

**Release Notice:**  
**X11R5**  
**DG/UX® X Window System**  
**for AViiON® Systems**

**Release 4.20**

**July 1997**

Part number 085-600112-00

This release notice applies to the following model:

P001A

## **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, duplication, 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 of Defense Federal Acquisition Regulation (DFAR) 252.227-7013 and in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause of Federal Acquisition Regulations (FAR) 52.227-19, whichever may apply.

DATA GENERAL CORPORATION  
4400 Computer Drive  
Westboro, Massachusetts 01580

**AViiON**, **DASHER**, and **DG/UX** are U.S. registered trademarks of Data General Corporation. All other trademarks mentioned herein are the property of their respective owners.

**Release Notice:**  
**X11R5**  
**DG/UX® X Window System for**  
**AViiON® Systems**  
**Release 4.20**

085-600112-00

July 1997

Revision History:

Effective with:

Original Release – July 1997    DG/UX System Release 4.20

# Contents

1	Introduction .....	5
2	Product description .....	5
3	Environment .....	6
3.1	Hardware .....	6
3.1.1	Supported Hardware .....	6
3.2	Software .....	7
4	Enhancements and changes .....	7
4.1	.snf fonts and font tools discontinued .....	7
4.1.1	X.desktop unbundled .....	7
4.2	xconfig .....	8
4.2.1	Improving Desktop Performance .....	8
4.2.2	UnixWare compatibility .....	9
4.2.3	Differences from the 88K Platform .....	9
5	Notes and warnings .....	9
5.1	Notes .....	10
5.1.1	X11R5 release organization .....	10
5.1.2	Using X with DG/UX information security .....	10
5.1.3	Internationalization support .....	11
5.1.4	X color management system .....	11
5.1.5	Athena widget set changed from X11R4 .....	11
5.1.6	Athena widget documentation .....	12
5.1.7	Keyboard support .....	12
5.1.8	SHAPE and MIT-SUNDRY-NONSTANDARD extensions .....	12
5.1.9	Run-time shared objects .....	13
5.1.10	Home and End Key interpretation in xterm .....	13
5.1.11	mwm WM_SAVE_YOURSELF protocol .....	13
5.1.12	/usr/opt/X11 structure .....	13
5.1.12.1	Traditional UNIX method .....	15
5.1.12.2	Method in <i>Porting and Developing Applications</i> .....	15
5.1.12.3	/usr/opt/X11 file system installation method .....	16
5.1.13	Applications that load files into /usr/opt/X11 .....	16
5.1.14	Xlib and Xt manual pages .....	16
5.1.15	editread and xterm stty settings .....	17
5.1.16	imake .....	17
5.1.17	Running UnixWare applications .....	17
5.2	Warnings .....	18

## Contents

5.2.1	X server swap usage .....	18
5.2.2	Use macros provided to write X extensions .....	18
5.2.3	xdgmail .....	18
5.2.3.1	Displaying xdgmail to remote systems .....	18
5.2.3.2	xdgmail and Application Modal dialog boxes .....	19
5.2.3.3	xdgmail and control buttons .....	19
5.2.3.4	Missing message view window .....	19
5.2.3.5	Conflict with mnemonics/accelerators .....	20
5.2.3.6	Segmentation fault if mailbox is truncated .....	20
5.2.4	Comments and blanks in resource files .....	20
5.2.5	X and Motif library changes .....	20
5.2.6	tearOffModel resource converter .....	21
5.2.7	Mislabeled fonts .....	21
5.2.8	Security in Trivial File Transfer Protocol .....	22
5.2.9	AVX-30 X terminal support .....	23
5.2.10	XrmParseCommand parameters .....	23
5.2.11	mterm and editread .....	23
5.2.12	xresedit .....	23
5.2.13	Running Motif applications on an X terminal .....	24
5.2.14	Motif handles pop-up menus incorrectly .....	24
5.2.15	Indicators disappear when XmNlabelType is XmPixmap .....	24
5.2.16	mwm requires one key binding .....	24
5.2.17	mwm redraws windows incorrectly .....	25
5.2.18	Terminating an X session .....	26
5.2.19	Manually Terminating the X Server .....	26
6	Documentation .....	27
6.1	Manuals .....	27
6.2	Intel-Related Documentation .....	28
7	Software distribution .....	28
8	Installation instructions .....	28
8.1	Any Platform .....	29
8.1.1	Saving and restoring .snf fonts .....	29
8.1.2	X11R5 images .....	29
8.1.3	Saving locally tailored files .....	30
8.1.4	Starting up and using the X Window System .....	31
8.2	Application Capture Option .....	32
8.2.1	Configuring the X Window System .....	33
9	Preparing a software trouble report .....	33

# 1 Introduction

This release notice describes the DG/UX® X Window System Revision 11 Release 5 and its installation. It includes information not available in the product manual.

Additional copies of this release notice can be printed. Use the file `/usr/opt/X11/release/X11_R4.20.rn` on the release CD-ROM.

## 2 Product description

The DG/UX X Window System combines the X Window System Release 5 and Motif Version 1.2.4.

The X Window System consists of

1. an X server, which manages a raster display and related input devices
2. a set of X clients, which communicate with the X server to perform terminal emulation, window management, and other necessary functions
3. programming libraries, which let you implement applications of your own
4. on-line documentation.

Motif, which is a derivative product of the Open Software Foundation, Inc.'s OSF/Motif, consists of

5. the **mwm** window manager
6. the User Interface Language (uil) compiler
7. subroutine libraries.

## 3 Environment

This section lists the hardware and software necessary for this revision of the DG/UX X Window System.

### 3.1 Hardware

The DG/UX X Window System is based on a client/server model. This model allows you to run the X Window System in two distinct environments, local display mode and remote display mode.

