

Release Notice:
DG/UX™ AView for AViiON®
Systems

Release 1.11 Revision 0.00

Part number 085-600305-00

This release notice applies to the following models:

P001
P002
H001

Copyright © Data General Corporation, 1989, 1990, 1991
Unpublished—all rights reserved under the copyright laws of the United States
Printed in the United States of America
August 1991
Licensed material—property of Data General Corporation

Restrictions and Trademarks

This software is made available solely pursuant to the terms of a DGC license agreement which governs its use.

Restricted Rights Legend: Use, duplications, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at [DFARS] 252.227-7013 (October 1988).

DATA GENERAL CORPORATION
4400 Computer Drive
Westboro, Massachusetts 01580

AViiON is a U.S. registered trademark of Data General Corporation.

DG/UX is a trademark of Data General Corporation.

UNIX is a U.S. registered trademark of American Telephone and Telegraph Company.

X Window System is a trademark of Massachusetts Institute of Technology.

Release Notice:

DG/UX™ AView for AViiON® Systems

Release 1.11

085-600305-00

August 1991

Contents

1	Introduction	1
2	Product Description	1
3	Environment	1
3.1	Hardware	1
3.2	Software	1
3.2.1	Operating Environment	2
3.2.2	Compiling Environment	2
4	Enhancements and Changes	2
5	Notes and Warnings	3
5.1	Notes	3
5.2	Warnings	4
6	Documentation	5
6.1	Manuals	5
6.2	Documentation-change files	5
7	Software Distribution	5
7.1	Media	5
7.2	Files	5
8	Installation Instructions	6
8.1	Installing The AView Graphics Library	6
8.2	Testing the Software Installation	6
9	Preparing a Software Trouble Report (STR)	7
9.1	Software Problems	8

1 Introduction

This Release Notice describes the DG/UX™ AView Graphics Library product and its installation. It also includes information not currently available in the product manual, such as the product directory tree.

Between releases, Data General may issue updates to the product. An update is essentially a partial release. This mechanism reduces the time needed to fix problems by providing a level of correction short of releasing the complete product. Each update of a product supersedes the previous update.

Additional copies of this release notice can be printed. Use the file `/usr/opt/aview/release/aview_1.11.rn` on the release tape. In the event of a difference between the on-line file and the hardcopy version of the release notice, the hardcopy notice takes precedence.

2 Product Description

AView is a full function 2- and 3-D graphics library. Its routines provide high-speed, direct access to the AViiON® graphics workstation hardware. It includes basic graphics primitives (line, circle, polygon), input control, color table control, double buffering, and Z-buffering.

3 Environment

3.1 Hardware

The AView Graphics Library runs on all AViiON workstations. It supports monochrome, 8-bit color, 24-bit color, and hardware Z-buffer options.

On all AV/400 and AV/4000 series workstations, prom revision 5.09 or greater is required.

Note: If you intend to use software Z-buffering in your applications, we recommend a minimum of 16MB of memory be installed on your system. This will reduce performance degradation due to swapping.

3.2 Software

3.2.1 Operating Environment

Revision 5.4 or later of the DG/UX operating system is a prerequisite. Consult the DG/UX release notice (085-600265) to determine its environmental requirements. Due to object file format changes, applications built with this release of the AView Graphics Library can only be run on DG/UX 5.4 or later.

3.2.2 Compiling Environment

DG/UX 5.4 is a multiple software development environment platform. The only DG/UX software development environment supported by the AView Graphics Library is m88kdguxelf (the default environment). For more information on software development environments, see the *Programmer's Reference for the DG/UX™ System*. When compiling applications that use the AView Graphics Library, the SDE_TARGET_ENVIRONMENT must be m88kdguxelf. Also, when using gcc, strict ansi and traditional mode are not supported. Use "gcc -DDGUX_SRC -ansi" or "gcc" without the -ansi or -traditional switches to compile your applications.

4 Enhancements and Changes

1. Support for monochrome has been added in this release.
2. Support for the 24-bit + 4-bit overlay graphics card has been added in this release.
3. An application would sometimes core dump when using gxFontDef. This has been fixed in this release.
4. When automatically starting the graphics daemon, an application could get out of sync with the graphics daemon, causing numerous problems. This has been fixed in this release. NOTE: This bug fix creates incompatibilities with previous releases. Specifically, applications built with this release of the library will hang when run with previously released versions of the grfxd.
5. An application would core dump when using gxMatrixConcat. This has been fixed in this release.
6. A drag-to-open window could be made larger than the XMaxSize and YMaxSize specified in the gxtWindow structure. This has been fixed in this release.
7. User defined line patterns are now displayed with the most significant bits used first. See the man page for gxLinePatternDef for more information.
8. A filled primitive would sometimes be filled using the default line pattern, rather than the default fill pattern. This has been fixed in this release.

