ESO Phase 3 automatic data validation: Groovy-based tool to assure the compliance of the reduced data with the Science Data Product Standard

L. Mascetti[a], V. Forchì[b], M. Arnaboldi[b], N. Delmotte[b], A. Micol[b], J. Retzlaff[b], and S. Zampieri[b]

[a]TERMA GmbH, Europahaus, Europaplatz 5, 64293 Darmstadt, Germany
[b]European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748 Garching, Germany

ABSTRACT

The ESO Phase 3 infrastructure provides a channel to submit reduced data products for publication to the astronomical community and long-term data preservation in the ESO Science Archive Facility. To be integrated into Phase 3, data must comply to the ESO Science Data Product Standard regarding format (one unique standard data format is associated to each type of product, like image, spectrum, IFU cube, etc.) and required metadata. ESO has developed a Groovy based tool that carries out an automatic validation of the submitted reduced products that is triggered when data are uploaded and then submitted. Here we present how the tool is structured and which checks are implemented.

Keywords: ESO Phase 3, reduced products, Data Products Standard, Science Archive, data validation, quality control, Groovy

1. INTRODUCTION

Phase 3[*] denotes the process of preparation, validation and ingestion of science reduced products for storage in the ESO Science Archive Facility and subsequent publication to the scientific community.

According to the ESO's policies governing Phase 3, returning reduced data to ESO is mandatory for ESO Public Surveys[1] and for ESO Large Programmes. For other ESO programmes there is no obligation, but PIs are invited to take advantage of Phase 3.

To ensure the successful integration of the new products into the ESO Archive, the data have to comply with the ESO Science Data Products Standard (SDPS)[†], a document that defines the structure and data format of reduced products starting from high-level requirements down to the definition of individual metadata items. The required FITS keywords are relevant for data characterisation, quality and processing provenance to trace back the original raw data. It also defines how to encode ancillary data files associated to the science products. The data provider is in charge of preparing the products according to that standard and then submit them to ESO.

Before the submitted data can be archived and published, they undergo a validation process, covered in details in the submitted contribution by N. Delmotte.[2] Part of the validation process is implemented in the Phase 3 software and automatically executed once the data have been uploaded to the ESO FTP server and it is the focus of this contribution. It performs a set of tests for the presence of mandatory keywords, type of such keywords, etc. After this step is successful an in-depth content validation is performed by the Archive Science Group (ASG) at ESO. More details are presented in Section 4. When the validation process certifies that the data are compliant, they can be archived and published. Hence, they become accessible to the community via the dedicated query forms as shown in Fig. 1.

Further author information: send correspondence to Laura Mascetti, Vincenzo Forchì or Archive Science Group (ASG)
Laura Mascetti, E-mail: lmascett@partner.eso.org
Vincenzo Forchì, E-mail: vforchi@eso.org
ASG, E-mail: usd-help@eso.org, subject: Phase 3

[*] http://www.eso.org/sci/observing/phase3.html
[†] http://www.eso.org/sci/observing/phase3/p3sdpstd.pdf