In local display mode, the X client and the X server run on the same machine, and they communicate using UNIX sockets or shared memory. The minimum configuration for an AViiON® workstation is a keyboard, a three-button mouse, a raster display, and a disk and CD-ROM drive or file server.

In remote display mode, the X server and clients execute on different machines connected by a local area network (LAN). The machine running the X server can be an AViiON workstation, an X terminal, a DGEM PC running eXceed, any other PC running a third-party X server package, or another manufacturer's workstation. This X display must be connected via the LAN to the system running the X client. The machine running the X client should be an AViiON server or workstation running the DG/UX X Window System.

Data General recommends a minimum of 32 MB for running the X Window System, and more if you use many graphical applications. Memory and CPU requirements are the same as those for the DG/UX system. Consult the DG/UX System Release Notice for these values.

#### 3.1.1 Supported Hardware

This release supports the AV2000 and AV3000 machines with either the integrated SVGA controller or the optional DG Model 7062 graphics controller.

## 3.2 Software

The remote machine presenting the displays generated by the X Window System must be equipped with support for TCP/IP LAN as well as an X server that supports the Version 11 protocol, release 4 or later.

The release notice for the DG/UX System Release lists the operating system requirements.

# 4 Enhancements and changes

This section contains information about enhancements and changes made in this release.

## 4.1 .snf fonts and font tools discontinued

In this release, the Server Natural Fonts (**.snf**) and the font tools **bdf2snf** and **showsnf** have been removed from the X Window System. The **.snf** fonts have been moved onto the Contributed Software for AViiON Systems, model R006AZA20D.

The X11R5 portable compiled fonts (**.pcf**) are the only fonts shipped and supported.

### 4.1.1 X.desktop unbundled

The package **xdt**, which contains the product X.desktop, is not included in this release. This product can be ordered separately. The model number (CD-ROM) is R073ABA1CD.

## 4.2 xconfig

The X server in this release provides a configuration utility called **xconfig**. **xconfig** lets you specify the screen resolution, type of mouse, and type of monitor connected to the system.

### 4.2.1 Improving Desktop Performance

The DG/UX MTS (Medium Term Scheduler) gives lower execution priorities to processes that seem compute-bound than to processes that seem io-bound. This model, when applied to the X Window System, causes the X server and any X clients using shared memory (**DGShm**) for communication (instead of sockets or TCP/IP) to appear compute-bound to the MTS. It in turn lowers their priority, thus lowering X Window System performance.

If you plan to use your workstation as an X desktop (not a server), implement the following kernel change (as **root**) as a workaround:

1. Edit the kernel configuration file using the **sysadm** menu System → Kernel → Build. The system will execute your editor loading the kernel configuration file.
2. Once in your editor, type **NOILEVEL** in the first line. This turns off the interactivity level.
3. Write **NOLANGUISHING** in the following line.
4. Write **MAXSLICE 50** in the following line. This sets the time slice to 50 for the process to execute, rather than 500 (the default).

The first lines of the file should look like this:

```
NOILEVEL
NOLANGUISHING
MAXSLICE 50
```

5. Save the file and then rebuild the kernel.
6. Reboot your system.



## 4.2.2 UnixWare compatibility

This release provides limited run-time support for some shrinkwrapped UnixWare applications. The links in `/usr/opt/X11/lib` and other locations link some UnixWare shared objects to the DG/UX X Window System equivalents. Thus, you may notice several additional files on an Intel-based platform that do not appear on an older, 88K-based platform. The 88K platform is not supported in this release.

This release also provides an Application Capture Option, `X11.aco`. This package is a collection of UnixWare shared libraries and links designed to provide a run-time environment for many shrinkwrapped UnixWare applications. Over time, this environment will be modified and improved to support a selected group of shrinkwrapped UnixWare applications.

For details and installation instructions, consult the release notice for the Application Capture Option.

## 4.2.3 Differences from the 88K Platform

The X servers in this release do not support the following features that were released in 88K versions of the DG/UX X Window System:

- X Input Extension
- X Color Correction

# 5 Notes and warnings

This section contains notes and warnings specific to this release.

## 5.1 Notes

This section includes information on features that may not work as expected but present no danger to your environment.

### 5.1.1 X11R5 release organization

X11 packages have been organized into “run-time” and “development” environments. This organization allows you to tailor your installation.

X11	Base X Window System package, containing executables, shared libraries, fonts, app-defaults, and bitmaps required for a running X Window System.
X11.man	X Window System runtime manual pages package, containing X and Motif man1 and man5 manual pages.
sdk.X11	X Window System development tools and components, containing libraries, header files, and documentation needed to build X clients as well as customized X servers.
sdk.X11.man	X Window System development documentation, containing X and Motif man3 manual pages.

Note: `sdk.x11` and `sdk.x11.man` are part of the Software Development Kit (SDK), which is bundled with the DG/UX system.

### 5.1.2 Using X with DG/UX information security

If you are using X with DG/UX information security, consult the release notice for the security product.

### 5.1.3 Internationalization support

In X11R5, routines have been added to **libX11** and **libXt** to aid in programming internationalized X clients. These include simple “locale” management routines to support:

- internationalized text drawing.
- internationalized window manager and text properties.
- localized databases.

Note that supported X clients have not incorporated any internationalization support.

### 5.1.4 X color management system

Until X11R5, color support was device-dependent; RGB values did not necessarily result in the same color from one display to another. R5 includes device-independent color with the X Color Management System (**Xcms**). **Xcms** functions were added to **Xlib** to support device-independent color spaces derivable from CIE XYZ color space (including CIE XYZ, xyY, L\*u\*v\*, L\*a\*b\*, and TekHVC). Monitors are characterized by data stored on **root** window properties. Set these properties with the new program **xcmsdb**. The default location of the “screen characterization database” loaded by **xcmsdb** is **/usr/lib/X11/dg6487.dcc**. A sample file is included with this release. However, even with this sample file, you will not get true device-independent color because physical characteristics of monitors change over time. Therefore, for accurate color reproduction, regularly calibrate your monitor and generate a similar file.

