

Easy Customization Without Programming

Customization Categories

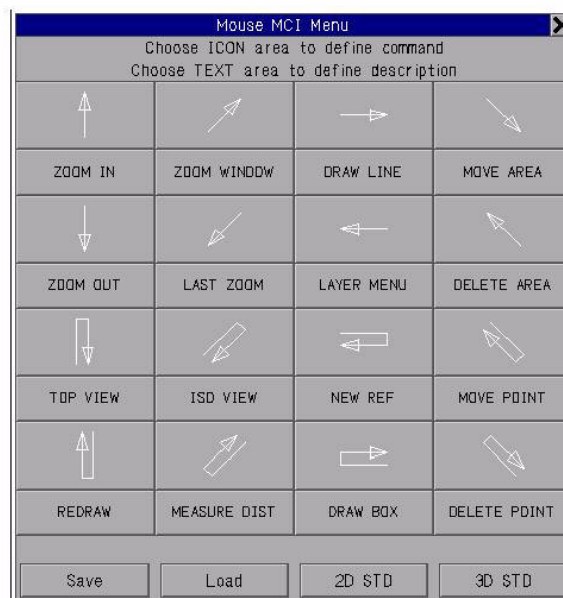
1. Mouse Motion Commands
2. Toolbar Menu Commands
3. Right-click Menu Commands
4. Alias Commands
5. Thumbnails for Entity Style Select Menus

Mouse Motion Commands

The ability to assign up to sixteen of your most commonly used ARRIS commands to a simple gesture of your mouse is arguably the most productive feature ever introduced to ARRIS. It is truly one of ARRIS' most unique features and once mastered, it can radically increase your drawing proficiency and output.

Below is brief explanation of the Mouse **M**otion **C**ommand **I**nterpreter (MCI) menu and its functions.

1. The first step is to become proficient at learning and using the default 16 commands below from the Mouse Motion menu



Notice the top two rows contain (8) single stroke gestures in 45 degree increments, while the bottom two rows contain (8) double stroke gestures in 45 degree increments.

Also notice that six of the boxes above have been assigned to commands that involve view control options (i.e. *Top View*, *Zoom Window*, *Zoom In*, *Zoom Out*, *Last Zoom and Redraw*). This is because manipulation of view control represents the majority of commands that are invoked in a typical CAD session so it makes sense to have them instantly accessible.

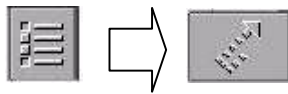
In order to speed up the process of memorizing each command and its respective gesture, I recommend printing out the page above and cutting out the *Mouse MCI Menu* icon. Then attach the cutout to your computer monitor for fast reference. This will greatly reduce the amount of time it takes for all the commands to become familiar to you. I strongly recommend committing a little time each day to assimilate at least one new mouse gesture into your repertoire.

2. The next step is to become so proficient at using the mouse gestures that you eventually want to replace some of the commands not used very often with commands that you use a hundred times a day.

To do this, you first have to discover the mnemonic command associated with the new command you'd like to assign. Typically this can be done by selecting the command from the menu and then entering a *Ctrl R* (the Control key together with the letter *R*) from the keyboard to bring to the mnemonic sequence to the ARRIS prompt line.

For example, let's replace the *Draw Box* command on the default menu with the *Area* command by following these steps:

- Select the *Area* button from the *Query* menu.
- Enter *Ctrl R* from the keyboard. The command string `:mn_desktop;query;:mn_area` appears at the prompt line. You can ignore the first part of this command string, `:mn_desktop;query;` which references its source menu and concentrate on the latter part which is `:mn_area`. This is the actual mnemonic command string for the *area* command.
- Now activate the *Mouse MCI* menu by selecting



and select the mouse gesture icon directly above the *Draw Box* label. This will display the prompt "*Enter command for box:*"

- You respond by typing `:mn_area` (derived from the command string above). Don't forget to include the `:` (semi colon) character at the beginning of the `mn_area` string.
- Now select the *Draw Box* label itself and replace it with the word *Area*. This word will now appear at prompt line when this gesture is invoked in the future.
- Finish by selecting the *Save* button at the lower-left corner of the menu and accept the default file name of `dg_user.st`.