5 Notes and Warnings

5.1 Notes

1. You do not have to start the graphics daemon manually. The library first checks to see if it is already running, and if not, looks for the `grfxd` executable in `/usr/bin` and `/usr/opt/aview/bin`. If the executable is found, the graphics daemon is started automatically.
2. If your application exits with an AView fatal error "Unable to create synchronization semaphore", then the graphics daemon must be started manually. Start the `grfxd` executable found in either `/usr/bin` or `/usr/opt/aview/bin`, then restart your application.
3. The library uses single precision floating point. The Makefile in the `aview/demo` directory uses the `gnu c` compiler (`gcc`) and illustrates the proper way to build applications. When using the `greenhills c` compiler (`ghcc`), the `-X167` option must be used to insure single precision floating point numbers are passed into the subroutines.
4. Previous revisions of the AView Graphics Library did not mask off the unused lower order bits of a 32 bit word when a line pattern shorter than 32 bits was defined. As a result, line patterns that were defined in the lower order bits of a 32 bit word would work contrary to the documentation. This has been fixed in this release and may require users to check line pattern definitions that are less than 32 bits and ensure that such definitions are placed into the higher order bits of a 32 bit word.
5. If your application is in `ZBufferMode` `gxcHardZ` or `gxcNotHidden`, has set a line pattern, and has not set a fill pattern, then primitives will be rendered using the line pattern, not the default fill pattern.
6. Resetting window parameters (by calling `gxWindowSet` with a non-null window pointer) will reset the color map, cursor, cursor display status (on or off), and display mask to the defaults.
7. Concave polygons are not fully supported. Some types will work but not all types. Use at your own risk.
8. The routine `gxWindowCreate` does not support the aspect ratio parameter of the `gxtWindow` structure.
9. The only devices currently supported by the `gxInputAccelerationSet` routine are:
`gxcMouse`
`gxcMouseX`
`gxcMouseY`
 Setting acceleration for any one of these devices sets it for all of them.
10. The routine `gxInputAccelerationGet` always returns `gxcMouse` as the value for the `Device` parameter.

11. The routine `gxCursorDef` does not yet support cross-hair cursors.
12. Software Z-buffering uses a large amount of memory and will be very slow on systems with less than 16MB of memory.
13. Error messages are directed to `stderr`.
14. Overlapping windows are not supported. If windows overlap, drawing from both windows will occur in the overlapped area.
15. A maximum of ten windows per AView graphics process can be opened.
16. The minimum window size is 20x20.
17. The maximum window size is 1280x1024.
18. On an 8-bit machine, `gxTri` color mode can not utilize the hardware Z-buffer. The software Z-buffer will be used even if hardware Z-buffer is requested.
19. `gxArc`, `gxEllipse`, and `gxCircle` will all immediately return with no action taken if the radius parameter is less than or equal to 0.
20. The only boolean that works correctly with software Z-buffering is `gxCopy`.

5.2 Warnings

1. The `grfxd` and `libaview.a` create files in `/tmp` that are used in the transfer of information between the `grfxd` and an AView application. If these files (`/tmp/.gxinptd` and `/tmp/.gx[process pid #]`) are deleted while the `grfxd` and the AView application are running, then input device service will hang.
2. You can not use applications built with this release of the library with earlier releases of the graphics daemon (`grfxd` executable).
3. The current version of AView does not run with X windows. Be sure to kill the X server before running any AView applications.
4. Calling `gxViewSet` to create a view other than the default view corrupts the text drawing position and requires the use of explicit calls to `gxTextPositionSet`, rather than relying on the default text drawing position.
5. Keyboard input may be lost while the bell is ringing.
6. When creating a window that has portions of the window located offscreen, the viewport for that window is resized to the onscreen region.

6 Documentation

6.1 Manuals

The following manuals are available for the AView Graphics Library product:

Publication	Part Number
AView Graphics Library Release Notice	085-600305-00
Programmer's Reference for the AView Graphics Library	093-000717-00

The Programmer's Reference for the AView Graphics Library is available in the `/usr/opt/aview/catman/A_man` directory and consists of several hundred man pages.

6.2 Documentation-change files

There are no documentation-change files associated with this release.

7 Software Distribution

7.1 Media

AView is included with model numbers P001, P002, and H001. See the media notices for these models for a description of the media.

