EXPLORATION & PRODUCTION

EPTG

TECHNOLOGY GROUP

Geoscience Technology Bulletin

XCAT - X11 Text Display Tool

By Joe M. Wade

F95-G-20
June 21, 1995
951720001-TUL

EPTG - Tulsa

Agreement D-95-2518, Seismic Signal Analysis, is funded by the following BUs: MCBU, OBU, Norway-Expl., Egypt, Egypt-Expl., UK-Expl., Trinidad, Netherlands, Netherlands-Expl., Canada Northwest, Canada Exploration, E/USA, LASA, MEA

XCAT - X11 Text Display Tool

Joe M. Wade
Geoscience Technology Division

F95-G-20
June 21, 1995
Project Agreement D-95-2518
951720001-TUL

Introduction

The **xcat** program was originally designed as a simple tool with which to display textual information while running a graphical interface under the X11 environment. The goal was simple - prevent diagnostic text of programs from being sent back to the window from which the application was started, since this was usually covered on the screen by the application itself. From these humble origins has developed a widely used program having a multitude of capabilities. It may be used as a simple display device, a stand-alone application, a front-end application, or as a tool from within another application. Anyone who works within an X11 environment should benefit from the ability provided by **xcat** to easily display textual information in a scrollable window with capabilities for editing, printing, or saving to a file the information. This note will first illustrate simple uses for xcat and progress on to more complex usage.

Use as a Simple Display Device

In its simplest function, **xcat** will receive piped data and display it in an X11 Motif scrollable text widget. Figure 1 is an example display created by running:

ls** /usr **| xcat

Figure 1

The text data that **xcat** receives as input can also be output from **xcat** through the *stdout* mechanism unchanged. This is enabled by the command line option *-**p***. An example of using this option would be to create a window into a processing stream where the data could be visually verified without interrupting the processing flow:

create_data* | *xcat -p* | *process_data

In this example, **create_data** and **process_data** represent programs which would create and process text data, respectively.

To simply display a file on a Unix file system, **xcat** works similarly to the Unix **cat** command. For instance, Figure 2 shows an example of executing:

***xcat** demofile*

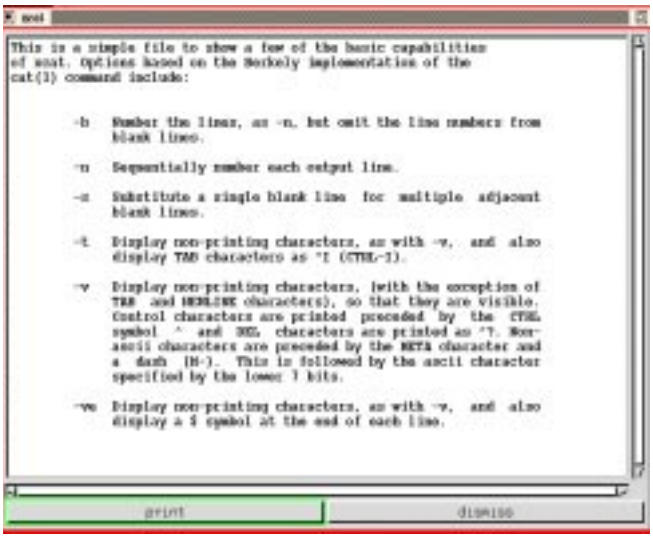

Figure 2

Here, *demofile* is a file containing common text and tab characters.

