

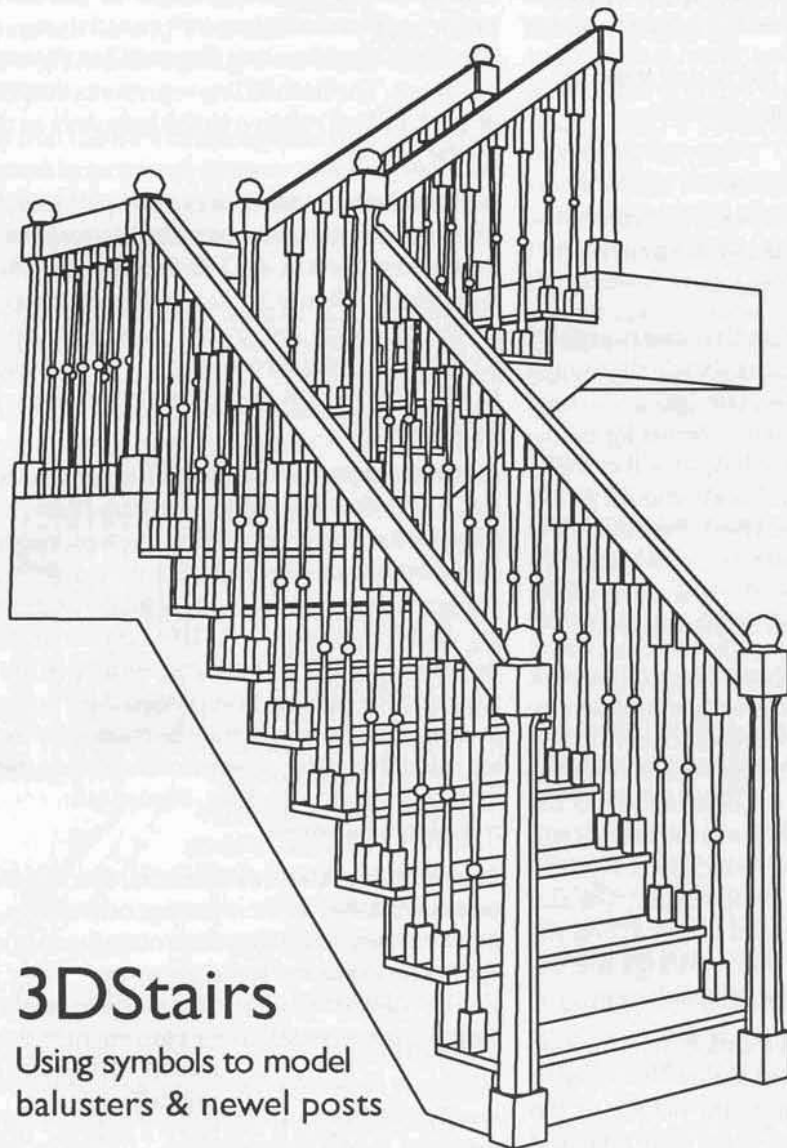


# Reference Point

Volume 2, Number 4

Technical Resources for DataCAD®

Fall, 1992



## 3DStairs

Using symbols to model  
balusters & newel posts

Maintenance customers will have recently received a new DCAL macro from Cadkey, Inc.: 3DSTAIRS.DCX. This macro enables parametric modeling of a variety of complex stair forms and offers to the user a broad range of definable variables. The operation of the macro is relatively straightforward and is supported by thorough documentation. Users should find that, after a bit of reading and some time spent experimenting with it, 3DSTAIRS will provide a powerful enhancement to 3D modeling.

The following is not intended as a comprehensive guide to the use of 3DSTAIRS; rather, it focuses on one specific aspect of its operation: the use of the SYMBOLS option for balusters and newel posts. User-created symbols may be tailored to a specific design and may be much more complex than the simple slabs or cylinders drawn by the macro. Consequently, the level of detail and specificity in the stair model may

### In this issue:

An article on the new 3DSTAIRS macro explores some of its more advanced features and covers some basic points along the way.

**DataCAD Directory and File Structure** discusses the file types utilized by DataCAD and the standard directory structure to which they are installed. Optional directories supported by the PATH NAMES menu in CONFIG are discussed. A method is presented for removing from the DRV directory unnecessary files, *saving about one megabyte of hard disk space*. Basic principles for management of template and symbol files are presented.

**A Configuration Documentation Form** is provided. Used in combination with the Default Drawing Documentation Form, from the Spring, 1992 issue, it makes a good basis for an 'Office Standards Notebook.'

The use of **QEMM-386** and **PC-Kwik**, two excellent third-party products, is discussed. The combination of these provides to the user simplified memory management and high performance RAM disk and disk cache drivers.

**User Input** from Bruce Kaplan continues the discussion of keyboard macros.

Two **batch files** for saving and loading copies of AUTOEXEC.BAT and CONFIG.SYS are provided.

An **Index** is provided to the first seven issues (volumes 1 & 2) of *Reference Point*.

### CONTENTS

3DStairs Macro	1
DataCAD Directory and File Structure	
Standard Directory Structure	4
Expanded Directory Structure	5
Template and Symbol Files	7
Configuration Documentation Form	8
QEMM-386 and PC-Kwik	9
User Input	
Bruce Kaplan on keyboard macros	10
KEEP.BAT and CHANGETO.BAT	10
Index	11
Editor's Note	12

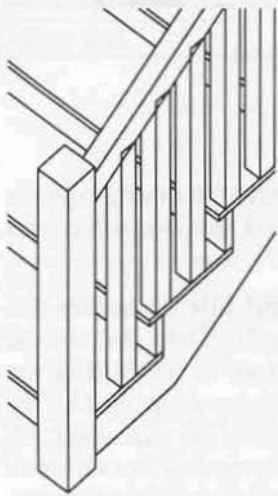


Figure 1

be greatly increased. In the process of examining this feature, some of the other, more basic, features of the macro will also be touched upon. There are four basic steps in using symbols in a stair model:

- Establish settings in the macro and place the stair normally, i.e. without symbols.
- Create the symbols.
- Erase the first stair and replace it with one drawn with the SYMBOLS option turned on.
- Use the REPLACE menu, under TEMPLATES, to place the symbols in the model.

**Establish settings and place the stair normally** before using symbols. The user should first adjust all of the parameters for the stair, leaving turned off the toggle for symbols in the menus for balusters and newel posts. The settings for (normally drawn) balusters and newel posts should reflect their 'gross,' or outline, size. The slabs or cylinders generated by the macro will be used as guides for the creation of the symbols; the size of the newel posts determines the length of the handrails.

The stair should be placed in a layer of its own. Viewing and editing it are greatly simplified by this. The stair should be examined closely at this point to verify that it has been constructed as desired by the user. Any modifications to the menu settings should be made and the stair should be placed again for further verification. Figure 1 illustrates a portion of an L-shaped stair. The size of the newel posts has been set to 4" x 4" and the balusters to 2" x 2". All other settings are the default values, as at the time of installation.

