

Philip A. McDonald<sup>†</sup>, Paula T. McCaslin, Edward J. Szoke<sup>\*</sup> and U. Herbert Grote  
NOAA Research-Forecast Systems Laboratory, Boulder, Colorado

<sup>\*</sup>[In collaboration with the Cooperative Institute for Research in the Atmosphere (CIRA),  
Colorado State University, Fort Collins, Colorado]

## 1. INTRODUCTION

For several years, the NOAA Forecast Systems Laboratory (FSL) has been developing an application for the 3-D display of meteorological model data (McCaslin et al. 1999 and 2000 and Szoke et al. 2001). The result of this work has been Display-Three Dimensions (D3D), a 3-D display counterpart to the Advanced Weather Interactive Processing System (AWIPS) 2-D display application Display-Two Dimensions (D2D). Although the core of D3D is Vis5D, a general purpose 3-D meteorological display application developed at the University of Wisconsin (Hibbard and Santek 1991), the intent has always been that D3D would be used in an AWIPS environment, or at least with an AWIPS data source.

Until very recently, the computational and graphics requirements of a 3-D display application exceeded the capabilities of typical AWIPS workstations. However, with the increasing performance of Linux personal computer (PC) workstations and their decreasing cost, coupled with a movement toward Linux AWIPS systems, the day of D3D in the forecast office is at hand.

Since D3D is a new and relatively unknown application, it is understandable that potential D3D users, as well as system administrators, may have concerns about the ramifications of running D3D on their systems. This discussion will address these concerns, without getting into the detail that one would find in an installation manual.

## 2. SYSTEM REQUIREMENTS

D3D development began on Hewlett-Packard systems running the HP-UX 10.x operating system. However, since early 2000, all D3D development has targeted Intel-based Linux PCs.

### 2.1 Platform

An Intel-based PC system with a central processor unit (CPU) speed of at least 500 MHz is recommended. Three-dimensional displays are computationally intensive, so users are advised to acquire the fastest system available. Since D3D makes no explicit use of multiple processors, these are optional. However, the operating system may take advantage of multiple processors and this may, in turn, improve D3D's performance.

### 2.2 Memory

Although D3D has been implemented on systems with as little as 128 MB of memory, 256 MB is the recommended minimum. Too little memory can drastically degrade D3D's performance, while, on the other hand, there is no such thing as too much memory.

### 2.3 Graphics Adapter

Hardware rendering, which utilizes the capabilities of a graphics adapter to their fullest extent, may require special device drivers, X servers, or even modified versions of the operating system. Although the accelerated performance of hardware rendering is desirable, these complications make it difficult for FSL to evaluate, implement, document, and support it. Consequently, at this time, D3D utilizes software rendering, that relies on the CPU to perform all of the rendering tasks. Software rendering is generally noticeably slower than hardware rendering. However, in this era of ever-increasing CPU speeds, the slowness of software rendering becomes less noticeable. Since D3D is not tied to any specific graphics adapter or adapter type, D3D works equally well with practically any graphics adapter.

### 2.4 Operating System

Currently, Red Hat Linux versions 6.2 through 7.1 are supported. Although the intent is to support as many versions of Red Hat Linux as possible, the D3D development staff cannot support them all forever. Consequently, as newer versions of Red Hat Linux are released, older versions may lose D3D support.

### 2.5 Other Software

Although D3D is distributed in as complete a form as possible, it does require some additional software to be available on the workstation. Most notable of these are the Tool Command Language/Tool Kit (Tcl/tk) and the C and C++ standard libraries. However, since these are generally distributed and installed with Linux, requiring their presence poses no additional burden on the workstation.

### 2.6 Data

D3D does not *require* D2D. However, D3D does need an AWIPS data tree and localization data for access to the model grids it displays. If D3D is running on a workstation that is networked to an AWIPS data server, these data can be made available to D3D by simply cross-mounting the data directories on the server to

---

<sup>†</sup> Corresponding author address: Phil McDonald, 325 Broadway R/FS4, Boulder, CO 80305-3328. Email: mcdonald@fsl.noaa.gov

the workstation. If the D3D workstation has no access to an AWIPS data server, then some other arrangement must be made to provide the data. Usually this means copying the desired AWIPS data and localization data to the D3D workstation.

