

SANDIA REPORT

SAND94-2391 • UC-705

Unlimited Release

Printed April 1995

xdamp: An IDL[®]-based Data Manipulation Program

William P. Ballard

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A04
Microfiche copy: A01

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

SAND94-2391
Unlimited Release

Distribution
Category UC-705

xdamp: An IDL®-based Data Manipulation Program

Printed April 1995

William P. Ballard

**Simulation Technology Research Department
Sandia National Laboratories, Albuquerque NM 87185**

ABSTRACT

The original DAMP (DAta Manipulation Program) was written by Mark Hedemann of Sandia National Laboratories and used the CA-DISSPLA™ (available from Computer Associates International, Inc., Garden City, NY) graphics package as its engine. It was used to plot, modify, and otherwise manipulate the one-dimensional data waveforms (data vs. time) from a wide variety of accelerators. With the waning of CA-DISSPLA and the increasing popularity of UNIX®-based workstations, a replacement was needed. This package uses the IDL® software, available from Research Systems Incorporated in Boulder, Colorado, as the engine, and creates a set of widgets to manipulate the data in a manner similar to the original DAMP. IDL is currently supported on a wide variety of UNIX platforms such as IBM® workstations, Hewlett Packard workstations, SUN® workstations, Microsoft® Windows™ computers, Macintosh® computers and Digital Equipment Corporation VMS® systems. Thus, this program should be portable across many platforms. We have verified operation, albeit with some IDL bugs, on IBM UNIX platforms, DEC Alpha systems, HP 9000/700 series workstations, and Macintosh computers, both regular and PowerPC™ versions.

Table of Contents

List of Figures	6
List of Tables.....	6
Introduction.....	7
PHILOSOPHY	7
GETTING STARTED.....	8
INSTALLATION	8
RESTRICTIONS	8
INITIALIZATION FILE.....	8
FINISHING UP.....	9
Screen Layout	11
File Menu.....	13
FILE.OPEN.....	13
FILE.CLOSE.....	13
FILE.APPEND.....	14
FILE.GET ARRAY	14
FILE.SAVE	14
FILE.SAVE AS	14
FILE.SAVE ASCII.....	15
FILE.SET FILTER.....	15
FILE.SET DATA TYPE.....	15
FILE.EXIT	15
Edit Menu.....	16
EDIT.STORE	16
EDIT.NEW	16
EDIT.COPY	16
EDIT.RENAME	16
EDIT.SELECT	16
EDIT.DELETE.....	17
EDIT.KEEP	17
EDIT.RESTORE	17
EDIT.NOTES	17
EDIT.FILE NOTES.....	17
EDIT.TITLE.....	17
Print Menu	18
PRINT.SCREEN	18
PRINT.ALL.....	18
PRINT.SELECTED	18
PRINT.SELECTED	18

PRINT.SUMMARY	18
PRINT.HEADER	18
PRINT.NOTES	18
PRINT.FILE NOTES	19
PRINT.PRINTER TYPE.....	19
PRINT.QUEUE OUTPUT	19
PRINT.PRINTER RESET	19
Options Window	20
CURSOR	20
AUTO PLOT	21
Clear Stack After Operations	21
POSTSCRIPT	21
ENCAPSULATED.....	21
PCL	21
PLOT #/PAGE	21
SHOT DATE.....	21
TODAYS DATE.....	21
SYMBOLS.....	22
GRID TYPE.....	22
LINE THICKNESS	22
HARDWARE FONT	22
VECTOR FONT	22
XFONT	22
GRAPHICS FONT SIZE.....	22
MAX/MIN	23
FWHM	23
RISE/FALL.....	23
BEST GUESS	24
PEAK DOWN.....	24
ENDS IN.....	24
FILE FILTER.....	24
DATA FORMAT	24
LINEAR.....	24
LOG	25
ROUNDED	25
EXACT	25
EXTENDED	25
SUPPRESS 0	25
MINOR TICKS AUTO.....	25
MINOR TICKS OFF	25
Tools Menu	26
TOOLS.MACRO	26

TOOLS.SHOW REGISTERS	26
TOOLS.SHOW LIMITS	27
TOOLS.SHOW NOTES	27
TOOLS.SHOW FILE NOTES	27
TOOLS.SHOW HEADER	27
TOOLS.XFONT	27
TOOLS.CALCULATOR	27
TOOLS.COLOR TABLE	27
TOOLS.PALETTE	28
TOOLS.XMANAGERTOOL	28
Help Window	29
Operators	30
OPERATOR OVERLOADING	30
ADD	31
SUBTRACT	32
MULTIPLY	32
DIVIDE	32
ABS	32
AVERAGE	32
EXPONENTS	32
POWER	32
INVERSE	33
SQRT	33
EXP	33
LN	33
PWR10	33
LOG10	33
YUNITS	33
CEILING	33
FLOOR	33
TRUNCATE BEFORE	33
TRUNCATE AFTER	33
TIMESHIFT	33
BASELINE	33
COMPARE	34
NORMALIZE	34
INTEGRATE	35
DIFFERENTIATE	35
TRANSFORMS	35
FFT	35
IFFT	36
CONVOLVE	36

CROSS CORRELATE.....	36
FILTERS	36
CABLES	37
CABLE COMPENSATE.....	37
DEDROOP	38
DEFIDU.....	38
GENERATE COMPENSATOR	38
MISCELLANY	38
INTERPOLATE.....	38
SPECTRUM	39
TIMEFLIP	39
CALORIMETER.....	39
Speed Buttons	40
ENTER.....	40
CLEAR	40
SELECT	40
ALL.....	40
PLOT.....	40
ZOOM.....	40
PRT SCRN	41
Waveform Buttons	42
Problems and Troubleshooting	43
Macintosh Specific Issues.....	45
HDF Data File Organization.....	46
HDF read and write IDL procedures	47
READ HDF FILE PROCEDURE	47
WRITE HDF FILE PROCEDURE.....	51
References.....	54
List of Distribution Files.....	55

List of Figures

Figure 1	xdamp Screen Layout	11
Figure 2	UNIX Pickfile Widget.....	13
Figure 3	Options Window Layout.....	20
Figure 4	Xfont widget	23
Figure 5	Help window example	29
Figure 6	EXPONENTS window	32
Figure 7	TRANSFORMS subwindow	35
Figure 8	FILTERS input window	37
Figure 9	CABLES subwindow.....	37
Figure 10	MISCELLANY subwindow	38

List of Tables

Table 1	xdamp.ini Initialization File Commands	9
Table 2	Operations that can use the cursor	20
Table 3	Automatic Register Names	30
Table 4	Register Names and Contents from COMPARE Operation .	34

PHILOSOPHY*Introduction*

xdamp relies, in as many places as possible, on IDL maintained routines. This choice should allow *xdamp* to age gracefully as Research Systems, Inc. will do most of the updating of the operations that actually modify the internal data. Nearly all of the procedures are precompiled when *xdamp* is started because the IDL macro programming language is interpreted at run time. Precompiling causes *xdamp* to pause while nearly everything is compiled but results in faster execution times when manipulating data.

xdamp loads all the data into a set of working waveforms and then closes the data file. It creates a button for each waveform and all manipulations can be done by pushing waveform buttons, followed by operator buttons. The original contents of the file are unchanged until you actively overwrite the original file. When attempting to overwrite a file, you are asked if you are sure you want to do this. However, *xdamp* minimizes how often it asks for confirmation of an action.

xdamp only saves data using the Hierarchical Data Format (HDF) developed by the National Center for Supercomputing Applications (NCSA). IDL provides all the requisite HDF file handling utilities for all of the computing platforms of interest. This format is already compressed so further data compression techniques are not needed; in fact, they tend to make the data file larger. Full libraries of FORTRAN and C routines are available from NCSA.

Whenever an operation is performed, the results are immediately and automatically plotted out, if the automatic plot feature is set on. Also, the relevant pulse parameters (maximum, minimum, pulse width, rise time, and fall time) are calculated. You can choose whether the stack is maintained between operations or is automatically cleared. *xdamp* keeps an audit trail of the operations performed on each waveform that change its contents. These are displayed whenever a single waveform is graphed and can be viewed and edited at will.

Automatic execution of command (macro) files is supported. This feature allows the user to create new procedures from combinations of existing operations. These files can be nested 10 deep to create complicated operations from a sequence of simple macros. *xdamp* automatically creates a journal file named *xdamp.jnl* in the current directory. The journal file can be copied and edited to create a macro file.

When performing any operation with an inherent order, the ENTER button must be used to define one of the waveforms. Usually this defines the waveform to be overwritten; for instance, when performing a waveform addition. In some instances the ENTER button is used to define the reference waveform (COMPARE, time aligning waveforms, and when generating cable compensators). In the case of inverse fast Fourier transforms (IFFTs), the ENTER button is used to designate the real portion of the waveform to be transformed.

The limits on the number of possible waveforms and their maximum size is relatively arbitrary. These limits can be seen by using the TOOLS.SHOW LIMITS menu selection. The system manager can increase these at will (in the *XDsys_ini.pro* module in the *XDsys_sp.pro* file). Many of the operators create new waveforms, if space permits. If

space does not permit, the original waveform is overwritten unless more than one waveform is generated, in which case the operation fails with an error message. To conserve waveform space, if an operation that creates new waveforms is repeated, and would duplicate some waveform names, the old waveforms are overwritten.

INSTALLATION

The *xdamp* source code is provided in a tar file named *xdampv1.xx* where *xx* is the version number. It should be installed in the directory *\$IDL_PATH/lib/xdamp* using the command "tar -xv -fxdampv1.xx". Be certain the directory has public read permission and that all files in the directory also have this permission. Almost all system-specific information is contained in the file *XDsys_sp.pro*. The system manager must edit the *XDsys_queue.pro* module in this file and fix the print queue command to use the appropriate printer. However, VMS systems use the *XDAMP\$PRINTER* logical for this and don't need editing, just definition of the logical. The limits for the number of waveforms (*maxarrays*), their maximum length (*maxsize*), and the default graphics area (*xysize*, *yysize*) can be changed in the *XDsys_ini* module in this file also. Finally, some customizing of the waveform button area is possible in this file.

RESTRICTIONS

xdamp must use IDL version 3.6 or later; otherwise the long variable and procedure names are not properly recognized. *xdamp* does not yet operate on windows computers because dynamic widgets are not yet supported on this platform. There are also some known Macintosh-specific problems discussed later.

GETTING STARTED

The IDL environment variables must be defined as described in the IDL installation instructions for the specific computer environment. The *xdamp* files must be installed in the (publicly accessible) *\$IDL_PATH/lib/xdamp* directory. Then the user can access *xdamp* simply by initiating IDL (type *idl* at the prompt) and then typing *xdamp* at the first IDL prompt. All subsequent inputs are funneled through the main *xdamp* window. If you install *xdamp* elsewhere, the system manager will need to modify the *!HELP_PATH* variable in the *XD_help* module in the *XDsys_sp.pro* file.

For UNIX systems, you can control the background color of the IDL widgets and the text font used for these widgets (the graphics font is completely independent). The following lines (with sample colors and font) may be inserted in your *Xdefaults* file. Be careful, if you choose too large a font, the graphics area will not be next to the controls and will be too small; if the screen does not appear similar to Fig. 1, then reduce the font size. I have had good success using a bold font for the widget font. Alternatively, reduce the graphics area size using the *GRAPHIC AREA=XSIZEExYSIZE* in the initialization file (see next topic). You can also reserve a block of color indices so that other applications won't destroy the color maps with the last entry in the list below.

```
Idl*background: LightSteelBlue  
Idl*foreground: Black  
Idl*fontList: ROMB12  
Idl.colors: 64 (the period is correct here)
```

INITIALIZATION FILE

There is an optional initialization file that may reside in the subdirectory in which you will run *xdamp*. Its name must be *xdamp.ini*. Different versions may be in different subdirectories. The following commands, one per line, may be in this file and will define the options preferences. Further information on these controls may be found in the Options section (page 20). A sample *xdamp.ini* file resides in *\$IDL_PATH/lib/xdamp*.

You must be exceedingly careful when setting the graphics hardware font to choose a legal definition. If in doubt, leave this blank and use the xfont utility in the OPTIONS window to find a legal font name and then include it in the initialization file.