Move the stair to its correct location in the model. The location of the **insertion point** (the point on the stair by which it is initially placed in the drawing) varies depending upon stair type and menu settings. The user should carefully note its location (i.e., front left edge of bottom riser, etc.) and, once the stair has been moved to its proper location, place a snapping point, in another layer, at the insertion point.

**Create symbols for balusters and newel posts.** In the example illustrated, two balusters (two per tread means two different sizes) and two newel posts are used. In the following, creation of the symbols for the balusters is detailed. Newel post symbol(s) creation is a similar process.

Use IDENTIFY to determine the z-minimum and z-maximum of the two baluster slabs on one of the treads. Subtract the minimum from the maxi-

imum to determine the height of the slabs. Use the macro 3DANGLE (available on the Cadkey bulletin board) to determine the angle of the handrail. Knowing that the balusters' overall dimension is 2" x 2", the base drawing illustrated in Figure 2 can be drawn. The dashed line represents the position of the handrail relative to the balusters as drawn by the macro.

Because the balusters in this example have sloped tops and because SURFACE OF REVOLUTION is used for the tapered portion, a 2D elevation of the balusters is developed (figure 3). Slabs at the top and bottom of the balusters are traced in a separate layer (LYRSRCH ON) with z-base set to -1" and z-height set to +1". The dome is drawn as a full dome, with its axis at Z=0.

In the REV SURF menu, two points are selected (at Z=0) to define the *surface boundary* (dashed lines in figure 3). Two points define the *surface center axis* (dot-dash lines in figure 3). For the domes and the surfaces of revolution, both primary and secondary divisions are set to 8. This helps to minimize the total number of surfaces, reducing the time required for hidden line processing. Smoother, more detailed surfaces may be created by increasing this value at the expense of increasing the time for hidden line processing. Figure 4 illustrates the completed balusters.

Before saving them as symbols, the 3D entities must be rotated to their proper orientation. Use the ROTATE menu in 3D to first rotate the entities 90° about the x-axis to 'stand them up.' They must also be rotated 90° about the z-axis to set them to their proper orientation for placement in the stair

model. These balusters are 'directional,' they have a specific orientation relative to the slope of the stair. When they are placed in the model in a later step, the ability of the macro to rotate them as the stair turns will be seen; for this to operate properly, the entities should have the correct orientation when the symbols are created.

Regardless of the orientation of the stair model in which they will eventually be placed, 'direc-

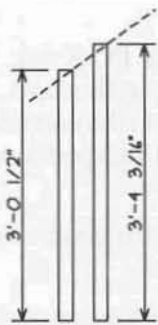


Figure 2



Figure 3

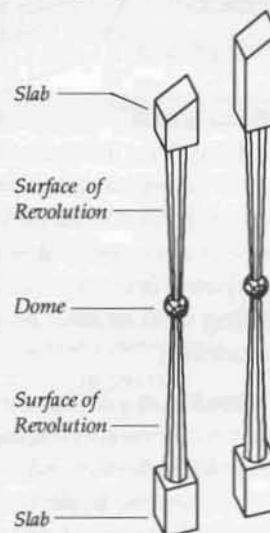


Figure 4

tional' symbols for balusters and newel posts should be created to the following rule: *The orientation of the entities comprising the symbol should be the same as their first instance on a stair placed orthogonally and having its first run rise in the positive Y direction.*

Verify that the base of the entities is at Z=0. They are saved in a normal manner to a template and symbol subdirectory established for their use; the insertion points for them are at the center of the base of each baluster at Z=0.

**Erase the original stair model.** Change the settings in 3DSTAIRS, toggling *on* SYMBOLS in the menus for balusters and newel posts. Select BEGIN from the main menu and place the stair again, using the

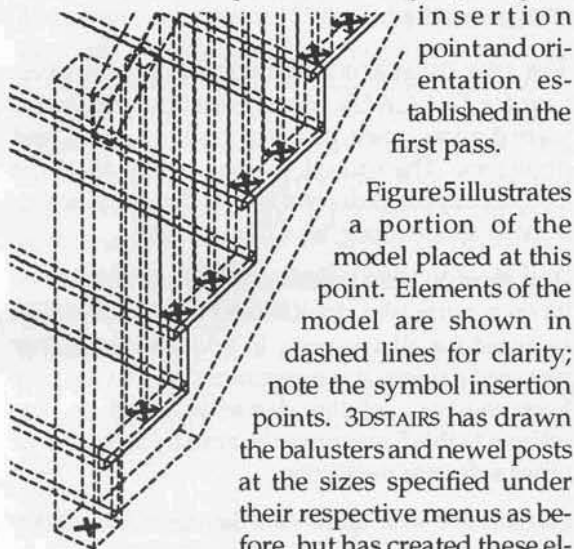


Figure 5

Figure 5 illustrates a portion of the model placed at this point. Elements of the model are shown in dashed lines for clarity; note the symbol insertion points. 3DSTAIRS has drawn the balusters and newel posts at the sizes specified under their respective menus as before, but has created these elements as symbols.

A look at SYMFILES under DIRECTORY shows that four different symbols have been placed in the drawing: STRBAL1, STRBAL2, STRNWL1, and STRNWL2. These are not externally referenced symbols, but ones created by 3DSTAIRS and placed *within* the active drawing. The number of symbols created is a function of the stair type drawn and the options selected within 3DSTAIRS. Were a second stair to be drawn within the same drawing file, the symbol naming convention would continue the numbering sequence (STRBAL3, STRBAL4,...).

If the original placement of the stair had been constructed with SYMBOLS turned on, and later erased, the second placement with SYMBOLS *on* would start the symbol naming at the higher numbers. The possibility for confusion is one of the reasons for recommending that symbols not be turned on until the final placement of the stair.

Use REPLACE, in the TEMPLATE menu to substitute the user-created symbols for balusters and newel posts for those drawn and placed by the macro. This is most easily accomplished by going to a PARALLEL view, and performing a WINDOW IN on the base of one of the symbols to assure proper selection. The 'symbol to replace' is one of those created by the macro. The symbol to 'replace with' is one of the user-created ones displayed in the template. Use the selection method option ALL to replace all instances of the selected symbol with the desired one. Care must be taken to assure proper coordination between macro- and user-created symbols. Figure 6 illustrates a portion of the model after the two baluster symbols have been replaced.

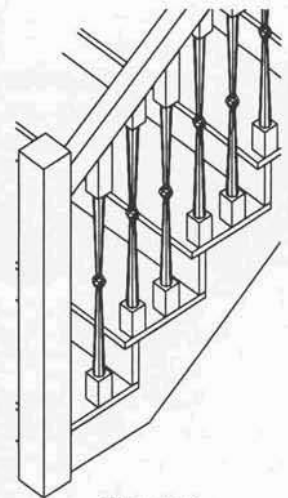


Figure 6

Figure 7 illustrates the balusters at the intermediate landing of the L-Shape stair (handrails and the newel post have been removed for clarity). Note that 3DSTAIRS has correctly rotated the baluster symbols at the turn. Baluster and newel post symbol rotation is controlled by 3DSTAIRS, hence the importance of correct orientation of the symbol elements at the time of symbol creation.

