

Las Vegas, Nevada
November 28 – December 1

Speaker Name: Bob Callori

Course: An Idiots Guide to AutoLISP

Course Description

This is the perfect class for anyone who has never written an AutoLisp program. Learn how to increase your productivity by using command line statements within a lisp file for simple tasks, such as create layers, offset lines, draw hatch patterns and insert attributed blocks. The instructor will furnish all attendees with a disk containing sample toolbars and routines that insert blocks with multiple arrows, leaders and more.

Bob Callori, AIA
San Francisco State University/
IA – Interior Architects, Inc
Email: callori@jps.net

This is the perfect class for anyone who has never written a Lisp program. Learn how to increase your productivity by using command line statements within a lisp file for simple tasks, such as create layers, offset lines, draw hatch patterns or polylines and insert attributed blocks.

- **Preliminary Setup Procedures**
- **Loading the ADESK.mns Custom menu**
- **Programming Exercise**
- **Basic Tools**
- **Create Two New Layers Named Wall4 and Wallc**
- **Creating A Macro**
- **Loading Lisp**
- **Draw A Line and Add A Prompt**
- **Offset the Line 4”**
- **Changing Layer to Wallc, Ltype to Hidden2 and Ltscale to 96**
- **Changing the Last Line (The Offset Line) From a Line to a 3.5” Polyline**
- **Saving and Restoring System Variables**
- **Inserting a Symbol Using Lisp**
 - Lisp Program No. 2: Ref00.lsp
- **System Variables - Self Paced Exercises**
 - Lisp Program No. 3: Ref01.lsp
 - Lisp Program No. 4: Ref01.lsp (modified)
 - Lisp Program No. 5: Ref01.lsp (modified)
 - Lisp Program No. 6: Ref01.lsp (modified)
- **Lisp Program Variations - Self Paced Exercises**
 - Lisp Program: Ref02.lsp (Insert an elevation symbol with attributes and a line)
 - Lisp Program: Ref03.lsp (Insert a section symbol with a tail)
- **Advanced Lisp Program Variations - Self Paced Exercises**
 - Lisp Program: Ref04.lsp (Insert a detail symbol with an arrow)
 - Lisp Program: Ref05.lsp (Insert a detail symbol with an arrow and a line)
 - Lisp Program: Ref06.lsp (Insert a detail symbol with an arrow, line and tail)
 - Lisp Program: Ref07.lsp (Insert an elevation symbol with multiple arrows)

PRELIMINARY SETUP PROCEDURES

This session requires users to type programming code using an ASCII text editor, such as Notepad. To make the class more productive, the instructor has created the programming files and a menu that invoke the lisp code. Attendees can select Toolbar icons accessible from the HAJC pulldown menu for these exercises. The LISP programs and related files must be copied from a disk provided by the instructor to your hard drive. You will then be shown how to use the AutoCAD Preferences command to set paths for your work.

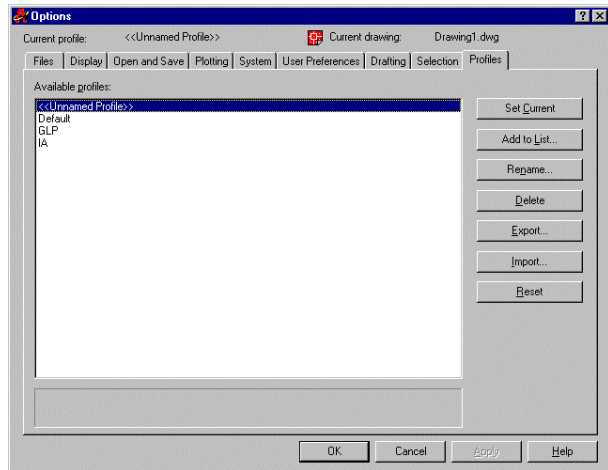
1. Copy the Library directory located on the disk to your hard drive. The disk contains four subfolders and includes a menu, lisp programming files and symbols (AutoCAD blocks).

| | |
|--|---|
| <pre> C:\Library - Bmp - Lisp - Support - Sym </pre> | <pre> (Bitmaps) (Lisp files) (for Adesk.mns menu and Acaddoc.lsp file) (Symbols) </pre> |
|--|---|

3. Open the Preferences dialog box by right clicking over the command prompt and choosing **OPTIONS...** when the cursor menu appears

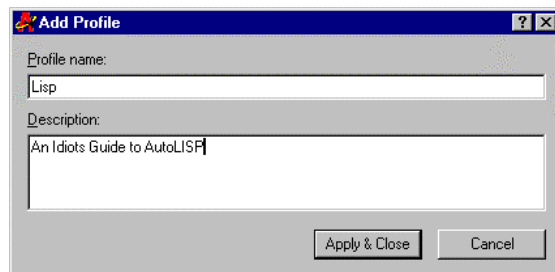


4. Pick the Profiles tab
 - Highlight <<Unnamed Profile>>.



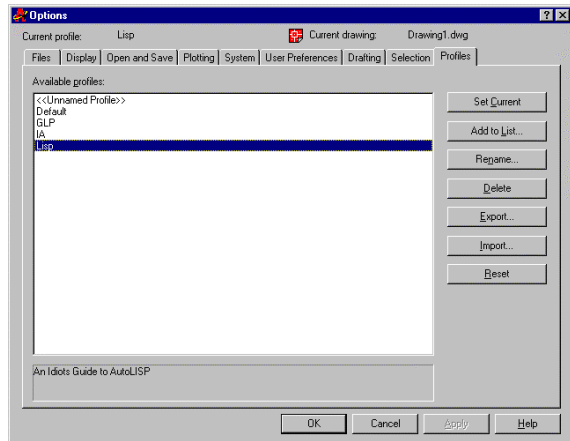
- Pick the **ADD TO LIST...** button to open the Add Profile dialog. Enter the following for each edit box, then pick **OK**.

Profile Name: **LISP**
 Description: **An Idiots Guide to AutoLisp**



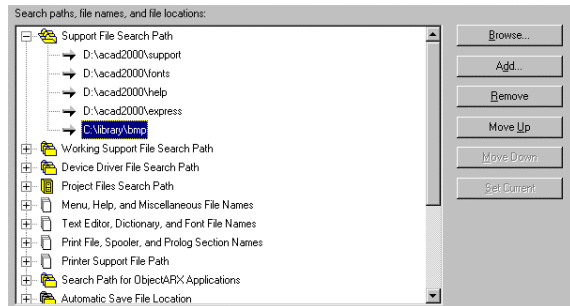
- Pick the **SET CURRENT...** button to set Lisp as the current profile.

Note: At the end of this class session, please return to the Options dialog box and set <<Unnamed Profile>> current.

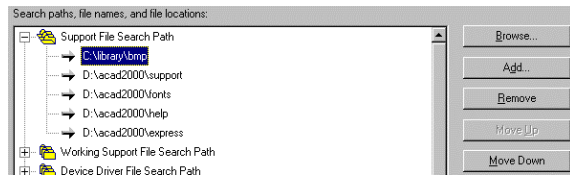


5. Pick the Files tab

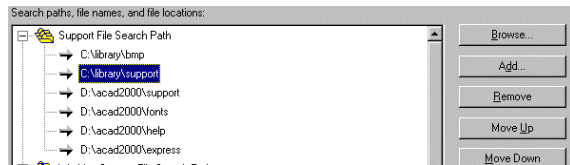
- Double click on **SUPPORT FILE SEARCH PATH** to expand this directory.
- Pick the **ADD...** button
- Type **C:\LIBRARY\BMP**, then press Enter.
- Pick the **MOVE UP** button to move **C:\LIBRARY\BMP** to the top of the directory list.