TABLE 1. *xdamp.ini* Initialization File Commands

Controls this feature	Default	Alternates
Cursor or prompt inputs	Cursor On	Cursor Off
Automatic plotting	Autoplot On	Autoplot Off
Clear Stack After Operation	Clear Stack Off	Clear Stack On
Print device	PostScript	Encapsulated, PCL
Plots/page	One/page	Two/page, Four/page
Date on plots	Todays Date	Shot Date
Plot with symbols	Symbols Off	Symbols ON
Grid appearance	Grid Off	Grid On, Grid Zero
Line thickness	Line thick=1.0	Number >0.2, >5.0
Graphics font type	Hardware Font	Vector Font
Graphics Hardware Font ^a	Gfont=FONT	Font must be fully qualified ^b
Vector font size multiplier ^c	Font size = 1.0	Number >0.2, <5.0
Display max/min information	Max/min On	Max/min Off
Display FWHM information	FWHM On	FWHM Off
Display rise/fall information	Rise/fall On	Rise/fall Off
Pulse parameter method	Best Guess	Peak Down, Ends In
Default file filter	File filter=*.hdf	Any wildcard string
File data encoding	HDF	DAMP, SICDAS, ASCII, IDR
Graphic area dimensions in pixels	Graphic Area=800x750	Can be blank and get default from system specific file
X-axis	Xlinear	Xlog
X-axis endpoints	Xrounded	Xexact, Xextended
X-axis zero in graphics area	Xzero On	Xzero Off
X-axis minor tick marks	Xminor Auto	Xminor Off
Y-axis	Ylinear	Ylog
Y-axis endpoints	Yrounded	Yexact, Yextended
Y-axis zero in graphics area	Yzero On	Yzero Off
Y-axis minor tick marks	Yminor Auto	Yminor Off

a. Valid only if Hardware Font selected

b. Example: -adobe-helvetica-medium-r-normal-14-140-75-75-p-77*

c. Valid only if Vector Font selected

FINISHING UP

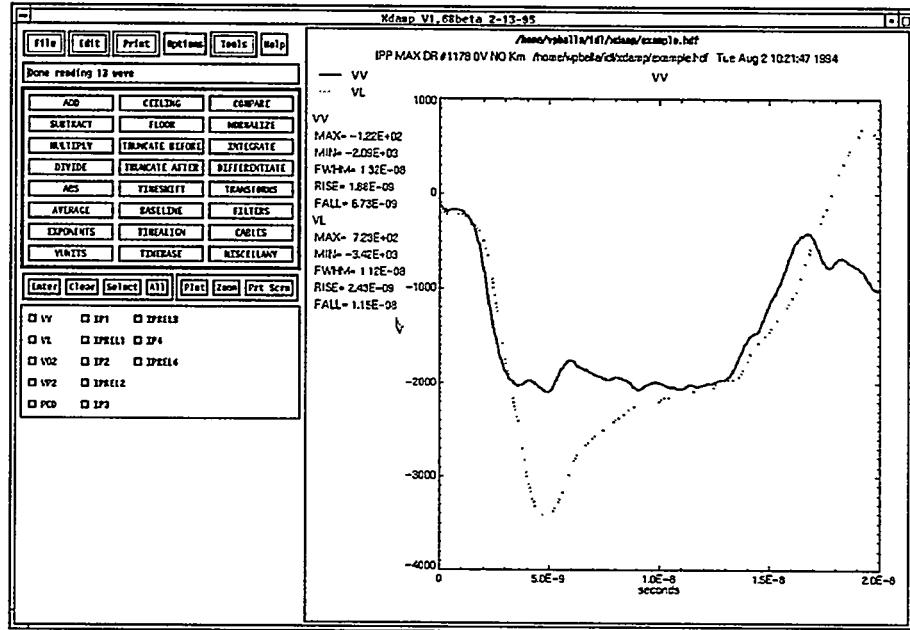
To leave *xdamp*, select FILE.EXIT from the *xdamp* menu. This action returns you to the IDL prompt. Then type EXIT to return to the system prompt. On a Macintosh, *DO NOT*

use the FILE.QUIT on the screen menu bar because this will not properly close the journal file or any other open files.

Screen Layout

The *xdamp* screen (Fig. 1) is designed so that input from a virtual terminal is never needed. The *xdamp* title bar displays the current version number and the date of that version. The left portion of the screen contains all the menus and buttons needed. The right portion of the screen contains the graphics region and a label area for the active file name.

FIGURE 1. *xdamp* Screen Layout



Across the top of the left portion of the *xdamp* screen is a series of "menu" buttons, not all of which actually lead to menus. The FILE, EDIT, PRINT, and TOOL buttons lead to menus, the OPTIONS button pops a window with available option selections, and the HELP button pops a widget that allows you to obtain help on any aspect of IDL or *xdamp*.

The space immediately below the menu bar contains a message area in which *xdamp* places output. Informational messages, general messages, error messages, and cursor values are shown in this area. For the sake of sanity, *xdamp* does not beep at you to alert you that a message is present.

Below the message area is the operator box. Each button causes the operation described to be performed. In a few cases, such as the TRANSFORM button, subsets of further buttons appear. Only one operator button can be pressed at a time.

Below the operator area is the speed button area where buttons used extremely frequently are placed. In most cases, the same operation executed by pressing one of these buttons is available from one of the menus.

Below the speed button area are the waveform buttons. One button is present for each of the waveforms. Multiple waveforms can be selected simply by pressing multiple buttons. Pressing an already set button deselects that waveform. Buttons may also be selected using wildcard strings by using the EDIT.SELECT menu selection or the Select speed button.

The following sections describe each of these actions in detail, beginning with the menus and proceeding down the left side of the screen. For the command equivalent, inputs that you choose are shown enclosed by the brackets <>. *The punctuation in the command equivalent is critical.* Also, although the command equivalents are shown with varying cases, the actual commands are converted to uppercase so the commands are case insensitive except for waveform names and file names. The capitalization comes from what appears in the xdamp.jnl journal file.

File Menu

FILE.OPEN

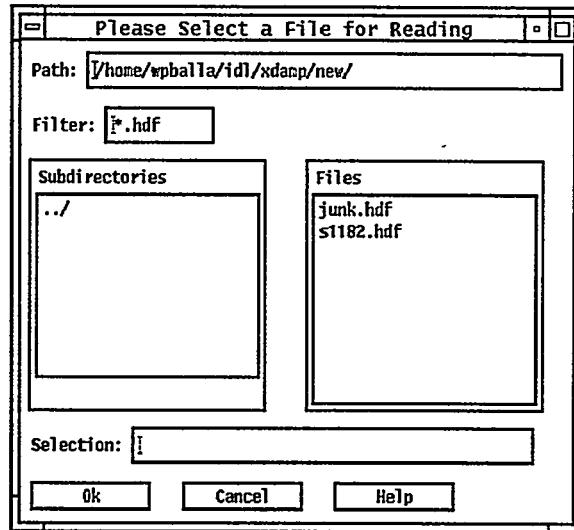
The FILE.OPEN menu selection pops a “pickfile” widget (Fig. 2) to allow you to choose the data file to open. The default extension is defined by the case sensitive file filter defined in the initialization file or by the FILE.SET FILTER menu selection. It may be temporarily edited within the pickfile widget. The widget allows you to move relatively freely between various directory tree structures. *Beware, only single mouse clicks are needed to move around on UNIX systems.* You should set the data storage type using the FILE.DATA TYPE menu item or the equivalent in the OPTIONS widget. This command automatically discerns if the data file was written in HDF, and reads in the data appropriately. DAMP files must be in the latest variable record length format. When opening or saving a DAMP file, it is assumed that the header file exists with the same name but with extension .list. If this is not true, you are prompted for the header filename with the pickfile widget. If records longer than the current maximum allowed are present, then the data are averaged over sets of points to reduce the record size. SIC-DAS data files from the SPHINX accelerator are in ASCII format but have the header information in the file and use a format that saves the starting and ending times of the waveforms, rather than the entire time data set. ASCII input data are in the format: name, horizontal axis units, vertical axis units, number of points and a set of x-y pairs, delimited by white space. The ASCII data need not be in order or have equally spaced points, *xdamp* will sort and then resample the waveform using the smallest time interval. In automatic execution mode, you will *not* be queried about whether to save a currently open file when opening a new file.

Command Equivalent

FILE.OPEN

<full path + file name>

FIGURE 2. UNIX Pickfile Widget



FILE.CLOSE

The FILE.CLOSE menu selection will close the active data set. If any modifications to the original data have been made, then you will be asked if you wish to save the data set

before the waveforms are cleared out. If you answer yes, then the SAVE AS submenu will appear to obtain a file name for saving the data. If you have already saved the file, then the NO in the command equivalent is not needed.

Command Equivalent

FILE CLOSE

NO

FILE.APPEND

The FILE.APPEND menu selection also uses the pickfile widget to choose a file to be appended to the currently opened file. Each waveform in the appended file will have the shot number from the file name (if any) prefixed to the waveform name. If there is no shot number, then one or more ampersands (&) are prefixed to the waveform name. The number of ampersands is determined by the number of append operations performed. The addition of the shot number or ampersands to each file name makes certain that the names are unique. When saving a file after an append operation, the default file name is still that of the initial file opened.

Command Equivalent

FILE APPEND

<full file name>

FILE.GET ARRAY

The FILE.GET ARRAY menu selection uses the pickfile widget to choose a data file. This file is opened and a widget with buttons for each waveform is presented. You choose which waveforms (which are automatically plotted) you want to append to the currently opened data set and then press the DONE button. The waveform names will either have the shot number from the file name prefixed to the waveform names or a number of ampersands (&) that depends on the number of files opened. These waveforms are added to the current data set. Do not use this command if the target file has only one waveform, instead use FILE.APPEND. The command equivalent is shown below for adding two waveforms "WVFM1" and "WVFM2".

Command Equivalent

FILE.GET ARRAY

<full file name>

WAVEFORM.WVFM1

WAVEFORM.WVFM2

DONE

FILE.SAVE

The FILE.SAVE menu selection overwrites the original data file with the current contents of the waveforms using the Hierarchical Data Format (HDF). If you have performed an append operation, the name of the first file opened is used. You are asked if you are certain that you want to overwrite the original file before continuing with this operation *unless* you are in automatic execution mode.

Command Equivalent

FILE SAVE

FILE.SAVE AS

The FILE.SAVE AS menu selection prompts for a new file name via the pickfile widget so you can alter the directory, file name, or file extension before actually saving the waveforms using Hierarchical Data Format (HDF). The default file extension for a save

as operation is .hdf. If the new and old file names are identical, you are asked if you are certain that you want to overwrite the original file before continuing with this operation.

Command Equivalent

FILE.SAVE AS

<full file name>

FILE.SAVE ASCII

The FILE.SAVE ASCII menu selection saves the selected waveforms in an x-y pair ASCII file suitable for editing and use by many external graphics packages. The output data format is the name, the horizontal axis units label, the vertical axis units label, the number of points, and then a series of x-y pairs separated by white space. If more than one waveform is selected, all waveforms are output in a single file. If no waveforms are selected, then all waveforms are written to the output file. The output file name is chosen by the pickfile widget with a default extension of .asc.

Command Equivalent

FILE.SAVE ASCII

<full file name>

FILE.SET FILTER

The FILE.SET FILTER menu selection asks for the new case sensitive file filter to be applied in future OPEN, SAVE, and SAVE AS operations. The file filter can be very sophisticated and use multiple wildcard characters (*) or simply specify the desired file extension. Example formats are *.hdf or saturn_023*.dat.

Command Equivalent

FILE.SET FILTER

<filter string>

FILE.SET DATA TYPE

The FILE.SET DATA TYPE menu selection pops a widget to set the format the data were stored in. Possible responses are HDF, DAMP, SICDAS, or ASCII formats as well as a CANCEL button if you don't want to change the current selection. File save operations always use the HDF format, no matter what data type is set.

Command Equivalent

FILE.SET DATA TYPE

HDF or DAMP or SICDAS or ASCII

FILE.EXIT

The FILE.EXIT menu selection exits *xdamp*. Prior to exiting, a FILE.CLOSE operation is performed, allowing you to save the current data set if you desire. Then, if any data remains in the print file, you are asked if you wish to spool it to the printer before exiting. This operation returns you to the IDL prompt, not the operating system. In command mode, the following sequence assumes you do not wish to save the active file. Also, an additional Yes or No may be needed if there is a print file open. The print question is asked before the question about saving the file.

Command Equivalent

FILE.EXIT

YES or NO

Edit Menu

In this context, all editing is done on waveforms or a set of special storage registers. There are two methods to create new waveforms, NEW and COPY. These are very similar in end effect. There is no method to create a new register; you must use the pre-defined registers #R0 through #R9. Be aware that *xdamp* is case sensitive for waveform names so the names VV Vv, vV, and vv are unique.

EDIT.STORE

The EDIT.STORE menu selection allows you to store a number in one of the pre-defined registers. A constant or the contents of another register can be stored in these registers.

