar#) to “OffsetPlus”. Toolbars (and all other Interface elements), like CUI commands, have their own set of properties (right).



2. Place commands on the toolbar by drag-and-dropping onto the “OffsetPlus” toolbar. It is possible to drag-and-drop multiple commands at a time using Shift+pick or Control+pick when selecting commands. It is not possible to drag items from one interface element to another (Ex. from Toolbar to a Menu).

Don't forget about the Find and Search tools mentioned in Section 6.2.

Once placed on the toolbar, commands can be rearranged by dragging them into the order desired. Separators can be added (to group like commands together) by right-clicking on an interface item.



Tip: When a CUI command is placed on a Toolbar, Menu or Dashboard, the Properties pane shows a new field, the “Display name”. This allows the name that appears on the Menu or as a tooltip to be different from the CUI's command name. In this way CUI command names can be structured in a way that makes sense when they are listed in the Command List pane, but different names can be used to improve how they appear in AutoCAD's interface.

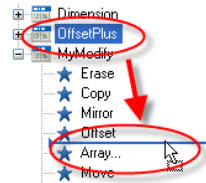


7.2 Flyouts

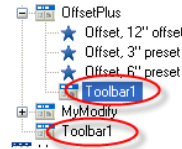
A flyout is a special toolbar that can be thought of as a “nested” toolbar, or a “toolbar within a toolbar”. The flyout is basically a reference to another (source) toolbar, in much the same way that an xref is a “nested drawing” that references another (source) drawing. As a reference, changes made to either the source or the flyout (reference) will be reflected in both toolbars. The flyout can have its own name (independent of the source toolbar’s name) and can be deleted without impacting the original toolbar (like detaching an xref). Deleting the source toolbar, however, will cause an error about a missing toolbar (i.e. the source).

Flyouts can be created two ways; via drag-and-drop technique or by right-clicking on an existing toolbar.

1. The Drag-and-drop method can be used if you have an existing toolbar that you would like to act as a flyout. In this example, we simply drag-and-dropped the “OffsetPlus” toolbar onto the “MyModify” toolbar (right). This new nested toolbar will then reference the original “OffsetPlus” toolbar.



2. If a toolbar does not already exist to be used as a flyout, you can create a new flyout directly. Right-click on an existing toolbar and select “New -> Flyout”. This will create a new “nested” toolbar (named “Toolbar#”) and will also create “source” toolbar (also named “Toolbar#”) which it references (right). Populating either one will update the other.



Tip: The “On By Default” property of the source toolbar is automatically set to “Hide”. Since this toolbar was created to act as a flyout, AutoCAD assumes that the source toolbar doesn’t need to be displayed as a floating toolbar.

7.3 Sub-menus

Similar to a toolbar and a flyout, a menu can contain a “sub-menu” (formerly called “cascaded menu”). The main difference between a flyout and a sub-menu, though, is that while a toolbar’s flyout is a reference to another source toolbar, a Sub-menu is a “child” menu that is completely independent. It exists only within its “parent” menu.

Sub-menus, like Flyouts, are created by drag-and-drop or by right-clicking on an existing menu and selecting New->Sub-menu. However, drag-and-dropping a menu onto another menu *moves* the menu, rather than creating a reference to the original, as with flyouts. To copy the menu instead, hold the Control key down while drag-and-dropping.

7.4 Shortcut Menus

Shortcut Menu (also called “right-click” or context menus) are displayed when you right-click on certain objects or in various areas of the screen. Shortcut Menu are almost identical to Menus, including the ability to contain Sub-menus, but they typically contain fewer interface items than a normal Menu.

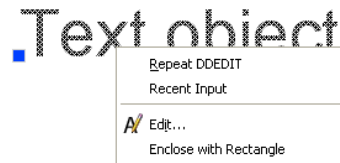
The key component of a Shortcut Menu is its Alias. The alias “associates” the Shortcut Menu with the desired entity. In the “Dimension Objects Menu” example (right), if you removed the “OBJECTS_DIMENSION” alias, the Shortcut Menu would cease to function when you right-click on a dimension.