By default, two buttons appear below the displayed text which each serve dual functions while the text is being received vs. after all text has been displayed. During processing of input data, the buttons will allow control of the scrolling in the text window and termination of the input data processing. The appearance of the main panel in this mode is shown later in Figure 5. After receipt of all the input data, the function of these buttons switches to that of a text processing button and panel dismissal button. By default, the processing button on the left is used for piping the text data on to a print command. This occurs automatically, assuming that the **PRINTER** or **LPDEST** variable is set, depending on the type of Unix system in use, in the user's environment. Alternately, an environmental variable called **XCAT_PRINT_CMD** may be set, which may contain a command string to which the entire contents of the text window will be piped. For example, to run the contents through the Unix command **enscript** to rotate the text rather than the using the default print command, this could be set to ***enscript -r***. It should also be noted that this command need not necessarily be a print command. It may be any command capable of processing text. For that reason, there are options for changing the label and function of this button via the command line arguments *-**pc*** and *-**pl***, or the X11 resources **printCmd*** or **printLabel***, respectively. In order to edit the text in the display window before further processing, the text widget must have been declared editable, either through the specification of *-**ed*** on the command line or via the ****editable*** resource specification. Further details on the main panel's processing buttons will be given later on in this paper.

Text Display Options

The **xcat** program has features for displaying text similar to the Unix **cat** utility. These option were based on the Berkeley implementation of **cat** and may be used for numbering output lines, compressing multiple blank lines, displaying non-printing characters, and displaying the end of line location. Redisplaying the *demofile* text using a few of these options shows how the appearance of the file can be changed. The command executed to get the display shown in Figure 3 was:

xcat -n -t -ve *demofile*

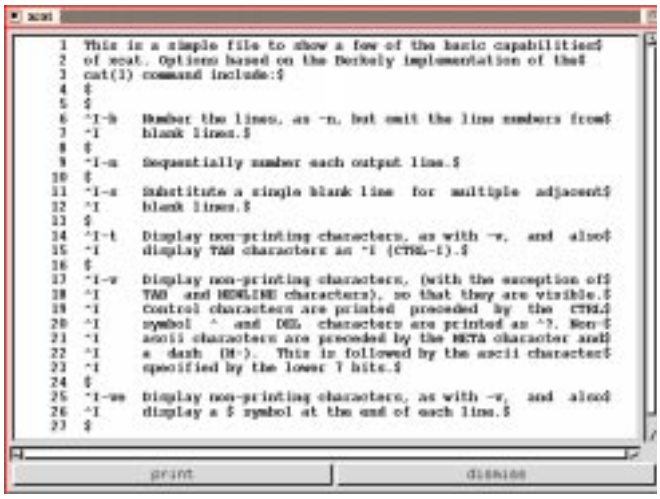

Figure 3

A special option was written into the program to take advantage of the graphical ability to highlight or underscore text. This is enabled by the ***-man*** command line option or the ***manPrinting*** X11 resource. Figure 4 shows an example of text displayed using this option in the command:

xcat -man *xcat.manpage*

Figure 4

Child Processing within Xcat

Xcat also has the capability of running child processes over which the user exercises some control. This is activated via the ***-e*** command line option, which was used to correspond to similar functionality within the X11 program **xterm.** All *stdout* and *stderr* output from the child process is displayed in the **xcat** text window. The child process may be terminated via the right-most button on the **xcat** main panel. The control buttons will not change to the ***print*** and ***dismiss*** functions until the child process has finished. Figure 5 shows an example of how the display will look while the child command is processing; in this case, a Unix **ping** command was run with the ***-s*** option to continually check the status of a target machine.

xcat -e ping -s *gsclus13*

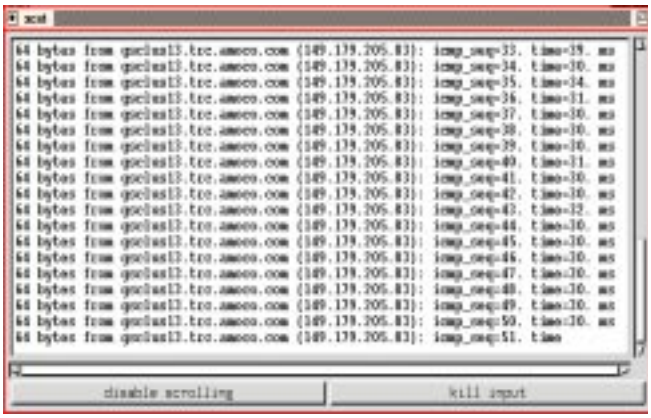

Figure 5

The child process, in this case **ping.** will run until the process exits normally, or is forcefully terminated by pressing the button labelled ***kill input.***

