EDOC102

EUROGAM PROJECT

VIVITRON DATA ACQUISITION SYSTEM

Spectrum Database Structure

Edition 0.1
September 1990

Service Acquisition et Traitement de Donnees
Centre de Recherches Nucleaires
Strasbourg

Spectrum Database Structure

Jean Zen

September 1990

1 Introduction

This document outlines a general structure of saved spectrum files in a disk file system (eg. NFS). It is also intended to be applied to online spectra, if the online spectra memory can be mapped to a file.

A spectrum file consists of a header bloc and a counts bloc(s). The first coded in ASCII and the second in binary are contiguous and accessible in direct access mode (Fortran) or in stream mode (C).

A spectrum file may also contain many spectra (main spectrum + subrange spectra). A such file is simply obtained by appending as many header and counts blocs to the main spectrum blocs. In this case, a pointer index table is necessary for quick access to a subrange spectrum. The index table may be machine dependent due to the access method used but resides in a seperate file whose name differs in prefix with that of the spectrum file (on Unix machines if the prefix is ".", the index file will be hidden to user). The location of the index table on a separate file permits no limitation on the number of subrange spectra.

2 Header bloc (multiple of 512 bytes)

All unused (trailing) field must be filled with 'space' characters.

First line (byte 1 to 80):
2:16 run name
18:24 run number
26:40 spectrum name
42:48 spectrum number
50:68 date of the form NN-MMM-YY HH:MM:SS
70:80 spectrum type (keywords : 1D, 2D, 3D or 4D)
Second line (byte 81 to 160):
80+2:8 precision

keywords : BIT*1, BIT*2, BIT*4, BIT*8, BIT*16, BIT*32, INT*2, INT*4, REAL*4, REAL*8 or REAL*16 (REAL + some BIT precisions not recommanded for exchange)

80+10:16 X range
80+18:24 X base (counted starting from 0)
80+26:32 Y range (0 if not applicable)
80+34:40 Y base (0 if not applicable)
80+42:48 Z range (0 if not applicable)
80+50:56 Z base (0 if not applicable)
80+58:64 T range (0 if not applicable)

80+66:72 T base (0 if not applicable)
80+74:76 byte ordering (keywords : MSB or LSB)
80+78:80 reserved
Third line (byte 161 to 240) :
Efficiency calibration
number of coefficients followed by the coefficients (in free floating point format)
Fourth line (byte 241 to 320):
240+2:8 application id (keyword : APPLIC the only one applicable)
240+10:16 header length in bytes (multiple of 512)
240+18:32 experiment name
240+34:40 version number
240+42:50 beam energy (optional)
240+52:60 beam ion species (optional)
240+62:70 target species (optional)
240+72:80 for free use
Fifth line (byte 321 to 400) :
X calibration
number of coefficients followed by the coefficients (in free floating point format)
sixth line (byte 401 to 480) :
Y calibration
number of coefficients followed by the coefficients (in free floating point format)
seventh line (byte 481 to 512 or farther)
min,max counts (to be elaborated)

3 Counts array bloc (multiple of 512 bytes)

The array elements hold the channel counts in the standard Fortran array element ordering (the X range subscript varies most rapidly). The number of elements is derived from the specification in the header bloc and completed to a multiple of 512 bytes.

4 Spectra index table

For a spectrum file consisting of more than one spectrum, a separate index table file is used for quick access to a subrange spectrum. This table is suggested to be structured as follows :

1. number of the existing subrange spectra in the file (in integer*4)
2. number of bytes written (= next free byte address in integer*4)
3. byte address of the 1st subrange spectrum header bloc (in INT*4)
4. byte address of the 2nd subrange spectrum header bloc (in INT*4)

... ...

5 Spectrum access routines

All capital arguments refer to outputs returned by the calls (eg. IERs are returncodes).

5.1 create and delete spectrm

spdef (IDTAG, nsub, ndim, nxyz, nbyte, namsp, IER)

spdel (idtag, nsub, IER)

idtag = a communication variable to identify the spectrum file

nsub = 0 (zero) to create or delete a spectrum file otherwise to add a new subrange spectrum (in this case idtag must be known by a previous spdef or sptag call) or to delete a last subrange spectrum

ndim = number of the dimensions of the spectrum to be created (1, 2 or 3)

nxyz = a three element array denoting X, Y and Z ranges

nbyte = number of bytes used for the spectrum counts (1, 2 or 4)

namsp = a 14 character string for the name of spectrum (and file too if nsub=0)

5.2 obtain spectrum idtag or subrange

sptag (IDTAG, namsp, IER)

spsub (idatg, NSUB, namsp, IER)

nsub = subrange number returned given the name of a subrange spectrum

5.3 inquire spectrum attributes

spinq (idtag, nsub, NDIM, NXYZ, NBYTE, IER)

spnam (idtag, nsub, NAMSP, IER)

5.4 read and write counts

spread (idtag, nsub, iy, ny, iz, nz, IARR, IER)

spwrit (idtag, nsub, iy, ny, iz, nz, iarr, IER)

iarr = counts array to be read or written (array elements must be contiguous, a minimum of a full range of X dimension is read or written)

iy = requested base on Y dimension

ny = requested range on Y dimension

iz = requested base on Z dimension

nz = requested range on Z dimension

5.5 read and write annotation strings

Most of the Daresbury SDB routines are applicable but should have an additional 'nsub' argument and likely be named differently.

6 Related documents

Introducing the SDB spectrum access system, Sept 1985, NSF, Daresbury

Accessing spectra, Feb 1988, NSF, Daresbury

Norme spectres sur disquettes, May 1988, IPN/CSNSM, Orsay

Systeme de gestion des fichiers de spectres, Nov 1988, CRN, Strasbourg