

Jul/Aug 2004



AUGIWorld

The Official Publication of Autodesk User Group International

A Discussion with CAD Managers

*For today's CAD managers,
it appears that one sure
constant is change*

- *On Formulas and Autodesk Revit Families*
- *Modifying AutoCAD's Context Menus*
- *Interior Lighting with Photometric Data*

www.AUGI.com

*Impressive AutoCAD
presentations quickly*

Sooner or Later?



Do the colors of your AutoCAD plots look different in print than they do on screen? M-Color 8 is the solution to color control problems. It saves colors in a standardized device-independent color space, giving you consistent colors with all your monitors and printers.

With M-Color it's quick and easy to create impressive and colorful plots from AutoCAD drawings. You can add translucent fills, gradients, and bitmap texturing with flexible options. M-Color allows you to apply graphic effects to whole AutoCAD layers and save color configurations for re-use with other drawings - an essential feature for productive CAD work. Scheduled printing enables you to process heavier jobs during off-hours. You can save your work in PDF, PostScript and EPS formats as well as all common bitmap formats such as TIFF and JPEG. In addition, M-Color 8 plots are easily exported to most page layout programs.

M-Color 8 supports both AutoCAD and AutoCAD LT.

NEW IN M-COLOR 8:

Color matching • Scheduled printing • Support for ECW and MrSID images • Rotate and modify texture fills • Usability improvements including undo/redo, font management and visual page layout setup • Password-protected MCL files



ORDER :

Motive Systems: 1-800-224-4074. www.motivesys.com • Microsoft Resources Corporation: 1-888-768-7568.
M-Color is a trademark of Motive Systems (e-mail: motivesys@motivesys.com). AutoCAD is a registered trademark of Autodesk, Inc.

Try M-Color for free at www.motivesys.com

Remember your first encounter with inspired design?



It takes more than inspiration to be a world-class engineer or exceptional architect. We make it easier to stay one.

To keep pace with the demands of your profession, as well as innovation around the world, contact your local Autodesk® Authorized Training Center (ATC®).

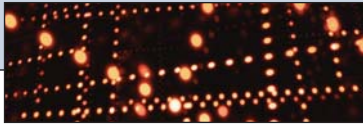
We teach AutoCAD®—and much, much more. Ask about classes on Autodesk® Architectural Desktop, Autodesk® Civil Design, Autodesk Inventor®, and Autodesk® Revit®. An ATC can help you stay on top of it all. To find the Autodesk Authorized Training Center nearest you, visit www.autodesk.com/atc, or call 1-800-964-6432.

Let Autodesk take care of your training.
So you can take care of building our world.

Autodesk, AutoCAD, Autodesk Inventor, ATC, and Revit are registered trademarks of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. (c) 2004 Autodesk, Inc. All rights reserved.

autodesk®
authorized training center

Table of Contents



feature

A Discussion with CAD Managers

*For today's CAD managers,
it appears that one sure
constant is change*

6



Editorial

Managing Editor

Marilyn Law
marilyn.law@augiworld.com

Technical Editor

David Harrington
david.harrington@augiworld.com

Contributing Editors

R. Robert Bell
Doug Broad
Dwayne D. Ellis
Christopher Fox
Robert Green
David Kingsley

Production

Tim Varnau
tim.varnau@augiworld.com

Advertising

Dan Teeter
dan.teeter@augiworld.com

Controller

Larry Teeter
larry.teeter@augiworld.com

Publisher

Rich Uphus
rich.uphus@augiworld.com

AUGI Board of Directors, 2004

Marv Muston, *President*
David Kingsley, *Senior Vice President*
John Moran, *Junior Vice President*
Richard Binning, *Secretary*
R. Robert Bell
Michael DeGraw
Yoshi Honda
Jill King

AUGIWorld is a trademark of Autodesk User Group International (AUGI) www.augi.com. AUGI and Autodesk are registered trademarks of Autodesk, Inc. in the U.S. and other countries. All other brand names, product names, or trademarks belong to their respective holders. *AUGIWorld* is published by Solidvapor, Inc., 19601 James Road, Noblesville, IN 46060.

Publications Mail Agreement # 40733013
Return Undeliverable Canadian Addresses to
Station A
PO Box 54
Windsor, ON N9A 6J5
Email: cpcreturns@wdsmail.com

**BPA International membership applied for
August 2002.**

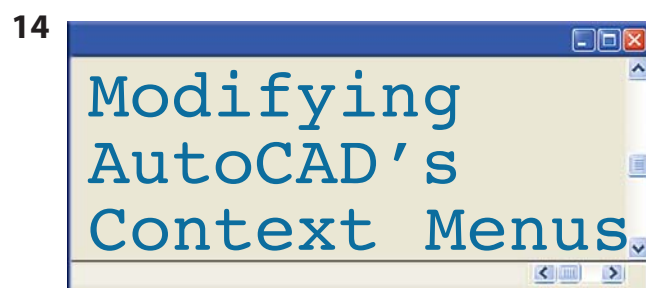
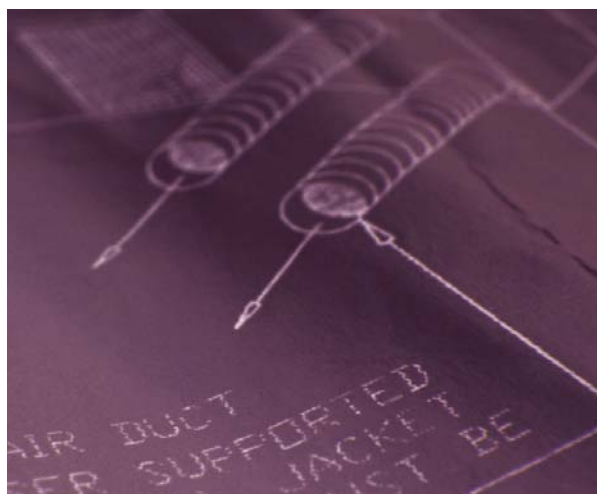
© Copyright 2004 Solidvapor, Inc. All rights reserved.

4 The CAD Manager

Do your lines have weight?

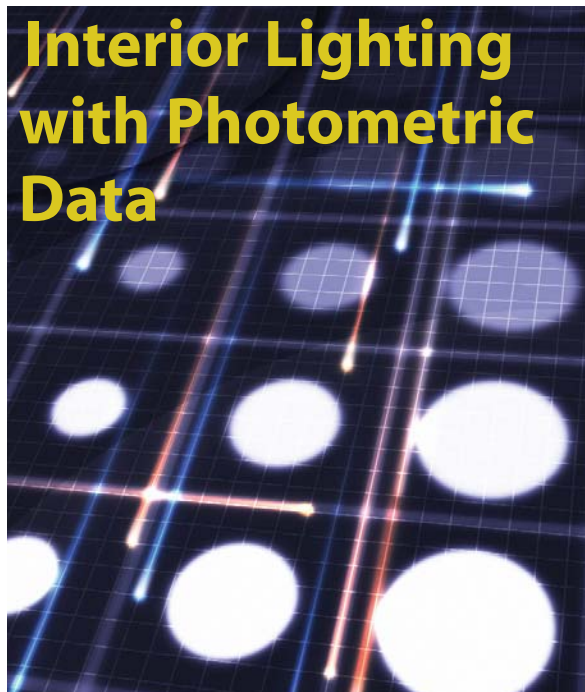


12 On Formulas and Autodesk Revit Families

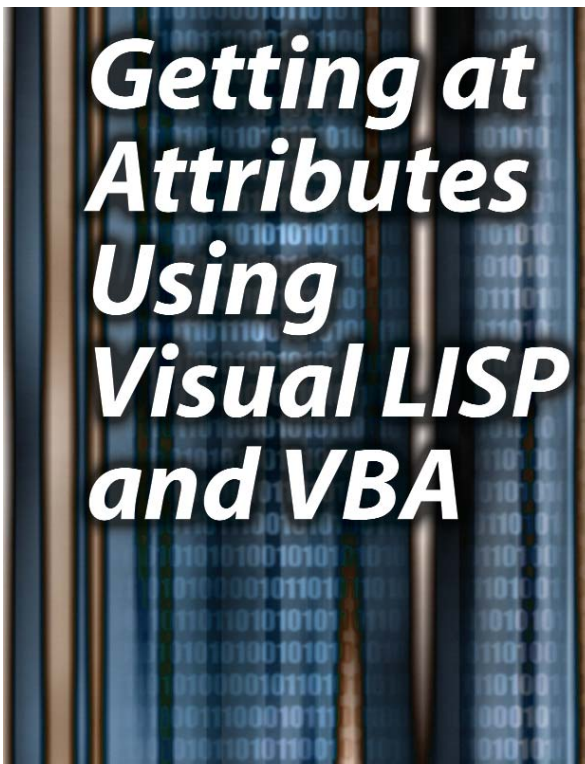


16 Using AutoCAD and Architectural Desktop Together

20 Interior Lighting with Photometric Data



22 Getting at Attributes Using Visual LISP and VBA



28 On The Back Page

Making the Old New Again

The Cad Manager



David Harrington

Do Your Lines Have Weight?

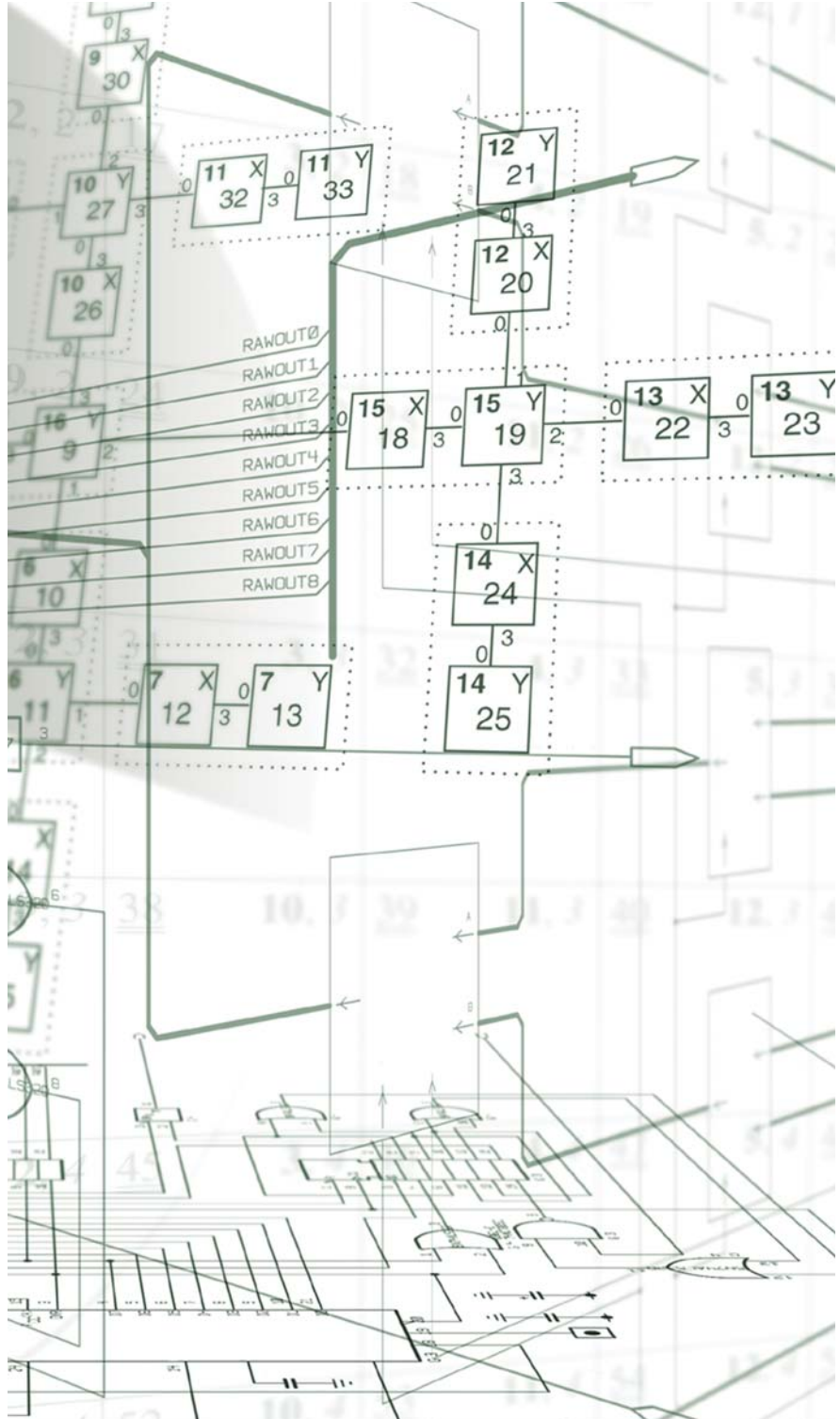
It is very likely that if you talked to your spouse about linewidths they would either have no clue or wonder how very little a printed line actually weighs and why anyone would care.