"Hidden" Features

Menu Bar Functions

By default, the user will see no menu bar on the **xcat** panel. This is because of **xcat**'s origins as a simple text display tool to be used in conjunction with other applications. In reality, there is a menu bar present which has a ***File*** pulldown containing functions for saving the textual information to a specified file and also a ***Help*** pulldown for accessing functions for various pieces of the program. The menubar may be made visible by entering ***-menu*** on the command line or specifying the ***menuBar*** resource in the application defaults file. Its appearance is shown in Figure 6 using:

xcat -menu *demofile*

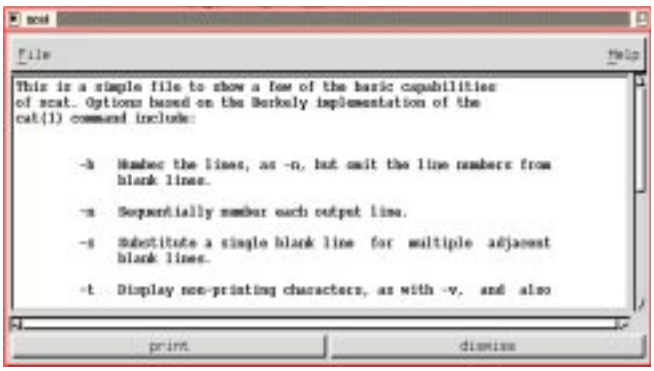

Figure 6

Either pulldown or any of the functions contained within may be accessed via the accelerator keys associated with them. The accelerator for the ***File*** pulldown is **MAlt<F>** and for the ***Help*** pulldown it is **MAlt<H>**, where **MAlt** refers to the Meta or Alt key, depending on the keyboard in use. Figure 7 shows how the ***File*** pulldown would appear on an **xcat** invocation which was started without a menubar. The accelerator functions for the individual functions contained in each pulldown are shown to the right of each pulldown button. For example, the ***Save*** function may be accessed at any time via the key sequence **MCtrl<S>**. This capability to access these functions even without the menubar enhances **xcat**'s functionality as a tool to be used by other applications.

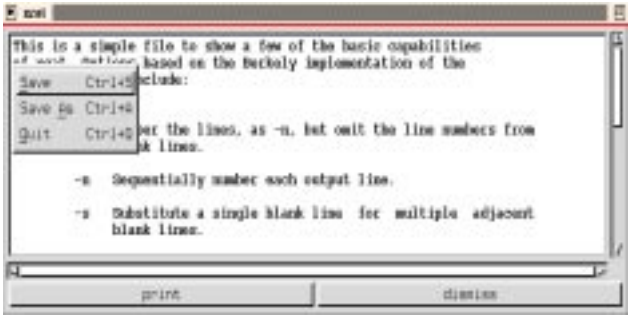

Figure 7

Processing Highlighted Text

There is also another main panel button which may be used for performing operations on highlighted text. The label for this button is set via the -***xl*** command line option or **xLabel*** resource. The actual command may be specified via -***xc*** option or a **xCmd*** resource specification. If no special instruction is given, highlighted text from the display window is appended to the specified command and the resultant command is executed upon selection of the middle button in the resulting main panel. This behavior may be altered by the use of the special character **%** within the specified command. To illustrate this, Figure 8 shows the results of the command*:*

xcat -title *Directories* ***-xl*** *'list files'* ***-xc "xcat*** **|**
-title *'Files in % Directory'* **|**
-e 'find % -type f -exec ls -ld {} |;' " **|**
-e 'find . -type d -exec ls -ld {} |;'.

The first **xcat** will list the directories under the current directory and the second **xcat** which will be run as a child will show the files under the highlighted directory. In the case shown, the ***sun4*** directory has been highlighted in the directory list and the button labelled ***list files*** pressed**.** The child **xcat** then shows a list of the files within the directory.