General	
Name	Dimension Objects Menu
Description	
Advanced	
Aliases	POP504_OBJECTS_DIMENSION
Element ID	PM_0011

Shortcuts Menus can have a “singular” alias, as in “OBJECT_BLOCKREF” or “OBJECT_HATCH”, or a “plural” alias as in “OBJECTS_BLOCKREF” or “OBJECTS_HATCH”. The Shortcut Menus having plural aliases only appear when more than one object is selected.

In the example below, we add an item to the Text Object Menu, using a pre-defined custom command.

1. Locate the “Text Object Menu” in the Shortcut Menus node.
2. In the Command List, locate the “Enclose with Rectangle” command and drag-and-drop it onto the “Text Object Menu”.



Comment [CSL8]: Do this using the AU Manager profile.

7.5 Keyboard Shortcuts


The Keyboard Shortcuts node is unique because it represents two different types of keyboard-based customizations; “shortcut keys” (also referred to as “accelerators”) and “temporary overrides” (introduced in 2006).

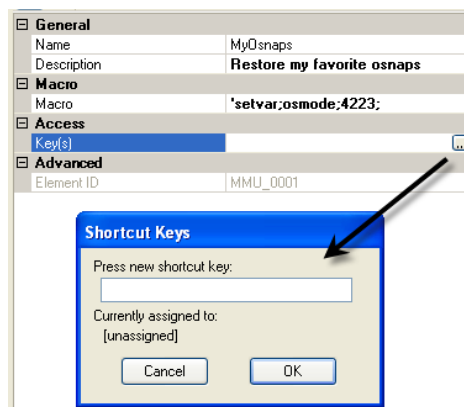
When you select the “Keyboards Shortcuts” node, the Shortcuts pane shows a complete list of both Shortcut Keys and Temporary Overrides. This list is useful to quickly see if a key (or key combination) has already been assigned and allows you to see examples of how they are defined.

7.5.1 Shortcut Keys

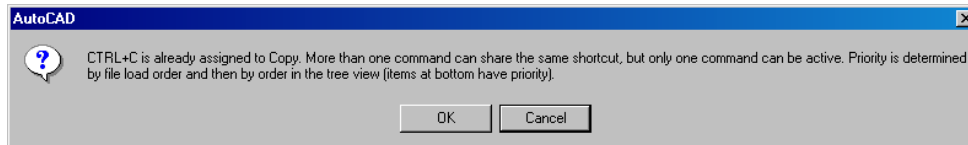
We’ve come to rely on “Shortcut Keys”, like the standard Windows shortcuts such as F1 (help) and Ctrl-C (copy-to-clipboard) to the AutoCAD specific shortcuts like F8 (ortho toggle) and Ctrl-3 (tool palettes toggle).

This example creates a new Shortcut Key using a pre-defined custom command that restores a certain combination of object snaps.

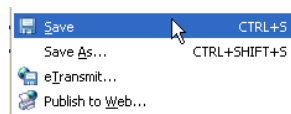
1. Oddly, you can’t select “New” from the right-click menu in the Shortcut Keys element the way you typically do on other elements. You must first drag-and-drop the command from the Command List onto the Shortcuts node.
2. Once placed in the Shortcut Keys element, the command’s properties now show a new property named “Key(s)”. Picking the Ellipsis  will display the “Shortcut Keys” dialog. This dialog also shows any existing shortcut assignments
3. Assign the shortcut key combination by pressing the “Control” and “M” keys together.



AutoCAD allows commands to share the same shortcut, but it warns you that only one of them can be active. The priority is determined by the order that the CUI files are loaded (see Section 5.3) and the order in which they are listing in the Shortcut Keys element.



