TurboCAD's Script Language

You can use TurboCAD's script language to automate many of your drawing tasks in TurboCAD. The script language employs simple commands that activate existing tools and commands in TurboCAD. For example, you can write a script that automatically draws a rectangle using the X, Y Coordinates Snap mode. It would look something like this:

```
SetSnapsScreenPos
DrawLineRectangle
ClickAt(1,1)
ClickAt(10,8)
```

As you use TurboCAD and find yourself performing the same task over and over again, you'll probably find it useful to write a script that makes the task automatic. Rather than repeat a series of steps, write a script that performs these steps at once. Then run the script using the File | Run Script command.

* **Important: TurboCAD for Windows and TurboCAD for DOS script files** are not compatible. Script files created for TurboCAD for Windows 1.x are not compatible with the current version.

* To write a script use a text editor; to run a script in TurboCAD, use the **File | Run Script command.**

Principles of creating script files

Script files are ASCII text files that you can create with your favorite text-editor. Here are some guidelines you need to follow when creating script files.

* To create the proper format, type a single command per line. All lines should be left-justified. Begin each line with the desired command, do not place any other characters at the start of a line, nor should you end a line with any punctuation marks. The commands are not case-sensitive, you can type all upper or lowercase.

* Script files must be saved as text-only files with the extension **.TWS**.

* The commands entered into script files should follow the same order that would be used during interactive use of TurboCAD. If you look at the earlier example you'll notice that a Snap mode was activated first (SetSnapsScreenPos), then a drawing tool (DrawLineRectangle), and finally the anchor and endpoints for the rectangle were identified.

* Most of the script commands require a drawing to be open when the script is run.

* Many commands require CLICK, CLICKAT, or INPUT commands to supply the actual points and/or values. The draw commands merely activate a drawing tool.

* Some commands require selection of entities. Entity selection is indicated by the [selection] indicator in the command definitions. Entities may be selected with the following commands (see each command later in this appendix for details):

```
Click
ClickAt
SelectAll
SelectByLayer
SelectByType
SelectNext
SelectNextAt
SelectWithWin
SelectNextWithWin
```

* Remember to set snap modes and defaults prior to construction of entities, as in the earlier example.

* Any errors in selection of entities or points (i.e., failure to select an appropriate entity) during script execution may result in incorrect operation of the remainder of the script, although in most cases the script will simply terminate. Pressing **<Esc>** or clicking the right mouse button during an INPUT command will terminate the script.

* All cursor positioning (and consequently drawing) must be within the visible portion of the drawing (the cursor can't be positioned outside the visible drawing area).

Declare Statements and Assign Statements

* If you will be creating and using variables in your scripting files, these variables must be identified at the beginning of the script. Use Declare statements to identify any variable you will be using. Use Assign statements to give a value (numeric or text) to those variables.

Comments in script files

Lines beginning with semi-colon ";" or blank lines are ignored as comments. All other lines must conform to script format.

Jump labels in script files

Lines beginning with a colon (:) define jump labels for use with the JUMP or JUMPIF commands. These labels are what TurboCAD searches for in a Jump or JumpIf command.

Example:
:Message Adios
Message(Time to take a break!,0,5000)
GetMouseClick
SaveDrawing(current.tcw)
Exit

Script file commands

* **Notes**: The [selection] indicator that appears in a number of the following descriptions is not a script command. It merely indicates entities must be selected prior to using the script command.

* Italicized words denote Variable Names. If a variable name is preceded by a dollar sign($), you must assign a name to this new variable that TurboCAD will use once the calculations are completed.

AddNums(*Num-1,Num-2,*$*Results*)
Add two variables together and store in a third variable

AlignLeft
Align all selected items to the left of their selection rectangle
Example: [selection]
AlignLeft

AlignTop
Align all selected items to the top of their selection rectangle
Example: [selection]
AlignTop

AlignRight
Align all selected items to the right of their selection rectangle
Example: [selection]
AlignRight

AlignBottom
Align all selected items to the bottom of their selection rectangle
Example: [selection]
AlignBottom

AlignMiddle
Align all selected items to the middle (vertically) of their selection rectangle
Example: [selection]
AlignMiddle

AlignCenter
Align all selected items to the center (horizontally) of their selection rectangle

Example: [selection]
AlignCenter

AssignNum(Value, $*Variable Name*)Assign a number to a variable

AssignStr(Value, $*Variable Name*)Assign a string to a variable

Bell Rings bell.

BringFront Bring selected item to the front of the entity stack
Example: [selection]
BringFront

CenterCursor Centers the cursor in drawing window. Useful in zoom & pan commands.

CenterSel(X,Y) Center selected items about a point
Example: [selection]
CenterSel(5,5)

Choose(Choice 1, Choice 2, ...Choice N)

Get a choice from user. Choices will be displayed in a Script Choice dialog box. Sets a value of 1..number of choices to be later used with JumpIfChoice command. Up to 10 choice strings of up to 25 characters may be entered.

