IT21765 Perfect Plots, Every Time, Every User

Frank Mayfield

CAD Administrator, Benham - a Haskell Company

Twitter: @theMadCadr

Class summary

Ensure that the hard work you put into developing your project page setups are preserved. Take the guess work out of plotting for your users. They'll love being able to simply select a Page Setup, and know that it's always right!

Key learning objectives

At the end of this class, you will be able to:

- Learn the workflow and application of autolisp coding to take back and apply at their office.
- Understand the need for setting and adhering to standard naming conventions.
- Consider other applications based on this programming philosophy.
- Demonstrate tangible results and time savings to management by applying these techniques.

About Me

- CAD Manager with Benham in Tulsa, Oklahoma
- AutoCAD user since v2.61 (1986)
- 15 years of CAD Management experience
- 5 years as a software developer utilizing Object Oriented Common Lisp (CLOS)
- AutoCAD All-Star Mentor
- 14th Autodesk University

@theMadCadr
https://www.linkedin.com/in/franklmayfield

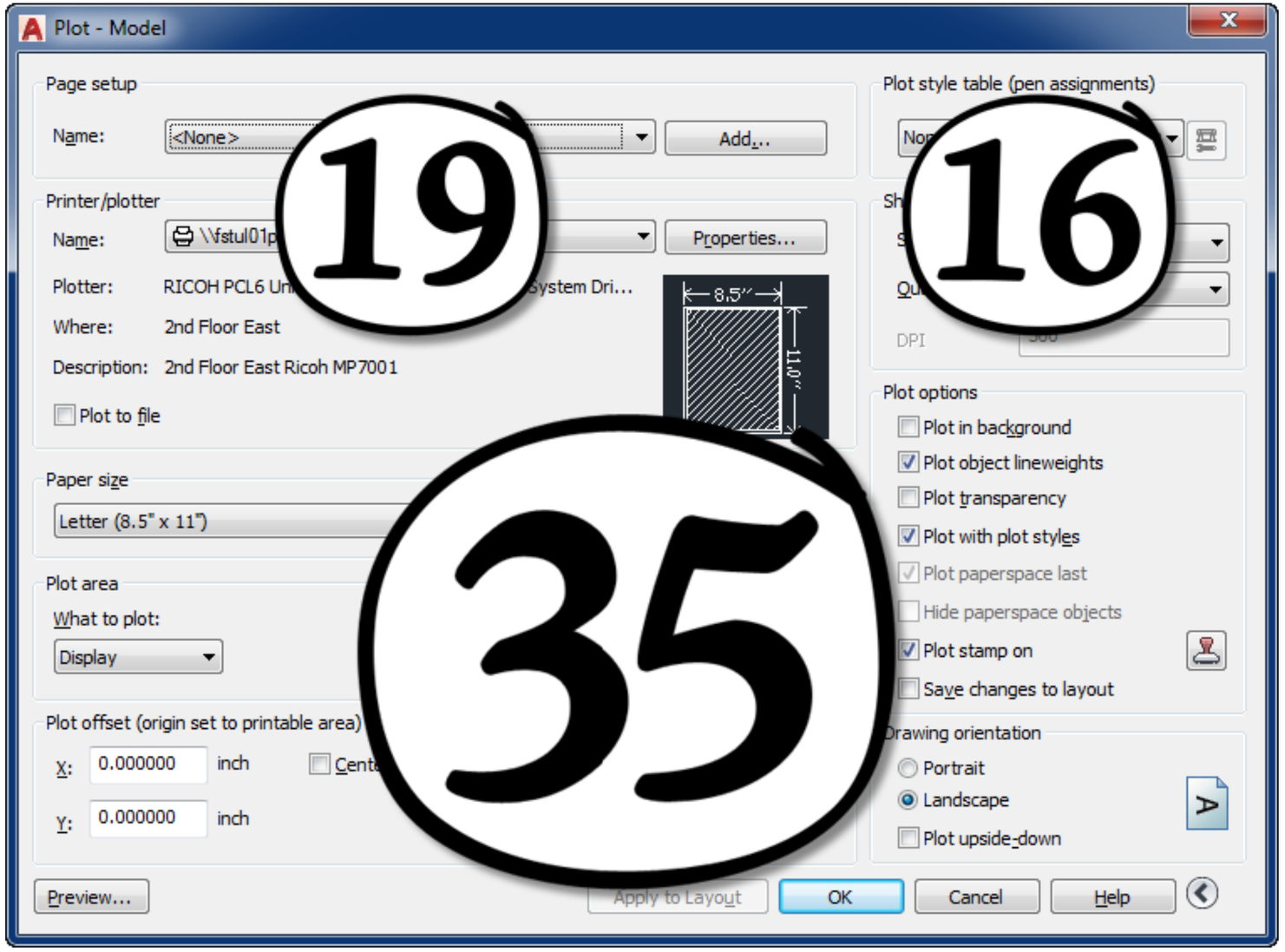

THE SET-UP

Page Setups: The Problem

- DISCLAIMER: Named Page Setups are great! They're a vast improvement over the old system (pre-AutoCAD 2005).
- Page Setups are saved in the .dwg file.
- Page setup *HPLaserJetPro* in drawing **A**, may not match *HPLaserJetPro* in drawing **B**.

```
_$ (eq (getPS "DWG_A" "HPLaserJetPro") (getPS "DWG_B" "HPLaserJetPro"))
   nil
```

Plotting Controls



My Scenario

- Multiple projects / Multiple clients
- Most use our Title Block and standards, some don’t.
- Different .ctb’s per discipline or drawing type, plus client .ctb’s
- Four printers to choose from, plus a plotter, plus PDF or DWF

My Scenario

- Embraced and used Sheet Set Manager
- Used a Page Setup Override file for printing (a .dwt)
- This was the method our users trusted.
- Soon the disconnect between the override file and the page setups in the sheets happened.