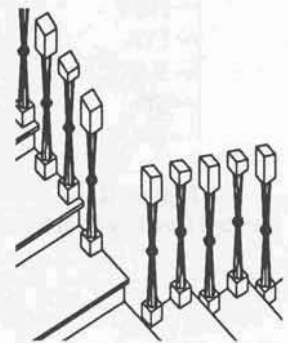


Figure 7

Figure 8 illustrates the baluster symbols on the intermediate landing at the horizontal handrail. Note that the handrail intersects the baluster at the lower edge of the sloped top. The hidden line removal process will hide the 'buried' portion of the baluster but, under some circumstances, the user might desire to have a more accurate baluster symbol for this condition.

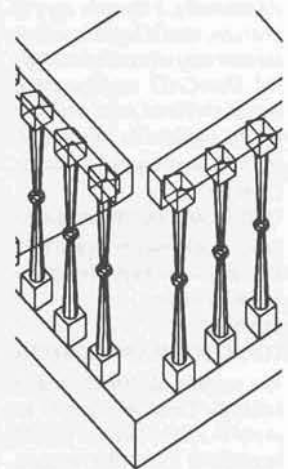


Figure 8

A third symbol, based on the first (shorter) one, could be created. It would be identical to the previous one, but with a flat top (at a z height equal to the bottom edge of the sloped top in the first symbol). In plan view, the user would REPLACE the appropriate symbols, selecting by AREA to insert this, third, baluster type.

The example illustrated in this article is fairly complex. It was chosen as a means of discussing some of the more difficult conditions that might be encountered in the use of the macro. Users should not be daunted by this complexity. For most stair models, the process will be much more straightforward. Models created with 3DSTAIRS, whether processed with hidden line removal and plotted or rendered with Velocity, will yield impressive results. Because the parametric tools are relatively comprehensive, the user may focus his/her efforts on the design of the stair, rather than on the complexity of 3D modeling.



# DataCAD Directory and File Structure

DataCAD v. 4.06 utilizes a large number of files and file types for its operation. The following is presented as a means by which the user may become (more) familiar with the purpose of these files and their organization.

First, the standard DataCAD directory structure, as implemented by the INSTALL routine, will be described. Next, the optional directories enabled by the PATH NAMES menu in CONFIG will be examined; the purpose of each directory and the file types supported by them will be described. Finally, the relationship between template files and symbol files will be discussed.

## Standard Directory Structure

In a standard installation, as initiated by the INSTALL routine, DataCAD utilizes the directory structure illustrated in Figure 1. The following describes the files installed to each directory:

**THE MTEC DIRECTORY** contains the executable file that runs DataCAD (DCAD.EXE) and its four overlay files (DCAD.OV1, DCAD.OV2, DCAD.OV3, and DCMODEL.OVR). It also contains DCAD.CFG, which holds the user-set configuration information and the CONFIG.EXE file used from DOS to enter the configuration menus. RUND CAD.BAT is the batch file created by CONFIG for loading drivers and entering DataCAD.

Two files are placed in the MTEC directory when DCP RINT is installed; DCP RINT.EXE is the executable file for the utility, DCP RINT.CFG is its configuration file. HPGLIII.DAT is an ASCII file; it contains parameters accessed by the driver for the Hewlett Packard LaserJet III driver. The file REVHIST.DOC contains the revision history information for DataCAD. EMMSTAT.EXE is an executable file that is used to check the availability of expanded memory when installing the software display list.

*REVHIST.DOC may be deleted, as may HPGLIII.DAT if the user does not require plotting to the LaserJet III.*

**THE CHR DIRECTORY** contains the font (.CHR) files supplied with DataCAD. Additionally, a number of .SHP files are installed to this directory. These 'shape' files are used in the process of defining fonts. Cadkey includes them as a way to illustrate this process. .SHP files are converted to .CHR files by using SHP2CHR.EXE, which is also installed to this directory.

*Users not interested in exploring this may safely delete the .SHP files and SHP2CHR.EXE.*

**THE DCP DIRECTORY** is created by the installation of DCP RINT, containing data files for the utility. Additionally, the installation of DCP RINT copies the driver UNIPLOT.DVP to the DRV directory.

**THE DCX DIRECTORY** contains the DCAL macros shipped with DataCAD (file extension .DCX) and the help files for the DCAD\_AEC macro (.HLP files). ARROW.DCX is accompanied by ARROW.DCS, the uncompiled DCAL code for the ARROW macro; it is supplied as a means of illustrating the structure of an uncompiled DCAL program.

*The .DCS file may be deleted if the user does not require it. The .HLP files may be deleted at the user's discretion; no help will be available from within the dcad\_aec macro, but its operation will not be effected and some disk space will be saved.*

**THE DRV DIRECTORY** contains the drivers necessary for DataCAD's communication with hardware devices: mice, plotters, graphics cards, and digitizers. The INSTALL routine copies all of the Cadkey supplied drivers to this directory, something over 1 megabyte of files, in total.

Any given system requires only those drivers for its own particular devices and seven other files required for all systems. In addition to his/her required drivers, it is recommended that the user keep the HPGL plotter driver installed as files written to this format may be read by a number of other software packages.

*Considerable disk space may be saved by deleting unnecessary files.*

The following is a suggested means for cleaning up this directory: Assuming that DataCAD is installed in C:\MTEC, the first step is to create a temporary directory and to move all of the files in the MTEC\DRV directory to it. In DOS, type:

```
CD\
MD \MTEC\HOLD
CD \MTEC\DRV
COPY *.* C:\MTEC\HOLD
DEL *.*
```

Next, copy the files necessary for all systems to the MTEC\DRV directory. The user should still be in DOS at the MTEC\DRV directory; type:

```
COPY C:\MTEC\HOLD\CADK*.MTM
COPY C:\MTEC\HOLD\DRIVERS.GD*
COPY C:\MTEC\HOLD\LOGON.DRV
```

Verify that the correct files have been copied to the MTEC\DRV directory by typing: DIR

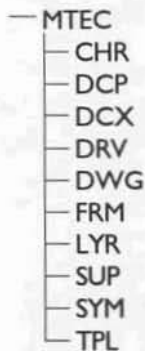


Figure 1

Standard Directory Structure

### General Warning:

*Before deleting or moving any files, it is strongly recommended that the user back-up all program and data files.*

*Additionally, a separate copy of DCAD.CFG should be made so that the user may return to the original DataCAD configuration should problems arise from the process outlined in this article. In DOS, save a copy of DCAD.CFG:*

```
CD \MTEC
COPY DCAD.CFG ORIGINAL.CFG
ORIGINAL.CFG may be copied back to return to the original configuration by typing:
```

```
CD \MTEC
COPY ORIGINAL.CFG DCAD.CFG
```

*The original distribution diskettes for DataCAD should be stored in a safe place and should be available if any files required for DataCAD's proper execution are accidentally deleted.*



The following files should be listed: CADKEY.MTM, CADKEYL.MTM, DRIVERS.GD0, DRIVERS.GD1, DRIVERS.GD2, DRIVERS.GD4, and LOGON.DRV.