Then there are those who interact with printed drawings. But they may have only a cursory understanding of linewidths – they know a fat line from a thin line and probably know that type of delineation means something specific.

This then leaves two more: drafters and AutoCAD® users. To a drafter, a linewidth is given to a particular pen stroke of a drawing to make it stand out from nearby elements. But to an AutoCAD user, a linewidth is a property value applied to a plot style and thereby to a layer and even onto an object. The end result of this is generally only realized when the drawing is plotted to paper. Then lo and behold, fat and thin lines. And ideally, arranged and assigned well that then conveys the information required in more pleasing style.

Is There A Problem?

I sense a problem. The problem I sense is the difficulty for people to clearly understand the linewidth that is being applied since it is somewhat difficult to see on-screen. In AutoCAD you can zoom in on something beyond the level of reproducing it on paper. You can also make things so huge you could never plot either. I recall back to a DWG I made of the solar system. When complete and plotted on an E size drawing, I had a small circle and a bunch of dots. Not helpful. I had to embellish the scale of the objects to allow a good level of comparison. Lineweights are the same.



India Ink or Bust!

Prior to the AutoCAD revolution most drafters used ink-based technical pens that came in widths ranging from 0.013mm to over 1.0mm wide. My recollection is that I had 10 pen sizes in my quiver of tools. I would use just five different sizes most of the time. But even that is a bit misleading since I could tweak a given line to be wider by placing additional strokes of the pen. I used the size I needed to convey the intensity I wanted, and then went further as required.

Pen Plotting Machine

In 1987 my company purchased an interesting device to go along with cool AutoCAD seats. Sounding a bit like a muscle car, the Calcomp 1043GT had a turret that would house up to eight pens. It also had a robotic type contraption that would grab one pen at a time and lay down lines of ink at lightning speed. It was cool to watch but it wasn't all that intelligent. It didn't know if a pen ran out of ink or got clogged. Good thing we were watching it plot since it often took 45 minutes for one plot!



The End of STYLE

The addition of AutoCAD and our pen plotter was the first encroachment into the demise of drafting style. Not only did we lose the number of available pens but the ability to tweak a line was lost as well. If nothing else, the Calcomp was consistent. A plot by me looked the same plotted by another. And boy was it fast! One effect we didn't lose was poche. To do shading we would place decorative adhesive film and trim out the excess. Alternatively, we would use a red pencil and manually shade the reverse side of a mylar drawing. Need the tone a bit darker? Press harder. When we plotted we were never actually done. We simply needed to get the DWG onto a drawing where it could be revised and finished.

Ink Spitting Machine

Our next plotter was an HP DesignJet 600. This little device was even faster than the 1043GT. It measured performance on how fast the paper came out rather than how fast a single line could be drawn, but could plot a DWG in about five minutes. However, it was not all golden. The inks used were not real India ink; it wasn't all

that dark. So when we did manual changes there were dramatic differences in appearance. As for the poche, we did finally have a solution for shading since the plotter could do any number of gradients and tone levels. The interesting thing is that even though we now had a plotter where we could do any lineweight, we didn't go back and add those missing pens.



Big Copy Machine

Our next plot device was a Xerox 8830 xerographic plotter and attached scanner. This baby was fast enough that we finally threw away our technical pens. It could crank out three to four plots a minute! Made a mistake? Plot it again. This introduction also coincided with our not plotting on mylar anymore. We got rid of the blueprint machine and just used the scanner portion to create copies. I don't miss the ammonia smell but I do miss the durability of mylar drawings. Another odd effect is that lineweights from our 8830 looked wider. The toner was not all that fine and the resolution was but 400 dpi.



Time for the Fix

So what is the fix? The fix is to stop letting technology interfere in how our draw-

ings look. Whenever you get a new output device, take a look at your supporting standards. See if you need to adjust your settings so that your drawings look the same. If they are already bad, take the initiative to redo them to how they should be. I just recently developed a technical pen based named plot style file. It is called KOH-I-NOOR.STB for obvious reasons.

It does have differences from the National CAD Standards pen table: it has more sizes! I don't know how the NCS committee decided on the palette of pens but why not use what drafters used for decades?

Size Doesn't Matter

In reality, the pen size variation isn't the problem inasmuch as how the sizes you have available are applied. Since in AutoCAD it is difficult to see pen weight, users (and I include myself) tend to assign a size based on what the element is rather than how the element needs to be presented.

For example, we use the same pen width for a slab edge 1/4" scale as we do 1/16" scale. It makes the drafting component in AutoCAD easy enough, get your right layer name and make it the color you know it should be. But shouldn't an element change width depending on how clear the drawing needs to be? Absolutely.


Free Your Mind

Ok, free your pens instead. Try using pen sizes based on how much of a statement the linework needs to make compared to adjoining lines. Take into consideration that closely spaced parallel lines should be thin so that the gap can be seen. Most of us have plotters that can plot any width we need. Increase the number of sizes you have at your disposal and then use them to make your drawings look cleaner and clearer! If you would like a copy of my STB file, just email me. I believe the use of named plot styles as opposed to color dependent plot styles can aid in the ability to make drawings look better. If you are using STBs, drop me an email about how you made the transition. I would love to hear your comments on this process. And as always, feel free to email me your thoughts on others subjects. Be and draw well!

A Discussion With CAD Managers



*For today's CAD managers,
it appears that one sure
constant is change*



What exactly does a CAD manager do? There is no clear answer to that question. CAD manager duties vary from company to company, industry to industry, individual to individual. Not only do the duties vary, but they often change as well, so that a CAD manager's "job description" when he or she first assumes the role often doesn't closely resemble what that job looks like a few years or even a few months later.

Sometimes, changes are forced upon CAD managers; sometimes, managers initiate the changes themselves. John Clauson, CAD Manager at Indak Manufacturing, a Northbrook, Illinois-based manufacturer of parts for the automotive and other industries, cites both technological advances and economics as primary reasons for a change in the duties

To help us get a handle on the role and responsibilities of CAD managers today, we polled a half-dozen AUGI members who are either CAD managers by title or carry out the responsibilities normally associated with CAD management. Our panel members filled out brief questionnaires asking about their top concerns, what they love about the job, what they don't love about the job, and so on.

“...many CAD managers have had to broaden their expertise in other aspects of their businesses and hone their communications skills.”

assumed by CAD managers today. Clauson says, “Some of the major changes, at least in my experience in the automotive market, have been in the areas of the pace and complexity of technological advances and the increased compression of the time available to deliver designs to market. Due to downsizing, I think that many CAD managers have had to broaden their expertise in other aspects of their businesses and hone their communications skills.”

In this article, we present the unvarnished truth about CAD management from the eyes of these CAD managers.

Chief Concerns

When asked about their primary concerns, responses varied, but all of the respondents report that turning out quality work is at the top of their priority lists. Faced with staffing issues that include underskilled or inexperienced employees, our panel nonetheless aims to keep pro-

ductivity up while providing assistance and training to their employees.

One manager finds that educating staff doesn't mean educating only *design* staff. "My number one biggest problem is that no one outside of the CAD users understands what we do. They believe either a computer can do CAD "automatically" or with a paint-type program that we got with a database," he says.

Maintaining CAD standards is a common concern and our panel has taken different approaches to ensuring that standards are met. One manager created a CAD Guide to "show how things are to be done." Another member of our panel notes, "I try to explain the overall idea of CAD standards so when a non-standard situation comes up, my drafting staff will have the tools to make a good judgment call about what should be done. I try to involve staff in discussions of why we do things the way we do so they will see the big picture and also see themselves as part of the team that can have a positive impact on the success of the company."

Training Goes with the Territory

Training employees is part and parcel of just about any CAD manager gig. Our panel is split among those who love this part of the job and those who would love nothing more than to delegate this part of the job.

Every member of our panel is involved in training to some extent. Some conduct formal monthly meetings, of which train-



ing is a part; others host weekly training sessions; all perform on-the-job training on an as-needed basis. Several of these managers take advantage of outside resources to help with training-related tasks. For these managers, outsourcing training means turning to Autodesk-sponsored training available from an Authorized Training Center (ATC), sending employees to the annual Autodesk University event, or using other forms of training available from third parties. The majority of the panel, however, prepares company-specific training material themselves and takes on the responsibility of presenting it to their employees.

One of the CAD managers who responded to our survey spends a good deal of time training employees, particularly new hires, which take up some 50 percent of his time. When it comes to training existing team members, this manager works smart, giving them tips & tricks weekly, hosting roundtable meetings with employees, and even forming a local user group (LUG) to introduce them to the CAD community. By contrast, another manager unabashedly declares that he "hates training" and outsources most of it.

One of the respondents has found a way

Where to Go for Help

There are a number of resources that can help fledgling CAD managers (and even polished pros) do their work more effectively. Here are a handful of them.

Amazon.com

<http://www.amazon.com>

Search for CAD management books, including Ralph Grabowski's *CAD Manager's Guidebook*.

The Forums at AUGI.com

<http://www.AUGI.com>

AUGI members can discuss CAD management issues at the AUGI Forums discussion boards.

Autodesk, Inc.

<http://www.autodesk.com>

On the Autodesk home page, click on Support, navigate to the Training section, and you'll find a number of CAD management articles on the "How-To" pages.

CAD Digest

<http://www.caddigest.com/index.htm>

"The reading room for CAD computer aided design" features a number of helpful articles on CAD management and other relevant topics.

CADinfo.net

<http://www.cadinfo.net/editorial/>

Read archived articles on CAD management topics at this website.

CAD-manager.com

<http://www.cad-manager.com/>

Consultant, author, and instructor Robert Green has created this web-based resource for CAD managers, which includes a collection of his articles.

TenLinks

<http://www.tenlinks.com>

The section on CAD management at this well-known web portal points visitors to other CAD management-related sites.

Your Local User Group

Some of the best support you'll receive comes from fellow CAD managers. Go to www.AUGI.com to find a Local User Group in your area.

“My favorite part of the job is the initial start of a big project—the thrill of preparing a strategy and outlining a plan of action.”

to save time spent on training. “I find that by winnowing through the new software to see how we will be using it in our office and then presenting only the features we need I can cut down significantly on the time we spend training,” she says. This manager also sees to it that all training materials are posted on her company’s intranet so that employees always have access to assistance.

While most of our panel indicated that they do not take advantage of online training, one does, explaining, “We have e-learning available company-wide for a variety of software and educational packages. This is an option for each individual to use as they need or want. Our group is allowed the freedom to do this any time they wish, as long as it does not interfere with their regular work.”

Unlike most of our panel, Indak Manufacturing’s Clauson is using e-learning more often these days. “Training is harder to manage today because of the recent economic climate, the increased complexity of software and reduced delivery times. We no longer have the luxury of sending people to off-site classes for a week at a time, so we’ve had to rely more on e-learning, which has thankfully become better. I have one-hour sessions almost every week for all of my CAD users to cover advanced topics and develop new strategies.”

What They Love About It

When asked about their favorite part of the job, most of our panel responded that their ability to help people increased their overall job satisfaction. Judging from the responses we received, CAD managers are problem-solvers at heart, and many enjoy this aspect of their jobs. Says one, “My favorite part of the job is the initial start of a big project—the thrill of preparing a

strategy and outlining a plan of action.”

Another respondent loves the challenge of investigating new technology and “experiencing the ‘aha’ moments when an idea changes the way we do business fundamentally.” And this is indeed a challenge, admits this manager. “The problems arise with people adopting technology that shifts the way work is done. They adopt technology that makes existing practices easier, but the paradigm shift is a lot harder.”

Clauson agrees. “When I started in the late 1980s, one of my prime concerns was learning how to ease people away from the drawing board and helping them to understand the digital environment. In retro-

spect, the primitive nature of software at that time made the cultural shift easier than it would be today. Imagine not knowing CAD at all and being pushed into trying to quickly become productive in the high-end solid modeling or building systems we have today.”