Now the *Area* command will be invoked each time you initiate this gesture in the future. Repeat these steps to reassign additional buttons in the future.

Toolbar Menu Commands

Customizing your ARRIS *Toolbar* menu is another terrific way to increase productivity by reducing the number of button clicks. Rather than browsing for commonly used commands that are buried behind multiple menus, why not map the command or function directly to your Toolbar to allow one-click access?

1. Redefining the Button

The easiest way to do this is to select the button candidate you would like to add to your Toolbar menu by clicking on it with your *middle-mouse* button. If you don't have a 3-button mouse, reach up and select the button by pressing the *F2* key. This will bring up the *Quick Help* menu for the button you selected. Notice there is an option on this popup menu labeled *Toolbar*. When you select this button your cursor will become a *box* icon attached to a *rubber band* followed by the ARRIS prompt "*Select any other menu or drawing area to exit*". Now drag your cursor down to the Toolbar menu and select any button you would like to replace with this new command by selecting it with your *left-mouse* button. Notice the Toolbar button representation now looks like the new command you just placed.

Another way to accomplish this same trick is to first identify the box on the Toolbar menu you would like to replace by selecting it with a *right-mouse* click. This will turn your cursor into a box icon with the word *Toolbar* below. Now select the desired new button from any ARRIS menu by selecting with the standard *left-mouse* button. This brings up the *Toolbar Setup* menu (more on this later). Simply click *OK* at the bottom of the popup menu and *voila*, the Toolbar button representation now looks like the new command you just reassigned.

2. Saving the Button Assignment(s)

To save this placement so it will reappear the next time you start an ARRIS session, select the Toolbar button labeled *TOOLBAR* (to the left of the *Stop Sign* icon). Then select *Save* from the menu, and then select the *def.* value which corresponds to your login name. Otherwise, your new Toolbar menu replacement button will only be temporary which may be okay if that was your intent.

3. Customizing Toolbar Button Assignments

Sometimes the commands stored on the Toolbar menu include both commands as well as built in responses. For example, the *Dist* button on the default Toolbar menu has a built in response for *2 points*, rather than the other possible response of *perimeter*. If that isn't the response you want, you can modify the response. Or you can modify other aspects of the Toolbar button command such as:

- The name of the command (including adding custom command names)
- The specific prompt response or responses
- The number of commands you wish to string together
- Display the button name as a *text string*
- Display the button as an *icon* using a repeated item

All these modifications can be accomplished by first selecting the Toolbar button with your *right-mouse* followed by a *left-mouse* click or *Enter* key. This brings up the *Toolbar Setup* menu mentioned in step 2 above which displays all the elements stored with the button such as the button *display mode* (i.e. text or icon) and the *command string(s)* and possible *prompt response(s)* invoked when the button is selected.

a) Using Text As A Toolbar Button Label

The Toolbar button mode can be changed by selecting *Icon* or *Text* from the upper left corner of the popup menu. If *Text* is selected, the desired button label will display whatever string of text is entered on the opposite line. To break the text into two lines, simply separate the two words with a ^ (*carat character*) such *Move^Area*.

b) Using an Icon As A Toolbar Button Label

If The Toolbar button mode is set to *Icon*, the desired button label will display whatever repeated item *name* and repeated item *library* are displayed in the two boxes on the opposite line respectively.

Using repeated items to graphically represent the button's function rather than just a cryptic string of text can be especially helpful in assisting new ARRIS users in coming up to speed faster (you know what they say about a picture's worth...). For example, I created icons of a text label, a footing detail and a company logo for my Toolbar buttons which bring up the RI Select menu for my notation RI's, my detail RI's, and my company standard RI's respectively.

