Astrophysikalisches Institut Potsdam
Optische Instrumentierung
An der Sternwarte 16, D-14482 Potsdam, Germany

Integral Field Spectroscopy Data Format

– Draft –

Input to the

OPTICON 3D Spectroscopy Working Group

Version: 0.2

Martin M. Roth

mmroth@aip.de

Tel: +49 (0)331 7499-313
Fax: +49 (0)331 7499-436

November 29, 2001

Contents

1 Revision History

Version	Date	Changes	Author
0.1	27/11/01	first draft	mmr
0.2	29/11/01	minor corrections	bom

2 About this Document

This document is a first attempt to present in a detailed form the **Integral Field Spectroscopy Data Format** whose general layout was proposed by the the OPTICON 3D Spectroscopy Working Group during a meeting in Lyon on June 25-26, 2001.

3 Abbreviations and Acronyms

- DEC - Declination
- DICB - ESO Data Interface Control Board
- FITS - Flexibel Image Transport System
- FTS - Fourier Transform Spectrometer
- HDU - Header and Data Unit
- IFS - Integral Field Spectroscopy
- IFU - Integral Field Unit
- NIR - Near infrared
- NOST - Nasa / Science Office of Standards and Technology
- RA - Right Ascencsion
- TBV - to be verified
- TBR - to be requested
- TBD - to be defined
- WCS - World Coordinate System

4 Applicable Documents

[1] **A User's Guide for the Flexible Image Transport System (FITS)**
Version 4.0
April 14, 1997
NASA/GSFC Astrophysics Data Facility
(available at `http://heasarc.gsfc.nasa.gov/docs/heasarc/fits.html`)

[2] **Definition of the Flexible Image Transport System (FITS)**
Standard NOST 100-2.0
March 29, 1999
NASA/Science Office of Standards and Technology
(available at `http://heasarc.gsfc.nasa.gov/docs/heasarc/fits.html`)

[3] **Data Interface Control Document**
GEN-SPE-ESO-19400-794/1.1/0
November 25, 1997
ESO Data Interface Control Board (DCIB)

[4] **Report on the Meeting of the OPTICON 3D Spectroscopy Working Group held in Lyon, Observatoire de Lyon, June 25-26, 2001**
July 3, 2001
J. Walsh
(available at `http://www.aip.de/Euro3D`)

[5] **MIFS: A Mega Integral Field Spectrograph for the VLT**
Bacon et al. 2001
astro-ph/0109060
to appear in: *Workshop on Future VLT Instrumentation, eds J. Bergeron and G. Monnet, ESO, Astrophysics Symposia (Springer)*

[6] **A Million Element Integral Field Unit (MEIFU)**
Morris et al. 2001
astro-ph/0108130
to appear in: *Workshop on Future VLT Instrumentation, eds J. Bergeron and G. Monnet, ESO, Astrophysics Symposia (Springer)*

5 Overview

5.1 Purpose

This document is intended to provide a practical basis for the detailed discussion and subsequent approval of a data format for Integral Field Spectroscopy (IFS), henceforth **IFS Data Format**, to be proposed and introduced by the OPTICON 3D Spectroscopy Working Group (WG).

The general layout of the IFS Data Format was discussed and endorsed by the WG Meeting in Lyon on June 25-26, 2001. The goal is to finalize a definition before the end of 2001.

5.2 Scope

Integral field spectroscopy is an emerging observational technique with a growing number of new instruments. It has the virtue of providing spectra for many points on the sky, or, seen from another perspective, images at many different wavelengths at the same time. Similar techniques are common e.g. for radio astronomy. However, in some wavelength domains like in the optical/NIR and for reasons of technological differences, standard tools for the analysis of IFS data are not available to date, which is one of the reasons why IFS has not yet become a widespread method in the astronomical community.

The OPTICON 3D Spectroscopy Working group has identified the need to create such tools, among which the definition of a common data format will be a first step to allow the transport of data between different users, thus facilitating the exchange of expertise. As a second step, the Euro3D RTN which is expected to be formed in 2002, will develop software tools for the analysis of IFS data. The availability of the IFS Data Format will be a necessary prerequisite for these developments. The IFS Data Format will also be necessary to conduct the research projects which form the second major part of the Euro3D RTN work plan.