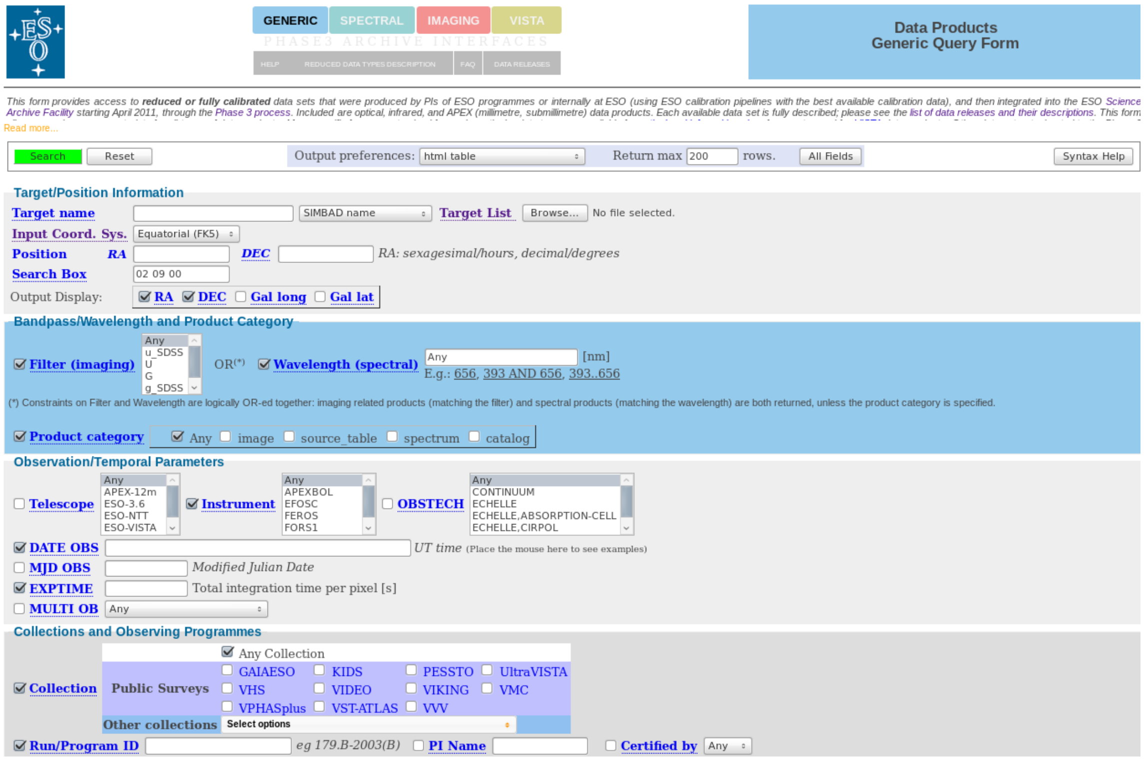

Figure 1. Thanks to the definition of the standard Phase 3 keywords, the data can be seamlessly accessed from a unique query form http://archive.eso.org/wdb/wdb/adp/phase3_main/form, independently of their original science programme.

2. DATA MODEL

The following data types are covered by the SDPS:

- **1-D Spectra**, `PRODCATG='SCIENCE.SPECTRUM'`[3]
  in binary table format: one primary header and one single extension (compliant to IVOA Spectral Data Model).
  No data in the primary HDU → `NAXIS=0 / Length of data axes`.
  Support for 2-D spectral frames as ancillary files.

- **Images**, `PRODCATG='SCIENCE.IMAGE'` or `'SCIENCE.MEFIMAGE'`
  Astrometrically and photometrically calibrated FITS images with associated confidence/weight maps; quality parameters required are for example limiting magnitudes (`ABMAGLIM` keyword) and PSF characterisation (`PSF_FWHM`).
  Single images are stored in the primary HDU.
  Support for multiple images stored in Multi-Extension FITS format (MEF image).

- **Sub-mm Flux Maps**, `PRODCATG='SCIENCE.IMAGE.FLUXMAP'`
  in order to support APEX/LABOCA products. The format is very similar to the one defined for IMAGE.

- **Source Lists**, `PRODCATG='SCIENCE.SRCTBL'`
  Single-band source catalogues directly extracted from the image to which they are associated to (via the PROV*i* keyword).

- **IFU 3-D Data Cubes**, `PRODCATG='SCIENCE.CUBE.IFS'`
  The data cube is stored in a FITS image extension, no data in the primary HDU.

- **Catalogues**, PRODCATG='SCIENCE.CATALOG'
  Uniform tabular structure including content descriptors (employing UCDs).
  Multi file format supported, especially for Large Surveys catalogues, Tile-by-Tile fashion: PRODCATG='SCIENCE.MCATALOG' and 'SCIENCE.CATALOGTILE'.
  They are also served via a dedicated query interface for catalogues, ESO Catalogue Facility[‡].

