



Becoming an AutoCAD® PowerUser

dave espinosa-aguilar

GD315-4 Course Summary: This course outlines 10 practical philosophies which lead to mastering any release of AutoCAD. Along the way, discover new and exciting techniques to take basic AutoCAD beyond your wildest dreams, such as how to reduce file sizes without losing information, share drawings with people who do not own or know how to use AutoCAD (or any other CAD program for that matter), create complicated linetypes and hatch patterns out of anything you can draw, create masking zones, create instant counts of block types, speed up your drawings without spending, accomplish regional layer control (learn to freeze an "area" of your drawing), and much more.

About the Speaker: A consultant in the CAD and Multimedia industry since 1986, dave espinosa-aguilar has trained architectural and engineering firms on the general use, customization and advanced programming of design and visualization applications from Autodesk including AutoCAD, AutoCAD Map, Architectural Desktop, Land Desktop & Civil 3D, 3D Studio MAX and Autodesk VIZ. dave's passion is streamlining and automating design production environments through onsite customization and programming, and he has authored the facilities management applications of several Fortune 500 companies using AutoCAD ObjectARX, VB/VBA, AutoLISP/DCL and MAXScript technologies. dave has also produced graphics applications and animations for Toxic Frog Multimedia and has co-authored several books including NRP's "Inside 3D Studio MAX" series. He has been a speaker at Autodesk University since its inception, and served on the Board of Directors for Autodesk User Group International for 6 years including the office of President in 1996. dave currently works for the City of Richland (Washington) GIS department as a software engineer.

Why i type:

1. it is the only consistent AutoCAD interface since its inception
2. it is the fastest interface to most settings and command options
 - a. the fastest users always type (using aliases)
 - b. examine DIMTXT vs DDIM dialog (6 clicks) to set dimension text height
3. it is the only interface to certain settings and command options including legacies
 - a. examine PURGE vs. -PURGE for Regapps
 - b. examine *_TOOLPALETTEPATH
 - c. examine -ARRAY/C, UCS/3P, etc
4. scripting cannot use dialog interfaces
5. if you ever get into programming, you MUST know typed versions of commands and system variables



Autodesk
University
2007

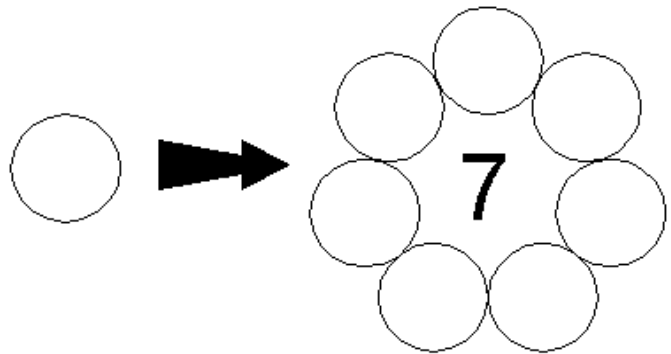
1. Watch Others Work...

- lots of techniques 'out there' that are not documented --- where does one get ahold of them?
- a co-worker sitting next to you may know something crucial to your work
- local user groups, office meetings, etc offer plenty of opportunities to discuss slow processes
- a newbie has no pre-conceived notions of do's/don'ts --- so newbies often work miracles!!
- some of the most productive people I know are the laziest! Watch lazy people who keep up!
- watching someone work transmits information a lot faster than reading about something

Examine the **ARRAY** command for some examples of how a command you may have thought you knew everything about runs into unexpected obstacles and challenges:

a. drafting technique: circular array (or "i don't have a clue" syndrome)

- draw a circle of arbitrary size
- array 7 perfectly tangential circles from it
- how do you do it?
- the ARRAY dialog won't help
- key: polygonal nature of question



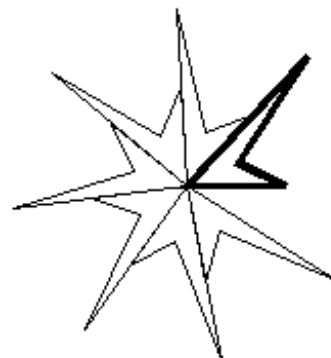
b. environment setup: arclinear array (or "why does it behave that way?" syndrome)

- draw arbitrary arc
- offset it
- draw a line between its right endpoints
- array 21 perfectly fit 21 lines along arc
- how do you do it?
- the ARRAY dialog won't help
- key: WCS/UCS/ECS



c. hidden functionality: osnap arraying (or "what did you just do?" syndrome)

draw "asteroid" PLINE shape
set GRIPS to <1>
array as many adjacent PLINES as will fit in circle
how do you do it?
the ARRAY dialog won't help
key: GRIPS<1>, pick basepoint, spacebar ROTATE mode
hold **CTRL** down, pick copying endpoint