### 3. SYSTEM RESOURCE USAGE

D3D has been designed to be as unobtrusive as possible, but it will, of course, use system resources.

#### 3.1 Processor

Since, on single processor systems, all applications have to share the same CPU, adding one more application (D3D in this case) will impact the performance of all the others. This may not be the case on a system with multiple processors if D3D can be assigned to a processor that is not in use. However, such an assignment would be a capability of the operating system, not D3D.

#### 3.2 Memory

The default memory limit that D3D will use is 200 MB. This is only the ceiling on the amount of memory that D3D will try to allocate, not the amount that it will try to allocate outright. If the amount of system's free memory is less than 200 MB, then D3D's limit will be the amount of system's free memory. Although a default limit of 200 MB has been found to be a comfortable one, it may be modified by users if a larger amount seems appropriate.

If the amount of memory available to D3D is less than D3D needs, D3D will attempt to free up memory by destroying old or unused graphical objects. However, if these objects are needed again, D3D will need to regenerate them. This process of generating, destroying, and then regenerating objects can have a severe impact on D3D's performance. Therefore, it is advisable to allocate an ample amount of memory.

#### 3.3 Disk

The D3D application and all of its ancillary data occupy about 20 MB of disk space. D3D documentation, including an online user's guide, occupy about 60 MB. In addition, D3D also needs access to the AWIPS localization data. If these data are not available, a minimal subset (consisting primarily of a grid source table, topographic data and map data) must be installed. This subset occupies about 20 MB.

During their use of D3D, users may save images and procedural bundles. These files will, of course, occupy disk space. If activity logging is enabled, log files will be created, taking up yet more disk space. Although it is difficult to precisely quantify the amount of disk space used by these types of files, several megabytes should be allocated for these purposes.

#### 3.4 Network

Only when D3D is networked to a remote data

server does D3D engage in any network activity, and then only to acquire the data requested by the user. D3D requires no notifications of any kind, nor does it broadcast messages of any kind. Although D3D user files, such as images and procedural bundles, may be written to the D3D workstation's disks, no files are written to the data server.

### 4. INSTALLATION AND CONFIGURATION

Initially, D3D should be installed from the distribution compact disk (CD). Requests for this CD can be made through the D3D release web page (<http://d3d.fsl.noaa.gov/release>). This CD contains everything necessary to assess D3D: the application, documentation, AWIPS localization data, and case data (from 3-4 May 1999). Running D3D from this CD is a simple and unobtrusive way for users to assess D3D and to become familiar with its use. However, using D3D in this way does not allow it to be used with real-time data or with additional data cases. For these capabilities, D3D needs to be installed on the workstation.

#### 4.1 Installation

The installation of D3D is accomplished by simply copying D3D from the distribution CD to the desired directory on the workstation. It is not necessary to copy the case data or the AWIPS localization data.

Once D3D has been installed, upgrades may be obtained through the D3D release web page.

#### 4.2 Configuration

After installation, the D3D main directory will contain a localization directory. The localization directory contains a file, **d3denv**, which specifies the values of the environment variables that D3D uses. To configure D3D, it is necessary to edit this file and provide appropriate values for the path of the AWIPS localization data (FXA\_HOME), the path of the real-time AWIPS data (FXA\_DATA), and the AWIPS local and ingest sites (FXA\_LOCAL\_SITE and FXA\_INGEST\_SITE). If D3D will be used with case data, the path of the case data also needs to be specified (FXA\_CASE).

### 5. CASE DATA

D3D users may find it useful to archive 3-D model grids and then later select them for use as training aids or for review purposes.

#### 5.1 Archiving

To archive a data case, the user need only copy the desired to the case data directory, usually on the D3D workstation. However, it is important to save the entire path of the dataset and not just the dataset itself. The Linux *tar* command is particularly well suited to this purpose.

For example, imagine that the date is 4 May 1999

and a user wants to archive the 0000 UTC Eta run from that date. First, the desired file and its path are saved in a tar file with a command similar to the following:

```
tar -cvf case.tar /data/fxa/Grid/SBN/netcdf/CONU
S211/Eta/19990504_0000
```

Next, a directory for the case data needs to be created:

```
mkdir /case/ok_tornadoes_04May1999
```

Finally, the current directory is changed to the case directory and the case data is untarred:

```
cd /case/ok_tornadoes_04May1999
tar -xvf case.tar
```

It should be noted that the AWIPS gridded datasets can be rather large (~50 MB or more). While being archived, a dataset will be traveling across the network and may impact the network performance. Also, the archived datasets will consume space on the D3D workstation.