3. RELEASE MANAGER

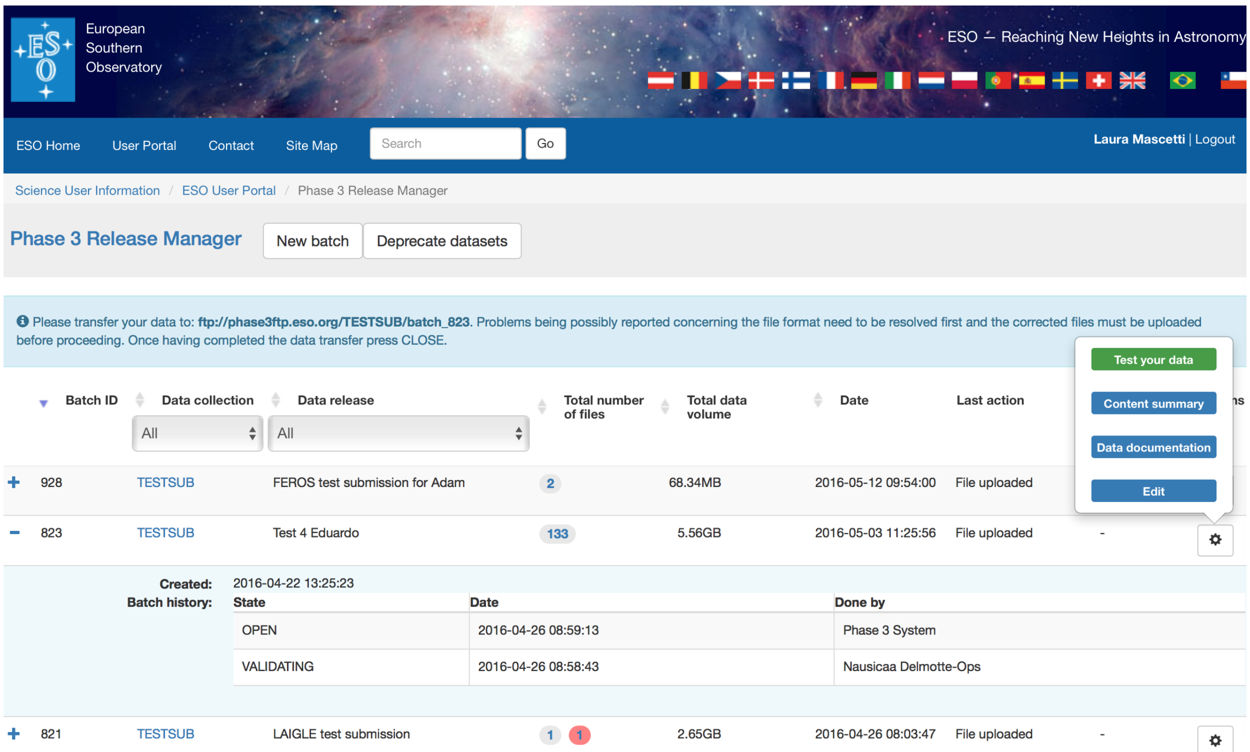

Figure 2. Screen-shot of the Release Manager.

The Release Manager (RM)[§] is a dedicated web application for controlling the entire Phase 3 data submission process by ESO operators and data providers. It allows to trigger the verification of the files, open new releases, delegate part of the Phase 3 tasks, upload the data documentation, visualise the summary of the release content and submit data. Fig. 2 shows the interface.

4. DATA VALIDATION

4.1 Ingestion Tool

The Phase 3 Ingestion Tool (IT) is the software underneath that governs the Phase 3 workflow. It is written in **Groovy**. Apache Groovy is an object-oriented programming language for the Java platform. It is a dynamic language with features similar to those of Python, Ruby, Perl and Smalltalk. It can be used as a scripting language for the Java Platform, is dynamically compiled to Java Virtual Machine (JVM) bytecode and interoperates with other Java code and libraries[¶].

[‡] http://www.eso.org/qi/
[§] http://www.eso.org/rm/
[¶] http://www.groovy-lang.org/

The IT runs as a daemon and it periodically checks for releases in OPEN, VALIDATING, CLOSED and ARCHIVING state and performs several operations, one of which is the data verification. After the data have been prepared and uploaded to the ESO server, fitsverify[‖] runs for each FITS file to check conformity with the FITS standard. The batch structure is also verified by reporting errors when `PRODCATG` keyword is missing, missing or duplicated `PROV`i are found or when the association of the science files with the ancillary files are incorrect or incomplete. The integrity of the non-FITS files is ensured by computing the MD5 hash for the file and comparing it with the value of the keyword `ASSOM`i stored in the header of the related science FITS file. See Appendix A for more details about the checks regarding the batch structure.