The possible applications of IFS encompass practically all areas of astrophysical research, from star formation in the milky way out to galaxies and the distant universe. A multi-wavelength approach (which incidently happens to define the very basis of IFS), e.g. the combination of UV-visual with NIR IFS data, also including radio, X-ray or other wavelength regimes, is likely to be demanded in the future. The spatial mapping of spectra from different methods of observing, and with different instrumental characteristics (e.g. orientation, resolution) should therefore be supported by a common data format, which eventually may develop into a widely accepted standard.

Taking into account several new instruments on 8-10m class telescopes which are in the process of commissioning and which will become operational soon, and given the ongoing efforts to develop a new generation of integral field instruments with largely increased scale and complexity, employing up to 1 million spatial elements (e.g. some IFUs proposed for 2nd Generation VLT instrumentation), a well-designed IFS Data Format will form an important basis for the future use of this technique.

This document lists the requirements and presents a definition of the IFS Data Format for subsequent discussion and iteration. The Appendix gives some examples of simple IDL scripts, which may be used to create sample FITS files to experiment with the IFS Data Format, and a first benchmark test of these routines.

6 Requirements

6.1 Completeness:

The IFS Data Format shall contain all the information which is necessary to retrieve the full content of the observational data.

6.2 Generality:

The IFS Data Format shall be simple and general enough to allow the distribution of IFS data from a wide variety of instruments. There shall be no specific assumptions, neither valid only for a subset of instruments, for a particular wavelength range, for certain observational methods, nor for any other constraints limiting the general use of the data format.

6.3 Reduced Data:

The IFS Data Format shall contain only fully reduced data, i.e. instrumental signatures must have been removed.

6.4 Preserving Original Information:

The IFS Data Format shall be designed to avoid any unnecessary loss of information which would be solely due to the requirement of fitting into the data format scheme.

6.5 Integral Field Data:

The IFS Data Format will contain spatially resolved spectral information for one or more contiguous two-dimensional fields on the sky. The spatial elements corresponding to these fields are defined by the technological method of how the spectra are being produced (e.g. fibers, lenslets, slicer, FTS, etc.). It is the spectra containing the physical information which must not be compromised by the method of fitting the data to a storage format. The IFS data format must also retain full information about the spatial mapping of the spectra on the sky to the best possible degree of accuracy and resolution.

6.6 Flexibility:

The IFS Data Format shall be flexible enough to allow upward compatibility with novel developments which may involve a much larger number of spectra ([5], [6]) than those of some existing instruments.

6.7 FITS:

The IFS Data Format shall comply with the FITS standard as described in "Definition of the Flexible image Transport System (FITS)" [2].

7 IFS Data Format Definition

7.1 Structure

The structure of the IFS Data Format consists of 4 mandatory, and optionally more FITS extensions.

In order to preserve the full information according to Requirement 5 (Section 6), the spectra will be stored as an image with `NAXIS=2` (Extension [1]), where each row represents a spectrum corresponding to a unique spatial element on the sky. The fastest varying index `NAXIS1` corresponds to wavelength. Each spatial element is identified by its row number `NAXIS2`. Additional information on the statistical error and on data quality for each spectral bin is provided in a similar kind of format in Extensions [2] and [3].

The spatial mapping on the sky at a certain reference wavelength `CRLAMBDA` is given by a position table (Extension [4]). Due to the presence of atmospheric refraction, the spatial mapping is, in the most general case, not constant but a function of wavelength. In those cases where the effects of atmospheric refrection result in non-negligible transformations, they must be accounted for in visualization and data analysis programmes.

The storage of data in datacubes with `NAXIS=3` is not supported because this method is not applicable for the most general cases, and since it is in contradiction to Requirements 2 and 5.

Further information may optionally be stored in Extensions [5], [6], ...

```
Extension
 Primary header             [0]
 Spectrum                   [1]
   Header
   2D Float image
 Statistical error          [2]
   Header
   2D Float image
 Data Quality               [3]
   Header
   2D Short integer image

 Table header
 Position table             [4]

 Table header                      | Optional
 Science table[1]           [5]    |

 Table header                      |
 Science table[2]           [6]    | Optional
 .
 .
```

7.2 Extension [0] : Primary Header

The primary HDU is reserved for the primary header and contains no data records. It is intended to provide general information about the origin and ownership of the data, and specific information describing observations, data reduction, and further processing history.

