ESO - EUROPEAN SOUTHERN OBSERVATORY

EUROPEAN SOUTHERN OBSERVATORY
Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

FITS format description for pipeline products with data, error and data quality information

VLT-SPE-ESO-19500-5667

Issue 1.0
11 July 2012

Prepared:	Martin Kümmel		
	Pascal Ballester		
	Harald Kuntschner		
	Name	Date	Signature
Approved:	Pascal Ballester		
	Name	Date	Signature
Released:	Pascal Ballester		
	Name	Date	Signature
Released:			
	Name	Date	Signature

Table of Contents

1. LIST OF ACRONYMS

VLT	Very Large Telescope
FITS	Flexible Image Transport System
CASA	Common Astronomy Software Applications
DICD	Data Interface Control Document
HDU	Header Data Unit
WCS	World Coordinate System
IFU	Integral Field Unit

2. INTRODUCTION AND SCOPE

Many of the current and the future ESO pipelines for VLT instruments deliver as the final products data with associated pixelwise error information and information on the data quality. This document discusses and proposes formats to store this kind of data in a FITS file such that other applications or tools can read and process them properly. An example for such an application is the ESO 3D viewer, which is currently being developed as part of the CASA viewer to display 3D cubes (two spatial plus one spectral dimension). Such a format definition does currently neither exist in the FITS standard (see http://fits.gsfc.nasa.gov/fits_standard.html) nor in the "Data Interface Control Document" (DICD, see http://arcdev.eso.org/dicb/DICB_internal/) of ESO. The proposed FITS format defines only the science data output of ESO pipelines (see Remark 1).

3. OVERVIEW ON THE DATA

The data products described here consist of the processed data values (called "science data" hereafter) and the error and data quality as additional or supporting information. Science data and their associated error and data quality information shall be stored in different Header Data Units (HDU) of the same FITS file (see Remark 2). The auxiliary data (error and data quality) are optional. If existing, the HDU with the auxiliary data must have the identical dimension as the according science data. Also the error and data quality HDU's must contain a World Coordinate System (WCS) which is identical to their according science data HDU.

4. THE ERROR

The pixel-by-pixel errors shall be used in applications such as the ESO 3D viewer to determine, via error propagation, the error for results from simple processing of the 3D data, such as extracted spectra or 2D reconstructed images. The error values can be given as different error types. Possible error types are:

- "MSE": mean squared error
- "RMSE": root mean square error
- "INVMSE": inverse mean squared error
- "INVRMSE": inverse root mean square error

An error HDU identifies its error type with the value of a FITS keyword as described in Sect. 6. The recommended error type is MSE.

5. THE DATA QUALITY INFORMATION

The data quality information offers the possibility to further qualify the data for every pixel. The data quality information is stored in a HDU with the identical dimension and WCS as the data HDU. There exist four different types of data quality HDU's, which are identified with the FITS keyword as described in Sect. 6:

- "MASKZERO": A data quality HDU of this type describes a pixel mask of any numeric FITS type (16-bit integer or 16-bit float). Good pixels have the value 0, all other values mark bad pixels;
- "MASKONE": A data quality HDU of this type describes a pixel mask of any numeric FITS type (16-bit integer or 16-bit float). Good pixels have the value 1, all other values mark bad pixels;
- "FLAG32BIT": As described in the Euro3D data format (http://www.aip.de/Euro3D/E3D/). Such a data quality HDU contains 32-bit integer values, and each bit can be set to describe a certain pixel defect (e.g. "no data", "outside the allowed range"). The decision on whether a

pixel is good is made via the bitwise operation "AND" of a pixel value with a mask value stored in a FITS keyword QUALMASK.

- "FLAG16BIT": As in the Euro 3D case, however with a different list of pixel defects (see Appendix A) and 16-bit integer values. The decision on whether a pixel is good is made via the bitwise operation "AND" of a pixel value with a mask value stored in a FITS keyword QUALMASK.

Pixel masks ("MASKZERO" and "MASKONE") can be used and understood rather intuitively. In contrast to this simple scheme, a more differentiated approach with 32-bit or 16-bit integer data quality values and quality mask value ("FLAG32BIT" and "FLAG16BIT") allows to select precisely which defects are considered to be serious and which are minor, such that the data is still useful. The data quality flagging in the Euro 3D format would allow to address 32 different defects, and it is possible to reject or accept each defect by adjusting the keyword QUALMASK.
In the fully processed data considered here, an individual pixel value often contains interpolated input from various original detector pixels. As a consequence, most of the pixels should be good, and a direct mapping of detector defects, e.g. "bad pixel", "hot pixel", for any individual pixel is often no longer possible. For the data quality type "FLAG16BIT" we have, starting from the list of defects defined in the Euro 3D format, extracted and compiled (see Appendix A) a list of defects that tend to be clustered and thus could likely be present even in interpolated, fully processed data. This reduced list of defects, including some deficiencies defined by the user, can easily be covered by a 16-bit number, which has the advantage to save, with respect to the full Euro 3D format, disk space and processing time.