Copy the correct graphics driver for the system (MYCARD.EXE, in this example), the mouse driver (MYMOUSE.DVM), plotter driver (MYPLOT.DVP), the software display list driver, the HPGL driver, and the DCPRINT driver (UNIPLLOT.DVP), if installed (the user should still be in the MTEC\DRV directory):

```
COPY C:\MTEC\HOLD\MYCARD.EXE
COPY C:\MTEC\HOLD\MYMOUSE.DVM
COPY C:\MTEC\HOLD\MYPLOT.DVP
COPY C:\MTEC\HOLD\DISPLIST.EXE
COPY C:\MTEC\HOLD\HPGLLDVP
COPY C:\MTEC\HOLD\UNIPLLOT.DVP
```

The proper operation of DataCAD should now be verified. RUNDCAD.BAT (or an alternate batch file used to load drivers and enter DataCAD) should be used. The graphics driver should install, as should the software display list, if used. The mouse should work properly and a test plot should run successfully. If problems occur, enter CONFIG to make sure that the proper device drivers are specified. In DOS, check that the drivers have been copied to MTEC\DRV.

If the graphics driver fails to load, RUNDCAD.BAT may be examined to verify that the correct driver has been copied to the DRV directory. In DOS, type:

```
CD\
CD\MTEC
TYPE RUNDCAD.BAT
```

The contents of the file will be displayed:

```
DRV\VESA256
DRV\DISPLIST
DCAD %I
DRV\VESA256 -R
DRV\DISPLIST -R
```

In this example, the driver that has been set up in CONFIG is VESA256.EXE. VESA256.EXE should appear in a directory listing of MTEC\DRV.

When all is working properly, the contents of MTEC\HOLD may be deleted and the directory may be removed:

```
CD\
CD\MTEC\HOLD
DEL *.*
CD\
RD \MTEC\HOLD
```

The above process yields to the user about 1 megabyte of additional disk space.

THE DWG DIRECTORY is used for drawing (.DC3) files. A new installation will have copied TUTORIAL.DC3 to this directory.

THE FRM DIRECTORY contains the form (.FRM) files used by the REPORTS function of the TEMPLATE menu. Reports generated 'to file' are saved with a .TXT file extension and are saved to the directory specified for text files (see the discussion of optional directories that follows).

THE LYS DIRECTORY is used for saving layer (.LYR) files. This directory is empty in a new installation.

THE SUP DIRECTORY contains support files for the operation of DataCAD. Some of these files may be moved to another directory (see the discussion of expanded directory structure that follows). A new installation contains these files:

DCAD.ANG	Angle values
DCAD.COP	DataCAD copyright screen
DCAD.DEC	Decimal values
DCAD.DIS	Distance values
DCAD.LBL	Menu item labels
DCAD3.LBL	Menu item labels (3D)
DCAD.LIN	Linetype definitions
DCAD.MCR	Keyboard macros
DCAD.MSG	Message line text
DCAD3.MSG	Message line text (3D)
DCAD.PAT	Hatch pattern definitions
DCAD.SCL	Scale values
METRIC.SCL	Metric scale values
DCAD.STL	Text style definitions

User modification of some of these files will be discussed in the section that follows on customizing the DataCAD file structure.

*None of these files should be deleted.*

THE SYM AND TPL DIRECTORIES will be discussed in a separate section later in this article.

## Expanded Directory Structure

Through the PATH NAMES menu in CONFIG (option 4 from the main menu), the user specifies the location of particular file types in DataCAD. In a standard installation, as discussed above, a number of the options under PATH NAMES are left blank.

Users may choose to create additional directories and assign file types to them. Note that the process of specifying a directory in PATH NAMES requires that the directory already exist. The directories must be created in DOS before they may be assigned in PATH NAMES.

See page 8 for a Configuration Documentation Form



## DataCAD Directory and File Structure

- MTEC
- CHR
- **CONFIGS**
- DCP
- DCX
- **DFLTDC3**
- **DFLTS**
- **DL**
- DRV
- DWG
- **DXF**
- FRM
- LXR
- SUP
- SYM
- **SWP**
- **TEMP**
- TPL
- **TXT**

Figure 2  
Expanded Directory Structure

To create a new directory, type:

```
MD \MTEC\DFLTDC3
```

This command should be executed from the root directory of the drive on which DataCAD is installed. DFLTDC3 is used as an example, the desired directory name should be substituted. Once the directory has been created, it may be specified in PATH NAMES. Figure 2 illustrates an expanded directory structure (added directories in bold type). In the following, each of the optional directories will be discussed.

**THE CONFIGS DIRECTORY** is one that is *not* supported by DataCAD through the PATH NAMES menu. It is a directory that is suggested for users who run DataCAD in more than one configuration. In CONFIGS, copies of DCAD.CFG, renamed, may be stored. Users who followed the suggestion for creating a backup copy of DCAD.CFG, outlined in the sidebar on page 4, should copy ORIGINAL.CFG to the CONFIGS directory and delete it from MTEC.

Users may have developed special configurations for running DataCAD under Windows, or for utilizing different display characteristics (resolution or screen colors). Articles in previous issues of *Reference Point* have touched upon this subject and have illustrated the contents of batch files used for switching between configurations.

**THE DFLTDC3 DIRECTORY** is created and specified for default drawings. See the Spring, 1992 issue of *Reference Point* for a discussion of default drawings. It is clearly advantageous to store default drawings to a location separate from project drawing files. If default drawings are currently stored in the same directory as project drawing files, it is *strongly recommended* that the user move them to one containing *only* default drawings.

**THE DFLTS DIRECTORY** is created and specified for default files (scales, angles, and distances), files which are initially installed in the SUP directory. To create the DFLTS directory and move the default files to it, in DOS type:

```
CD\
MD \MTEC\DFLTS
CD\MTEC\SUP
COPY *.ANG C:\MTEC\DFLTS
COPY *.DIS C:\MTEC\DFLTS
COPY *.SCL C:\MTEC\DFLTS
DEL *.ANG
DEL *.DIS
DEL *.SCL
```

After moving the files as described above, the user should enter CONFIG, change the path name for default files to DFLTS, and exit CONFIG, saving the configuration. Enter DataCAD and verify that these files are being accessed properly.

The scale, angle, and distance files may all be modified from within DataCAD through the EDITDEFS menu under SETTINGS. Additionally, edited versions of these files may be saved and loaded to any drawing file. Many users have developed customized versions of these files that they use for particular purposes/drawing types; *saving them to a distinct directory is advisable as a means of good file maintenance.*

Of the files remaining in the SUP directory, the .COP, .LBL and .MSG files are used by DataCAD's own operation and should not be modified by the user. Linetype definitions may be modified by the user with an ASCII editor in DOS, as may the keyboard macro and hatch pattern files. The text style file is written to and read from STYLES under TEXT.

**THE DL DIRECTORY** is used for display list overflow. It is strongly recommended that the software display list *not* be configured to hard disk; rather it should be run in expanded memory. If the size of the display list exceeds the amount of RAM specified for it, though, configuring the overflow to hard disk provides a measure of safety.