Tip: If a CUI command has been assigned a Shortcut Key, the shortcut is automatically displayed when the command is placed on a menu. In the example at the right, no special coding was required to add the “CTRL+S” or “CTRL+SHIFT+S” next to the menu items. The shortcut is shown on the menu automatically. If the command’s shortcut is changed, the menu item will be updated as well.



7.5.2 Temporary Overrides


Temporary overrides, introduced in AutoCAD 2006, are a unique form of keyboard shortcut because they can perform a function when the key is pressed (“key down”), when the key is released (“key up”), or both. Because they are referred to as “temporary”, they have the ability to change (or override) something momentarily when a key is pressed and then reset it when the key is released. The override is only “active” as long as the associated keys are pressed down. An example of this type behavior is when you hold down the Control key to prevent a toolbar or palette from “docking”. While Control is pressed, the “auto-dock” tendency is temporarily suspended.

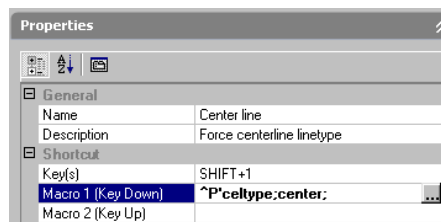
Some overrides that will be very popular are the object snap overrides. For example, even if the Endpoint object snap is not on, you can use the Shift+E override to temporarily activate the Endpoint object snap.

Creating a Temporary Override, goes as follows:

Comment [CSL9]: Trivia: TO's are the only customization that do not use CUI commands!

1. On the Temporary Override Keys node, right-click and select New > Temporary Override.
2. The new override's default properties are displayed in the Properties pane. This example will create an override that will force the current linetype to Center.

- Name: Centerline
- Description: Force centerline linetype This description is option and is not displayed in AutoCAD.
- Key(s): Click the Ellipsis  to display the Shortcut Keys dialog. Press the “Shift” and “1” keys.
- Macro 1: ^P'celtype;center;
- Macro 2: *leave blank*



Here are some rules that apply to Temporary Overrides:

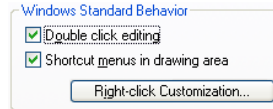
- The Shift key is the only "function" key that can be used.
- Overrides can be given multiple key assignments. The "Object Snap Override: Center" override, for example, is listed twice, once with a "Shift+C" and once with a "Shift+,". Apparently the logic for creating the multiples is to accommodate which side of the keyboard you work from most. The additional overrides can be safely removed. Unfortunately, it is not possible to assign multiple keys to a single override. You must create a separate override item for each different key combination.

Tip: If you press & release the Shift+<key> combination too swiftly, AutoCAD interprets it as a capital letter. When using overrides, press and hold the keys for a brief second to activate the override's key-down macro.

7.6 Mouse Buttons

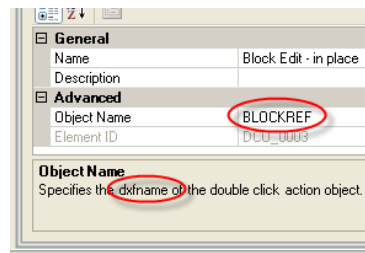
All mouse buttons, except for the left button, can be defined in the CUI. There are a few challenges when working with mouse buttons.

- Today's mice often have their own means of assigning button actions through special drivers. Some of the button-related customization may have to be done at the system level, not within the CUI editor.
- The right-click (Click-Button 2) will only display shortcut menus if the "Shortcut menus in drawing area" option (in Options\User preferences) is checked.
- If AutoCAD's mbuttonpan system variable is set to value of 1, the middle button's pan function will take precedence over the middle button's CUI assignment (Click-Button 3).
- Since each button can have multiple actions assigned (click, shift+click, control+click and shift+control+click), knowing which button does what is sometimes hard to determine. Allow some time for trial and error when customizing them.