2. Trust Your Instincts...

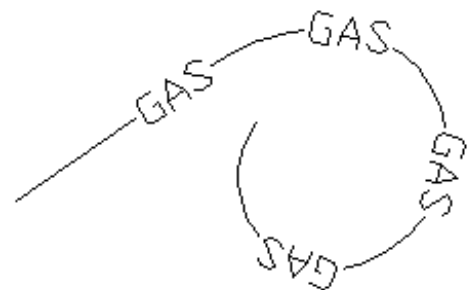
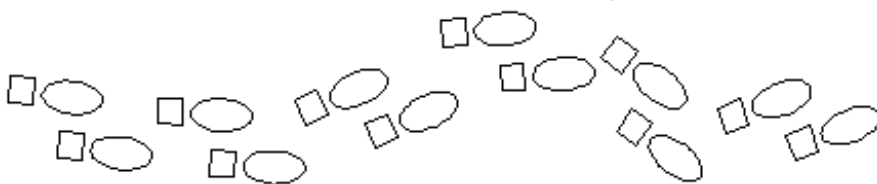
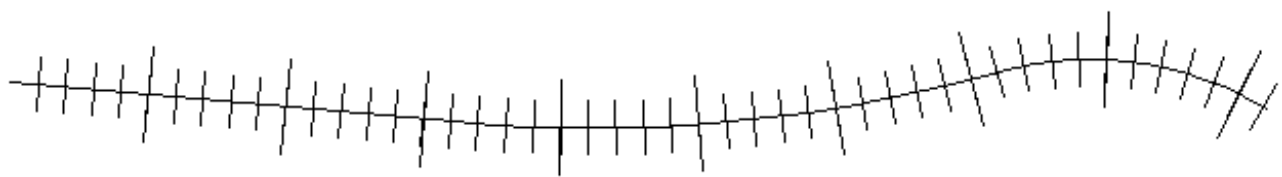
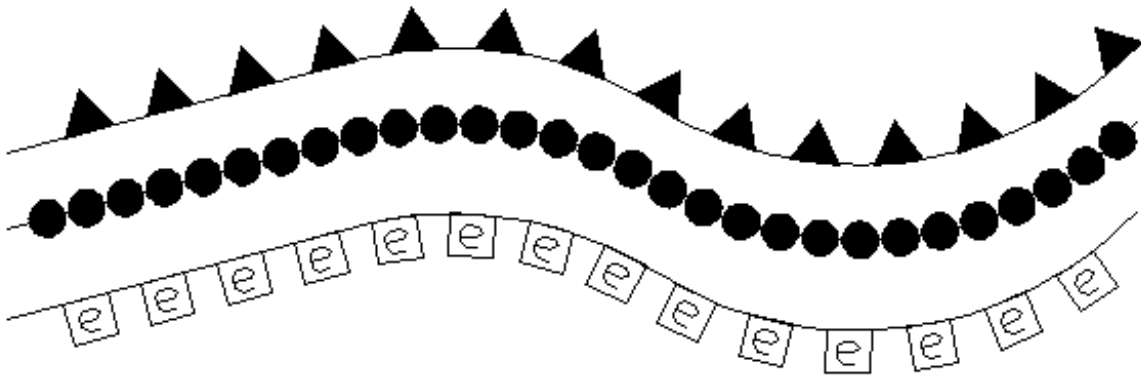
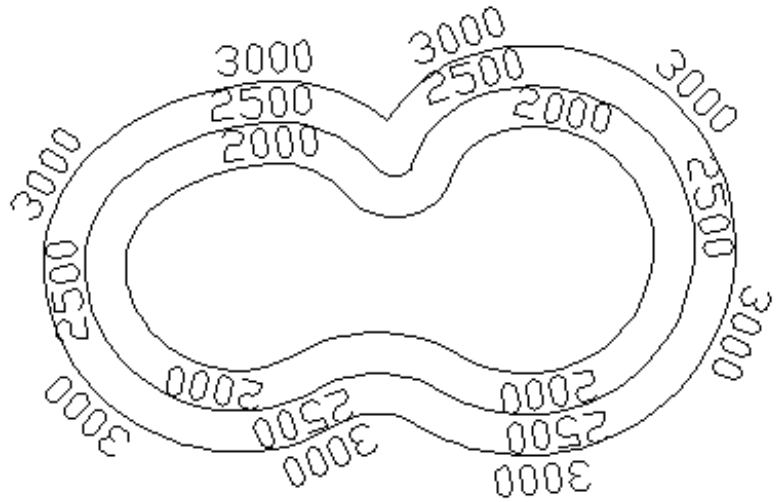
- pay attention to your boredom
- when brain thinks "there must be a better/faster way to do this"... it is usually right
- some will knowingly continue to work using the only methods/commands they know. don't!
- a poweruser knows that his/her knowledge is always unfinished.

If you had to create hundreds of labeled contours, how would you do it?

DTEXT, rotate a million times?

"Polytracing" is an old technique used to make linetypes out of anything you can draw in seconds:

- bypasses shape-making limitations
- no need to send additional LIN and SHP files with a drawing to make it appear properly.
- complete control over spacing (not hardcoded in shape definition file)— simply retrace!
- Leave original tracing polyline on no-plot layer or freeze it for later re-tracing if needed. Collect all 'linetype' elements with FILTER or QSELECT (use a dedicated LAYER or BLOCK name for the traces)



If this technique works with TEXT entities, will it work with MTEXT entities? Will it honor MTEXT properties like... background masking??

