This project is supported by the European Commission's FP7 Capacities Programme for the period April 2013 - March 2017 under the Grant Agreement number 312495.

www.solarnet-east.eu

DELIVERABLE D20.4

Document on Standards for Data Archiving and VO

WP20 Integrated Operation and Exploitation of Solar Physics Facilities and Coordination with other Research Infrastructures

1ST Reporting Period

November 2014

This project is supported by the European Commission's FP7 Capacities Programme for the period April 2013 - March 2017 under the Grant Agreement number 312495.

PROJECT GENERAL INFORMATION

Grant Agreement number: 312495

Project acronym: SOLARNET

Project title: High-Resolution Solar Physics Network

Funded under: FP7-INFRASTRUCTURES: INFRA-2012-1.1.26 - Research Infrastructures for High-Resolution Solar Physics

Funding scheme: Combination of Collaborative Project and Coordination and Support Action for Integrating Activities

From: 2013-04-01 to 2017-03-31

Date of latest version of Annex I against which the assessment will be made: **13/02/2013**

Periodic report: 1st **X** 2nd ☐ 3rd ☐ 4th ☐

Period covered: from **01/04/2013** to **30/09/2014**

Project's coordinator: Dr. Manuel Collados Vera, IAC.

Tel: (34) 922 60 52 00

Fax: (34) 922 60 52 10

E-mail: mcv@iac.es

Project website address: http://solarnet-east.eu/

Document on Standards for Data Archiving and VO

SOLARNET WP 20.3 deliverable D20.4

Preface

This work package is meant to establish standards for an adequate dissemination of data to the community. This includes considerations from three main viewpoints:

1) User interaction with future Solar Virtual Observatories (SVOs):
   a) The standards should provide for flexible and efficient searches.
   b) The standards should include clear descriptions of data quality criteria that may be used to select data.
2) Usability of the data files:
   a) Data files should be easy to use and as self-sufficient and self-explanatory as can reasonably be achieved, without reference to additional information.
   b) It should be possible to develop generic tools for data visualisation and analysis.
   c) It should be possible to continue using existing software and services that rely on data files in a format not consistent with this standard.
   d) Data files should contain quality information.
3) Pipeline design:
   a) The standards should be specific enough to avoid differences in interpretations between different pipelines.
   b) The standards should be forward-looking, giving mechanisms and principles that are flexible yet strict enough to prevent divergence among future pipelines handling new instruments and data types that do not yet exist.
   c) It must be possible to implement the standards – i.e. the standards should not require mandatory information that may not be available.
   d) The standards should take repeatability into account, by requiring that the applied processing is described in a proper manner.

Section 1 is written from the standpoint of user interaction with an SVO, asking and answering the question “what should an ideal SVO be able to do”. It does not go into great amounts of detail, and the contents is not to be viewed as a “standard” per se: The most important point can best be summarised in the idea of an ideal SVO that may find any desired combination of multi-instrument observations and events/features (if they exist). This is a quite daunting task, far beyond the capabilities of any existing SVO, and a full implementation might easily take several years. We have chosen not to give any *formal* description of a data model, since this is implementation dependent, but rather say in general that all metadata given as FITS keywords (see the FITS Standard v. 3.0), should be available for searching - including those that are instrument-specific. Also, questions such as most of those relating to the user interface of an SVO are left open, as it is hard to say what the best solutions will be without the ability to

experiment with an SVO with the mentioned capabilities. In fact, designing a good user interface for such an SVO may be the hardest part of the implementation.

Nevertheless, Section 1 is very important in that it sets the stage for the rest of the document, which goes in *much* more detail as an explicit standard/convention[1] for how data files are to be composed in order to fulfil the overall objectives of the work package, including those in Section 1.

Although the rest of the document is written as a standard/convention, it is not in any way a final one. First of all, it has *never* been applied to any data set or pipeline yet - which will undoubtedly bring to light shortcomings that must be amended. Second, the world of existing FITS file conventions is incredibly large, and it may very well be that we have overlooked some that overlap with what we have written. Whenever such overlaps are discovered, a review should be done to see if the existing convention should be adopted instead. This will be determined by comparing the two conventions' flexibility, expandability, well-definedness, ease of use, and the extent of adaptation of the existing convention. Thus it is to be expected that Part 2 will be revised at some point.

The primary audiences for this document are designers of pipelines that produce data for distribution through a Solar Virtual Observatory (SVO), designers of utilities (web services and software) that further process and visualise such data, and designers of SVOs.

[1] In the FITS community, the word "standard" means an officially adopted FITS standard, normally published in a peer-reviewed journal. The word "convention" is normally used for more unofficial "standards". De facto, what is described in this document are FITS *conventions*.