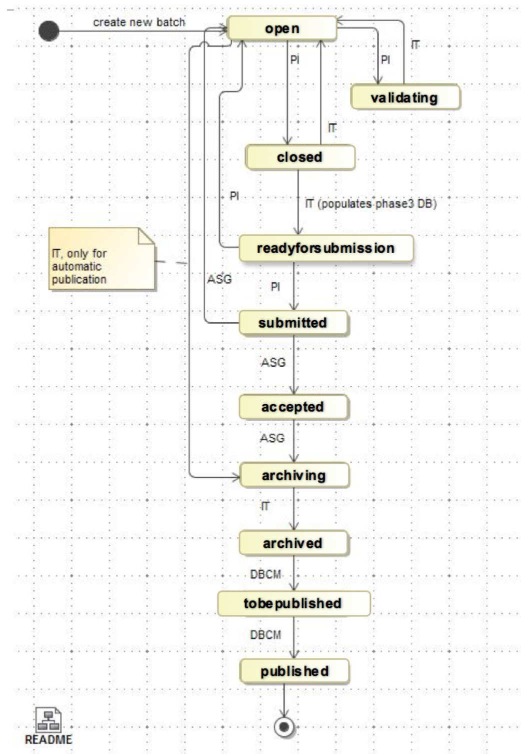

Figure 3. Ingestion Tool State Machine. It represents the status of a batch and who can trigger changes of its status. The first actions are triggered by the Principal Investigator (PI), the IT that runs as a demon reacts to the changes and ASG is in charged of reviewing the content and accept or reopen the batch, then trigger the archiving process. The Database Content Management group (DBCM) is taking care of the publication process after the archiving.

The IT then performs the *Phase 3 format and provenance verification* for each file. To do this, the validation rules (see Appendix B for more details regarding the checks involved) are dynamically added to the class path, this means that the tool does not need to be restarted if the rules are updated. Possible issues concerning the files are displayed in the RM and need to be fixed before proceeding. The data provider has then to close the submission via the RM. By closing the Phase 3 batch, the respective FTP directory turns into read-only mode and data cannot be added nor modified by the data provider. This signal triggers again the verification process that, at the end of its execution, sends a notification E-mail with possible issues and output from the automatic checks performed (all the notification are configurable and can be disabled or enabled). If problems are reported as a result of the Phase 3 format and provenance verification process, these need to be fixed by the data provider, and the data need to be replaced on the Phase 3 FTP server before proceeding. When the verification is successful, the data have still to be revised by ASG, which performs an in depth content validation, including cross-checks with the provided documentation and several other checks that cannot be automatically

[‖] https://heasarc.gsfc.nasa.gov/docs/software/ftools/fitsverify/

implemented, like spotting outliers. For a complete overview of the sequence of steps of the Phase 3 process please refer to the Phase 3 web pages, in particular to
http://www.eso.org/sci/observing/phase3/overview.html.

4.2 Rules for the verification

```
case "SCIENCE.IMAGE":
case "SCIENCE.MEFIMAGE":
    return ScienceImage.class
case "SCIENCE.SPECTRUM":
    return ScienceSpectrum.class
case "SCIENCE.SRCTBL":
    return ScienceSourceTable.class
case "SCIENCE.CUBE.IFS":
    return ScienceCubeIfs.class
case { ParserUtils.isCatalog(it) }:
    return ScienceCatalog.class
case "SCIENCE.IMAGE.FLUXMAP":
    return ScienceImageFluxmap.class
case { it.startsWith("ANCILLARY.") &&
    it.replaceFirst("ANCILLARY\\.","") in [ "PREVIEW", "HARPSTAR", "FEROSTAR", "WEIGHTMAP", "GAINMAP",
                                            "MASK", "SPECTRUM", "2DSPECTRUM", "IMAGE", "README", "MOSSPECTRA",
                                            "MOSSPECTRA.CALSIM", "MOSSPECTRA.SKY", "MOSSPECTRA.PREVIEW",
                                            "RMSMAP", "SNRMAP", "SRCTBL", "RESMAP", "SRCMASK", "FILTERED",
                                            "SPECTRUM.TELLURIC","SPECTRUM.CONTINUUM", "MUSE.PIXTABLE_REDUCED",
                                            "KMOS.PIXTABLE_REDUCED", "VARMAP" ] }:
    return Phase3FitsFile.class
default:
    throw new Exception("No validator found for category $category")
```

Figure 4. The class ValidationFactory assigns a class to each category of FITS files, as defined by the header keyword `PRODCATG`.