If this technique will harness the power of BLOCKC, will it also work with Dynamic Blocks? (ex: Align Parameter)

3. See Beyond the Documented Purpose...

- for over 20 years powerful AutoCAD techniques have evolved in the workplace (AutoCAD "Lore")
- these techniques are well understood – yet there is nothing about them in the manuals
- these techniques are widely used – yet they are not a part of any educational curriculum
- they typically formed out of a need to solve a crisis
- they involve commands you already know how to use
- this is true not only for AutoCAD, but any application (ex: Excel)

a. the ultimate PURGE command:

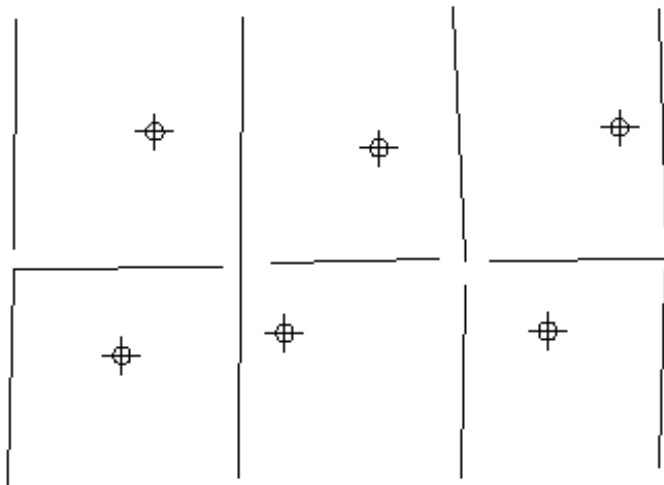
The first poweruser story ever told at this annual event: (NAAUG wishlist meeting 1990 — a user approached the microphone before autodesk technical panel and asked for the PURGE command. The panel says it didn't exist. another user approached the microphone and said to use **WBLOCK** to do it. And this is **still the laziest fastest way to purge** a drawing of unwanted data.

b. Hyperlink used as a non-BLOCK ATTRIBUTE:

Ever wish there was a way to assign ATTRIBUTES to non-BLOCKS? The Hyperlink field (rarely used for anything) can double as an ATTRIBUTE for any entity type.

Data Extraction will even honor it, and "hovering" over the entity even reports the "ATTRIBUTE!"

Count	Name	PN#
1	Line	PN#2000
3	Line	PN#1000



c. a way to approach data counts:

Ever had to deal with "crapCAD?" How many parking lots are there to the right? If you had hundreds drawn this poorly, how would you count them all? Consider the use of entities for data gathering purposes, not drafting purposes: create dedicated count layer, set **DDPTYPE** to something visible, and use **MULTIPLE POINTS** to count later using **FILTER** or equivalent command.

d. EXCEL as an AutoCAD polyline builder:

Very few users realize that cell range data in Excel can be pasted directly to AutoCAD for drawing linework!! If you have columnated COGO-type X,Y,Z data, you can insert a new column in front of all your coordinate data rows, and paste a CONCATENATE function in all cells of this column to build the XYZ data into 3D coordinates AutoCAD can understand.

	A	B	C
1	1	1	0
2	2	2	0
3	2	1	0
4	3	2	0

The CONCATENATE function "glues" the other column data into a (X,Y,Z) format. The entire column can then be pasted into the Command prompt (following a **3DPOLY** statement). Simply "close" when finished.

	A1	=CONCATENATE(B1,",",C1,",",D1)		
	A	B	C	D
1	1,1,0	1	1	0
2	2,2,0	2	2	0
3	2,1,0	2	1	0
4	3,2,0	3	2	0

4. Learn the Program. All of It. You PAID for All of It!

- maybe you own R2008, but you may be using it like R12DOS
- the real upgrade happens in your head... upgrade that before you buy more "upgrades"
- most users today do not even know, understand or exploit over 25% of the technologies they've already paid for
- the "shrinking pie syndrome:" how many of the features that you paid for do you use?



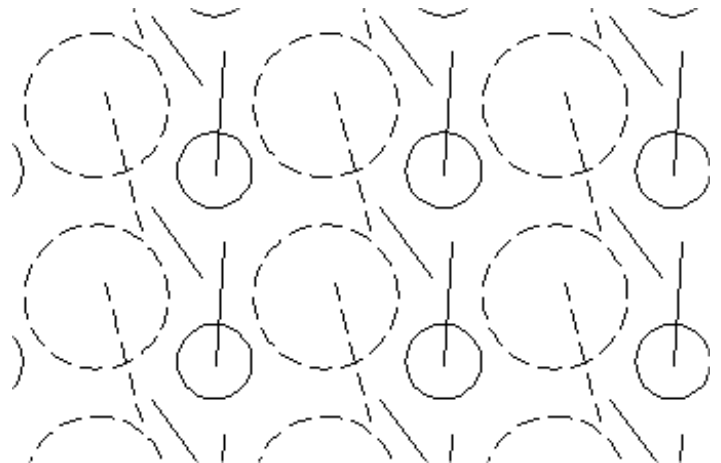
Regions? Solids? Rendering? DbConnect? OLE Objects? Diesel? Scripting? AutoLISP or Visual LISP? Visual Basic for Applications? ObjectARX? The Express Tools? 3D? Data Extraction?