There are many ways to address this problem, but one manager handles it this way: “We undertake a massive effort to address the concerns and explain the benefit, but we usually focus on a few individuals who fight the hardest, get them on board, and the rest usually follow.”

Good Advice

We asked our panel to tell us the best job-related advice they ever received. Here’s what they told us.

“Work is work. When you leave work every day, leave work at the office and enjoy your time with your family. The work will still be here in the morning.

“Try not to do everything yourself. Delegate and train others.”

“Keep it simple.”

“Be sure to praise people for a job well done.”

“Don’t take it personally.”



SOMETIMES YOU HAVE TO DRAW THE LINE...

...and if you design parking facilities we’re talking about a lot of lines. From the makers of AutoTURN software comes ParkCAD – the world’s premier parking layout design software. ParkCAD makes it easy to generate conceptual layouts and test “what-if” scenarios in seconds. Say goodbye to tedious calculations, ‘guessing’ stall counts, and one-parking-stall-at-a-time drafting.

ParkCAD - IT’S ABOUT TIME YOU DREW MORE THAN JUST LINES.

Download a **FREE TRIAL** version of ParkCAD today!
www.transoftsolutions.com/DrawTheLine

autodesk
authorized developer

AutoCAD is a registered
trademark of Autodesk, Inc.

1.888.244.8387
info@transoftsolutions.com

Transoft
solutions



Sometimes, good advice comes not from word, but from deed, as this manager illustrates: “I worked for an old guy who would tell me stories and do little things like fill the copy paper and make coffee. He was a brilliant man, and even though he was a senior director in the company, no job was beneath him.”

Sharing Their Expertise

Our panel, which has more than 40 years combined experience in CAD management and related jobs, has advice of their own. Here’s what they tell other CAD managers.

“I offer this advice. Always work knowing that over your shoulder are three very important people: 1) A judge—so keep track of what you do and be prepared for a detailed explanation or justification; 2) An English teacher—this will keep your grammar correct; there is nothing worse than poor grammar, especially on a record survey; and 3) A surveyor or engineer—depending on your skill set, be ready to answer the ‘why did you do it that way?’ questions.”

“Break everything down. If you have to develop a block catalog with 2,000+ blocks, remember that it starts with just one. If you develop CAD standards, do one page at a time.”

“Keep your documentation in order and your people informed of standards. Keep

them motivated to do well for you. Give praise often for good work.”

“Recognize that you have to start with each individual at his or her current level of competence, no matter how much you wish they were further along. You need to recognize that everyone learns differently and has different skills to offer. Appreciate the differences.”

“Don’t try to ‘own’ the company CAD standards. Always let them be fluid and open to other people’s input. Your job is to facilitate, not dictate. Listen, listen, listen.”

“Delegate and train others. It is not a requirement that you be the smartest at your job; rather, have your employees do the work as you direct them.”

“The hardest part of being a CAD manager is proving your ‘worth’ to upper management and owners. For example, I used to get a lot of problems because the company thought that my job cost them money. At first you might agree, but realize that if you spend time investigating new technology and find something that will save the company, say, 20 minutes a month per employee on CAD work. If your company has 30 employees, that’s 10 hours a month or 120 hours a year. If it takes you a week to investigate, implement, and train people on whatever it is, you have just saved your company 80 hours in that year alone.”

Check out Robert Green’s sidebar “CAD Manager Top 10 Tips and Tricks” for more good advice.

Welcome the Challenge

The ever-changing role of CAD manager can stress you out if you let it. One member of our panel chooses not to. “As a CAD manager, I think your skills/duties always change. Personally mine have changed and evolved over time. We are constantly evolving the way we do business and my role is to help drive those changes and evolutions. I started out programming routines in LISP and customizing menus, then I learned VBA so I started implementing those new routines. Then I would go to Autodesk University and learn other ways of streamlining our office. All of that created an evolution in my job skills and description.”

Clauson, too, suggests welcoming the challenge that comes from a changing position. “My job has become dramatically different over the years. It is certainly more dynamic because it changes almost daily. I think that the ability to manage change is in itself a prime skill that modern CAD managers must develop well. Also, it helps if one can have the attitude of enjoying the challenge of changing, rather than seeing it as a burden. Every day brings something new, so there’s little danger of boredom. May as well enjoy it.”

Marilyn Law is managing editor of AUGIWorld.

About the CAD Manager Panel

Our six panel members have more than 40 years combined experience in CAD management and related positions in a variety of industries including civil engineering, architecture, real estate, and retail. In addition, we welcome the expertise of both John Clauson, a CAD manager who was AUGI president in 2003, and Robert Green, CAD manager, author, and instructor. We appreciate the participation of everyone in the development of this article.

CAD Manager Top 10 Tips and Tricks

10

Standardize Your Software

Standardizing your work environment begins with standard software deployment. Locate customized routines on network drives so that all software behaves consistently across your company. Standard software means updates can be driven from a central location rather than each machine. The more consistent your software is the less time you'll spend supporting it (and your users).

9

Implement Minimum CAD Standards

Instead of trying, yet again, to achieve a grand and overarching set of CAD standards, push for acceptance of minimal standards you can build on. Take care to target your standards toward items that improve productivity and design automation to achieve real cost savings. Remember that it is better to have minimal standards that are actually followed than grandiose standards that are ignored. And once your minimal standards are in place you'll have a solid base of standards to build on over time.

8

Enforcing Your Standards

Now that you have minimal productivity based standards in place, how do you deal with renegades that won't get with the program? Assuming you've crafted your standards to address the productivity of your users, and have communicated it to upper management, enforcement now takes a surprising turn in your favor: Anyone who won't follow the standards is simply wasting money. Think about that. Renegades aren't being ornery, stubborn or defiant; they're simply wasting the company's money. Do you think upper management will support you in your desire to get standards followed now?

7

Managing for Savings

By looking for opportunities to save time, money and effort you'll gain a great management reputation. As you manage your

CAD team you can sidestep political and personnel issues by becoming the Pied Piper of efficiency. After all, if you manage your team in a way that gets work done quickly, efficiently, and with less expense, you're obviously doing something right. And in today's marketplace efficiency protects you from being outsourced as well!

6

Train the Right Way

Do fight for a training budget but target that budget towards implementing CAD standards and work practices that address productivity issues. Build a training topics list that gives the best financial payback for your company. Avoid generic upgrade training that doesn't address your company's particular needs. As you implement your training program make sure users know that they will be expected to participate in training and master the subject matter. Lastly, use your power users to assist you in developing and delivering your training program to better leverage your time.

5

Maintain a Solid Budget

Your budget says a lot about you. Upper management will draw conclusions about your thoroughness and vision based on how well your budget holds up. Remember that some upper management team members may only know you by your budget documentation. There is no way you can write something once a year and remember every great thought you ever had, so keep recording your thoughts and format your budget documents so they are concise, readable, and professional looking.

4

Evaluate Technology on Finances

The difference between evaluating technology on features and on finance is the separator between being a technologist and a manager. You may find some really neat software that just won't pay for itself and have to veto it even when users may be clamoring for it. On the other hand, you may very

well find low-priced utility software or home-developed routines that can have tremendous payback potential. The key is to put your technology budget into products that will save time and money for your company.

3

Get Purchase Requests Approved

If you follow my advice on item number 4 all you have to do is write down your justifications and submit them to management and you should find your purchase orders being approved much more easily. The fact is that you'll be submitting purchase requests that are well grounded in financial performance that will be viewed as a business decision. Bottom line: Justify what you need rather than asking for what you want.

2

Stop Fighting Fires

OK, there had to be one item in the top ten list that I don't have a clever answer for. However, if you take steps to standardize your work environment and train for productivity you should see your fire fighting reduced over time. If anyone continues to create problems by not adhering to standards, you should resolve the issue using your management clout rather than simply fighting the same fire over and over.

1

Getting Management to Back You

If you follow the suggestions I've made so far, upper management will see you as a level-headed, business-focused, results-oriented CAD manager that has the company's financial interest at heart. When your CAD department delivers more work using fewer labor dollars job after job, you won't have to worry about managerial support, you'll have to worry about being promoted instead!

Robert Green writes for Cadalyst magazine and provides consulting and public speaking services throughout the U.S. and Canada. Contact him at rgreen@greenconsulting.com.

On Formulas and Autodesk Revit Families

Autodesk® Revit® differs from CAD drafting in many ways, which can be confusing or intimidating to people who are used to creating drawings with lines, arcs, and circles. One of the most fundamental and biggest differences between a drafting application and a parametric design application lies in the manipulation of objects. CAD applications start with vectors (linework) and add intelligence to them. Using LISP, ARX, and VBA, clever programmers over the years have devised many ways to create and manipulate AutoCAD® entities. The Autodesk “Desktop” series of applications are built on leveraging information into and back out of linework to create 3D objects and spaces.

In Autodesk Revit, many of the object specifications or manipulations that formerly would have taken at least a passing knowledge of programming are available right in the program in ways that designers can grasp without turning geeky.

First, let’s review very quickly what parameters are, and why they are so useful. Nearly anything you can think of that will define a building component by name or hold a value, such as Height, Width, Length, Code, Cost, or Comments, has been built into Autodesk Revit already, and you can easily create your own at any time.

Family creation is the place where parameters and their uses are most apparent to Revit users. There is not enough space in this article to give more than an overview of parameters in families. The tutorials included with Revit are always a good place to start when you’re trying to learn more about the program. In this article, we’ll survey Revit’s tutorial exercises that deal with parameters, particularly the latest wrinkles, parameters that obey con-

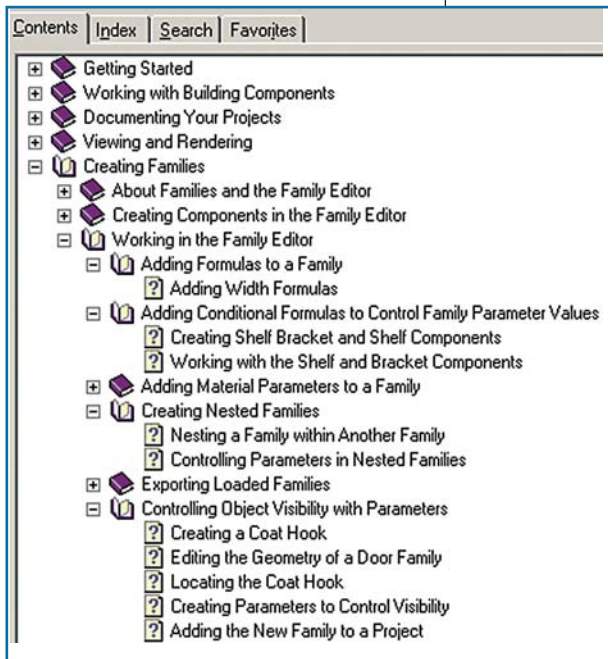


Figure 1

ditional (yes/no) and logical (if/then, greater/lesser) operations. Be warned—there is some programming-type syntax you will have to master to make the best use of logical parameters.

All the tutorials we will discuss can be found by using the menu pick Help>Tutorials. The dialog that opens has four tabs. On the Content tab expand Creating Families>Working in the Family Editor. Expand Working in the Family Editor to see individual lessons. See Figure 1.

We will look at the lessons on:

- Adding Formulas to a Family
- Adding Conditional Formulas to Control Family Parameter Values
- Controlling Parameters in Nested Families
- Controlling Object Visibility with Parameters.

Simple Formulas

Open the exercise Adding Width Formulas under the tutorial lesson Adding Formulas to a Family. This exercise shows you how to define the width of two side window units in relation to the width of the center unit in a window assembly.

This example is simple, but powerful. When you combine formulas with Family Types, you can very quickly create many different rule-based variations of a component type, each one of which allows for further controlled variations.

For instance, suppose you create two Family Types in the Palladian window family file, one in which the side windows are equal in width to the center unit and one

in which the side windows are one half the width of the center unit. Each time you load the family into a project file you have these two types to choose from. Each instance of each family could have a different height, but you can't mistakenly change the width of any window instance.

Nested Families

A step up in complexity from the Window width exercise is the exercise on Controlling Parameters in Nested Families, inside the lesson on Creating Nested Families. This exercise shows you how to load a

transom family into a door family, and how to key the Width parameter of the transom to the Width of the door.

Again, this example is deceptively simple. Associating a Family Parameter with another Parameter is a powerful tool. You can have whole families inside other families, each one keyed to something different or all keyed to different relations to the same element. Think of a door with an eyebrow transom window. You can create versions with different height eyebrows as options, and swap out all instances in your project by cycling through family types.