In addition, you can access client-side color name databases similar to **rgb.txt**, with the **Xcms** functions in **Xlib**. The default location for this file is **/usr/opt/X11/Xcms.txt**, although you can specify a different file in the **XCMSDB** environment variable.

### 5.1.5 Athena widget set changed from X11R4

The X11R5 version of Athena widgets (as supplied by Data General) changed significantly from X11R4. R5 supports both the normal Athena look and feel and the Motif look and feel. Use the “lookAndFeel” resource to select the Look and feel that a client (linked with the R5 **Xaw** library) uses at run time. Add one of the following lines to your **.Xdefaults** file to select a “look

and feel”:

```
*lookAndFeel: WindowManager  
*lookAndFeel: athena  
*lookAndFeel: motif
```

If “WindowManager” is specified and **mwm** is running when the client is invoked, the default Look and feel is “motif”; otherwise, it is “athena”.

Note: The Motif look and feel works only with Athena-based clients that are set up to use it. All Data General-supplied Athena-based X clients support the Motif look and feel.

### 5.1.6 Athena widget documentation

Although the Athena widget set is included in X11R5, **Xaw** manual pages are not. For Athena widget documentation, refer to the O’Reilly X Window System manuals, volumes 4 and 5 (see the order form enclosed with your shipment).

### 5.1.7 Keyboard support

X11R5 and X11R4 use the Mode\_Shift feature to handle Alt key sequences. As a result, many applications built with X11R3 libraries interpret keyboard input incorrectly with foreign keyboards. To correct this problem, relink the applications with the new libraries.

### 5.1.8 SHAPE and MIT-SUNDRY-NONSTANDARD extensions

The MIT SHAPE and MIT-SUNDRY-NONSTANDARD extensions are supported in the extensions library **libXext**.

## 5.1.9 Run-time shared objects

The manual *Porting and Developing Applications on the DG/UX System*, which is provided on line, lists the shared libraries included in this release.

### 5.1.10 Home and End Key interpretation in xterm

In DG/UX System 5.4, **xterm** incorrectly interpreted the Home and End keys. Because the VT102 terminal that **xterm** is emulating does not have either key, **xterm** was changed so that it does not generate any escape sequences for these keys. You can override this default behavior with the “translations” resource. For details, see the **xterm(1X)** manual page.

### 5.1.11 mwm WM\_SAVE\_YOURSELF protocol

With Motif release 1.2 and later, **mwm** handles the WM\_SAVE\_YOURSELF protocol independently from the WM\_DELETE\_WINDOW protocol. A client expressing interest in both protocols is notified for each (first WM\_DELETE\_WINDOW, then WM\_SAVE\_YOURSELF) when **f.kill** is invoked on a window. A client expressing interest in neither protocol has its connection to the X server terminated. After **mwm** sends a WM\_SAVE\_YOURSELF client message, it sets a timeout, the value of which is specified by **mwm**'s quitTimeout resource. The client's connection to the server is terminated when the timeout expires or when the client updates its WM\_COMMAND property.

### 5.1.12 /usr/opt/X11 structure

In DG/UX System 5.4, the directories **/usr/bin/X11**, **/usr/lib/X11**, and **/usr/include/X11** were symbolic links to directories in the **/usr/opt/X11** file system. This caused problems when installing third-party software, for the following reasons:

- When **/usr/opt/X11** is remote-mounted, installation of third-party X11 software fails. Software installation is performed by the user **root**. Unfortunately, across an NFS mount, the user **root** is translated to the user **nobody** on the remote system. The user **nobody** does not have write access.

- Files added to these directories actually are added to the **/usr/opt/X11** file system instead of the **/usr** file system. This makes disk space instructions incorrect. For example, a third-party package may instruct you to have a certain amount of free space in the **/usr** file system, while the free space actually is required in **/usr/opt/X11**.
- Relative symbolic links (links starting with **..**) added to these directories could be incorrect. For example, symbolic links created in the directory **/usr/bin/X11** are actually created in **/usr/opt/X11/bin**. The problem is that **/usr/opt/X11/bin** is four directory levels below root (**/**), instead of three as **/usr/bin/X11** implies. The manual *Porting and Developing Applications for the DG/UX System* recommends that you use absolute symbolic links (links starting with **/**) to avoid this problem.

Since DG/UX release 5.4.1, these directories are real directories in the **/usr** file system, and the individual files in the directories are linked, instead of the directories themselves. The following is the list of directories created in the **/usr** file system:

```
/usr/bin/X11
/usr/include/X11
/usr/include/X11/bitmaps
/usr/lib/X11
/usr/lib/X11/app-defaults
/usr/lib/X11/fonts
/usr/lib/X11/fonts/100dpi
/usr/lib/X11/fonts/75dpi
/usr/lib/X11/fonts/misc
```

The setup script **/usr/sbin/setup.d/usr/X11\_0.X11.do** creates the directories listed above, and creates symbolic links for each of the files contained in the corresponding **/usr/opt/X11** directory (for example, **/usr/opt/X11/bin** for **/usr/bin/X11**).

You can use the following methods to install third-party software:

- Traditional UNIX method.
- Method described in *Porting and Developing Applications for the DG/UX System*.
- Direct access of the **/usr/opt/X11** file system.

The following examples use the package **foo** installing the file **bar** into the directory **/usr/bin/X11**.

