Autodesk University 2007

EAT TEXT: Data Extraction in AutoCAD® with Novacaine

dave espinosa-aguilar – City of Richland, WA

GD205-4P: The new Data Extraction technology in AutoCAD provides amazing capabilities for the average office to generate intelligent and associative tables of drawing information gleaned from your linework. Discover how to link your design with spreadsheets and maintain bidirectional data flows for generating schedules, part lists, and bills of materials. Learn how to use Column Filters, Formula Columns, and other powerful features to report everything from survey points, alignment tables, and lot acreages to ductwork and pipe sizings, door and windows schedules, and facilities management costs. We'll also discuss undocumented tricks that will take the Extraction tools to the next level of functionality. It's everything you need to know about out-of-the-box Excel integration with AutoCAD.

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

Autodesk®


Autodesk
University
2007

Autodesk University 2007

EXCEL: An Old "AutoCAD Wishlist Item"

Since the very first release of AutoCAD was launched, users have been wanting Excel functionality built into it. There are obvious reasons for this since drafting and design data is often organized in columns and rows and reported this way to many agencies in the form of Schedules, Bill of Materials, Inventory Lists, Analysis Trends and Charts… the list goes on and on.

For the first decade of AutoCAD's existence, most users manually drew their own tables of information with OFFSETted LINEs and center-justified TEXT entities, painstakingly counting out and maintaining their data sets in this DWG-based file format. If the design changed, the whole mind-numbing process potentially started from scratch -- and this is why users searched and searched for clever ways to automate these tallying processes.

Excel has been around a long time. It does what it does extremely well, and it's on practically everyone's desktop already. In a typical office, there are plenty of people who don't own and/or use AutoCAD but who do use Excel daily (and very powerfully) and they have typically had no access to that column-row-based information stored in AutoCAD. The only reliable method for sharing that information was either to export somehow using commands like ATTEXT (ATTRIBute extract) or thru customized progamming (AutoLISP routines and the like). There were very few options that came in AutoCAD "out of the box" to make this kind of data-sharing possible.

In the second decade of AutoCAD's existence, numerous technologies came along which seemed to promise true integration with AutoCAD and Excel so that the data needed by users of both applications might reliably be exchanged. Unfortunately, DDE, OLE and other technologies like them all had severe limitations and very dangerous quirks about them – in a nutshell, it took an analy-retentive mega-geek to make these technologies work correctly (if at all) for anyone, and consequently very few organizations adopted these technologies as integral parts of their design and drafting production processes and strategies.

And yet, no one in the AutoCAD world denied that there was huge value and benefit to true integration between these two software products. Year after year, the AUGI AutoCAD Wishlist would say the unsayable ("we want Excel functionality in AutoCAD"), and slowly but surely, users began to see this request emerge in the form of more robust and stable OLE implementations, the introduction of TABLE entities, and the debut of specialized data extraction commands like EATTEXT. When Autodesk unveiled its latest version of AutoCAD, lo and behold, there it was: Data Extraction as a main feature set in the progam and potentially a huge reason to upgrade from prior releases.

As with any new major technology feature set introduced in any popular Windows application, Data Extraction in AutoCAD 2008 faced some serious challenges when it was first introduced. Several myths about it sprang up overnight regarding its misbehaviors, incompatibilities and bugs… some of these myths arising from true problems in the software that had not been anticipated by Autodesk (ex: Dynamic Blocks misbehaved with it), some of these myths arising from AutoCAD veterans who were struggling with the new interfaces and concepts and were flat out using them incorrectly or in ways the technology was never intended to be used. The exact same thing happened when Paperspace was first introduced. And Solids Modeling. And dbConnect. And the Sheet Set Manager. And and and and and.

Let's face it: some of AutoCAD's most revolutionary technologies are not intuitive on their first examination and they can require a lot of reading, playtime and experimentation to understand and properly implement. So did learning to ride a bicycle. With patience, practice, and perseverance, amazing productivity can result from all of these more sophisticated toolsets.

The first Service Pack for the AutoCAD 2008 product series has been out for a while now, it has resolved most of the problems originally found in Data Extraction, and everyone including Autodesk has been goofing around with it for months now. The discussion forums are alive with accurate information about how to use it, what to be careful with about it, and even advice on workarounds for the next desired wave of functionality to be included in it.

So crack your knuckles, remind yourself that you've mastered initially bewildering but eventually powerful concepts in the past with AutoCAD, and engage this new approach to handling data between AutoCASD and Excel today. Don’t just get your hands dirty with it. Get them filthy with it. Make a real mess of things. That's how PowerUsers do things. Sometimes a trek of this type can be downright painful at first as users struggle with their attachments to doing things the old way, the remembered pain of past implementations when technologis weren't fully understood, and the fear of boldly going where they have never gone before. This class is intended to minimize that atachment, that pain and that fear by exploring the

big benefits and obvious uses of the technology to anyone. The 'Novocaine," the new stuff we'll be injecting you with, will clarify and align this technology for you so that when you leave this dentist's office, you'll be smiling widely.

As I started every **Integration of AutoCAD VBA with Microsoft Excel** class at this event for the past 5 years, we begin this class by asking ourselves:

"What Are The Real Benefits of AutoCAD/Excel Integration?"

