SAX data analysis software

A study about Fortran compatibility

Prepared for the
SAX data analysis working group
by

L.Chiappetti (IFCTR)

Draft - August 1986

This document is file FORTRAN.SAX

1 : Introduction

This document is intended as the first in a collection of notes for discussion within the SAX data analysis working group. The final aim of this group should be the production of a set of rules, which should enable the different software modules written at the various SAX hardware institutes to be a) easily integrated in a common SAX data analysis package at the Scientific Data Centre (CDS); b) easily exchanged between the hardware institutes (and the scientific community at large ?).

A driving consideration has been so far (August 1986) the fact that the hardware institutes presently have different computers, and that this situation is not likely to change on a short term (the hardware at the CDS - and the location and status of the CDS itself - is moreover not yet decided). Therefore there is no immediate compatibility between software packages running at the different institutes. An effort should therefore be made to produce a machine-independent software. This goal is worthwhile to be mantained even if a transition occurs towards a situation of hardware compatibility between the institutes of the SAX Consortium, as it is currently under discussion for the Italian Institutes. Three good reasons exist for this. The first reason concerns the past (that is, the present) software modules written in the current situation will have to be adapted to run on the new "unique" machines. The second reason concerns the present (of the SAX operational phase) : a problem of hardware compatibility may still exist for part of the Consortium, or at least for the scientific community. Moreover it may still be desirable to run some modules on large mainframes. The third reason concerns the future (beyond the SAX mission) : if we succeed in creating a "general purpose X-ray astronomy package", we would like to propagate it further on newer machines. A sound programming style may also help conversion to new standards (is Fortran 88 forthcoming ?).

The goal of machine-independence is basically achieved by avoidance of the use of machine- or operating system-dependent features (or enclosing them into isolated routine, if at all necessary), restricting oneself to a common subset of the language selected (ANSI Fortran 77), applying some limited restrictions to the format of the data to be exchanged between the insitutes.

In order to do this a set of rules and recommendations should be integrated into a "manual of style". As a preliminary step towards the preparation of such a manual, an investigation is necessary about the existing degree of compatibility between the various features of the Fortran languages available at the different hardware institutes.

Document for which no computer readable source exists any longer

Partial scan of original hardcopy (available on request) supplied

My original approach (November '85) was to include both the preliminary investigation (in the form of a cross-correlation between different Fortrans) and the set of rules in the same document. A somewhat different approach has been taken by M.Morini in his draft document of June '86 (defining a common set of rules first, and the machine-dependent prohibitions later). I presently (August '86) still stand in my original point-of-view, that a detailed presentation of the current situation is necessary, but have now decided that the preliminary investigation should be done in a separate document. I also would stress here, that a plain comparison of different Fortran manuals is not sufficient without a thorough practical experience on a particular machine. There are often features or nuances which are totally or partially undocumented, or site-dependent, and become known only with practice.

The present draft document therefore contains a description (in the order normally followed by Fortran manuals) of similarities and differences between the Fortran 77 implementations on the computers currently available at IFCTR Milano. The information contained in the quoted document by Morini, is taken into account where necessary, and is assumed to correspond to the implementation available at IFCAI Palermo. Any further information about implementations at other sites will be greatly appreciated.

Some other system dependent features, indirectly related to Fortran, (e.g. linking/loading programs, memory restrictions, file handling etc.) are described in further sections.

A separate draft document (in a more preliminary form) will attempt to put down some rules, as a consequence of the above investigations, and some general style recommendations.

The layout of both documents is also intended as a proposal for the layout of the final document to be produced by the data analysis working group.

2 : Cross correlation between Fortran languages

This section describes the similarities and differences between the Fortran 77 implementations available at IFCTR Milano, which are :

Hewlett Packard Fortran 7X Rev. 2401 (running on the HP 1000 F computer under the RTE 6 VM operating system, rev. 2440, with FMGR)

IBM VS Fortran release 4.1 (running on the IBM-compatible mainframe Siemens-Fujitsu 7.865 under the VM/SP (CMS) operating system, release 3.13).

For the IFCAI Palermo, site reference is made through the document by Morini to:

DEC VAX Fortran V. 4.0 (running on the DEC VAX 11/750 computer under the VMS operating system, V. ?.?)

Users of similar computer system are advised to check whether the above is consistent with the situation at their site (e.g. are they using CI instead of FMGR on an HP system ? are they using another VM/SP release on an IBM system ? etc.).

The layout of this section is intended to easily enable the insertion of notes about other computer systems. The text is broken into sub-sections, each one regarding a single language item (items are listed in an order similar to the one used by most Fortran manuals). Each subsection includes separate paragraphs about the different systems, where the same features are listed in a fixed order, with a clear indication whether full, limited or no compatibility exists. Cases of limited or no compatibility are indicated by the use of **boldface** script.