What to do?

- In a perfect world, page setups wouldn’t live in .dwg files.
- I needed to create a near perfect world. Where the page setups in the drawings would always match those in the override file.
- I wanted the whole thing to be transparent to the user, and require no interaction on their part.

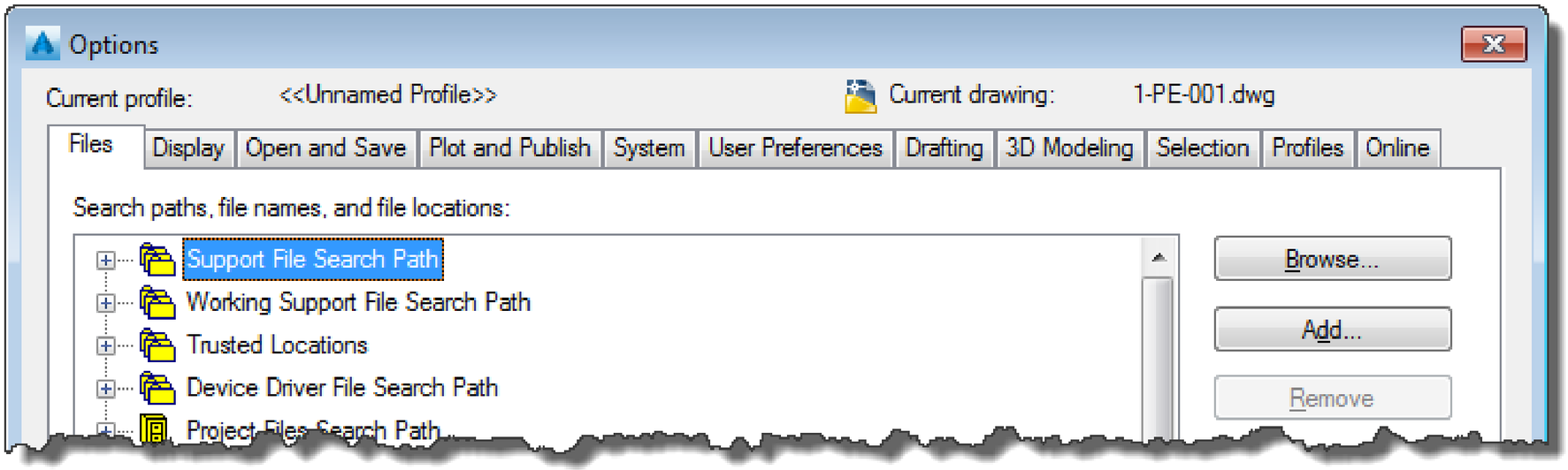

THE TALE

The Concept

- It needs to load every time a drawing file is opened.
  - ACADDOC.lsp
- Analyze the file being opened. What is it? Where is it?
  - Essential that you have a standardized folder and file naming convention
- Based in the analysis, find your master page setup.
- Manage the synchronization of the page setups with lisp.
  - Delete the existing, possibly wrong page setups
  - Import the good, standard set from your master file

Breaking down the steps (1)

- ACADDOC.lsp = Every time a file is opened.
- ACAD.lsp = When a new AutoCAD session is started.



Breaking down the steps (2)

- Analyze the file

```
(getvar "dwgprefix")
(getvar "dwgname")
```

P:\FDB\1650-TUL\IND\ClientName\316363 AU_CLASS\20_DESGN\40_CAD\PIPING

L:\Industrial\Projects\33841 Large Mart\201 Design\20 AutoCAD\Arch

B:\2016\Retail\Buy More\Burbank\91501\Eng\CAD\MEP

W:\ …

G:\ …

Z:\Davids team\zombie apocalypse1

E:\USB20FB\My Files

C:\Users\mayfieldf\Documents\Work Stuff\CAD\

Breaking down the steps (3)

- Build a path to your master page setups.
  - This could be very easy or quite complex.
  - You situation will be different!

```
(defun *dwgPath* () (strcase (getvar "dwgprefix")))

(defun *isBuyMore?* () (wcmatch (*dwgpath*) "*BUY MORE*"))
(defun *isLargeMart?* () (wcmatch (*dwgpath*) "*LARGE MART*"))
(defun *OrangeOrange?* () (wcmatch (*dwgpath*) "*ORANGEORANGE*"))
```

Breaking down the steps (3)

- Build a path to your master page setups.
  - This could be very easy or quite complex.
  - You situation will be different!

```
(cond
  ((*isBuyMore?*)
   (setq
     PageSetupFile
      "B:\\2016\\RETAIL\\BUY MORE\\SUPPORT\\BUYMOREPAGESETUPS.dwg"))
  ((*isLargeMart?*)
   (setq
     PageSetupFile
      "L:\\2016\\RETAIL\\LARGE MART\\SUPPORT\\LARGEMARTPAGESETUPS.dwg"))
  ((*OrangeOrange?*)
   (setq
     PageSetupFile
      "O:\\2016\\RETAIL\\ORANGEORANGE\\SUPPORT\\ORANGE2XPAGESETUPS.dwg")))
```

Breaking down the steps (3)

- Build a path to your master page setups.
  - This could be very easy or quite complex.
  - You situation will be different!

```
(defun *dwgPath* () (strcase (getvar "dwgprefix")))
(defun *dwgName* () (strcase (getvar "dwgname")))
(defun *dwgDriveLtr* () (substr (*dwgPath*) 1 1))
```

Breaking down the steps (3)