There are actually numerous benefits and uses for integrating Excel functionality with AutoCAD, and some are more obvious than others. But this class emphasizes three benefits in particular (and their potential uses) because these three benefits apply to anyone in any business scenario which involves AutoCAD.:

1. Small = Fast

The smaller a data file is in Windows, the faster it is to use. This is true of any Windows application. AutoCAD is never faster than the moment when you first launch it. s you begin added more and more linework to the session, things begin to slow down. This is why it takes longer to load, regenerate and save larger drawings than it does to load, regenerate and save an empty drawing. Anything you can do to minimize the sheer size of a DWG file equates to gained speed. By moving certain types of data out of AutoCAD (without losing access to that data in AutoCAD), you shrink the size of the data being handled by AutoCAD and therefore you increase the speed of AutoCAD.There is a sweetspot where this concept becomes truly beneficial depending n how large your files are and the typical types of data you store in AutoCAD… but the truth in this concept is undeniable: data size drives speed of data handling. Data Extraction makes it possible to move certain types of data in AutoCAD to a different location without losing that information in AutoCAD. This eternaliztion of data accelorates everything in AutoCAD.

2. Shared Data = Used Data

If you store information in a DWG file, and only people who own and know how to use AutoCAD can see and use that information, then you have digitally castrated your non-AutoCAD workforce. You may have excellent reasons for not making DWG information available to people outside the AutoCAD realm, but are you keeping that information hidden or locked down because it is a deliberate choice, or because you percieve there being no other reliable options? If the latter, why severely limit that information's potential to be used to its fullest and most productive extent? Information is everything these days, and when you can provide it in a format that everyone can see and understand and use, new techniques, processes and opportunities avail themselves to your entire organization. AutoCAD Map users (and indeed GIS engineer) have understood this principle since the product's inception, but this power is now available in all the AutoCAD 2008 flavors thru the Data Extraction technology. When your people have access to data, they get new ideas about that data which lead to new abilities with that data. And that tends to create new opportunities with that data… which generates new profits from that data. The equation can be that simple: make the data more available, and more people will use it. There is a saying that "if you build it, they will come." Well, the same is true of digital constructs: "if you expose them, they will come."

3. Excel Functionality = AutoCAD Functionality

You don't need to wait for AutoCAD to build literal Excel functionaliy into itself to benefit by Excel. If you already have Excel… use Excel! Excel does a ton of things that AutoCAD doesn't. It does them very well. This isn't a failing of AutoCAD The ttwo products simply do very different things very well. You can have the best of both works without requiring each to be the other. Consider for a moment how impractical it might be for an Excel user to want AutoCAD-type functionality in Excel. How long would that wishlist item last? Excel-type functionality is growing in AutoCAD… but will AutoCAD ever fully be Excel? Not likely. So the next best thing is to leverage the daylights out of Excel itself, as it stands, and find ways to merge that functionality with AutoCAD. That is exactly what Data Extraction provides. In fact (I'll go out on a limb here…), a better name for Data Extraction in AutoCAD would be **Data Integration** because the Data Extraction functions actually do far more than just extract data. They maintain data. They re-import data. They align data. They report data. They organize and format data. Once you have this kind of a connection between two applications, the wait is over for them to be each other. They are already each other, at full strength. So what does Excel do still that AutoCAD doesn't? Excel charts information in ways AutoCAD doesn't. It offers functions in its cells, ranges and spreadsheets which AutoCAD doesn't. It organizes and formats information in ways AutoCAD doesn't. This is not a problem for anyone. The more powerful your Excel skills get, the more powerful your integration with AutoCAD gets. Stop waiting on AutoCAD to become Excel, and focus instead on integrating Excel (and Word, and Access and any other Wndows application you typically use). AutoCAD may already be as powerful as your entire desktop.

This Is The Gold

The remainder of this class examines the various commands, techniques, and very sexy technology that makes Excel integration possible with AutoCAD, but without serious attention to the above concepts, it's all pointless. The real challenge to benefiting by Data Extraction technology comes from stepping back and re-examining your entire daily process and asking yourself how the above benefits might be applied to your typical work. Are there slow aspects to your AutoCAD workflows that would benefit by drawings being smaller? Externalize the data. Is the data you're storing in AutoCAD held hostage from other groups of people in your company who could beneficially be working with it? Get it to them. Is there functionality in Excel that you've been waiting on to be implemented in AutoCAD? Stop waiting. Play with Data Extraction. Experiment with it. Apply it to the real world of data you work with. It's going to take an expert in your office who knows how things are currently done to see how Data Extraction can streamline processes and expand the office's capabilities. And I'm betting that person is you.

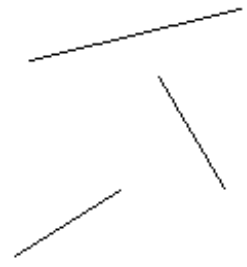
To get a grip on Data Extraction concepts and methods in AutoCAD 2008, let's start by examining data extraction methods in prior releases of AutoCAD, some powerful and clever techniques that came out of that primorial DWG soup, and how those techniques still apply to effectively govern Data Extraction in the latest release. If your brain is thinking "I came to learn about new stuff, not old stuff!" … maybe it's time to re-think the "old stuff."