Example:

```
:ChooseJumpChoose(Script 1, Script 2, Stop Script)
JumpIfChoice(1, Jump1)
JumpIfChoice(2, Jump2)
JumpIfChoice(3,Jump3)
Jump(ChooseJump)
:Jump1
ExecScript(SCRIPT1.TWS)
:Jump2
ExecScript(SCRIPT2.TWS)
:Jump3
Stop
```

Click Simulates left mouse button click at current cursor location. Used when a construction point is required during entity creation and the cursor is already positioned where desired. Also used to select entities for some commands.

ClickAt(x,y) Simulates left mouse button click at x,y in current drawing units. Used whenever a construction point is required during entity creation. Also used to select entities for some commands.
Example: ClickAt(5,5)

CopyArray(X scaling factor, Y scaling factor, rotation angle, number of columns, number of rows)

Copies marked entities in an array.

Required: [selection]
Reference point
Destination point

Example: [selection]
CopyArray(2,2,90,10,10)
ClickAt(5,5)
ClickAt(6,6)

CopyLinear(X scaling factor, Y scaling factor, rotation angle, number of copies)
Copies marked entities linearly.
Required: [selection]
Reference point
Destination point
Example: [selection]
CopyLinear(0.5,0.5,45,8)
ClickAt(5,5)
ClickAt(6,6)

CopyRadial(X scaling factor, Y scaling factor, number of copies, rotation angle, step angle)
Copies marked entities radially.
Required: [selection]
Reference point
Centerpoint
Example: [selection]
CopyRadial(1,1,12,0,30)
ClickAt(5,5)
ClickAt(6,6)

DeclareNum($*Variable Name*) Declare a numeric variable. MUST BE AT THE BEGINNING OF THE SCRIPT!

DeclareStr($Variable Name) Declare a string (alpha) variable. MUST BE AT THE BEGINNING OF THE SCRIPT!

DeleteAll Deletes all entities. DOES NOT PROMPT FIRST!

DeleteMarked Deletes marked entities. Must be in Selection Mode.
Required: [selection]

DeselectAll Deselects all entities.

DivNums(*Num-1*,*Num-2*,$*Results*) Divide two variables and store the result in a third variable.

DoubleClick Simulates left mouse button double-click at current cursor location. Used to finish certain drawing commands.

DoubleClickAt(x,y) Simulates left mouse button double-click at x,y in current drawing units. Used to complete some drawing commands.
Example: DoubleClickAt(5,5)

DrawArcArcTan(radius of tangent arc, start angle, end angle) Draws an arc tangent to an arc.
Required: An arc to be tangent to
Tangent point for arc

Example: DrawArcArcTan(2.5,0,215)
ClickAt(5,5)
ClickAt(6,6)

DrawArcCenterPt(start angle, end angle) Draws an arc using centerpoint and point on radius.
Required: Centerpoint of the arc
Point on radius
Example: DrawArcCenterPt(0,270)
ClickAt(5,5)
ClickAt(6,5)

DrawArcConcentric(start angle, end angle) Draws concentric arcs
Required: Centerpoint
First Point on radius
Second Point on radius
Nth Point on Radius
Example: DrawArcConcentric(0,180)
ClickAt(5,5)
DoubleClickAt(6,6)

DrawArcDoublePt(relative start angle, relative end angle) Draws an arc using two points.
Required: First point
Second point
Example: DrawArcDoublePt(0,360)
ClickAt(5,5)
ClickAt(7,5)

DrawArcEllipse(relative start angle, relative end angle) Draws an ellipse.
Required: Center Point of the Ellipse
Major axis point
Minor axis point
Example: DrawArcEllipse(0,360)
ClickAt(5,5)
ClickAt(7,5)
ClickAt(5,6)

DrawArcLineTan(radius of tangent arc, start angle, end angle) Draws an arc tangent to a line.
Required: A line to be tangent to
Tangent point on line
Example: DrawArcLineTan(1,0,180)
ClickAt(5,5)
ClickAt(5,5)

DrawArcTriplePt(start angle, end angle) Draws an arc using three points.
Required: First point
Second point

Third point
Example: DrawArcTriplePt(90,315)
ClickAt(5,5)
ClickAt(6,6)
ClickAt(7,5)

DrawArcTan3Arcs(start angle, end angle) Draws an arc tangent to three existing arcs.
Required: First Arc to be tangent to
Second Arc to be tangent to
Third Arc to be tangent to
Example: DrawArcTan3Arcs(0,270)
ClickAt(3,4)
ClickAt(4,5)
ClickAt(3,6)
DrawBitmap(name of bitmap file) Inserts a bitmap file into your drawing.
Required: Origin point
Example: DrawBitmap(C:\GARDEN.BMP)
ClickAt(3,3)