If this sounds interesting, simply create an RI of any size, scale or color that you would like to use as a Toolbar icon. Keep it very simple since it will become greatly reduced to fit your Toolbar button. Now save the RI into a library that will always be accessible. I add my Toolbar RI's to a library named (oddly enough) *toolbar.ri* and store it in my *STANDARDS* directory.

To add the new RI as a Toolbar button icon, *right-click* on any Toolbar button, then hit *Enter*. Now select the *ICON* box at the top of the popup menu, then select the first empty box to the right and enter the name of your new RI. Now select the next box to the right and enter the name of the RI library into which it was stored.

Don't forget to save the Toolbar menu after customizing it by selecting the button labeled *Toolbar* (next to the *Stop Sign*), then choose *SAVE*, then select the *def:* option (which is typically your login name). This way each user can design their

own icons or they can be shared by selecting the *save to STANDARDS* then *load from STANDARDS* options.

4. Customizing Toolbar Button Assignments

Have you ever wished you could map more than just 23 functions to the ARRIS TOOLBAR menu? One easy way to map more buttons is to add a *Cycle* button that activates a second TOOLBAR menu with another 22 functions. Each time the *cycle* button is pressed, it 'toggles' back and forth between the two saved menus. If this sounds interesting, just follow these steps:

Set Up the First Level TOOLBAR Menu

Assign all the desired functions to your favorite TOOLBAR menu locations. Reserve one button (the far right, for example) to be the *CYCLE 2* button which will take you to your "alternate" or *Second Level* menu.

Set Up CYCLE 2 Button

Right-click on the 'reserved' button (see above) and drag your cursor out to the "drawing" area of your screen and then left-click your mouse to activate the *Toolbar Setup* menu.

First select the *CLEAR BUTTON* to erase all previous button mapping selections.

Now select the *gray box* opposite the word *TEXT* and enter the following string using the keyboard:

CYCLE²

(The '^' is a SHIFT^6 (carat) and will place the word *CYCLE* above the number 2 in the box for aesthetic purposes)

Now select the green box that says, *add new command string here* and type the following string using the keyboard:

NOTE: The middle character is an 'l' (el), not a '1' (one).

:mn_user;l;user2

Now save this modified version of the TOOLBAR menu by selecting the button on the far left of the TOOLBAR labeled *TOOLBAR*.

At the first prompt, pick *save* from the menu choice

At the second prompt, *save toolbar?*, type in the following:

NOTE: The last character is a '1' (one), not an 'l' (el).

def: login_name [the default name is your login name]

Now *drag and drop* different commands to the TOOLBAR to create another variation that you would like to "cycle" to. But don't overwrite your CYCLE 2 button until the next step!!

When you are finished, right-click on the *CYCLE 2* button and drag your cursor out to the "drawing" area of your screen and then left-click your mouse to activate the *Toolbar Setup* menu again.

Now select the box labeled *CYCLE 2* and change it to say

CYCLE 1

Now click on the string of text (opposite the 'X' box) entered in Step 5 and replace it with the following string:

NOTE: The last character is a '1' (one), not an 'l' (el).

```
:mn_user;login_name
```

Now save this modified version of the TOOLBAR menu by selecting the button on the far left of the TOOLBAR labeled *TOOLBAR*.

At the first prompt, pick *save* from the menu choice
At the second prompt, *save toolbar?*, type in the following:

```
user2
```

If everything was configured correctly you should be able to toggle between menus by selecting the *CYCLE 1* and *CYCLE 2* buttons.

Remember that future updates to the first version (*CYCLE 1* menu) must re-saved as *login_name*.

Changes made to the second menu must be re-saved as *user2* (See Steps 6 and 10 respectively)

Right-click Menu Commands

Are you tired of answering the same prompts or picking the same options for commands that you typically answer the same way every time? To reduce the number of mouse clicks and speed up your productivity, you can customize a *right-click* for certain menu buttons that automatically respond to the prompts the way that you usually do.