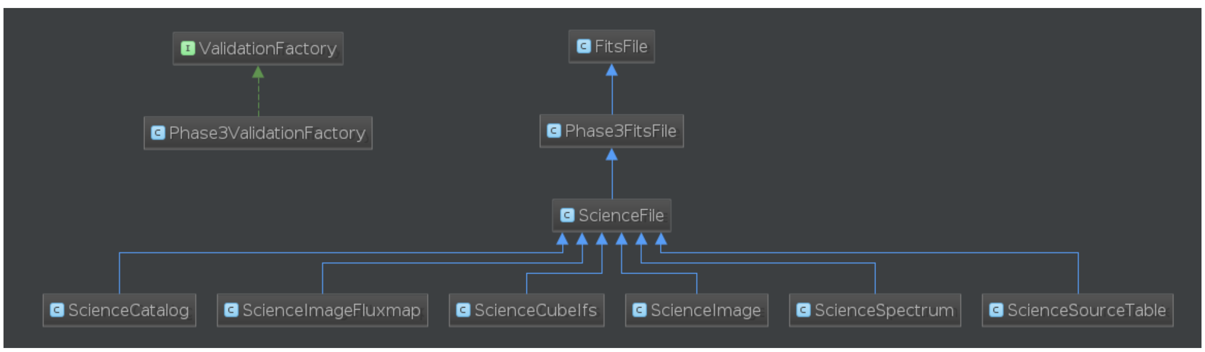

Figure 5. Class hierarchy: schematic view of the Groovy classes used for the file verification.

The class FitsFile is part of the core of the Phase 3 software. It defines an interface with the FITS files, allows for easy access to the keywords and data in the various FITS HDUs. This class makes heavy use of the dynamic features of Groovy because it creates dynamic properties based on the keywords defined in the FITS file. This allows the users to access the keywords in a more compact way (the batch content summary can be displayed via the RM). These properties are then cached to minimise the access to the disk.

The class Phase3ValidationFactory assigns to each file, reading the header keyword `PRODCATG`, the rules to be applied, in the way shown in Fig. 4.

The class hierarchy, shown in Fig. 5, defines all the tests that a certain file category must undergo: auxiliary files, for example, must only comply to the tests performed in Phase3FitsFile, while a science catalog has to comply also to what is defined in ScienceFile and ScienceCatalog. All the FITS files, besides being valid FITS files (passing fitsverify) must have `CHECKSUM` and `DATASUM` present in each HDU, as checked by the class Phase3FitsFile.

The specific checks for each category regard:

- presence of mandatory keywords,
- or keywords not allowed (like BUNIT in spectra),
- then for each mandatory keywords the function *checkMandatoryKeywords* calls an other function, *check-KeywordType*, that checks the type of the keyword value according to a specified dictionary.
- Moreover, the exact value of some keywords is checked, or whether the value is in a given range, allowing certain tolerance or whether the value is within a list of values. There are also cross-header checks that involve keywords in different HDUs.
- Additional checks on the data themselves are performed, this is for example the case of 1-D spectra where it is required that the wavelength array is strictly increasing.
- Checks between data and metadata are also performed (for example, this is the case of WAVELMIN and WAVELMAX in the spectral case, that are checked against the first and last data values in the wavelength array).
- The existence of required extensions is also checked, by extension name.

4.3 Future Improvements

It is foreseen to expand the automatic checks to include the consistency of some keyword values with the provenance and other files part of the batch (like the INSTRUME, TELESCOP and ORIGIN values, NCOMBINE and other information that have to be propagated or derived from other files), including them in a new part that can be seen as a batch verification, in addition to the already implemented file-by-file verification. This will significantly improve the efficiency of the Phase 3 workflow, by decreasing the time between submission and publication. Data provider will be notified at an earlier stage of possible issues with the data and the workload for ASG concerning all the manual tests performed will decrease. For more details on this aspect, please refer to the submitted contribution by J. Retzlaff.[4]

APPENDIX A. AUTOMATIC CHECKS IMPLEMENTED IN THE BATCH STRUCTURE VERIFICATION

Checks implemented to verify the batch structure.

```
- Run fitsverify for FITS files

- Check for the following errors in the phase3 keywords:

  1. Files without PRODCATG

  2. ASSO[CNM]i keyword in non SCIENCE file

  3. Missing ASSONi keywords (e.g. ASSOC1, ASSOC3) for SCIENCE files

  4. Presence of ASSOCi or ASSOMi in case ASSONi points to a FITS file

  5. Absence of ASSOCi and ASSOMi in case ASSONi is not a FITS file

  6. Duplicated ASSONi

  7. Missing and orphaned ASSOCi and ASSOMi

  8. Missing PROVi keywords (e.g. PROV1, PROV3)

  9. Duplicated PROVi entries

- Flag empty batches

- Flag duplicate filenames

- Flag outliers in the batch directory (files not belonging to any dataset)

- In case of updates: check for conflicts, duplicates, unresolved files

- Verify checksum of non FITS files

- Resolve provenance (check that the files mentioned are already archived or uploaded)

- Flag mixed ways of encoding provenances within the same file
```