ATTEXT, "Block Tagging", Proxies and "CrapCAD"

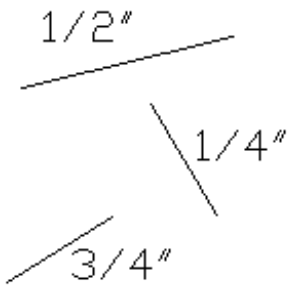
We start by examining one of the oldest data extraction technologies available in AutoCAD: the **ATTEXT** command. It requires an ASCII template file (.TXT) to be built which uses specialized codes (shown below) to designate which aspects of a BLOCK (with ATTRIButes—it must have ATTRIButes) should be exported. It's a one-way street with data… the command is invoked, the AutoCAD drawing is surveyed (the user selects which BLOCKs with ATTRIButes to examine), the template file is consulted and the information is exported in a CDF (Excel-compatible), SDF or DXF format. Fairly straightforward stuff.

BL:NAME Cwww000 (Block name)
BL:LEVEL Nwww000 (Block nesting level)
BL:X Nwwwddd(X coordinate of block insertion point)
BL:Y Nwwwddd(Y coordinate of block insertion point)
BL:Z Nwwwddd(Z coordinate of block insertion point)
BL:NUMBER Nwww000 (Block counter; the same for MINSERT)
BL:HANDLE Cwww000 (Block handle; the same for MINSERT)
BL:LAYER Cwww000 (Block insertion layer name)
BL:ORIENT Nwwwddd(Block rotation angle)
BL:XSCALE Nwwwddd(X scale factor)
BL:YSCALE Nwwwddd(Y scale factor)
BL:ZSCALE Nwwwddd(Z scale factor)
BL:XEXTRUDE Nwwwddd(X component of block extrusion direction)
BL:YEXTRUDE Nwwwddd(Y component of block extrusion direction)
BL:ZEXTRUDE Nwwwddd(Z component of block extrusion direction)
numeric Nwwwddd (Numeric attribute tag)
character Cwww000 (Character attribute tag)

If the drawing changed any, then the ATTEXT command would have to be re-used and the exported data would have to overwrite previous data. Also, if any linework in AutoCAD needed to be reported/exported but did not exist as a BLOCK with ATTRIButes, you were/are pretty much out of luck with the ATTEXT command.



The figure at the right shows three lines. If you needed their information reported (such as their "pipe sizes", there was no way to do this with ATTEXT unless you made each pipe into its own BLOCK with ATTRIButes. However, never tell an AutoCAD user a thing can't be done. Over the years, several clever techniques evolved which made reporting/exporting non-BLOCKs, BLOCKs-without-ATTRIButes and even collections of entities possible. The most popular technique of these was/is "Block Tagging."

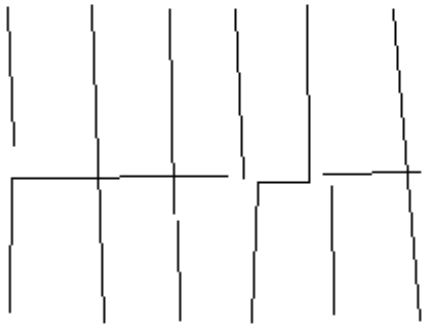


Briefly stated, "Block Tagging" is the use of legal BLOCKs with ATTRIButes as "proxies" for anything needed. If a thing exists (say a LINE, a CIRCLE, a POLYLINE , a BLOCK with no ATTRIButes or even a collection of entities) which ATTEXT won't recognize but still needs to report on, then a legal "tagging" BLOCK with ATTRIButes is INSERTed which represents it --- strictly at a data level. This tagging BLOCK is typically given its own frozen-at-plot-time

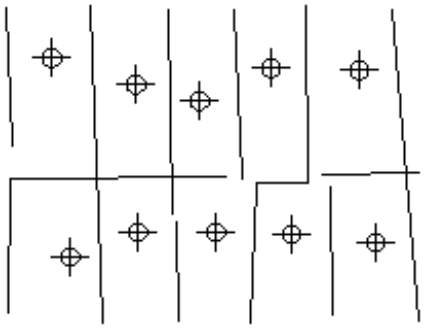
LAYER so that the tagging BLOCK doesn't plot, and the tagging BLOCK only is exported when you need information about the thing. There is no actual "link" between the tagging BLOCK and the thing it represents (except perhaps visual proximity to "associate" it with). An elegant workaround really -- which still has simple and profound ramifications for the Data Extraction in AutoCAD 2008. The figure on the right shows the same three "pipes" with nearby ATTRIBute-only BLOCKs reporting their pipe sizes.

The essential gem to keep handy from this era is this:

You can represent (and therefore export) any data element in AutoCAD if you simply establish a dedicated proxy element for it which has nothing to do with plotted linework.



Even though ATTEXT is an old and limited command, it gave birth to this very powerful concept. There will always be cases where the linework you need to report on does not perfectly accomodate data extraction the way you need it to. A common case: "CrapCAD." The figure to the right has the audacity of representing 10 parking lots. No data extraction commands in any release of AutoCAD will tell you there are 10 parking lots there. When circumstanes like these arise (and there's a LOT of CrapCAD out there), you need to be armed with technique as well as technologies. Data Extraction functionality in AutoCAD 2008 is extremely powerful, but it is no match for horrible drafting and therefore it cannot cover every possible scenario you might run into. And you definitely don't want to have to redraw an entire design if you're just trying to get the numbers out to someone. There is a much more efficient way to handle this reality in any release of AutoCAD.



