"What is the relationship between CADKEY and Reference Point, anyway?"

Since the appearance of the first issue of *Reference Point*, this question, in one form or another, is the one that has been asked of me *to the exclusion of all others.*

I would like to take the opportunity to respond to the direct, and underlying, questions. You, the reader, deserve to know exactly where the editorial control of *Reference Point* lies, whose "voice" is speaking to you. The answer is not a simple one, for the editorial control of the publication is shared between me and various people at CADKEY.

Under the terms of an agreement between CADKEY and Moore/Weinrich Architects, I provide to CADKEY "camera ready art" which they print and distribute. The name and format of the publication, though designed by me, are owned by CADKEY. The editorial content is of my choosing, but is subject to review by CADKEY.

The review process has not been formalized into any hard-and-set structure. Rather, it is intended to be responsive to the particular issues at hand.

When the publication was first being designed, I made a set of proposals suggesting format, regular features, and editorial emphasis (i.e. the focus on *technique*). People from CADKEY, primarily Lou Bodnar, Clay Rogers, and Frank Simpson, provided feedback and made suggestions based on their perceptions of the needs of the DataCAD user base.

As the articles for the first issue were written, I passed them first to Clay Rogers, who, along with Mark Hyjek, Mark White, and Ken Parish, reviewed them for technical accuracy. This group also made suggestions for improving the *techniques* described.

This process resulted in minor revisions to a couple of the articles and major changes to the overall content of one: the piece on cutting a door into a curved wall.

Technical input from CADKEY was essential to maintaining the level of accuracy desired by everyone involved. Debate over the validity of particular solutions to drawing problems resulted in a more comprehensive presentation than my first draft accomplished.

The final step in the review process consisted of copy editing by Frank Simpson. He reviewed the final layout for spelling, grammar, and format. He made a number of suggestions that improved the quality of the final product.

The relationship between me and CADKEY is analagous to that of an architect and his/her client. In both cases, considerable give-and-take is required in the process of bringing the finished "project" to reality.

Production of *Reference Point* entails a considerable amount of "review and redesign." Overall, I feel that the level of input from CADKEY, both technical and editorial, has not only been appropriate, but essential to establishing and maintaining the level of quality that I want *Reference Point* to have.

Reference Point is not, by its nature, a vehicle for opinion. My primary responsibility, as I see it, is to provide technical information to DataCAD users. Generally, accomplishing this does not involve the expression of my opinions. However, there are subject areas in which clearly stated opinions are not only unavoidable, but important to the process of communicating information. I can only say that, when appropriate, my opinions will be clearly expressed in these pages.

Philip Hart

CONTENTS

DOING WINDOWS

The installation of Microsoft Windows® is a relatively straightforward process. Many mail order houses sell systems with Windows pre-installed. Dealers will load it as part of a purchase agreement. Users can install it using Microsoft's comprehensive Setup routine.

Windows requires that the DOS configuration be optimized for it as the primary DOS application.

Windows controls so many aspects of the hardware resources of the system that it is crucial that its installation be correct. For Windows to operate efficiently, it should not be forced to compete for resources with other DOS applications.

Assuming that Windows has been properly installed and configured, the installation of Windows applications is very simple. Very little modification of the DOS-level configuration (typically, only the addition of directories in the AUTOEXEC.BAT Path statement) is required. These applications relate directly to Windows, relying upon it for graphics drivers, printer drivers, etc. They do so by reading (and writing, at the point of installation) entries in two Windows "initialization" files: SYSTEM.INI and WIN.INI.

Other .INI files are used by Windows to hold settings for its own applications. Additionally, many Windows applications include .INI files for their own use. SYSTEM.INI and WIN.INI, however, are the files containing the settings that control the basic operation of Windows. The settings in WIN.INI pertain primarily to user preferences, those in SYSTEM.INI to hardware set-up.

Many of the entries in the .INI files are modified through menu settings in Windows (Control Panel, Setup, etc.). As they are ASCII text files, they are easily edited directly by the user, as well. Documentation, on a line-by-line basis, of the entries in these files is contained in a series of .TXT files that the user should print out and study as needed.

Fine tuning of the operation of Windows (and, therefore, Windows applications) is accomplished by changing settings established in AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, and SYSTEM.INI. The user can edit them using Notepad in Windows or any ASCII editor.

Additionally, Microsoft supplies with Windows a very nice (undocumented) utility, System Edit. When it is run from Windows (by selecting in Program Manager: File; Run; and specifying \system\sysedit.exe as the file to run), a program window is opened containing four document windows: AUTOEXEC.BAT, CONFIG.SYS, SYS.INI, and WIN.INI.