APPENDIX B. AUTOMATIC CHECKS IMPLEMENTED IN THE FILE-BY-FILE VERIFICATION

Detailed checks implemented for each file category. The header keywords are defined in the SDPS.

```
All Phase3 files
        CHECKSUM and DATASUM in all extensions

SCIENCE.*
        LONGSTR and CONTINUE are not allowed
        Mandatory keywords: ORIGIN, PROG_ID, TELESCOP, INSTRUME, OBSTECH, DATE,
                            MJD-OBS, MJD-END, PROCSOFT
        DATE format yyyy-MM-dd
        MJD-END >= MJD-OBS
        check MJD-OBS value. It cannot be < 40587
        CDELT[123] are not allowed
        PC[12]_[12] are not allowed
        PROG_ID=MULTI if PROGID1 is defined
        if PROG_ID is MULTI PROGID1 and PROGID2 must be defined
        EXPTIME >= 0, if defined
        EXPTIME < (TEXPTIME+0.01)
        EXPTIME <= TEXPTIME, if defined
        TL_OFFAN is >= -999, if defined
        if ORIGIN not like 'ESO%' and not APEX then NOESODAT must be T
        FLUXCAL in ABSOLUTE, UNCALIBRATED
        RA/DEC in the correct range
        SKY_RES, SKY_RERR, ABMAGLIM in the correct range
        WAVELMIN/MAX consistent
        SPEC_RES not negative
        PIXNOISE not negative
        PROV0* not allowed
        OBID0* not allowed
        EXTNAME defined in all extensions, PHASE3PROVENANCE and PHASE3CATALOG must be unique.
        ORIGFILE only in primary HDU
        PRODCATG only in primary HDU

SCIENCE.IMAGE
        Mandatory keywords: BUNIT, FILTER, OBJECT, RADECSYS, FLUXCAL, PHOTSYS, REFERENC
        Mandatory keywords: RA, DEC, EQUINOX, EXPTIME, TEXPTIME, M_EPOCH
        if ORIGIN == GRANTECAN
            TELESCOP == GTC
            NOESODAT == true
            else
            mandatory keywords: NCOMBNE, OBID1
        if MEF
            NAXIS == 0
            ext:NAXIS == 2
            Mandatory keywords in extenstions: BUNIT, PHOTZP, PHOTSYS, ABMAGLIM, ABMAGSAT,
                                               PSF_FWHM, and ELLIPTIC
            Mandatory keywords in extenstions: CTYPE[12], CRVAL[12], CRPIX[12], CD[12]_[12]
        else
            NAXIS == 2
            Mandatory keywords: BUNIT, PHOTZP, PHOTSYS, ABMAGLIM, ABMAGSAT, PSF_FWHM,
```

```
                            and ELLIPTIC
            Mandatory keywords: CTYPE[12], CRVAL[12], CRPIX[12], CD[12]_[12]
        FLUXCAL in [ABSOLUTE, UNCALIBRATED]
        if VIRCAM or OMEGACAM
            Mandatory keywords: IMATYPE, TL_RA, TL_DEC, TL_OFFAN, EPS_REG, ASSON1
            if MEF
                Mandatory keyword in extensions: ELLIPTIC
            else
                Mandatory keyword: ELLIPTIC
            IMATYPE = TILE
            if not deepProduct
                Mandatory keyword: DIT
                if VIRCAM
                    Mandatory keywords: NJITTER, NOFFSETS, NUSTEP, NDIT
            if OMEGACAM
                Mandatory keywords: NCOMBINE, TL_ID
                TELESCOP = ESO-VST
                ORIGIN = ESO-PARANAL
                OBSTECH = IMAGE,DITHER
        NDIT is int, if defined

SCIENCE.SRCTBL
        Mandatory keywords: FILTER, OBJECT, RADECSYS, 1:PHOTSYS, IMATYPE, REFERENC
        Mandatory keywords: RA, DEC, EQUINOX, OBID1, M_EPOCH, ISAMP
        INSTRUME in OMEGACAM, VIRCAM, HAWKI, VIMOS
        if multiBand
            mandatory keywords: FILTER1, FILTER2, PROV2, APMATCHD
            mandatory keywords: ext:FPRA[1-4], ext:FPDE[1-4], ext:MAGLIM[1-2]
        else