Asking the really hard questions:

- Is Autodesk Design Review (formerly DWF Composer) enough?
- Is AutoCAD LT enough?
- Examine your 2008 list of new technologies... how many of you are already using the sheet set manager? The plot and publish tools? Tables? Fields? Markups? New notation symbols?
- Test yourself: take a tally of all the commands you use in a week... then compare them to the complete list of commands available in the program. All of them. What's your percentage?
- By taking on one command or system variable a workday, you can move thru the entire syntax in a matter of months and make an informed decision as to what you really need to know.

Knowing one command might make all the difference in your world:

a. Imagine having to get rid of the dashed circles in this image to the right. Yes, you could ZOOM/PAN your way to selecting every one of them, but wouldn't it be easier to "filter" them?



FILTER? What's that?

I don't see **FILTER** in a pulldown menu!!!

FILTER rocks. Not only does it go light years beyond **QSELECT**, you can even **save NFS files** for repeated filtering operations.

b. Ever wish you could select BLOCKS by their ATTRIBUTE values? **QSELECT** and **FILTER** won't do it... but **FIND** will!!!! If you use the current selection (Search in:) mode and select ALL entities first, note that **Zoom To** and **Select All** options are then available for automatic block selection!

Never customized before? Scared to? Start simple with Scripting!!

"If you can type it, you can script it!" Consider a block re-definition (simplification) script for speeding along editing, or a system variable settings script for configuring any AutoCAD station you have to work on just the way you like it setup. Use command line copy/paste techniques to generate scripts with less memorizing of sequences.

a. a "scrub" operation per client (MAGIC.SCR) which renames layers, deletes information from particular layers, thaw's all layers and zoom to extents!

Does anyone sing the praises of the 'CAL command anymore? Does anyone even know of it?

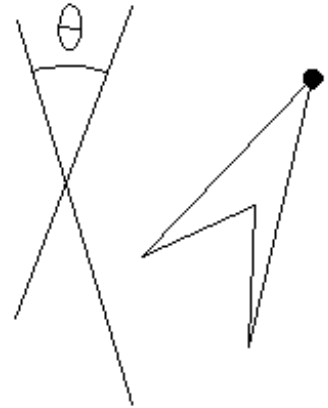
Yes, since AutoCAD 2005 we've had the "midpoint between 2 endpoints" object snap mode to get us to a midpoint between two endpoints. But... we've actually had this ability in the form of the 'CAL command's **MEE** function since AutoCAD R12 DOS days. **CAL** does this kind of thing (calcing distances, angles, etc) very well and it really pays to know its **over 100 subfunctions**. Here is just a taste of its power:

Case #1:

Rotate the enclosed polyline about the solid dot angle theta.

Technique:

Use **ROTATE** command per usual.
when prompted for rotation angle, invoke '**CAL**
use **ang(int,end,end)** to describe angle

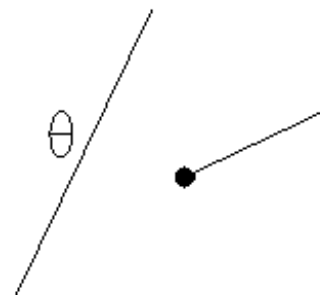


Case #2:

Rotate the short line about the solid dot so that it is flush with angle theta.

Technique:

use **ROTATE** command per usual.
use **Reference** mode to align the line with cursor
when prompted for rotation angle, invoke '**CAL**
use **ang(end,end)** to describe angle

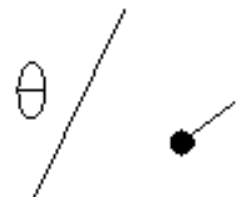


Case #3:

Extend the short line a distance of theta in the short line's direction.

Technique:

Use **LENGTHEN** command per usual. Use **DElta** option.
invoke '**CAL** using **dist(end,end)** and select long line endpoints

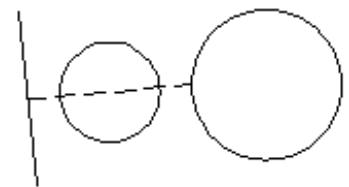


Case #4:

Draw a new circle whose center is exactly halfway between a line's midpoint and a circle's quadrant which has a radius 2/3 the size of the original circle.

Technique:

use **CIRCLE** command. When prompted for centerpoint invoke '**CAL**.
use **(mid+qua)/2** to find centerpoint.
When prompted for radius, invoke '**CAL**.
use **rad*2/3** and select original circle.

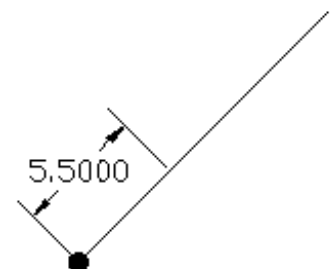


Case #5:

Draw a line beginning 5.5 units away from the solid dot along the line.

Technique:

use **LINE** command. When prompted for point, invoke '**CAL**.
use **pld(end,end,5.5)** picking endpoints to begin line.



