Status report on XAS software libraries

(DAWG-REP 19/92)

L. Chiappetti - IFCTR - Fri, 28 Aug 1992

0. Executive summary

This report presents the work done in the last few months about the VOS and XAS file i/o routines. The main highlights are recalled here.

The VOS library prototype is nearly complete. All functions required on all machines are implemented (while some functions required as auxiliary functions on some machines only are provisionally implemented only on those machines) and working. There are still some limited problems with environment access on Unix (the problem is being investigated by D.Dal Fiume, to which I am indebted for lot of precious help with Fortran-to-C interfacing).

The VOS library includes routines for: access to the run string (partially also in the GENERAL library), access to global variables, basic file access (open and inquire), program control and miscellaneous utilities. As an unforeseen addition I have also developed a successful prototype of a dynamic memory allocation routine.

A working set of essential routines to access all main types of XAS files are available in the GENERAL library. This includes higher level open and close, and access to header keywords. Some more prototype routines are available in the demo sources (see below), but have not yet been moved to the library.

During the development of the file access routines, some refinements to the specification of the file format (extensively documented in DAWG-REP 20/92) have been made. They are summarized below in section 1.4. The main points are: usage of INTEGER*2 keywords is now deprecated, support to mini-header spanning record boundaries, support to arrays of keywords (according to a request of D.Dal Fiume, although their usage is not officially encouraged; support to 3-d images and in general packing images, spectra etc. in a single file is also formally compatible with this s/w although not officially supported). Support to automatic conversion of system dependent format (e.g. to transparently read on Unix a file written on the VAX) is planned for the future, but not implemented yet.

A set of demo programs have been prepared and are available to all people interested. These demos include representative demos and functional demos. The first group exercises writing and reading sample files (images, spectra, time profiles and photon lists) in XAS format. The second group provides prototype utilities of general use, and include a general tool for file header display, a sample header editing program, and a prototype converter to the FITS format.

The demos are mainly intended for scientific programmers, to be studied in order to have an idea of how essential file, runstring and environment access is done, without necessarily looking into lower level routines (which however are available for the courageous).

It is worth mentioning that the files produced have all been accessed (either directly or through the FITS conversion process) with SAOImage, IDL, MIDAS and IRAF, therefore practically demonstrating the feasibility of an "open system" as planned.

Of course it shall also be said that the work done has proven the worth of the VOS approach, since all programs (and GENERAL library routines) are compatible at source code level between VMS and Unices, and in fact I systematically use a single physical source (residing on VMS and accessed via NFS for compilation).
Document for which no computer readable source exists any longer

Partial scan of original hardcopy of 32pag. (available on request) supplied
## Table of content

0. Executive summary

1. Report on work done
   1.1. The demos
      1.1.1. functional demos
      1.1.1.1. Header display program
      1.1.1.2. Programmers' hints for header content listing
      1.1.1.3. Header editing program
      1.1.1.4. Programmers' hints for header editing
      1.1.1.5. FITS conversion program
      1.1.1.6. Programmers' hints on format conversion
      1.1.2.1. Creating and reading an image file
      1.1.2.2. Programmers' hints on images
      1.1.2.3. Creating a time profile
      1.1.2.4. Plotting a time profile
      1.1.2.5. Programmers' hints on time profiles
      1.1.2.6. Creating a spectrum
      1.1.2.7. Plotting a spectrum
      1.1.2.8. Programmers' hints on spectra
      1.1.2.9. Creating and reading a photon list
      1.1.2.10. Programmers' hints on writing and reading a photon list
      1.1.2.11. Programmers' hints on reading and writing to a generic table
      1.1.2.12. Copying file headers
      1.1.2.13. Programmers' hints on copying headers across files
      1.1.2.14. Further possibilities: extend file, truncate file
      1.1.2.15. Future example demos
   1.2. Introduction to the three library layers
      1.2.1. the GENERAL library
      1.2.2. the VOS library upper level Z
      1.2.3. the lower level ZC for Unix (also VAX)
   1.3. Library routine description
      1.3.1. Routines for environment access
      1.3.1.1. Runstring
      1.3.1.2. Global environment
      1.3.2. Dynamic memory allocation
      1.3.2.1. Usage
      1.3.2.2. Implementation
      1.3.3. Routines for file access
      1.3.3.1. at VOS level
      1.3.3.2. at XAS file level
      1.3.4. Prototype routines in demos
      1.3.5. Routine calling sequences
   1.4. changes to XAS file doc