Table of Contents

1. An ideal Solar Virtual Observatory (SVO)

Traditionally, solar observation archives and SVOs have been used primarily to locate data from data sets that researchers have already known existed. Most likely, the majority of searches have been to find and retrieve specific observations that the researcher already knew about – at least when excluding searches for any "supporting observations" from sources already known by the researchers.

However, the number of data sets available has grown, and will continue to grow as more and more ground-based observations are made available. As a result, researchers may not know about all the potentially useful observations that are available. The use of multi-instrument analysis of solar phenomena has grown over the last decade or two (as witnessed by e.g. the many Joint Observing Programs and campaigns run by SOHO instruments, often in collaborations with ground based observatories) – but the ability of SVOs to locate multi-instrument observations not already known to the researcher has not grown with it.

An ideal SVO should be able to locate "the ideal set of observations" for an observer - i.e. to find sets of successful observations matching a hypothetical ideal observation proposal (if such observations exist). This includes joint observations of specific targets/events from multiple instruments - even from multiple positions within the solar system. Such a scenario may even involve observations that do not overlap in time, e.g. solar disk observations of events vs. *in situ* observations of particles/shocks/interactions at a later time.

The searches might even include "success criteria", describing the quality of the resulting data such as effective seeing, cadence variations, the number of dropped frames etc.

For such search capabilities to be meaningful, the searches should – as far as possible – be "instrument agnostic" - i.e. formulated to find observations based on their physical characteristics, not their origin. This means emphasis must be put on making sure data pipelines output all the necessary characteristics, in a standardised manner.

If no matching data exists, the user might be given feedback on how to relax the search criteria to find matches that are close to the ideal scenario. In case parts of a search has been specified with reference to specific instruments, the SVO might also be able to suggest data sets from other instruments with similar (or better) characteristics.

1.1. A search example

Although the above paragraphs describe an ideal SVO, an exhaustive, detailed description of such an SVO is beyond the scope of this work package. The number of different search criteria that may occur in an ideal SVO search is in principle "unbounded", in the sense that each instrument may have different relevant settings and properties (gain, focus setting, readout mode, etc). Whenever observations from a new instrument is added to the SVO, there may be entirely new search criteria that become relevant. Nevertheless, below we give a moderately complex example of a set of search criteria in a semi-structured format, to give a flavour of the searches that should be possible. It is divided into two sections - one for observations, targets, and events, and one for the required ***relationships*** between the matching observations/targets/events:

Observations:

Subsearch 1, Name: FILTERGRAMS

Type: Filtergram
Wavelength: 304Å
Filter width: 3Å
FOV size: > 100”x100”
Minimum spatial pixel resolution: (1”,1”)
Minimum effective (seeing+diffraction+focus-restoration) seeing conditions: 2” FWHM
Signal-to-noise ratio (average): >5
Cadence: <5s
Duration: >1h

Subsearch 2, Name: CDS

Type: Spectral raster
Instrument: SOHO/CDS
SOHO/CDS observation program: O_LOOP3

Targets:

Subsearch 5, Name: ACTIVE_REGION

Type: Active Region
McIntosh Z: >C
McIntosh p: a,h,k

Events:

Subsearch 3, Name: FLARE

Type: Flare (event)
Class: >M5
Max distance from disk centre: 500”

Subsearch 4, Name: SHOCK:

Type: SOHO/CECLIAS/MTOF Solar Wind Shock
Confidence level: >90%
Parameters:
Propagation model: Standard
FOV diameter (on solar surface): 200 arcsec
Time interval: (-10h, +5h)

The name given to each sub-search is freely chosen, for use in the specification of relationships between search results of different types. The searches for observations, targets, and events should be mostly self-explanatory, except for the SHOCK specification:

The type indicates that it must be a shock detected in data from the SOHO/CELIAS/MTOF instrument, with a confidence level of more than 90% (see http://umtof.umd.edu/pm/).

The interpretation of the propagation model, field of view diameter, and duration parameters is this: An in situ shock detection is a space-time event. Using the chosen propagation model, any matching events will be back-projected and transformed into another space-time event on the solar surface at a specific time. A circular “field of view” (diameter 200 arcsec) and a time interval are added, transforming the event into a kind of “remote sensing observation”. The time interval is specified to be from 10 hours prior to the back-projected event time until 5 hours after

the back-projected event time. Now, the “shock remote sensing observation” can be related to the other remote sensing observations in terms of field of view/time overlaps.