The **mandatory keywords** of the Primary Header are given in the following example:

```
SIMPLE  =                    T / IFS Data Format (Euro3D)
BITPIX  =                    8 /
NAXIS   =                    0 /
EXTEND  =                    T / File contains extensions
```

7.3 Extension [1] : Spectra

This HDU contains the spectra in the form of `NAXIS=2` images as described in Section 7.1. The order of spectra is arbitrary, but must be consistent with the numbering in the Position Table of Extension [4] (see below). The data are given as `float` values. Wavelength information is provided with keywords for the reference wavelength bin, wavelength increments, and units.

Header example:

```
XTENSION= 'IMAGE   '           / IMAGE extension
BITPIX  =                  -32 /
NAXIS   =                    2 /
NAXIS1  =                  998 /
NAXIS2  =                  254 /
PCOUNT  =                    0 / NO GROUP PARAMETERS
GCOUNT  =                    1 / ONE TABLE
CTYPE1  =           'ANGSTROM'/ wavelength scale along NAXIS1
CRVAL1  =              4197.80 / reference wavelength
CDELT1  =              2.63985 / wavelength increment per bin
CRPIX1  =                  1.0 / position of wavelength reference bin
EXTNAME = 'IFS2DSTK'           / IFS stacked spectra (2D image representation)
END
```

Mandatory keywords:
`XTENSION = IMAGE`: type of extension
`BITPIX = -32` : floating point data values
`NAXIS = 2` : 2D data array
`NAXIS1 = [INTEGER]`: number of wavelength steps
`NAXIS2 = [INTEGER]`: total number of spectra
`PCOUNT = 0`:
`GCOUNT = 1`: one table
`CTYPE1 = [STRING]`: wavelength scale along `NAXIS1` (e.g. `ANGSTROM, NM, MICRON,...`)
`CRVAL1 = [FLOAT]`: reference wavelength
`CDELT1 = [FLOAT]`: wavelength increment per bin
`CRPIX1 = [FLOAT]`: position of wavelength reference bin
`EXTNAME = 'IFS2DSTK'`: Extension name "*IFS 2D stacked spectra*"

7.4 Extension [2] : Statistical Error (Variance)

This HDU contains the statistical error for each pixel of the data array in Extension [1], respectively. The array formats must therefore be identical. The variance values must have been determined as part of the data reduction process. The data are given as `float` values. Wavelength information is not repeated in this header (TBV).

Header example:

```
XTENSION= 'IMAGE   '           / IMAGE extension
BITPIX  =                  -32 /
NAXIS   =                    2 /
NAXIS1  =                  998 /
NAXIS2  =                  254 /
PCOUNT  =                    0 / NO GROUP PARAMETERS
GCOUNT  =                    1 / ONE TABLE
EXTNAME = 'IFS2DVAR'           / IFS stacked spectra variance
END
```

Mandatory keywords:
`XTENSION = IMAGE`: type of extension (IMAGE).
`BITPIX = -32` : floating point data values
`NAXIS = 2` : 2D data array
`NAXIS1 = [INTEGER]` : number of wavelength steps
`NAXIS2 = [INTEGER]`: total number of spectra
`PCOUNT = 0`:
`GCOUNT = 1`: one table
`EXTNAME = 'IFS2DVAR'`: Extension name "*IFS stacked spectra variance*"

7.5 Extension [3] : Data Quality

This HDU contains a 16-bit integer as data quality indicator for each pixel of the data array in Extension [1], respectively. The formats must therefore be identical, except for the data values which correspond to `BITPIX = 16`. Individual detector flaws, or technical anomalies, detected cosmic ray hits, etc. are flagged with the data quality bits according to the table below. The Data Quality array is also used to mask out pixels that do not contain data, e.g. for instruments that do not produce spectra of equal spectral range for all spatial elements. Individual flags must have been set during the data reduction process. Wavelength information is not repeated in this header (TBV).