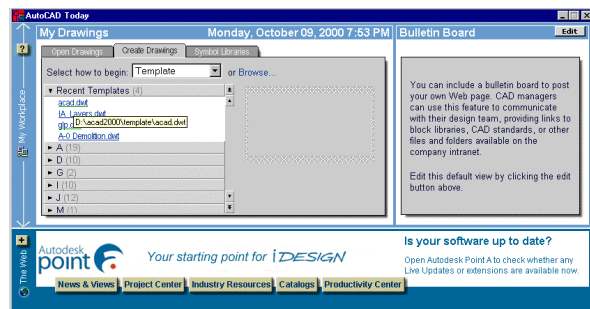
- Pick the **ADD...** button again
- Type **C:\LIBRARY\SUPPORT**, then press Enter



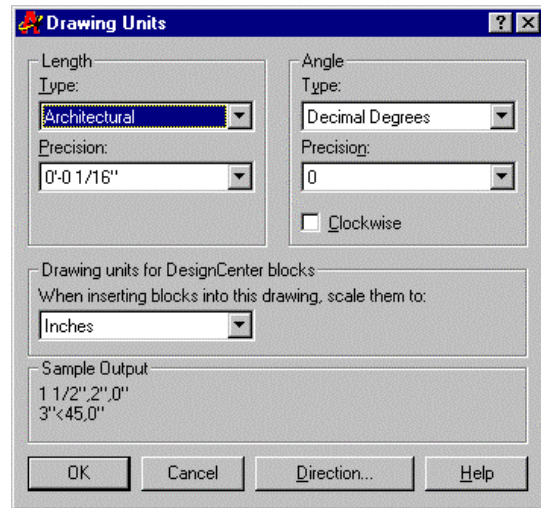
- Pick the **MOVE UP** button to move **C:\LIBRARY\SUPPORT** as the 2nd directory in the list.



- Pick **Ok** to close the dialog.
- Pick **NEW** from the File menu or enter New at the command line to open AutoCAD Today and start a new drawing so that your new paths will take effect.
- Select the **ACAD.DWT** file



- Use the **DDUNITS** command to set your Units to Architectural.



- Pick ORTHO from the status line to Set Ortho **ON**
- Use the Zoom command to set your screen display size.

Command: **zoom**

All/Center/Dynamic/Extents/Previous/Scale(X/XP)/Window/<Realtime>: **.07x**

- For the last setup procedure, enter the command or system variable Cmdecho and set it to 0 (On). This will prevent your programming code from displaying at the command prompt.

Command: **cmdecho**

New value for CMDECHO <1>: **0**

When Cmdecho is **1** (ON), a typical prompt will appear with the command sequence as part of the prompt:

Command:

Pick points to draw a wall line.line From point:

To point: 16'

To point:

When Cmdecho is **0** (OFF), the prompt will appear with the programming prompt only:

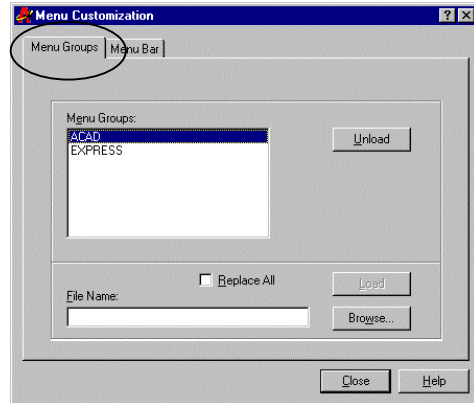
Command: (load "c:/library/lisp/dbline03")

Layer WALLC already exists.

Pick points to draw a wall line.16'

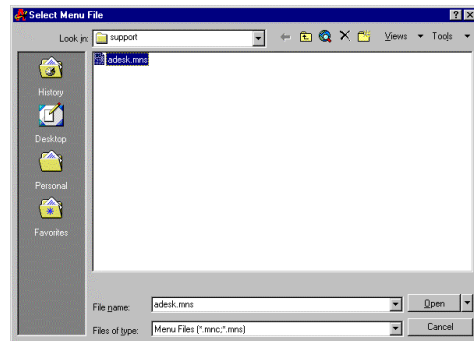
LOADING THE ADESK.MNS CUSTOM MENU

1. Enter the **MENULOAD** command to open the Menu Customization dialog box and click on the Menu Groups tab.

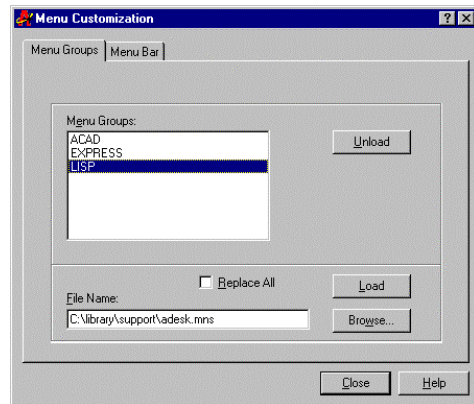


2. Pick **BROWSE...**

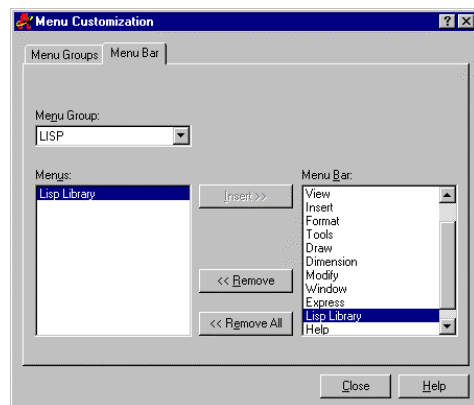
3. Locate the ADESK.MNS file in the C:\Library\Support folder, then click **OPEN**.



4. Click **LOAD** and the ADESK.mns file will display LISP in the Menu Groups list box.



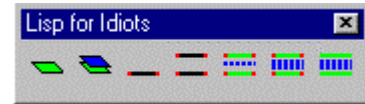
5. Click the Menu Bar tab.
6. Highlight Lisp Library in the Menus list box at the left.
7. Highlight Help in the Menu Bar list box at the right.
8. Click the **INSERT>>** button to place the Lisp Library menu in the menu bar.



8. Pick **CLOSE** to exit the dialog.

9. Three toolbars should appear on your display screen. We will begin this session using the Lisp for Idiots toolbar, so you can close the other two toolbars picking the dash mark at the upper left corner of these toolbars.

Lisp for Idiots Toolbar



Lisp for Idiots II Toolbar



Reference Toolbar



PROGRAMMING EXERCISE

A simple exercise has been created for this session that prompts you to draw a single line, then prompts you to offset it twice. The result provides you with two parallel lines drawn on layer Wallc, then one line placed in the center on layer Wall4 with linetype hidden2 and ltscale 96.

BASIC TOOLS

TOOL 1: Command - used to identify the "command" prompt line within a lisp file.

TOOL 2: Parenthesis - allows command to be typed at command line

TOOL 3: Quotation marks - allows execution of a command and command options

Apply tools 1, 2 and 3 by typing the series of commands with command options at the prompt line, separating the command name and response option within quotes "" and enclosing the entire line within parenthesis (). Quoted options can be separated by pressing the spacebar. Double quotes (") imply a return.

CREATE NEW LAYERS NAMED WALL AND WALL4 AT THE COMMAND PROMPT

1. Invoke the Layer command by typing it at the command prompt and create a layer named WALL with the color green and linetype continuous. Use the “make” option to set the layer current. After completing the command sequence, pick the Layer icon from the Object Properties Toolbar to open the Layer & Linetype Properties dialog box and view the results.

Note: To invoke the layer command at the command prompt, preface it with a hyphen.
Command: **-LAYER**

```
Command: -layer
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock: new
New layer name(s): wall
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock: color
Color: green
Layer name(s) for color 3 (green) <0>: wall
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock: ltype
Linetype (or ?) <CONTINUOUS>: ↵
Layer name(s) for linetype CONTINUOUS <0>: wall
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock: ↵
```

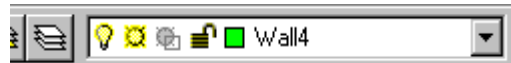
2. You can execute the same procedure by typing it as a lisp statement at the command line using the AutoCAD “command” function enclosed in parenthesis and quotes as shown. Make a layer named WALL4, color Green, linetype Continuous by typing it as shown.