- Build a path to your master page setups.
  - This could be very easy or quite complex.
  - You situation will be different!

```
(if (= (*dwgDriveLtr*) "B")
  ;; Then, we know it's a Buy More project
  (setq
    PageSetupFile
     "B:\\2016\\RETAIL\\BUY MORE\\SUPPORT\\BUYMOREPAGESETUPS.dwg")

  ;; Else, it has to be Large Mart (our only two clients)
  (setq
    PageSetupFile
     "L:\\2016\\RETAIL\\LARGE MART\\SUPPORT\\LARGEMARTPAGESETUPS.dwg"))
```

P:\FDB\1650-TUL\IND\ClientName\316363 AU_CLASS\20_DESGN\40_CAD\PIPING

L:\Industrial\Projects\33841 Large Mart\201 Design\20 AutoCAD\Arch

B:\2016\Retail\Buy More\Burbank\91501\Eng\CAD\MEP

W:\ …

G:\ …

Z:\Davids team\zombie apocalypse1

E:\USB20FB\My Files

C:\Users\mayfieldf\Documents\Work Stuff\CAD\

Breaking down the steps (3)

- Build a path to your master page setups.
  - This could be very easy or quite complex.
  - You situation will be different!

```
(defun *dwgPath* () (strcase (getvar "dwgprefix")))
```

Breaking down the steps (3)

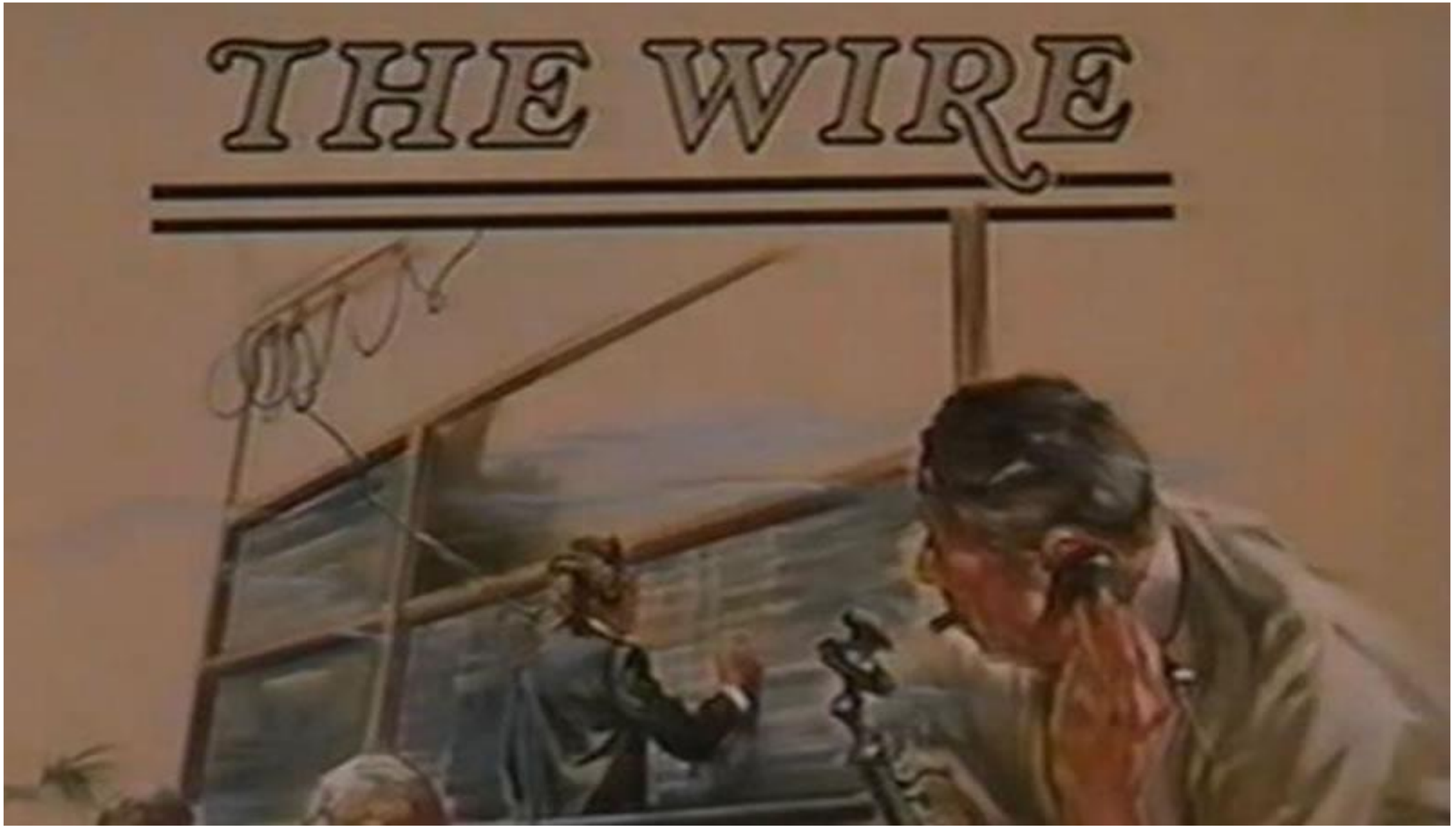
- Build a path to your master page setups.
  - This could be very easy or quite complex.
  - You situation will be different!

```
;; Set the path of the file being opened to the var 'dpath'.
(setq dpath (*dwgPath*))
> "B:\\2016\\Retail\\Buy More\\Burbank\\91501\\Eng\\CAD\\MEP\\"

;; Find the start position of the folder where the Page Setup file resides:
(setq startPos (vl-string-search "CAD\\" dpath));; Note CAD\\ is 4 digits.
> 42

;; From the front of the search string, go 4 digits past where CAD is found.
(setq PageSetupPath (substr dpath 1 (+ 4 startPos)))
> "B:\\2016\\Retail\\Buy More\\Burbank\\91501\\Eng\\CAD\\"

;; Concatenate the path and file together, while also making sure it exists.
(setq PageSetupFile (findfile (strcat PageSetupPath "MasterPageSetup.dwt")))
> "B:\\2016\\Retail\\Buy More\\Burbank\\91501\\Eng\\CAD\\MasterPageSetup.dwt"
```