Let's say you're creating a preliminary drawing with lots of overlapping construction lines and you'd like to join the corners without having to perform a lot of trim and extend commands. As most of you know, setting the *fillet* or *chamfer* command to a *zero* radius will automatically join (i.e. double trim/extend) any two selected line endpoints. How about customizing the *Chamfer* button so that when you *right-click*

it, it will automatically input *zero* chamfer values and quickly prepare you to start joining any two line endpoints? Just follow these easy steps:

1. Select the *Customization Menu* button (near the upper-right corner of your ARRIS menu next to the *Load Plug-in* button) and then select the *Right Mouse Command* menu (the mouse icon with the right button colored in red). Now drag your special *right-click* cursor down and select the *Chamfer* button under the *Place* label from the *Line* menu. This will bring up the *Right Click Setup* menu.

(Note: Some menu buttons will generate the message: *RC Command Uncustomizable* which means that you can't perform this operation here)

2. Under the *USER DEFINED* half of the menu, select the empty blue box to the right of the *DESCRIPTION* label and type in a description such as: *Chamfer with zero radius or Join Lines*.
3. Now select the empty blue box to the right of the *COMMAND* label and type: `<:mn_chamfer;0;0`

(Note: The '<' is a special character which instructs ARRIS to first clear any pop-up menus that may be displayed in the drawing area. The `:mn_chamfer;0;0` tells ARRIS to type the *chamfer* command and enter a *zero* in response to the two prompts which ask for the two chamfer distances)

4. Now select *Accept* from the pop-up menu, and then select *Exit*.

Now, anytime you want to automatically join two line endpoints to form a corner, simply right-click on this box and avoid having to enter the chamfer values. Notice that the description you entered in step 2 above is displayed on the message line when this *right-click* command is initiated. A normal *left-click* on the box will still prompt you for the chamfer values as usual. If you want to return to the original behavior, repeat steps 1-3 above and when prompted to enter the command, select the "USE DEFAULT" option which will reset the originally assigned operation.

Alias Commands

For those users who love to type, ARRIS 8.2 contains 15 predefined one and two-letter alias shortcut commands. These single character commands can be typed at the keyboard followed by the *Enter* key:

a	(arc)	r	(rotate)
l	(single line)	t	(text0)
d	(distance)	z	(zoom)
c	(copy)	ci	(circle)
e	(erase)	bh	(cross hatch)
m	(move)	ri	(repeated item)
o	(offset)	tr	(trim/extend)

If you want to change the existing alias commands or add more, simply copy the *C:\Program Files\ARRIS 82\li\ sys_aliases.st* file into your *Home* or *Standards* directory and rename it *alias.st*. That way if you update your ARRIS software you won't run the risk of having your customized changes overwritten.

Below are two sample lines from the *alias.st* file. The first line describes the content for each field of the *alias* file and the second is a line which defines the alias command for *single line*. Notice each of the four fields are separated by a ; (semi colon).

```
#ALIAS;letters;description;command  
ALIAS;l;SINGLE LINE;:mn__main;='LINE';:mn_line;='s'
```