(command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")

```
Command: u
LAYER GROUP
Command: (command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
```

Typing layer command sequence at command prompt

The layer name will appear in the Object Properties dialog box.



3. Enter the AutoCAD U command to undo this procedure.
4. You can type the same statement **(command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")** and save it in a Lisp text file using an .LSP extension.
5. At the AutoCAD command prompt, type Notepad and type the file **DBLINE01.LSP** from the **C:\LIBRARY\LISP** directory:

```
Command: notepad
File to edit:C:\Library\Lisp\Dbline01.lsp
```

6. We will now discover how to use Lisp and save a file in an ASCII text editor, such as Notepad, to create a Wall4 layer defined with the same color and linetype properties, as well as set it current.

- After opening the Dbline01.lsp file, you will see the same statement previously entered at the AutoCAD command prompt. Let's execute it using a toolbar icon.

```
(command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
```

Dbline01.lsp

- Select the icon 1-Make a Layer from the Lisp for Idiots Toolbar to invoke it.



- Pick the Layer icon from the Object Properties Toolbar to open the Layer & Linetype Properties dialog box and verify that the new layer WALL4 was created.

CREATING A MACRO

- You can add a function called “defun” to convert your lisp file into a macro or shortcut key and invoke it at the command line.

```
(defun c:DBLINE02 ()
  (command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
)
```

- The defun function is described as follows:

```
(defun c:DBLINE02 ()
  |      | |      | ---Defines an argument, used for advanced lisp programming
  |      | | --- Save the lisp program filename as Dbline02
  |      | | --- Assign function (name) as an AutoCAD command
  | --- Autolisp programming language meaning define the function
```

- Open the file Dbline02.lsp by double clicking on the name using the Explorer or with your Notepad entering the name and path at the **FILE TO EDIT:** prompt on the AutoCAD command line.
- An additional layer named WALLC, color Yellow, linetype Continuous to be offset as a 3.5” wide polyline centered within the 4” wide wall lines on layer Wall4 has been added. The layer command is shown using the NEW option to avoid setting the layer current.

```
(command "layer" "new" "wallc" "color" "yellow" "wallc" "ltype" "continuous" "wallc" "")
```

```
(defun c:dbline02 ()
  (command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
  (command "layer" "new" "wallc" "color" "blue" "wallc" "ltype" "continuous" "wallc" "")
)
```

Dbline02.lsp

DRAW A LINE AND ADD A PROMPT

1. Open the Dblines03.lsp file from inside AutoCAD or the Explorer.
2. The Line command has been added, followed by a function called **PAUSE** allowing you to draw a line. The command will pause, so that a user can pick first and second points or enter values at the command line, then exit the command.

```
(command "line" pause pause "")
|         |         |         | --- return to exit command
|         |         |         | --- AutoCAD "pause" function allows for user input
|         |         |         | --- AutoCAD Line command
|         |         |         | --- AutoCAD "command" function
```

If typed at the command line, the prompts to draw a horizontal 16' long line would read:

```
Command: line
From point: [pick first point (requires a pause for user input)]
To point: @16'<0 [pick second point (requires another pause for user input)]
To point: [press enter to exit command (invoked with double quote marks)]
```

Using direct distance entry to draw the line, the command sequence would be:

```
Command: line
From point: [Pick first point]
To point: 16' [With orthomode on, the distance value is entered]
To point [press enter to exit command]
```

3. A statement, **Pick points to draw a wall line.**, has also been added to prompt the user to draw the line. "Princ" prints your text or prompt on the command line. The statement is enclosed in parenthesis, includes the function "princ", and a lower case "n" with a backslash (\n). The "\n" function instructs AutoCAD to place the prompt on a new line.

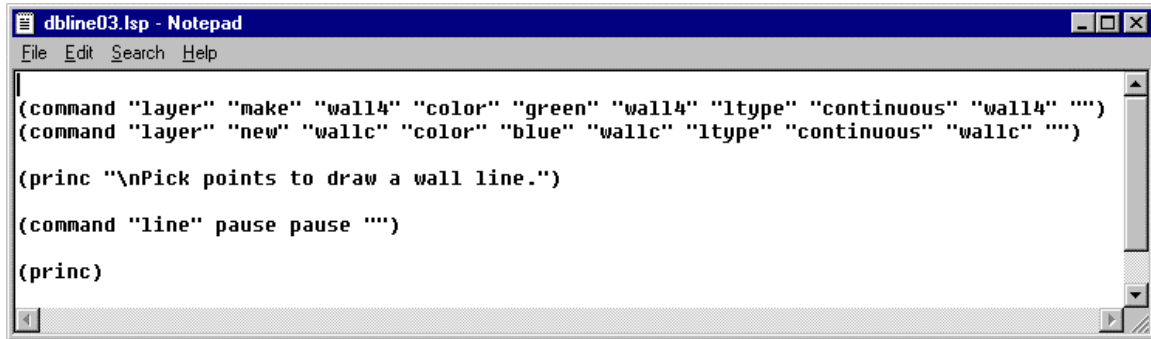
```
(princ "\nPick points to draw a wall line.")
| | | --- Prompt instructing user what to do
| | | --- Lisp function meaning "next line"
| | | --- Lisp function to display or print prompt at command line
```

Note: The statement might also read "\nPick points or enter a value:"

4. The function "princ", enclosed in parenthesis, is used to prevent "nil" from appearing at the command line. The function is placed before the ending parenthesis.

```
(princ)
```

5. The Dbline03.lsp lisp file appears as shown.



```
{command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" ""}
{command "layer" "new" "wallc" "color" "blue" "wallc" "ltype" "continuous" "wallc" ""}

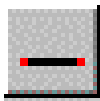
{princ "\nPick points to draw a wall line."}

{command "line" pause pause ""}

{princ}
```

Dbline03.lsp

6. Select the icon 3-Line/Prompt from the Lisp for Idiots Toolbar to invoke it.



2-LINE/PROMPT

7. Pick points and draw a horizontal, using direct distance entry, 16' long line.

Command: (load "c:/library/lisp/dbline03")

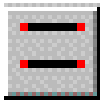
Layer WALLC already exists.

Pick points to draw a wall line.16'

4. The Dbline04.lsp lisp file now appears as shown:

Dblne04.lsp

5. Select the icon 4-Offset Line from the Lisp for Idiots Toolbar to invoke Dblne04.lsp.



4-OFFSET LINE

6. The command line reads as shown:

```
Command: (load "c:/library/lisp/dblne04")
Layer WALLC already exists.
Pick points to draw a wall line.16' [Using direct distance entry to draw 16' long line]
Pick wall line [Instructs user to pick the wall line]
Next, pick side to offset. [Instructs user to pick side to offset the wall line]
```



CHANGING THE LAYER TO WALLC, LTYPE (LINETYPE) TO HIDDEN2 AND LTSCALE TO 96

1. We will change the Offset distance to 2", then offset one of the two wall lines 2", locating it within the 4" wide wall, then change its properties. You can use the Offsetdist system variable, the Offset command or the lisp "setvar" function as shown below:

Offsetdist system variable used as a command:

```
Command: offsetdist  
New value for OFFSETDIST <0'-4">: 2
```

Offset used as a command:

```
Command: offset  
Offset distance or Through <0'-4">: 2  
Select object to offset:  
Side to offset?  
Select object to offset:
```

Using the lisp Setvar function to set the value to 2":

```
(setvar "offsetdist" 2)  
|          |          | --- The value can be entered without quotes  
|          | --- Name of system variable  
| --- Setvar command
```