This is how you start to build a catalogue of products, options, or standards efficiently.

Controlling Visibility with Yes/No

Not only can you alter the relationships of the family components that stay "on" all the time, you can have components appear and disappear. The Revit tutorial lesson "Controlling Object Visibility With Parameters" steps you through exercises to create components to insert in a door—a hook and a kick plate—with "switches" to turn them on and off.

This ability deepens the usefulness of nested families—you can build complex assemblies with control over the parts, as in cabinetry with end panels that

show or not to meet wall conditions. You can build a catalogue of your most used components with sizes and features built in and selectable.

Conditional Formulas

Autodesk Revit 6.1 adds to the ability to work with parameters. With logical operations now available, components can auto-correct to a much higher degree. Revit's tutorial lesson "Adding Conditional Formulas to Control Family Parameters" shows you how to vary the thickness of a shelf based on the length, and how to have a support bracket that appears or not based on the length of the shelf.

This is where you have to learn some formula operators and their special syntax. This is not exactly programming, but something closer to algebra. Command-line type attention to detail and punctuation will save you time and frustration. By way of example, here is the formula, in the exercise "Creating Shelf Bracket and Shelf Components," that controls the thickness of the shelf:

If (Length < 4', 0' 0 1/2", If (Length > 12', 0' 1", 0' 0 3/4"))

Parameter	Value	Formula
Thickness (default)	0' 0 3/4"	= If (Length < 4', 0' 0 1/2", If (Length > 12', 0' 1", 0' 0 3/4"))
Length (default)	4' 0"	=
Height	3' 0"	=
Depth	2' 0"	=

Figure 2

Those of you with AutoCAD experience can compare this formula with the lines of LISP that would be required to create a 3D object to represent a shelf that situates itself at the desired height, with a desired length and correct thickness. Autodesk Revit does the heavy lifting—your instructions, as always, need to be precise, but are getting simpler and more powerful with each improvement in parameters.

Now your components can not only grow or shrink proportionally, but also display options or subsets based on anything from length to family type



Christopher Fox is a freelance architectural drafter, an educator, author, and frequent contributor to AUGIWorld. He can be reached at lcf@archimagcad.com.

Modifying AutoCAD's Context Menus

Context shortcut menus are a powerful tool. All those common tasks that you perform in a specific context are right there with a right-click of the mouse. And editing menus is easy to do. In fact, it is so easy to edit a menu in the normal manner, by using a text editor to edit the .mns/.mnu file directly, that you may wonder why this article introduces an alternate approach: to use Visual LISP to add or remove items to context-sensitive menus.

One reason why you may consider this approach is that most context-sensitive menus still reside in the default AutoCAD® menu in many offices. Quite often the AutoCAD menu is considered “off-limits” by the CAD manager and you may not be permitted to modify it. Granted, a context-sensitive menu may be copied to a partial menu for further editing. However, this may be overkill if all you want to do is add one item a single context-sensitive menu. And if you want to remove an item from a context-sensitive menu, without permanently changing the menu, what can you do? This article will show you how to use Visual LISP code to perform these tasks without editing the menu files.

Another reason might be that you want to provide a context-sensitive menu item only under particular conditions, such as when a user works on a particular type of drawing. The changes made to a context-sensitive menu via the code presented here are not permanent; they are only active during the current AutoCAD session. Demonstrating this alternative also gives you the opportunity to see the advantages that the ActiveX interface can give your programs.

The code itself is posted on AUGI's website (www.AUGI.com) in the *AUGIWorld* forum. You may wish to download it so that you can refer to the code while reading this article. The code on the website is fully commented.

Listing the Context-Sensitive Menus

Context-sensitive menus are defined in a .mns/.mnu file as a POP menu, starting at POP500 thru POP999. (POP0 is the Object Snap shortcut menu.) To my surprise, the number isn't so important—it is the menu's alias name that makes a POP menu context-sensitive. Details are in the Customization Guide if you wish to create context-sensitive menus in the traditional manner.

In order to add or remove an item from a context-sensitive menu located in the AutoCAD menu, you will need to know how to identify those elements. Most of the context-sensitive menus labels in the AutoCAD menu start with “Context.” The only exceptions are the Object Snap and Grips menus.

The function (I:ListMenus) will list all AutoCAD context-sensitive menus and their items to the text window. The menus are numbered, and these numbers are an index number to each menu. Because there are also menus that are not context-sensitive, these numbers will have gaps in the sequence. The list

also identifies each item in each context-sensitive menu by an index number and its label. Separators in the context-sensitive menu will list as a blank line. Please note that the index numbers begin at zero. In many programming languages indexes start at zero, not one, and the ActiveX interface, which this code uses, is zero-based.

A partial sample of the result is shown in Figure 1. The ampersand (&) in a label indicates the accelerator key used to execute the menu item. Because it is part of the label, it needs to be included when identifying the menu item by its label.

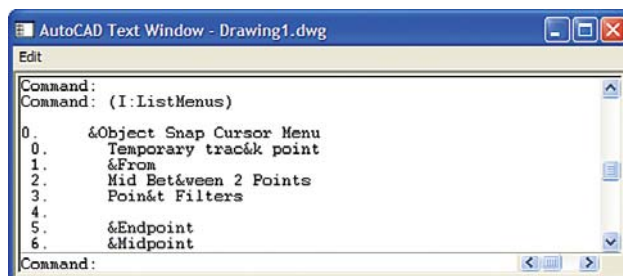


Figure 1

The code used later in the article will work with either the labels or the index numbers. However, some context-sensitive menus have the same label, such as the menu that displays for a single selected XRef, and the menu that displays when multiple XRefs are selected. Therefore, at times you may need to use the index numbers. However, index numbers will change as you add and remove items, so exercise care when using index numbers.

Adding Items to a Context-Sensitive Menu

A menu item needs three things: a home, a name, and a job. The “home” for the item can be provided by using either the context-sensitive menu’s label or its index number. The item’s “name,” or label, is a string that must not already exist in the context-sensitive menu. The “job” is a normal menu macro, formatted as a string. Details on how to make a menu macro are described in the Customization Guide. However, it simply breaks down to providing the correct command, options, cancels, pauses, and <Enter>s. The function provided for this article supports all those items mentioned. Support for other special control characters normally available to menu macros has not been provided in the functions for this article.

(I:AddToContextMenu) is the primary function for adding an item to a context-sensitive menu. It requires the three arguments described earlier (I:AddToContextMenu menu name macro): the menu, as a string or index number; the name, as a string; and the menu macro as a string. If successful, the command line will display the result.

The function depends on some subroutines included in the .lsp file, which is available on the AUGI website. The subroutine (I:IsItemThere) checks to see if the item exists, and if it does, returns that object, or nil if it does not exist. The subroutine (I:FormatMacro) makes sure that the string provided is correctly formatted so that the ActiveX interface will accept the macro. That subroutine further depends on the subroutine (I:SubstAll), which replaces all occurrences of a substring in a string with a given substring. So this subroutine replaces your “^C” with the ASCII code that the ActiveX interface requires.

Removing Items from a Context-Sensitive Menu

Removing items from a context-sensitive menu is a simple process. Note that this is not a permanent deletion. The next time the menu loads, the item will be back. This gives you the power to temporarily affect what a context-sensitive menu displays. If you want the change to be made every time AutoCAD starts, you could add this code to your Acad.lsp file.

The function (I:RemoveFromContextMenu) does the work of removing an item from the specified context-sensitive menu. It requires two arguments (I:RemoveFromContextMenu

menu name): the menu, as a string or index number; and the name, as a string or index number. Remember, you can get the index numbers from the listing provided by (I:ListMenus). If successful, it will display the result.

The function depends on one of the same subroutines that (I:AddToContextMenu) uses; (I:IsItemThere). If the menu item is found, it is returned as an object, and that object is deleted (remember, not permanently!).

Samples

Would you like to see some samples now? Two of the following examples will add items to the Edit mode context-sensitive menu. The final sample will remove the Options item from the Default mode context-sensitive menu.

Zoom object

AutoCAD 2005 added a useful Zoom option: the ability to zoom to selected objects. However, it is sad that this zoom option is not available from the context-sensitive menu when objects are selected. Look at the list of menus returned by (I:ListMenus). It is obvious that the

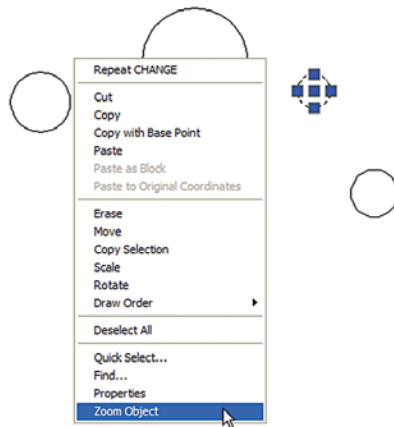


Figure 2

“Context menu for edit mode” is the menu we are interested in. (If it is not obvious, you can compare the list to the menu itself when an object is gripped.) Here is the sample (and you can see the result in Figure 2):

```
(I:AddToContextMenu  
  “Context menu for edit mode”  
  “Zoom Object”  
  “_.Zoom _Object”)
```

Modify Linetype

Perhaps you often modify an object’s linetype to one or two specific linetypes. Why not add this to the context-sensitive menu for editing objects? In this case, the

menu macro will require two explicit <Enter>s at the end of the macro, one to exit the Linetype option, and one to exit the ChProp command. In this sample, all <Enter>s are represented by semi-colons (;), per the guidelines for menu macros given in the Customization Guide. Semi-colons are preferable to spaces as it is easier to count semi-colons than to count spaces.

```
(I:AddToContextMenu  
  “Context menu for edit mode”  
  “Hidden Object”  
  “_.ChProp;_LT;Hidden;;”)
```

Remove Options

Perhaps you don’t want the Options item to be displayed on the default menu. If your company is like many others, after the options have been set to your satisfaction, there is not much need to run that command often. So simply remove the item from the context-sensitive menu. Note that the item to be removed needs to be provided exactly as shown in the listing of context-sensitive menu items returned by the (I:ListMenus) function.

```
(I:RemoveFromContextMenu  
  “Context menu for default mode”  
  “&Options...”)  
Conclusion
```

Wrap-up

The alternative described in this article gives you a way to modify AutoCAD’s standard context-sensitive menus in a manner that does not require you to manually edit the Acad.mns/.mnu files. This approach gives you the power to modify the base setup without the need to maintain copies of the menus that change with each release of AutoCAD. It also gives you self-documenting code that describes the changes you make to the way the Acad context-sensitive menus display. This is a real plus over attempting to maintain copies of the Acad menu files and documenting the changes made within those files.

Please leave comments on the usefulness of this article in the AUGIWorld forum at www.AUGI.com.



R. Robert Bell is network administrator/programmer for MW Consulting Engineers in Spokane, Washington. He has used AutoCAD since v2.18 (AutoLISP!). He is on AUGI’s Board of Directors and can be reached at robert.bell@augi.com



Using AutoCAD and Autodesk Architectural Desktop Together

» *How Can Those with AutoCAD-based Products Work Together In Building Design?*

The Autodesk® 2005 family has added productivity features and improved construction document (CD) organization without changing the drawing database. Therefore, the 2004 family of products can be used without special adaptation for most work. Autodesk® Architectural Desktop 2005 improved detailing, sheet set management, and publishing. With object enablers, AutoCAD® 2004/2005 can open, modify, review, plot, and make minor changes to Autodesk Architectural Desktop and Autodesk® Building Systems drawings (Architectural Desktop objects can be edited; i.e., move, copy, rotate, trim, etc.)

AutoCAD 2005's sheet set manager (SSM) and Architectural Desktop/Architectural Desktop 2005's project navigators (PN) have a similar functionality and can both be used to access project files. The 2005 publishing features allow for frequent virtual plotting to a project website where all involved can keep up to date with the project changes. To round it out, the free DWF viewer and

low-cost DWF markup products allow project managers and clients to guide the process without a large investment and without requiring the plotting and mailing delays and expenses.

Architectural Desktop is best for building modeling. Autodesk Building Systems, which contains all Architectural Desktop features, is best for PME design, but AutoCAD may seem more familiar. AutoCAD is sufficient for general drafting. The 2004 products can be used with 2005 products, but should not be used to change fields or tables if their automated capabilities are to be maintained.