Relationships:

1. **OBSERVATIONS CDS**, **FILTERGRAMS** and **SHOCK**
   JOINT FOV OVERLAP: > 50” x 50” (intersection of all FOVs)
   FOV TIME OVERLAP: >1h (intersection of all time periods)
2. **FLARE** and **CDS**:
   FOV OVERLAP CDS ➔ FLARE: 100%
   TIME OVERLAP: >100min
3. **CDS** and **ACTIVE_REGION**:
   FOV OVERLAP CDS ➔ ACTIVE_REGION: 50%
4. **FILTERGRAMS** and **FLARE**:
   TIME OVERLAP FILTERGRAMS ➔ FLARE: > (-30min, +30min)
5. **FLARE** and **SHOCK**:
   JOINT FOV OVERLAP: >0
   TIME OVERLAP: >0

Relationship specifications are used to eliminate results that do not have e.g. the required relations in time and space to other results. Since these specifications are more complicated, a brief explanation of each one is given below:

Number 1 requires that there must be an area of size 50”x50” that is covered by all three observation types (CDS, FILTERGRAM and the back-projection of the SHOCK). Also, all three instruments must simultaneously cover this field of view for 1 hour or more.

Number 2 requires that the CDS observations overlap the entire flare area, and that this overlap lasts for longer than 100 minutes.

Number 3 requires that the CDS observations overlap 50% of the active region area.

Number 4 requires that the flare and (back-propagated) shock “fields of view” (or rather “areas”) overlap (though no minimum area specified), and that they overlap in time (no minimum duration).

As mentioned before, the construction of an SVO with all of the abilities above is a tremendous task, and would easily take several years.

1.2. Presentation of search results

An important facet of how users interact with an SVO is the ability to group the search results. E.g. if a search matches 1000 observations, but they are part of only 5 different observation series (e.g. different filters or wavelength regions), it makes sense to have an optional grouping mechanism to “collapse” the result listing into only 5 lines, showing some form of “summary” of the underlying files for each series - if the user so chooses.

Different grouping methods or levels will be useful for different purposes and at different stages in the search for data. Thus an SVO should offer a variety of grouping methods. This is further discussed in Section 8.

1.3. Visualisation

Given a successful search, a quick-look capability should exist, to visualize each data set matching the search criteria - through images, or perhaps showing movies of selected results side by side, etc. This could be achieved either internally or by using external web services, some of which already exist.

1.4. Types of Observations, Targets, and Events

In order to enable searches like those described above, the different types of observations, targets, and events must be identified, and their relevant attributes must be defined. We note, however, that the issue of targets and events is outside the scope of this work package.

With regard to observations, attributes that are common to more than one type should be identified, creating a classification scheme. It might be, for example, that spectrometer rasters with dimensions `(x,y,lambda)` should also qualify as an image according to the search criteria above, and should be returned as a match if it covers the 304Å line, since it does contain an "image" in 304Å. For now, at least, it seems wise to avoid performing any such classification inside data processing pipelines - they should instead focus on an accurate physical description of the data, leaving the classification to an SVO.

1.5. Instrument-specific parameters

In principle, generic observation attributes should suffice to describe the data, but there will always be other parameters that may be relevant, even though only those who know an instrument fairly well will want to use them.

In many cases, a researcher will be much more familiar with one (or more) instruments than others, and might use instrument-specific search criteria - e.g. "Study A version 2" - simply because they have experience with such data. In that case, it would be nice if the archive could extract the generic parameters matching those search criteria, and suggest alternative search criteria based on this.

2. Defining the output of pipelines

In order to fulfil the "vision" of an ideal SVO presented above, it is necessary to ensure that the data to be served contains the necessary metadata. This document describes a standard, intended to be applicable to all solar data pipeline outputs. However, since the rest of the document applies to FITS files (see the FITS Standard v. 3.0), we will below refer to the rules and guidelines as *"conventions"*, not "standards". In the FITS community, the word "standard" is reserved for officially accepted standards that in most cases have been published in peer-reviewed journals.

The scope is in principle limited to ground-based, remote sensing observations. We have, however, tried to include some very minor aspects to ensure that also space-based remote sensing data products are covered by this description.

3. Previous work and existing utilities

There have been/are numerous projects with SVO-like characteristics and goals (e.g. VSO, HELIO, Helioviewer, and the Hinode archive in Oslo). We have tried to take advantage of the lessons learned by these. In addition, we also try to maximise the ability of a future SVO to take advantage of existing utilities - such as services provided through HELIO and more or less generic analysis and visualisation software.

However, inherent differences and inconsistencies between existing pipeline outputs, services, and SVOs make this very hard or even impossible through a simple merging of the conventions in use.

Instead, we concentrate on using the most generally accepted FITS standards (e.g. the World Coordinate System) to describe the physical aspects of the data as accurately and exhaustively as possible.