Each file may be directly edited in its document window. If changes are made to a file, upon exiting System Edit, the user is asked if he/she wishes to save the changes made to the particular file(s).

The user can simultaneously use Notepad to open and display the contents of the appropriate .TXT file as a guide while editing either of the .INI files.

Creating a program icon for System Edit makes the utility quickly available. The following steps enable the user to do this:

In Program Manager, select File, then New

Select Program Item

Click OK

Enter: System Edit in the Description dialog box

Enter: c:\windows\system\sysedit.exe in the Command Line dialog box

Click on Change Icon to scroll through available icons. Click on OK to select the desired one.

Click OK to exit

A new icon labelled System Edit should now appear in the Program Group window that was active when the process was begun. It may be dragged to any Group window.

Changes made to AUTOEXEC.BAT or to CONFIG.SYS require that the user reboot the system for them to take effect. Changes to the .INI files generally require that the user exit and then re-enter Windows for them to take effect.

The process of fine-tuning a Windows configuration requires a lot of trial-and-error work; there are *many* settings in the .INI files. Fortunately, they are reasonably well documented in the .TXT files and, once set correctly, control the operation of Windows and *all installed Windows applications.*

UPGRADING TO DOS 5.0

Improved performance provides a significant reason for the **DataCAD** user to upgrade to DOS 5.0. As its release has been widely covered by the computer press, only one new feature will be discussed here: *DOS 5.0 can be installed to occupy less space than its predecessors.*

On 286, 386, and 486 systems with at least one megabyte of RAM, a portion of the DOS 5.0 operating code may be loaded into the high memory area (between 640k and 1024k). Compared to DOS 3.3, roughly 32k more *conventional memory* is made available to applications. As this region of RAM is where applications actually operate, more available RAM equates to faster program execution.

Two possible reasons for *not* upgrading:

- DOS 5.0 has compatibility problems with some early Phoenix 286 BIOS versions.

- Some older software may not run properly. Users should check with the manufacturers of applications installed on their systems to assure compatibility with DOS 5.0. Particular attention should be paid to hard disk utilities, communications software, and DOS shells. Undoubtably, upgrades are available.

Microsoft has provided a safety net: The Setup process *does not* overwrite the existing version of DOS (it saves it to a new sub-directory). It also creates a bootable floppy with which 5.0 may be un-installed. Thus, the user may easily restore his/her system to the previous DOS version should 5.0 present problems.

On most systems, the installation routine for DOS 5.0 (optionally) modifies CONFIG.SYS, adding, among other things, the following lines:

DEVICE=HIMEM.SYS
DOS=HIGH

HIMEM.SYS is Microsoft's memory manager; it enables DOS=HIGH, which in turn loads most of the DOS code to high memory. Other memory managers are reported to be capable of enabling the DOS=HIGH statement; users of QEMM-386®, 386Max®, etc. should experiment.

With no other changes to the system configuration, the **DataCAD** user should realize a gain of approximately 32k RAM in conventional memory. This equates to 7-8 additional page frames available for allocation by **DataCAD** and *noticeably improved performance.*

Editor's Note:
On my own system, I went from 32 page frames available to DataCAD under DOS 3.3 to 40 page frames under DOS 5.0, using Quartedeck's QEMM-386. The performance gain is noticeable, even on a fast machine.

TECH SUPPORT

The importance of a really "clean" transfer of data between DataCAD and another CAD program (used by a consultant, for instance) cannot be overstated. DXF file transfer can be a source of difficulties for the DataCAD user. The following article was prepared by Clay Rogers and addresses the major areas in which problems are known to occur. Where appropriate, the editor has added commentary from the user's point of view; these comments appear in italicized text.

Drawing eXchange Format (DXF) is an intermediary file format through which CAD systems may transfer drawing information to one another. DXF is one of a number of intermediary standards for CAD, but it is by far the most prevalent one. The DXF standard is defined by AutoDesk and updated with new releases of AutoCAD®. The DataCAD translator currently works to most of the standards established by AutoCAD Release 10.

The need for a commonly agreed upon standard for CAD file transfer is evident when one looks at the

number of CAD systems on the market (roughly 90-100 major systems). If each system were required to communicate directly with each of the others, their programmers would spend all of their time writing translators. By accepting one standard, each CAD vendor can focus its efforts on providing a good translator to read and write to that standard alone.