Examine the figure to the right. Using MULTIPLE POINT, it would only take a user seconds to place POINT entities on a dedicated "counting" LAYER where every parking lot should be. The linework doesn't have to be edited either. In this case, a BLOCK isn't even used, just a POINT entity on a dedicated layer. ATTEXT cannot extract this information, but commands like FILTER can easily count the number of POINTs found on the dedicated LAYER. More importantly, Data Extraction functions in AutoCAD 2008 can also find these POINTs easily and tally them without any modification to the CrapCAD they represent. We saw this kind of power coming ou way when AutoCAD 2007 introduced …

"EAT TEXT"

The new EATTEXT ("extended ATTEXT") command **in AutoCAD 2007** took the old ATTEXT command to the next level by providing a Wizard dialog interface that made extraction parameters much easier to describe on the fly. The EATTEXT command still exists in AutoCAD 2008, but it launches the DATAEXTRACTION command now instead of the original AutoCAD 2007 Wizard. The AutoCAD 2007 implementation of EATTEXT only surveyed selected BLOCKs again, but by using "tagging BLOCKs" it could still report on anything in the drawing file. However, there were some very subtle but profound changes in EATTEXT:

- EATTEXT no longer required BLOCKs to have ATTRIButes in them. It could report on any BLOCKs.
- EATTEXT could filter which surveyed BLOCKs were reported (such as those only on a certain LAYER).
- EATTEXT could export the data either to AutoCAD TABLE entities or to CSV (Excel-compatible) files, or both.
- EATTEXT could use its own templates to remember the extraction parameters specified for re-use.
- EATTEXT could use TABLE templates to export to certain TABLE styles.
- The TABLEs EATTEXT extracted to were now associative. If BLOCK counts changed, their TABLEs updated!
- The order in which BLOCK data was reported could be previewed, sorted, reorganized and even hidden.
- BLOCKs with new FIELDs in their ATTRIButes could have FIELD data exported.
- The new Dynamic BLOCKs and their information could also be exported.
- EATTEXT could survey BLOCKs found in external sources outside the master drawing and report them as well.

The data extraction functionality in AutoCAD 2007 is still a one-way process really. If the design changes, then the data extraction process has to be re-done. Even though EATTEXT introduced associative TABLEs, the updating process does require a REGENeration. Sometimes a TABLE link does break and a TABLE has to be re-created. Even so, if you're a user of AutoCAD 2007, not only is EATTEXT excellent preparation for AutoCAD 2008 Data Extraction, but by using tagging BLOCKs with it, it is very powerful.

Three significant capabilities came out of the AutoCAD 2007 EATTEXT technology which lend themselves directly to efficient use of Data Extraction in AutoCAD 2008.

1. Data Extraction does not need to be a singular process. Instead of using it once on everything, it can be used several times on subsets of drawing elements with common properties in the drawing (thru the use of EATTEXT filtering).
2. With properties of the BLOCKs being reported in "hide-able" columns, a user can rely on a BLOCK property to collect and organize BLOCKs in a drawing but not necessarily show the criteria in the extracted data itself.
3. EATTEXT reports its surveyed information based on the unique combined visible values it reports. If items do not have unique data associated with them, then they are "collected" into summation rows and reported as single row counts.

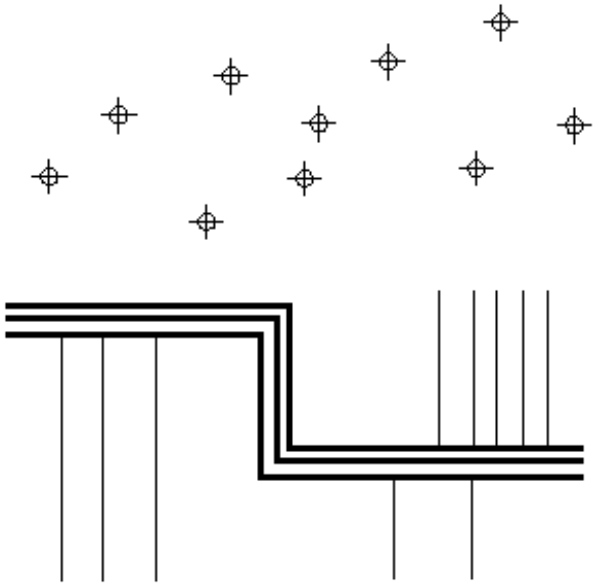
As with other sophisticated technologies when they are first introduced in new releases of AutoCAD, these types of revolutionary changes may demand that we re-examine our daily processes, our existing standards, and the way we create, organize and maintain drawings to exploit them fully. Consider how Paperspace radically overhauled the way we organize annotative and scaled linework. Data Extraction (DE) is that revolutionary, and you may find yourself overhauling your BLOCK libraries (since DE can report to any entity type now), your BLOCKs with ATTRIButes (since they aren't required for extraction anymore) and your TABLE styles (since they can accommodate new DE functionality).