            mandatory keywords: EXPTIME, TEXPTIME, ext:ABMAGLIM, ext:ABMAGSAT,
                                ext:PSF_FWHM, ext:ELLIPTIC
        IMATYPE in [PAWPRINT, TILE, VSTRIPES]
        if VIRCAM or OMEGACAM
            Mandatory keywords: TL_RA, TL_DEC, TL_OFFAN, EPS_REG
            if OMEGACAM
                Mandatory keywords: TL_ID
                TELESCOP = ESO-VST
                ORIGIN = ESO-PARANAL
                OBSTECH in [IMAGE,DITHER, IMAGE,JITTER, IMAGE,OFFSET, IMAGE,STARE]
        mandatory keywords TFORMi, TTYPEi, for every column in every table extension

SCIENCE.IMAGE.FLUXMAP
        Mandatory keywords: BUNIT, FEBE1, FILTER, OBJECT, RADECSYS, TIMESYS, FLUXCAL, ASSON1
        Mandatory keywords: REFERENC, MAPMODE, SKY_RES, BNOISE
        Mandatory keywords: CTYPE[12], CRVAL[12], CRPIX[12], CD[12]_[12]
        Mandatory keywords: NAXIS1, NAXIS2, RA, DEC, EQUINOX, FLUXERR,
                            WAVELMIN, WAVELMAX, NCOMBINE
        Mandatory double keywords:
        FLUXCAL is ABSOLUTE
        TELESCOP = APEX-12m
        ORIGIN = APEX
        if FEBE1 is LABOCA-ABBA
```

```
            FILTER is 870u
            WAVELMIN is 7.994E05
            WAVELMAX is 9.517E05

SCIENCE.SPECTRUM
        cross-header checks
        abs(0:RA-1:RA) <= 1e-7
        abs(0:DEC-1:DEC) <= 1e-7
        0:OBJECT = 1:OBJECT
        1:TELAPSE >= TEXPTIME (tol=0.1s)
        1:TELAPSE = MJD-END - MJD-OBS (tol=0.1s)
        1:TMID = (MJD-OBS + MJD-END)/2 (tol=1e-4)
        1:SPEC_VAL = (WAVELMIN+WAVELMAX)/2 (tol=1e-3)
        1:SPEC_BW = WAVELMAX-WAVELMIN (tol=1e-3)
        SPEC_BIN = (WAVELMAX - WAVELMIN)/(1:NELEM-1)) (tol=1e-3)
        primary HDU
            mandatory keywords: DISPELEM, SPECSYS, OBJECT, RADECSYS, FLUXCAL, REFERENC
            mandatory keywords: BITPIX, NAXIS, RA, DEC, EQUINOX, WAVELMIN, WAVELMAX,
                                SPEC_BIN, FLUXERR, EXPTIME, TEXPTIME, SNR, SPEC_RES
            mandatory keywords: SIMPLE, EXTEND, M_EPOCH, TOT_FLUX, CONTNORM
            if ORIGIN == GRANTECAN
                TELESCOP == GTC
                NOESODAT == true
            else
                mandatory keywords: NCOMBNE, OBID1
            BUNIT, CD1_1 are not allowed
            NAXIS = 0
            abs(EXPTIME-TEXPTIME) <= 0.01
            if FLUXCAL = ABSOLUTE
                FLUXERR = -2 or FLUXERR in [0..100]
            if not XSHOOTER
                mandatory keywords: EXT_OBJ
        extension HDU header
            mandatory keywords: VOCLASS, OBJECT, EXTNAME, TITLE
            mandatory keywords: TTYPE[1-3], TUTYP[1-3], TFORM[1-3], TUNIT[1-3], TUCD[1-3]
            mandatory double keywords: RA, DEC, APERTURE, TELAPSE, TMID, SPEC_VAL, SPEC_BW,
                                       TFIELDS, NELEM, TDMIN1, TDMAX1
            BUNIT is not allowed
            NAXIS = 2
            NAXIS2 = 1
            TUTYP1 matches (Spectrum.|spec:|eso:)Data.SpectralAxis.Value
            TUTYP2 matches (Spectrum.|spec:|eso:)Data.FluxAxis.Value
            TUTYP3 matches (Spectrum.|spec:|eso:)Data.FluxAxis.Accuracy.StatError
            TTYPE1 in WAVE, FREQ, ENER
            TTYPE2 begins with FLUX
            TTYPE3 begins with ERR
            TFORMi must all be the same
            VOPUB = ESO/SAF
            GCOUNT = 1
            PCOUNT = 0
        extension HDU data
            the data in the first column is strictly increasing