1. Character set

HP Upper case alphabetic characters (A-Z) are part of the Fortran set.
Numeric characters (0-9) are part of the Fortran set.
Lower case alphabetic characters (a-z) **are accepted** and translated to upper case (in normal Fortran statements). They are not translated in character and Hollerith constants (this is an extension of ANSI 77 standard).
The following non-alphanumeric characters are part of the Fortran set (common to HP and IBM) : space ( ) = + - * / ' . , " : $
The following non-alphanumeric characters are part of the Fortran

82. **Usage of Hollerith strings**

HP Hollerith values may be freely used if shorter than 8 character, else are allowed only in formats and as subroutine arguments. This may be required to pass arguments to older system-dependent routines, which do not accept character arguments.

IBM Hollerith values may be used only in DATA and format statements.

VAX NO info available.

83. **Suppression of line feed**

HP An **underscore** (as last character of a character string) in an output list inhibits the carriage return and line feed automatically generated at each end of record).

IBM There is **no known way** to inhibit carriage return and line feed at the end of record in an output list.

VAX The **$ format descriptor** is used to inhibit carriage return and line feed at the end of record in an output list.

3 : Cross correlation between file systems

In this section we give an overview of the file system and file i/o and management by Fortran programs for HP and IBM systems (see beginning of section 2 for details on current operating systems). No attempt is made here to reference the VAX case, due to lack of information. Boldface script is not used to evidentiate differences (actually most of the items are different indeed).

1. **Disk structure**

HP Each user may access a number of virtual disk areas (cartridges), identified by a logical unit number and/or a 2-character identifier.
No tree directories are supported under FMGR.

IBM Each user may access a number of virtual disk areas, identified to the system by a virtual address, and to the user by a single filemode letter.
No tree directories are supported.

2. **Disk access**

HP Each user may access in read/write private, temporary (pool), group and system cartridges. Group and system cartridges are shared among some (all) users. Access to some system cartridges is limited.

IBM Each user may access in read/write his own disks, and temporary disks. He may access in read only system disks, and other users' disk (if a link password is defined and an explicit link action is taken). Shared read/write access is subject to limitations.

3. **File names**

HP A file is identified under FMGR by a 6-character name. Additional specifications include a 2-char security code, the cartridge reference number, and a numeric file type (see below). No files with the same name may exist on the same cartridge, irrespective of the other specifications.
A file name may not begin with a numeric character. Non-alphanumeric characters in file names are generally allowed.

17. Spooling

Spooling is generally defined as a way of sending data to a shared slow device (generally a printer) by many users, enqueueing the data on a dedicated disk buffer. However system dependent peculiarities exist.

HP Spool files may be set-up via FMGR commands, or programmatically. They are later handled by SMP (Spool Management Program). FMGR or programmatic interfaces with SMP exist. Essentially spooling associates a logical unit number (user defined or assigned by the system) to a file (named by the user or assigned by the system from a pool) in a permanent way until explicitly closed (beyond a single program run). At closure the file may be purged or kept, released to user access, or outspooled to a designated device (generally, but not necessarily, a printer).

IBM Under VM/SP an user has spooled access to a **virtual** reader, a vritual punch and a virtual printer. Named and unnamed spool files may be transferred from and to the spool area, and managed by CP and CMS commands. The spool areas are mainly used as temporary stores, or for communication between different virtual machines (including remote nodes in Earnet). Spooling data to printers (more files may be chained in a single spool file) is generally accomplished transferring the data to the virtual reader of a dedicated virtual machine, which manages the physical printer.

4 : Cross correlation of system dependent features

In this section we give an overview of some system dependent features which may be relevant to usage and execution of Fortran programs for HP and IBM systems (see beginning of section 2 for details on current operating systems). No attempt is made here to reference the VAX case, due to lack of information. Boldface script is not used to evidentiate differences (actually most of the items are different indeed).

1. Internal representation of INTEGER*2 variables

HP 1 sign bit + 15 bit binary value

IBM 1 sign bit + 15 bit binary value

2. Internal representation of INTEGER*4 variables

HP 1 sign bit + 31 bit binary value

IBM 1 sign bit + 31 bit binary value

3. Internal representation of REAL*4 variables

HP **1 mantissa sign bit + 22 bit mantissa + 6 bit exponent + 1 exponent sign bit.**

IBM **1 sign bit + 7 bit exponent + 24 bit mantissa**

4. Internal representation of REAL*8 variables

HP **1 mantissa sign bit + 54 bit mantissa + 6 bit exponent + 1 exponent sign bit.**

IBM **1 sign bit + 7 bit exponent + 56 bit mantissa**

5. Memory size

HP HP 1000 computers use 16-bit addressing. The **maximum core space** which may be used by a simple program (i.e the maximum size of an extended background partition) is 32 pages (i.e. **65 kbytes**). This area includes code, data and system overheads. Ways to overcome this limitations are described below.