**THE DXF DIRECTORY** is created and specified for DXF file transfers. Files written or read by DataCAD should be to and from a specific directory.

**THE SWP DIRECTORY** is shown for users who run systems with limited amounts of RAM and configure DataCAD's path for the virtual, or swap, files to hard disk. For those users, it is a good practice to write the .SWP files to a specific directory. When experiencing anomalous behavior in a drawing file, forcing an exit to DOS without saving the drawing file, .SWP files will be left on the hard disk. If their location is known (SWP directory), they may be deleted, recovering disk space.

**THE TEMP DIRECTORY** is created and specified for temporary files. DataCAD writes temporary files when the user specifies plotting in the background.

**THE TXT DIRECTORY** is created and specified for text output and input. The FILE I/O function of the TEXT menu reads from and writes to this directory. Plots created TOFILE and template reports directed TOFILE are written to this directory. Again, use of this directory is a matter of good file maintenance.



## Template and Symbol Files

The storage of template and symbol files on hard disk requires a special set of considerations for the DataCAD user. Because of the relationship between a template file and the symbols which it accesses, DataCAD provides to the user a complex set of tools for their creation and management. The following discussion focuses on these files from a disk management point of view.

A **TEMPLATE FILE** is an ASCII file that provides to DataCAD an 'index' of the addresses for the symbol files accessed through it. When a template file is loaded to DataCAD, it provides the parameters for the display of the box grid and the symbols displayed within it.

To illustrate, the content of the BEDS template (\MTEC\TPL\FURN\BEDS.TPL), supplied as part of the dcad\_aec macro, is shown below:

```

DATACAD TEMPLATE FILE. VERSION 01.10.
1
4
*
SYM\BEDS\TWIN
SYM\BEDS\FULL
SYM\BEDS\QUEEN
SYM\BEDS\KING

```

The first line is a header included in all template files. The lines containing the numbers 1 and 4 tell DataCAD to display the template as 1 division in the X direction and 4 divisions in the Y direction. The \* is a separator. The four lines of text describe the accessed symbols, including their DOS paths. In this case, the symbols TWIN.SM3, FULL.SM3, QUEEN.SM3, and KING.SM3 are found in the SYM\BEDS directory under the directory from which DataCAD is executed (C:\MTEC, normally).

This is an example of the use of **relative path names** for the symbols. Because DataCAD is operating from C:\MTEC, C:\MTEC\SYM\BEDS, the full, or **absolute path name** is not specified. The distinction between relative and absolute path names is an important one.

Many users find it beneficial to store symbol files in a location other than under the MTEC directory from which they run DataCAD; this facilitates back-up procedures or, in the case of storage to floppy disks, saves on hard disk space. But, it requires the use of absolute path names in template files. When relative path names are used in such cases, subdirectories under C:\MTEC are searched for the symbols and they are not found.

The AUTOPATH function under SAVE SYMBOL in the TEMPLATES menu should be mentioned in this context. When AUTOPATH is on, DataCAD creates a subdirectory immediately under the one specified in CONFIG for symbol files. It gives this subdirectory the same name as the active template file. If the user places all symbol directories at this level, then AUTOPATH is of great assistance in maintaining the directory structure.

If, however, the user wishes to establish a more complex directory structure for symbol files, AUTOPATH should be supplemented by selecting NEW PATH. The user would select SAVE SYM, verify that AUTOPATH is on, and select NEW PATH. Then he/she may navigate through the directory structure and verify the selection of the proper directory (displayed in the message line).

**GOOD FILE MANAGEMENT** of symbols and templates depends upon an orderly approach. It is extremely important that the symbol files accessed by any given template be stored together in a discreet location, i.e. a single directory. The corollary of this is that each symbol file should appear in one, and only one, template.

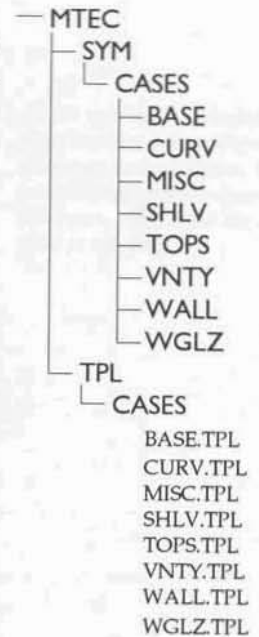
Bill Coppock's Cases Templates product provides a good example. This set of 400+ symbols is controlled by 8 template files, installed in the directory structure illustrated in Figure 3. The template files which access corresponding symbol directories are listed under the \MTEC\TPL\CASES directory. It would be entirely possible to organize these files in another manner. All of the symbol files, for instance, could be located in a single directory and accessed by the 8 templates; but updating the symbols and accessing them would quickly become a nightmare. Additional templates might also be created, organizing the same symbols files in different groupings for the sake of facilitating their access for particular jobs. This sort of cross-listing, though, would lead to another level of confusion.

Orderly management of template and symbol files is extremely important to controlling the operation of DataCAD. Three simple principles can assist the user in avoiding a potentially confusing situation:

*One template file = one symbol directory*

*One symbol file = one template entry*

*Template directory structure should parallel symbol directory structure*



**Figure 3**  
Cases Templates  
Directory Structure



# Configuration Documentation Form

File backed up to: \_\_\_\_\_

Batch file for swapping to: \_\_\_\_\_

## 1. DISPLAY DEVICE OPTIONS

- 1. Display device: \_\_\_\_\_
- 2. Monitor Configuration:  single  dual
- 3. Aspect ratio:  adjusted  not adjusted
- 4. Screen resolution: \_\_\_\_\_
- 5. Text resolution: \_\_\_\_\_
- 6. Menu configuration: \_\_\_\_\_
- 7. Display list configuration:  on  off
  - Amount of EMS left free: \_\_\_\_\_ k
  - Amount of disk space left free: \_\_\_\_\_ k

## 2. MOUSE OR DIGITIZER OPTIONS

- Driver: \_\_\_\_\_
- Port: \_\_\_\_\_

## 3. PLOTTER OPTIONS

- Driver: \_\_\_\_\_
- Port: \_\_\_\_\_

## 4. PATH NAMES

- 1. Drawing files: \_\_\_\_\_
- 2. Template files: \_\_\_\_\_
- 3. Symbol files: \_\_\_\_\_
- 4. Template report forms: \_\_\_\_\_
- 5. Macro files: \_\_\_\_\_
- 6. Character sets: \_\_\_\_\_
- 7. Support files: \_\_\_\_\_
- 8. Driver files: \_\_\_\_\_
- 9. Layer files: \_\_\_\_\_
- 10. Text output files: \_\_\_\_\_
- 11. Temporary files: \_\_\_\_\_
- 12. Virtual files: \_\_\_\_\_
- 13. Default files: \_\_\_\_\_
- 14. DXF files: \_\_\_\_\_
- 15. Default drawings: \_\_\_\_\_
- 16. Display list files: \_\_\_\_\_
- 17. Configuration files: \_\_\_\_\_