Command Equivalent

EDIT.STORE
<value or #register>
<0 through 9>

EDIT.NEW

The EDIT.NEW menu selection creates a new waveform. It prompts for the desired name of the new waveform. This name must be different from all other names in the current data base. All other data about the waveform is assumed to be blank until some operation targeting this waveform is performed. Then, the new waveform takes its properties (number of points, time span) from the other waveform in the operation.

Command Equivalent

EDIT.NEW
<waveform name>

EDIT.COPY

The EDIT.COPY menu selection copies the last waveform selected to a new waveform. You are prompted for the new waveform name. If the new name is identical to an existing one, you will be asked if you are certain you wish to overwrite the old waveform. Should you answer affirmatively, the copy command will proceed, writing over the old data with the selected waveform. All other data are identical to the old waveform.

Command Equivalent

EDIT.COPY
<waveform name>

EDIT.RENAME

The EDIT.RENAME menu selection renames the last waveform selected to a new name obtained from a prompt window. The new name is forced to be unique compared to all current waveforms or the operation is not performed.

Command Equivalent

EDIT.RENAME
<waveform name>

EDIT.SELECT

The EDIT.SELECT menu selection asks for a wild card waveform designation and then selects all of the waveforms meeting that criteria. At least one asterisk must be present in the response. For example, PIN* as the wild card input would select PIN, PIN1, and PIN007. If more than 8 waveforms are selected, then they will NOT be automatically plotted. If 8 or fewer are chosen, autoplot operates normally. This feature prevents plot-

ting, which is slow, when you are preparing to delete a large number of waveforms. The PLOT button still operates normally if you do wish a plot of the selected waveforms.

Command Equivalent

EDIT.SELECT

<wildcard string>

EDIT.DELETE

The EDIT.DELETE menu selection deletes all the selected waveforms and their buttons. There is no "Are you sure?" prompt.

Command Equivalent

EDIT.DELETE

EDIT.KEEP

The EDIT.KEEP menu selection deletes all waveforms except the selected waveforms. There is no "Are you sure?" prompt.

Command Equivalent

EDIT.KEEP

EDIT.RESTORE

The EDIT.RESTORE menu selection restores all the currently selected waveforms from the data file to their state after the last save operation.

Command Equivalent

EDIT.RESTORE

EDIT.NOTES

The EDIT.NOTE menu selection allows you to edit notes attached to each waveform. (Notes appear in the lower left corner of the graphics area if a single waveform is plotted.) *xdamp* automatically uses the notes for creating an audit trail of operations performed on every waveform. Editing areas for notes about all of the selected waveforms, or all waveforms if none are selected, will appear. In automatic execution, this command returns control to you for the input of notes. Automatic execution continues when you press the DONE button.

Command Equivalent

EDIT.NOTES

EDIT.FILE NOTES

The EDIT.FILE NOTES menu selection allows you to edit notes attached to the overall data set. An editing area for the notes will appear. In automatic execution, this command returns control to you for the input of file notes. Automatic execution continues when you press the DONE button.

Command Equivalent

EDIT.FILE NOTES

EDIT.TITLE

The EDIT.TITLE menu selection allows you to edit the shot title information attached to the overall data set. An editing area for the title will appear. In automatic execution, this command returns control to you for the input of the shot title. Automatic execution continues when you press the RETURN key or DONE button.

Command Equivalent

EDIT.TITLE

Print Menu

All printing is done using PostScript, Encapsulated PostScript, or PCL and goes to a temporary file named *xdamp.out* in your current directory. To minimize paper, all printing occurs in append mode until you request that the file be queued. *xdamp* overrides the font selections when using PostScript output to be certain output will fit on the page.

PRINT.SCREEN

The PRINT.SCREEN menu selection causes the current graphics view to be redrawn to a file that can be printed later (see PRINT.PRINTER QUEUE).

Command Equivalent

PRINT.SCREEN**PRINT.ALL**

The PRINT.ALL menu selection causes all of the waveforms to be drawn to a file for later printing using the current selection of one, two, or four plots per page.

Command Equivalent

PRINT.ALL**PRINT.SELECTED**

The PRINT.SELECTED menu selection causes all of the selected waveforms to be drawn to a file for later printing. The number of plots per page selection is honored for this process but only one plot to a plot area.

Command Equivalent

PRINT.SELECTED**PRINT.REGISTERS**

The PRINT.REGISTERS menu selection causes the contents of the registers to be sent to the print file.

Command Equivalent

PRINT.REGISTERS**PRINT.SUMMARY**

The PRINT.SUMMARY menu selection places a summary sheet in the file. The summary sheet contains the waveform name, maximum, minimum, vertical units, full width at half maximum, rise time, fall time and time units for each selected waveform. If no waveforms are selected, the information for all waveforms is printed. This command also creates an ASCII file called *xdamp.ss* suitable for inclusion in a spreadsheet using single spaces as delimiters.

Command Equivalent

PRINT.SUMMARY**PRINT.HEADER**

The PRINT.HEADER menu places the header (if one exists) of the original file opened, in the file for later printing.

Command Equivalent

PRINT.HEADER**PRINT.NOTES**

The PRINT.NOTES menu places the notes for all the selected waveforms in the file for later printing. If no waveforms are selected, then all of the notes are printed.

Command Equivalent

PRINT.NOTES

PRINT.FILE NOTES

The PRINT.FILE NOTES menu places the file notes in the printer file for later printing.

Command Equivalent

PRINT.FILE NOTES**PRINT.PRINTER TYPE**

The PRINT.PRINTER TYPE menu selection pops a selection widget to choose the type of printer you have. You can either have PostScript, Encapsulated PostScript, or PCL type printers as well as a Cancel button if the current selection is adequate. This can also be changed in the OPTIONS widget.

Command Equivalent

PRINT.PRINTER TYPE**POSTSCRIPT or ENCAPSULATED or PCL****PRINT.QUEUE OUTPUT**

The PRINT.QUEUE OUTPUT menu selection closes the current print file and queues it to the line printer for UNIX systems. For Macintosh computers, you need to print out the *xdamp.out* file manually. One way to accomplish this is to use the shareware program DropPs available from *mac.archive.umich.edu* via anonymous ftp.

Command Equivalent

PRINT.QUEUE OUTPUT**PRINT.PRINTER RESET**

The PRINT.PRINTER RESET menu selection closes the current print file and does not queue it to the printer. The second argument in the command equivalent is only required if there is an uncleared print file in the queue.

Command Equivalent

PRINT.PRINTER RESET**YES or NO**

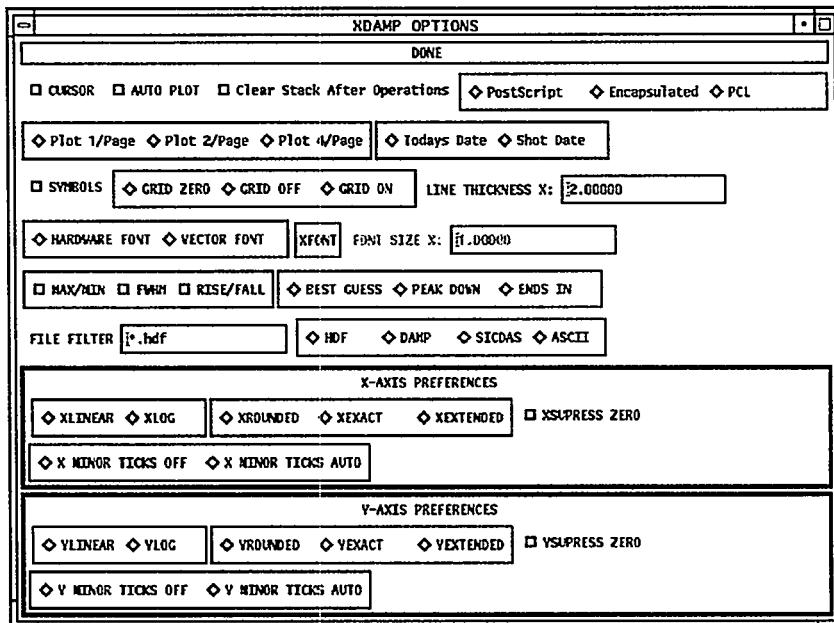
Options Window

Unlike the prior menus, the OPTIONS menu button pops up a window (Fig. 3) with a series of push buttons and input boxes. These allow you to select your preferred mode of viewing the data. All of the OPTIONS options may be set in the *xdamp.ini* file. The x- and y-axis preferences selections are identical and are only discussed under the equivalent x-axis heading. Multiple OPTIONS options may be placed in a series with the block of commands surrounded by the following commands. However, to perform even a single options command, you must surround it by these.

Command Equivalent

```
OPTIONS
Other commands here
DONE
```

FIGURE 3. Options Window Layout



CURSOR

The CURSOR preference button determines how inputs will be given to *xdamp* for truncation, time shifting, zooming, and baselining operations. If the cursor button is down it is ON and the cursor will become active for these inputs. The current cursor coordinates are shown continuously in the message window. You must press a mouse button to pick the final desired cursor-input position. If cursor mode is not selected, a numeric input box will automatically appear. Only a few operators, shown in Table 2, use the cursor.

TABLE 2. Operations that can use the cursor

```
BASELINE
TIMESHIFT
CEILING
```

TABLE 2. Operations that can use the cursor

FLOOR
TRUNCATE BEFORE
TRUNCATE AFTER
DEFIDU
CALORIMETER

The CURSOR command equivalent toggles the current state. In command mode, if the cursor preference is on, the automatic execution pauses for cursor input. If the cursor preference is off, the cursor position was saved (in data units) and is reused.

Command Equivalent

CURSOR**AUTO PLOT**

The AUTO PLOT preference button determines whether selected waveforms will be immediately plotted, or if the PLOT button must be used to make a plot. This is a toggle button. The command equivalent to toggle the state is shown below.

Command Equivalent

AUTO PLOT**Clear Stack After Operations**

The Clear Stack after Operations preference button determines whether the stack will be automatically cleared after each operation. This is a toggle button. The command equivalent to toggle the state is shown below.

Command Equivalent

CLEAR STACK**POSTSCRIPT
ENCAPSULATED
PCL**

The POSTSCRIPT, ENCAPSULATED, and PCL buttons are mutually exclusive. These determine what type of output device is in use. PCL mode creates (large) bitmap files.

Command Equivalent

POSTSCRIPT or ENCAPSULATED or PCL**PLOT #/PAGE**

The PLOT 1/PAGE, PLOT 2/PAGE, and PLOT 4/PAGE buttons are mutually exclusive. These determine whether one, two, or four plots per page will be shown both on the screen and for hardcopies.

Command Equivalent

PLOT 1/PAGE or PLOT 2/PAGE or PLOT 4/PAGE**SHOT DATE
TODAYS DATE**

The SHOT DATE and TODAYS DATE buttons are mutually exclusive. If a shot date was encoded in the data file, then that date and time are shown on each plot when SHOT DATE is selected. Otherwise, if TODAYS DATE is selected, the current date is placed on each plot.

Command Equivalent

SHOT DATE or TODAYS DATE

SYMBOLS

The **SYMBOLS** button determines whether symbols will be used on each plot in addition to the line. If it is pressed, then symbols will be used. In command mode, this is a toggle command.

Command Equivalent

SYMBOLS**GRID TYPE**

The **GRID ZERO**, **GRID OFF** and **GRID ON** buttons are mutually exclusive. If **GRID OFF** is selected, no grid is shown. If **GRID ON** is selected, then a dotted grid is shown at the major tick marks. If **GRID ZERO** is selected, then only the x and y axis zero values of the grid are drawn as bold lines.

Command Equivalent

GRID ON or GRID OFF or GRID ZERO**LINE THICKNESS**

The **LINE THICKNESS MULTIPLIER** determines how thick the graphics lines will be. The default is 1.0 and making lines too thick significantly slows down the graphics response. Valid values are between 0.2 and 5.0. The comma in the command equivalent is important.

Command Equivalent

LINE THICKNESS, <Line Thickness Multiplier>**HARDWARE FONT**
VECTOR FONT

The **VECTOR FONT** button selects the IDL Hershey vector fonts for the graphics area. The **HARDWARE FONT** button selects the terminal default as specified in the .Xdefaults file. The hardware fonts can be changed through either the **XFONT** button or the **TOOLS.XFONT** menu item.