### 5.1.12.1 Traditional UNIX method

The traditional UNIX installation method views the **/usr** file system as one large file system containing all the software on a system. This method is derived from earlier versions of UNIX, which did not have symbolic links, and had limited support for Logical Disk Units (LDUs). Follow the instructions in the third-party software to place the files into the appropriate directory in **/usr/bin/X11**.

### 5.1.12.2 Method in *Porting and Developing Applications*

The method described in *Porting and Developing Applications on the DG/UX System* installs the files of an optional package in the **opt** subdirectory of a file system, such as **/usr/opt/foo** for the package **foo**. Symbolic links are then added in “public” directories (for example, **/usr/bin/X11**) that point to files in the **opt** subdirectories. This method recommends that you use absolute symbolic links in the X11 directories. For example, use the following command to link the file **bar** in the package **foo**:

```
ln -s /usr/opt/foo/bar /usr/bin/X11/bar ↵
```

The X11 directory change in DG/UX releases allows you to use relative symbolic links. However, use absolute symbolic links for downward compatibility with 5.4 systems. All of the following **ln** commands work on a Release 5.4.2 or later system, but only the last works on a Release 5.4 system:

```
ln -s ../../opt/foo/bar /usr/bin/X11/bar ↵
ln -s ../../../usr/opt/foo/bar /usr/bin/X11/bar ↵
ln -s ../../../../../../usr/opt/foo/bar /usr/bin/X11/bar ↵
```

Also, a symbolic link in the following form works on a Release 5.4 system only:

```
ln -s ../../../opt/foo/bar /usr/bin/X11/bar ↵
```

### 5.1.12.3 `/usr/opt/X11` file system installation method

This method directly accesses the `/usr/opt/X11` file system by moving and linking the executables:

```
ln -s /usr/opt/foo/bar /usr/opt/X11/bin/bar ↵
ln -s ../../foo/bar /usr/opt/X11/bin/bar ↵
ln -s ../../../opt/foo/bar /usr/opt/X11/bin/bar ↵
ln -s ../../../../usr/opt/foo/bar /usr/opt/X11/bin/bar ↵
```

An optional package should never write in the installation area of another package. The `/usr/opt/X11` file system is intended only to segregate the X11 package files from the base DG/UX system. If a third-party software package does install its files in this manner, you can re-run the setup script `/usr/sbin/setup.d/usr/X11__0.X11.do[ne]` to remedy the situation. This script creates symbolic links for all new files. Since this script automatically runs when a system is upgraded, rerun it only if you install a package that loads directly into `/usr/opt/X11` after a system is upgraded.

### 5.1.13 Applications that load files into `/usr/opt/X11`

After you load an application that loads files directly into `/usr/opt/X11`, become **root** and rerun the X11 setup script:

```
/usr/sbin/setup.d/usr/X11__0.X11.done / /usr ↵
```

This script creates the necessary symbolic links in `/usr/bin/X11`, `/usr/lib/X11`, and `/usr/include/X11` to let the package run correctly.

### 5.1.14 `Xlib` and `Xt` manual pages

The `Xlib` and `Xt` manual pages included in this release are incomplete. The X Consortium provided several functions without proper on-line documentation. For a list of functions without manual pages, refer to the file `/usr/opt/X11/release/manpages.missing`. The complete `Xlib` and `Xt` documentation is available in the O'Reilly X Window System manuals, volumes 2 and 5.



### 5.1.15 **editread and xterm stty settings**

If **editread** is turned on when an **xterm** is started in the background (for example, **xterm &**), **xterm** can pick up the **stty** settings either before or after **editread** sets them up. This is because **xterm** reads the **stty** settings of the parent terminal (**/dev/tty**) to establish the settings it will use. **editread**, however, sets up the **stty** settings during the reinitialization, after the background **xterm** is launched. Therefore, either **xterm** or **editread** can establish the **stty** settings. This condition appears often in the **cs**, but rarely in the **sh**.

To work around this situation, specify a **.cshrc** file that resets your **stty** settings in **xterm** after the C shell comes up, or to turn **editread** off before you launch the **xterm** window. If you turn off **editread** before the launch, you cannot turn it back on.

### 5.1.16 **imake**

**imake** and **imake** support are now part of the standard DG/UX X Window System. For more information, see the **imake(1)** and **makedepend(1)** manual pages.

### 5.1.17 **Running UnixWare applications**

If you run shrinkwrapped UnixWare applications and receive the error `Cannot open X Display`, try setting the following environment variables to work around the problem. You should submit an STR on this failure (even if this workaround works) so that Data General can improve the product. If you submit an STR, please specify whether or not the ACO package was in use.

Set the following environment variable before starting the application (Bourne shell syntax):

```
DISPLAY=`hostname`:0; export DISPLAY
XLOCAL=TCP;export XLOCAL
```

If this first setting does not work, try:

```
xhost +localhost
DISPLAY=localhost:0;export DISPLAY
```

## 5.2 Warnings

This section includes information on software and hardware that may not work as expected and may affect your data.

### 5.2.1 X server swap usage

The X server can monopolize large quantities of system swap space. This is most noticeable when running clients that demand large numbers of small- to medium-sized pixmaps, or several large pixmaps.

You should restart the server regularly for the system to reclaim the used memory. To do so, press the “restart” button on the **xdm** login panel.

### 5.2.2 Use macros provided to write X extensions

With Data General’s shared memory transport implementation, you must use the standard macros provided in the X library to write data to the client’s output buffer. These standard macros include `GetReq` (for getting the next available X request packet) and `Data` (to place data in the buffer and pad appropriately). Writing directly to the buffer and manipulating the buffer pointers (`bufptr`, `buffer`, `bufmax`) without using the standard macros results in output buffer corruption and unpredictable failures.

### 5.2.3 xdgmail

This section discusses known **xdgmail** problems.