Header example:

```
XTENSION= 'IMAGE   '           / IMAGE extension
BITPIX  =                   16 /
NAXIS   =                    2 /
NAXIS1  =                  998 /
NAXIS2  =                  254 /
PCOUNT  =                    0 / NO GROUP PARAMETERS
GCOUNT  =                    1 / ONE TABLE
EXTNAME = 'IFS2DDQU'           / IFS stacked spectra data quality
END
```

Mandatory keywords:
`XTENSION = IMAGE`: type of extension (IMAGE).
`BITPIX = 16` : 16-bit integer data values
`NAXIS = 2` : 2D data array
`NAXIS1 = [INTEGER]` : number of wavelength steps
`NAXIS2 = [INTEGER]` : total number of spectra
`PCOUNT = 0`:
`GCOUNT = 1`: one table
`EXTNAME = 'IFS2DDQU'`: Extension name "*IFS stacked spectra data quality*"

Data Quality Flags:

bit	flag value	quality condition
0	1	
1	2	affected by isolated hot pixel/column
2	4	affected by hot pixel area (> 5 sigma median dark)
3	8	affected by dark pixel (CCD trap)
4	16	affected by dark pixel area (< 20% of average sensitity)
5	32	affected by saturated detector pixel
6	64	affected by bad pixel from calibration frame
7	128	affected by cosmic ray removal
8	256	affected by ghost/stray light at >10% intensity level
9	512	affected by electronic pickup noise
10	1024	
11	2048	
12	4096	
13	8192	no data value (inactive detector area, mosaic gap)
14	16384	no data value (outside of spectral range)

7.6 Extension [4] : Position Table

This HDU serves the purpose of mapping each spectrum at a given wavelength to a position on the sky. This is accomplished by means of a table which lists spatial element positions (e.g. fiber, lenses, fraction of a slicer, ...) in an instrument-based reference frame, which may be transformed to WCS conventions by using additional header information. The mapping is not immediately given in WCS values because in the most general case differential atmospheric refraction leads to non-trivial and time-dependent shifts as a function of wavelength whose correction requires spatial rebinning and is therefore incompatible with the stacked spectra format.

In those cases where atmospheric refraction is negligible, a proper selection of the instrumental coordinate system (e.g. RA, DEC, arcsec scale) will map naturally into WCS with no correction necessary.

The Position Table contains 4 columns, and as many rows as there are spectra. Column 1 contains the 2D spectrum row number, corresponding to a spatial element, whose instrumental coordinates are listed in Columns 2 and 3. Column 4 contains a group index (1,...,n) which collectively identifies n groups of spectra belonging to the same contiguous IFU(s), which is (are) destinct from another IFU, or several other IFUs. This identification is useful to allow a visualization tool or data analysis software to automatically distinguish between different deployable IFUs, or to separate the main IFU from special purpose spectra (e.g. calibration spectra). When there is only one contiguous set of spectra in a single IFU, this parameter must be set to 1 for all rows.

The transformation of the instrumental coordinate system to WCS is performed through keywords `CRVAL1/2`, `CDELT1/2`, `CTYPE1/2`, `CROTA1/2` according to the explanation in the Data Interface Control Document (reference [3]), Section 4.2, page 18.

Along with the positional information, details of the spatial sampling may be optionally communicated through keywords `PIXTYPE`, `PIXSIZE1`, `PIXSIZE2`, and `PIXANGLE`. `PIXTYPE` may take the values `SQUARE`, `RECTANG`, `HEXAGON`, `CIRCLE`, `POLYGON`, `SPECIAL`, encoding square, rectangular, hexagonal, circular, and polygon shapes, respectively. `POLYGON` requires an additional table to list the coordinates for each spatial element individually. `SPECIAL` is reserved to allow for the encoding of mixed shapes or more complicated arrangements, and also requires one or more additional tables to hold this information. Any details of such peculiar modes are beyond the scope of this document.

The size of spatial elements may be given by `PIXSIZE1`, and also `PIXSIZE2` where appropriate. See Fig. 1 for a description.

The orientation of non-symmetrical spatial elements is encoded through `PIXANGLE`, giving the angle from the instrumental coordinate system X axis to the major axis of the spatial element, see Fig. 2.