Command Equivalent

VECTOR FONT or HARDWARE FONT**GRAPHICS FONT SIZE**

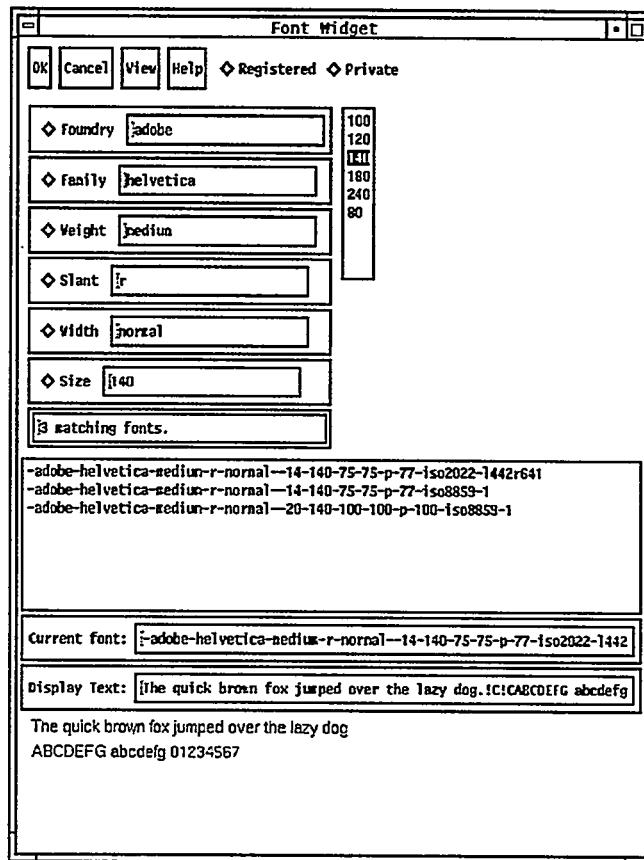
The **GRAPHICS FONT SIZE MULTIPLIER** is only active for vector fonts. It modifies all the text in the graphics window. Valid values are 0.2 to 5.0. If you have selected hardware fonts, this field is inactive and the font size multiplier is set to 1.0. The comma in the command equivalent is important.

Command Equivalent

FONT SIZE, **XFONT**

The **XFONT** button pops up the xfont selector widget (Fig. 4) if **HARDWARE FONT** has been selected. This widget allows you to preview and select a hardware font from all of those available to your terminal. This font becomes the graphics area font. This application has no command equivalent but a desired font can be placed in the **xdamp.ini** file. This application does not function for Macintosh or Windows computers.

FIGURE 4. Xfont widget

**MAX/MIN**

The MAX/MIN toggle button selects whether the maximum and minimum values of the selected waveforms will be shown on the left side of each plot (in ONE/page mode). This button is not mutually exclusive with either the FWHM or RISE/FALL buttons. These quantities are calculated every time a waveform is plotted, whether or not MAX/MIN is selected, and are stored in registers.

Command Equivalent**MAX/MIN****FWHM**

The FWHM toggle button selects whether the full-width at half-maximum (FWHM) values of the selected waveforms will be shown on the left side of each plot (in ONE/page mode). This button is not mutually exclusive with either the MAX/MIN or RISE/FALL buttons. This quantity is calculated every time a waveform is plotted, whether or not FWHM is selected, and is stored in a register.

Command Equivalent**FWHM****RISE/FALL**

The RISE/FALL toggle button selects whether the 10%-to-90% rise-time and fall-time values of the selected waveforms will be shown on the left side of each plot (in ONE/

page mode). This button is not mutually exclusive with either the MAX/MIN or FWHM buttons. These quantities are calculated every time a waveform is plotted, whether or not RISE/FALL is selected, and are stored in registers.

Command Equivalent

RISE/FALL

BEST GUESS

The BEST GUESS button is mutually exclusive with the PEAK DOWN and ENDS IN buttons. It selects the mode used to calculate the FWHM and rise and fall times. The algorithm used calculates all parameters using both the PEAK DOWN and ENDS IN techniques. The longest FWHM and the shortest rise and fall times are then selected.

Command Equivalent

BEST GUESS

PEAK DOWN

The PEAK DOWN button is mutually exclusive with the BEST GUESS and ENDS IN buttons. It selects the mode used to calculate the FWHM and rise and fall times. The algorithm is to calculate all parameters starting from the waveform peak and proceeding outward to find the first 90%, 50%, and 10% values for the calculations.

Command Equivalent

PEAK DOWN

ENDS IN

The ENDS IN button is mutually exclusive with the BEST GUESS and PEAK DOWN buttons. It selects the mode used to calculate the FWHM and rise and fall times. The algorithm is to calculate all parameters starting from the beginning and the end of the waveform to find the first 10%, 50% and 90% values for the calculations.

Command Equivalent

ENDS IN

FILE FILTER

The FILE FILTER entry box allows you to change the default file filter to be applied in future OPEN, SAVE, and SAVE AS operations. This is identical to changing the filter in the FILE.SET FILTER menu selection. The file filter can be very sophisticated and use multiple wild card characters (*) or simply specify the desired file extension. Example formats are *.hdf or saturn_012*.dat.

Command Equivalent

FILE FILTER, <filter string>

DATA FORMAT

The DATA FORMAT selection buttons allow you to change the default data structure for the input file. This is identical to changing the selection in the FILE.SET DATA TYPE menu selection. Whatever the selection is, HDF files are always automatically recognized for reading.

Command Equivalent

HDF or DAMP or SICDAS or ASCII

LINEAR

The LINEAR button is mutually exclusive with the LOG button. This selects whether the axis will be linear or logarithmic.

Command Equivalent

XLINEAR or YLINEAR

LOG	The LOG button is mutually exclusive with the LINEAR button. This selects whether the axis will be linear or logarithmic. For log axes, negative and zero values are ignored.
	Command Equivalent <u>XLOG or YLOG</u>
ROUNDED	The ROUNDED button is mutually exclusive with the EXACT and EXTENDED buttons. This selects rounded axis limits where IDL chooses the axis limits after examining the waveforms.
	Command Equivalent <u>XROUNDED or YROUNDED</u>
EXACT	The EXACT button is mutually exclusive with the ROUNDED and EXTENDED buttons. This selects exact axis limits and IDL chooses the axis limits after examining the waveforms.
	Command Equivalent <u>XEXACT or YEXACT</u>
EXTENDED	The EXTENDED button is mutually exclusive with the ROUNDED and EXACT buttons. This selects extended axis limits where IDL chooses the axis limits after examining the waveforms and then adds approximately 20% to both ends.
	Command Equivalent <u>XEXTENDED or YEXTENDED</u>
SUPPRESS 0	The SUPPRESS 0 toggle button is not mutually exclusive with the ROUNDED, EXACT and EXTENDED buttons. Normally, IDL will choose to have the value 0 included in the axis limits for both axes. This button removes this restriction and is akin to ac-coupling an oscilloscope.
	Command Equivalent <u>XSUPPRESS ZERO or YSUPPRESS ZERO</u>
MINOR TICKS AUTO	The MINOR TICKS AUTO button is mutually exclusive with the MINOR TICKS OFF button. This enables minor axis tick marks and IDL chooses the spacing after examining the waveforms.
	Command Equivalent <u>X MINOR TICKS AUTO or Y MINOR TICKS AUTO</u>
MINOR TICKS OFF	The MINOR TICKS OFF button is mutually exclusive with the MINOR TICKS AUTO button. This disables minor axis tick marks.
	Command Equivalent <u>X MINOR TICKSOFF or Y MINOR TICKS OFF</u>

Tools Menu

TOOLS.MACRO

The TOOLS.MACRO menu has two sub-menu selections: Single File and Sequence of Files. Both of these modes require a command file (the default extension is .dcf) composed of a sequence of *xdamp* commands. The appropriate command file equivalent is listed at the end of each command description. You can incorporate comments into your command files by placing a semicolon at the beginning of the line. Whitespace lines are ignored and can be inserted to make the command file easier to read but do not end a file with whitespace or comments. The single file version of this command can be used to nest macros up to 10 deep. You cannot recursively call the sequence of files version but single macros can be executed from within a sequence of files macro.

The single-file mode will pop the pickfile widget to select an *xdamp* command file. This command file will then be executed and you will be returned to *xdamp*. The *xdamp.ini* file is not executed and the waveform buttons are left as is. Thus, you can select a waveform and then perform a calculation on that waveform. To add a macro within another macro, use the following command syntax.

Command Equivalent

TOOLS.MACRO.SINGLE FILE
<full file name>

The sequence-of-files selection pops the pickfile widget three times. The first time, you are asked to pick the beginning data file of a sequence, the second time you pick the final data file, and the third time you choose the command file to operate on each of these files. Each file is automatically opened and closed, so you do not need to perform these operations in the command file. However, you must perform an explicit FILE.SAVE operation in the command file or the results will not be permanently saved. The *xdamp.ini* file will be automatically executed prior to opening each file so that a known initial state is present. This option allows an easy way to translate a group of files to HDF format. Simply create a one line command file containing the command FILE.SAVE. Modify your *xdamp.ini* to reflect the data format and extension that you will be translating from. When you perform automatic multiple file execution, the original data files will be overwritten with HDF data files.

You need to be very careful of the state of *xdamp* when you begin an automatic execution. *xdamp* does not clear all of the waveform selection buttons and does not initialize many variables in single execution mode. However, the initialization file is executed for the initial command file of a nested set and for each pass through a sequence of files loop. If you are assuming a particular waveform selection state, it is wise to perform a CLEAR operation in your command file.

TOOLS.SHOW REGISTERS

The TOOLS.SHOW REGISTERS menu selection pops a widget that shows the name and contents of each register. No action is taken and the widget remains visible until you press the DONE button. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

TOOLS.SHOW REGISTERS

TOOLS.SHOW LIMITS

The TOOLS.SHOW LIMITS menu selection pops a widget that displays the current limits on the number and length of waveforms. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

TOOLS.SHOW LIMITS**TOOLS.SHOW NOTES**

The TOOLS.SHOW NOTES menu selection pops a widget that shows the name and contents of each note associated with a waveform. No action is taken and the widget remains visible until you press the DONE button. You cannot edit the notes with this selection, instead use EDIT.NOTES. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

TOOLS.SHOW NOTES**TOOLS.SHOW FILE NOTES**

The TOOLS.SHOW FILE NOTES menu selection pops a widget that shows the file notes. No action is taken and the widget remains visible until you press the DONE button. You cannot edit the file notes with this selection, instead use EDIT.FILE NOTES. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

TOOLS.SHOW FILE NOTES**TOOLS.SHOW HEADER**

The TOOLS.SHOW HEADER menu selection pops a widget that shows the file header information. No action is taken and the widget remains visible until you press the DONE button. You cannot edit the header information with this selection, instead use EDIT.HEADER. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

TOOLS.SHOW HEADER**TOOLS.XFONT**

The TOOLS.XFONT menu selection pops up the xfont selector widget. This widget allows you to preview and select a hardware font from all of those available to your terminal. This font becomes the graphics area font. This application has no command equivalent but can be placed in the *xdamp.ini* file.

TOOLS.CALCULATOR

The TOOLS.CALCULATOR menu selection pops up a calculator widget. The results of all calculations are only visible in the widget window and are not returned to *xdamp*. This application has no command equivalent.

TOOLS.COLOR TABLE

The TOOLS.COLOR TABLE menu selection pops up a widget used to load color tables. Use this application to change to reverse video mode in the graphics area. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

TOOLS.COLOR TABLE

TOOLS.PALETTE

The TOOLS.PALETTE menu selection pops up a widget used to select color palettes. This selection has the command equivalent shown below but you must manually press the DONE button to continue with automatic execution.