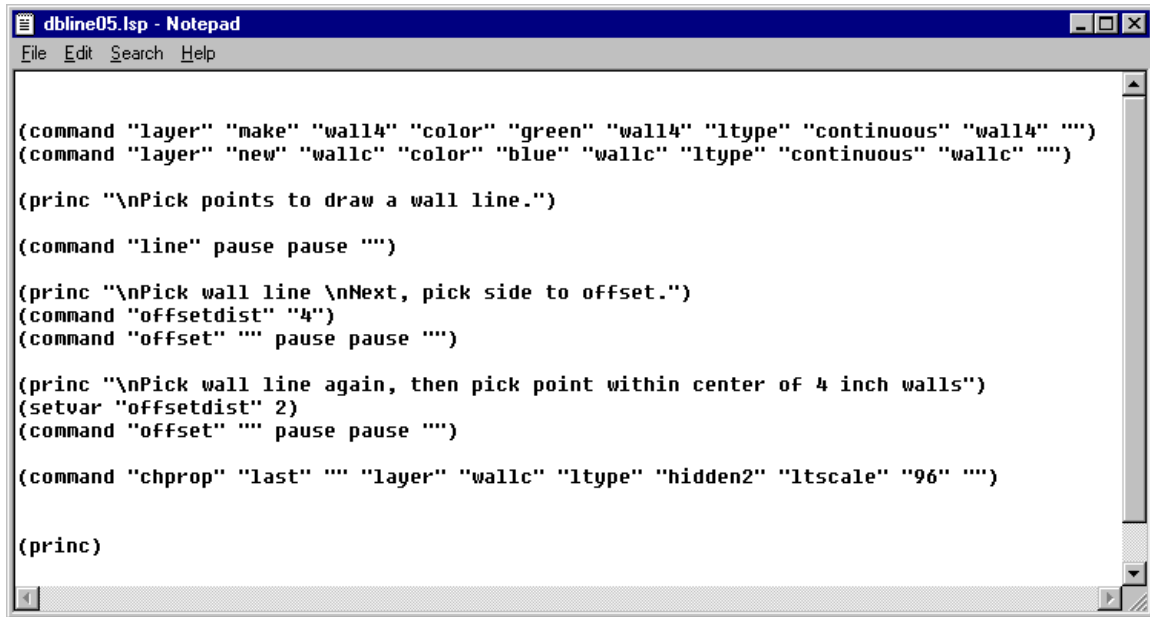
2. Use the Chprop command to change the properties of the last line drawn (the offset line). Change the line's Layer to WALLC, the Ltype to HIDDEN2, LtScale to 96. Assume that the linetype is already loaded.

```
Command: chprop  
Select objects: last 1 found  
Select objects: ↵  
Change what property (Color/LAyer/LType/lTScale/Thickness) ? layer  
New layer <0>: wallc  
Change what property (Color/LAyer/LType/lTScale/Thickness) ? ltype  
New linetype <CONTINUOUS>: hidden2  
Change what property (Color/LAyer/LType/lTScale/Thickness) ? ltScale  
New linetype scale <0'-1">: 96  
Change what property (Color/LAyer/LType/lTScale/Thickness) ? ↵
```

3. The Chprop command is added to the lisp file as shown.

```
(command "chprop" "last" "" "layer" "wallc" "ltype" "hidden2" "ltScale" "96" "")
```

4. Open the Dbline05.lsp lisp file from the C:\Library\Lisp folder and it will appear as shown:



```
(command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
(command "layer" "new" "wallc" "color" "blue" "wallc" "ltype" "continuous" "wallc" "")

(princ "\nPick points to draw a wall line.")

(command "line" pause pause "")

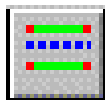
(princ "\nPick wall line \nNext, pick side to offset.")
(command "offsetdist" "4")
(command "offset" "" pause pause "")

(princ "\nPick wall line again, then pick point within center of 4 inch walls")
(setvar "offsetdist" 2)
(command "offset" "" pause pause "")

(command "chprop" "last" "" "layer" "wallc" "ltype" "hidden2" "ltscale" "96" "")

(princ)
```

5. Select the icon 5-Change Layer from the Lisp for Idiots Toolbar to invoke it.



5-CHANGE LAYER

6. The first line is drawn on layer Wall4 (the current layer), the line is offset 2", then its properties changed to layer Wallc, linetype hidden2, and ltscale 96.

Command: (load "c:/library/lisp/dblne05")

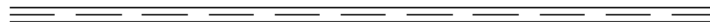
Layer WALLC already exists.

Pick points to draw a wall line. 16' [*Using direct distance entry to draw 16' long line*]

Pick wall line [*Instructs user to pick the wall line*]

Next, pick side to offset. [*Instructs user to pick side to offset the wall line*]

Pick wall line again, then pick point within center of 4 inch walls [*Pick one of the two wall lines, then pick a point inside the two parallel lines to add a third line between them*]



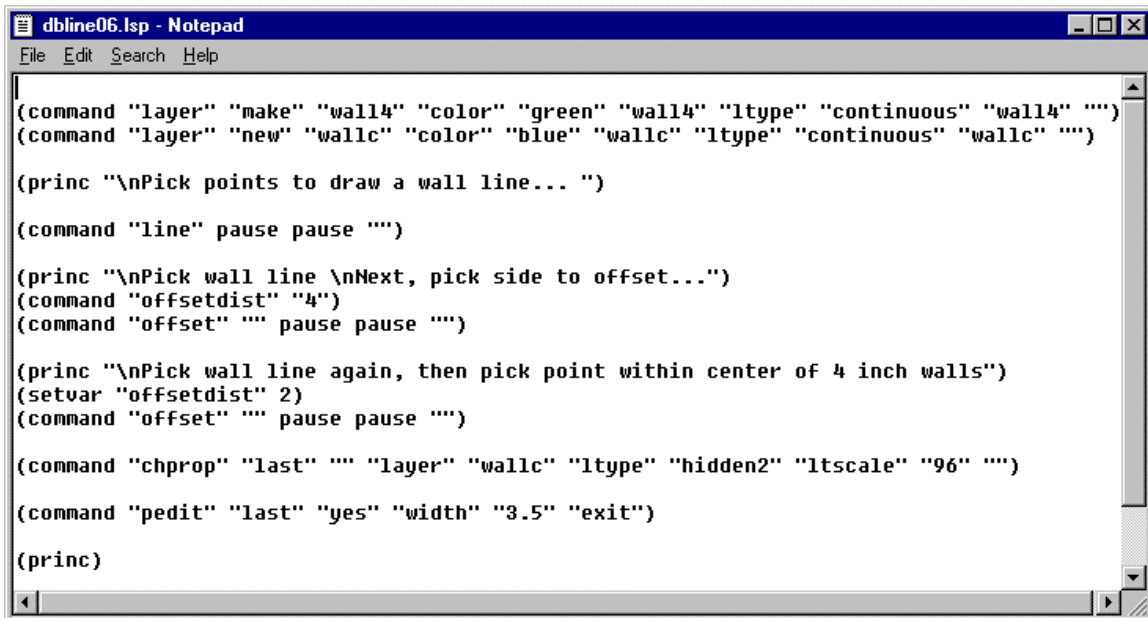
CHANGING THE LAST LINE (THE OFFSET LINE) FROM A LINE TO A 3.5" WIDE POLYLINE

1. Open the Dblines06.lsp file. The last line drawn is changed into a polyline with a width of 3.5". The command line prompts for editing a polyline are:

```
Command: pedit
Select polyline: last [last line drawn]
Object selected is not a polyline
Do you want to turn it into one? <Y> [yes]
Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo/eXit <X>: w
Enter new width for all segments: 3.5
Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo/eXit <X>:
```

The lisp programming format is shown:

```
(command "pedit" "last" "yes" "width" "3.5" "exit")
```



```
dblins06.lsp - Notepad
File Edit Search Help

|
(command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
(command "layer" "new" "wallc" "color" "blue" "wallc" "ltype" "continuous" "wallc" "")

(princ "\nPick points to draw a wall line... ")

(command "line" pause pause "")

(princ "\nPick wall line \nNext, pick side to offset...")
(command "offsetdist" "4")
(command "offset" "" pause pause "")

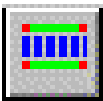
(princ "\nPick wall line again, then pick point within center of 4 inch walls")
(setvar "offsetdist" 2)
(command "offset" "" pause pause "")

(command "chprop" "last" "" "layer" "wallc" "ltype" "hidden2" "ltscale" "96" "")

(command "pedit" "last" "yes" "width" "3.5" "exit")

(princ)
```

2. Select the icon 6-Change to Pline from the Lisp for Idiots Toolbar to invoke it.



6-CHANGE TO PLINE

3. Execute the program, responding to the prompts as in the previous exercise, and this time the 2nd line being offset line will be drawn as a 3.5" wide polyline.