Although the majority of people using DataCAD's DXF translator are able to transfer data with little or no problems, there have been requests from a number of people for a description of the known difficulties. There are a number of quirks associated with transferring CAD drawings through DXF that are not necessarily caused by "bugs." DXF is a "bottleneck" and it will always have limitations that may cause minor problems. Some are experienced in writing a DXF file from a **DataCAD** drawing, while others are experienced in reading DXF files into **DataCAD**. Below are a few notes concerning the transfer of data through DXF.

WRITING TO DXF

Mirroring Symbols

When you use DXF to translate a drawing that contains mirrored symbols you will experience unpredictable results. After passing through DXF the symbols often appear rotated rather than mirrored.

Do not use mirrored symbols in your drawings. Since objects are typically not mirror images of themselves anyway, you should create a second symbol that is the mirror image of the first. If you have a symbol that you want to place as a mirror image of its current form, try the following procedure. By using this method you can avoid the unpredictable results of mirroring symbols.

1. Turn on EXPLODE in the Template menu.
2. Place an instance of the symbol you wish to mirror in the drawing.
3. Mirror it about the desired axis.
4. Save it as a new symbol with a different name.
5. You now have two symbols that are mirror images of one another.
6. Turn EXPLODE off.
7. Insert the new symbol as desired within the drawing.

This procedure does not work if the symbol that you are exploding contains instances of other symbols. This is known as a "nested" symbol. When a nested symbol is exploded, only the highest level is exploded. The symbol(s) within the symbol remains a symbol.

Attempts to explode previously placed mirrored symbols as a solution to this problem will usually result in a similar problem. The entities resulting from the "explosion" will revert to their "as created" orientation, appearing to have rotated about the symbol's insertion point. This is true for all of the currently available "symbol exploder" macros with the exception of the one included in The Detail Master°, which will explode mirrored symbols to their "as mirrored" orientation.

Symbol Attributes

Symbol attributes are neither written nor read through DataCAD's DXF translator. This creates problems when attempting to import symbol libraries from building component manufacturers. All non-graphic data such as part number, description, and cost are stripped off making the symbols less than useful.

User-Defined Linetypes

DataCAD supports user-defined linetypes (insulation, shingle, shiplap,etc.) but AutoCAD does not. If you transfer a drawing that contains user-defined linetypes from **DataCAD** to AutoCAD, these lines will become solid.

Only the following linetypes translate properly: solid, dotted, dashed, and dot-dash.

Text Transfer Limitations

A common problem with transferring DXF files relates to text. The text font name, size and aspect ratio settings interact to affect the overall size of the text. The symptom of the problem usually manifests itself as text not fitting properly in the same space it occupied prior to being transferred through DXF. This can be the result of using a different text font in **DataCAD** and AutoCAD or it may be a result of how DataCAD and AutoCAD differ in their interpretation of the size of a given text font.

Text Size (Height): The size for most fonts in DataCAD (except ROMAN2 and ORIG2) corresponds to the distance from the bottom of a descending character to the top of a capital letter. In AutoCAD the text size is from the bottom of a capital letter to the top.

When you transfer a drawing containing text from **DataCAD** to AutoCAD the resulting text will be larger unless the font used in DataCAD is either ROMAN2 or ORIG2. These two DataCAD fonts interpret the size setting in the same way that AutoCAD fonts do.

Aspect Ratio: For text of normal proportions, based on the font definition, the aspect ratio should be set to 1. Changing the aspect ratio affects the relative height to width of the characters. Since the aspect ratio affects the width of the characters based on the height, if the text size (height) is wrong the width of the characters, and therefore the length of the line of characters, will also be wrong.

Another thing to be aware of is that the value representing the aspect ratio in **DataCAD** has an opposite effect of the same setting in AutoCAD. In **DataCAD** as the value decreases the character width increases, but in AutoCAD, as the value decreases the character width decreases. **DataCAD's** DXF translator compensates for this difference.

READING FROM DXF

ReadDXF/Scale

DataCAD users will often read DXF files into DataCAD at the wrong scale. The symptom may appear as a bunch of small x's (symbol insertion points) displayed on top of one another. Another symptom is the display scale may go to 1:1000 and the message line read "Invalid View Specification" with nothing displayed. This is because **DataCAD** automatically performs a "WindowIn/Extents" and the drawing may be covering an area larger than can be displayed at 1:1,000 scale.