Architectural Desktop 2004/2005 drawings must be exported to those using AutoCAD 2002/Architectural Desktop 3.3 and earlier. Such drawings are "dumbed down," losing their Architectural Desktop intelligence permanently. Those sending plans to engineers should use the screened display configuration. Exported drawings are primarily useful for background templates. Those using exported drawings should xattach them rather than working directly in the exported drawings to facili-

tate future change.

Some consultants using 2004/2005 products may prefer to work on exported plans due to their 2D nature.

What Is a Sheet Set?

Sheet sets, available to users of the 2005 family of products and Architectural Desktop/Autodesk Building Systems 2004 users, are used to organize, cross-reference, and publish the set of drawings. This feature helps everyone see where and how individual views fit into the Construction Document Set. Sheet sets can be created from the sample templates, from existing drawing sets, or from scratch. With Architectural Desktop 2005, according to Chris Yanchar of Autodesk, "you can [also] assign a Sheet Set Template File under Options > AEC Project Defaults. This is how out-of-the-box and new Architectural Desktop Project has certain subsets defined on the Sheet Sets tab. When you open an Architectural Desktop 2004 project in 2005 for the first time, a sheet set is created for you by importing the first layout of each sheet drawing." Architectural

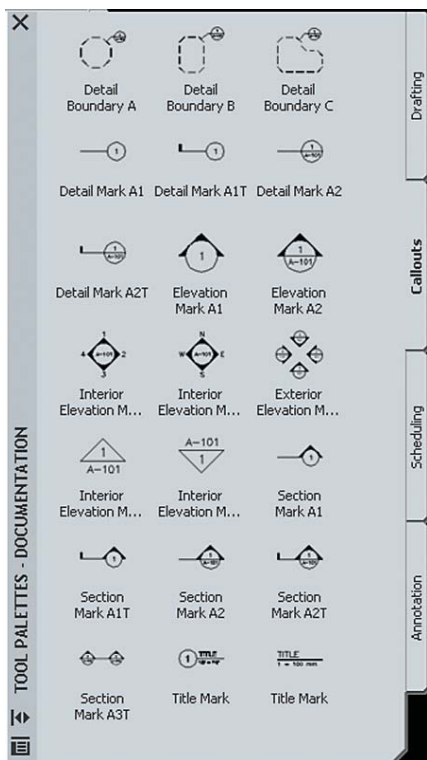


Figure 1

Desktop 2005 can automate the cross-referencing of drawings with special callouts and label blocks using a new feature called fields and tables. A "sheet list" table object, which contains the drawing index, can be placed on any selected sheet. If sheets are added, right-clicking it and choosing "Update" updates the sheet list.

Sheets are paperspace layouts registered with the sheet set (.dst) that usually contain a titleblock and one or more sheet views. Sheet Views are paperspace viewports on sheets. Sheet drawings are files that contain preferably one sheet but possibly more sheets. Each layout may be registered as a separate sheet. Sheet sets are collections of those sheets that can be organized into subsets.

Both the SSM and the PN automate the process of attaching or overlaying xrefs and creating the paperspace viewports. They both use template files to create sheets.

How Is SSM Different from PN?

Both AutoCAD software's Sheet Set Manager (SSM) and Architectural Desktop software's Project Navigator (PN) can manage sheet sets. While the SSM and PN both produce sheet set files (.dst), the Project Navigator has advantages with respect to project awareness and callout management. According to Yanchar, "The biggest difference, and area of potential

confusion for mixed offices, is in views, callouts, title marks, and how we handle field codes within them." In Architectural Desktop, he further notes:

1. There are dedicated tools for callouts and title marks of all flavors [and] specific routines for placing a section callout with line, tag, and tail...or a detail boundary for circle, rectangular, or freeform.
2. You are allowed to place callouts and title marks in model space.
3. ADT creates its own temporary field code for callouts and title marks which supports the process of having a link to a model space view and then transfer that to the regular AutoCAD Sheet Set field codes once placed on a sheet.
4. 'Named model space views' each can have their own scale for scaling annotation/callouts.
5. You can post-link a callout or title mark to a model space view or sheet view by drag/dropping the symbol to the navigator.
6. ADT has specific tools for placing Callouts and Title Marks with various creation parameters, including the ability to generate sections, elevations, and

details from the building model. In the AutoCAD Sheet Set Manager there is the concept of Label Blocks that allow the user to associate individual blocks that can represent the callout symbol.

ADT callouts and title marks (available from the callout tool palette) can generate views. (See Figure 1 for PN Callouts). AutoCAD callouts and label blocks (available from SSM) are label, cross-reference and document the sheet view scale (Callouts and Label Blocks must be selected manually if the sheet set was created using the PN. See Figure 2, next page.). In ADT 2005, model space views in "view" drawings can be labeled with title marks and associated with "sheet views." AutoCAD/Architectural Desktop 2004 has no fields or tables and should NOT be used to key the set.

How Are Views Generated and Updated?

With AutoCAD, multiview and pictorial drawings are created by adherence to drafting practices. The view command is used to specify the boundaries and layer settings of those drawings. Coordination between views is solely the drafter's responsibility.

MANAGE YOUR FILES WITH AUTOEDMS™

Simplify Security and Control

- Security and control of your
- AutoCAD® drawings save time and
- money, and can also help prevent the
- headaches of errors. AutoEDMS
- tracks, profiles and audits complete
- document activity with automatic
- filenames, while integrating
- seamlessly with all popular Windows
- programs. As files are revised and
- approved, they become "master" files
- and all previous versions are rolled-
- back to guarantee current version use.
- Take control and secure your files
- today with AutoEDMS.



Call us at 800-325-0425 Fax: (310) 755-6050
2463 208th Street, Torrance, Ca 90501-6238 USA
Email: info@acssoftware.com Web: acssoftware.com

© 2004 ACS Software, Inc. All rights reserved. AutoEDMS is a trademark of ACS Software, Inc.
AutoCAD® is a registered trademark of Autodesk, Inc.

ADT users draw plans (inherently 3D) as “construct” drawings. Each plan is associated with a level and a division. Together, all the levels and divisions make the “project.” Drawing elements that are not associated with a particular location are called elements. Special spanning constructs include more than one level (stairs and

facades for instance). Views are separate reports of the 3D building model.

In Architectural Desktop, view drawings are often created by right-clicking on a folder in the PN Views tab and by choosing New View. The Architectural Desktop user typically creates a top view view drawing for each level construct by choosing

the context and view direction for that view. An additional view drawing, often named model, should be generated that contains all the levels and divisions. This view drawing is typically used for generated building elevations and sections. Architectural Desktop callout tools are used to generate these report views.

Architectural Desktop callout tools give the option of generating elevations or sections in new view drawings, existing view drawings, or the current drawing. Architectural Desktop callout tools also allow a choice of scale for that view and an option to include a title mark. I have found using the current drawing option to be very satisfactory in Architectural Desktop 2005. It’s nice to see the elevations side-by-side.

Architectural Desktop view drawings are the recommended place for annotation. With Architectural Desktop, generated views can be updated as the model changes but depend on the drafter to choose Refresh or Regenerate.” Special “live sections” stay up-to-date.

How Are Views Placed on Sheets?

View drawings or model space views can be dragged-in-dropped from the PN Views tab

onto a sheet to become sheet views. Using the SSM, views are dragged from the Resource Drawings Tab. (See Figure 3.) The SSM adds label blocks automatically to each sheet view. In Architectural Desktop, the fields in pre-existing title marks and callout fields are resolved and hyperlinked when the sheet view is created. Drawings too large for a single sheet view should be divided into separate model space views and matchlined. Architectural Desktop layout grids are helpful for subdividing the views. PN callouts are linked to the first associated sheet view

Changing the Scale and Layer Settings of Views

The PN scale settings for view drawings and model space views become the default scale for new sheet views. To change the scale before placing the sheet view, right-click on the view name in the PN Views tab, choose properties, and then the new scale. Sheet views can be changed by display unlocking the viewport and choosing the new scale. The scale and title fields in title marks and label blocks are refreshed at regenall. Modelspace annotations, however, must be re-scaled manually.

Layer snapshots between model space and sheet views are not updated. To change the layers of a sheet view, delete the sheet view (the viewport). Deleting a sheet view breaks the links with its associated Architectural Desktop title mark and callout, causing the associated fields to display ### symbols. To relink them, drag the title marks and callouts to the associated sheet view name. With the SSM, delete the label block associated with the sheet view and create a new sheet view. A new label block is placed automatically. Adjust the VP scale and rename and renumber the sheet view name as required. Then place new callouts or edit the fields in the existing ones (that’s a bit complicated).

What Can the PN Do That SSM Can’t?

Unlike SSM, the PN can be used to delete drawings from the project directory. SSM can be used to remove sheets but the associated drawing files are not deleted. This means that the drafter using AutoCAD would still need to use Windows Explorer to do such tasks. The PN Sheets tab has two viewing modes (Sheet View and Explorer View). (See Figure 4.) When in the Sheet Explorer mode, legacy AutoCAD drawings can be dragged into

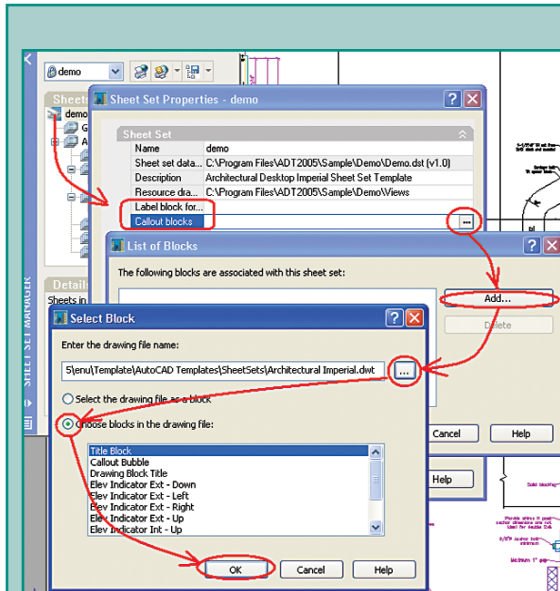


Figure 2

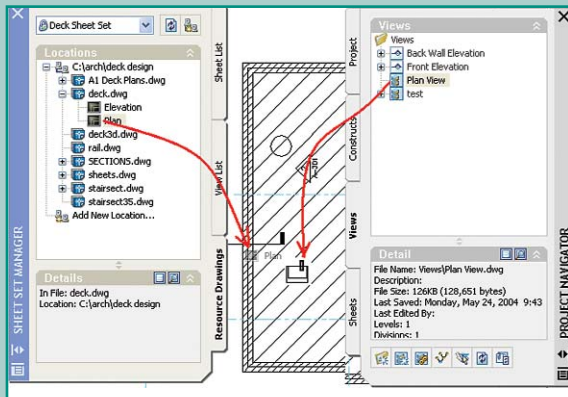


Figure 3

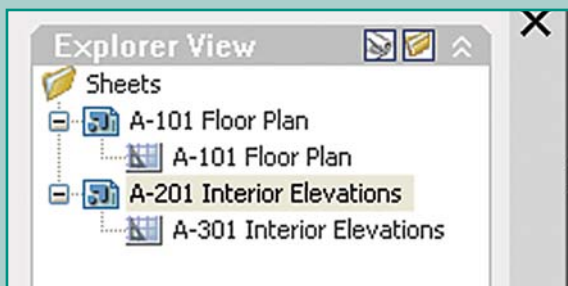


Figure 4

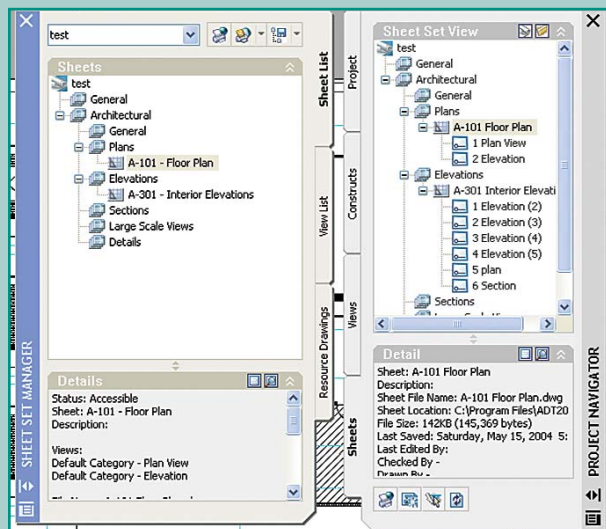


Figure 5

the sheet set and sheet files can be deleted without using the Windows Explorer. Take caution when deleting files using the PN. The files are not sent to the recycle bin and therefore are unrecoverable.