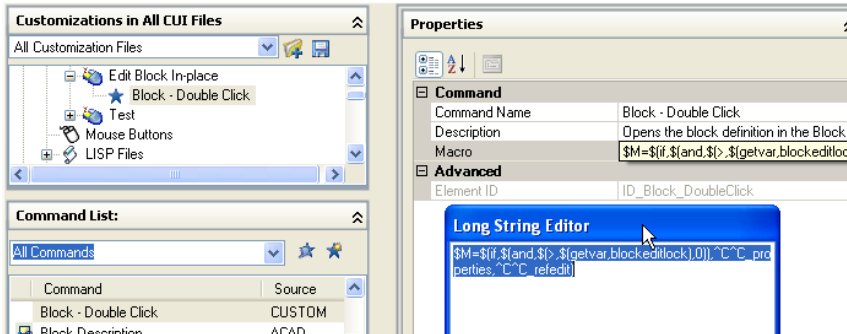
7.7 Double Click Actions

The Double Click Action controls the event that occurs when you double-click on an object. The Double Click Action can be named anything, but it has to be associated with an object, similar to the Shortcut Menu. The "Object Name" must reflect the proper "dxfname" of the object you are double-clicking on. For a fairly comprehensive list of objects and their dxfnames, refer to the Double Click Actions node in the ACAD.cui.



In 2008, the default Double Click Action on a block is to open the Block Editor. I prefer to have it execute the "Edit Block In-place" (Refedit) function. This can be changed by modifying the CUI command that was placed on the Double Click Action for a Block object. The "Bedit" command in the macro is changed to "Refedit".

Comment [CSL10]: Show example of Circle DCA starting the Circle command.



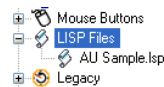
7.8 Tool Palettes and the CUI

When the CUI was introduced with AutoCAD 2006, one of the complaints was that it didn't support customizing tool palettes. Since 2007, it is now possible to populate a tool palette by dragging commands from the Command list onto tool palettes. The CUI can also be launched from a tool palette by right-clicking on a palette, and selecting the new "Customize Commands" option.

Gotcha: The only disconnect that remains is that a command placed on a tool palette is not linked to the CUI command. So any commands placed on a palette will not be updated if the original CUI command is modified.

7.9 LISP Files

CUI files still support MNL (menu lisp) files the same as before, but there is a new option. In the "Customizations In" pane, there is a new "LISP Files" node. This allows you to "associate" any AutoLISP file (LSP, MNL, VLX or ARX) with a particular CUI file. When the CUI is loaded, the associated AutoLISP file(s) is loaded as well.



Gotcha: There are no properties for LISP items. It would be nice to be able to modify the path of an item. Currently, if an AutoLISP file moves or is renamed, you will have to delete the item and recreate it in order to repath it.

8 Workspaces

Workspaces are one of those new CUI concepts that take a while to sink in. It's not a critical concept when first getting into the CUI, but is not something that should be ignored either. Essentially, a workspace is a named environment consisting of a combination of interface elements (toolbars, menus, etc.).

8.1 Workspaces vs Profiles

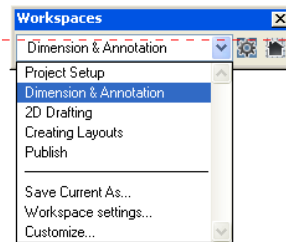
Everyone new to the CUI wrestles with the distinction between workspaces and profiles. Here is a brief comparison:

- Workspaces define what interface elements are visible. It is essentially a way to streamline your "AutoCAD desktop", eliminating unneeded toolbars, menu, palettes, etc. Changing workspaces does not change which CUIs are loaded, it just changes how much of them you actually see.
- Profiles contain what CUIs are loaded, but do not control what parts of the CUI are visible. Many of the settings controlled by a profile are invisible to the user (system settings, etc.), whereas a workspace is all about the organization of the AutoCAD environment.