DrawCircleCenterPt Draws a circle using centerpoint and radius point.
Required: CenterpointPoint on radius
Example: DrawCircleCenterPt
ClickAt(5,5)
ClickAt(7,5)

DrawCircleCircleTan(radius of tangent circle) Draws a circle tangent to an arc.
Required: An arc to be tangent to
Tangent point for circle
Example: DrawCircleCircleTan(1)
ClickAt(5,5)
ClickAt(6,6)

DrawCircleConcentric Draws concentric circles.
Required: Centerpoint
First Point on radius
Second Point on radius
Nth Point on radius
Example: DrawCircleConcentric
ClickAt(5,5)
ClickAt(6,6)
DoubleClickAt(7,7)

DrawCircleDoublePt Draws a circle using two points.
Required: First point
Second point
Example: DrawCircleDoublePt

ClickAt(5,5)
ClickAt(7,5)

DrawCircleLineTan(radius of tangent circle) Draws a circle tangent to a line.
Required: A line to be tangent to
Tangent point on line
Example: DrawCircleLineTan(2)
ClickAt(5,5)
ClickAt(5,5)

DrawCircleTan3Arcs Draws a circle tangent to three existing arcs.
Required: First arc to be tangent to
Second arc to be tangent to
Third arc to be tangent to
Example: DrawCircleTan3Arcs
ClickAt(3,4)
ClickAt(3,6)
ClickAt(5,5)

DrawCircleTriplePt Draws a circle using three points.
Required: First point
Second point
Third point
Example: DrawCircleTriplePt
ClickAt(5,5)
ClickAt(6,6)
ClickAt(7,5)

DrawCurveBezier Draws a Bezier curve
Required: Start point
Next curve point
Next curve point(until DoubleClick)
Example: DrawCurveBezier
ClickAt(1,1)
ClickAt(3,2)
ClickAt(5,3)
DoubleClickAt(4,8)

DrawCurveSpline Draws a Spline curve.
Required: Start point
Next curve point
Next curve point(until DoubleClick)
Example: DrawCurveSpline
ClickAt(1,1)
ClickAt(3,2)
ClickAt(5,3)

```
                                DoubleClickAt(4,8)

DrawDimAngular        Draws an angular dimension.
                      Required:     Base point
                                    Reference point
                                    Position for the dimension
                      Example:      DrawDimAngular
                                    ClickAt(3,3)
                                    ClickAt(4,5)
                                    ClickAt(5,7)

DrawDimArrow          Draws an arrow dimension.
                      Required:     Arrow tip point
                                    Point opposite tip
                      Example:      DrawDimArrow
                                    ClickAt(5,5)
                                    ClickAt(8,8)

DrawDimDiameter       Draws a diameter dimension for an arc or circle.
                      Required:     [selection]
                                    Point of label
                      Example:      DrawDimDiameter
                                    [selection]
                                    ClickAt(5,5)

DrawDimHorizontal     Draws a horizontal dimension.
                      Required:     If Auto Dimension On:
                                              A line
                                              Position of dimension text
                                    If Auto Dimensions Off:
                                              Point 1
                                              Point 2
                                              Position of dimension text
                      Example:      If Auto Dimension On:
                                              DrawDimHorizontal
                                              ClickAt(5,5)
                                              ClickAt(5,3)
                                    If Auto Dimension Off:
                                              DrawDimHorizontal
                                              ClickAt(3,5)
                                              ClickAt(7,5)
                                              ClickAt(5,3)

DrawDimParallel       Draws a parallel dimension.
                      Required:     If Auto Dimension On:
                                              A line
```

Position of dimension text
If Auto Dimension Off:
Point 1
Point 2
Position of dimension text
Example: If Auto Dimension On:
DrawDimParallel
ClickAt(5,5)
ClickAt(5,3)
If Auto Dimension Off:
DrawDimParallel
ClickAt(3,5)
ClickAt(7,7)
ClickAt(6,4)

DrawDimRadius Draws a radius dimension for an arc or circle.
Required: [selection]
Position of dimension text
Example: DrawDimRadius
ClickAt(5,5)
ClickAt(7,7)

DrawDimVertical Draws a vertical dimension.
Required: If Auto Dimension On:
A line
Position of dimension text
If Auto Dimensions Off:
Point 1
Point 2
Position of dimension text
Example: If Auto Dimension On:
DrawDimVertical
ClickAt(5,5)
ClickAt(3,5)
If Auto Dimension Off:
DrawDimVertical
ClickAt(3,3)
ClickAt(3,7)
ClickAt(5,5)

DrawDLineDropLine Draw perpendicular double line from point to line.
Required: Selected line
Endpoint for double line
Example: DrawDLineDropLine
ClickAt(5,4)
ClickAt(5,7)