Breaking down the steps (4)

- Finally, simply delete the existing page setups, and import the good.

```
(vl-load-com)

(vlax-for
  ps
  (vla-get-plotconfigurations
    (vla-get-activedocument (vlax-get-acad-object)))
  (vla-delete ps))


(setvar "CMDECHO" 0)

(command "-PSETUPIN" PageSetupFile "*")

;; say no to redefining if prompted.
(while (not (zerop (getvar "CMDACTIVE")))
  (command "N"))

(setvar "CMDECHO" 1)
```


THE WIRE

Interlude: Organizing your code

- ACADDOC.lsp

```
(setvar "XLOADCTL" 2)
(setvar "SAVETIME" 10)
(setvar "REMEMBERFOLDERS" 1)


;; Set SAVETYPE to 2013 format
(if (setq opensaveObj
      (vla-get-opensave
        (vla-get-preferences (vlax-get-acad-object))))
  (vla-put-saveastype opensaveObj 60))
(if opensaveObj (vlax-release-object opensaveObj))


(autoload "northingeasting" '("NE"))
```

Interlude: Organizing your code

- ACADDOC.lsp

```
(load "managePageSetups.lsp")
(deleteAndImportPageSetups)


(vl-load-com)
(acad-push-dbmod)

(load "globals.lsp")
 :
(code)
 :
(acad-pop-dbmod)
```

Interlude: Organizing your code

- GLOBALS.lsp

  Sets variables and functions that are used in other routines.

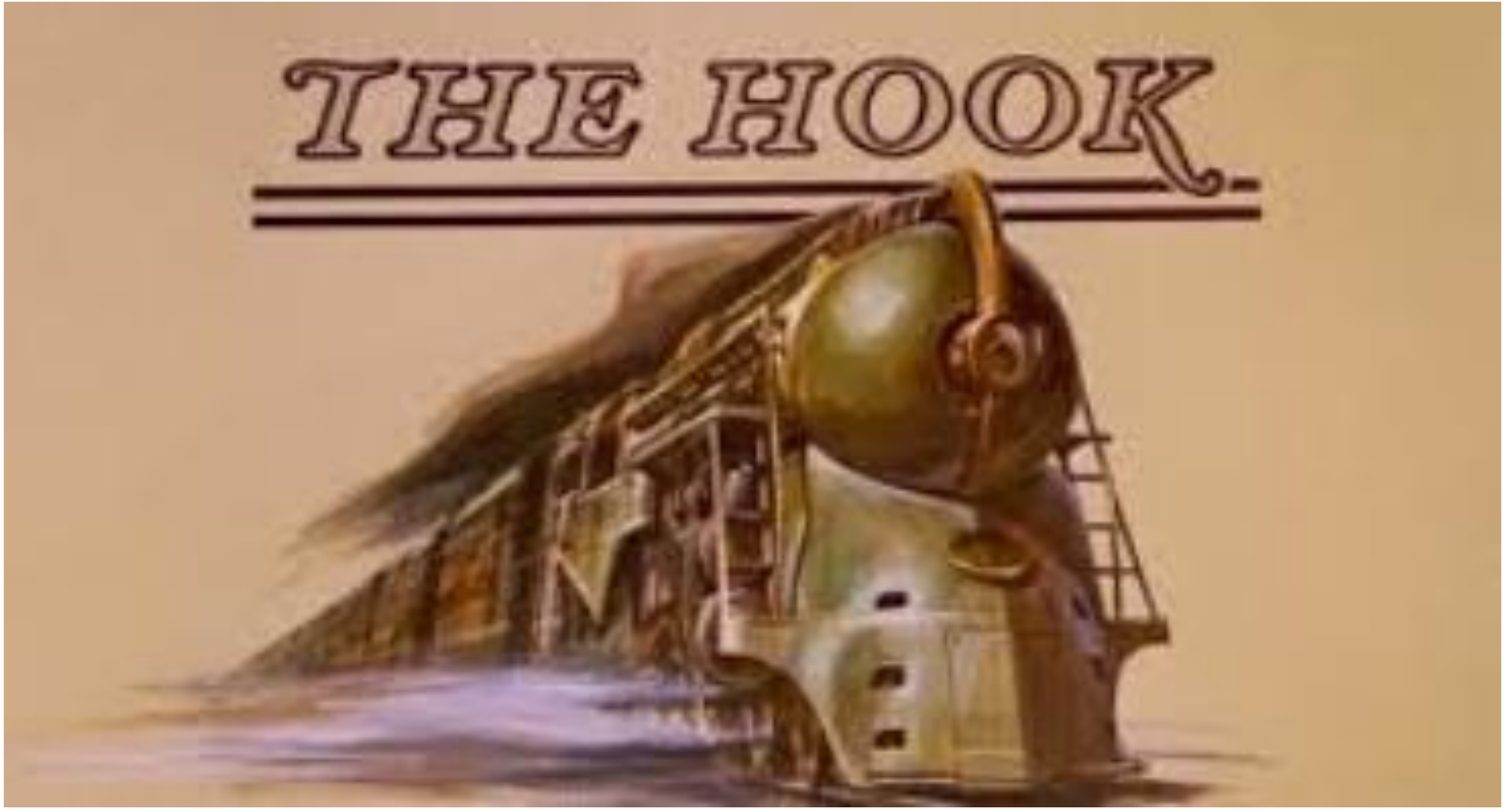
```
(defun *dwgPath* () (strcase (getvar "dwgprefix")))
(defun *dwgName* () (strcase (getvar "dwgname")))
(defun *dwgDriveLtr* () (substr (*dwgPath*) 1 1))


(defun *is-Houston?* () (wcmatch (*dwgPath*) "*2203-HOU*"))
(defun *is-StLouis?* () (wcmatch (*dwgPath*) "*1648-STL*"))


(setq *acadVer* () (getvar "acadver"))
(defun *is-2016?* () (= (substr *acadVer* 1 4) "20.1"))
(defun *is-2017?* () (= (substr *acadVer* 1 4) "21.0"))
```