SAVING AND RESTORING SYSTEM VARIABLES

System variables are used to control command settings. This section explains how you can temporarily save your settings inside a lisp file during the programs execution. You will want to set various values in your lisp program while it is running and then restore their original values prior to exiting your program so that the values in your drawing session at not changed.

1. The system variable for **OFFSETDIST** changes from 4” to 2” during the execution of the lisp program. The drawings **OFFSETDIST** original value of 0 should not be effected by your program and remain unchanged in your drawing session.
2. First you must use a **setq** function to **save** the drawing’s original (current) offset distance system variable **OFFSETDIST** to a name, such as “**OLDOFFSETDIST**”. This line is placed at the beginning of the lisp program.

```
(setq oldoffsetdist (getvar "offsetdist"))
| | | | - - - System variable for offset distance
| | | | - - - Requests AutoCAD to get the variable offsetdist (AutoCAD stores the current layer name)
| | - - - The current value of offsetdist is saved to a group of letters or a name such as oldoffsetdist
| - - - This function saves the value (set quote) of the drawing current offset distance.
```

3. Next, you can set one or more values for your offset distance using a setvar command as previously described.

```
(setvar "offsetdist" 2)
| | | | - - - The value can be entered without quotes
| | | | - - - Name of system variable
| - - - Setvar command
```

4. Finally, at the end of the program, you should restore the original or “saved value” with the **setvar** function..

```
(setvar "offsetdist" oldoffsetdist)
| | | | - - - The name of the old current offsetdist system variable
| | | | - - - AutoCAD restores the offsetdist system variable to the oldoffsetdist value
| - - - AutoCAD sets the variable for the current offset distance
```

5. Open the Dbline07.lsp file to view the file with the save and restore functions added.

```
dbline07.lsp - Notepad
File Edit Search Help

(setq oldoffsetdist (getvar "offsetdist"))
(setq oldlayer (getvar "clayer"))

(command "layer" "make" "wall4" "color" "green" "wall4" "ltype" "continuous" "wall4" "")
(command "layer" "new" "wallc" "color" "blue" "wallc" "ltype" "continuous" "wallc" "")

(princ "\nPick points to draw a wall line.")

(command "line" pause pause "")

(princ "\nPick wall line \nNext, pick side to offset.")
(command "offsetdist" "4")
(command "offset" "" pause pause "")

(princ "\nPick wall line again, then pick point within center of 4 inch walls")
(setvar "offsetdist" 2)
(command "offset" "" pause pause "")

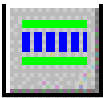
(command "chprop" "last" "" "layer" "wallc" "ltype" "hidden2" "ltscale" "96" "")

(command "pedit" "last" "yes" "width" "3.5" "exit")

(setvar "offsetdist" oldoffsetdist)
(setvar "clayer" oldlayer)
(princ)
```

6. To test the program, enter the following settings, then pick the icon 7 - Variables to load the Dbline07.lsp file.

- Set the current layer to 0.
- Enter the offsetdist command and set the value to 0.



7 - VARIABLES

7. Verify that the layer 0 remains unchanged and that the offsetdist value is set to 0.

```
Command: layer
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock: set
New current layer <WALL4>: 0
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock: ↵
```

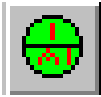
```
Command: offsetdist
New value for OFFSETDIST <0'-2">: 0
```

Now that you understand some of the basic concepts, the next series of programs will demonstrate how to insert one or more symbols or blocks into a drawing. Additional exercises explain how to add lines or polylines to draw various reference symbol types, such as sections, elevations and area references. We may not have time to complete all these exercises, so they are included as Self Paced Exercises that you may try on your own time. Use can use the Reference and Lisp for Idiots II toolbar icons in the HAJC menu to invoke them.

INSERTING A SYMBOL USING LISP - Self Paced Exercise

Class Exercise: Write a program to create a layer and insert a symbol. Save the new file name to REF01.LSP.

The program can be invoked by picking the Detail icon on the Lisp for Idiots II toolbar.



PROGRAM No.2

Lisp Program No. 2: REF01.lsp

The AutoCAD **Insert** command places a symbol in a drawing. The symbols path can be hard coded using a single forward slash / or two backward slashes \\ within the command statement:

C:/LIBRARY/LISP/REF01

Syntax description:

| | |
|---------|--------------------------|
| C: | drive letter |
| / or \\ | directory name separator |
| LIBRARY | directory name |
| LISP | subdirectory name |
| REF01 | lisp program |

Understanding the prompt lines for the Insert command

Use the **-Insert** command to place a symbol in your drawing named REFTAG from the C:/LIBRARY/LISP directory.

Command: **-insert**
Block name (or?): **reftag**
Insertion point: **[pick a point]**
X scale factor <1> / Corner / XYZ: **1**
Y scale factor (default=X): **1**
Rotation angle <0>: **0**
Enter attribute values
Detail ID (1): **1**
Detail No. (A12.10): **A12.10**



reftag symbol

You can type the statements for creating a layer and inserting a block at the command line to test them, then paste each line into the Notepad. Save the file to the name Ref01.lsp as shown.

Command: **(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")**

Command: **(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFTAG" "9,5" "1" "1" "0")**

The symbol is placed into the drawing, but the attribute dialog does not open.

SYSTEM VARIABLES - Self Paced Exercises

You now know the basics of Lisp programming. Class time will probably not allow us to complete the remaining exercises, so I have included them as Self Paced Exercises so that you can continue your Lisp training and experimentation at your leisure on your own time.

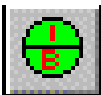
TOOL 4: System Variables - use system variables to set controls.

System variables and values that affect this specific program include:

| System variable | Description | Value |
|-----------------|---|--|
| cmdecho | Used with Autolisp to control what is displayed at the prompt line | 0=not echoed to screen; 1=echoed |
| clayer | Displays/sets current layer | |
| attmode | Controls the attribute display mode | 0=off; 1=normal;2=on |
| attreq | Controls the prompt for attributes | 0=no prompt or dialog box for attributes;1=normal prompt or dialog box upon attribute insertion |

Lisp Program No.3: REF01a.lsp

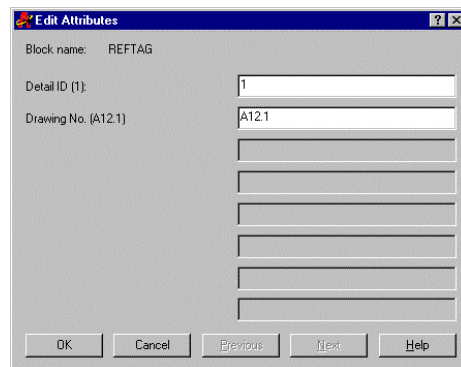
1. Modify the previous lisp program Ref01.lsp and insert a symbol into a drawing at the coordinates 9,5 on the A-tag layer with the attribute system variable Attdia set to On.
2. Save the revised lisp file to Ref01.lsp and load it using from the Reference toolbar.



PROGRAM NO. 3

The symbol is inserted into the drawing on layer a-tag and the Edit Attribute dialog box opens.

Use a semi-colon after your lisp statement to add remarks or comments.



```
(COMMAND "LAYER" "Make" "A-TAG" "Color" "Green" "A-TAG" "")
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFTAG" "9,5" "1" "1" "0")
```

Use the Ddatte command to open the Edit attributes dialog after the block is inserted. (The selection set "Last" in the program acts on the last object, which is the Reftag symbol.

```
(COMMAND "DDATTE" "LAST")
```

Lisp Program No.4: REF01b.lsp:

Modify the lisp program Ref01a.lsp to allow the user to input the symbol at a user defined location. Include a statement prompting the user to pick a pick.