5. Small = Fast

- the golden CADD rule: the smaller the drawing, the faster it is to load, save, and edit
- AutoCAD is never faster than when it first boots up. From then on everything just gets slower.
- *demandloading* is an internal technology, but it can also be a user strategy!!
- several techniques have evolved over the years which are founded on this observation.

a. Examine how TEXT can be manipulated to speed up drawing performance:

STYLE command can be used to temporarily set much simpler fonts for heavily-used styles.

QTEXT can speed up all text without losing anything if linework is the priority.

Text can be imported thru the **MTEXT** command at any time... meaning it doesn't need to be there when you're needing speed for linework.

b. Examine the **WBLOCK** and **BLOCK** commands for some examples of how again commands you may have thought you knew everything about offer amazing undocumented capabilities.

1. **WBLOCK** as a file reduction tool:

Not only can **WBLOCK** be used as a purging tool for unwanted data, but it can also be used to reduce a drawing to its minimum size. Simply **WBLOCK** the drawing out to itself from the origin.

2. **WBLOCK** as a tool for pieceworking:

why work on UNIVERSE.DWG... or TEXAS.DWG... when you can open DALLAS.DWG?

why work at the speed of **clutter** when you can work at the speed of **clarity**?

the essential principle: if you don't need it **NOW**, get it out of the session.

there is a sweetspot involved: compare the time of benefit with the time of preparation.

make sure the temporary pieced files you generate are **SAFE** from accidental erasures.

this is how users on R12 DOS 386s outperform AutoCAD 2005 users on pentiums.

the only investment is preparation time and organization of compartmentalized data.

Three types of pieceworking:

regional pieceworking: examine the drawing for areas you wish to work in. **WBLOCK** them out to temporary files from the same point (use origin to keep things simple). Be sure to **WBLOCK** out the remainder of the drawing from the origin for reconstruction.

layer pieceworking: what do you think is faster to edit... frozen entities... or nothingness? examine the drawing for layers you wish to work on. **WBLOCK** them out to temporary files from the same point (use the origin to keep things simple). Be sure to **WBLOCK** out the remainder of the drawing from the origin for reconstruction.

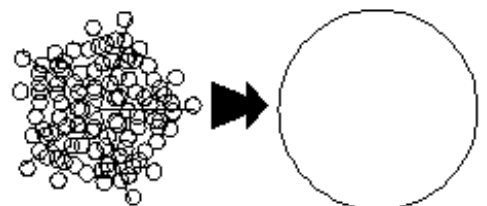
entity pieceworking: do you really need that titleblock there? How about those specification notes? How about those details, key legends, hatched in borders, background fillers, or XREFs? examine the drawing for entity types you wish to work on. **WBLOCK** them out to temporary files from the same point (use the origin to keep things simple). Be sure to **WBLOCK** out the remainder of the drawing from the origin for reconstruction. Need to see where you should draw? Use a temporary border.

c. **WBLOCK** / **BLOCK** / **INSERT** used for block substitutions:

Swap complex blocks with simplistic versions for faster editing purposes, redefine them later for finalized plotting/archiving/etc create equivalent simple blocks from same insertion point

WBLOCK the complex versions out to drive for later re-insertion

Use **INSERT/=** to redefine in drawing when needed.



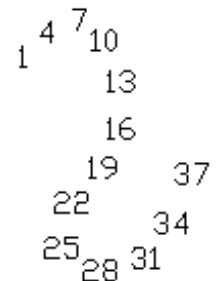
6. Programming is PowerDrafting

- All the great third-party apps and utilities down thru the years started out as AutoLISP routines.
- If you can master PEDIT, you can master AutoLISP. It's not hard stuff to learn.
- There's a ton of sample code "out there" in the world to learn from.
- No waiting around until AutoCAD 2009.
- There's a ton of people who would love to teach it to you.

a. Incremented numeric TEXT

Imagine having to create the labels shown to the right... hundreds of them. In sequence. Manually. In about 3 minutes of typing you'd be able to add this kind of functionality to AutoCAD. If you know AutoLISP.

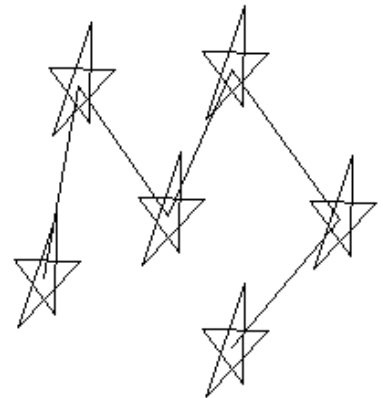
```
(defun C:ITEXT ()
  (setq myheight (getreal "\nEnter text height: "))
  (setq mynumber (getint "\nEnter beginning number: "))
  (setq myincrement (getint "\nEnter increment step: "))
  (setq mypoint (getpoint "\nPick point: "))
  (while mypoint
    (command "text" "j" "m" mypoint myheight "0" (itoa mynumber))
    (setq mynumber (+ mynumber myincrement))
    (setq mypoint (getpoint "\nPick point: "))
  )
)
```