Interlude: Organizing your code

- GLOBALS.lsp

  Sets variables and functions that are used in other routines.

```
(defun *is-Pipe?* () (wcmatch (*dwgPath*) "*PIPING*"))
(defun *is-Struct?* () (wcmatch (*dwgPath*) "*STRUCTURAL*"))
(defun *is-Elec?* () (wcmatch (*dwgPath*) "*ELECTRICAL*"))
(defun *is-Iso?* () (wcmatch (*dwgPath*) "*ISOMETRICS*"))


(defun *isPlant3D?* ()
  (and (getvar "plantcontentfolder")
       (*is-2015?*)))

(defun *isCADWorx?* ()
  (or (member "cadworxplant2013.arx" (arx))
      (member "cadworxplant2015.arx" (arx))))
```


THE HOOK

Real World Examples - or - What you might run into…

Easy Peasy

- Every drawing will always use the same page setup or set of page setups.
- No need to analyze anything.
- Simply delete and replace every time.

Easy Peasy

- Loop to delete all Page Setups:

```
(vl-load-com)

(vlax-for
  ps
  (vla-get-plotconfigurations
    (vla-get-activedocument (vlax-get-acad-object)))
  (vla-delete ps))
```

- Import all Page Setups from a file:

```
(setvar "CMDECHO" 0)

(command "-PSETUPIN" PageSetupFile "*")

;; say no to redefining if prompted.
(while (not (zerop (getvar "CMDACTIVE")))
  (command "N"))

(setvar "CMDECHO" 1)
```

The Next Best Thing…

- Multiple clients or programs, but each always uses a single set of page setups.
- The client page setup file resides in a dedicated support folder.
- Your analysis may simply be a drive letter:

```
(defun *dwgPath* () (strcase (getvar "dwgprefix")))
(defun *dwgDriveLtr* () (substr (*dwgpath*) 1 1))
```

- Where “B” is the “Buy More” client and “L” is Large Mart.

The Next Best Thing…

- Multiple clients or programs, but each always uses a single set of page setups.
- The client page setup file resides in a dedicated support folder.
- Or you may need to find a key string in the path:

```
(defun *isBuyMore?* () (wcmatch (*dwgpath*) "*Buy More*"))
(defun *isLargeMart?* () (wcmatch (*dwgpath*) "*Large Mart*"))
(defun *OrangeOrange?* () (wcmatch (*dwgpath*) "*OrangeOrange*"))
```

- Anything more than two, use a cond statement.

The Next Best Thing…

- Tip: Group conditions by most probable to least probable

```
(cond
  ((*isBuyMore?*)
   (setq
     PageSetupFile
      "B:\\2016\\Retail\\Buy More\\Support\\BuyMorePageSetups.dwg"))
  ((*isLargeMart?*)
   (setq
     PageSetupFile
      "L:\\2016\\Retail\\Large Mart\\Support\\LargeMartPageSetups.dwg"))
  ((*OrangeOrange?*)
   (setq
     PageSetupFile
      "O:\\2016\\Retail\\Orange2x\\Support\\Orange2xPageSetups.dwg")))
```

- Anything more than two, use a cond statement.

A Bit More Complicated

- Multiple projects, multiple clients – company and client page setups.

B:\2016\Retail\Buy More\Burbank\91501\Eng\CAD\MEP
B:\2015\Retail\Large Mart\Tulsa\74103\Eng\CAD\Arch
B:\2015\Retail\JR Nickels\Buffalo\14212\Eng\CAD\Struct
F:\2016\Food\Breezy's\Austin\78710\Eng\CAD\Civil

- As the CAD Mgr, you know what folder the master page setup file should be in. In our examples above… "\CAD\"

A Bit More Complicated

- Simply parse the path down to where you page setup file resides.

```
;; Set the path of the file being opened to the var 'dpath'.
(setq dpath (*dwgPath*))
> "B:\\2016\\RETAIL\\BUY MORE\\BURBANK\\91501\\ENG\\CAD\\MEP\\"

(if (and (wcmatch dpath "*\\ENG\\CAD\\*")
         (setq startPos (vl-string-search "CAD\\" dpath)))
> T

  (progn
    (setq PageSetupPath (substr dpath 1 (+ 4 startPos)))
> "B:\\2016\\RETAIL\\BUY MORE\\BURBANK\\91501\\ENG\\CAD\\"

    ;; Concatenate the path and file together,
    ;; while also making sure it exists.
    (setq PageSetupFile
           (findfile (strcat PageSetupPath "MasterPageSetup.dwt")))))
> "B:\\2016\\RETAIL\\BUY MORE\\BURBANK\\91501\\ENG\\CAD\\MasterPageSetup.dwt"
```

Welcome to My World

- Wherein you have any or all of the previous scenarios, along with further complications.
  - Much more detail in the handout

1. You may need to bypass or load a different set of page setups for certain types of files or disciplines.

2. There may be page setups you don’t want to delete.

3. One or more projects may be an outlier – they just don’t fit your standard.

1 – Bypass or load something different.