The file can be invoked from the Lisp for Idiots II toolbar



PROGRAM No.4

```
(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")  
(PRINC "\nPick point for symbol.")  
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFTAG" pause "1" "1" "0")
```

- Load the lisp file from the Lisp for Idiots II toolbar.
- The symbol is inserted from the directory C:\LIBRARY\SYM at a user specified location into the drawing on layer a-tag.
- System variables for attmode and attreq should also be included in the lisp file as they effect the status of attributed symbols during insertion.
- System variables, should be saved, reset during execution of the lisp program, then restored. Including a “setq” function can save the old values and a “setvar” function can restore the old values at the start and end of a lisp program.

Tool 5: **Functions** - special symbols that can be invoked in lisp files.

We have already learned some of the functions, here are some reminders.

| Code | Description |
|---------|--|
| Command | The most general of the Autolisp functions that access AutoCAD |
| "" | Enter or return |
| pause | Pauses for user input if an Autocad command is in progress |
| ; | Allows for comments within Autolisp programs |
| \n | Newline character (lower case “n” is required) |
| princ | Displays a message at the command line |
| setq | Sets a value to a function |
| setvar | AutoCAD command |

Lisp Program No.5: REF01c.lsp:

Modify the last lisp program to save the current layer (set at 0), insert a symbol on the layer a-tag, then restore the layer (0). You can load it from the Lisp for Idiots II toolbar.

1. Use Cmdecho to prevent the lisp program from being displayed at the prompt line.
2. Allow users to select a point for the symbol and include a prompt instructing users to "Pick a point for the symbol".
3. Save and set system variables with SETQ and SETVAR statements.

```
;This section saves current system variables
(SETQ OLDCMDECHO (GETVAR "CMDECHO"))
(SETQ OLDLAYER (GETVAR "CLAYER"))
(SETQ OLDATTREQ (GETVAR "ATTREQ"))
(SETQ OLDATTMODE (GETVAR "ATTMODE"))

;This section sets system variables specific to lisp the program
(SETVAR "CMDECHO" 0)
(SETVAR "ATTREQ" 1)
(SETVAR "ATTMODE" 1)

;Issues AutoCAD command to make new layer and set it current.

(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")

;Prints a remark at the command line

(PRINC "\nPick point for symbol.")

;Issues AutoCAD command to insert a block (symbol)

(COMMAND "INSERT" "C:/LIBRARY/SYM/REFTAG" pause "1" "1" "0")

;This section restores original system variable values
(SETVAR "CLAYER" OLDLAYER)
(SETVAR "ATTREQ" OLDATTREQ)
(SETVAR "ATTMODE" OLDATTMODE)
(SETVAR "CMDECHO" OLDCMDECHO)

;Prevents display of "nil" at command line confirming command has been properly ;executed.
(PRINC)
```

Lisp Program No.6: REF01d.lsp:

This lisp program replaces **pause** with **pt1** when requesting user input. It is an alternate method to lisp program #5 and allows users to save the X and Y coordinate points in the drawing for additional block insertions at the same coordiante. You can load it from the Lisp for Idiots II toolbar.



PROGRAM NO. 6

```
(SETQ OLDCMDECHO (GETVAR "CMDECHO"))
(SETQ OLDLAYER (GETVAR "CLAYER"))
(SETQ OLDATTREQ (GETVAR "ATTREQ"))
(SETQ OLDATTMODE (GETVAR "ATTMODE"))

  (SETVAR "CMDECHO" 0)
  (SETVAR "ATTREQ" 1)
  (SETVAR "ATTMODE" 1)

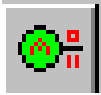
  (COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")
  (SETQ pt1 (GETPOINT "\nPick point for symbol.))
  (COMMAND "-INSERT" "C:/LIBRARY/SYM/REFTAG" pt1 "1" "1" "0")
  (COMMAND "DDATTE" "LAST")

(SETVAR "CLAYER" OLDLAYER)
(SETVAR "ATTREQ" OLDATTREQ)
(SETVAR "ATTMODE" OLDATTMODE)
(SETVAR "CMDECHO" OLDCMDECHO)

(PRINC)
```


LISP PROGRAM VARIATIONS - Self Paced Exercises

Experiment with the next two lisp programs as self paced exercises to demonstrate how to insert a symbol and draw a line on a specific layer. Ref02.lsp demonstrates the use of the PRINC statement with “pause” to instruct user input. The lisp function SETQ instructs the variable **pt1** (or Point 1) to display the prompt **Pick first point (QUADpoint of symbol)...** when a user is asked to GETPOINT. Pick the icon from the Reference toolbar.



ELEVATION WITH ATTRIBUTE/LINE

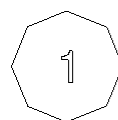
Lisp Program : REF02.lsp

Create a lisp program to insert an elevation symbol with attributes and a line.



ELEV.dwg

ELEVATION
1/4"=1'-0"



ELEV.dwg with polyline

ELEVATION
1/4"=1'-0"

```
(DEFUN C:REF02 () ;Creates program as macro or short cut key
  (SETQ OLDCMDECHO (GETVAR "CMDECHO"))
  (SETQ OLDLAYER (GETVAR "CLAYER"))
  (SETQ OLDATTREQ (GETVAR "ATTREQ"))
  (SETQ OLDATTMODE (GETVAR "ATTMODE"))
  (SETQ OLDORTHOMODE (GETVAR "ORTHOMODE"))
  (SETQ OLDOSMODE (GETVAR "OSMODE"))
  (SETQ OLDPLINEWID (GETVAR "PLINEWID")) ;Save current pline width

  (SETVAR "CMDECHO" 0) ;Set echo off
  (SETVAR "ORTHOMODE" 1) ;Set orthomode on
  (SETVAR "ATTREQ" 1) ;Control attribute prompting 1=on
  (SETVAR "ATTMODE" 1) ;Controls attribute display 1=on
  (SETVAR "OSMODE" 16) ;Sets object snap 16=quadrant
  (SETVAR "PLINEWID" 0.0125) ;Set polyline width

  (PRINC "\nPick a point for symbol...") ;Display command line prompt

  (COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")

  (COMMAND "-INSERT" "C:/LIBRARY/SYM/ELEV" pause "1" "1" "0") ;Inserts Circle block
  (COMMAND "DDATTE" "LAST") ;Open Edit Attribute dialog

  (SETQ pt1 (GETPOINT "\nPick first point (QUADpoint of symbol)...")) ;Get a point

  (SETQ pt2 (GETPOINT pt1 "\nPick second point...")) ;Continue, get a second point

  (COMMAND "PLINE" pt1 pt2 "") ;Draw a polyline from pt1 (point 1) to pt2 (point 2)

  (SETVAR "CLAYER" OLDLAYER) ;Restore all variables
  (SETVAR "PLINEWID" OLDPLINEWID) ;Restore original pline width
  (SETVAR "ATTREQ" OLDATTREQ)
  (SETVAR "ATTMODE" OLDATTMODE)
  (SETVAR "OSMODE" OLDOSMODE)
```

```
(SETVAR "ORTHOMODE" OLDORTHOMODE)  
(SETVAR "CMDECHO" OLDCMDECHO)
```

```
(PRINC) ;Prevents 0 or nil response at completion of routine  
) ;Closing paranthesis
```

Lisp Program: REF03.lsp

Create a lisp program to insert a section symbol with a tail.

Use the References toolbar to load the file.