b. Blocks at PLINE vertices

Imagine having to place blocks from their insertion points at every vertex of a polyline. Manually. In about 3 minutes of typing you'd be able to add this kind of functionality to AutoCAD. If you know AutoLISP.

```
(defun C:BPLINE ()
  (setq blockname (getstring "\nEnter the name of the block: "))
  (setq plinelist (entget (car (entsel "\nPick PLINE: "))))
  (setq counter 0 total (length plinelist))
  (while (< counter total)
    (setq listitem (nth counter plinelist))
    (if (= (car listitem) 10)
      (command "insert" blockname
        (list (cadr listitem) (caddr listitem) 0.0) "" "" ""))
    (setq counter (+ counter 1))
  )
)
```



c. explode all GROUPs

Explode all the groups in a drawing.

```
(defun C:DELGROUPS ()
  (setvar "cmdecho" 0)
  (setq objdict (namedobjdict) grpdict (dictsearch objdict "ACAD_GROUP") grplist nil)
  (foreach listitem grpdict
    (setq grpcode (car listitem))
    (if (= grpcode 3) (setq grplist (cons (cdr listitem) grplist)))
  )
  (setq ca 0 ta (length grplist))
  (repeat ta
    (setq grp1 (nth ca grplist)) (command "group" "explode" grp1) (setq ca (+ ca 1))
  )
  (princ)
)
```

7. Stay Task-Focused

What if you need to:

- share your drawings with people who can't use your DXF, DWG or DWF files?
- share your drawings with people who don't need to edit but see/discuss your drawings?
- share your drawings with people you don't want changing your drawings?
- share your drawings with people you really don't want having your drawings in the first place?
- share your drawings with people using very old versions of AutoCAD or other CAD apps?
- share your drawings without giving away proprietary resources (block libraries, fonts, etc)
- share your drawings (including xrefs) with people who have very different directory structures?
- share your drawings without sending horrendously large files?
- share your drawings with idiots, technophobes and people still using 286s's with Win 3.1?

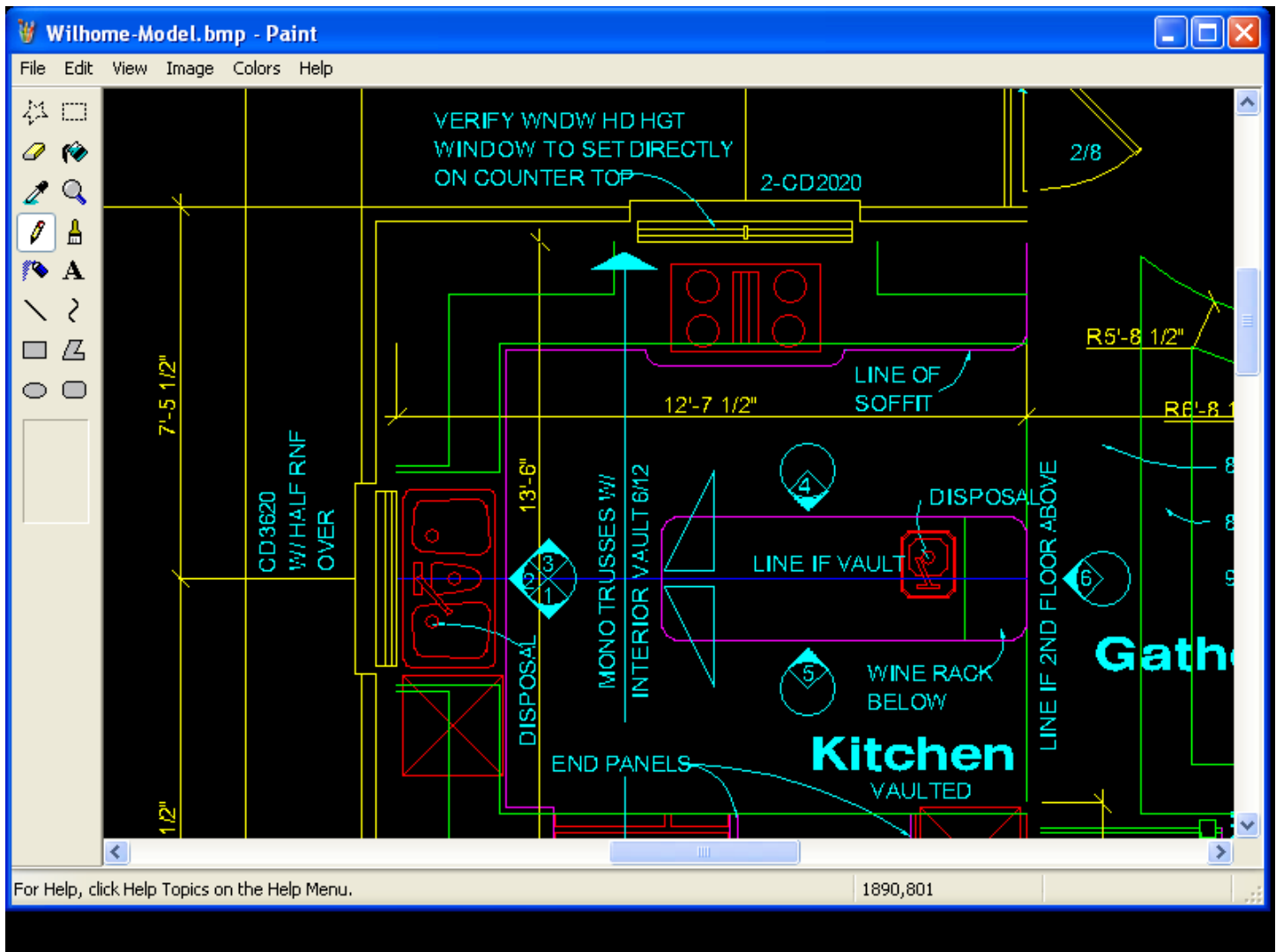
Consider a Screen Capture!!!