5. CHARACTER SET: \_\_\_\_\_

6. DEFAULT DRAWING NAME: \_\_\_\_\_

## 7. SCREEN COLORS

### 1. Menu and status text:

- Border: \_\_\_\_\_
- Function key labels: \_\_\_\_\_
- Function keys: \_\_\_\_\_
- Function key highlight: \_\_\_\_\_
- Current program level: \_\_\_\_\_
- Message: \_\_\_\_\_
- Error: \_\_\_\_\_
- Distance readout: \_\_\_\_\_
- User input: \_\_\_\_\_
- Scale: \_\_\_\_\_
- Directory: \_\_\_\_\_
- Busy light: \_\_\_\_\_

### 2. Palette of colors:

Use default:  1  2  3  4

Lt. Blue R:_____ G:_____ B:_____	Lt. Cyan R:_____ G:_____ B:_____	Lt. Mag. R:_____ G:_____ B:_____	Yellow R:_____ G:_____ B:_____
Lt. Gray R:_____ G:_____ B:_____	Dk. Gray R:_____ G:_____ B:_____	Lt. Red R:_____ G:_____ B:_____	Lt. Green R:_____ G:_____ B:_____
Blue R:_____ G:_____ B:_____	Cyan R:_____ G:_____ B:_____	Magenta R:_____ G:_____ B:_____	Brown R:_____ G:_____ B:_____
Back-ground R:_____ G:_____ B:_____	White R:_____ G:_____ B:_____	Red R:_____ G:_____ B:_____	Green R:_____ G:_____ B:_____





# QEMM-386 & PC-Kwik

## Configuring for DataCAD with third-party software

In the last issue of *Reference Point*, the process of creating an optimized system configuration for DataCAD was explored. Settings in CONFIG.SYS and AUTOEXEC.BAT were established through the use of tools available in MS-DOS 5.0, alone. Loading drivers to high DOS memory proved to be awkward. Limited memory reporting tools in DOS 5.0 required a degree of trial-and-error testing to load all drivers to high DOS memory.

Using a third-party memory manager, the task is simplified. A number of good products are available; Quarterdeck's QEMM-386® is one of the most popular, and is consistently rated highly by the computer press. As well as being simpler to use than the DOS tools, QEMM-386 yields better performance (more Page Frames) in DataCAD. Likewise, both in terms of convenience and performance, Multisoft's PC-Kwik® Power Pack provides advantages over DOS 5.0's RAM disk and disk cache drivers.

An examination of CONFIG.SYS and AUTOEXEC.BAT with QEMM-386 and PC-Kwik installed provides a means for discussing their use. The content of the files was arrived at by first creating 'vanilla' versions of them and then running QEMM-386's OPTIMIZE routine. OPTIMIZE tests drivers and loads them to the high DOS memory area (LOADHI statements), locating each driver to a specific memory address (R:X statements).

OPTIMIZE does not always work perfectly. This configuration required manual editing of the order in which drivers load for OPTIMIZE to load them all high. QEMM-386's memory analysis and reporting tool, MANIFEST, is the best of its type and proved superior to the DOS MEM command.

### CONFIG.SYS

```
DEVICE=C:\QEMM\QEMM386.SYS R:1 RAM ST:M
DOS=HIGH
FILES=30
BUFFERS=10
STACKS=0,0
DEVICE=C:\QEMM\LOADHI.SYS /R:2 . . .
. . .C:\PCKWIK\PCKRAMD.SYS /S:6116
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /P
```

### AUTOEXEC.BAT

```
PATH C:\QEMM;C:\DOS
PROMPT $P$G
C:\QEMM\LOADHI /R:2 C:\STAR\UTIL\VMODE VESA
C:\QEMM\LOADHI /R:2 C:\COMFILES\MOUSE
C:\QEMM\LOADHI /R:1 C:\MTEC\DRV\VESAS256
C:\PCKWIK\SUPERPCK /A+ /T:8 /L:1024
C:\QEMM\LOADHI /R:2 C:\MTEC\DRV\DISPLIST
```

In the first line of CONFIG.SYS, QEMM.SYS is installed; replacing DOS's HIMEM.SYS and EMM386.SYS for management of high DOS memory and extended memory. The RAM statement enables QEMM to access high DOS memory and configures extended memory as expanded memory. ST:M defines the method by which QEMM remaps ROM to RAM. QEMM386.SYS supports DOS 5.0's DOS=HIGH statement. FILES, BUFFERS, STACKS, and SHELL lines remain as they were in the 'DOS only' version.

The line installing PC-Kwik's RAM disk driver, PCKRAMD.SYS, includes the parameter /S:6116. This establishes a *maximum* size for the RAM disk: 6 megabytes. In AUTOEXEC.BAT, PC-Kwik's disk cache driver, SUPERPCK.EXE installs. The /A+ parameter tells the cache to use expanded memory. /T:8 sets track buffering to 8 sectors per track, required for compatibility with QEMM-386. /L:1024 tells the cache to loan up to 1024k of extended memory to any program requesting it (software display list).

Note that in the parameters for both the RAM disk and the disk cache's lending, the memory allocations are stated as limits, not fixed quantities. When DataCAD is run from this configuration, the virtual (.SWP) files are configured to the RAM disk, which *may not exceed* 6 megs. PC-Kwik resizes the RAM disk, keeping it only as large as the data written to it. The software display list, similarly, uses *up to* 1 meg of extended memory. RAM not used by the RAM disk or the software display list is allocated to the disk cache.

In a 'DOS tools only' configuration, the amount of RAM allocated to the RAM disk is fixed, as is that allocated to the disk cache and software display list. The flexible utilization of RAM by PC-Kwik means that *all* available RAM is being put to use.

The performance of PC-Kwik is terrific. At the point of exiting (and saving) one drawing file and entering another, the change is remarkably fast. Unless both drawing files are very large, the save (from the RAM disk) is 'held' by the disk cache until the new .DC3 has been read in. Similarly, AUTOSAVES seem to happen in the background.

QEMM-386 facilitates loading drivers to high DOS memory, freeing conventional memory for DataCAD's execution (39 Page Frames). PC-Kwik dynamically allocates RAM to the RAM disk, disk cache, and software display list, reducing hard disk activity and maximizing RAM utilization. Combined, the two yield impressive performance gains over a 'DOS tools only' configuration.

### Disclaimer:

Cadkey, Inc. does not formally endorse or support either of these products.

Many users, including the editor, have had great success in using QEMM-386 & PC-Kwik and swear by them.

At this writing, PC Magazine plans to review disk caching software in an upcoming issue. Users anticipating the purchase of a third-party disk cache are advised to look for it.

QEMM-386 v.6.02  
Quarterdeck Office Systems  
150 Pico Blvd.  
Santa Monica, CA 90405  
(310) 392-9851

PC-Kwik v.2.13  
Multisoft Corporation  
15100 SW Koll Parkway  
Beaverton, OR 97006  
(800) 234-KWIK

**Do you have a favorite tip or trick?**

Users are encouraged to share with others through these pages some of the techniques that they have developed for efficient use of DataCAD.