8.2 Working with Workspaces

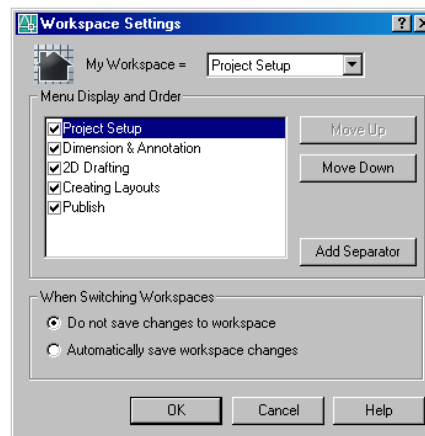
Sometimes an example is worth a thousand words. The "acadSampleWorkspaces.cui" file that ships with AutoCAD has some nice pre-configured workspaces to give us an introduction. It must be designated as the Main or Enterprise customization file in order to use the workspaces defined in it. Once you get the general idea of what a Workspace is, then you can begin to modify them or create you own.

The Workspaces toolbar is central to choosing and managing workspaces. It contains a drop-down for switching between defined workspaces. With the "Save Current As" option you can save the current interface configuration so that it can be restored later by name.



Comment [CSL11]: The workspaces defined in this example CUI are "task" related, but they could easily be client, project or discipline related.

The Workspace Settings icon  on the toolbar displays the Workspace Settings dialog for general management of workspaces. In this dialog box, you control existing workspaces. The "My Workspace" value can be used to choose a "default" or preferred workspace which can easily be restored by picking the  icon on the Workspaces toolbar.



The Menu Display and Order box controls which workspaces should be visible and in what order they should appear in the list. So, it is possible to have more workspaces defined than are actively being used.

When switching workspaces, you can choose to automatically save any environment changes to the current workspace. If the "Do not save" option is chosen, changes will be lost when you switch to a different workspace.

8.3 Editing Workspaces

Some workspace modifications can be “saved” by simply switching between workspaces. Other environment changes, though, must be made in the CUI editor.

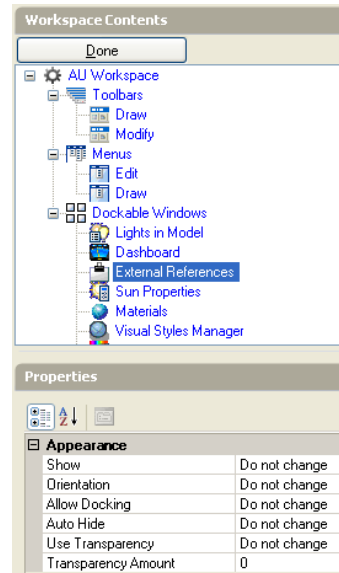
Similar to other interface elements, workspaces can be created by a right-clicking on the Workspace node and selecting New>Workspace. Workspaces can also be imported from other CUI files using the Transfer tab.

Editing a workspace is done in the “Workspace Contents” pane. When you select the “Customize Workspace” button, it puts the CUI in a “workspace customization” mode. In this mode, no changes can be made to the loaded CUI files, nor can new customizations be added. You are simply allowed to select the interface elements that are available in the loaded CUI files that you want associated with this workspace.



Workspaces are where you determine which menus appear on the Menu bar and the order in which they appear. The order they appear in the Menus node reflects how they will appear on the menu bar.

The “Dockable Windows” node allows control of things like Palettes and Dashboards. The Appearance properties in the Properties pane determine whether the dockable window will display or not be changed at all when the workspace is set current. If “Do not change” is chosen, that particular appearance property of the dockable window will not be affected when this workspace is set current. Setting the “Show” property to “No” will turn off that dockable window when that workspace is set current.



9 Conclusion

There are a plenty of resources available and which I relied on for the contents of this document. AutoCAD’s on-line help was always a first stop. In 2008, the addition of video clips in the Help is extremely helpful! The AUGI Forums were another no-brainer, as were Autodesk’s Newsgroups. A number of blogs provide thought-provoking examples, publications, white-papers, and videos. I hope this class has given you the foundation to begin putting the CUI to work for you. Happy CUI-ing!