Print Screen Button (clipboard), paste into email and send, discuss. Done.

OR

Consider PLOTTING to a Raster File Format

Open drawing(s), create/configure plotting device, **PLOT** to desired resolution, depth and format.
No hassles or technological hurdles to jump, quick, small, easy and safe (no access to digital data).

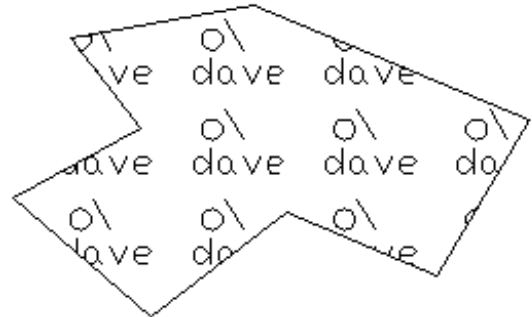


8. Experiment!!!

- “Playing Around” and “Goofing Off” with AutoCAD is **HIGHLY PRODUCTIVE**
- Call it “R&D” if you have to, but make time every week to fiddle around with AutoCAD
- Beer and AutoCAD go very well together...time for the 3am XCLIP story...

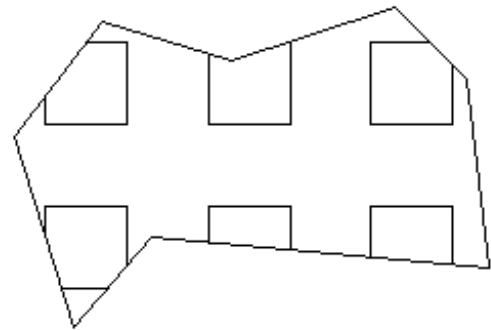
HATCH Pattern Example:

- Draw POLYLINE boundary for area to be hatched
- Draw hatch “cell”
- Make BLOCK out of the cell
- MINSERT it over the entire area. Change Properties as desired.
- XCLIP the MINSERT. WHAT? YOU CAN XCLIP A MINSERT? Yes. You can XCLIP a BLOCK too!
- No .PAT file needed in transit to make it work.
- Fast to make.
- More versatile even than **SUPERHATCH!**



Try it with Dynamic Block Arrays!!!

- BEDIT to enter the Block Editor (call the block MYHATCH)
- Draw a RECTANG from 0,0 to 1,1
- Add an XY Parameter that uses the RECTANG and stretches from 0,0 to 1,1
- Add an Array Action that arrays 2.0 units in the X & Y axis
- Save the BLOCK and exit the Block Editor
- Draw a PLINE boundary for the hatching area
- INSERT the MYHATCH Block and stretch it over the desired area to be "hatched"
- Use XCLIP to clip the dynamic block



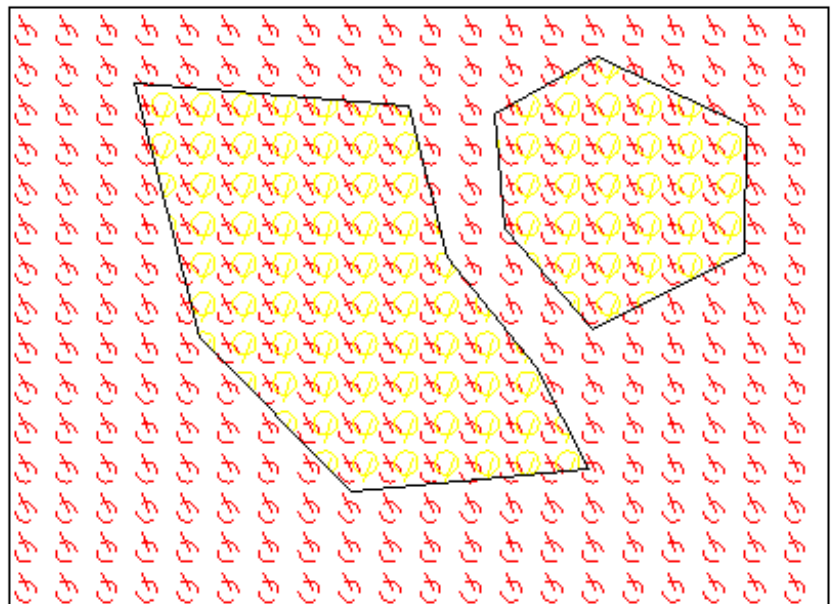
Regional LAYER Control:

Can you freeze a layer in an area of a drawing? (No.)