DATAEXTRACTION

Sometimes, the best way to learn a new technology is to see it in action. What follows is a complete tutorial which goes thru fundamental data extraction concepts, step by step, in a reproducible and simple way. It requires AutoCAD 2008 (with Service Pack 1) and Microsoft Excel to be installed, and requires no supplemental files since everything in this tutorial is created and manipulated from scatch. By completing the tutorial, you will be introduced to majority of data extraction functionality, and it's enough to get you thinking seriously about your own potential implementations.

Preparation: this section sets up the folders, the files and the thinking before data extraction gets underway. This avoids the need for supplemental files to be used since the tutorial can be performed anywhere at any time with zero file and system preparations.

1. at the operating system level (Windows Explorer), create a folder called C:\DXESTUFF
2. in AutoCAD, create a new drawing. use **DDTYPE** to set the points mode to something easy to look at. Draw 5 points on the default Layer "0", save the drawing as DXE1.DWG in the new c:\dxestuff folder, and close the drawing. The figure to the right shows an example of DXE1.DWG.
3. in AutoCAD, create a new drawing. use RECTANG from 0,0 to 50,50 to draw a plinar rectangle. Zoom to extents. Create two layers, one named "1-2-pipe" and another named "3-4-pipe". Use **PLINE** with ORTHO turned on to draw an orthogonal zigzag PLINE on layer "3-4-pipe" inside the rectangle, representing a mainline pipe. Use OFFSET to create a second PLINE offset from the first.
4. use LINE command with ORTHO turned on to draw orthogonal LINEs of different lengths on layer "1-2-pipe" branching off the two polylines. Copy these lines several times so that each length has differing amounts of copies. Delete the rectangle, save this drawing as DXE2.DWG in the new c:\dxestuff folder, and close the drawing. A figure to the right shows an example of DXE2.DWG (PLINEs are shown boldfaced).



5. in AutoCAD, create a new drawing and save it as DXE3.DWG in the root directory (not the c:\dxestuff directory). You are now ready to begin.

Extraction: this section demonstrates how the DATAEXTRACTION command creates a populated TABLE in DXE3.DWG and an external Excel spreadsheet file DXE3.XLS from the two external drawings DXE1.DWG and DXE2.DWG. Data Extraction can operate on the current drawing, but it can also operate on other specified drawing files (UNC coding may be required especially for network accesses), entire directories of drawings, external reference files, and nested BLOCks.

1. in AutoCAD in the DXE3.DWG drawing, use the **DATAEXTRACTION** command.
2. You are on page 1 of the Data Extraction Wizard (the Begin page). Select the Create a New Data Extraction option and click the Next button. When prompted for a filename for the extraction file, use DXE-ALL.DXE.
3. You are on page 2 of the Wizard (Define Data Source). In the Data source frame, select Drawings/Sheet set option, uncheck Include current drawing toggle, and click the Settings button. The Additional Settings dialog opens to reveal that objects wll be extracted from BLOCKs, XREFs, and XREFs will not be included in the counts.The Extract from frame reports that All objects in the drawing (not just those in Modelspace) wll be extracted. Click OK. You are back to the Define Data Source page. Pick the Add Folder button. The Add Folder Options dialog appears. Pick the Browse button and point to C:\DXESTUFF. In the Options frame, Check the Automatically include new drawings added in this folder to the data extraction toggle. Also, Check the Include Subfolders toggle. Leave the Utilize wild card toggle off. Click OK. You can now see DXE1.DWG and DXE2.DWG reporting in the Drawing folders and files frame. Click Next.
4. You are on page 3 of the Wizard (Select Objects). Right -click over any column to see the four options (Check All, Uncheck All, Invert selection and Edit Display Name). Lines, Points and Polylines are shown and checked. By double-clicking on a column header, you can change the sorting order from Ascending to Descending. In the Display Options frame below, notice that you can determine if all objects or BLOCK (or non-BLOCKs) only are treated. You can also isolate only those BLOCKs with ATTRIButes, or those objects currently in use. Leave everything here with their default settings and click Next.
5. You are on page 4 of the Wizard (Select Properties). By rich-clicking ov er a column, the same 4 selection/naming options are available as the previous page. All properties derived from LINEs, POINTs, and PLINEs (the objects availabl) are shown and divided into 5 categories:
    - 3d visualization: materials,
    - drawing: think DWGPROPS including custom fields (ex: drafter)
    - general: layer, color, ltype, linewt, plot sty, thick
    - geometry: coords, length, angle, area
    - misc: closed