If the drawing covers an area larger than what can be displayed at the largest predefined scale (1:1,000) it may require you to change one of your existing scales to 1:10,000 or 1:50,000. This can be performed in the submenu under Settings/EditDefs/Scales/Change.

In the ReadDXF menu the option Scale can be selected. When picked, it asks you: "What is the value of one DXF file unit?" What it is asking for is the value of one "unit" in the AutoCAD drawing. In AutoCAD, the value of a "unit" is arbitrary and user definable. By default, one "unit" in AutoCAD is equal to one inch. This is also the default setting when reading a DXF file into DataCAD. However, it is not uncommon, with large building or site plans generated in AutoCAD, to have one "unit" equal to one foot. In cases like this, when you read the DXF file into DataCAD you should change Scale in ReadDXF to 1.0 (1 foot) instead of 0.1 (1 inch).

Correcting Fatal Errors Caused by Astronomical Coordinates

Periodically, tech support gets calls from people that have trouble editing DXF files they have read into **DataCAD** because of the absolute coordinates of the drawing. This can result in system crashes and Fatal Error messages (Real Math Overflow). If you are experiencing such problems you should check the coordinates of the entities in your drawing. If the coordinates of the entities in the drawing are astronomical you will need to move the entire drawing closer to absolute 0,0.

Coordinate Identification: Examine the drawing and pick an entity near the center of your drawing (in Ortho/Plan view). Perform a Coordinate Identification by pressing the **?** key

and then snapping to the point on the entity you want to identify. If the X=800,000 and Y=1,200,000 or some other very large numbers you should move the entire drawing closer to absolute 0,0.

Move Drawing to Absolute 0,0: Make a note of the X and Y coordinates that you identified at the center of the drawing. Then press the <Ins> key until you have selected Absolute Cartesian Input Mode. Go to the Move menu and snap to the point at the center of the drawing as the "first point of the distance to move." For the "second point of the distance to move" press the space bar and enter an X value of 0 and a Y value of 0. Then select the entire drawing by Area with LyrSrch on. This will move the entire drawing to an area surrounding absolute 0,0.
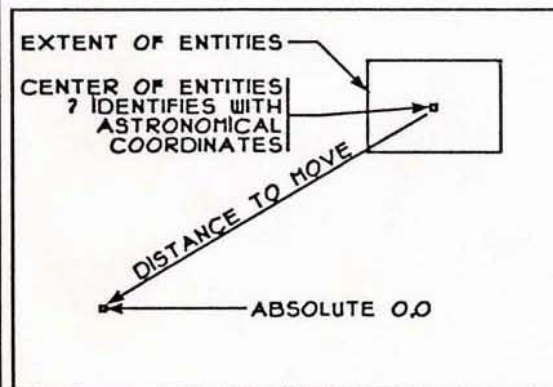
Figures 1 and 2 illustrate:

Figure 1
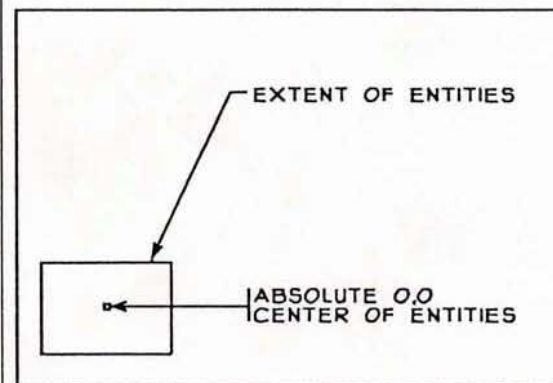*Entities imported at astronomical coordinates*

Figure 2
After being moved to Absolute 0,0 point

3D ENTITY TRANSFER LIMITATIONS

Since DXF is a data transfer mechanism designed around AutoCAD's database, it is as limited as AutoCAD when representing 3D information. These limitations become apparent most frequently when **DataCAD** users are attempting to transfer a 3D model from **DataCAD** to a third-party rendering program or other CAD software. The DXF "bottleneck" becomes most severe when transferring 3D data from **DataCAD** to AutoCAD and back into **DataCAD** again. Once the integrity of the data is lost, there is no practical way to get it back. Below is a list of DXF limitations commonly observed when transferring 3D models.

DataCAD supports slabs and polygons with up to 36 vertices. AutoCAD only supports polygons with 3 or 4 vertices. Currently, any polygon or slab surface that has more than 4 vertices will not transfer from **DataCAD** to AutoCAD. However, DataCAD does output the perimeter of the surface as 3D lines. In order to transfer polygons with more that 4 vertices from **DataCAD** to AutoCAD it would require a complex algorithm that breaks the large polygon into a collection of smaller polygons which would have no associativity. There are conditions that even this would not properly address.