Note that the title specified for the child instance of **xcat** also contains a dynamic window title containing the specified directory's name. This begins to show the capabilities of xcat as a tool rather than a simple display window. By taking the example shown above one step further and putting in an option to run an xterm with a vi editing session on the picked file, we have built a simple type of file manager:

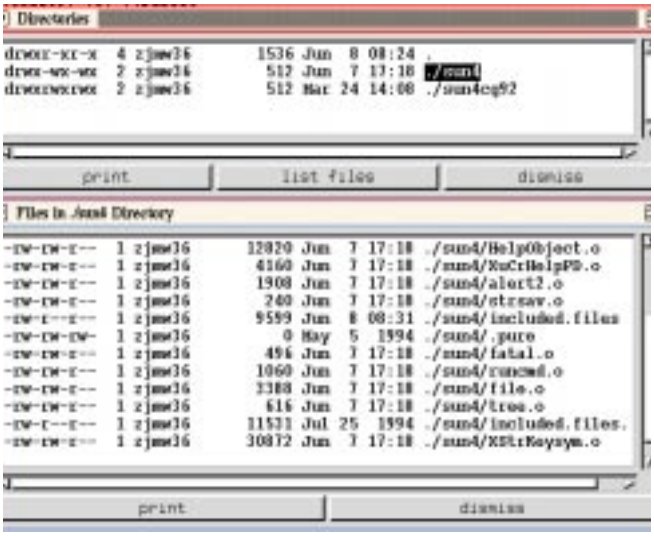

Figure 8

xcat -title *Directories* ***-xl*** *'list files'* ***-xc*** **|**
"***xcat -title*** 'Files in % Directory' ***-xc 'xterm*** \
-e vi ' -xl 'edit file' ***-e 'find % -type f*** \
-exec ls -ld {} |;' " -e 'find . -type d **|**
-exec ls -ld {} |;'

The results are shown in Figure 9. In both examples, note the use of single quotes on the command line is to get around interpretation by the current shell of special characters such as **\,** which is used in the **find** command**.**

Figure 9

Application Interfaces to Xcat

For the basic user community, the capability of **xcat** to work in conjunction with other programs may hold little interest. However, by widespread use by application programs, the users benefit from a more standard interface through familiar functionality. Programs may write text to an **xcat** window by executing **popen**(2) on an **xcat** process, opening it for writing, and then just writing data to the file stream which was returned. The **xcat** application will even inherit the X11 class name of the calling program by the use of the -***C*** command line option. By using this, X11 resources such as color, font, etc. will be the same as the main application and transparency to the user may be attained. Standard uses of **xcat** in this manner may be the display of informational messages, file lists from which to pick items with another action assigned via use of the -***xc*** option, etc. Its limits are basically that of our imagination.

Summary

Xcat has attained a reputation among those who use it regularly as the “swiss army knife” of application programs. Its broad range of functions permits it to serve a wide variety of functions, even serving as a type of second-hand utility to other programs.

Users are encouraged to simply try using some of these functions and experiment with the options available. A quick list of the options may be obtained by entering:

xcat -h

A more detailed manual page is available on the Sun systems by entering:

man -M */explprod/xaprtools/phase2/man xcat*

The program has been installed on Sun, Cray, SGI, HP, Convex, and IBM AIX platforms under the exploration products area.

Addendum

This application was developed with valuable suggestions and programming help from Steve Farmer and Phil Fincannon. Its evolution from a simple display window to a multi-faceted tool is largely due to their input.

Agreement D-95-2518 is funded by the following BUs: MCBU, OBU, Norway-Expl., Egypt, Egypt-Expl., UK-Expl., Trinidad, Netherlands, Netherlands-Expl., Canada Northwest, Canada Exploration, E/USA, LASA, MEA

Geoscience Technology
Joe M. Wade
jwade@amoco.com
SOCON 422-4387.
or 918-660-4387.