DrawDLineMulti Draws multiple connected double lines.
Required: Start point
Next endpoint
Next endpoint(until DoubleClick)
Example: DrawDLineMulti
ClickAt(1,1)
ClickAt(10,1)
ClickAt(10,8)
ClickAt(1,8)
DoubleClickAt(1,1)

DrawDLineParallel(parallel double line distance) Draws a double line parallel to another line
Required: Select a line to be parallel to
Example: DrawDLineParallel(1)
ClickAt(5,5)

DrawDLinePolygon(number of sides) Draws an n-sided double line polygon.
Required: Centerpoint
Radial point
Example: DrawDLinePolygon(3)
ClickAt(5,5)
ClickAt(5,7)

DrawDLineRectangle Draws a double line rectangle.
Required: First corner
Opposite corner
Example: DrawDLineRectangle
ClickAt(1,8)
ClickAt(10,1)

DrawDLineSingle Draws a single 2 point double line.
Required: Start point
Endpoint
Example: DrawDLineSingle
ClickAt(1,1)
ClickAt(5,5)

DrawHatch Draw a hatch.
Required: Successive points to define boundary. The last point should be equal to the first.
If hatching an arc, use OnArcPt snap mode and just select arc.
Example: DrawHatch
ClickAt(1,1)
ClickAt(3,1)
ClickAt(3,3)
ClickAt(1,3)

ClickAt(1,1)

DrawLineArcPtTan Draws a line tangent to an arc.
Required: Origin point for line
tangent point on an arc
Example: DrawLineArcPtTan
ClickAt(5,5)
ClickAt(2,2)

DrawLineDropLine Draws perpendicular line from point to line.
Required: A line
Endpoint for line
Example: DrawLineDropLine
ClickAt(5,4)
ClickAt(5,7)

DrawLineLine2Arcs Draws a line tangent to two arcs.
Required: First Arc (line’s origin)
Second Arc (line’s endpoint)
Example: DrawLineLine2Arcs
ClickAt(3,3)
ClickAt(7,7)

DrawLineMulti Draws multiple connected lines.
Required: Start point
Next endpoint
Next endpoint ..... (until DoubleClick)
Example: DrawLineMulti
ClickAt(1,1)
ClickAt(10,1)
ClickAt(10,8)
ClickAt(1,8)
DoubleClickAt(1,1)

DrawLineParallel(parallel line offset distance)Draws a line parallel to another line.
Required: A line to be parallel to
Example: DrawLineParallel(1)
ClickAt(5,5)

DrawLinePolygon(number of sides)Draws an n-sided polygon.
Required: Centerpoint
Radial point
Example: DrawLinePolygon(3)
ClickAt(5,5)
ClickAt(5,7)

DrawLineRectangle Draws a rectangle.
Required: First corner
Opposite corner
Example: DrawLineRectangle
ClickAt(1,8)
ClickAt(10,1)

DrawLineSingle Draws a single 2 point line.
Required: Start point
Endpoint
Example: DrawLineSingle
ClickAt(1,1)
ClickAt(5,5)

DrawLineTan(length of line)Draws a line tangent to an arc.
Required: An arc to be tangent to
A tangent point on the arc
Example: DrawLineTan(3)
ClickAt(5,5)
ClickAt(5,7)

DrawMode Set Drawing Mode On, this equivalent to choosing Draw.

DrawPointCross Draws a cross point.
Required: Location for point
Example: DrawPointCross
ClickAt(5,5)

DrawPointDot Draws a dot point.
Required: Location for point
Example: DrawPointDot
ClickAt(5,5)

DrawPointSquare Draws a square point.
Required: Location for point
Example: DrawPointSquare
ClickAt(5,5)

DrawPointStar Draws a star point.
Required: Location for point
Example: DrawPointStar
ClickAt(5,5)

DrawText(text contents) Draws text.
Required: Location for text
Example: DrawText(Some Text)

ClickAt(5,5)

EditColor(Color#) Change the color of selected entities.
Required: [selection]
Example: [selection]
EditColor(6)

EditDLinePipe Pipe the first double line into the second double -”clean” an intersection.
Required: [1st double line selection]
[2nd double line selection]
Example: EditDLinePipe
ClickAt(5,5)
ClickAt(6,6)

EditDLineUnpipe Unpipe the first double line from the second double line - “close” an intersection.
Required: [1st double line selection]
[2nd double line selection]
Example: EditDLineUnpipe
ClickAt(5,5)
ClickAt(6,6)

EditDLineChamfDD(Distance,Distance) Creates a double line chamfer using distance and distance.
Required: [1st double line selection]
[2nd double line selection]
Chamfer position
Example: EditDLineChamfDD(1,1)
ClickAt(5,5)
ClickAt(7,7)
ClickAt(6,6)