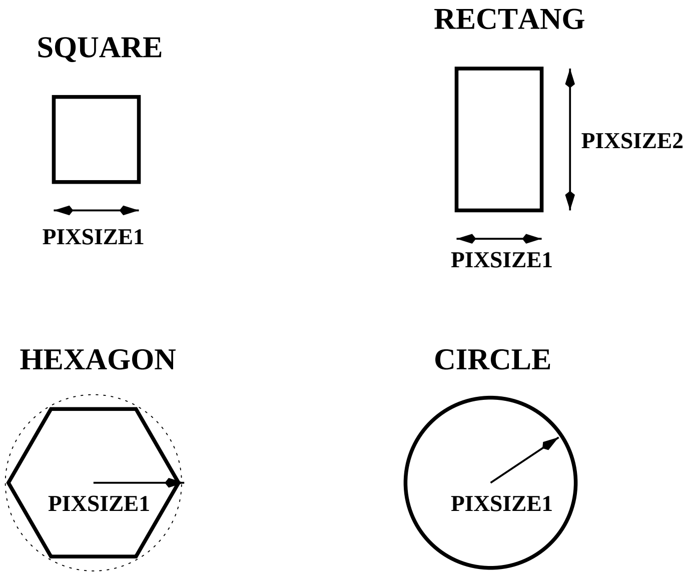

Figure 1: Illustration of PIXTYPE='SQUARE', 'RECTAN', 'HEXAGON', 'CIRCLE'. The fill-factor and details of the sampling for a given seeing PSF can be derived from this kind of information.

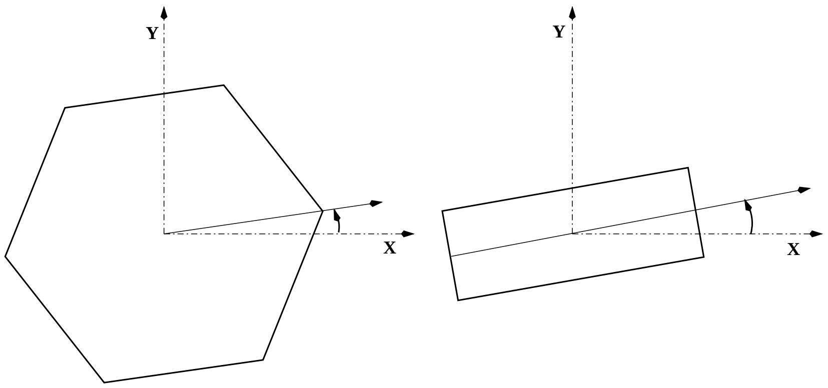

Figure 2: Encoding the orientation of non-symmetrical spatial elements with PIXANGLE.

Simple header example:

```
XTENSION= 'BINTABLE'           /IFS Data Format Position Table
BITPIX  =                    8 /
NAXIS   =                    2 /Binary table
NAXIS1  =                   16 /Number of bytes per row
NAXIS2  =                  254 /Number of rows
PCOUNT  =                    0 /Random parameter count
GCOUNT  =                    1 /Group count
TFIELDS =                    4 /Number of columns
EXTNAME = 'IFSSEPOS'           /IFS spatial element position
TFORM1  = '1J      '           /Integer*4 (long integer)
TTYPE1  = 'ROWNUM  '           / IFS spatial element number (row in 2D spectra)
TFORM2  = '1E      '           /Real*4 (floating point)
TTYPE2  = 'XPOS    '           /IFS spatial element X position
TFORM3  = '1E      '           /Real*4 (floating point)
TTYPE3  = 'YPOS    '           /IFS spatial element Y position
TFORM4  = '1J      '           /Integer*4 (long integer)
TTYPE4  = 'IFU_NO  '           /IFU group number
...
END
```