Command Equivalent

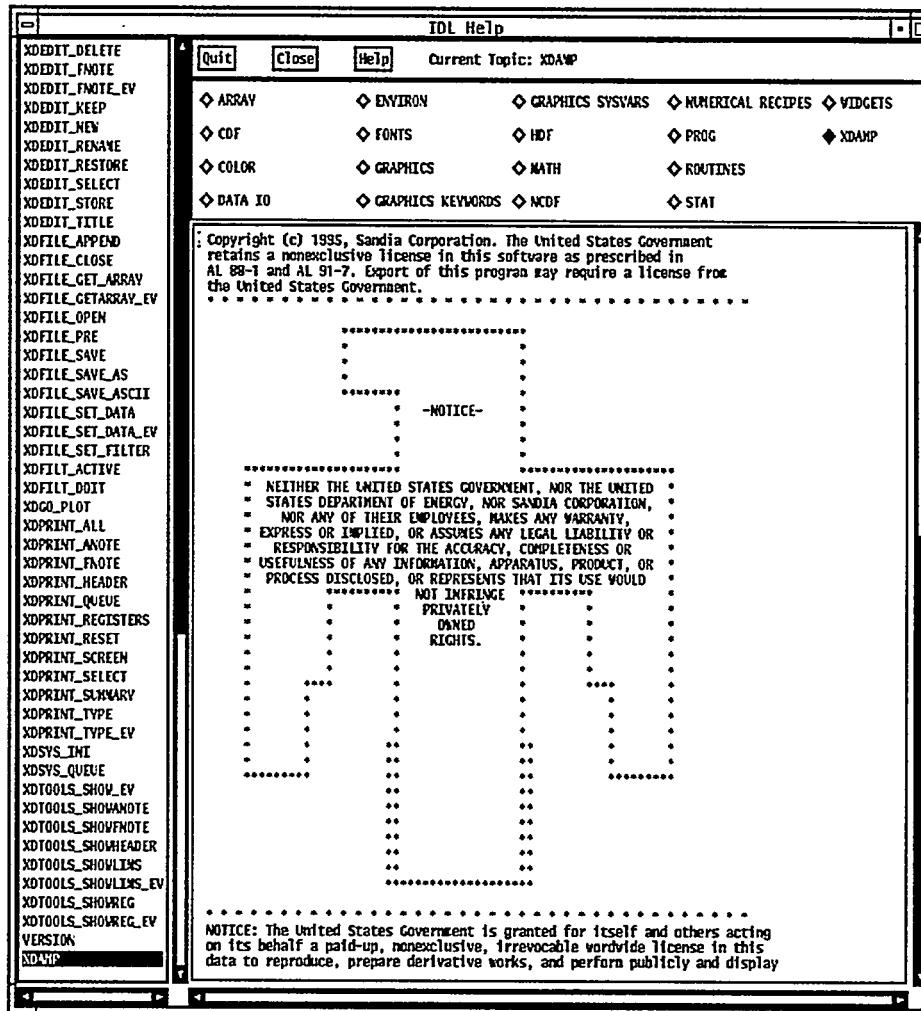
TOOLS.PALETTE**TOOLS.XMANAGERTOOL**

The TOOLS.XMANAGERTOOL menu selection pops up the widget used to investigate what widgets are currently managed by the IDL Xmanager. This application has no command equivalent.

Help Window

Pressing the **HELP** button will activate the IDL help widget (Fig. 5). This is equivalent to typing a question mark at the IDL prompt. By pressing the **xdamp** button in the widget, all comments in the **xdamp** procedures are available for perusal. The **HELP** button is ignored in automatic execution mode.

FIGURE 5. Help window example



Operators

OPERATOR OVERLOADING

There are two types of operators in xdamp; single-waveform operators and multiple-waveform operators. As a consequence, the behavior of the operators is slightly different. For single-waveform operators (i.e. ABS, TIMESHIFT, BASELINE), if multiple waveforms are selected, the same operation is performed on all of the selected waveforms. For multiple-waveform operators, if multiple waveforms are selected, then a constant to operate on all the selected waveforms is requested. If you wish to perform a multiple waveform operation on two waveforms (i.e. add two waveforms), then the first waveform must be selected and then ENTERed. If no waveforms at all have been selected, then it is assumed that you wish to perform register arithmetic on the pre-defined registers #R0 through #R9 for the following operators: ADD, SUBTRACT, MULTIPLY, DIVIDE, ABS, INVERSE, POWER, SQRT, LOG, LN, EXP, 10^A . Thus, many of the operators are triply overloaded and operate on registers, single waveforms, or multiple waveforms in two possible modes.

TABLE 3. Automatic Register Names

Register Name	Quantity
#N	Number of points in waveform
#DT	Δt for waveform
#MIN	Minimum of waveform
#MAX	Maximum of waveform
#PEAK	Peak value of waveform (sign sensitive)
#TATMIN	Time of minimum
#TATMAX	Time of maximum
#TATPEAK	Time of peak
#FWHM	Full-width-at-half-maximum of waveform
#RISE	10%-90% rise time of waveform
#FALL	90%-10% fall time of waveform
#TSHIFT	Time shift from COMPARE
#SCALE	Scale factor from COMPARE
#OFFSET	Offset from COMPARE
#CURSORT	Time from last Cursor operation
#CURSORY	y value of last Cursor operation
INTERCEPT	Intercept from Calorimeter operation
#R0 - #R9	User number registers

When entering a constant to operate on single or multiple waveforms there are two possibilities. You can enter an explicit number in free format, or you can use one of the registers (assuming it is loaded). The registers are all prefixed with the character #. A variety of registers are automatically loaded during each plot operation when the pulse parameters are calculated. These are shown in Table 3. *The values come from the last waveform selected and plotted.* The peak register will be the maximum if the absolute

value of the maximum is greater than the absolute value of the minimum, otherwise it is the minimum value.

All of the single- and multiple-waveform operators have similar command file sequences. The single-waveform operators allow the selection of as many waveforms as desired and then use the command sequence below to execute the command "COMMAND".

Command Equivalent

COMMAND

If the operator requires a constant input, it is provided in the following manner on the next line of the command file. If performing a cursor operation and the cursor preference is on, the command file temporarily becomes interactive. You may wish to dynamically prompt for a numeric input during automatic execution. To do this, replace the <number or #register> in the following table with a question mark followed immediately by a prompt string without quotes. The prompt string will become the title of the input box and you will be prompted for the information to be loaded in the register.

Command Equivalent

<number or #register> or ?PROMPT STRING

Multiple-waveform operators all operate in one of two modes: operating on two waveforms or operating on many waveforms with the same constant. When a first waveform is selected, followed by an ENTER, and then a second waveform is selected, followed by pressing an operator button, the second waveform is added to the first. The first waveform is overwritten by the operation on both waveforms after truncating the first waveform to the common time domain and resampling the second waveform if necessary. The second waveform is not modified. The command equivalent form of multiple-waveform operations for the COMPARE routine is shown next, including the waveform selection portion, where test1 is the reference waveform and test2 is to be compared to test1.

Command Equivalent

WAVEFORM.test1

ENTER

WAVEFORM.test2,

COMPARE

In the second mode of operation, if one or more waveforms are selected without an ENTER, then you are prompted for a constant to operate on all of the waveforms selected. All of them are modified by this process. The command equivalent will require a <variable value> line after the operator command to perform correctly unless the operator uses the cursor and cursor mode is on. In this situation, the program pauses for cursor movement so you can interact with *xdamp*.

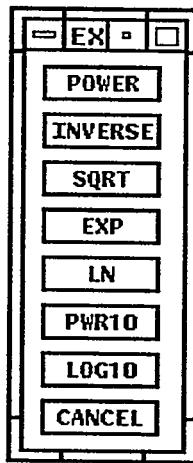
ADD

ADD is a multiple-waveform operator that adds waveforms or numbers to registers. This command modifies the ENTERed waveform or all of the selected waveforms.

SUBTRACT	SUBTRACT is a multiple-waveform operator that subtracts waveforms or numbers from registers. This command modifies the ENTERed waveform or all of the selected waveforms.
MULTIPLY	MULTIPLY is a multiple-waveform operator that multiplies waveforms or numbers to registers. This command modifies the ENTERed waveform or all of the selected waveforms.
DIVIDE	DIVIDE is a multiple-waveform operator that divides waveforms or numbers into registers. This command modifies the ENTERed waveform or all of the selected waveforms. Zero values of the divisor are masked out so numerical errors are trapped.
ABS	ABS is a single-waveform operator that takes the absolute value of waveforms or registers. This command modifies all of the selected waveforms.
AVERAGE	AVERAGE is a multiple-waveform operator that averages all selected waveforms and places the result in a new waveform with the name of the first selected waveform prefixed by "AVG_". This command does not modify any of the waveforms.
EXPONENTS	EXPONENTS pops a window (Fig. 6) to select among the various power, logarithm, and inverse logarithm single-waveform operators: POWER, INVERSE, SQRT, EXP, LN, 10^x , LOG. These modify all of the selected waveforms or the register chosen. A typical command sequence is shown, assuming the waveforms have already been selected.

Command Equivalent

EXPONENTS
POWER

FIGURE 6. EXPONENTS window

POWER	POWER is a single-waveform operator that takes a waveform or register to an <i>integer</i> power. This command modifies all of the selected waveforms.
--------------	--

INVERSE	INVERSE is a single-waveform operator that takes the inverse (1/x) of waveforms or a register. This command modifies all of the selected waveforms.
SQRT	SQRT is a single-waveform operator that takes the square root of waveforms or registers. This command modifies all of the selected waveforms.
EXP	EXP is a single-waveform operator that takes e raised to the waveform or e to the register. This command modifies all of the selected waveforms.
LN	LN is a single-waveform operator that takes the natural logarithm of the waveform or register. This command modifies all of the selected waveforms.
PWR10	PWR10 is a single-waveform operator that takes 10 raised to the waveform or register. This command modifies all of the selected waveforms.
LOG10	LOG10 is a single-waveform operator that takes the base-10 logarithm of the waveform or register. This command modifies all of the selected waveforms.
YUNITS	YUNITS is a single-waveform operator that allows you to change the vertical-axis-units label for one or more waveforms. This command modifies all of the selected waveforms. Its command equivalent requires an additional input.

Command Equivalent

YUNITS
<vertical units string>

CEILING	CEILING is a single-waveform operator that truncates the waveforms at values above those determined by the cursor or manual input. This command modifies all of the selected waveforms.
FLOOR	FLOOR is a single-waveform operator that truncates the waveforms at values below those determined by the cursor or manual input. This command modifies all of the selected waveforms.
TRUNCATE BEFORE	TRUNCATE BEFORE is a single-waveform operator that truncates the waveforms prior to a time determined by the cursor or manual input. This command modifies all of the selected waveforms.
TRUNCATE AFTER	TRUNCATE AFTER is a single-waveform operator that truncates the waveforms after a time determined by the cursor or manual input. This command modifies all of the selected waveforms.
TIMESHIFT	TIMESHIFT is a single-waveform operator that timeshifts the waveforms according to the cursor or an entered value. For cursor inputs, the cursor location becomes the new location of time zero. For entered inputs the value is subtracted from the starting time of the waveform. This command modifies all of the selected waveforms.
BASELINE	BASELINE is a single-waveform operator that “baselines” the waveforms according to the cursor or an entered value. The cursor or entered input defines a time at which the integral of all the selected waveforms is set to zero. Thus, offset before the main pulse

can be removed from the overall integral. This command modifies all of the selected waveforms.

TIMEALIGN

TIMEALIGN is a multiple-waveform operator that time aligns the selected waveforms using the cross-correlation algorithm of the COMPARE routine. This command modifies all of the selected waveforms except the ENTERed reference waveform.

TIMEBASE

TIMEBASE is a single-waveform operator that modifies the time axis by a multiplicative constant and/or allows you to change the time units. This command modifies all of the selected waveforms. Because this accepts two inputs, the command equivalent has an additional line as follows.

Command Equivalent

```
TIMEBASE
<multiplier>
<units string>
```

COMPARE

COMPARE is a multiple-waveform operator that compares two waveforms. This command *must* have the reference waveform selected and ENTERed. Then the variable waveform to be compared to the reference is selected and the COMPARE button is pressed. A copy of the variable waveform is used for all operations so the original is not modified. First, the variable waveform is resampled to the same time intervals as the reference waveform. Then a cross-correlation operation is performed to find the optimum time shift to best time align the two waveforms and the variable waveform is time shifted and truncated to completely overlap the reference waveform. Finally, an IDL polynomial fitting routine is used to fit the equation:

$$\text{REFERENCE} = \text{SCALE} * \text{VARIABLE} + \text{OFFSET}. \quad (\text{EQ 1})$$

Here, the scale is the multiplier that should be applied to the variable waveform to best fit it to the reference waveform and the offset is the constant that must be added to it. The reference waveform and the best fit variable waveform are plotted with the scale, offset, time shift, standard deviation, and correlation coefficient of the fit shown on the left side of the graphics area. The fitting parameters are stored in registers that can be used as input for subsequent operations according to the following scheme.

TABLE 4. Register Names and Contents from COMPARE Operation

Register Name	Quantity
#SCALE	Scale factor
#OFFSET	Baseline offset
#TSHIFT	Time shift

After the compare operation is complete, you can press the PRINT button to generate a hardcopy or press the ZOOM button to zoom in on the compared plot (and print this if you desire).

NORMALIZE

NORMALIZE is a single-waveform operator that divides each selected waveform by the absolute value of its peak value thus maintaining the waveforms polarity. This command modifies all of the selected waveforms which will afterwards range from -1 to 1.

INTEGRATE

INTEGRATE is a single-waveform operator that integrates the waveforms point by point using Simpson's rule. A new waveform is created for each one integrated if sufficient waveform memory is available; otherwise, the old waveforms are overwritten. These new waveforms are named the same as the original waveforms except the names are prefixed with "INT_". This command does not modify any of the waveforms if sufficient memory exists.