7.2 Files

This table briefly describes the organization of the product release tree for the AView Graphics Library.

AView Graphics Library Directory Structure	
<code>/usr/opt/aview</code>	top level directory
<code>/usr/opt/aview/bin</code>	AView executable files
<code> /grfxd</code>	graphics daemon
<code>/usr/opt/aview/include</code>	AView include directory
<code> /aview.h</code>	AView include file
<code>/usr/opt/aview/release</code>	AView release notice
<code>/usr/opt/aview/sbin</code>	AView sysadm setup scripts
<code>/usr/opt/aview/lib</code>	AView library directory
<code> /libaview.a</code>	AView library
<code>/usr/opt/aview/catman/A_man</code>	AView on-line documentation
<code> /man3</code>	graphics library docs

<code>/usr/opt/aview/tutorial</code>	tutorial code
<code>/usr/opt/aview/demo</code>	demos and example makefile directory

8 Installation Instructions

8.1 Installing The AView Graphics Library

AView currently requires 10000 blocks of file space to load. Make sure you have adequate space available prior to installing this package. There are a large number of demo programs provided in source form only. Enough space is available to compile a few of these at a time. However, if you wish to compile all of the demos at once, you will need an additional 40000 blocks of space. Since there is little need to have all the executables at one time, we do not recommend that you allocate this extra space.

This package will load into `/usr/opt`. We recommend that a separate logical disk be created so that the release does not load directly into `/usr`. If `/usr/opt/aview` has not been created on your system, refer to *Installing the DG/UX™ System (093-701052)* for creating a logical disk, and adding the file system. Once the file system has been setup and mounted, load the AView Graphics Library with *sysadm*. Once the package has finished loading, use *sysadm* to install AView. Following package setup, the installation of the AView Graphics Library is complete.

8.2 Testing the Software Installation

Currently, AView does not run within X windows. If you are running X, kill the X Server and get back to the single (white on blue if on color, white on black if on mono) terminal emulation window that was present before X was started.

To test the installation of the AView Graphics Library, go to the `/usr/opt/aview/demo` directory, build any of the demos that are there, and run it.

If the software does not function correctly, first review the installation to make sure you have not overlooked any steps. In particular, make sure the graphics daemon (`grfxd` in `/usr/opt/aview/bin`) is running. If you feel you have followed the instructions correctly call your local Customer Service Representative.

9 Preparing a Software Trouble Report (STR)

If you believe you have found an error in the DG/UX AView Graphics Library or its documentation, or if you have a suggestion for enhancing or improving the product, use a Data General Software Trouble Report (STR) to communicate this to DG.

STR forms are available from the nearest DG office or DG representative, the Software Support Center, or can be found on your system in the file STR_form in /usr/release. If your contract permits, you may report the information called for in this section to your Data General representative. To help us process STRs quickly, **please include only one problem or suggestion on each STR form.** Please follow these guidelines when filling out your Software Trouble Report:

1. List the product name (AView), model number (H001A), and revision number as shown on the title page of this release notice. Also, include the revision of the DG/UX System that you are running. If you are running an update or patch, include its number as well.
2. Decide what kind of STR you are writing:
 - **Enhancement:** describe the proposed enhancement clearly and tell why you want it. The better we understand your desire, the easier it is for us to evaluate your request.
 - **Documentation Error:** list the title and part number of the document or man page and list the page and paragraph (or section) containing the error. Please state exactly why you think there is an error.
 - **Software Problem:** clearly and specifically state the problem so that support personnel can try to reproduce it. See the section **Software Problems** below for more details.
3. On the STR form provide all of the following information:
 - Date
 - Name and revision of the product
 - DG/UX revision being used
 - CPU type
 - Hardware configuration (if relevant)
 - Names and revisions of other software running on the system
 - The command line or scenario that caused the problem
 - The action(s) necessary to reproduce the problem
 - How often the problem occurs and how serious it is

4. If the problem occurred soon after installing a new revision of software or new hardware, please note this.
5. If you received an error message, please write down the exact text (and number, if present) of the message.

9.1 Software Problems

Report any particular activity or program running on the system that seems to cause the problem. If the program is supplied by DG, report in detail the exact steps used to reproduce the problem. If the program is supplied by another vendor or written by an installation, include a copy of the program and its source code, if possible.

Again, report in detail the exact steps used to reproduce the problem.

If your system panics, hangs or halts, see the DG/UX release notice for instructions on system dumps and DG/UX STRs.

End of AView Release Notice