- You may need to bypass a certain file type or discipline…

```
(defun *is-Iso?* () (wcmatch (*dwgPath*) "*ISOMETRICS*"))


(if (and (wcmatch dpath "*\\ENG\\CAD\\*")
         (not (*is-Iso?*))
         (setq startPos (vl-string-search "CAD\\" dpath)))
```

1 – Bypass or load something different.

- You may need to bypass a certain file type or discipline…

```
(defun *is-Iso?* () (wcmatch (*dwgPath*) "*ISOMETRICS*"))
```

```
(if (and (wcmatch dpath "*\\ENG\\CAD\\*")
         (not (*is-Iso?*))
         (setq startPos (vl-string-search "CAD\\" dpath)))
```

- Based on the file, you may just load a different set…

```
(if (*is-Iso?*)
  (setq PageSetupFile (findfile (strcat PageSetupPath "iso-template.dwt")))
  (setq PageSetupFile (findfile (strcat PageSetupPath "MasterPageSetup.dwt"))))
```

1 – Bypass or load something different.

- You may need to bypass a certain file type or discipline…
  - Another example

```
(defun *is-Houston?* () (wcmatch (*dwgPath*) "*2203-HOU*"))
(defun *is-StLouis?* () (wcmatch (*dwgPath*) "*1648-STL*"))
```

```
(if (not (or (*is-Houston?*)
             (*is-StLouis?*)))
  (progn
    ;; ... continue with your code here.
    ))
```

2 – You can’t just delete all the existing Page Setups.

- Analyze each page setup on the fly… three examples

```
(setq *MasterPageSetupList* '("HPLaserJetPro" "OCEPlotWave360"
                              "EpsonMX80DotMatrix" "Gutenberg1440"))

(vlax-for
  ps
  (vla-get-plotconfigurations
    (vla-get-activedocument (vlax-get-acad-object)))

  ;; get the current page setup name, and check it against the list
  (if (member (vla-get-name ps) *MasterPageSetupList*)
    (vla-delete ps)))
```

2 – You can’t just delete all the existing Page Setups.

- Analyze each page setup on the fly…
  - Example two

```
(vlax-for
  ps
  (vla-get-plotconfigurations
    (vla-get-activedocument (vlax-get-acad-object)))
  (if (not (wcmatch (vla-get-name ps) "*HOU*,*STL*"))
    (vla-delete ps)))
```

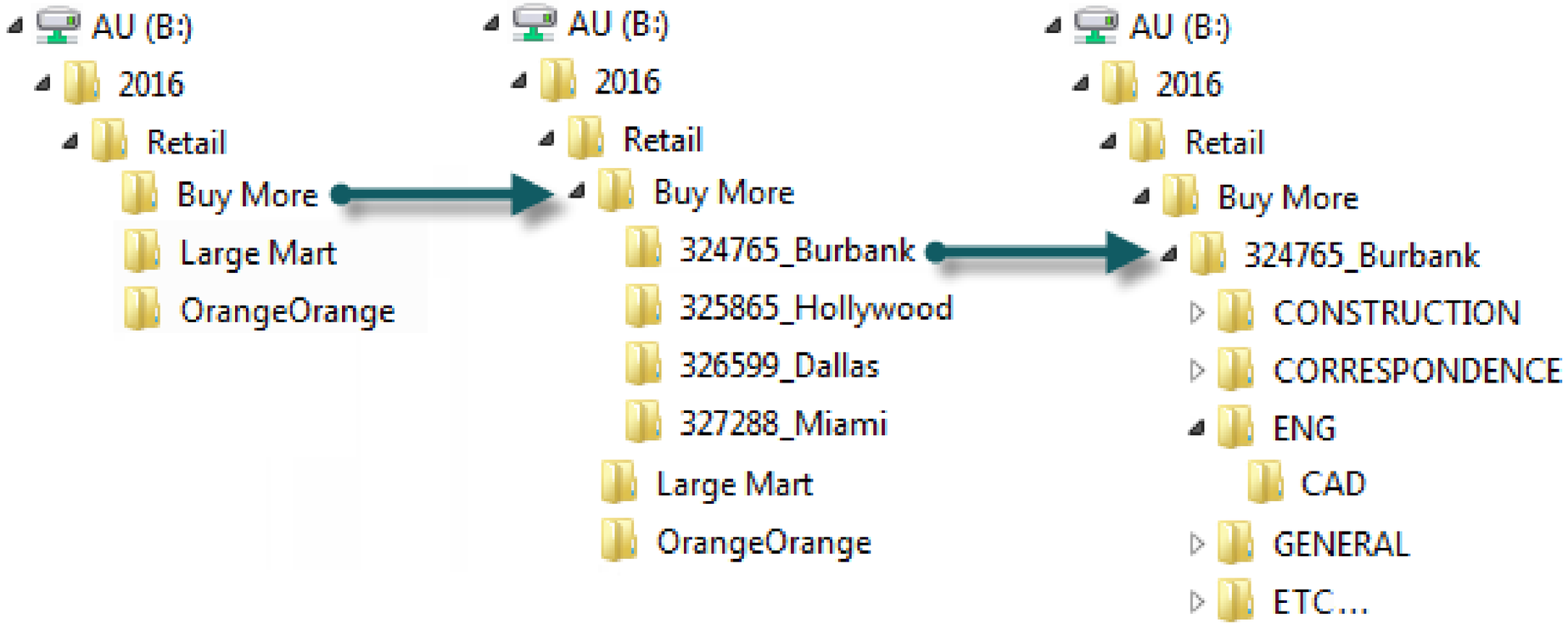
2 – You can’t just delete all the existing Page Setups.