DIFFERENTIATE

DIFFERENTIATE is a single-waveform operator that differentiates the waveforms point by point using a three-point central differencing technique. A new waveform is created for each one differentiated if sufficient waveform memory is available; otherwise, the old waveforms are overwritten. These new waveforms are named the same as the original waveforms except the names are prefixed with "DIF_". This command does not modify any of the waveforms if sufficient memory exists.

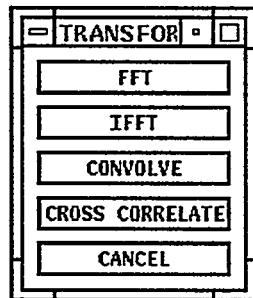
TRANSFORMS

The TRANSFORMS button pops a window (Fig. 7) with all available transforms accessed by buttons. This was done to save window space for the infrequently called routines: FFT, IFFT, CONVOLVE, and CROSS CORRELATE described below. The command mode operation is a bit different for these operators because an additional level of widgets exists. The following sequence would work for FFT with similar sequences for the others.

Command Equivalent

WAVEFORM.name
TRANSFORMS
FFT

FIGURE 7. TRANSFORMS subwindow

**FFT**

FFT is a single-waveform operator that takes the discrete Fourier transform of the selected waveforms. It returns complex waveforms, so for each waveform with name "DATA" three new waveforms ("RE_FFT_DATA", "IM_FFT_DATA", and "MAG_FFT_DATA") are created. The first waveform contains the real component, the second contains the imaginary component, and the third contains the magnitude of the FFT. Time-phase information is lost during this process because it cannot be easily removed from the inverse operation IFFT. Therefore, if you transform a waveform and then invert the transform you will recover the shape but the waveform will start at t=0. FFT results are displayed from the negative Nyquist frequency to the positive Nyquist frequency.

IFFT

IFFT is a multiple-waveform operator that takes the inverse discrete Fourier transform of the selected waveforms. The real component must be designated with the ENTER button, followed by selecting the imaginary component. This operation returns complex waveforms, so for each waveform "DATA" two new waveforms ("RE_IFFT_DATA" and "IM_IFFT_DATA") are created. The first waveform contains the real component and the second contains the imaginary component of the IFFT. Time-phase information is lost during this process. Therefore, if you transform a waveform and then invert the transform you will recover the shape but the waveform will start at t=0. A typical command sequence is shown below.

Command Equivalent

```
WAVEFORM.RE_Waveform
ENTER
WAVEFORM.IM_Waveform
TRANSFORMS
IFFT
```

CONVOLVE

CONVOLVE is a multiple-waveform operator that performs the mathematical convolution operation by serial addition (rather than an FFT/IFFT pair). The first waveform must be ENTERed even though the operation is commutative. A new waveform, containing the convolution of the two will be created with its name based on the ENTERed waveform name prefixed with "CONV_". The new waveform is longer than either original waveform.

CROSS CORRELATE

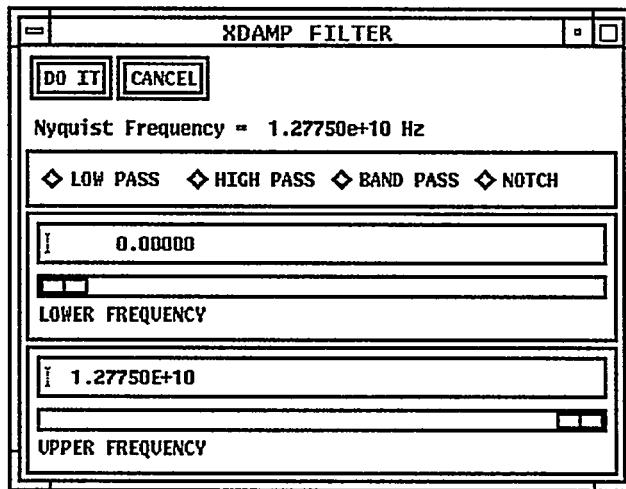
CROSS CORRELATE is a multiple-waveform operator that performs a cross correlation by serial addition (rather than an FFT/IFFT pair). The first waveform must be ENTERed because this operation is not commutative. A new waveform, containing the cross correlation of the two will be created with its name based on the ENTERed waveform name prefixed with "XCOR_". The new waveform is longer than either original waveform.

FILTERS

FILTERS is a single-waveform operator that performs low-pass, high-pass, band-pass and notch filtering. The FILTERS button pops a widget (Fig. 8) to obtain information about the desired filter and the filter frequencies desired. The original waveforms are overwritten. The filter algorithm uses a non-recursive digital filter with a 50dB Gibbs peak roll off obtained from a Kaiser windowing function. For notch filters in the command equivalent, simply set f_{lower} greater than f_{upper} . These frequencies are in Hz in the command file.

Command Equivalent

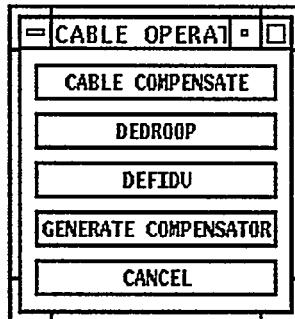
```
FILTERS,
<lower frequency>, <upper frequency>
```

FIGURE 8. FILTERS input window**CABLES**

The CABLES button pops a window (Fig. 9) with all available cable operations accessed by buttons. This was done to save window space for the infrequently called routines (CABLE COMPENSATE, DEDROOP, DEFIDU, and GENERATE COMPENSATOR) described below. The command mode operation is a bit different for these operators because an additional level of widgets exists. The following sequence would work for DEDROOP with similar sequences for the others.

Command Equivalent

WAVEFORM.name
CABLES
DEDROOP

FIGURE 9. CABLES subwindow**CABLE COMPENSATE**

CABLE COMPENSATE is a multiple-waveform operator that performs the mathematical convolution operation by serial addition of a response waveform with a cable compensator waveform. The first waveform must be ENTERed and then the cable compensator selected. The original waveform is overwritten and is not lengthened.

DEDROOP

DEDROOP is a single-waveform operator that removes an RC type roll off from a set of waveforms. This is done by serial convolution with a correction waveform. The original waveforms are overwritten.

DEFIDU

DEFIDU is a single-waveform operator that removes a positive fiducial marker from a set of waveforms. If the cursor preference is on, a cursor appears to define the portion of the waveform containing the cursor. Otherwise, it is assumed that the fiducial marker is within the first 10% of the waveform. The fiducial is assumed to define time zero and the waveform is deleted up to the point at which the fiducial drops to 10% of its peak plus three points. The original waveforms are overwritten.

GENERATE COMPENSATOR

GENERATE COMPENSATOR is a multiple-waveform operator that performs the mathematical deconvolution operation to find the cable compensator waveform. The reference waveform (directly into the digitizer) must be ENTERed and then data with the cable in place is selected. Next you are prompted for the truncation fraction (0.5 is a good start) and the desired length of the compensator array. The desired length defaults to a multiple of 500 but may need to be a power of two for some digitizers. This operation performs the serial deconvolution operation described in Boyer's report SAND87-3072. The cable compensator is given the name of the second waveform prefixed by "COMP_". The reference data, cable data, and compensated cable data are then plotted. If you repeat the operation with a different truncation fraction, then the compensator waveform is overwritten.

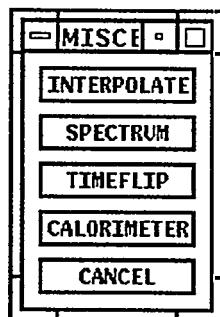
MISCELLANY

The MISCELLANY button pops a window (Fig. 10) with additional miscellaneous operations, accessed by buttons. This was done to save window space for the infrequently called routines: INTERPOLATE, SPECTRUM, TIMEFLIP and CALORIMETER described below. The command mode operation is a bit different for these operators because an additional level of widgets exists. The following sequence would work for TIMEFLIP with similar sequences for the others.

Command Equivalent

WAVEFORM.name
MISCELLANY
TIMEFLIP

FIGURE 10. MISCELLANY subwindow

**INTERPOLATE**

INTERPOLATE is a single-waveform operator that allows you to change a waveform to the values linearly interpolated between two defined times. In cursor mode, a box to

define the times appears and in non-cursor mode, you are prompted for the two times. The box may be moved by pressing and holding the left mouse button, resized by pressing and holding the middle mouse button, and exited by pressing the right mouse button. (Mac users will need to use special keys with the single mouse key to perform these acts.) This command modifies all of the selected waveforms.

SPECTRUM

SPECTRUM is a dual-waveform operator to calculate the electron spectrum given the voltage and current waveforms in a diode. You must select the voltage waveform and ENTER it followed by the current waveform. The procedure assumes that these two waveforms are properly time aligned. Then you are prompted for the number of spectral bins and the desired maximum energy. If the voltage or current are negative polarity, they are converted to positive polarity so the maximum spectral energy is always positive. SPECTRUM creates two new pseudo-waveforms, ESPECTRUM and CUMSPECTRUM. ESPECTRUM contains the differential electron energy number spectrum, normalized to an integral of one electron. The CUMSPECTRUM pseudo-waveform contains the cumulative number spectrum integral of ESPECTRUM, normalized to unity. The standard INTEGRATE command is not proper because of some tricks used in the binning of the electron spectrum to make the ESPECTRUM histogram look nice. The bottom bin begins at $\Delta E_{bin}/2$ and will have the number of particles with energies between 0 and ΔE_{bin} . All other bin widths are ΔE_{bin} . The peak bin ends at the maximum energy plus $\Delta E_{bin}/2$. For CUMSPECTRUM, the bottom bin begins at 0.0 and the top bin ends at the maximum energy. Note that any waveform with SPECTRUM in the name will be plotted as a histogram. The original two waveforms are not altered by this process. If you repeat the operation, the existing ESPECTRUM and CUMSPECTRUM waveforms will be overwritten unless you rename them.

TIMEFLIP

TIMEFLIP is a single-waveform operator that time reverses the waveforms while maintaining their start and end times. This command modifies all of the selected waveforms.

CALORIMETER

CALORIMETER is a single-waveform operator that calculates the value of the selected waveform at the time $t=0$ by linear extrapolation. The peak of the waveform is used as one point and the second is determined by the cursor or a delta time can be entered in an input box if the cursor preference is off.

Speed Buttons

A series of speed buttons are placed immediately above the waveform button area and just below the operator area. These are for the most commonly used tasks and some of the buttons are also available from the menus.

ENTER

The ENTER button is used to designate one waveform as special. For operations adding, subtracting, multiplying, or dividing two waveforms, the ENTER button designates the waveform to be overwritten. It is best thought of as a reverse Polish notation (RPN) style ENTER button. For operations that use one waveform as a reference (i.e. COMPARE or cable compensator creation) the ENTER button designates the reference waveform.

CLEAR

The CLEAR button deselects all waveforms, clears the plot area, and clears the message box.

Command Equivalent

`CLEAR`**SELECT**

The SELECT button selects waveforms based on an input string containing wildcard characters (*) in the specification. For instance V* would select all waveforms beginning with the capital letter V. The string specification is case sensitive.

Command Equivalent

`SELECT`
`<wildcard selection string>`**ALL**

The ALL button selects all of the waveforms.

Command Equivalent

`ALL`**PLOT**

The PLOT button causes all waveforms selected to be immediately plotted. If no waveforms have been selected, then all the waveforms are plotted sequentially with a two second delay.

Command Equivalent

`PLOT`**ZOOM**

The ZOOM button replots the selected waveforms and places a zoom box on the screen if the cursor preference is on. The zoom box may be moved by pressing and holding the left mouse button, resized by pressing and holding the middle mouse button, and exited by pressing the right mouse button. (Mac users will need to use special keys with the single mouse key to perform these acts.) The zoomed image is then plotted. If the cursor preference is off, you are prompted for the four limits of the zoom plot. In the command equivalent, the last four inputs are only needed if you are not in cursor mode.

Command Equivalent

`ZOOM`
`<tmin in data units>, <tmax>, <ymin>, <ymax>`

PRT SCRN

The PRT SCRN button causes whatever is in the current plot window to be placed in a print file. This is also accessible via the PRINT menu as PRINT.SCREEN. The print file is not spooled until you exit the program or force a spool through the PRINT menu.

Command Equivalent

PRT SCRN

Waveform Buttons

The names of the waveform buttons vary with the application. The technique used to select any waveform is identical; simply push the button. To deselect a waveform, simply push the button again. The command syntax is shown below for a waveform named Example. Note that the names are case sensitive.

Command Equivalent

WAVEFORM.Example

If too many waveforms exist to fit in the waveform button box, then the box becomes a scrolling region and you may need to use the scroll bars to find the waveforms to select.

Beware of two things. The waveform selection process is a toggle that may cause problems in automatic execution mode if a waveform is selected for a second time and becomes deselected. Also, waveform names are case sensitive.