Mandatory keywords:
XTENSION= 'BINTABLE' : type of extension (binary table)
BITPIX = 8: binary table
NAXIS = 2: table with rows and columns
NAXIS1 = [INTEGER]: number of bytes per row
NAXIS2 = [INTEGER]: number of rows
PCOUNT = 0: always 0
GCOUNT = 1: always 1
TFIELDS = 4: number of columns
EXTNAME = 'IFSSEPOS': extension name "*IFS spatial element position*"
TFORM1 = [FORMAT-STRING]: format = Integer*4 (long integer)
TTYPE1 = [INTEGER]: IFS spatial element row number (in 2D spectra)
TFORM2 = [FORMAT-STRING]: format = Real*4 (floating point)
TTYPE2 = 'XPOS ': IFS spatial element X position
TFORM3 = [FORMAT-STRING]: format = Real*4 (floating point)
TTYPE3 = 'YPOS ': IFS spatial element Y position
TFORM4 = [FORMAT-STRING]: format = Integer*4 (long integer)
TTYPE4 = [INTEGER]: IFU group number
CRVAL1 = [FLOAT]: IFS coordinate system X reference (origin)
CRVAL2 = [FLOAT]: IFS coordinate system Y reference (origin)
CDELT1 = 2.7777E-4: IFS coordinate system X increments (1 arcsec)
CDELT2 = 2.7777E-4: IFS coordinate system Y increments (1 arcsec)
CTYPE1 = 'RA--TAN': WCS projection method
CTYPE2 = 'DEC-TAN': WCS projection method
CROTA1 = [FLOAT]: rotation angle
CROTA2 = [FLOAT]: rotation angle

Additional keywords:
PIXTYPE = 'SQUARE': spatial element = square, measuring PIXSIZE1 on a side
or: PIXTYPE = 'RECTANG': rectangular, PIXSIZE1×PIXSIZE2
or: PIXTYPE = 'HEXAGON': hexagonal, inscribing circle of radius PIXSIZE1
or: PIXTYPE = 'CIRCLE': circular with radius PIXSIZE1
or: PIXTYPE = 'POLYGON': polygon, defined in separate table (Extension [5])
or: PIXTYPE = 'SPECIAL': special definition in separate table(s) (Extension [5], ...)
PIXSIZE1 = [FLOAT]: size parameter of spatial element
PIXSIZE2 = [FLOAT]: size parameter of spatial element
PIXANGLE = [FLOAT]: angle of major axis of spatial element wrt to X axis.
EQUINOX = [FLOAT]: epoch of observation, fraction of a year
PARANGLE = [FLOAT]: parallactic angle, needed to correct for atmospheric refraction
AIRMASS = [FLOAT]: airmass, needed to correct for atmospheric refraction
CRLAMBDA = [FLOAT]: reference wavelength for mapping to WCS
ADSPCORR = [logical]: 'T' if atmospheric dispersion has been corrected, 'F' otherwise. May be set 'T' for certain instruments, e.g. equipped with an optical Atmospheric Dispersion Compensator, or when the effect of atmospheric dispersion is negligible, e.g. due to a small wavelength range.

7.7 Extension [5] : Science Table

TBD

A Simple IDL script: read_IFS.pro

This simple script illustrates the minimal amount of required coding under IDL for reading a file according to the IFS Data Format definition as given above. This script makes use of routines from the **IDL Astronomy User's Library**, provided by NASA/GSFC.

Note that the astrolib FITS I/O routines are extensively supported and up-to-date, see: http://idlastro.gsfc.nasa.gov/news.html.

```
fn='~/idl/JOBS/testfile.fits'

; --- Primary Header [Extension 0] ------------------------------------------
r = readfits(fn,hdr_0,exten_no=0)

; --- 2D Stacked Spectra [Extension 1] ---------------------------------------
r_img = readfits(fn,hdr_1,exten_no=1)


; --- 2D Variance [Extension 2] ----------------------------------------------
r_var = readfits(fn,hdr_2,exten_no=2)


; --- 2D Data Quality [Extension 3] ------------------------------------------
r_dqu = readfits(fn,hdr_3,exten_no=3)


; --- Binary Table Spatial Element Positions [Extension 4] -------------------
fxbopen,lu,fn,4,hdr_4
fxbhelp,lu
fxbread,lu,num,'ROWNUM'
fxbread,lu,x,'XPOS'
fxbread,lu,y,'YPOS'
fxbread,lu,ifuno,'IFU_NO'
fxbclose,lu

help,r_img,r_var,r_dqu,num,x,y,ifuno

END
```