EditDLineChamfDA(Distance,Angle) Creates a double line chamfer using distance and angle.
Required: [1st double line selection]
[2nd double line selection]
Chamfer position
Example: EditDLineChamfDA(1,45)
ClickAt(5,5)
ClickAt(7,7)
ClickAt(6,6)

EditDLineChamfLA(Length,Angle)Creates a double line chamfer using length and angle.
Required: [1st double line selection]
[2nd double line selection]
Chamfer position
Example: EditDLineChamfLA(1,45)
ClickAt(5,5)
ClickAt(7,7)

ClickAt(6,6)

EditFill(fill#, color#)
Change the fill pattern for selected entities.
Required: [selection]
Example: [selection]
EditFill(10,6)

Fill:
0 - Solid
1 - Invisible
2 - Vasterisks
3 - Vbrick
4 - VCeilingGrid
5 - VCheckerPlate
6 - Vconcrete
7 - Vearth
8 - Vherringbone
9 - VHoneyComb
10 - Vsteel
11 - VTriangleGrid
12 - Tcw10Hatch
13 - Ten
14 - Fifteen
15 - Thirty
16 - Fifty
17 - Seventy
18 - Eighty5
19 - Horizontal
20 - ThinHorz
21 - Vertical
22 - ThinVert
23 - Diagonal
24 - ThinDiag
25 - Hatch
26 - ThinHatch
27 - Pebble
28 - ThinPebble
29 - Brick);

Color: 1 through 16

EditLayer(Layer)
Sets the layer of the selected entities. Layer can be 0 - 255 or Named Layer.
Required [selection]
Example [selection]
EditLayer(Walls)

EditLineChamfDD(Distance,Distance)
Creates a single line chamfer using distance and distance.
Required: [1st line selection]
[2nd line selection]
Chamfer point
Example: EditLineChamfDD(1,1)
ClickAt(5,5)
ClickAt(7,7)
ClickAt(6,6)

EditLineChamfDA(Distance,Angle)
Creates a single line chamfer using distance and angle.
Required: [1st line selection]
[2nd line selection]
Chamfer point
Example: EditLineChamfDA(1,45)
ClickAt(5,5)
ClickAt(7,7)
ClickAt(6,6)

EditLineChamfLA(Length,Angle)Creates a single line chamfer using length and angle.
Required: [1st line selection]
[2nd line selection]
Chamfer point
Example: EditLineChamfLA(1,45)
ClickAt(5,5)
ClickAt(7,7)
ClickAt(6,6)

EditLineStyle(line style#) Changes the line style of the selected entities.
Required: [selection]
Example: [selection]
EditLineStyle(4)
Line Style:
0 - Solid
1 - Invisible
2 - tcDashed
3 - Hidden
4 - Center
5 - Phantom
6 - Dot
7 - DashDot
8 - Border
9 - Divide
10 - DDashDDot

EditMode Sets Edit Mode On. Equivalent to choosing Edit | Line/Double Line or the Edit tool.

ExecScript(filename) Executes the named TurboCAD script. Useful to chain script files together. The filename extension must be **.TWS**.
Example: ExecScript(MYSCRIPT.TWS)

Exit Exits TurboCAD. Any drawings that have been modified will not be saved, so SaveDrawing should be used first.

Explode Explode the selected polyline.
Required: [selection]
Example: [selection]
Explode

ExportFile(filename) Exports the marked entities to the named file. The type of write depends on the file extension specified. See File | Export command for details.
Required: [selection]
Example: SelectionMode
[selection]
ExportFile(TEST.DXF)

GetAngleSel($*Angle*) Get the angle of the selected line.
Required: [selected line]

Example: [selection]
GetAngleSel($*Angle*)

GetArcCenter($*X*, $*Y*) Assigns the center point of the arc to variables X and Y.
Requires: [selected arc]

GetArcMajRad($*MajRad*) Get the major radius of the selected arc.
Required: [selected arc]
Example: [selection]
GetArcMajRad($*MajRad*)

GetArcMinRad($*MinRad*) Get the minor radius of the selected arc.
Required: [selected arc]
Example: [selection]
GetArcMinRad($*MinRad*)

GetArcStrtAng($*StrtAng*) Get the starting angle of the selected arc.
Required: [selected arc]
Example: [selection]
GetArcStrtAng($*StrtAng*)

GetArcEndAng($*EndAng*) Get the end angle of the selected arc.
Required: [selected arc]
Example: [selection]
GetArcEndAng($*EndAng*)

GetClick($*X*, $*Y*) Assigns mouse click coordinates to variables X and Y.