Problems and Troubleshooting

If you experience a problem, *first* look in the message area for an informational message about anything that may be illegal such as performing an operation that requires an ENTER without performing the ENTER or opening a file with the incorrect format.

If your display does not appear like Figure 1, "xdamp Screen Layout," on page 11, then you probably need to reduce the widget font size in your .Xdefaults file (See "GETTING STARTED" on page 8.) and then reboot. If this doesn't work, reduce the size of the graphics font area according to your display's limitations by using the GRAPHIC AREA command in the xdamp.ini file.

If *xdamp* fails when you first try to plot a waveform, then you probably have an illegal font name in your xdamp.ini file. Remove the GFONT=font_string line from this file and try *xdamp* again.

Difficulties with the UNIX pickfile widget are usually due to rapid multiple clicking on selections. This widget wants *single* clicks (on UNIX systems) and is not particularly fast because it must spawn system commands to obtain the directory information. Be patient. The pickfile widget is also case sensitive.

A common difficulty is not recognizing that waveform names *are* case sensitive when defining a select string with wild cards. Similarly, the file filter definition must contain an asterisk like *.hdf and the file names are also case sensitive. Without a wild card, nothing will be selected.

The single file automatic execution mode does not initialize many variables (for maximum flexibility). Therefore, you should start your automatic execution file assuming that the waveform selection state is ill-defined.

When using automatically loaded registers, remember that the pulse parameter values come from the last waveform selected, not the first waveform selected. When in doubt, just plot one waveform at a time. The pulse parameters are generated whether or not you elect to have the results displayed on the screen.

The xfont widget application is not very robust. Frequently, it will have conflicts if another application (FrameMaker) is using display PostScript. It is best to set the font by using this widget *immediately* after starting only IDL, reading the font in the message window and placing this in the xdamp.ini file with the GFONT=font_string command.

Color table problems can abound in this application. A specific symptom is if the zoom box is not visible. Other applications and IDL can use the same color map, not always with benign interactions. For UNIX systems, use the Idl.colors: 64 command in your .Xdefaults file to reserve some colors to IDL. For other systems, the order in which applications are started can affect what happens.

In automatic execution mode, if you are automatically clearing the stack after each operation and are using a journal file generated with this setting disabled, very strange

things can occur. Many of the operators are overloaded and think that trying to add without any waveform selected means that you want to add a constant to a register. When there is no valid information for the register arithmetic, the whole automatic execution will fail. This cannot be trapped easily so you must generate command files with the stack clearing preference you will use when executing them.

Do not end a macro file with a comment or blank line. *xdamp* thinks another command follows this and fails when it reaches the end of file unexpectedly.

Do not use FILE.GET ARRAY to add files with only a single array. Instead use the FILE.APPEND which is faster and requires less input from the user.

Macintosh Specific Issues

On Macintoshes, there is a bug in IDL that causes very slow compilation. This should be fixed in the release after 3.6.1 but you can speed things up by replacing every occurrence of an at sign (@) with a fully qualified path to the IDL xdamp source code in all the *.pro files. This will markedly speed things up.

If your display does not appear like Figure 1, "xdamp Screen Layout," on page 11, then you probably need to reduce the widget font size via the preferences section. If this doesn't work, then reduce the size of the graphics font area in the file XDSys_sp.pro or with the GRAPHIC AREA command in the xdamp.ini file according to your display's limitations.

Do not use the close window box to complete data entries. Use the DONE button in the widget instead. Otherwise, IDL will not receive an event to process the input stream.

Macintosh output cannot be automatically spooled. You must print the xdamp.out file yourself after performing a PRINT.QUEUE command. This can be done using the shareware program DropPs available from mac.archive.umich.edu via anonymous ftp.

On Macintoshes, IDL cannot handle the waveform button widget very well in this release. Therefore, if you have more than about 80 waveforms, you will exceed the screen size. Converting this area to a scrolling region doesn't work (yet). Try to keep the number of waveforms down or use a smaller font.

Macintosh journal file output will appear on a single line if you are using the Power PC version. Using an editor, change all the linefeeds (ASCII 10) to carriage returns and continue to edit the file normally. This is a known IDL bug.

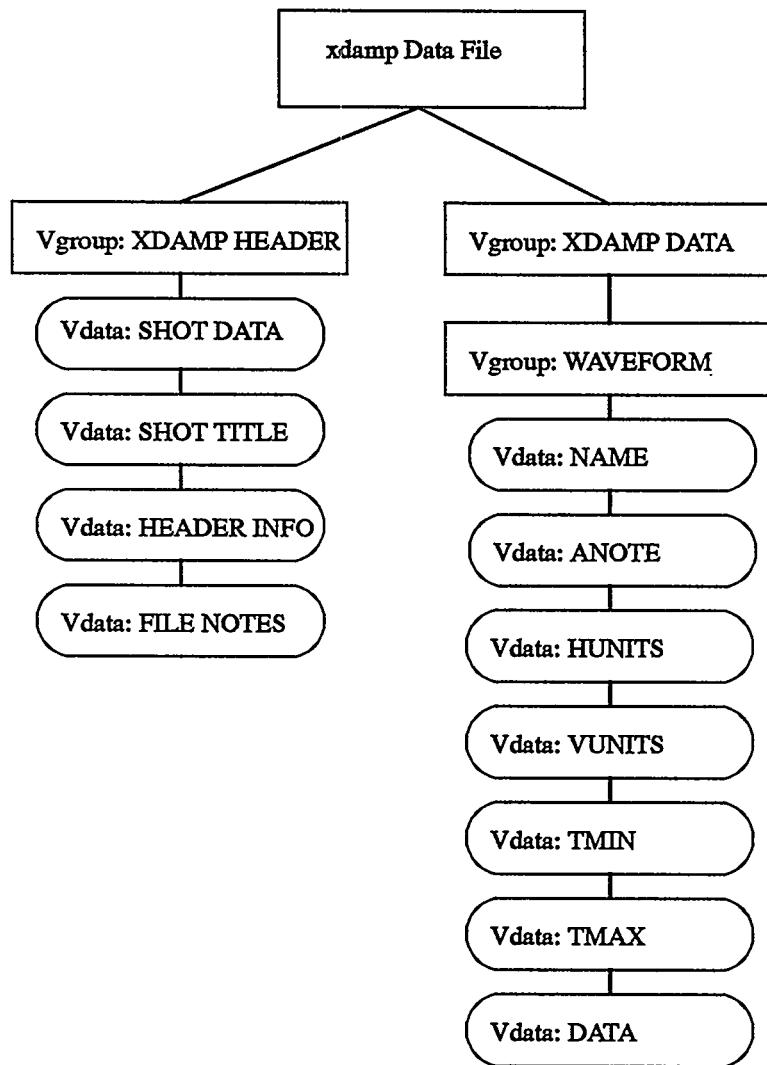
To use the ZOOM, you need three mouse buttons. The left button is the normal Mac button, the center button is the Option key with the mouse button and the right button is the Apple key with the mouse button.

The Xfont application does not function for Macintosh computers. Instead, use the preferences section of the Macintosh menu bar to set the desired font for the application.

On a Macintosh, DO NOT use the FILE.QUIT on the screen menu bar because this will not properly close the journal file.

HDF Data File Organization

Vgroup and Vdata structure for xdamp HDF file format with field names.



*HDF read and write IDL procedures***READ HDF FILE
PROCEDURE**

```

:+
; Copyright (c) 1995, Sandia Corporation. The United States Government
; retains a nonexclusive license in this software as prescribed in
; AL 88-1 and AL 91-7. Export of this program may require a license from
; the United States Government.
;+++++
; procedure XD_readfile, filename, append keyword, overwrite keyword,
;   getarray keyword
;
; Internal routine that opens the file, reads the data
;
PRO XD_readfile, fname, APPEND = append, OVERWRITE = overwrite, $
    GETARRAY = getarray

@XD_bases.cmn
@XD_data.cmn
@XD_jml.cmn
@XD_stack.cmn

XD_print, "Reading "+fname

WIDGET_CONTROL, wform_base, MAP=0

IF KEYWORD_SET(OVERWRITE) THEN over = 1 ELSE over = 0
IF KEYWORD_SET(APPEND) THEN appnd = 1 ELSE appnd = 0
IF KEYWORD_SET(GETARRAY) THEN geta = 1 ELSE geta = 0

;*** figure out the prepend string
str = XDfile_pre(fname)
IF (geta OR over OR appnd EQ 1) THEN str = ""

;*** loop through the file reading the information and saving it
IF (narrays EQ 0) THEN BEGIN
    i=1
    shot_title = ""
    shot_date = ""
ENDIF ELSE i=narrays+1
IF (NOT over AND NOT appnd AND NOT geta) THEN BEGIN
    shot_title = ""
    shot_date = ""
ENDIF
anote = ""
s_buffer = "abcdefghijklmnopqrstuvwxyz" + $
    "abcdefghijklmnopqrstuvwxyz" + $
    "abcdefghijklmnopqrstuvwxyz" + "acdefgh"

;*** HDF FORMAT

```

```

IF (HDF_ISHDF(fname)) THEN BEGIN
  hdf_handle = HDF_OPEN(fname, /READ)

;*** find header Vgroup
vgxh = -1L
REPEAT BEGIN
  t = vgxh
  vgxh = HDF_VGETID(hdf_handle, t)
  vid = HDF_VATTACH(hdf_handle, vgxh, /READ)
  HDF_VGET, vid, NAME=tname
  HDF_VDETACH, vid
ENDREP UNTIL (tname EQ "XDAMP HEADER")
vgxhh = HDF_VATTACH(hdf_handle, vgxh, /READ)

;*** find shot information Vgroup
vgxi = -1L
inote = 0
ihead = 0
REPEAT BEGIN
  t = vgxi
  vgxi = HDF_VGETNEXT(vgxhh, t)
  IF (HDF_VISVS(vgxhh, vgxi)) THEN BEGIN
    vid = HDF_VSATTACH(hdf_handle, vgxi, /READ)
    HDF_VSGET, vid, FIELDS=tname
    IF (NOT appnd) THEN BEGIN
      CASE tname OF
        "SHOT DATE": k = HDF_VSREAD(vid, shot_date, FIELDS="SHOT DATE")
        "SHOT TITLE": k = HDF_VSREAD(vid, shot_title, FIELDS="SHOT TITLE")
        "HEADER INFO": BEGIN
          k = HDF_VSREAD(vid, stg, FIELDS="HEADER INFO")
          header(ihead) = STRING(stg)
          ihead = ihead+1
        ENDCASE
        "FILE NOTES": BEGIN
          k = HDF_VSREAD(vid, stg, FIELDS="FILE NOTES")
          fnotes(inote) = STRING(stg)
          inote = inote+1
        ENDCASE
        ELSE:
      ENDCASE
    ENDIF
    HDF_VSDETACH, vid
  ENDIF
ENDREP UNTIL (vgxi EQ -1L)
shot_date = STRING(shot_date)
shot_title = STRING(shot_title)
HDF_VDETACH, vgxhh

;*** find data Vgroup
vgxd = -1L

```

```

REPEAT BEGIN
  t = vgxd
  vgxd = HDF_VGETID(hdf_handle, t)
  vid = HDF_VATTACH(hdf_handle, vgxd, /READ)
  HDF_VGET, vid, NAME=tname
  HDF_VDETACH, vid
ENDREP UNTIL (tname EQ "XDAMP DATA")
vgxdh = HDF_VATTACH(hdf_handle, vgxd, /READ)

;*** find waveform Vgroup
vgwd = -1L
REPEAT BEGIN
  t = vgwd
  vgwd = HDF_VGETNEXT(vgxdh, t)
  IF (HDF_VISVG(vgxdh, vgwd)) THEN BEGIN
    vid = HDF_VATTACH(hdf_handle, vgwd, /READ)
    HDF_VGET, vid, NAME=tname
    HDF_VDETACH, vid
  ENDIF
ENDREP UNTIL (tname EQ "WAVEFORM")
vgwdh = HDF_VATTACH(hdf_handle, vgwd, /READ)

;*** now loop through the waveforms
id = -1L
done = 0
id = HDF_VGETNEXT(vgwdh, id)
REPEAT BEGIN
  flag = 0
  REPEAT BEGIN
    vds = HDF_VSATTACH(hdf_handle, id, /READ)
    HDF_VSGET, vds, FIELDS=testfield
    CASE testfield OF
      "NAME": k = HDF_VSREAD( vds, aname, FIELDS="NAME")
      "ANOTE": k = HDF_VSREAD( vds, anote, FIELDS="ANOTE")
      "HUNITS": k = HDF_VSREAD( vds, hunit, FIELDS="HUNITS")
      "VUNITS": k = HDF_VSREAD( vds, vunit, FIELDS="VUNITS")
      "TMIN": k = HDF_VSREAD( vds, tmin, FIELDS="TMIN")
      "TMAX": k = HDF_VSREAD( vds, tmax, FIELDS="TMAX")
      "DATA": BEGIN
        k = HDF_VSREAD( vds, ary, FIELDS="DATA")
        flag = 1
      ENDCASE
      ELSE:
    ENDCASE
    HDF_VSDETACH, vds
    id = HDF_VGETNEXT(vgwdh, id)
  ENDREP UNTIL flag
  dbase(0).name = str+STRING(aname)
  dbase(0).anote = STRING(anote)
  IF (STRLEN(STRCOMPRESS(dbase(0).anote,/REMOVE_ALL)) EQ 0) THEN $