2. How to retrieve and use the demos
   2.1. What to retrieve
   2.2. How to install
   2.3. How to use

3. Practical examples, prescriptions and recipes
   3.1. How to read from terminal or runstring
   3.2. How to access global values
   3.3. How to terminate a program
   3.4. How to open a generic file (usually not done)
   3.5. How to open a XAS file
   3.6. How to handle errors
   3.7. How to handle a typical XAS file
   3.8. How to handle header keyword

4. Compatibility with external packages
   4.1. Using images with SAOImage, IDL, MIDAS, IRAF
   4.2. Using tables with IDL, MIDAS, IRAF
   4.3. Access to header in IDL, MIDAS, IRAF
1. Report on work done

The work done has moved essentially on three lines: a) to freeze a working prototype of the VOS routines with almost full functionality; b) to generate a working and complete prototype of essential XAS file access routines; c) to experiment with some higher level XAS file access routines. In particular the latter point intended to demonstrate the ease and flexibility of programming for the general user: for this purpose a set of demo programs have been produced, and are available to everybody.

We give here: i) a description of the demos (section 1.1); ii) an introduction to the software libraries (section 1.2); iii) more details on individual routines (section 1.3).

1.1. The demos

The demos include representative demos and functional demos. The first group exercises writing and reading sample files (images, spectra, time profiles and photon lists) in XAS format. The second group provides prototype utilities of general use, and include a general tool for file header display, a sample header editing program, and a prototype converter to FITS format.

The demos are mainly intended for scientific programmers, to be studied in order to have an idea of how essential file, runstring and environment access is done, without necessarily looking into lower level routines (which however are available for the courageous).

Instructions on network retrieval of the demos are provided below in section 2. In 2.3 we also give an introductory dry run to the demos. You might want to skip to there to know what the demos do, or go on reading here a more detailed description of how the demos work.

For all demos we give first user's hint (how to call the demo, what it does), followed by a programmers' hint section.

1.1.1. functional demos

There are three demos which can be classified as "functional", as they provide more or less rough prototypes of actual tools which can be used as support to scientific analysis. They include:

1.1.1.1. Header display program

The demo is a functional prototype of an utility to inspect the header of a XAS file. The program is called as:

demo_header_display [filename]

where the filename will be prompted if not supplied. The demo will then produce a self-explanatory listing on the terminal screen. After some general information, each keyword will be shown on one or more lines as in the following examples:

(C) OBSERVER = L.Chiappetti - IFCTR
(J) I4VECTOR( 2) = 72

for a scalar keyword
for an array element

The code in parenthesis indicates the data type (C=character, I=integer*2, J=integer*4, R=real, D=double etc.), followed by the name, the subscript (for arrays) and the value, appropriately formatted.

1.1.1.2. Programmers' hints for header content listing

The demo_header_display source contains a prototype version of H_NEXT_KEYWORD (to be possibly moved to the GENERAL library). This routine performs a scan of the header of the current file, and returns the name of the next keyword. Its usage should be necessary only in the few dedicated applications (like this program) which need a systematic scan.

The program then calls H_FIND_KEYWORD to find the type and length of the next keyword (once the name is known), and H_READ_KEYWOD to get its value. This program makes a pecular usage of the
endif
print,' there are ',izoff,' bytes in miniheader'
skip=izoff+reclen*datasize
point_lun,1,skip
;
; this reads the header
loop:
  code=0B
  length=0B
  name='12345678'
  value='NOT FOUND'
  readu,1,code,length,name
  if code eq 0 and length eq 0 then begin
    close,1
    return
  endif
  ; prepare space for value
  if code eq 0 then temp=string(replicate(32b,length))
  if code eq 1 then begin
    length=length/2B
    temp=intarr(length)
  endif
  if code eq 2 or code eq 3 then length=length/4B
  if code eq 2 then temp=lonarr(length)
  if code eq 3 then temp=fllarr(length)
  if code eq 4 then begin
    length=length/8B
    temp=dblarr(length)
  endif
  readu,1,temp
  print,' type=’,code,’ length=’,length,’ ',name,’=’,temp
  if not eof(1) then goto,loop
  ; close file
  close,1
  return
end

Complete access to XAS header information is possible within MIDAS (as descriptors) or IRAF (as keywords) if the file is converted to FITS (with demo_tofits) and read into MIDAS or IRAF with usual commands (this is not possible for tables in IRAF yet). However the handling of array keywords, which are not FITS standard, is current subject to limitations, which are described in 1.1.1.5.