```

```
            1:NELEM = #elements in the first column
            if TTYPE1 == WAVE
                WAVELMIN = data[0][0] (tol = 0.0001), handles only nm, um and ang
                WAVELMAX = data[0][-1] (tol = 0.0001), handles only nm, um and ang
                TDMIN1 = data[0][0] (tol = 0.0001)
                TDMAX1 = data[0][-1] (tol = 0.0001)

SCIENCE.CUBE.IFS
        cross-header checks
    primary HDU
        Mandatory keywords: BUNIT, OBJECT, RADECSYS, FLUXCAL, REFERENC, DATE-OBS,
                            PROV1, OBSTECH, ASSON1
        Mandatory keywords: RA, DEC, EQUINOX, EXPTIME, TEXPTIME
        Mandatory keywords: NCOMBINE, OBID1, WAVELMIN, WAVELMAX, SPEC_RES, SKY_RES
        FILTER is not allowed
        specific checks for MUSE
            check WAVELMIN/MAX range
            check SPEC_RES range
    extension HDU header
        mandatory keywords: NAXIS3,
        EXTNAME='DATA' must be present in one extension (the other extensions are optional)
        if there is an extension named STAT
            STAT:SCIDATA must be DATA
            DATA:SCIDATA must be STAT
        NAXIS=3 in all extensions

SCIENCE.CATALOG, SCIENCE.MCATALOG, SCIENCE.CATALOGTILE
        primary HDU
        FILTER=MULTI if FILTER1 is defined only for OBSTECH=*IMAGING*
            if FILTER is MULTI FILTER1 and FILTER2 must be defined
            PRODCATG is catalog main file
                Mandatory keyword: REFERENC
            PRODCATG is catalog tile
                Mandatory keywords: RA, DEC, OBJECT, FPRA1, FPDE1, SKYSQDEG
            NAXIS = 0
            if PRODCATG is SCIENCE.CATALOG
                if APEXBOL
                    ORIGIN = APEX
                    TELESCOP = APEX-12m
                    Mandatory keywords: FILTER, TIMESYS, WAVELMIN, WAVELMAX, SKY_RES, BNOISE
        extenion HDU header
            XTENSION = BINTABLE
            EXTNAME = PHASE3CATALOG
            Mandatory keyword: TFIELDS
            TDMINi < TDMAXi, if both defined
            TCOMMi must be defined and cannot be empty
            TFORMi must be defined and must follow the standard
            TSCALi is not allowed
            TTYPEi must be defined, cannot be an SQL reserved keywords
                   and must match the pattern [A-Za-z][A-Za-z_0-9]*
            TUCDi must be defined, composed of the UCD valid atoms
                   and must define one and only one identifier (meta.id;meta.main)
```

```
    TUNITi must be defined
    TZEROi is not allowed
    TXLNKi must be ARCFILE, ORIGFILE or CATALOG
    TTYPEi must be unique, case insensitive
extension HDU data
        ID column values must be unique and not null
        numeric values must be within TDMIN and TDMAX, if they are defined
```

ACKNOWLEDGMENTS

We would like to thank Ignacio Vera, Database Architect at ESO for his efforts and support. A special thanks to Cristiano Da Rocha from Terma GmbH for his support.

REFERENCES

[1] Arnaboldi, M., "Public Surveys at ESO" Proc. SPIE 9910-11 (2016), this volume.

[2] Delmotte, N., "Validation of ESO Phase 3 data submissions" Proc. SPIE 9910-7 (2016), this volume.

[3] Micol, A., "ESO Science Data Product Standard for 1D Spectral Products" Proc. SPIE 9910-121 (2016), this volume.

[4] Retzlaff, J., "Publication of science data products through the ESO Archive: lessons learned and future evolution" Proc. SPIE 9910-8 (2016), this volume.