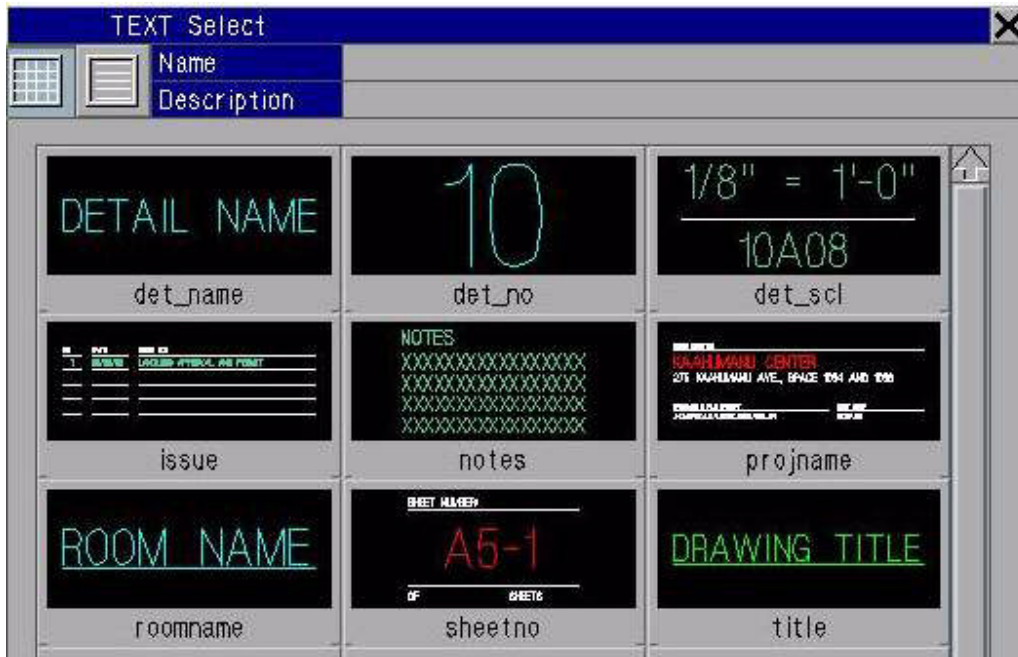
1. Each new line must start with the word *ALIAS*
2. The next field declares the letter or letters to be used as the alias, in this case the letter *l* (el) for line.
3. The next field contains a brief description of the command such as *SINGLE LINE*
4. The last field contains the actual command being aliased, in this case *:mn__main;='LINE';:mn_line;='s'* which invokes the Line menu then the *single line* command.

Just modify any existing line to quickly change the alias character and/or command string associated with the alias. Or add new lines for more alias commands as the need arises.

Thumbnails for Entity Style Select Menus

In addition to creating custom RI icons for your TOOLBAR buttons, you can also create custom RI's for the sample thumbnails that ARRIS uses to represent your user-created Entity Styles for *text*, *line*, *pattern* and *dimension* entities. Why would you want to do this you ask? Well, to make it easier to associate the thumbnails with their usage of course. Trust me, new users love this.

For example, the Gap (ARRIS user extraordinaire) customized their ARRIS text style thumbnails to display the text sample in the context for which it would be used. For *Note* style text, they made an icon of a small paragraph of text. For *Sheet No.* text, they created the string **A5-1** in the color and font they used for sheet numbers. The same for *drawing label text*, *detail bubble text*, etc. (see partial *Text Select* menu below)



To do this, create an RI of any size, scale or color that you would like to use as a style sample for your thumbnail menu. Again, keep it very simple since it will be reduced to fit your sample thumbnail window.

The repeated item library, where ARRIS stores the sample thumbnails for the Entity Styles, is named *mn_sfikons.ri* and is located below the *styles.dir* directory in a sub directory named *default.prj* (or *xxxxx.prj* if you have renamed your directory to something more meaningful).

If you view the *mn_sfikons.ri* library via the *ARRIS Repeated Item Select* menu, you will see that all of the RI have special 3-letter prefixes such as *@dm* for dimension styles, *@ln* for line styles, etc. Although ARRIS won't let you create a new RI that starts with an *@* character, it will let you place any RI with an *@* character already in its name, freeze it, update it, and remake it using the existing name as the default value. This will allow you to overwrite the default entity thumbnail with one of your own design. You will be stuck with the previous RI name (derived from the style name given when it was created), so make sure the style names are accurate. Also, you are limited to 8-character style names so that ARRIS can put the 3-letter *@xx* prefix in front of the corresponding RI name.

Once you have updated the appropriate *mn_sfikons.ri* library with your custom RI's, be sure to make a backup copy of it in the same directory. I say this because the original will be overwritten the first time somebody accidentally selects the *Regen Ikons* button at the bottom of the Entity Select menu.