#### 5.2.3.1 Displaying xdgmail to remote systems

If you use **xdgmail** with the resource `XdgMail*editCmd` set to something other than `builtin`, you may not be able to create or edit messages. This happens when you start **xdgmail** on one machine and display it on another across the network. In this instance, you must set the shell `DISPLAY` variable before you start **xdgmail**. For example, if you use `/bin/sh` to display **xdgmail** on the server “foo” to the workstation “bar,” you must execute the following commands

on the workstation (“foo”):

```
env DISPLAY=bar:0 xdgmail & ↵
```

Note: Using the `-display` command line option will not work, since this applies only to `xdgmail` and not to other processes (such as `xterm`) started from within `xdgmail`.

### 5.2.3.2 `xdgmail` and Application Modal dialog boxes

Because many of the dialog boxes in `xdgmail` are application-modal (that is, they must be dismissed before `xdgmail` will respond to any other input), it may appear at times that `xdgmail` is inexplicably hung. If `xdgmail` stops responding to input, look for boxes related to `xdgmail` that are hidden by other windows. Also, make sure all boxes other than the main window and the most current dialog box have been dismissed. Since the error and warning message boxes are relatively small, one or more of them is typically the culprit.

### 5.2.3.3 `xdgmail` and control buttons

If you set the `XdgMail*buttons` resource to `none`, X displays the following warning:

```
Warning:
  Name: controlMenu
  Class: XmRowColumn
  failure of geometry request to "almost" reply
```

This warning does not affect the behavior of `xdgmail`. (This problem will be corrected in a later release of `xdgmail`.) To avoid the problem, keep at least one button defined in the `XdgMail*buttons` resource.

### 5.2.3.4 Missing message view window

Sometimes when `xdgmail` starts up after a system reboot, it appears to be missing the message view window. Actually, the movable grip between the message button window and the message view window has moved to the bottom of the `xdgmail` main window. To correct this, identify the small square “grip” in the lower right corner of the `xdgmail` window and drag it up the

screen with the mouse until the message view window and the button box are the desired size relative to one another.

### 5.2.3.5 Conflict with mnemonics/accelerators

If you use a menu button mnemonic identical to an accelerator or another operation, you get both selections. For example, if you press “h” while in the menu bar, the help window and the search dialog box pop up. This problem will be corrected in a later release of **xdgmail**.

### 5.2.3.6 Segmentation fault if mailbox is truncated

If the end of a mailbox is truncated, a currently running **xdgmail** session may exit with a segmentation fault. You cannot restart **xdgmail** until one or more empty lines are appended to the mailbox.

## 5.2.4 Comments and blanks in resource files

Do not use C-style comments (*/\* \*/*) and shell-style comments (*#*) inside resource files. Instead, use exclamation marks (!) at the beginning of a line. For more information, see the **xrdb** manual page.

Do not put or leave extra blanks at the end of resource definitions because trailing blanks are treated as part of the resource value.

## 5.2.5 X and Motif library changes

Because of changes in the X and Motif libraries, some applications that use shared libraries may no longer run on X11R5. Because the X Consortium removed some undocumented functions from the public namespace, any application that references those functions will not work. Data General is attempting to provide compatibility between X11R4 applications and X11R5 shared libraries. If you experience dynamic loader errors when running an X11R4-based application on an X11R5 machine, contact Data General Customer Support. To correct the problem, Data General will send you the X11R4 library. In general, this is how old versions of shared libraries are supported.

## 5.2.6 **tearOffModel resource converter**

The `tearOffModel` resource converter is available only in applications that provide it. For details, see the `XmBulletinBoard(3)` manual page.