6. ORGANIZATION OF THE FITS FILE

The science data plus their associated error and data quality information shall be stored in the HDU's of a single FITS file. If the file contains several HDU's, the primary HDU of the FITS files shall not contain data (see Remark 3). It is allowed to store several sets of HDU's with associated science data, error and data quality in a single FITS file.
If there are several sets in a FITS file, it must be possible to derive the identification of the set (such as the chip number or the IFU number) either from the extension name (keyword EXTNAME) or the combination of extension name and extension version (keyword EXTVER) of each science data HDU (see Remark 3). It is recommended that the identification is also part of the error and data quality extension names (see Appendix B). The order of the HDU's is recommended to be science data, error, data quality for each set representing the data of an IFU unit. However also a random order or the sorting according to function (e.g. all science data HDU's followed by all error HDU's followed by all data quality HDU's) shall be possible.

7. IDENTIFICATION OF DATA SETS WITH FITS KEYWORDS

Any application that uses science data with errors and data quality information must be able to identify within the list of available HDU's which extensions form a set and which role (science data/error/data quality) each HDU has. This information is provided in the FITS keywords of the various HDU's.
The FITS keywords:

- identify the type of every HDU;
- provide further type information for an error or data quality HDU;
- point to the HDU with the complementary information.

We propose to use a FITS keyword naming convention which was introduced in the high energy community (http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/docs/ofwg_recomm/r8.html). In order to specify the keyword convention, every HDU contains the fixed keywords:

- HDUCLASS="ESO"
- HDUDOC="DICD"
- HDUVERS="DICD version 6"

Further information is provided in hierarchical keywords HDUCLAS**n** with the index number **n**. For the various HDU types, these keywords contain:

HDU type	HDUCLAS1	HDUCLAS2	HDUCLAS3
science data	"IMAGE"	"DATA"	
error	"IMAGE"	"ERROR"	"MSE"
			"RMSE"
			"INVMSE"
			"INVRMSE"
data quality	"IMAGE"	"QUALITY"	"MASKZERO"
			"MASKONE"
			"FLAG32BIT"
			"FLAG16BIT"

In addition, specific keywords point to the HDU with the complementary information:

- in a science HDU:
  - ERRDATA "errext": identifies the extension name of the corresponding error HDU (if existing)
  - QUALDATA "dqext": identifies the extension name of the corresponding data quality HDU (if existing)
- in an error HDU:
  - SCIDATA "sciext": identifies the extension name of the corresponding science data HDU
  - QUALDATA "dqext": identifies the extension name of the corresponding data quality HDU (if existing)
- in a data quality HDU:
  - SCIDATA "sciext": identifies the extension name of the corresponding science data HDU
  - ERRDATA "errext": identifies the extension name of the corresponding error HDU (if existing)

Providing the full information on associated HDU's in each individual HDU replicates redundant information. However then each HDU possesses sufficient information to group itself in the global context. Appendix B shows an overview of a FITS file following this scheme. If the extension version keyword is used for the unique identification of an HDU (not done at ESO), see Remark 4.

8. REMARKS

- **Remark 1:** While the FITS format proposed here can certainly be applied to the final output products from all instruments, there might be cases where this is not appropriate. An example for such an case would be output products where the data quality information is shared by many sets of science data. Then the format proposed here would require the replication of identical data quality information for each science data set. Similarly, the "data-set" concept

of Phase 3 data products (see GEN-SPE-ESO-33000-5335) goes much beyond the format described here and allows more information to be included also in separate files.

- **Remark 2:** In the format proposed here the science data, the error and data quality are stored in the same FITS file, which is different from what some ESO pipelines currently deliver. E.g. FLAMES/ARGUS has a file for the data and another for the errors. The advantage of our approach is that the storage of the entire data (science+error+data quality) in a single file provides coherent data sets. It also allows a convenient access for applications such as the ESO 3D viewer. This avoids interactive input from the user, such as "1. load science data from file A, 2. load error from file B, 3. load data quality from file C.
- **Remark 3:** In the format proposed here it is mandatory that each HDU with data can be identified via extension name and, if necessary, extensions version (keywords `EXTNAME` and `EXTVER`). Consequently, if the primary HDU does contain data these keywords need to be set as well. While primary HDU's with extension name are not very popular and even not possible to be generated with the current (Oct. 2011) version of cpl, it is allowed in the FITS standard (see Sect. 4.4.2.6 in http://fits.gsfc.nasa.gov/fits_standard.html).
- **Remark 4:** If the FITS files use the extension name plus extension version (keyword `EXTVER`) to uniquely identify the HDU's, the extension version shall be appended to the extension name in the keywords that point to the HDU's with complementary information (e.g. `SCIDATA "sciext,1"; ERRDATA "errext,1"`).