GetCornerSel($X, $Y) Get the corner of the selected items.
Required: [selection]
Example: [selection]
GetCornerSel($*X*, $*Y*)

GetDeltaXYSel($*DeltaX*, *$DeltaY*) Get delta-X and delta-Y of the selected line.
Required: [selected line]
Example: [selection]
GetDeltaXYSel($*DeltaX*, $*DeltaY*)

GetGrpEntityCnt($*EntityCnt*)Get the group entity count of the selected group.
Required: [selected group]
Example: [selection]
GetGrpEntityCnt($*EntityCnt*)

GetLengthSel($*Length*) Get the length of the selected line.
Required: [selected line]
Example: [selection]

GetLengthSel($*Length*)

GetMouseClick Pauses until mouse is clicked

GetOriginSel($X, $Y) Get the origin of the selected items.
Required: [selection]
Example: [selection]
GetOriginSel($*X*, $*Y*)

GroupBreak Breaks marked groups. The Selection Mode must be active.
Required: [selection]
Example: SelectionMode
[selection]
GroupBreak

GroupMake Makes a group of marked entities. Must be in Selection Mode.
Required: [selection]
Example: SelectionMode
[selection]
GroupMake

GroupSetInfo(new info) Sets info field of a symbol (group) to be displayed on symbol preview.
Required: [selection]
Example: [selection]
GroupSetInfoValue(New Info)

ImportFile(filename, X scaling factor, Y scaling factor, rotation angle)
Imports the named file into the current drawing. The type of read depends on the file extension specified. See File | Import command for details.
Required: Reference point
Example: ImportFile(TEST.DXF,1,1,0)
ClickAt(0,0)

InputNum($*Prompt*, $*Value*) Brings up a dialog box containing the *Prompt*, the assigns inputted numeric value to a variable.

InputString($Prompt, $Value) Same as above, except the input is in alpha characters.

InsertSymbol(X scaling, Y scaling, rotation angle)
Inserts symbol from clipboard into drawing. The symbol must be previously selected from a symbol library.
Required: Location for symbol
Example: InsertSymbol(1,1,90)
ClickAt(5,5)

Jump(jump label) Unconditional jump to a defined label.
Example: Jump(Label 1)

JumpIfChoice(Choice Index, Label) See Choose

JumpIfEqual(*Num-1*, *Num-2*, Label) If the variables are equal, then jump to label.

JumpIfLess(Num-1, Num-2, Label) If variable 1 is less than variable 2, then jump to label.

JumpIfLessNEq(*Num-1*, *Num-2*, Label) If variable 1 is less than or equal to variable 2, then jump to label.

JumpIfGreater(Num-1, Num-2, Label) If variable 1 is greater than variable 2, then jump to label.

JumpIfGreaterNEq(*Num-1*, *Num-2*, Label) If variable 1 is grater than or equal to variable 2 then jump to label.

JumpIfNot Eq(Num-1, Num-2, Label) If the variables are not equal, then jump to label.

MaxWindow Maximizes the current drawing window.

Message(text, loc, ms) Puts message on screen for milliseconds (ms) at location as follows:
Loc (Screen Location):
0=Center
1=Upper left
2=Upper right
3=Lower right
4=Lower left

MoveSel(X scaling factor, Y scaling factor, rotation angle)Moves marked entities.
Required: [selection]
Reference point
Destination point
Example: [selection]
MoveSel(2,2,90)
ClickAt(5,5)
ClickAt(6,6)

MoveCursor(x,y) Moves drawing cursor to x,y in current drawing units. Can not be moved outside of visible drawing area. This command is useful before pan or zoom commands.
Example: MoveCursor(5,5)

MultNums(*Num-1*, *Num-2*, $*Results)* Multiplies variable 1 by variable two and assigns the result to a third variable

NewDrawing Opens a new drawing window.

OpenDrawing(filename) Loads named drawing. The type of file load depends on the extension used. See the File | Open command for valid extensions.
Example: OpenDrawing(LRPLAN.TCW)

OpenLibrary(filename) Opens named symbol library in select mode. The filename extension must be **.SLW**.
Example: OpenLibrary(BATH1.SLW)

PanToPoint Pans to the next user specified point. If the desired point is on the screen ClickAt can be used with the ScreenPos Snap Mode, otherwise Value should be used with another Snap Mode.

Example:
```
PanToPoint
SetSnapScreenPos
ClickAt(1,3)
```

Example:
```
PanToPoint
SetSnapCoordsAbs
Value(1)
Value(3)
```

Pause(ms) Pauses for ms milliseconds.1 second = 1000 milliseconds.

PlotDrawing Plots current drawing.

PrintDrawing Prints current drawing.

PrintFit2Page(true/false) Set the print fit to page option.

Redraw Re-draws the current drawing.

RepeatScript Repeats the entire script. Use if looping through script is desired.

RestoreNamedStyle(style name) Restores a named style.

RestoreNamedView(view name) Restores a named view.

RestorePreviousView Restores immediately previous view.

RotateSel(angle) Rotates the selected entities.

Required: [selection]

Example:
```
[selection]
RotateSel(45)
```

SaveDrawing(filename) Saves current drawing as filename. If a drawing with the same name exists, it will be overwritten without warning. The type of file save depends on the extension. See the File | Save As command for details.