Wouldn't it be incredible... if we could have **REGIONAL LAYER CONTROL???**

Paperspace makes it possible: an “x-ray” into layer information in a zone!

- Create layer 1 and 2 (red solid yellow dashed) with lines and circles in Modelspace.
- In paperspace create layer VIEWS.
- Create RECTANG for viewport boundary.
- Create Viewport using MVIEW/Object.
- Turn GRIPS on.
- Copy viewport onto itself. This maintains exact overlap.
- Grip stretch to interior boundary.
- VPLAYER/Freeze select outer polyline.
- This x-rays the layout.
- Use PEDIT to modify the x-ray boundary in any way you wish.
- This incredible benefit has nothing to do with doing multiple scales on the same sheet.



9. Don't Buy Into the Myths...

- There are a lot of myths circulating out there about:
- how certain commands "should be used"
- how certain commands "should not be used"
- what certain commands "are only used for"

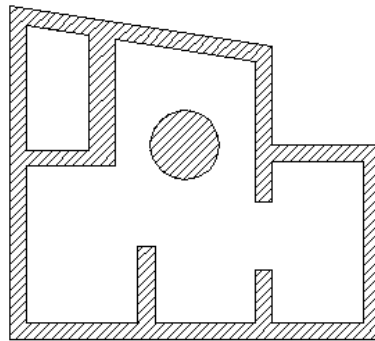
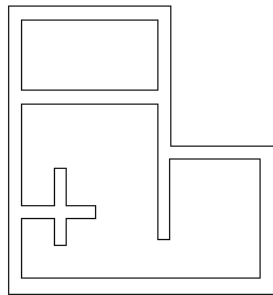
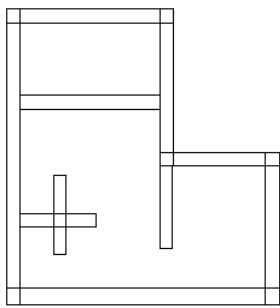
This causes:

A great deal of AutoCAD capability to go unexploited

A great deal of productivity to get bypassed

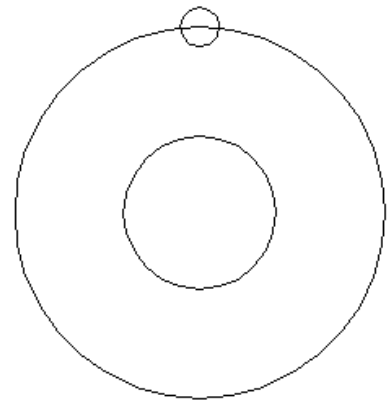
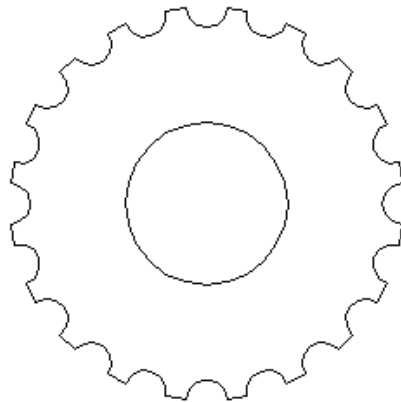
Users to stop thinking outside the box.

Solids Modeling is only for people who do 3D modeling.

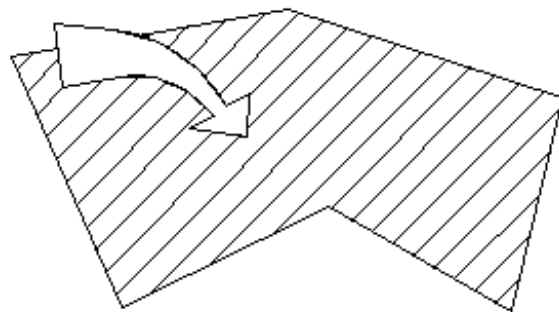
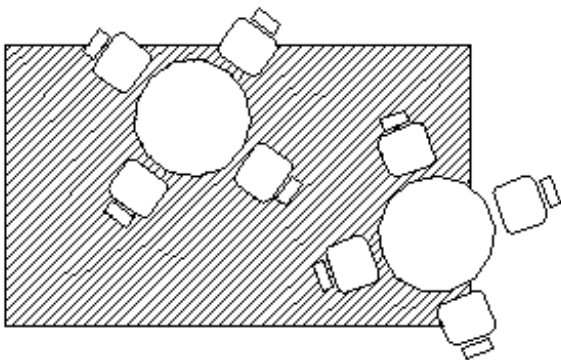


Walls cleaned up
instantly
Areas and perimeters
found in seconds.
Fast modeling

Try drawing this manually without the
use of Solids Modeling functions!



This technique is an incredible bypass to the limitations on internal masking applications and functions:



These are all 2D CAD applications.

10. It's not just what you know... it's who you know.

- A long time ago I gave up the notion that I could "know it all about AutoCAD"
- Everyone specializes. You're as powerful as the people you can turn to.
- New techniques are born everyday. Exposure is the key to learning them.
- The person sitting next to you knows something about AutoCAD you don't. Same for your co-workers.
- Autodesk forums, AUGI forums, etc... think of them as interactive books!
- Turn to your neighbor... get his/her card, put a face to an email address!!