## 5.2.7 **Mislabeled fonts**

The following fonts are erroneously classed as 8-bit iso8859-1 fonts. This means that only the lower 128 glyphs (characters) are accessible. Renaming the fonts or “completing” them would correct the situation but would put Data General’s implementation at odds with the X Consortium. “Pretending” that they are 7-bit fonts is the common usage solution to this problem, because this font set has existed in this state since X11R4.

```
-misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1
-misc-fixed-bold-r-normal--13-100-100-100-c-80-iso8859-1
-misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1
-misc-fixed-bold-r-normal--13-120-75-75-c-80-iso8859-1
-misc-fixed-bold-r-semicondensed--13-100-100-100-c-60-iso8859-1
-misc-fixed-bold-r-semicondensed--13-120-75-75-c-60-iso8859-1
-misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1
-misc-fixed-medium-r-normal--10-70-100-100-c-60-iso8859-1
-misc-fixed-medium-r-normal--13-100-100-100-c-70-iso8859-1
-misc-fixed-medium-r-normal--13-100-100-100-c-80-iso8859-1
-misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-1
-misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1
-misc-fixed-medium-r-normal--20-140-100-100-c-100-iso8859-1
-misc-fixed-medium-r-normal--20-200-75-75-c-100-iso8859-1
-misc-fixed-medium-r-normal--8-60-100-100-c-50-iso8859-1
-misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
-misc-fixed-medium-r-normal--9-80-100-100-c-60-iso8859-1
-misc-fixed-medium-r-normal--9-90-75-75-c-60-iso8859-1
-misc-fixed-medium-r-semicondensed--12-110-75-75-c-60-iso8859-1
-misc-fixed-medium-r-semicondensed--12-90-100-100-c-60-iso8859-1
-misc-fixed-medium-r-semicondensed--13-100-100-100-c-60-iso8859-1
-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
-schumacher-clean-bold-r-normal--10-100-75-75-c-60-iso8859-1
-schumacher-clean-bold-r-normal--10-100-75-75-c-80-iso8859-1
```

```

-schumacher-clean-bold-r-normal--12-120-75-75-c-60-iso8859-1
-schumacher-clean-bold-r-normal--12-120-75-75-c-80-iso8859-1
-schumacher-clean-bold-r-normal--13-130-75-75-c-80-iso8859-1
-schumacher-clean-bold-r-normal--14-140-75-75-c-80-iso8859-1
-schumacher-clean-bold-r-normal--15-150-75-75-c-90-iso8859-1
-schumacher-clean-bold-r-normal--16-160-75-75-c-80-iso8859-1
-schumacher-clean-bold-r-normal--8-80-75-75-c-80-iso8859-1
-schumacher-clean-medium-i-normal--12-120-75-75-c-60-iso8859-1
-schumacher-clean-medium-i-normal--8-80-75-75-c-80-iso8859-1
-schumacher-clean-medium-r-normal--10-100-75-75-c-50-iso8859-1
-schumacher-clean-medium-r-normal--10-100-75-75-c-60-iso8859-1
-schumacher-clean-medium-r-normal--10-100-75-75-c-70-iso8859-1
-schumacher-clean-medium-r-normal--10-100-75-75-c-80-iso8859-1
-schumacher-clean-medium-r-normal--12-120-75-75-c-60-iso8859-1
-schumacher-clean-medium-r-normal--12-120-75-75-c-70-iso8859-1
-schumacher-clean-medium-r-normal--12-120-75-75-c-80-iso8859-1
-schumacher-clean-medium-r-normal--13-130-75-75-c-60-iso8859-1
-schumacher-clean-medium-r-normal--13-130-75-75-c-80-iso8859-1
-schumacher-clean-medium-r-normal--14-140-75-75-c-70-iso8859-1
-schumacher-clean-medium-r-normal--14-140-75-75-c-80-iso8859-1
-schumacher-clean-medium-r-normal--15-150-75-75-c-90-iso8859-1
-schumacher-clean-medium-r-normal--16-160-75-75-c-80-iso8859-1
-schumacher-clean-medium-r-normal--6-60-75-75-c-40-iso8859-1
-schumacher-clean-medium-r-normal--6-60-75-75-c-50-iso8859-1
-schumacher-clean-medium-r-normal--6-60-75-75-c-60-iso8859-1
-schumacher-clean-medium-r-normal--8-80-75-75-c-50-iso8859-1
-schumacher-clean-medium-r-normal--8-80-75-75-c-60-iso8859-1
-schumacher-clean-medium-r-normal--8-80-75-75-c-70-iso8859-1
-schumacher-clean-medium-r-normal--8-80-75-75-c-80-iso8859-1

```

### 5.2.8 Security in Trivial File Transfer Protocol

Improved security in TFTP allows access to the files in /tftpboot only. X terminals should use NFS rather than TFTP to access configuration and font files.

## 5.2.9 AVX-30 X terminal support

Support for AVX-30 X terminal has been removed from this release of the DG/UX X Window System.

## 5.2.10 XrmParseCommand parameters

In X11R4, `XrmParseCommand()` let you pass a string of the form *AppName.class* as the fourth parameter (name), but the fourth parameter could only be the application's name. Since this use of `XrmParseCommand()` was not documented and is no longer available in X11R5, applications (such as the contributed client `xlock`) that used this feature must be modified to work properly with X11R5. An application that has this problem will not act on some command line options. To resolve this problem, change your `app-defaults` file instead of using the command line option.

## 5.2.11 mterm and editread

If you use `editread`, start an `mterm` session, change mode, and then press a character prior to pressing the Return (Enter) key, `editread` starts replicating the character you pressed and beeps uncontrollably. You can kill this `mterm` session with either the `mwm` window menu close button or `dg_kill(1)`. To avoid this problem, either disable `editread` before starting `mterm` or press the Return key after changing emulation mode.

## 5.2.12 xresedit

When you enter the `xresedit` edit window to edit a font resource, the window lists the wrong number of fonts that match the default font string. From the edit window, you can pick almost any value for each element of the font string even if that choice does not match any font. If you pick an invalid option, the window will not update properly until you create a valid font string. To avoid this problem, always pick one of the font elements and select the `*` choice or the already selected choice. When you make this selection, the window lists valid options. In this way, you can avoid selecting invalid font elements.

Following is an example workaround of this problem. If you are editing a resource with the following font string:

```
-misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-i
```

select the **font** pull-down menu and then select either \* or **misc** to get a valid font. Now, you can select only valid options for each element.

### 5.2.13 Running Motif applications on an X terminal

When running Motif applications on an X terminal, turn off the option “Retain X Settings” for the applications to run properly.

### 5.2.14 Motif handles pop-up menus incorrectly

Motif handles pop-up menus incorrectly under certain conditions. When running an application that supports pop-up menus, you might “lose” the keyboard by bringing up the pop-up menu and pressing the Escape key. Switch to another window and then back to the original window to restore the focus.

### 5.2.15 Indicators disappear when XmNlabelType is XmPixmap

When XmNlabelType is set to XmPixmap at the creation of a ToggleButton, but the actual pixmap is added later using SetValues, the indicator button disappears. This is not a problem if a pixmap is designated at creation time. It also is not a problem if one creates the ToggleButton with XmNlabelType set to the default, sets the pixmap, and then sets XmNlabelType to XmPixmap.