Users are also encouraged to write in with questions about problems they are having with DataCAD or areas of its operation that they find difficult to understand.

Drop me a note. My address listed on page 12.

Thanks,  
Philip

**Bruce Kaplan continues to explore the use of keyboard macros. In a posting to the B.S.A. Bulletin Board, he details the following:**

In an effort to speed up the preliminary design phase of a project and yet have usable and accurate information I have always wanted a quick, easy way to insert doors and cut the walls. Our standard is for the door hinge to be 6" from a wall. I have finally settled on an approach that is both quick and flexible. Going back to our old friend the keyboard macro, it appears you can even automate drawing routines. By placing your cursor inside the corner of a room and invoking the following macro, a 3'-0" door (in-swing) will be drawn 6" to the right of the room corner.

```
R^a^f9^s3^'^N^ ^6^$^0$^ ^3$^0$^$^$^;
```

The macro (ALT+R) goes through the following steps: ARCHITECTS menu, DOOR SWING, SINGLE; in response to "enter door hinge point," the macro performs: REFERENCE POINT, OBJECT SNAP, Space Bar, 6" in X direction, 0" in Y direction. In response to "enter door strike," the macro performs: Space Bar, 3' in the X direction, 0" in the Y direction. In response to "enter inside of door swing," the macro executes a pick (\$); in response to the prompt "enter outside of door," the macro also executes a pick.

The advantages: Very, very quick! The doors are drawn dynamically so you can have jambs if you wish. Obviously there are a couple of caveats to this approach: you need to be in RELATIVE CARTESIAN input mode; your cursor needs to be close enough to the corner (but still inside the room) to

get a positive snap; and you will need four different macros to cover up, down, left, and right. (If doors on angles are a frequent occurrence for you just create macros for that, use RELATIVE POLAR input mode!). Of course this macro won't account for every door size in a project, and it wasn't intended to; but you can get a majority of doors in any single project.

The use of David Pendery's EDITKY.DCX (available from Cheapware) will allow the user the flexibility to modify the door offset and size from within DataCAD. I have added these door drawing macros to my preliminary design macro file which also include direct access to the L, T, and X intersection cleanup commands. I use David Pendery's other gem NEWKYMCR.DCX (also available from Cheapware) to switch between keyboard macro files.

*Thanks, Bruce, I can envision a keyboard macro file that would contain lines for placing doors in this manner for a variety of conditions. Separate lines would enable cutting the door with left or right swings in vertical or horizontal walls, for a total of 16 conditions. Additionally, I might add two lines to the keyboard macro file:*

```
A^X^S|^F2^S4^S|^S0^S0
B^X^F3^F7^F8^S4^F0^S0^S0
```

ALT+A resets the OBJECT SNAP settings to END PNT only and sets the MISDIST (range of pixels searched) to a relatively large number, 20 in the example. ALT+B resets to more normal picking (END PNT, MID PNT, PERPEND, and INTSECT) and a MISDIST of 10 pixels. ALT+A would be invoked before cutting doors, ALT+B after the operations were completed.

## Batch files for saving and loading configurations

**Note:**

The PAUSE command is entered in CHANGETO.BAT to avoid a problem resulting from the use of a disk cache that incorporates 'write caching.' Both PC-Kwik and Microsoft's SMARTDRIVE.EXE can cache writes to the hard disk, delaying the actual 'write.'

In the case of CHANGETO.BAT, the system might reboot before AUTOEXEC.BAT and CONFIG.SYS are actually written to the hard disk. PAUSE introduces enough of a delay to assure that the write actually is made to hard disk before the system reboots and, hence, that the new files are in place at boot-up.

Users who run their systems in more than one configuration frequently load different versions of AUTOEXEC.BAT and CONFIG.SYS to run specific applications. Also, while fine tuning parameters, they may need to save versions of AUTOEXEC.BAT and CONFIG.SYS. Two batch files can be of help:

**KEEP.BAT**

```
C:\
CD\
COPY AUTOEXEC.BAT C:\CONFIGS\AUTOEXEC.%I
COPY CONFIG.SYS C:\CONFIGS\CONFIG.%I
```

KEEP.BAT copies AUTOEXEC.BAT and CONFIG.SYS to the C:\CONFIGS directory with a specified file extension. The command 'KEEP CAD' saves them as AUTOEXEC.CAD and CONFIG.CAD.

**CHANGETO.BAT**

```
C:\
CD\
@ECHO OFF
IF NOT EXIST C:\CONFIGS\AUTOEXEC.%I GOTO ERROR
IF NOT EXIST C:\CONFIGS\CONFIG.%I GOTO ERROR
COPY C:\CONFIGS\AUTOEXEC.%I AUTOEXEC.BAT
COPY C:\CONFIGS\CONFIG.%I CONFIG.SYS
PAUSE
REBOOT
:ERROR
ECHO SPECIFIED FILES DO NOT EXIST
ECHO HERE ARE THE ONES THAT ARE ON FILE:
DIR C:\CONFIGS\*.* /W
@ECHO ON
```

Executing the command 'CHANGETO CAD' copies AUTOEXEC.CAD and CONFIG.CAD to the root directory as AUTOEXEC.BAT and CONFIG.SYS; it then reboots the system, using a shareware program, REBOOT.EXE.

**INDEX**

The following is a cumulative index to topics covered in volumes one and two of Reference Point (seven issues). The syntax of the listings is: volume: number, page. The article on ABC Draw, for instance, is found in volume II, number 1, page 2.