APPENDIX A. LIST OF PIXEL DEFECTS FOR QUALITY TYPE "FLAG16BIT"

Bit #	flag_value	quality_condition
0	0	good data
1	1	affected by telluric feature (corrected)
2	2	affected by telluric feature (uncorrected)
3	4	ghost/stray light at >10% intensity level
4	8	saturated data value
5	16	bad pixel (general)
6	32	Na-laser notch filter affected
7	64	user defined
8	128	...
9	256	...
10	512	...
11	1024	...
12	2048	...
13	4096	...
14	8192	missing data
15	16384	outside data range

APPENDIX B. FITS EXAMPLE

The file FITS_templatesIII.fits contains the following extensions:

```
--> catfits FITS_templatesIII.fits
EXT#  FITSNAME      FILENAME              EXTVE DIMENS       BITPI
OBJECT
0     FITS_template                                          8
1       IMAGE       IFU1.SCI                    30x30x30     -32
2       IMAGE       IFU1.ERR                    30x30x30     -32
3       IMAGE       IFU1.DQ                     30x30x30     16
4       IMAGE       IFU2.SCI                    30x30x30     -32
5       IMAGE       IFU2.ERR                    30x30x30     -32
6       IMAGE       IFU2.DQ                     30x30x30     16
7       IMAGE       IFU3.SCI                    30x30x30     -32
8       IMAGE       IFU3.DQ                     30x30x30     -32
```

The first data extension has the keywords:

```
--> imhead FITS_templatesIII.fits[IFU1.SCI] lo+
FITS_templatesIII.fits[IFU1.SCI][30,30,30][real]:
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [30,30,30], length of user area 1377
s.u.
Pixel file "FITS_templatesIII.fits" [ok]
PCOUNT  =                    0 / number of parameters
GCOUNT  =                    1 / number of groups
EXTNAME = 'IFU1.SCI'           / Extension name
HDUCLASS= 'ESO'                / Identification of the data format
HDUDOC  = 'DICD'               / Document describing the data
format
HDUVERS = 'DICD version 6'     / Specific version of the document
HDUCLAS1= 'IMAGE   '           / Image data format
HDUCLAS2= 'DATA    '           / Identification as science
extension
ERRDATA = 'IFU1.ERR'           / Name of its error extension
QUALDATA= 'IFU1.DQ '           / Name of its data quality extension

        / World coordinate system start:
CRPIX1  =                  5.0 / Reference pixel
CRPIX2  =                  5.0 / Reference pixel
.
.
```

The keyword ERRDATA points to the extension 'IFU1.ERR' as the associated error array, which has the keywords:

```
--> imhead FITS_templatesIII.fits[IFU1.ERR] lo+
FITS_templatesIII.fits[IFU1.ERR][30,30,30][real]:
No bad pixels, min=0., max=0. (old)
```

```
Line storage mode, physdim [30,30,30], length of user area 1418
s.u.
Pixel file "FITS_templatesIII.fits" [ok]
PCOUNT  =                    0 / number of parameters
GCOUNT  =                    1 / number of groups
EXTNAME = 'IFU1.ERR'           / Extension name
HDUCLASS= 'ESO'                / Identification of the data format
HDUDOC  = 'DICD'               / Document describing the data
format
HDUVERS = 'DICD version 6'     / Specific version of the document
HDUCLAS1= 'IMAGE   '           / Image data format
HDUCLAS2= 'ERROR   '           / Identification as science
extension
HDUCLAS3= 'MSE     '           / Error type: Mean Squared Error
SCIDATA = 'IFU1.SCI'           / Name of its data extension
QUALDATA= 'IFU1.DQ '           / Name of its data quality extension

        / World coordinate system start:
CRPIX1  =                  5.0 / Reference pixel
CRPIX2  =                  5.0 / Reference pixel
.
.
```

In both, the data extension and the error extension, the extension `'IFU1.DQ'` is identified as the associated data quality extension, which has the keywords:

```
--> imhead FITS_templatesIII.fits[IFU1.DQ] lo+
FITS_templatesIII.fits[IFU1.DQ][30,30,30][short]:
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [30,30,30], length of user area 1458
s.u.
Pixel file "FITS_templatesIII.fits" [ok]
PCOUNT  =                    0 / number of parameters
GCOUNT  =                    1 / number of groups
EXTNAME = 'IFU1.DQ '           / Extension name
HDUCLASS= 'ESO'                / Identification of the data format
HDUDOC  = 'DICD'               / Document describing the data
format
HDUVERS = 'DICD version 6'     / Specific version of the document
HDUCLAS1= 'IMAGE   '           / Image data format
HDUCLAS2= 'QUALITY '           / Identification as science
extension
HDUCLAS3= 'FLAG16BIT'          / Data quality type
QUALMASK=                16383 / Mask number to identify bad pixels
SCIDATA = 'IFU1.SCI'           / Name of its data extension
ERRDATA = 'IFU1.ERR'           / Name of its error extension

        / World coordinate system start:
CRPIX1  =                  5.0 / Reference pixel
CRPIX2  =                  5.0 / Reference pixel
.
.
```