### 5.2.16 mwm requires one key binding

**mwm** now requires that at least one key binding be present in a “Keys” definition in the **.mwmrc** file. In DG/UX 5.4, empty Keys definitions were legal and worked as expected.



```
RKeys MyKeyBindings
{
    !
    ! mwm defaults deleted
    !
}
```

Along with an X resource of `Mwm*keyBindings: MyKeyBindings`, this had the desired effect of eliminating all `mwm` key bindings. However, in DG/UX 5.4.2 and later revisions, when the same definitions are used, `mwm` reports this error:

```
Rmwm: Key bindings MyKeyBindings not found, using builtin key binding
```

If you add one key definition to the Keys entry, the message is eliminated. Thus, `.mwmrc` now looks like this:

```
Keys MyKeyBindings
{
    !
    ! mwm defaults deleted
    !

    <Key>F1    root    f.nop
}
```

### 5.2.17 `mwm` redraws windows incorrectly

In the Motif 1.2.4 version of `mwm` (in this release), if you first move a window so that the frame border of the window being moved is beneath the pop-up box that shows the screen coordinates and then release the mouse button, you may get a short different color line in the window. This bug in `mwm` will be fixed in a future Motif release. Issue the `xrefresh` command to redraw the window correctly.

### 5.2.18 Terminating an X session

You can no longer terminate an X session by killing the window manager.

In previous releases, the files **system.Xsession** and **sample.Xsession** ended with the line

```
exec mwm
```

This allowed the session to be terminated by killing **mwm**.

In this release, these files end with the lines

```
mwm &  
wait
```

This takes **mwm** out of the role of holding up the session and solves problems where defunct processes were accumulating. However, you cannot terminate the session by killing **mwm**.

To terminate the session, replace the **exec mwm** line in your personal **Xsession** files with the following two lines:

```
mwm &  
wait
```

You can then use the root menu option **Logout** to terminate the session.

### 5.2.19 Manually Terminating the X Server

If the X server unexpectedly terminates, the Kernel Terminal Emulator may not be able to restore its state. If this happens, you must power down the system and reboot it from a cold power up, to allow the graphics hardware to reset itself correctly. Therefore, do not use **kill -9** if you can avoid it. If you feel that you must kill the X server, use **kill -1** first so that the X server will shut down and allow the Kernel Terminal Emulator to restore its state.

## 6 Documentation

### 6.1 Manuals

The following documents ship in printed form with the DG/UX X Window System:

Publication	Part Number
X Window System Release Notice	085-600421
X Window System User's Guide, OSF/Motif Edition	069-100229
X Window System Document Order Form	069-100392
OSF/Motif Document Order Form	069-100411

The X Window System Document Order Form lists additional manuals that you can order from O'Reilly & Associates and from Prentice Hall.

See also various on-line documents in `/usr/opt/X11/doc`.

The DG/UX documentation is on CD-ROM. The **DG/UX System Release 4.20 Documentation CD-ROM** contains DG/UX documentation and the WorldView Viewer software, which enables you to view the DG/UX documentation on-line. The *Quick Start Guide for Viewing DG/UX Documentation*, which you received in printed form with your DG/UX system, discusses how to use WorldView.

The manual *Porting and Developing Applications for the DG/UX System*, which is on the documentation CD-ROM, contains a chapter on the DG/UX X Window system.

Printed copies of all DG/UX manuals can be ordered at additional cost. From the WorldView collection contents page, click on the selection bar for descriptions of DG/UX manuals, ordering information, and a list of topics and tasks.

## 6.2 Intel-Related Documentation

The following document contains information related to this release and can be ordered from:  
 ATI Graphics  
 33 Commerce Valley Drive East  
 Thornhill, Ontario Canada L3T 7N6

Publication	Part Number
ATI Graphics Pro Turbo User's Guide	ATI P/N 10701F1610

## 7 Software distribution

This release of the DG/UX X Window System package is part of the general DG/UX operating system release package. Please see the DG/UX release notice for exact media details.

For a complete list of files in the DG/UX X Window System package, see the following files under `/usr/opt/X11/release` after the system is loaded (associated package in brackets).

```
X11_R4.10.fl           [X11]
X11.man_R4.10.fl      [X11.man]
sdk.X11_R4.10.fl      [sdk.X11]
sdk.X11.man_R4.10.fl  [sdk.X11.man]
```

## 8 Installation instructions

Installation of the DG/UX X Window System is covered in the manual *Installing the DG/UX System*. This section contains information necessary for setting up the X Window System.

## 8.1 Any Platform

This release of the DG/UX X Window System can be installed on both stand-alone workstations and OS server-client configurations using **sysadm**. This release contains Release 5 of the X11 product and replaces the current contents of **/usr/opt/X11**.

### 8.1.1 Saving and restoring .snf fonts

If you have **.snf** fonts of interest, you must save them before installing this release.

### 8.1.2 X11R5 images

X11R5 has been divided into two logical environments: run-time and development. This differentiates between what is needed to run X and what is needed to do X development (that is, X clients and server customization). The development package is part of the Software Development Kit (SDK).

- X Window System Image [X11]

This required image contains the necessary files to support the DG/UX X Window System for a run-time environment. That is, it contains all X11 executables, shared objects, fonts, bitmaps, resource files, and other miscellaneous files needed for a functioning X environment. This image loads into the directory **/usr/opt/X11**. The current contents of **/usr/opt/X11** are overwritten during loading.

- X11 Manual Pages [X11.man]

This optional image contains all the section 1 (user commands) and section 5 (miscellaneous) manual pages supplied by the X Consortium, OSF, and Data General. This image fits in the space allocated for the X11 package in **/usr/opt/X11**.

- X Software Development Environment Image [sdk.X11]

This optional image contains all header files and static libraries needed to develop clients and customized X servers. This image also contains all on-line documentation (excluding manual pages) in both PostScript and printable formats.

- X Software Development Manual Pages [sdk.X11.man]

This optional image contains the X software development manual pages. This package contains all the section 3 (subroutine) manual pages supplied by the X Consortium, OSF, and Data General.