SYMBOL WITH A LINE/TAIL

```
(DEFUN C:REF03 ()
  (SETQ OLDCMDECHO (GETVAR "CMDECHO"))
  (SETQ OLDLAYER (GETVAR "CLAYER"))
  (SETQ OLDATTREQ (GETVAR "ATTREQ"))
  (SETQ OLDATTMODE (GETVAR "ATTMODE"))
  (SETQ OLDORTHO (GETVAR "ORTHOMODE"))
  (SETQ OLDOSMODE (GETVAR "OSMODE"))
  (SETQ OLDPLINEWID (GETVAR "PLINEWID"))
```

```
(SETVAR "CMDECHO" 0)
(SETVAR "ORTHOMODE" 1)
(SETVAR "ATTREQ" 1)
(SETVAR "ATTMODE" 1)
(SETVAR "OSMODE" 16)
(SETVAR "PLINEWID" 0)
```

```
(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")
```

```
(PRINC "\nPick a point for symbol..")
```

```
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFTAG" pause "1" "1" "0")
```

```
(COMMAND "DDATTE" "LAST")
```

```
(SETQ pt1 (GETPOINT "\nPick first point (QUADpoint of symbol)."))
```

;Prompts for first point at circle quad point

```
(SETQ pt2 (GETPOINT pt1 "\nPick second point, pick third point to orient tail..."))
```

;Prompts for second point to set line length

```
(COMMAND "PLINE" pt1 pt2 "")
```

;Pick quad point at circle, then pick second point

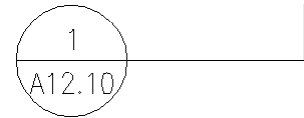
```
(SETVAR "PLINEWID" 0.06)
```

;Set pline width of .06

```
(COMMAND "PLINE" pt2 pause "")
```

;Pause for user input ;to draw tail

```
(SETVAR "CLAYER" OLDLAYER)
(SETVAR "ORTHOMODE" OLDORTHO)
(SETVAR "PLINEWID" OLDPLINEWID)
(SETVAR "ATTREQ" OLDATTREQ)
(SETVAR "ATTMODE" OLDATTMODE)
(SETVAR "OSMODE" OLDOSMODE)
(SETVAR "CMDECHO" OLDCMDECHO)
(PRINC)
)
```



REFTAG.dwg with a line and a
polyline

ADVANCED LISP PROGRAM VARIATIONS - Self Paced Exercises

The following lisp programs are variations of lisp programs Ref02 and Ref03. They are provided to encourage your continued development and understanding of the Lisp programming language.

Use the References toolbar to load it.



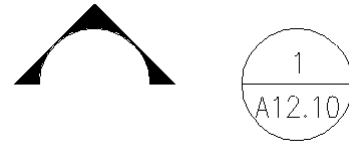
SYMBOL WITH ARROW

Lisp Program: REF04.lsp

Create a lisp program to insert a detail symbol with an arrow.

```
(DEFUN C:REF04 ()
  (SETQ OLDCMDECHO (GETVAR "CMDECHO"))
  (SETQ OLDLAYER (GETVAR "CLAYER"))
  (SETQ OLDATTREQ (GETVAR "ATTREQ"))
  (SETQ OLDATTMODE (GETVAR "ATTMODE"))
  (SETQ OLDORTHO (GETVAR "ORTHOMODE"))
```

```
(SETVAR "CMDECHO" 0)
(SETVAR "ATTREQ" 1)
(SETVAR "ATTMODE" 1)
(SETVAR "ORTHOMODE" 1)
```

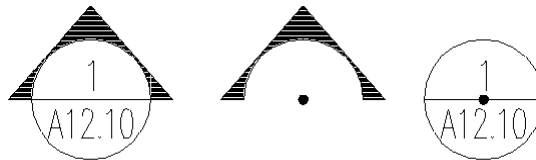


Combining Arrow.dwg and Reftag.dwg

```
(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")
(SETQ pt1 (GETPOINT "\nPick a point for symbol..."))
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFTAG" pt1 "1" "1" "0")
;Inserts circular symbol

(COMMAND "DDATTE" "LAST")
(PRINC "\nPick or enter direction to orient arrow <0>:")
(COMMAND "INSERT" "C:/LIBRARY/SYM/ARROW" pt1 "1" "1" pause)
;Inserts arrow and allows rotation
```

```
(SETVAR "CLAYER" OLDLAYER)
(SETVAR "ATTREQ" OLDATTREQ)
(SETVAR "ATTMODE" OLDATTMODE)
(SETVAR "ORTHOMODE" OLDORTHO)
(SETVAR "CMDECHO" OLDCMDECHO)
(PRINC)
)
```



Combined Arrow and Reftag symbol. Dot identifies insertion point for each symbol

Lisp Program: REF05.lsp

Create a lisp program to insert a symbol with an arrow and line.

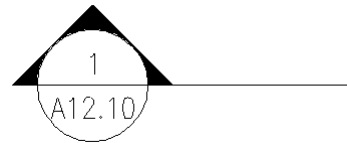
Use the References toolbar to load it.



SYMBOL WITH ARROW AND LINE

```
(DEFUN C:REF05 ()
  (SETQ OLDCMDECHO (GETVAR "CMDECHO"))
  (SETQ OLDLAYER (GETVAR "CLAYER"))
  (SETQ OLDATTREQ (GETVAR "ATTREQ"))
  (SETQ OLDATTMODE (GETVAR "ATTMODE"))
  (SETQ OLDORTHO (GETVAR "ORTHOMODE"))
  (SETQ OLDOSMODE (GETVAR "OSMODE"))
```

```
(SETVAR "CMDECHO" 0)
(SETVAR "ATTREQ" 1)
(SETVAR "ATTMODE" 1)
```



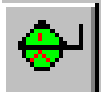
Reftag, Arrow and line

```
(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")
(SETQ pt1 (GETPOINT "\nPick a point for symbol..."))
;Prompts for location for circular symbol
(COMMAND "INSERT" "C:/LIBRARY/SYM/REFTAG" pt1 "1" "1" "0")
;Insert circular symbol
(PRINC "\nPick or enter direction to orient arrow <0>:")
;Prompts for direction of circular symbol
(SETVAR "ORTHOMODE" 1)
;Set orthomode on
(COMMAND "-INSERT" "C:/LIBRARY/SYM/ARROW" pt1 "1" "1" PAUSE)
;Insert arrow
(COMMAND "DDATTE" "LAST")
(SETVAR "OSMODE" 16)
;Set osnap to quadpoint
(SETQ pt2 (GETPOINT "\nPick QUADpoint of circle..."))
;Prompts to draw line's first point
(SETQ pt3 (GETPOINT pt2 "\nPick second point"))
;Prompts to draw line's second point
(COMMAND "LINE" pt2 pt3 "")
;Draw lines at first and second point

(SETVAR "CLAYER" OLDLAYER)
(SETVAR "ATTREQ" OLDATTREQ)
(SETVAR "ATTMODE" OLDATTMODE)
(SETVAR "ORTHOMODE" OLDORTHO)
(SETVAR "OSMODE" OLDOSMODE)
(SETVAR "CMDECHO" OLDCMDECHO)
(PRINC)
)
```

Lisp Program: REF06.lsp

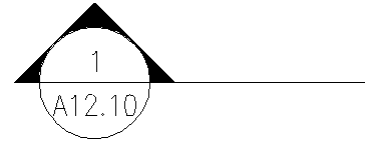
Create a lisp program to insert a detail symbol with an arrow, line and tail.



SYMBOL WITH ARROW/LINE/TAIL

```
(DEFUN C:REF06 ()  
(SETQ OLDCMDECHO (GETVAR "CMDECHO"))  
(SETQ OLDLAYER (GETVAR "CLAYER"))  
(SETQ OLDATTREQ (GETVAR "ATTREQ"))  
(SETQ OLDATTMODE (GETVAR "ATTMODE"))  
(SETQ OLDORTHO (GETVAR "ORTHOMODE"))  
(SETQ OLDOSMODE (GETVAR "OSMODE"))  
(SETQ OLDPLINEWID (GETVAR "PLINEWID"))
```

```
(SETVAR "CMDECHO" 0)  
(SETVAR "ATTREQ" 1)  
(SETVAR "ATTMODE" 1)  
(SETVAR "PLINEWID" 0)
```



Reftag, Arrow, Line and Polyline

```
(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")  
(SETQ pt1 (GETPOINT "\nPick a point for symbol..."))  
(COMMAND "INSERT" "C:/LIBRARY/SYM/REFTAG" pt1 "1" "1" "0")  
;Insert circular symbol  
(PRINC "\nPick or enter direction to orient arrow <0>:")  
;Prompts for arrow orientation  
(SETVAR "ORTHOMODE" 1)  
(COMMAND "-INSERT" "C:/LIBRARY/SYM/ARROW" pt1 "1" "1" PAUSE)  
;Insert arrow  
(COMMAND "DDATTE" "LAST")  
(SETVAR "OSMODE" 16)  
(SETQ pt2 (GETPOINT "\nPick QUADpoint of symbol circle..."))  
(SETQ pt3 (GETPOINT pt2 "\nPick second point OR enter direction <0>:\nPick next point,  
then pick to orient tail"))  
(COMMAND "PLINE" pt2 pt3 "w" "0.06" "0.06" pt3 pause "")  
  
(SETVAR "CLAYER" OLDLAYER)  
(SETVAR "ATTREQ" OLDATTREQ)  
(SETVAR "ATTMODE" OLDATTMODE)  
(SETVAR "ORTHOMODE" OLDORTHO)  
(SETVAR "OSMODE" OLDOSMODE)  
(SETVAR "PLINEWID" OLDPLINEWID)  
(SETVAR "CMDECHO" OLDCMDECHO)  
(PRINC)  
)
```

Lisp Program: REF07.lsp

Create a lisp program to insert an elevation symbol with multiple arrows.