B Simple IDL script: `write_IFS.pro`

This is another script, illustrating an example of how to write a file according to the IFS Data Format definition as given above. This script makes extensive use of routines from the **IDL Astrolib Library**, provided by NASA/GSFC.

```
fn = '~/idl/JOBS/testfile.fits'; filename to be written

; It is assumed that we have the following arrays available from previous
; I/O or processing, ready to be written to disk:
;
; img: 2D image of stacked spectra [FLOAT]
; hdr_in: header from original raw data input file
; varimg: 2D image of variance values, corresponding to img [FLOAT]
; dqimg: 2D image of data quality flags, corresponding to img [SHORT]
; no: 1D array of row numbers of spectra [LONG]
; xpos: 1D array of spatial element X positions [FLOAT]
; ypos: 1D array of spatial element Y positions [FLOAT]
; ifu_gr: 1D array of spatial element IFU group number [LONG]e


@fxbintable

; --- make primary header [EXTENSION 0] -------------------------------------
fxhmake,hdr0,/EXTEND,/DATE ; create basic header
hdr_copy,hdr_in,hdr0,list ; copy selected keywords from input data file
fxwrite,fn,hdr0 ; write Extension [0]


; --- write 2D stacked spectra [EXTENSION 1] --------------------------------
mkhdr,hdr1,img,/IMAGE ; create basic header
fxaddpar,hdr1,'EXTNAME','IFS2DSTK',' IFS stacked spectra (2D image representation)'
writefits,fn,img,hdr1,/APPEND


; --- write 2D variance [EXTENSION 2] ---------------------------------------
mkhdr,hdr2,varimg,/IMAGE
fxaddpar,hdr2,'EXTNAME','IFS2DVAR',' IFS stacked spectra variance'
writefits,fn,varimg,hdr2,/APPEND


; --- write 2D data quality [EXTENSION 3] -----------------------------------
mkhdr,hdr3,dqimg,/IMAGE
fxaddpar,hdr3,'EXTNAME','IFS2DDQU',' IFS stacked spectra data quality'
writefits,fn,dqimg,hdr3,/APPEND
```

```
; --- write spatial element position table [EXTENSION 4] --------------------
s = size(img)
n = s(2)
fxbhmake,hdrbt1,n,'IFSSEPOS','IFS spatial element position'
fxbaddcol,index,hdrbt1,no(0),'ROWNUM',' IFS spatial element number (row in 2D spectra)'
fxbaddcol,index,hdrbt1,xpos(0),'XPOS','IFS spatial element X position'
fxbaddcol,index,hdrbt1,ypos(0),'YPOS','IFS spatial element Y position'
fxbaddcol,index,hdrbt1,ifu_gr(0),'IFU_NO','IFU group number'

fxbcreate,unit,fn,hdrbt1

for i=0,n-1 do begin
fxbwrite,unit,no(i),1,i+1
fxbwrite,unit,xpos(i),2,i+1
fxbwrite,unit,ypos(i),3,i+1
fxbwrite,unit,ifu_gr(i),4,i+1
endfor

fxbhelp,unit
fxbfinish,unit

; --- finished. --------------------------------------------------------------


print,'----------------------------------------------------------'
print,'printing INFO on this FITS file:
print,'----------------------------------------------------------'
fits_help,fn

END
```

C Benchmark using `read_IFS` and `write_IFS`

A simple benchmark test has been carried out in order to provide an impression of typical read and write times for IFS data of different sizes.

The tested formats include MPFS (15x16 spatial elements + 14 calibration spectra, 998 spectral elements), PMAS (16x16, 2048) in its mosaic camera configuration, assumed VMOS (80x80, 550), and 1/10 of MIFS (proposed 2nd Generation VLT instrument) with 9000 spatial elements of 1600 spectral pixels.

The computer used for this test was a laptop with Pentium-III processor at 497 MHz, 256KB cache, and with 256MB Memory. Note that the write routine includes some overhead for the simulation of variance through a `sqrt` operation.

IFU	File Size [bytes]	write	read	spatial	spectral
MPFS	2 571 840	<1s	<1s	998	254
PMAS	5 279 040	1s	1s	16x16	2048
VMOS	35 449 920	12s	3s	80x80	550
0.1x MIFS	144 342 720	97s	58s	9000	1600