<b>ABC Draw</b>	II: i, 2	<b>Default Drawing Form</b>	II: ii, 4
<b>Absolute 0 Problems</b>		<b>Default Drawings</b>	II: ii, 2
DXF file transfer	I: ii, 5	<b>Directory and File Structure</b>	II, iv, 4
trimming	I: i, 8	standard directory structure	II, iv, 4
window\In\extents problem	II: iii, 4	template and symbol files	II, iv, 7
<b>Alias Upfront</b>	II: i, 4	<b>Disk Cache</b>	
<b>Architect's Toolkit One</b>	II: ii, 1; II: iii, 11	configuring	II: iii, 6
		PCKWIK	II, iv, 9
<b>Beginning DataCAD</b>	II: i, 12	<b>DOS 5.0</b>	
<b>Bulletin Boards</b>		configuring for DataCAD	II: iii, 6
B.S.A.	I: i, 8	upgrading to	I: ii, 3
Cadkey	I: i, 8	<b>DOS Configuration</b>	
RADIUS	I: i, 8	Windows 3.x	I: ii, 2
<b>Cases Templates</b>	II: iii, 11	<b>Drawing Sheet Border</b>	I: iii, 4
<b>Cheapware</b>	I: i, 7; II: i, 11	<b>DXF File Transfer</b>	I: ii, 3
<b>Configuration Documentation Form</b>	II, iv, 8	3D entities	I: ii, 6
<b>Configuring for DataCAD</b>	II: i, 6	astronomical coordinates	I: ii, 5
batch files	II, iv, 10	mirrored symbols	I: ii, 4
disk cache	II: i, 7; II, iv, 9	read DXF	I: ii, 5
DOS memory utilization	II: i, 6	text transfer limitations	I: ii, 4
page frames	II: i, 6	to AutoCAD	II: iii, 4
RAM disk	II: i, 7; II, iv, 9	user-defined linetypes	I: ii, 4
with QEMM-386 & PC-Kwick	II: iv, 9	<b>Gleason Partnership</b>	II: iii, 11
<b>Confusing Words in DataCAD</b>	I: i, 6	<b>Graphics Drivers</b>	II, iv, 4
<b>Curved Wall</b>		S3 Graphics Driver	II: iii, 4
3D	II: ii, 9	<b>Hewlett Packard DesignJet</b>	
doors in	I: i, 4	installation	II: i, 3
<b>DataCAD for the Architect</b>		<b>Internal Diagnostic Fault 40</b>	II: iii, 4
2nd ed. reviewed	I: ii, 8	<b>Keyboard Macros</b>	
<b>DCAD.CFG</b>		2d / 3D tip	II: ii, 12
for Windows 3.x	I: iii, 3	customizing	II: i, 9
menu blacking	I: i, 3	cutting doors	II, iv, 10
screen colors	I: i, 3	toggling sel sets	II: i, 11
<b>DCAD.MCR.</b>	See Keyboard Macros	<b>Line Types</b>	
<b>DCAL Macros</b>		multi-weight dashed arcs	I: iii, 3
3Dangle	II: ii, 12	<b>Logitech Mouseman</b>	
3DStairs	II: iii, 3; II, iv, 1	installation	I: i, 3; II: i, 3
AEC_MODEL		<b>Mannequin</b>	II: i, 2
Default Settings	II: i, 3	<b>Memory Management</b>	
Dhatch	II: iii, 12	QEMM-386	II, iv, 9
Editkey	II: i, 11; II, iv, 10	<b>Neo Graphix</b>	II: iii, 11
Layout 40	I: iii, 7	<b>Object Snap</b>	
LyrUtil		sel set toggle	I: iii, 3
Input Mode Problem	II: i, 3	<b>Organic CADD</b>	II: iii, 11
Newkymcr	II: i, 11; II, iv, 10		
Plot 40	I: iii, 7		
Roof Builder 2	II: iii, 12		
Roofer	II: iii, 12		
Roofit	II: iii, 2		
Setpens	II: iii, 5		
Touchup	II: iii, 11		



Index continued from page 11

<b>P</b> age Frames			
maximizing	II: iii, 6; II: iv, 9	Selection Sets	I: ii, 7
<b>PC-Kwik</b>	II, iv, 9	mask	I: ii, 7
<b>Plot</b>		overview	I: iii, 3
drawing sheet border	I: iii, 4	sel set 8	II: i, 1
layout	I: iii, 4	<b>Soft-Tech</b>	II: i, 6
multiple scale	I: iii, 4	<b>Software Display List</b>	I: iii, 6
software display list	I: iii, 6	plot layout	II: iv, 7
<b>Publications</b>		<b>Symbols</b>	II, iv, 7
<i>Beginning DataCAD</i>	II: i, 12	and template directories	II: iv, 7
<i>Cadkey Applications Guide</i>	I: i, 7; II: iii, 12	piping	II: iii, 5
<i>Cheap Tricks</i>	I: i, 7	<b>T</b> raining Centers	I: iii, 8; II: i, 12
<i>DataCAD for the Architect</i>	I: i, 7	listing	II: ii, 11
<i>WindowIn on DataCAD</i>	I: i, 7	<b>U</b> ser Groups	II: i, 3
<b>Q</b> EMM-386	II, iv, 9	<b>V</b> elocity Tips	II: ii, 10
<b>R</b> AM Disk		<b>V</b> esa Drivers	
configuring	II: iii, 6	<b>W</b> indows 3.x	I: ii, 2
PCKWIK	II, iv, 9	installation	I: ii, 2
<b>Reference Point</b>		overview	I: iii, 2
distribution	II: ii, 1	running DataCAD under	II: i, 8
editorial policy	I: ii, 1	sources of information	I: ii, 2
<b>RenderMan Interface</b>	II: ii, 1; II: iii, 5	System Edit	II: i, 8
<b>S</b> 3 Graphics Driver	II: iii, 4	<i>Windows 3 Secrets</i>	II: i, 8

**Editor's note:** With this issue, the second volume of Reference Point is completed. Writing the newsletter has been an interesting experience for me. Feedback from DataCAD users has been generally favorable, but the amount of it has been disappointingly small. I would like Reference Point to be more responsive to the needs of its readers, but this requires a more active level of communication than has occurred over the last year and a half. To this end, I would appreciate some feedback from you, the reader.

- What topic areas you are interested in seeing covered in future issues? 2D? 3D?
- Would you like to see coverage of menu 'basics' or more advanced topics?
- Do you want to see articles on DataCAD-specific third-party products (i.e. DCAL macros, etc.)?
- Are you interested in discussions/reviews of hardware products and their use with DataCAD?
- What is the most spectacular use to which you have put DataCAD? Most cost efficient? Most satisfying?
- Are you willing to share with other users examples of your work and/or techniques through these pages?

These few questions are meant just as a means of stimulating your thoughts on the contents of Reference Point. A quick note from you outlining any thoughts that you might have will be very much appreciated. Thanks, Philip

## PUBLICATION INFORMATION

**Reference Point**  
is published quarterly by:

**Cadkey, Inc.**  
4 Griffin Road North  
Windsor, CT 06095

**Managing Editor:**  
**Editor:**

Frank Simpson  
Philip Hart

Additional copies, changes in mailing address, and other business communications: Tel: (203) 298-8888; Fax: (203) 298-6401

**Editorial Review Committee:**

Lou Bodnar  
Clay Rogers  
Eric Smith  
Mark White

**Editorial Communication:**

**Philip Hart**  
9 High Street  
Brunswick, ME 04011  
(207) 729-5466

© Copyright 1992, by Cadkey, Inc. All rights reserved.

## CADKEY CONTACT

<b>Cadkey:</b>	(203) 298-8888
<b>Lou Bodnar:</b>	ext. 6425
<i>A/E/C Marketing</i>	
<b>Mark Hyjek:</b>	ext. 7157
<i>Technical Support</i>	
<b>Clay Rogers:</b>	ext. 7100
<i>Quality Assurance</i>	
<b>Frank Simpson:</b>	ext. 6443
<i>Marketing Publications</i>	
<b>Mark White:</b>	ext. 6455
<i>A/E/C Product Management</i>	
<b>Tech Support FAX:</b>	(203) 298-6404
<b>Cadkey Bulletin Board:</b>	(203) 298-6405
(8 bits, no parity, 1 stop bit)	

**Address correction:**

The listing in the last issue of Reference Point for Computerland in Lancaster, PA as a DataCAD training center was incorrect. It should have been:

The Business Advisory Group, Inc.  
217 Granite Run Drive  
Lancaster, PA 17601  
Terry Bergen  
(717) 560-9600 ext.101

Note that this is also the contact information for the DETAIL Users Group.