SaveNamedStyle(style name) Save the current attribute settings as a style.

SaveNamedView(view name) Save the current view.

ScaleSel(X scaling factor, Y scaling factor) Scales the selected entities.

Required: [selection]

Example:
```
[selection]
ScaleSel(0.5,0.5)
```

SelectAll — Selects all entities on active layers.

SelectByLayer(layer) — Select all entities on a given layer.

SelectByType(type) — Select all entities of a given type: arcs, lines, text, dimensions, and so on.
Types are as follows:
1 = All
2 = Layer
3 = Line
4 = Arc
5 = Dim
6 = Point
7 = Group
8 = Text
9 = Curve
10 = DblLine
11 = Bitmap

SelectionMode — Forces program into Selection Mode. Issue this command prior to commands the require entity selection before issuing the specific action, e.g., Delete, Export, etc...

SelectNext — Similar to Click for selection of entities but adds to current selection list.

SelectNextAt(x,y) — Similar to ClickAt for selection of entities but adds to current selection list.

SelectNextWithWin — Similar to SelectWithWin but adds to current selection list.
Required: Origin point of selection window
Corner point of selection window
Example: SelectNextWithWin
ClickAt(2,2)
ClickAt(6,6)

SelectWithWin — Selects entities within a window.
Required: Origin point of selection window
Corner point of selection window
Example: SelectWithWin
ClickAt(2,2)
ClickAt(6,6)

SendBack — Send the selected entity to the back of the entity stack.
Example: [selection]
SendBack

SetAngleSel(angle) — Set the angle of the selected entity.
Example: [selection]
SetAngleSel(45)

SetArcLayer(#) — Sets the default arc layer, -1..255. If -1 is used, Status Bar Setting will be used.

SetArcMajRad(major radius) — Set the major radius of the selected arc.
Required: [selected arc]

Example: [selection]
SetArcMajRad(5)

SetArcMinRad(minor radius) Set the minor radius of the selected arc.
Required: [selected arc]
Example: [selection]
SetArcMinRad(5)

SetArcEndAng(end angle) Set the ending angle for the selected arc.
Required: [selected arc]
Example: [selection]
SetArcEndAng(45)

SetArcStrtAng(start angle) Set the starting angle for the selected arc.
Required: [selected arc]
Example: [selection]
SetArcStrtAng(45)

SetAutoDim(on/off) Sets auto dimensions on/off.

SetCornerSel(X, Y) Set the corner of the selected entity, resizing it.
Example: [selection]
SetCornerSel(1,4)

SetCPtPen(#) Sets default const point pen, -1 or 1..16. If -1 is used, Status Bar selection will be used.

SetCursor(On/Off) Sets drawing cursor type.
Cursor Types:
On = Large cross
Off = Small cross

SetDecs(#) Sets drawing decimals places, 0..4.

SetDimArrow(#) Sets the default dimension arrow type.
Arrow Types:
0 = normal arrow
1 = closed normal arrow
2 = slash
3 = cross

SetDimCSize(h) Sets the default dimension text size in current drawing units.

SetDimDecs(#) Sets the default dimension decimals, 0..4.

SetDimFont(name) Sets the default dimension font to the named font. If the named font can not be found, no change will occur.

SetDLineOffsetRef(#) Sets the default offset reference for new double line construction.
Offsets are:
0 = left
1 = center
2 = right

SetDLineSeparation(#) Sets the default distance between the 2 lines of a double line.

SetEllipseLines(#) Sets number of ellipse lines, 3..255.

SetFastText(on/off) Sets fast text on/off.

SetFillAngle(angle) Set the angle option for the current vector-based fill pattern.

SetFillCross(On/Off) Set the crossed option for the current vector-based fill pattern.

SetFillCur(fill, color) Set the current fill pattern for the drawing.
Fill:
0 - Solid
1 - Invisible
2 - Vasterisks
3 - Vbrick
4 - VCeilingGrid
5 - VCheckerPlate
6 - Vconcrete
7 - Vearth
8 - Vherringbone
9 - VHoneyComb
10 - Vsteel
11 - VTriangleGrid
12 - Tcw10Hatch
13 - Ten
14 - Fifteen
15 - Thirty
16 - Fifty
17 - Seventy
18 - Eighty5
19 - Horizontal
20 - ThinHorz
21 - Vertical
22 - ThinVert
23 - Diagonal
24 - ThinDiag
25 - Hatch
26 - ThinHatch
27 - Pebble
28 - ThinPebble
29 - Brick
Color: 1-16

SetFillScale(scale) Set the scale option for the current vector based fill pattern.

SetGrid(on/off) Sets grid on/off.

SetGridFreq(x,y) Sets the default grid display frequency, 1...999.

SetGridPen(#) Sets the default grid pen, -1 or 1..16. If -1 is used, Status Bar selection will be used.