## 5.2 Selection

If D3D has been configured for use with case data, a case data selector, similar to the one in Figure 1, will appear when D3D is started. This selector allows the user to set the data source to real-time data or any of the cases archived in the case data directory.

If D3D has been configured for use with real-time data only, the case data selector will not appear.

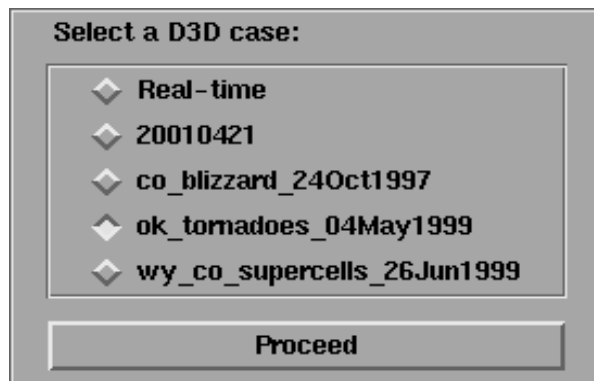


Figure 1. D3D case data selector, shown at D3D startup. The user has selected *ok\_tornadoes\_04May1999*

## 6. ADDING A MODEL

When considering adding a local, or any other, model to those that D3D will display, it is extremely important to bear in mind that D3D reads data from the AWIPS data tree and relies on the AWIPS localization data for information about the data in that tree. Therefore, it is imperative that the new model be added to the AWIPS data tree and localization data in accordance with AWIPS procedures set forth in the *AWIPS System*

*Manager's Manual*.

Once this has been accomplished, D3D can be modified to recognize the new model. This procedure is not difficult, but it involves editing several Tcl/tk and ancillary files. Most users should be able to accomplish this task without much difficulty. For a detailed description of this procedure, users should consult the D3D documentation found on the distribution CD. First, the D3D Volume Browser menu files need to be edited to associate the new model with an appropriate display scale. Next, a model configuration file for the new model will need to be placed in the D3D localization data directory. A D3D model configuration file contains descriptive information such as the model name, dimensions, vertical levels, available fields, and domain mapping parameters. Generally, the model configuration file for the new model can be copied from that of another model with similar properties and then modified appropriately.

## 7. SUMMARY AND CONCLUSIONS

D3D is an evolving technology. Throughout its development two things have been, and will continue to be, borne in mind. First, the application must provide a service to the user. To meet this end D3D needs to be not only useful and easy to use, but it must continue to adapt to users' needs, either through the users' means or through those of the D3D developers. Second, D3D must in no significant way impede or interfere with any aspect of any system on which it is installed. This must be especially true when D3D is installed in a forecast office AWIPS environment. The D3D development team believes that D3D succeeds on both counts.

## 8. ACKNOWLEDGMENTS

Grateful thanks are extended to Nita Fullerton and Patrice Kucera of FSL for their reviews of this paper.

## 9. REFERENCES

- Hibbard, W. R. and D. Santek, 1991: The Vis5D system for easy interactive visualization. *Seventh International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, New Orleans, LA, Amer. Meteor. Soc., 129-134.
- McCaslin, P. T., P. A. McDonald, and E. J. Szoke, 1999: Developing and testing a 3D visualization workstation application at FSL. *15th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, Dallas, TX, Amer. Meteor. Soc., 498-501.
- McCaslin, P. T., P. A. McDonald, and E. J. Szoke, 2000: 3D Visualization development at NOAA Forecast Systems Laboratory. *Computer Graphics*, February 2000, Vol. 34 No. 1, 41-44.
- Szoke, E. J., U. H. Grote, P. C. Kucera, P. T. McCaslin, P. A. McDonald, and W. F. Roberts, 2001: D3D: A potential 3D visualization tool for the National Weather Service. *17th International Conference on Interactive*

*Information and Processing Systems for Meteorology,  
Oceanography, and Hydrology*, Albuquerque, NM, Amer.  
Meteor. Soc., 10-14.