The PN automates the xref attachment and overlay process so that constructs are associated with levels (stories) and divisions. It uses subfolders: elements, constructs, views, and sheets to categorize drawing files. Views are reports of the 3D building model. An XML file is generated for each construct, view, and sheet file.

The PN can e-transmit project (APJ) files and can be used to send object-exploded versions of the project.

The PN automatically create a sheet set when an Architectural Desktop 2004 project is first loaded.

How Are the SSM and PN Tabs Different?

The SSM Views List tab shows only sheet views organized by view type as specified with the view command. The PN Views Tab shows all view drawings in the project view folder and their nested model space views (See Figure 5). The SSM Sheets List tab is similar to PN Sheets Tab but shows only sheets, without associated sheet views (See Figure 6). Views can be dragged to sheets from the PN Views tab or from the SSM Resource Drawings Tab (See Figure 3).

Can a Sheet Be Used on More Than One Project?

Each sheet should only be used on a single project. Standard information that is

used on multiple projects should be copied to individual projects or should be xref'd into a sheet for each project. This procedure is explained in the User's Guides.

How Can I Set Up My Title Block on a Sheet?

Sheet templates that come with AutoCAD/Architectural Desktop 2005 include multiple sheet size layouts in the same file. To customize, start a new drawing with the sheet template using the same plot units and replace the paperspace graphics and page setups with yours. Use fields for such items as Sheet Number Sheet Name and Date. Save the drawing as a new template.

If you have different titleblocks for each project, then open your newly created sheet template and save it as a drawing to the current project folder. Use that copy as your subset sheet template. (Sheet templates may be DWG files.) The sheet files that come with ADT include fields that access many project parameters (name, address, number...) managed by ADT. If you use the same field codes in your titleblocks, then they will behave just as the OOTB ones.

Since each sheet subset can have its own sheet template, right-click on the each subset node and add your custom sheet template. This facilitates division of responsibility.

Where to Learn More

Read AutoCAD 2005 User's Guide Part 4 and Architectural Desktop 2005 User's Guide, Chapters 6 and 40.

Read Chris Yanchar's Blog

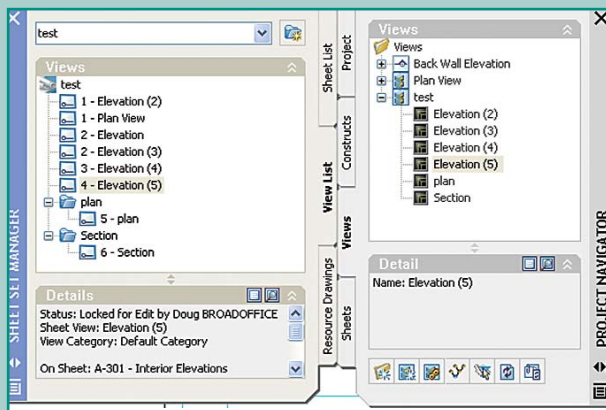


Figure 6

http://autodesk.blogs.com/between_the_walls/2004/03/tutorial_intero.html

Learn the PN features quickly by opening the sample project under C:\Program Files\<ADT2005>\Sample\Autodesk Architectural Desktop Sample Project.

Create a small dummy project with a rectangular plan construct and a single view drawing with a few elevation callouts. Manually create some views by within the view drawing. Drag title marks from the palette onto the new views. Create a sheet. Drag the views from the view tab of PN on to the open sheet.

In Conclusion

Upgrading to a 2005 product does not mean incompatibility with 2004. It is possible to use both product families together without dumbing down Architectural Desktop drawings. The SSM and PN are fully compatible, but it is advantageous to use Architectural Desktop 2005 to key the set. The 2004 family of products can be used, but avoid changing annotation that includes fields. Use SSM to create a sheet set from existing projects where PN was not used. Architectural Desktop has both features. AutoCAD has only SSM.



Doug Broad is a registered architect and instructor at Nash Community College. He has used AutoCAD since v2.6 and currently uses Architectural Desktop and does customization in AutoLISP and VBA. He can be reached at dbroad@earthlink.net.

Interior Lighting With Photometric Data

The Approach

There are essentially two general approaches for a digital artist to accomplish realistic interior illumination. The first approach involves using every tool, technique, or trick to muddle your way through and arrive at something believable and “sort of” realistic but not entirely accurate from a physics standpoint. The second approach uses reality or physics-based tools that are simply digital translations of what happens in the real world. Most often in the computer graphics world the adage “If it looks right, it is right” prevails, though this isn’t always the case when it comes to lighting. At times you need to legitimize your rendering using scientifically accurate techniques.

The Tools

There are a multitude of tools available now to realistically simulate how light behaves. The two main tools required for realistic lighting in Autodesk VIZ® are Global Illumination/Radiosity rendering and Photometric data also known as Luminaires. Global illumination and Radiosity allow for indirect diffuse lighting...which is really just light bouncing around as well as color bleed between surfaces. The newest incarnation of VIZ now has the powerful Mental Ray renderer by Mental Images.

Photometric data comes in the form of IES (Illuminating Engineering Society) files which describe the performance and light distribution patterns of actual manufactured light fixtures. These luminaire files are helpful insofar as the artist has a tangible value, measured in lumens, candelas, colour temperature) attached to a lights intensity instead of some arbitrary value.

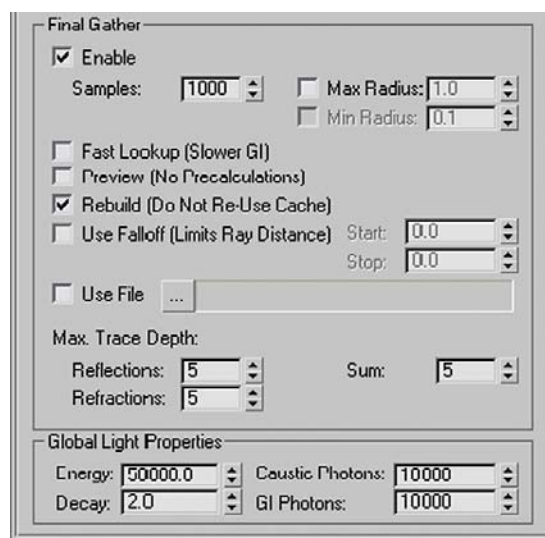
Photometric Data

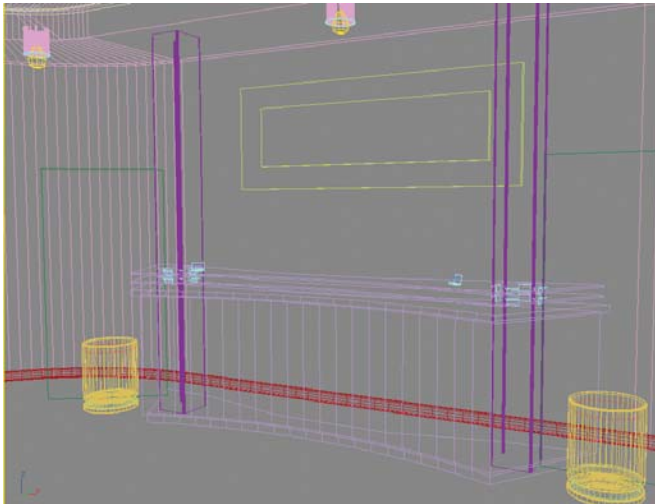
Photometric lights come in three different forms—point, linear, and area—with

all three types containing a free and targeted version for easy manipulation during setup.

The use of each light type is pretty straightforward. Point lights represent your basic light bulb or single point light sources. Linear lights correspond to tube type lights like fluorescent bulbs. Area lights typify lights that are plate or panel like in form.

Photometric lights make light placement and positioning during setup a lot easier. All you have to do is place them





where you would place a real-world light and you're done...well almost. To really take advantage of this light type we need to acquire some IES files that I mentioned earlier. There are literally hundreds of them available from all the light/light fixture manufacturers' websites free of charge.

Once you have found the IES files that are appropriate for your light setup you need to select the light and change the distribution method to "web." Once you have done this you can then load the file by going to the Web Parameters rollout, clicking the button and locating your IES file. The contents of these files describe physical attributes of the light as well as the intensity of emission in the volume around the light, so you get all the unique patterns that are characteristic of that particular light. These files by themselves will make your lighting setup more realistic...but we aren't going to stop there.

Global Illumination...

Now that we have our lights illuminat-

ing the way that we want and need them to, we can use Mental Ray to bounce the light around a bit to create some nice soft shadows while maintaining the light distribution from the IES file. Like any other global illumination renderer there is almost an infinite amount of tweaking and trade offs that can be done for speed and quality. This takes practice as well as patience but the default render values will get you going in the right direction. In the General Parameters rollout, be sure to select Mental Ray Shadow maps for our shadow type. Indirect illumination must be turned on as well as final gather. Now you can render away!

The End...

This method of creating interior lighting is a must have in the arsenal of every 3d artist regardless of whether your lighting simulations need to be physically accurate or not. The really great thing about Autodesk VIZ 2005 is that it provides the artist a myriad of tools to get the job done. In a production environment it is essential to have viable options.

Quite simply, using photometric data and Mental Ray together generates great overall illumination allowing you to take your interior renderings to the next level in realism.



Dwayne D. Ellis is the lead 3d computer animator at Hrycay Consulting Engineers, a firm specializing in Motor Vehicle Accident Reconstruction. He is also the founder of Lifeseyes Studios and can be reached at dwayne_ellis@yahoo.com.



The Logical Choice for CAD Managers.

Why fight over using PDF or DWF when AcroPlot Pro allows you to do both from:

**DWG : DXF : PLT : Word : Excel : TIFF :
CALC : GP4 : JPG : BMP : and more...**



**Lines Merge
Support for
PDF. Need
we say more?**

- **Batch converts with or without AutoCAD.**
- **Creates intelligent bookmarks from your titleblock attributes.**
- **Includes a full PDF Driver for use from any Windows application.**

At CADzation it's always been about more than just creating a PDF or DWF. It's about saving our customer's time, money, and making their jobs easier. And if we have our way you'll forget all about things like plt files and plotter configurations. We'll create the PDF or DWF and you can use it for high resolution printing just as easy as you can send it to your clients.

Download a full working trial version today



www.CADzation.com
© 2004 CADzation All rights reserved. All brand names, product names & trademarks belong to their respective holders.

Getting at Attributes Using Visual LISP and VBA

» The options available for customizing AutoCAD are varied. A common question is: "What is the best programming language to use for my customization?" However, that question cannot be answered simply. It is far better to think of the various options as tools in a toolbox. You could not convince a carpenter that the only tool he needs is a hammer, even if it is the tool he uses most often. Similarly, you should not limit yourself to only one tool for customization.

Let us take the common task of modifying a specific attribute to demonstrate how you could use the two languages, Visual LISP and VBA, and the ActiveX interface (a mechanism to use the VBA object model from Visual LISP) to perform the task. (Note: all the code in this article, and the sample A-Rm block, are available for download.) I have provided a simple menu macro below that permits users to modify an attribute. The limitation of the menu macro is that users must pick the specific attribute that they wish to modify. Therefore, if the attribute has no value, and so is not visible, the user cannot use the menu macro to modify the attribute. It also makes no provision for a bad pick, such as selecting a portion of the inserted block. Also, what if users want to use this macro from the command prompt?

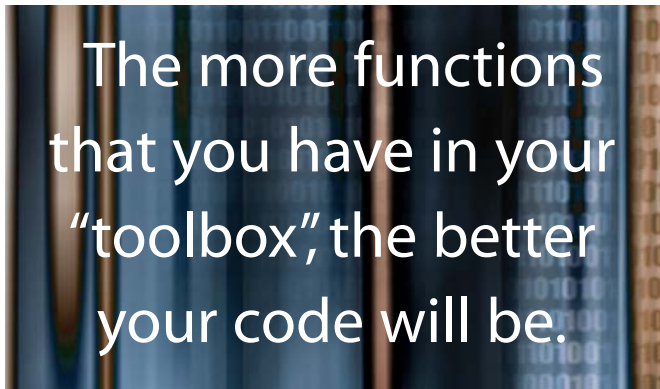
```
[Room #]*^C^C._-AttEdit;_y;A-Rm;Room;;\;_v;_r;\;
```

So a natural extension of the menu macro would be to use one of the programming languages to permit users to select any part of the insert(ed block), and then modify the attribute regardless of what part of the insert the user selected. In this article, I will cover how to do this with Visual LISP and VBA. In the next issue, I will demonstrate using the ActiveX interface to Visual LISP to do the same thing as the VBA macro. I will present that code last because the help files for the ActiveX interface are written for VBA, and it takes a mental translation to convert the VBA help files to the Visual LISP functions used by the ActiveX interface. However, the mental translation is easy after doing it a few times. The advantage to using the ActiveX interface is that you may write code in Visual LISP, if you need some of the fea-

tures from that language, yet still use the VBA object model, which has some benefits.