- Analyze each page setup on the fly…
  - Example three

```
(vlax-for
  ps
  (vla-get-plotconfigurations
    (vla-get-activedocument (vlax-get-acad-object)))
  (setq deviceName (vla-get-configName ps))
  (if (wcmatch deviceName "*PTUL*")
    (vla-delete ps)))
```

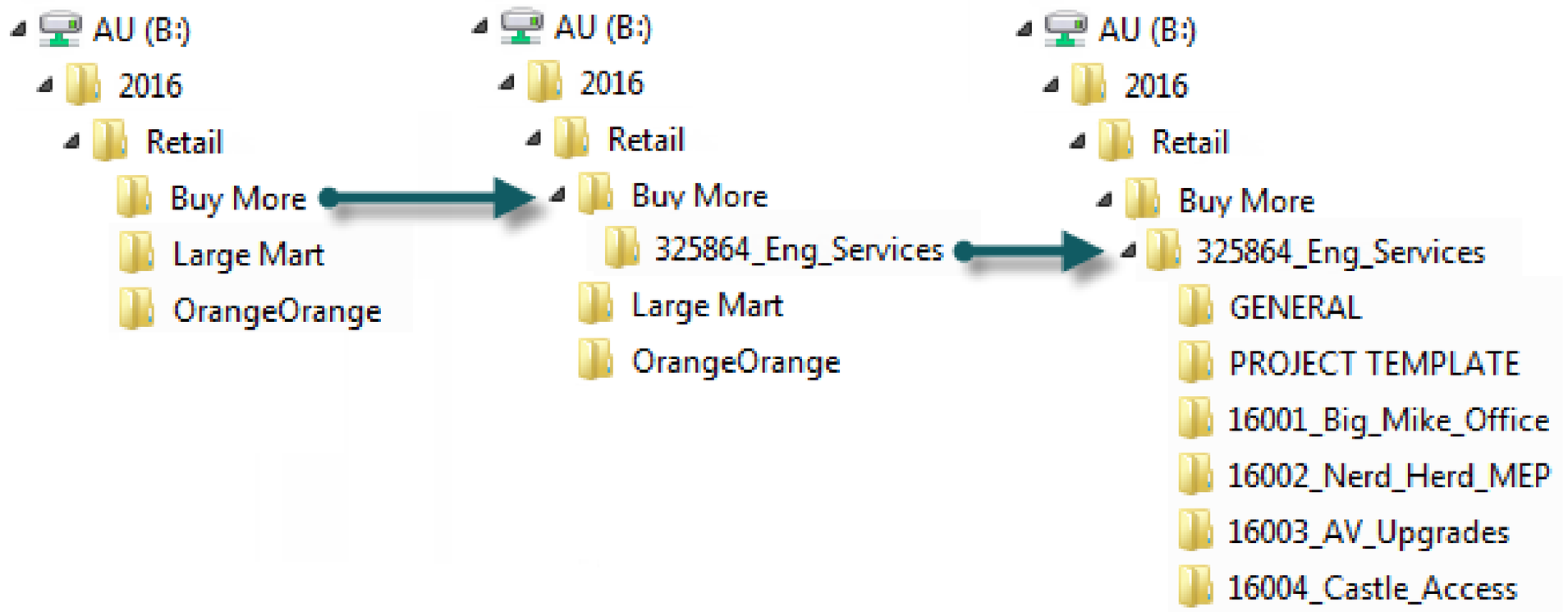
3 – The Outlier. Again, ‘cause there’s always one.

- A typical project folder structure – expanded…



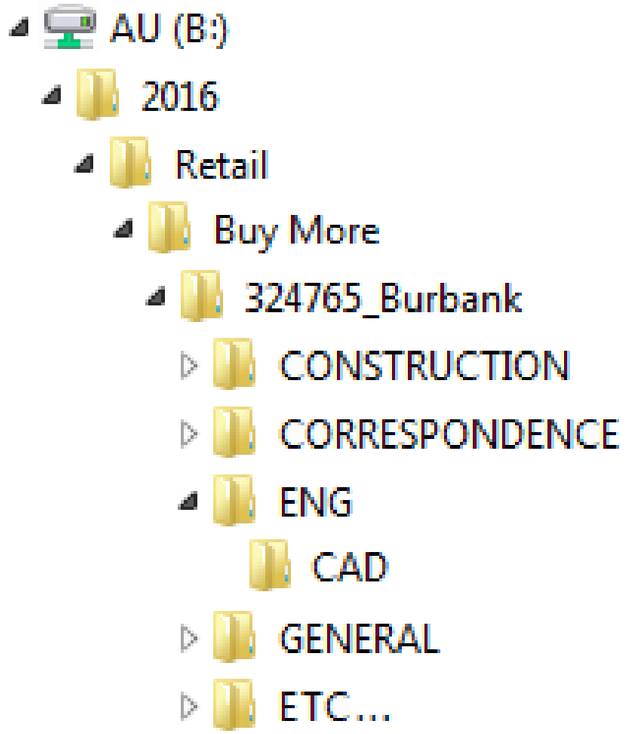
3 – The Outlier. Again, ‘cause there’s always one.