    Leaving all 5 categories checked (active), right-click over any column and Uncheck All. Then manually check Layer (a common property to all three entity types), Length (a property from the LINEs and PLINEs), and Position X, Position Y, and Position Z (a property of the POINTs). Click Next.
6. You are on page 5 of the Wizard (Refine Data). All available data values are reported according to your selections. Count column reports each LINE, POINT and PLINE as a unique row. By picking on a column header and moving it over another header, you can swap the order of columns, and as with similar columns, by double-picking on a header, you can change its sorting order from Ascending to Descending. Sorting can have nested criteria to it. Fo example, you can sort by Name column alphabetical values first, and then by Position X values second. You can do this either by picking the Sort Columns Options button, or by right-clicking over a column and selecting Sort Columns Options. Bring up the Sort Columns Options dialog. Select the Column "Select column" combobox and pick "Name". Leave the Adjacent Order combobox value at "Ascending." Pick the Add button which adds another row. Pick the new "select column" combobox and set its value to "Position X". Leave its adjacent Order combobox value at "Ascending." You can remove or re-organized these sorting elements at any time. Click OK. Notice now that LINEs come before POINTs which come before PLINEs, and that the Points are sorted according to ascending values. Richt-click over the Position X header, and use Rename Column to name it "Easting." Use the same process to rename the Position Y column to "Northing" and the Position Z column to "Elevation." Right-click over the Name header and use the Hide Column function. The Name column no longer shows. Right-click over any header and use the Show Hideen Columns/Show All Hideen Columns function. Al hideen columns are visible again. Notice that LINEs and PLINEs report counts greater than 1. This is because the Combine Identical Rows toggle is checked in the bottom left area of the dialog. If you uncheck this toggle, every element is listed with its own row. You can also hide the Counbt column and Name column by checking the other two corresponding toggles. For now, leave all three toggles checked. Click the Full Preview button to see the resulting tabular data. We will return to the Link External Data button later. Click Next.
7. You are on page 6 of the Wizard (Choose Output). Check both the Insert data extraction table into drawing toggle and the Output data to external file toggle. For the external file value, use C:\DXESTUFF\DXE-ALL.XLS. Click Next.
8. You are on page 7 of the Wizard (Table Style). For now, leave all default values alone and Click Next.

9. You are on page 8 of the Wizard. At this point or any other page you can us the Back button to reconfigure the extraction. For now, click the Finish button. Pick a point in the AutoCAD graphics screen area for the upper left corn er of the new TABLE to be inserted. A notification bubble alerts you that Data Extraction updates may be required if other linked TABLEs are in the session. You can zoom to extents now to see the resulting linked TABLE, and we are finished with the Wizard.
10. Minimize your AutoCAD session, launch Excel, and open the new DXE-ALL.XLS file to verify the file export. You will see the exact same tabular data in the opening Sheet. Also notice that if you highlight all cells with values in them, a named range "Summary" has also been created in the Sheet. Close Excel.

Updates: this section demonstrates how changes to the drawings involved in a data extraction can automatically update in linked tables and external spreadsheet files. Key concept here: linked TABLEs are updated with AutoCAD link-updating functions (like DATALINKUPDATE or the Update All Data Links function in the status bar link icon). External files on the other hand are updated by re-using the DATAEXTRACTION command on an existing Data Extraction file (DXE). This two-fold process confuses more people than any othe concept when they are first introduced to Data Extraction. Therefore, the tutorial will break this process down into two processes.

To update the TABLE:
1. save DXE3.DWG (now with the new TABLE in it) and close the drawing.
2. open DXE1.DWG, select all the POINTs and COPY them to double the number of POINTs.
3. save DXE1.DWG and close the drawing.
4. open DXE3.DWG. Notice that the table is not updated (it shows the original count of POINTs). In the status bar lower right hand area is a new Link icon. if you right click on it, two options become available. Select the Update All Data Links option. The TABLE suddenly updates to reflect the new number of POINTs.

To update the Excel Spreadsheet:
5. Minimize AutoCAD and launch Excel. Open DXE-ALL.XLS and notice that this fle has not been updated with the new POINTs count even though the AutoCAD TABLE has been updated. Updating TABLES and Excel files is a two-fold process, and unlike TABLE associative behavior, you literally need to overwrite the old XLS fle to update it. To do this, use the **DATAEXTRACTION** command to start the Wizard again. This time on page 1, select Edit an existing data extraction and use the browse button by this option to select the DXE-ALL.DXE file. Click Next. Continue to Click Next thru all of the pages until you get to page 6 (Choose Output). Uncheck Insert data extraction table into drawing since you already have a TABLE in your drawing), but leave the Output data to external file toggle checked. Make sure it is pointing to the existing C:\DXESTUFF\DXE-ALL.XLS file. If prompted to overwrite it, say Yes. Click Next. Click Finish.
6. Minimize AutoCAD and launch Excel. Open the XE-ALL.XLS and notice that now the latest extraction values are there. So if you ever want to "update" an existing exported spreadsheet, use **DATAEXTRACTION** to re-export and overwrite the existing one. Using typical "Update All Links" functions (in the status bar, pulldown menus, pop up menus, dialogs and other interfaces) do not accomplish this. The online documentation makes this process very clear. It requires **DATAEXTRACTION** (run on an existing DXE file) to "update" (overwrite) existing XLS files.

Filtering: this section demonstrates how filtering functionality in Data Extraction can isolate reported elements in a design, and therefore isolate elements reported in a TABLE and in external spreadsheets. It begins by isolating POINT entities in the design and extracting their information in the original TABLE and the original exported XLS file by modifying the existing extraction file (C:\DXESTUFF\DXE-ALL.DXE). A new secondary extraction file will use the modified original extraction file as a template for isolating piping elements so that most of the parameters are already configured and the process can move along more quickly.

First, the survey points (the existing extraction and associated TABLE will be sized down to only report on survey points):
1. In AutoCAD, use the **DATAEXTRACTION** command. On page 1 of the Wizard, select Edit an existing data extraction option and point to C:\DXESTUFF\DXE-ALL.DXE.