The simplest form that a Visual LISP program can take is to duplicate the menu macro using the (*vl-CmdF*) function. You will need to define the program's command name, which is done using the (*defun*) function. The "C:" tells the (*defun*) function that you are creating a function that should be treated like a command, and the "RM" is the command name.

```
;;; command name is "RM"
(defun C:RM ()
  (vl-CmdF ". _AttEdit"
    "_Y"
    "A-Rm"
    "Room"
    "")
  (nentsel) ; get sub-entity, e.g. attribute
  "_V"
  "_R"
  (getstring "\nSpecify new room #: ") ; get new #
  "")
  (princ)) ; clean exit
```



The more functions
that you have in your
"toolbox", the better
your code will be.

I also used a few other functions, for which I provided comments. The advantage to using Visual LISP instead of a simple menu macro is that now the command is accessible from the keyboard, and provides a friendly prompt to tell the user what to do next. You can place this code in your customized menu's .mnl file, so that it is loaded with your menu. You may then edit the menu macro to use the Visual LISP program instead.

```
[Room #]*^C^CRM
```

But the same limitations exist with this program — there might be no visible attribute to pick, and the user can still make a bad pick. So we need to make a function that will let the user pick only the insert (really, the block name) we want, and once we have that insert, find the attribute regardless of visibility. We will tackle the issue of the pick first.

I advocate making a separate user-function for this, because this function will be useful for many other programs too. The more functions that you have in your "toolbox", the better your code will be. You will use the (*defun*) function to define the user-function, but this time you will not use the "C:" prefix. This is because this function will not be a command to be typed at the command prompt by the user. Rather, it will be a function that will be used in the Visual LISP code itself. We will want to be able to tell this function what block name we are looking for, so we need to make provision for an argument (which is like a variable in Visual LISP).

```
;;; return only matching inserted block, or nil if <Esc>
(defun GetInsert (BlkN / ; provide string argument
                  ssFound) ; selection set
  (vl-Catch-All-Apply ; trap any error
    (function
      (lambda ()
        (while ; loop until good pick
          (not
            (setq ssFound (ssget "+.E:S" ; point mode, in pick-
                                box, single mode
                                (list ; filter on...
                                  (cons 0 "INSERT") ; inserts
                                  (cons 2 BlkN)))))) ; and
          provided block name
            (if ssFound (ssname ssFound 0))) ; return 1st item in
          selection set
```

I used a Visual LISP function, (*vl-Catch-All-Apply*), to trap any error, to handle the case of the user hitting the <Esc> key. Therefore, the function will only return one of two values, either *nil* for no pick, or the *ename* (entity name) of the selected insert. The work is done inside the while loop. I am using some of the infrequently used options for (*ssget*), and one undocumented option "+.", which forces (*ssget*) into point selection mode. These three options make the (*ssget*) function act just like a single entity selection with the benefit of filtering the selection.

We are now going to rewrite the code for C:RM to use this new function. At this stage, I am only going to make the program return the value from the sub-routine, because the "attribute finding" sub-routine is not written yet. You will be able to test the effectiveness of the GetInsert sub-routine by using the C:RM program.

```
;;; command name is "RM"
(defun C:RM (/ objInsert) ; entity name of insert
  (setq objInsert (GetInsert "A-Rm")))
```

Give the new command a try. You should see either an entity name or *nil* at the command line after running the C:RM program. Note that if you pick the wrong type of object, or even any inserted block other than one named "A-Rm", the selection is filtered out, and you still have to opportunity to select the correct inserted block.

```
Command: RM
Select objects:
<Entity name: 4007cea0>
Command:
Command: RM
Select objects:
1 was filtered out.
Select objects:
1 was filtered out.
Select objects: *Cancel*
nil
Command:
```

The next function will take the entity name provided by the GetInsert sub-routine and the tag name of the desired attribute, find the attribute, and return the entity data (and not just the entity name) of that attribute.

```
;;; Return only matching attribute entity data, or nil
(defun GetAttrib (eName ; provide EName argument
                  AttTag / ; provide string argument
                  eData ; entity data list
                  DxfVal ; entity type
```

```

        Done) ; flag to exit loop
(while (not Done) ; loop until done
  (setq eData (entget eName))
  (cond ((and (= "ATTRIB" (setq DxfVal (cdr (assoc 0
eData)))) ; if attribute
    (= (cdr (assoc 2 eData)) AttTag)) ; and
matches tag
  (setq Done T)) ; then exit loop
  ((= "SEQEND" DxfVal) ; if at end of insert's
entities
  (setq Done T ; then exit loop
    eData nil))) ; clear entity data
  (if (not Done) ; if not done
    (setq eName (entnext eName)))) ; then get next entity
eData) ; return entity data

```

The reason that we need to use the while loop is that attributes are actually separate entities that reside “below” the insert entity. So we loop thru the insert’s sub-entities, using the (*entnext*) function, looking for the correct attribute. If the loop doesn’t find the attribute, and reaches the SeqEnd entity that marks the end of the insert’s sub-entities, the sub-routine will return *nil*.

We will now add this sub-routine to our program. The program, at this stage, will return the entity data of the correct attribute, or *nil* if that attribute is not found.

```

;;; command name is "RM"
(defun C:RM (/ objInsert ; entity name of insert
  eDataAttrib) ; entity data of attribute
  (if (setq objInsert (GetInsert "A-Rm"))
    (setq eDataAttrib (GetAttrib objInsert "ROOM"))))

```

Remember that an attribute tag is in all upper-case letters, therefore the above code is using “ROOM” as the AttTag argument. When you run the revised program, you should get something similar to what is shown below.

```

Command: RM
Select objects:
((-1 . <Entity name: 4007cea8>) (0 . "ATTRIB") (330 .
<Entity name: 4007cea0>) (5 . "85") (100 . "AcDbEntity")
(67 . 0) (410 . "Model") (8 . "A-RM") (100 . "AcDbText")
(10 1.0 1.0 0.0) (40 . 0.1) (1 . "") (50 . 0.0) (41 .
0.75) (51 . 0.0) (7 . "S") (71 . 0) (72 . 4) (11 1.0 1.0
0.0) (210 0.0 0.0 1.0) (100 . "AcDbAttribute") (2 .
"ROOM") (70 . 0) (73 . 0) (74 . 0))

```

Now that we have the correct attribute data, all that is left to program is the code to get the user’s input for the new room number, and apply that to the attribute itself.

```

;;; command name is "RM"
(defun C:RM (/ objInsert ; entity name of insert
  eDataAttrib ; entity data of attribute
  Inp) ; user's input
  (and
    (setq objInsert (GetInsert "A-Rm"))
    (setq eDataAttrib (GetAttrib objInsert "ROOM"))
    (setq Inp (getstring "\nSpecify new room #: "))
    (entmod (subst (cons 1 Inp) (assoc 1 eDataAttrib)
eDataAttrib))
    (entupd objInsert))
  (princ))

```

Notice that I dropped the use of the *if* statement, and instead placed all the sequentially dependent code within one *and* statement. This makes it easy to quietly end a program when one of the necessary steps returns *nil* because of an error or, as in this case, a cancelled pick or missing attribute.

With VBA

Now we will perform the same task in Visual Basic for Applications (VBA). This will give you the opportunity to examine the similarities, and the differences, between both languages. For information on VBA’s object model, see the Object model in the ActiveX and VBA Reference available under AutoCAD’s Developer Help menu.

This time around we have the benefit of having programmed the application once already. Therefore, we know that we want the program to have functions that:

1. Permit the user to select an inserted block,
2. Retrieve the desired attribute,
3. And modify the attribute.

Start the VBA Manager and select the New button. Select the Visual Basic Editor button. This will bring up the VBA Integrated Development Environment (IDE). (A shortcut would be to use the command VBAIDE.) We need to make sure a couple of the editor’s options are set correctly. Go to the Tools menu, and select Options. In the Editor tab, clear the option “Auto Syntax Check,” and select the option “Require Variable Declaration.” The first option, although it may seem to be a good idea, just interferes with the process of writing code. Incorrect syntax is still highlighted in the editor regardless of the state of this option. This option, when active, makes it difficult to copy and paste code. The second option, when active, forces you to declare, or Dim, your variables before you use them. This helps you easily find spelling mistakes in your code, because the editor will complain about an undefined variable when you run the code.



If you do not see the Project Explorer, hit <Ctrl>+<R> to display it. Double-click on the ThisDrawing object in the explorer (you may need to expand the project to find it under the folder “AutoCAD Objects”). This action will bring up a code window. If the code window is completely blank, type in the line:

Option Explicit

This line is automatically added to all modules by the “Require Variable Declaration” option we selected above for all new modules. Since this module existed before you (possibly) changed the option, it might not exist in this current module.

Let us begin creating the macro that the user will use. We will be adding to this macro as the program develops.

```
Public Sub ModifyRoomNumber()
```

```
End Sub
```


Note that if you type “public sub ModifyRoomNumber” and hit <Enter>, that the parentheses and the End Sub are added for you automatically. Also note that the statements “Public” and “Sub” capitalized themselves. This automatic capitalization is a great feature of the IDE. You will quickly get used to typing mostly in lower-case, and letting the editor capitalize for you. The only capitalization you need to do manually is when specifying procedure/function names and declaring your variables. We used the expression “Public” to make sure that this procedure will be visible to the user when they look at the macro list. We will add more to this procedure later.

Let us tackle the first function, GetInsert. We know that we want this function to permit the user to select only the block that we desire. Therefore, we need to pass the block name to the function. This function needs to return the BlockReference (or inserted block) object. I am going to arrow down past our public macro to a blank line. Now type in exactly what I have below, but using the keys only where I indicate with angled brackets, e.g. <Ctrl>+<Space> means to hold the Control key down and press the Space bar.

```
private function GetInsert(BlockName as str<Ctrl>+<Space>)
as acadbl<Down><Enter>
```

This should give you the following code:

```
Private Function GetInsert(BlockName As String) As
AcadBlockReference
```

End Function

Do you see how easy it is to write code with this auto-complete feature, called IntelliSense? Once you get used to writing code this way, it is hard to go back to primitive IDEs like the Visual LISP IDE.

You may wonder what is the difference between a Sub and a Function. A function will return a value, whereas a sub cannot. Since we want this function to return a BlockReference, we use the Function statement. We also do not want the user to be able to see this function, so we declare it to be Private to this module. Below is the complete code for the function.

```
'Return only matching BlockReference,
or Nothing
Private Function GetInsert(BlockName
As String) As AcadBlockReference
    Dim FilterType(1) As Integer
    Dim FilterData(1) As Variant
    FilterType(0) = 0: FilterData(0) =
"INSERT"
    FilterType(1) = 2: FilterData(1) =
BlockName
    With
    ThisDrawing.PickfirstSelectionSet
        .Clear
        .SelectOnScreen FilterType,
FilterData
        If .Count > 0 Then Set GetInsert
= .Item(0)
    End With
End Function
```

I want to filter out incorrect selections, just as we did in the Visual LISP code.

You need to use two arrays to do this in VBA. We will be filtering on two different values, the object type “INSERT” and the insert’s name. Therefore, we need to have both arrays contain two elements. Because arrays are zero-based in VBA 5/6, we declare the arrays as you see above, with the (1). The first array will hold only integers. The second array can hold many different data types, so we declare it to hold variants. I use the next two lines to fill the arrays with the required data. The FilterType array holds the DXF codes, and the FilterData array holds the actual data values. These data values are the ones found in the DXF Reference, and not the ones found in the object model. If you think this is more complicated than Visual LISP, it isn’t, it is just “wordier.” Here is a side-by-side comparison (assuming that I’m going to use variables in the Visual LISP code):

```
(defun GetInsert (BlockName /
Filter1
Filter2)
(setq Filter1 (cons 0 "INSERT"))
(setq Filter2 (cons 2 BlockName))
Function GetInsert(BlockName As String)
Dim FilterType(1) As Integer
Dim FilterData(1) As Variant
FilterType(0) = 0: FilterData(0) = "INSERT"
FilterType(1) = 2: FilterData(1) = BlockName
```

As you can see, when you get as close as you can to comparing apples-to-oranges, the process is similar.