### 8.1.3 Saving locally tailored files

Any files that were customized for your local site will be destroyed during the installation process. You should back up these files to tape before starting the installation process. This backup of selected files is in addition to a complete system backup, discussed in a later section. For a convenient means of restoring your customizations without the cumbersome process of restoring a few files off a complete system backup, keep a separate copy of these files (made with **cpio** or **tar**).

The following is a suggested list of such files. It is not an exhaustive list, nor is it implied that the system administrator should automatically save every item on the list. This list is a reminder of possible places where the system administrator may have customized the local environment:

**/usr/opt/X11/include/X11/bitmaps**

(In this release, only the bitmaps received from the X Consortium distribution are included).

**/usr/opt/X11/lib/fonts/**

(In this release, only the X Consortium distribution fonts are supplied).

**/usr/opt/X11/xtd/\***

**/usr/lib/X11/app-defaults/\***

**/usr/lib/X11/rgb.\***

**/usr/opt/X11/lib/sample.Xdefaults**

**/usr/opt/X11/lib/xstart/xstart.sh**

Note: User sites should not modify any files in the directory structure. If modifications are necessary, find the symbolic link under `/usr/bin/X11`, `/usr/lib/X11`, or `/usr/include/X11` that has the same name as the actual file and delete it. Copy the original file from `/usr/opt/X11` into the same filename under `/usr/bin/X11` (or the other directories as necessary) and make the modifications there.

### 8.1.4 Starting up and using the X Window System

Once the installation steps for your system are complete, use the `init` command to bring the system back to run level 3. For information on changing run levels, see the manual *Managing the DG/UX System*. During the process of going to run level 3, the X server (`X`), and the X Display Manager (`xdm`) are started. When the software packages were set up, an entry was automatically added into `/etc/inittab` to start up the X Display Manager. `xdm` will present the `xdm` login window, which is used for login purposes. For more information about using and controlling `xdm`, consult the `xdm` and `XDMCP` manual pages.

`xdm` has replaced the `xstart` interface as the preferred method for starting up an X session. When a user first logs into the system using `xdm`, his or her private `.xstart.sh` (or other such startup file) is not used. `xdm` uses an `.Xsession` file to start up a tailored environment. You can reestablish your environment in one of two ways.

You can get backward compatibility with `xstart(1X)` by copying the file `/usr/lib/X11/xdm/xstart.Xsession` to your `$HOME` directory as the file `$HOME/.Xsession`. However, the preferred method is for the user to copy the file `/usr/lib/X11/xdm/sample.Xsession` to your `$HOME` directory as the file `$HOME/.Xsession`. Then, using an editor (such as `ed(1)`, `vi(1)`, etc.), enter your standard client set into this file, replacing the three clients (`xterm`, `xclock`, and `xbif`) provided by default. For further information about these two files, consult the `Xsession(5)` manual page.

Before using the DG/UX X Window System, make sure that `/usr/bin/X11` is on your path. Your path contains the list of ordered directories to search for programs to be executed.

To check your path under the Bourne shell, execute the following:

```
echo $PATH
```

To check your path under the C shell, execute the following:

```
printenv PATH
```

If `/usr/bin/X11` does not appear in your environment variable, you must add `PATH` then `/usr/bin/X11` to your search path.

For the Bourne shell, use the following command to correctly set the `PATH` environment:

```
PATH=$PATH:/usr/bin/X11; export PATH
```

For the C shell, use the following command:

```
setenv PATH ${PATH}:/usr/bin/X11
```

You may want to edit your `.login` or `.cshrc` files to initialize this variable.

For information about how to log in and use the DG/UX X Window System, see Volume 3 of the O'Reilly X Window System Series.

User accounts created prior to installation of this release may use the Athena “lookAndFeel” by default (unless you set the resource `*lookAndFeel: motif`). New user accounts where `sysadm` is used to create the `$HOME` directory will have the motif “lookAndFeel” by default. To do this, set the resource `*lookAndFeel: motif` in the `.Xdefaults` file that is copied from the `/etc/skel` directory during creation of the `$HOME` directory by `sysadm`.

## 8.2 Application Capture Option

The optional Applications Capture Option [X11.aco] image contains UnixWare shared objects and supporting files that provide a run-time environment that allows a selected number of shrinkwrapped UnixWare applications to run. This package adds files to `/usr/opt/X11` and places some links in `/usr`. Before installing, make sure that `/usr/opt/X11` has 140,000 blocks of disk space allocated to it. If not, use `sysadm` to expand `/usr/opt/X11`. This package requires approximately 2,000 blocks of space in `/usr/opt/X11`. This package uses approximately 50 I-nodes in the `/usr` file system.



## 8.2.1 Configuring the X Window System

In this release, the DG/UX X Window System does not start the local X server by default. To set up your X server, run the **xconfig** program. This program prompts you for information, such as the type of mouse and monitor connected to your system. If you need help at any point, type a question mark (?). If you do not install the X Window system from the console, you must reboot before running **xconfig**.

The program **xconfig** modifies a set of files in the directory **/var/X11/Xserver** and sets **Xdg**, which is a link in the directory **/usr/opt/X11/bin**. Do not modify these files manually.

# 9 Preparing a software trouble report

If you believe that you have found an error in the DG/UX system or its documentation, consult the manual *Preventing and Fixing Problems on the DG/UX System*. This manual, which is provided in both printed and CD-ROM versions with your system, explains the procedures for filing a software trouble report (STR).

A copy of the standard STR form suitable for printing on a line printer is in the file **/usr/release/STR\_form**.

1. The product name is **DG/UX X Window System for AViiON Systems**.
2. The model is **H001A**.
3. The revision is **DG/UX R4.20**.
4. If you were running a UnixWare application, indicate whether or not the ACO package was in use.

To comment on the documentation, fill out the reader comment form in **/usr/release/doc\_comment\_form** and send it to the address on the form.

End of DG/UX X Window System Release Notice