- The Outlier



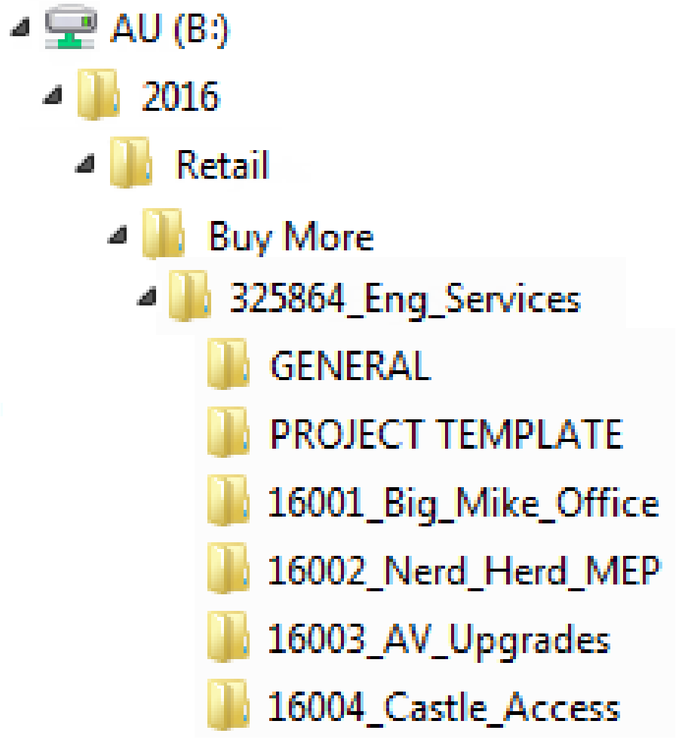
3 – The Outlier. Again, ‘cause there’s always one.

Normal Project



VS.

Projects within a project



3 – The Outlier. Again, ‘cause there’s always one.

```
(defun *isBuyMoreIDIQ?* () (vl-string-search "325864" dpath))


;; Here we use the value returned by *isBuyMoreIDIQ?*
(if (not (setq startPos (*isBuyMoreIDIQ?*)))

  ;; Then, if not a Buy More IDIQ, and it is a standard path...
  (if (and (wcmatch dpath "*\\ENG\\CAD\\*")
           (setq startPos (vl-string-search "CAD\\" dpath)))

    ;; Do your normal parsing here...
    (progn
      ;; find a common string in std. paths, and get its starting position.
      (setq startPos (vl-string-search "CAD\\" dpath))
      ;; from the front of the full path, go 4 digits past from "CAD".
      (setq *projPath* (substr dpath 1 (+ 4 startPos)))))

  ;; Else, it's the outlier IDIQ Buy More Services project.
  ;; Since the predicate returns the position, just add the needed folder.
  (setq *projPath* (strcat (substr dpath 1 (+ 20 startPos)) "GENERAL\\")))
```


THE STING

Thinking outside the box …

- Our focus today has been on Page Setups.
- Maintaining them, and importing them is the easy part.
- The key is analyzing the file.
- What else can you learn about the it?
- What else can you do when you know what it is?

Thinking outside the box …

- What is the extension?
  - DWG
  - DWT
  - DWS
    - You may only want to run your code on DWGs and bypass the others.
- Does the file name tell you anything?
- Is there anything embedded in the file itself?

Thinking outside the box …

- What discipline does it belong to?
  - We've already used Isometrics as an example.
  - What else?
    - Architectural
    - Structural
    - Electrical
    - Plumbing
    - Mechanical
    - Civil
    - Etc.
  - You may write code to return the discipline…

```
(defun *get-disc*  (/ dp disc)
  (setq dp (*dwgPath*))
  (if
    (cond
      ((vl-string-search "\\ARCH\\" dp)
       (setq disc "Arch"))
      ((vl-string-search "\\STRUCT\\" dp)
       (setq disc "Struct"))
      ((vl-string-search "\\ELEC\\" dp)
       (setq disc "Elec"))
      ((vl-string-search "\\MECH\\" dp)
       (setq disc "Mech"))
      ((vl-string-search "\\PLUMB\\" dp)
       (setq disc "Plumb"))
      ((vl-string-search "\\CIVIL\\" dp)
       (setq disc "Civil"))))
  disc)
```

Thinking outside the box …

- Now you know the discipline… what can you do?
  - Verify or load the proper .DWS standards file
  - Load the corresponding Tool Palette
  - Load or verify layers
  - Load or verify styles
    - Text
    - Dimensions
    - Multi Leaders
    - Tables
  - Set defaults in custom program’s dialog

Thinking outside the box …

- A case study in… a particular project, and lazy users.
  - The problem: 3D piping models saved with elements turned off.
    - 3rd party software tools.
    - Visual style for performance
    - Multiple MS viewports
    - *In addition to being a pain for others, it affected our Navisworks files*
  - My Analysis:
    - What was the client / project?
    - Was 3rd party (CADWorx) running?
    - Was it a model file? (folder)

Thinking outside the box …

- The solution: a custom CLOSE command in ACADDOC.lsp

```
(command "._undefine" "._CLOSE")

(defun c:CLOSE (/)
  (if (and
        (*isLargeMart?*)
        (*isCADWorx?*)
        (*is3DModel?*)
        (*isTulsa?*))
    (progn
      (command "-vports" "SI")
      (command "vscurrent" 2) ;2D wireframe
      (command "_.script" "LargeMart_CLOSE.scr") ;; performs a CADWorx LINEISOLATE to Show All

      (vla-sendcommand
        (vla-get-activedocument
          (vlax-get-acad-object)) ".close "))

    (progn
      (vla-sendcommand
        (vla-get-activedocument
          (vlax-get-acad-object)) ".close ")))
  (princ))
```

Conclusion

How did I do?

- Your class feedback is critical. Fill out a **class survey** now.
- Use the AU mobile app or fill out a class survey online.
- Give feedback after each session.
- AU speakers will get feedback in real-time.
- **Your feedback results in better classes and a better AU experience.**

More Questions? Visit the AU Answer Bar

- Seek answers to all of your technical product questions by visiting the **Answer Bar**.
- Open daily from **8am-6pm Tuesday** and **Wednesday**; **8am-4:30pm Thursday**.
- Located outside **Hall C, Level 2**.
- Meet Autodesk developers, testers, & support engineers ready to help with your most challenging technical questions.