SetGridStep(x,y) Sets the default grid step distance in current drawing units.

SetGridStyle(#) Sets the Grid Style to lines or dots/points : 0 = Lined, 1 = Point

SetLayer(#, on/off) Sets layer number 1..255 on/off.

SetLayerCur(#) Sets the current drawing layer, 0..255.

SetLineStyleCur(#) Sets the default line style, -1..10. If -1 is used, Status Bar selection will be used.
Styles are:
0 - ls_Solid
1 - ls_Invisible
2 - ls_tcDashed
3 - ls_Hidden
4 - ls_Center
5 - ls_Phantom
6 - ls_Dot
7 - ls_DashDot
8 - ls_Border
9 - ls_Divide
10 - ls_DDashDDot

SetLineThickCur(#) Sets the default line thickness, -1..15 If -1 is used, Status Bar selection will be used.

SetNearSearch(on/off) Sets near search mode on/off.

SetOrthoAngle(degrees) Sets drawing ortho angle, 1..180.

SetOrthoMode(on/off) Sets ortho mode on/off.

SetPaperSize(#) Sets drawing paper size.
Paper Sizes:
0 - ANSI_A
1 - ANSI_AP
2 - ANSI_B
3 - ANSI_BP
4 - ANSI_C
5 - ANSI_CP
6 - ANSI_D
7 - ANSI_DP
8 - ANSI_E
9 - ANSI_EP
10 - ISO_A4
11 - ISO_A4P
12 - ISO_A3
13 - ISO_A3P
14 - ISO_A2
15 - ISO_A2P
16 - ISO_A1
17 - ISO_A1P
18 - ISO_A0
19 - ISO_AOP
20 - CUSTOM
21 - MATCHPRINTER

SetColorCur(Pen#) Sets the current Pen Color. 1...16

SetProxPerc(#) Sets snap proximity percentage, 1..100.

SetRubberBand(on/off) Sets rubber banding on/off.

SetScale(number, page units, number, world units) Set the scale for the current drawing.

Page units are as follows: 0=Millimeters
1=Inches

World units are as follows: 0=Millimeters
1=Centimeters
2=Meters
3=Kilometers
4=Inches
5=Feet
6=Fractional Inches
7=Fractional Feet
8=Yards
9=Miles
10=Microns
11=Angstroms

Example: SetScale(0.25,1,1,6)

SetSnapArcAngle(Angle) Sets snap mode to point on arc and an angle.

SetSnapEndPtArc Sets snap mode to endpoint of an arc.

SetSnapGrid Sets snap mode to snap to grid.

SetSnapIntersect Sets snap mode to intersection of line-line, line-arc, or arc-arc.

SetSnapLinePt Sets snap mode to a point on a line.

SetSnapMidPtLine Sets snap mode to midpoint of a line.

SetSnapNearPt Sets snap mode to near point.

SetSnapOnArcPt Sets snap mode to a point on an arc.

SetSnapPolar(radius, angle) Sets snap mode to polar coordinates.

SetSnapRelative(delta X, delta Y) Sets snap mode to relative coordinates.

SetSnapScreenPos Sets snap mode to screen position.

SetTextAngle(degrees) Sets the default text angle, 0..360.

SetTextCSize(font height) Sets the default text character height in current drawing units.

SetTextFont(font name) Sets the default text font to the named font. If the named font can not be found, no changes are made.

SetTextJustify(#) Sets default text justification.

Justification:

1 = Left, Bottom
2 = Left, Center
3 = Left, Top
4 = Center, Bottom
5 = Center, Center
6 = Center, Top
7 = Right, Bottom
8 = Right, Center
9 = Right, Top

SetWinSelOpen(on/off) Sets window select open mode on/off.

Stop Immediately stops script execution.

SubNums(*Num-1*, *Num-2*, *Results*) Adds variable 1 to variable 2 and stores the sum in a third variable.

Undo Undoes the last insert, delete, or edit.

ViewPrintMargin Toggles the print margin display on and off.

ViewRuler Toggles the ruler display on and off.

ViewSnaps Toggles the Snaps palette display on and off.

ViewToolbar Toggles the toolbar display on and off.

WinClose Closes current window. Modified drawings are not saved. Use SaveDrawing first.

WinCloseAll Closes all windows. Modified drawings are not saved. Use SaveDrawing first.

WinSelect(window name) Selects a drawing window.

ZoomExtents Zooms window to drawing extents.

ZoomFull Zooms window to full drawing size.

ZoomIn Zooms window in one level and attempts to keep drawing cursor centered.
ZoomOut Zooms window out one level and attempts to keep drawing cursor centered.

ZoomReset Zooms window to the view when the drawing was first opened.

ZoomWindow Zooms window to defined window.
Requires: Upper left cornerLower right corner
Example: ZoomW