The rest of the function uses the Pickfirst selection set, to avoid creating an unneeded named selection set. Many VBA programmers of AutoCAD think that they *need* to use a named selection set (from the SelectionSets collection), but that is not

Do You Need To Clean, Edit, And Convert Your Legacy Paper Documents?

GTX Corporation offers the most efficient, professional, and advanced imaging technology available today.

Use:

- GTXScanClean™ for Office, designed for automatic cleanup of scanned or copied office documents.
- GTXScanClean™ for Engineering, a revolutionary software that cleans and straightens scanned or copied engineering drawings with the single click of button.
- GTXImage CAD™ PLUS 2000, a stand-alone software to expand your CAD (computer-aided design) environment. It includes raster editing with superior raster selection tools.
- GTXRaster CAD® Series, for your AutoCAD® 2000, 2000i, 2002, and 2004 environments. It includes superior raster editing and raster conversion tools.

Don't delay your projects any longer!

Visit our Web site at www.gtx.com or call

GTX
www.gtx.com



**SAVE UP
TO 50%
OF YOUR
TIME.**

Civil Engineering Software

Easy to use, time saving tools for engineers, designing, drafting and surveying – sites, roads, rivers and subdivisions using autodesk Land Desktop.

ROBERT STELTMAN INC.
autodesk®
authorized developer

**Download the
FREE 30 day
evaluation
version today.**

steltman
SOFTWARE

Toll free 1-866-260-2514
www.rsteltman.com or
email at civilinfo@rsteltman.com

true. The Pickfirst selection set will do the job just fine, thank you. I use the With statement to avoid the need to use ThisDrawing.PickfirstSelectionSet over and over. There are also performance implications that make it wise to use the With statement. I clear the selection set, in case it has anything in it. I then use the SelectOnScreen method, with the filters. Now this highlights a weakness in VBA. We are not able to accomplish *exactly* the same behavior as the Visual LISP code. You currently cannot specify the point mode (+.), single mode (:S), or the cursor box mode (:E) in VBA. Therefore, the user will be able to pick more than one object, and *must* hit <Enter> when they are done selecting objects. However, I have cheated by having the function return only the first object in the selection set. There is also the Utility object's method GetEntity method, which *will* only permit one pick, but that method doesn't permit filtering directly.

So let us add this function to our macro!

```
Public Sub ModifyRoomNumber()
    Dim Insert As AcadBlockReference
    Set Insert = GetInsert("A-Rm")
End Sub
```

We may now test the code. Make sure you have a few objects in your drawing, including an insert of the block A-Rm. Place your cursor somewhere inside the ModifyRoomNumber procedure. Now hit <F8>. We have entered "Step Mode" and the program will advance one statement at a time every time you hit <F8>. If you display the Locals windows you can watch the status of your variables.

Now we need to make a function that takes the BlockReference and the attribute Tag, and returns the AttributeReference object itself.

```
'Return only matching AttributeReference, or Nothing
Private Function GetAttrib(Insert As AcadBlockReference, _
    TagName As String) As AcadAttributeReference
    If Insert.HasAttributes Then
        Dim Attribs As Variant
        Attribs = Insert.GetAttributes
        Dim Attrib As AcadAttributeReference
        Dim Count As Integer
        For Count = 0 To UBound(Attribs) 'assume zero-based
            Set Attrib = Attribs(Count)
            If Attrib.TagString = TagName Then
                Set GetAttrib = Attrib
                Exit For
            End If
        Next Count
    End If
End Function
```

This code is pretty straightforward. A BlockReference object has the method GetAttributes. This method returns an array of attributes, which I'm saving to the variable Attribs. I then declare the variable Attrib, which will hold an AttributeReference object each time thru the loop. I also declare the counter that will be used in the loop. I set up a For loop that will go thru all the attributes of the insert. When it finds the correct attribute, it returns that attribute and exits the loop.

Now let's add this function to our macro!

```
Public Sub ModifyRoomNumber()
    Dim Insert As AcadBlockReference
    Set Insert = GetInsert("A-Rm")
    Dim Attrib As AcadAttributeReference
    If Not Insert Is Nothing Then Set Attrib =
    GetAttrib(Insert, "ROOM")
End Sub
```

Once again, I declare the new variable, and then I test to make sure we have a BlockReference from the previous function before running the new function. (And to think that your English teacher told you to never use double negatives!)

All that's left is the code to get the user's input on the new value for the attribute.

```
Public Sub ModifyRoomNumber()
    Dim Insert As AcadBlockReference
    Set Insert = GetInsert("A-Rm")
    Dim Attrib As AcadAttributeReference
    If Not Insert Is Nothing Then Set Attrib =
    GetAttrib(Insert, "ROOM")
    If Not Attrib Is Nothing Then Attrib.TextString = _
    ThisDrawing.Utility.GetString(False, vbCrLf & "Specify
new room #: ")
End Sub
```



Save the current project wherever you keep your customizations. I chose to name the file "RoomNumbers". This implies that this project may have other macros in the future that work with room numbers (such as making a room schedule).

We may run this macro from the Macros dialog box (<Alt>+<F8>) or from a menu item or the command line with the following Visual LISP code. Yes, in an ironic twist, in order to run VBA macros from a menu macro or the command line, we need to use Visual LISP! Place the code in your customized menu's .mnl file, so it is loaded with your menu.

```
;;; command name is "RM"
(defun C:RM ()
    (vl-load-com)
    (vla-load-dvb (vlax-get-acad-object)
    "C:\\Temp\\RoomNumbers.dvb")
    (vla-run-macro (vlax-get-acad-object) "ModifyRoomNumber")
    (princ))
```

I hope you enjoyed this programming example. Stay tuned for the next installment on doing this same thing in Visual LISP, but this time using the ActiveX interface. If you have any questions or comments, please feel free to e-mail me.

R. Robert Bell is network administrator/programmer for MW Consulting Engineers in Spokane, Washington. He has used AutoCAD since v2.18 (AutoLISP!). He is on AUGI's Board of Directors and can be reached at robert.bell@augi.com.

Reader Service Information

Fast...Easy...Efficient

Welcome to a new, easier way to get more information about the companies, products, and services featured in this issue of **AUGIWorld**. Here's all you need to do:

Go to www.augi.com/adinfo. Here, you can:

- send personal emails to the companies found in this issue or previous issues
- visit the websites of the companies featured in **AUGIWorld**
- send a general email requesting more information from any of the companies mentioned in this and other issues of **AUGIWorld**

Communicating with software developers has never been easier! At www.augi.com/adinfo, you're a click away from any company featured in **AUGIWorld**.

Advertising Index

Company	Web Address	Page#
ACS Software	www.acssoftware.com	17
Autodesk Training Center	www.autodesk.com/atc	1
Autodesk University	www.autodesk.com/au	Back Cover
CADzation	www.cadzation.com	21
COADE, Inc.	www.coade.com	27
GTX Corporation	www.gtx.com	25
IMAGINiT Technologies	www.rand.com/imaginitt	Inside Back Cover
Motive Systems	www.motivesys.com	Inside Front Cover
Robert Steltman, Inc.	www.rsteltman.com	25
Transoft Solutions	www.parkcad.com	9

CADWorx®

Plant
Design
Suite

Easy
Open
Scalable

CADWorx PIPE provides the flexibility of 2D or 3D plant design, handling small jobs and complex projects with ease. It also offers bi-directional links to engineering analysis.



CADWorx P&ID makes the creation of intelligent process flow diagrams and automated instrumentation diagrams effortless.

CADWorx Equipment uses parametric tools that make equipment modeling and editing easy, accurate and fast.

COADE®
ENGINEERING SOFTWARE

1-800-899-8787
sales@coade.com

**DOWNLOAD
FREE DEMO**

www.coade.com

AutoCAD is a registered trademark of Autodesk, Inc. ©2003 COADE, Inc.

On The Back Page



David Kingsley

Making the Old New Again

I often write about the history of CAD, but this time I'm going to write a little about plain old history.

I have been working on a project that spans an entire century. The engineers that started it were born more than a century ago and spent the latter part of their careers in the 1950s. The folks that inherited it from them are just reaching retirement. The product they designed is still produced today. As I wrote when I first described the premise of this column, many times things are done right the first time and there is no good reason to change them. In this case, I am keeping that thought in mind and treading lightly on some respectable engineering documentation.

The products of which I speak are power tools used by auto body repair professionals. They were originally designed during WW2. As I understand it, these are the "Mercedes" of their type; they don't break or wear out.

As you may or may not know, GD&T, aka ANSI Y14.5, aka Geometric Dimensioning and Tolerancing, was developed during World War II to improve, and ultimately assure, parts interchangeability. It was critical that out in the field, sea, or sky, the parts from one mechanism were interchangeable with another of the same model. Prior to this, things were still somewhat "handcrafted" in this regard.

My task has been to transcribe a set of paper drawings into AutoCAD. On many of these drawings, the last "Engineering Change Order" was issued in 1954-1958. These parts have been manufactured for 50 years without an engineering modification. The documentation practices and discipline of the era are clearly evident in the drawing package. GD&T was developed in WW2, but is still not widely accepted today. As such, this drawing package did not incorporate it.

The new CEO of the manufacturer initially contacted me. His father had recently

retired and turned over the reigns of the company to him. He had an engineering degree and had been patiently waiting for his turn. He told me that we would be "leapfrogging the company from the 19th to the 21st century." Some of the senior manufacturing technicians were still working, but would soon be following their recently retired leader.



The CEO's task (and ultimately mine) was to capture all the product knowledge these people had in their heads and get it documented. To this point, most of the work was done in-house. He wanted to outsource more of it and transform the in-house operations into assembly and test only. He couldn't do that with the documentation package he had. Another impetus was that there had recently been a fire in the building, and their only set of original drawings had come very close to being destroyed.

I must admit that I made a bold attempt to use raster-to-vector conversion. There were a lot of linework changes to be made as well as incorporating GD&T dimensioning. I found the software tools to be reasonable, but I determined the entire process was just not suited for mechanical drawings. Even with all the raster scaling tools available, it was

very difficult to produce an accurate 2D model. It was far easier to just build the drawings from scratch. This is not to say that R2V conversion has no application. I know cartographers and civil engineers who couldn't work without it.

Marvin, the technician I worked with, is 78 years old and as sharp as a tack. I can only hope that I am as spry and quick of mind as he when I reach that age. He had every change, tolerance, and process methodically logged in a three-ring binder. He could answer any question on any of 200 drawings in a few page turns. Marvin has made a lot of metal chips and seen a lot of technological advances over his career. When he came to my office, I couldn't resist giving him a demo of Autodesk Inventor®. I banged out a couple of basic parts and then opened the engine assembly sample drawing. When I constrained the connecting rod to the crankshaft, and then dragged the crank and moved the piston up and down, he was awestruck. He immediately got it. Almost the next words out of his

mouth were, "You could build and test your entire prototype right on the computer, then you would almost never make a bad part!"

All in all, it has been an interesting experience bringing something really old into the new millennia. The most fun though, was watching Marvin's reaction to Inventor.

David Kingsley is senior vice president on the AUGI Board of Directors and can be reached at david.kingsley@augi.com.

IMAGINiT Technologies

CHOOSE, CUSTOMIZE AND IMPLEMENT
THE RIGHT TECHNOLOGY FOR
ALL YOUR ENGINEERING AND
DESIGN SOLUTIONS NEEDS

- Complete training, support and consulting services
- Best practices
- Business processes

The advantage of Global Reach with
Local Touch — IMAGINiT Technologies,
a Rand Worldwide company. For more
information please visit

www.rand.com/imaginit



IMAGINiT
TECHNOLOGIES
A RAND Worldwide Company

autodesk
authorized systems center

AUGI Members early registration for Autodesk University begins July 26th!

Many classes fill up quickly. Be among the first to review the new courses and secure your spot for AU 2004.

A BOOSTER SHOT FOR THE BRAIN



AUTODESK
UNIVERSITY
2004

12th Annual Autodesk University

The premier Autodesk learning and networking event

November 30–December 3
MGM Grand Conference Center
Las Vegas • USA

To register visit <http://www.autodesk.com/au>