   Note: Here we are recycling the DXE-ALL.DXE for this survey points extraction. It goes without saying that you could also leave the original extraction file DXE-ALL.DXE alone (which extracts everything) and instead create a new extraction (ex: DXE-POINTS.DXE) from it which only extracts points. This process is outlined below in steps 3 and 4, but for this step, let us modify DXE-ALL.DXE to only handle POINTs so that you can see the process of modifying an existing extraction file and see the resulting TABLE behavior.

Click Next till you reach page 3. Check POINTs only. On page 4, select Position X, Position Y and Position Z only. On page 5, uncheck Show Name Column and Show Count Column toggles and hide all columns except Position X, Position Y and Position Z. If needed, rename Position X column to Easting, Position Y column to Northing and Position Z colyumn to Elevation. On page 6, do not insert a TABLE since you already have one) but do output data to an external file. Save this file as C:\DXESTUFF\DXE-POINTS.XLS. Finish out the Wizard.

2. The original TABLE (which once reported everything) now only shows Positional information for the POINTs. The notification bubble alerts you to the update, and if you minimize AutoCAD and view the new DXE-POINTS.XLS file, it will only show the POINT information. The original DXE-ALL.XLS file remains untouched even though the extraction file that created it has been modified. Close all XLS files, widen the AutoCAD TABLE and size the rows equally if needed.

Second, the piping (a new extraction will use the existing one as a template):

3. In AutoCAD, use the **DATAEXTRACTION** command. On page 1 of the Wizard, select Create a new data extraction option, check Use previous extraction as a template toggle and point to C:\DXESTUFF\DXE-ALL.DXE. You'll be prompted for a new extraction file name. Use C:\DXESTUFF\DXE-PIPES.DXE. Click Next till you reach page 3. Check LINEs and POLYLINEs only. On page 4, select Layer and Length only. On page 5, unhide all columns and then hide everything but the Count, Layer and Length columns. On page 6, do insert a new TABLE since you don't have one for piping yet. Also do output data to an external file. Save this file as C:\DXESTUFF\DXE-PIPES.XLS. Finish out the Wizard.
4. The new TABLE only shows pipe information now for the LINEs and PLINEs. If you minimize AutoCAD and view the new DXE-PIPES.XLS file, it will only show the piping information. Close the XLS file, widen the AutoCAD TABLE and size the rows equally if needed.
5. In AutoCAD, use the **DATAEXTRACTION** command. On page 1. use Edit an existing data extraction option and point to the new DXE-PIPES.DXE file. Click Next till page 5. Right-click over any column and select Combine Record Mode/Sum Values. Notice how all unique Layer values ("3-4-pipe" and "1-2-pipe") are now totalled in single rows. Click Next. Do not insert a new TABLE (since you have one already) but do Output data to external file (DXE-PIPES.XLS again) and finish out the Wizard. The notification bubble updates the TABLE, and if you verify, the DXE-PIPES.XLS file has been updated to reflect the summed rows as well.

Linking: this section demonstrates how to use the Data Link Manager to merge "matching" external spreadsheet data with data extraction values to produce new calculated tabular data. In this example, the pipes have a cost associated with their sizes and their lengths (all of which are currently extracted). The basic equation for calculating the cost of any size pipe in this application is <pipe length> * <price of unit length>. New columns will be created in data extraction which import the pricing information per pipe layer and which multiply the corresponding price by the pipe length in each row. A final total cost of all piping will be calculated as well. Once this Link is created, the completed cost of the piping in the design can be determined and updated as the design evolves. Nifty.

Note: Linking requires one shared unique column value in each data realm (the data extraction values and the external spreadsheet values) in order to match up the imported spreadsheet row content correctly. In this example, the layer names will be used for matching row values.

1. Launch Excel, and in Sheet 1 put the following values in these cells:

A1: Layer	B1: Price
A2: 3-4-pipe	B2: 2.0
A2: 1-2-pipe	B3: 1.0

2. save the file in the C:\DXESTUFF folder as PIPECOST.XLS, and close Excel.
3. In AutoCAD, use the **DATATEXTRACTION** command. On page 1, use the Edit an existing data extraction option and point to DXE-PIPES.DXE. Click Next all the way to page 5. Click the Link External Data button.
4. You are now in the Link External data dialog. To choose the source of the external data, you need to set up a Link with the pricing spreadsheet and tell it where in that sheet the pricing information can be found. Click on the Launch Data Link Manager button.
5. You are now in the Data Link Manager dialog (which can also be invoked with the **DATALINK** command). All existing links are reported here in the Links frame. Pick on the Create a new Excel Data Link option and provide the name "pipecost." A mini dialog will appear. Pick the browse button to point to the C:\DXESTUFF\PIPECOST.XLS file. You are now in yet another dialog where you can select the location of the data in the XLS file. You can select Entire Sheet, a named range (if it exists in the sheet already), or a specified range using cell values. Select Link Entire Sheet. Notice atg the bottom right corner of this dialog is an expand button (>). Pick it to see additional cellgoverning options

such as cell formatting. Now pick OK. You are back in the main Data Link Manager dialog and "pipecost" is now listed as a valid link. Click Ok

6. You are now back at the original Link External Data dialog, and if you click on the Data link combobox (which formerly had a None value) you will now see pipecost as an option. As soon as you select it, most controls in this dialog come alive. In the Data matching frame, click on the Drawing data column combobox. All available column headers for ata extraction are listed. Select Layer since that is the common unique "row value" in your current data extraction and the PIPECOST.XLS file. Click on the External data column combobox and both column top row values found in PIPECOST.XLS are listed.