```

```

        dbase(0).anote = dbase(0).name+":"
        dbase(0).npts = N_ELEMENTS(ary)
        dbase(0).xunits = STRING(hunit)
        dbase(0).yunits = STRING(vunit)
        dbase(0).tmin = tmin
        dbase(0).tmax = tmax
        dbase(0).yarray(*) = 0.0
        dbase(0).yarray = ary
        IF (over) THEN BEGIN
            FOR j = 0, next_stack-1 DO BEGIN
                k = stack(j)
                IF dbase(k).name EQ dbase(0).name THEN dbase(k) = dbase(0)
            ENDFOR
        ENDIF ELSE BEGIN
            dbase(i) = dbase(0)
            i = i+1
            IF (i GT maxarrays) THEN BEGIN
                id = -1
                XD_print, "Too many waveforms in data file."
            ENDIF
        ENDELSE
    ENDREP UNTIL (id EQ -1L)
    HDF_VDETACH, vgwdh
    HDF_VDETACH, vgxdh
    HDF_CLOSE, hdf_handle
ENDIF

IF (NOT over) THEN narrays = i-1

;*** run through strings, stripping out any exclamation points, they are bad
FOR i = 1, narrays DO BEGIN
    j = STRPOS(dbase(i).name,"!")
    WHILE (j NE -1) DO BEGIN
        a = dbase(i).name
        STRPUT, a, "|",j
        dbase(i).name = a
        j = STRPOS(dbase(i).name,"!")
    ENDWHILE
    j = STRPOS(dbase(i).yunits,"!")
    WHILE (j NE -1) DO BEGIN
        a = dbase(i).yunits
        STRPUT, a, "|",j
        dbase(i).yunits = a
        j = STRPOS(dbase(i).yunits,"!")
    ENDWHILE
    j = STRPOS(dbase(i).xunits,"!")
    WHILE (j NE -1) DO BEGIN
        a = dbase(i).xunits
        STRPUT, a, "|",j
        dbase(i).xunits = a
    ENDWHILE

```

```

        j = STRPOS(dbase(i).xunits, "!")
ENDWHILE
ENDFOR

IF (narrays EQ 0) THEN XD_print, "Incorrect data format, select another." $
ELSE XD_print, "Done reading "+STRTRIM(STRING(narrays),2)+ $
        " waveforms."
RETURN
END;===== end of XD_readfile routine =====

```

**WRITE HDF FILE
PROCEDURE**

```

;+
; Copyright (c) 1995, Sandia Corporation. The United States Government
; retains a nonexclusive license in this software as prescribed in
; AL 88-1 and AL 91-7. Export of this program may require a license from
; the United States Government.
;+++++
; procedure XD_savefile, filename
;
; Internal routine that actually writes a file in HDF format.
;
PRO XD_savefile, new_filename
@XD_data.cmn

;*** declare variables
XD_fields = ["NAME", "ANOTE", "VUNITS", "HUNITS", "TMIN", "TMAX",
"DATA"]
XD_ftype = ["BYTE", "BYTE", "BYTE", "BYTE", "FLOAT", "FLOAT", "FLOAT"]
nfields = N_ELEMENTS(XD_fields)
s_buffer = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"+ $
"abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"+ $
"abcdefghijklmnopqrstuvwxyzacdefgh"
f_buffer = 1.0

;*** open a new HDF file, get the file handle
hdf_handle = HDF_OPEN(new_filename, /WRITE, /CREATE)

;*** create a vgroup id, attach it and set the name
vg_header = HDF_VATTACH(hdf_handle, -1L, /WRITE)
HDF_VSETNAME, vg_header, "XDAMP HEADER"

;*** do another for the data
vg_data = HDF_VATTACH(hdf_handle, -1L, /WRITE)
HDF_VSETNAME, vg_data, "XDAMP DATA"

;*** this will go inside data in case other types are added someday
vg_wvfm = HDF_VATTACH(hdf_handle, -1L, /WRITE)
HDF_VSETNAME, vg_wvfm, "WAVEFORM"

```

```

;*** first write shot date and shot title
vds = HDF_VSATTACH(hdf_handle, -1L, /WRITE)
HDF_VSFDEFINE, vds, "SHOT DATE", /BYTE
HDF_VSWRITE, vds, "SHOT DATE", STRTRIM(shot_date)
HDF_VINSERT, vg_header, vds
HDF_VSDETACH, vds
vds = HDF_VSATTACH(hdf_handle, -1L, /WRITE)
HDF_VSFDEFINE, vds, "SHOT TITLE", /BYTE
HDF_VSWRITE, vds, "SHOT TITLE", STRTRIM(shot_title)
HDF_VINSERT, vg_header, vds
HDF_VSDETACH, vds

;*** now write in header stuff
FOR i = 0, N_ELEMENTS(header)-1 DO BEGIN
  temp = STRTRIM(header(i))
  IF (STRLEN(temp) GT 0) THEN BEGIN
    vds = HDF_VSATTACH(hdf_handle, -1L, /WRITE)
    HDF_VSFDEFINE, vds, "HEADER INFO", /BYTE
    HDF_VSWRITE, vds, "HEADER INFO", temp
    HDF_VINSERT, vg_header, vds
    HDF_VSDETACH, vds
  ENDIF
ENDFOR

;*** now add in the file notes to the header
FOR j = 0, N_ELEMENTS(fnotes)-1 DO BEGIN
  temp = STRTRIM(fnotes(j))
  IF (STRLEN(temp) GT 0) THEN BEGIN
    vds = HDF_VSATTACH(hdf_handle, -1L, /WRITE)
    HDF_VSFDEFINE, vds, "FILE NOTES", /BYTE
    HDF_VSWRITE, vds, "FILE NOTES", temp
    HDF_VINSERT, vg_header, vds
    HDF_VSDETACH, vds
  ENDIF
ENDFOR

;*** now build a waveform description
;*** first get a set of vdata ids to put a waveform in
FOR j = 1, narrays DO BEGIN
  FOR i = 0, nfields-1 DO BEGIN
    vds = HDF_VSATTACH(hdf_handle, -1L, /WRITE)
    CASE XD_ftype(i) OF
      "BYTE": HDF_VSFDEFINE, vds, XD_fields(i), /BYTE
      "FLOAT": HDF_VSFDEFINE, vds, XD_fieldS(i), /FLOAT
    ENDCASE
    CASE XD_fields(i) OF
      "NAME": HDF_VSWRITE, vds, "NAME", dbase(j).name
      "ANOTE": HDF_VSWRITE, vds, "ANOTE", dbase(j).anote
      "HUNITS": HDF_VSWRITE, vds, "HUNITS", dbase(j).xunits
      "VUNITS": HDF_VSWRITE, vds, "VUNITS", dbase(j).yunits
    ENDIF
  ENDFOR
ENDFOR

```

```
"TMIN": HDF_VSWRITE, vds, "TMIN", dbase(j).tmin
"TMAX": HDF_VSWRITE, vds, "TMAX", dbase(j).tmax
"DATA": BEGIN
    temp = dbase(j).yarray(0:dbase(j).npts-1)
    HDF_VSWRITE, vds, "DATA", temp
    END
    else:
ENDCASE
HDF_VINSERT, vg_wvfm, vds
HDF_VSDETACH, vds
ENDFOR
ENDFOR

;*** finished writing stuff, close file
HDF_VINSERT, vg_data, vg_wvfm
HDF_VDETACH, vg_wvfm
HDF_VDETACH, vg_data
HDF_VDETACH, vg_header
HDF_CLOSE, hdf_handle

RETURN
END ;===== end of XD_savefile routine =====
```

References

IDL User's Guide, Interactive Data Language, Version 3.6, Research Systems, Inc. April 1994.

IDL Reference Guide, Interactive Data Language, Version 3.6, Research Systems, Inc. April 1994.

IDL Mathematics Guide, Interactive Data Language, Version 3.6, Research Systems, Inc. April 1994.

IDL Scientific Data Formats, Interactive Data Language, Version 3.6, Research Systems, Inc. April 1994.

NCSA HDF Specification and Developer's Guide, National Center for Supercomputing Applications, 152 Computing Applications Building, 605 East Springfield Avenue, Champaign, IL 61820.

NCSA HDF Vset, National Center for Supercomputing Applications, 152 Computing Applications Building, 605 East Springfield Avenue, Champaign, IL 61820.

R. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, New York, 1965.

R. B. Spielman, *A Non-Recursive Lowpass Digital Filter Algorithm*, Sandia National Laboratories Report No. SAND82-0930, November 1982.

W. B. Boyer, *Computer Compensation for Cable Signal Degradations*, Sandia National Laboratories Report No. SAND87-3072, December 1987.

List of Distribution Files

xdamp.pro	Main procedure
xdamp.ini	sample xdamp initialization file
manual.ps	PostScript version of manual
xdamp.help	IDL style help file
example.hdf	sample hdf data file
example.cat	sample sicdas data file
example.damp	sample damp data file
example.list	sample damp header file
XD_auto.pro	automatic execution files
XD_edit.pro	edit menu procedures
XD_etest.pro	error testing procedure
XD_file.pro	file menu procedures
XD_filtr.pro	filter procedures
XD_input.pro	input procedures
XD_legnd.pro	legend procedure
XD_ops_1.pro	single waveform operators procedures
XD_ops_2.pro	dual waveform operators procedures
XD_optns.pro	options procedures
XD_print.pro	print menu procedures
XD_pulse.pro	pulse parameter calculation procedures
XD_query.pro	query box procedures
XD_subs.pro	sub-widget box procedures
XD_tools.pro	tools menu procedures
XD_utils.pro	utility procedures
XDsys_sp.pro	system specific procedures
XD_anote.cmn	array notes common
XD_bases.cmn	widget bases common
XD_data.cmn	data block common
XD_filtr.cmn	filter widget common
XD_fnote.cmn	file notes common
XD_geta.cmn	get array common
XD_jrnl.cmn	journal common
XD_optns.cmn	options widget common
XD_plot.cmn	plot information common
XD_query.cmn	query widget common
XD_regs.cmn	registers common
XD_stack.cmn	data stack common

Distribution

1	MS1106	J. S. Gustwiller, 9342
1	MS1106	R. E. Pepping, 9342
1	MS1155	W. Beezhold, 9303
1	MS1159	W. H. Barrett, 9311
1	MS1159	J. I. Greenwoll, 9312
1	MS1159	M. A. Hedemann, 9311
1	MS1159	R. E. Klingler, 9311
10	MS1160	R. O'Rourke, 9322
10	MS1160	R. Broyles, 9322
1	MS1167	T. A. Zarick, 9351
1	MS1168	W. B. Boyer, 9321
1	MS1168	P. F. McKay, 9321
1	MS1168	J. W. Lee, 9321
1	MS1169	R. E. French, 9322
1	MS1169	D. B. Hughes, 9322
25	MS1179	W. P. Ballard, 9341
1	MS1179	D. E. Beutler, 9341
1	MS1179	R. E. Craven, 9341
1	MS1179	J. R. Lee, 9341
1	MS1179	T. B. Parson, 9341
1	MS1186	D. B. Seidel, 1242
1	MS1187	L. P. Mix, 1271
1	MS1188	G. C. Tisone, 1212
1	MS1188	M. Trahan, 1212
1	MS1188	J. S. Wagner, 1212
1	MS1194	C. Deeney, 1273
1	MS1194	D. H. McDaniel, 1273
1	MS1194	T. J. Nash, 1273
1	MS1194	L. Ruggles, 1273
1	MS1194	J. F. Seamen, 1273
1	MS1194	R. B. Spielman, 1273
2	MS1194	J. L. Porter, 1273
1	MS9018	Central Technical Files, 8523-2
5	MS0899	Technical Library, 13414
1	MS0619	Print Media, 12615
2	MS0100	Document Processing, 7613-2 For DOE/OSTI
3	MS0161	Patent and Licensing Office, 11500