DataCAD supports voids in polygons and slabs but AutoCAD does not. DXF has no concept of what a void is. As a result, voids do not transfer from **DataCAD** to AutoCAD.

DataCAD does not currently support the **AutoCAD Release 10 User Coordinate System (U.C.S.), Entity Coordinate System (E.C.S.) or View Dependent Entities.** If a DXF file originating from AutoCAD contains these types of entity definitions, the resulting entities may have unpredictable orientation and coordinates when read into **DataCAD**.

The method used by DataCAD to define a variety of 3D entities is not fully supported by AutoCAD. These entity types are: 3D arc, dome, torus, cone, truncated cone, cylinder, surface of revolution, and symbols. Consequently, **DataCAD's** DXF translator must explode the 3D entities into polygons when transferring from **DataCAD** to AutoCAD. As a result, the integrity of the data is lost and cannot be recovered.

Additionally, if any entities are rotated about the X or Y axes in **DataCAD** there may be unpredictable results after transferring the drawing into AutoCAD.

EDITOR'S NOTES ON DXF TRANSFER

The following comments are based on my own experience, suggestions from fellow users (notably discussions in the DataCAD Boston Users Group), and a series of articles published in *WindowIn on DataCAD*. My experience with DXF transfer consists primarily of writing out DXF files for use by consultants as base drawings. CAD systems that I have translated to are: AutoCAD and VersaCad®(both PC and Mac versions). I have had little experience importing DXF and my experiences with 3D DXF transfer have been highly unsatisfactory.

At the point of writing out a DXF file to a consultant with whom I have not previously established a set of standards, I proceed as follows:

I provide to the "target" CAD system a trial DXF file. It is based on a simple .DC3 file containing examples of all 2D entity types, text fonts, colors, spacings, etc. that are normally used by our office. Multiple layers (with standard naming conventions) are used. A plot of the drawing file and complete written documentation accompany the DXF file.

By carefully comparing the original .DC3 file with the drawing generated on the "target" system, problems are identified & documented.

With these limitations in mind, I make a copy of the "source" .DC3 file and edit it to use as the DXF base, adhering to some simple rules:

- One Layer=One Color
- Each layer contains discreet information
- Delete all unnecessary information
- Explode all symbols & associative dimensions
- Change all line weight to 1
- Change all (necessary) text to ROMAN2, aspect=1
- Document everything thoroughly

Resulting DXF transfers have been successful; when problems have occurred, they have been easily tracked down and corrected.

SOME THOUGHTS ON SELECTION SETS

An interesting, and underutilized, feature of DataCAD is Selection Sets. It offers to the user a powerful means of entity selection for various editing processes. The following is not meant to be a complete discussion of its use. Rather, it suggests some cautions about its use and illustrates two of its more common utilizations.

A Selection Set is *not* an entity, rather it is a means of *grouping* entities for selection through editing commands. Entities in **DataCAD** have attributes, among which are color, line type, line weight, etc.

Another entity attribute is a "flag" identifying membership in a Selection Set (or multiple Selection Sets). When a Selection Set is Active, its name appears in the Status Window and choosing (F5) Sel Set as the picking method in an editing menu causes the command to operate on entities "flagged" with membership in that Selection Set. The editing operation occurs across layers and affects entities both in layers that are turned on *and in those that are turned off*.

The user might be unaware of the result of an inadvertent "pick" by Sel Set in an editing menu when the command acts on entities in layers that are currently turned off or out of the currently displayed view. For instance, a user attempting to select entities by Area (F3) might accidentally press F5 (Sel Set) resulting in the entities in the Selection Set being acted on unintentionally and *blindly*. For this reason, a few cautions are offered:

- Make a Selection Set active only at the time of its use.

- Clear a Selection Set before adding to it, or...

- Display the contents of a Selection Set before adding to it.

- Toggle Append on only as needed.

The most commonly used feature of the Edit Sets menu is the **Mask/Color** option for adding entities to a Selection Set. When chosen, the user may specify a color, toggle LyrSrch on or off, and select entities by Entity, Group, Area, Fence, or Sel Set for inclusion in the specified Selection Set. The user might, thus, pick by area, with layer search on, all light green entities for inclusion in Sel Set 1. With Sel Set 1

active, the command Move, some specified distance, F5 (Sel Set) moves all light green entities that distance.