   Note: there is a toggle at the bottom of this dialog which reads Use top row of external data as column names. When it is checked, you can use the top row of an external spreadsheet as a column header name for easy data matching.

   Select Layer again. As soon as Layer is selected in both Data matching comboboxes, the Check Match button becomes available. Click it. An alert of "The Key pairing was successful" should appear. This button makes it possible to check for unique shared row values in both worlds. If you ever get an error, it typically means you don't have unique values shared between both worlds.
7. in the Additional columns for data table frame are listed all column header values found in the external spreadsheet (Layer and Price). We only want the Price values aded to the data extraction (per unique layer), so uncheck Layer but leave Price checked. Click OK. Now the magic starts revealing itself. Look at the data extraction grid. The Layer and new Price columns both have a fancy link icon in their headers indicating that the Layers column values are matched between the two realms and the Prices values are being properly imported per layer.
8. Right-click over any column, and select Insert Formula Column. This brings up a mini-dialog. Type "Totals" for the Column name. Double-pick the Length entry so that the formula window reports <Length>. Now pick the asterisk button once (multiplication) so that the formula window reports <Length> *. Finally, double-pick the Price entry so that the formula window reports <Length> * <Price>. Pick the Validate button to see if this formula is legal. It is. Click OK.
9. Examine the new "Totals" column. It correctly multiples the extracted length values with the associated pricing values per layer. Right-click over the new totals column, and select Insert Total Footer/Sum. A new row with a sole total value appears. Right-click over the same column again and select Set Column Data Format option. This mini-dialog looks very similar to the formatting dialog found in the FIELDs dialog and works pretty much the ame way. Select "Currency" from the data types window and click OK. The totals now show dollar values.
10. Finish out the Wizard, not inserting a TABLE but outputting the DXE-PIPES.XLS file and confirm the updated data. Beautiful ain't it?

External File Updates: the magic wouldn't be complete without making an external data file change and seeing it reflected in our new formula-based TABLE. This section modifies the linked spreadsheet pricing data and shows theupdating results in the TABLE and resulting exported spreadsheet.

1. Launch Excel and open the PIPECOST.XLS file. Change the 3-4-pipe price to 10.0 and the 1-2-pipe price to 5.0. save the file and exit Excel.
2. A notification bubble alerts us of the change. Use **DATALINKUPDATE** command (or any other interface like the status bar or TABLE properties) to Update data links (all data links) and the new pricing information now reports in the TABLE along with the newly calculated costs.
3. Save DXE3.DWG and close the drawing. open DXE2.DWG and make some modifications to the linework: stretch some LINEs, copy more lines, and perhaps OFFSET one of the PLINEs.
4. Save DXE2.DWG and close it. Open DXE3.DWG. Use **DATALINKUPDATE** again to update the TABLE. The new length and count values (as well as all calculations) are updated.

Final Trick: This section demostrates two more important ideas: it is possble to "lock down" a TABLE so that it does not update any further, and it is possible to use the TABLE command itself to accomplish the Linking process.

1. Click inside the TABLE, and within the TABLE select all the cells that reflect linked data only (if you hover the cursor over the cells, you will see the locked value icon appear over them). Right-click to see TABLE options including Data Extraction which has three suboptions including Detach Data Link. Select this detaching option and the TABLE is now "dead" and will not update any further. If you need a TABLE to update again, Use **DATAEXTRACTION** command with an existing extraction file to re-create a linked associative updating TABLE and toast the dead one. This is a most reliable and lazy way to accomplish this task.
2. Use the TABLE command itself to bring up the TABLE dialog and notice that its Insert options frame includes "From a Data Link" (and pipecost is listed in its combobox). If you select this option, the pricing information will be brought in as a TABLE (which can be a very handy reference!). If you select "From object data in the drawing" you are basically issuing the **DATAEXTRACTION** command and away you go with the Wizard to create your TABLE. There are a lot of ways in which these commands and functions can be accessed. Playtime will make them all known to you.

Data Abstraction

Things to consider now that the essentials have been tackled:

- **DATAEXTRACTION** has a **–DATAEXTRACTION** version of itself… which means it is scriptable!
- The **DXEVAL** system variable controls data extraction notification. Its integer values are:

0	No notification
1	Open
2	Save
4	Plot
8	Publish
16	eTransmit/Archive
32	Save with Automatic Update
64	Plot with Automatic Update
128	Publish with Automatic Update
256	eTransmit/Archive with Automatic Update

- **DATAEXTRACTION** can use DXE files or the older BLK extraction format files.
- **DATAEXTRACTION** can maintain Excel cell formatting
- if a data link misbehaves due to files moving or data matching not behaving, toast it and recreate it.

The remainder of the class focuses on several real world applications of this technology including sophisticated PowerUser techniques which implement:

- Dynamic Blocks (tagging BLOCKs)
- Proxy Entities
- Hyperlinked pseudo-ATTRIButes
- FIELD TABLE-Referencing
- Optimizations