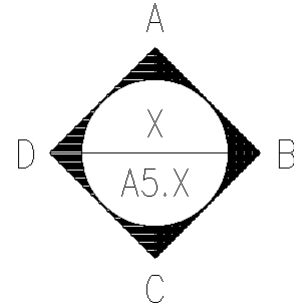


SYMBOL WITH MULTIPLE ARROWS

```
(DEFUN C:REF07 ()
  (SETQ OLDCMDECHO (GETVAR "CMDECHO"))
  (SETQ OLDCLAYER (GETVAR "CLAYER"))
  (SETQ OLDATTREQ (GETVAR "ATTREQ"))
  (SETQ OLDATTMODE (GETVAR "ATTMODE"))
  (SETQ OLDORTHOMODE (GETVAR "ORTHOMODE"))

  (SETVAR "CMDECHO" 0)
  (SETVAR "ATTREQ" 1)
  (SETVAR "ATTMODE" 1)
  (SETVAR "ORTHOMODE" 1)
  ;Allow for a Yes/No answer
  (COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG"
    "")
  (SETQ pt1 (GETPOINT "\nPlace symbol,\nthen pick or enter angle to orient arrow... "))
  (COMMAND "INSERT" "C:/LIBRARY/SYM/REFCIR" pt1 "1" "1" "0")
    ;Insert symbol
  (SETQ ans "Yes") ;Set Yes to a variable named "ans"
  (WHILE (= ans "Yes") ;If answer is Yes, continue inserting REFCIR
    (COMMAND "-INSERT" "C:/LIBRARY/SYM/REFARROW" pt1 "1" "1" "0" PAUSE)
    ;Insert arrow symbol REFARROW
  (COMMAND "DDATTE" "LAST")
  (INITGET "Yes No") ;The "initget" function initializes all valid values that can be answered
    to the "getkword" function that follows. The keyword function in this example is
    set to "Yes No"
  (SETQ ans "No") ;When the answer is No, stop insertion of symbol of arrow
    REFARROW
  (SETQ ans (GETKWORD "\nTo continue enter [Y]es, press RETURN, then enter angle or
    pick orientation (RETURN when done)..."))

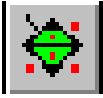
    ;"getkword" requests a predefined keyword from the ;user
  )
  (SETVAR "CLAYER" OLDCLAYER)
  (SETVAR "ATTREQ" OLDATTREQ)
  (SETVAR "ATTMODE" OLDATTMODE)
  (SETVAR "ORTHOMODE" OLDORTHOMODE)
  (SETVAR "CMDECHO" OLDCMDECHO)
  (PRINC)
  )
```



Refcir and Refarrow

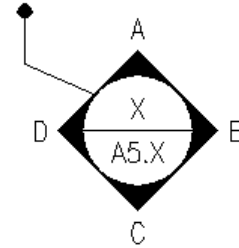
Lisp Program: REF08.lsp

Create a lisp program to insert an elevation symbol with multiple arrows and leader line.



SYMBOL WITH MULTIPLE ARROWS/LEADER

```
(DEFUN C:REF08 ()
  (SETQ OLDCMDECHO (GETVAR "CMDECHO"))
  (SETQ OLDLAYER (GETVAR "CLAYER"))
  (SETQ OLDATTREQ (GETVAR "ATTREQ"))
  (SETQ OLDATTMODE (GETVAR "ATTMODE"))
  (SETQ OLDOSMODE (GETVAR "OSMODE"))
  (SETQ OLDORTHO (GETVAR "ORTHOMODE"))
  (SETQ OLDDIM (GETVAR "DIMSCALE"))
  ;Save current Dimscale value
  (SETVAR "CMDECHO" 1)
  (SETVAR "ATTREQ" 1)
  (SETVAR "ATTMODE" 1)
  (SETVAR "ORTHOMODE" 1)
  (SETVAR "OSMODE" 0)
  (SETVAR "DIMSCALE" 1) ;Set Dimscale value to 1
```



Refcir with arrow and leader

```
(COMMAND "LAYER" "M" "A-TAG" "C" "G" "A-TAG" "")
```

```
(SETQ pt1 (GETPOINT "\nPick point for detail symbol...\nPick direction to orient arrow <0>..."))
```

```
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFCIR" pt1 "1" "1" "0")
```

```
(COMMAND "DDATTE" "LAST")
```

Note: A block can be made scaleable by replacing the "1" "1" values, which prompt for the X and Y scale factor with (GETVAR "DIMSCLALE")

Command: insert
Block name (or ?) <REFCIR>: C:\LIBRARY\SYM\REFCIR
Insertion point: X scale factor <1> / Corner / XYZ: 1 [Use (getvar "dimscale").]
Y scale factor (default=X): 1 [Enter """" (double quotes) for a return]
Rotation angle <0>:

For example:

```
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFCIR" pt1 (getvar "dimscale") (getvar "dimscale") "0")
```

or

```
(COMMAND "-INSERT" "C:/LIBRARY/SYM/REFCIR" pt1 (getvar "dimscale") "" "0")
```



```

(SETQ ans "Yes")
(WHILE (= ans "Yes") ;Include a space after the equal sign (=)
  (COMMAND "-INSERT" "C:/LIBRARY/SYM/REFARROW" pt1 "1" "1" PAUSE)
  (INITGET "Yes No")
  (SETQ ans "No")
  (SETQ ans (GETKEYWORD "\nPick or enter direction to orient arrow... \nEnter
[Y]es or [N]o to insert another arrow (Press RETURN when done)<0>:"))
)

```

```

(SETVAR "ORTHOMODE" 1)
(COMMAND "LAYER" "M" "LDR" "C" "7" "LDR" "")
(SETQ p1 (GETPOINT "\nStart point of leadered dot. "))
(SETQ p2 (GETPOINT p1 "\nSelect second point, then place tag: "))
(COMMAND "DONUT" "0" (* 0.0625(GETVAR "DIMSCALE"))) p1 ""

```

Note: Use the dimscale variable with a multiplier. This will allow you to resize the ;dot by a specific value in relation to your dimscale value. The * (asterisk) is a ;multiplier and the 3 is the number it multiplies by the dimscale value.

```

(COMMAND "LINE" p1 p2 "")

```

```

(SETVAR "ORTHOMODE" 0)
(SETVAR "OSMODE" 17)

```

```

(SETQ p3 (GETPOINT p2 "\nPick point along circle"))
(COMMAND "LINE" p2 p3 "")
(SETVAR "CLAYER" OLDCLAYER)
(SETVAR "ATTREQ" OLDATTREQ)
(SETVAR "ATTMODE" OLDATTMODE)
(SETVAR "OSMODE" OLDDOSMODE)
(SETVAR "ORTHOMODE" OLDORTHOMODE)
(SETVAR "DIMSCALE" OLDDDIM) ;Restore dimscale value
(SETVAR "CMDECHO" OLDCMDECHO)
(PRINC)
)

```