The most significant drawback to using such a sequence on a regular basis is that a very long series of menu choices must be selected to activate and clear the desired Selection Set and then add to it with Mask, by color. However, since it is a very useful command option, users may want to write a keyboard macro to automate the process, as follows:

S	EditSet menu
F0	Set Active
F8	Sel Set 8
S0	Exit
F0	Set Active
F1	Sel Set 1
S0	Exit
F4	Clear
F1	Sel Set 1
F5	Yes
F2	Add To
F1	Sel Set 1
F0	Mask
F2	Color
S9	All Off

Note: This sequence assures that even if Sel Set 1 is active at the point of executing the keyboard macro, it is positively toggled to be the active one.

Another possible keyboard macro would clear Sel Set 2, make it active, and toggle Append on:

S	EditSet menu
F4	Clear
F2	Sel Set 2
F5	Yes
F0	Set Active
F8	Sel Set 8
S0	Exit
F0	Set Active
F2	Sel Set 2
S0	Exit
F9	Append
S0	Exit

Assuming that Append was toggled off at the point of executing the keyboard macro, subsequently created entities are appended to the (cleared) Sel Set 2. A series of entities may be created, across layers, and then edited (copied, mirrored, etc.) as a linked set.

The potential of Selection Sets lies in its flexible identification and linkage of entities across layer "boundaries." Therein also lies its potential dangers. We encourage users to explore (with care) the possibilities afforded by Selection Sets and utilize them on a day-to-day basis.

Editor's Note:
Reference Point actively seeks User Input on uses for Selection Sets.

POINTS OF REFERENCE

DataCAD for the Architect
2nd Edition
Carol Buehrens
TAB Books
$26.95

The first edition of *DataCAD for the Architect*, published in 1989, was aimed primarily at the new user of **DataCAD**. It was structured around a series of lessons that the user followed to generate a set of typical drawings for a small residential project. New commands were introduced in each of the lessons, building on those used in earlier exercises.

The second edition maintains the same structure as the first, using the same basic project and lessons. The lessons are expanded and revised to reflect changes to **DataCAD** with the release of version 4.0.

Lessons on the use of the **3D Modeler** and the **AEC Modeler** macro are included for the first time. Where appropriate, lessons are annotated with warnings to users of **DataCAD 128**, indicating areas in which its use differs from the standard program.

Changes in its graphical format and the addition of many new illustrations make it a much more readable book. As in the first edition, an appendix titled "The DataCAD Reference Guide" provides documentation of menu structure, scales, hatch patterns, etc.

The new lesson on the Modeler is very good, providing an introduction to 3D entities, views, and editing commands. Many users have shied away from using the Modeler, because they have not understood how to use it. *DataCAD for the Architect* may facilitate access to this portion of the software for many users.

In a "lesson and exercise" format, it is difficult to discuss thoroughly all of the options available in any given menu; to do so would only confuse the new user. Buehrens is necessarily limited to offering specific methodologies for specific drawing tasks. Advanced users, however, may disagree with a number of the methodologies presented in the book.

Further, as a user's skill level increases, so does the need for more in-depth information. An appendix titled "The DataCAD Operations Guide" outlines steps for accomplishing specific tasks in the format: "if you want to do this...follow these steps." Inclusion of this section (expanded in the 2nd edition) is critical for the advanced user.

"The DataCAD Operations Guide," though, is less than entirely satisfactory. The information that it provides is limited in scope. Those areas covered, as in the basic lessons, tend to be handled in a manner suggesting that *one and only one method* is appropriate to any given drawing task.

Ideally, *DataCAD for the Architect* would be a two volume work: one volume a new user "lesson book," the second an expanded "operations guide." In such a format, Buehrens could provide both an introduction to **DataCAD** for the new user and a more comprehensive discussion of menu/method options for the more advanced user.

Notwithstanding the above reservations, the second edition is a welcome improvement on the original book. It provides to new users a good self-guided tutorial. The lessons also can be integrated successfully into in-house training programs.

Experienced users may be disappointed in the lack of comprehensive technical information and the limited range of methodologies presented. They will, however, find that *DataCAD for the Architect* provides a solid introduction to unfamiliar menu areas and, thereby, may broaden their use of **DataCAD**.

Note:
The DataCAD Maintenance Program coupon book, sent to users in April, includes a $5.00 discount coupon for DataCAD for the Architect.

Reader Poll:
Do you prefer the paper on which this issue is printed or that used for the first issue? Please respond to Frank Simpson or Philip Hart. Thank you.