In principle, an exhaustive description of the physical characteristics of the observations should suffice to make new SOLARNET FITS files backwards compatible with existing analysis or visualisation utilities, *given* some relatively simple utility-specific "translation" routines. These are often needed in order to convert values or rename keywords in order to match the conventions and practices in use by the legacy data set for which the utilities were written.

On the other hand, there must also be translation routines that make legacy data compatible with SVOs/utilities that depend on SOLARNET conventions - i.e. the conventions presented in this document. This would be done *without* modifying the original files, but rather by "synthesising" a set of metadata compatible with the SOLARNET conventions, in order to ingest the data in an SVO. Utilities that rely on SOLARNET-type headers would then be able to retrieve a synthesised header from the SVO (either when downloading the data or later, during analysis and visualisation).

Note that this means that whenever this document says "must", then strictly speaking it only applies to fully compliant files.

Quite a few existing utilities *also* rely on additional data/metadata not described in the SOLARNET conventions. Since legacy data and legacy utilities *must* be covered by the SOLARNET project, the conventions *must* allow for such data/metadata to be included. This is also important for new data sets where additional data is needed in order to e.g. interpret or calibrate/recalibrate the data.

However, such additional data would in general not be possible to "synthesise" for observations other than those for which the utilities were originally written, since there is no way to "guess" what those data should be.

Note that in the above discussion, "legacy" data sets include those that are currently being produced by existing pipelines. In general, we do not expect such pipelines to be modified in order to produce SOLARNET-compliant files, due to a lack of resources and the potential need for reformatting of very large data sets

For new pipelines, it is desirable that there is a convergence towards:

1. Including as much as possible of the metadata described by SOLARNET conventions
2. Excluding superfluous and sometimes ill-defined keywords
3. Using only the SOLARNET definitions of all SOLARNET keywords

4. File format, file names, and HDUs

Based on common practice in the solar remote sensing community, we highly recommend using the FITS file format for disseminating solar observations. This document describes how to include the metadata content through keywords inside FITS files. That does not preclude the use of other file formats, *if* there are strong arguments for doing so. As long as the requirements for the metadata information content are met, an automated translation between the representations in different formats is achievable. However, until such a translation utility exists, it may be impossible to submit and ingest such data into an SVO.

4.1. File naming conventions

Although file naming conventions are of little consequence to an SVO, they can be a big help for users to get a manual overview of files stored locally, so we give some recommendations/rules below:

We strongly recommend using the widespread convention of keeping file names in all lowercase. File names must only contain letters A-Z (deprecated) and a-z, digits 0-9, periods, underscores and minus signs. Each component of the file name should be separated with an underscore – not a minus sign. File name components with numerical values must be a) preceded with one or more identifying letters, and b) given in a fixed-decimal format, e.g. (00.0300). Variable-length string values should be post-fixed with underscores to a fixed length.

Another common convention has been to start the file name with the "instrument name" - typically defined in a consistent manner only on a *per mission* or *observatory* basis - i.e. collisions may appear with other missions. Thus we recommend prefixing the instrument name with a mission or observatory identifier (e.g. **`iris`** for IRIS or **`sst`** for SST). After the instrument name, the data level is normally encoded as e.g. "l0" and "l1" for level 0 and 1. Note, however, that the concept of data level is normally *entirely* project/instrument-specific and does not by itself uniquely identify what kinds of processing have been applied.

Within each data set it is often very useful to have file names that can be sorted by time when subject to a lexical sort (such as with "**`ls`**"). That means the next item in the file name should be the date and time (**`YYYYMMDD_HHMMSS[_ddd]`**). The "**`ddd`**" part is milliseconds, and is optional. Even more decimals can be added, if necessary.

If the data may be made available (simultaneously) with e.g. different processing emphasis (e.g. trade-offs between resolution and noise level), an alphanumeric identifier[2] of the processing mode must be added in order to ensure uniqueness of the file name.

What comes next is highly instrument-specific, but attributes that specify the type of content should definitely be encoded, e.g. which filter has been used, which type of optical set-up has been used, etc.

File names *must* be unique - i.e. all files must be able to coexist in a single directory. If the above conventions/suggestions do not result in a unique name, some additional information *must* be added.

4.2. FITS File Header/Data Units (HDUs)

FITS files may contain one or more data arrays, each with its separate header containing metadata in keyword-value pairs. In this document, we use the notation **`ABCXYZ=5`** to say that "the keyword **`ABCXYZ`** is/must be equal to **`5`**". Similarly, **`SOLARNET>0`** means that "the keyword **`SOLARNET`** is/must be greater than zero". When keyword names are given using notations such as "**`KYnnna`**", or "**`KEYn_i`**", **`nnn`** stands for a *non-zero-padded* numerical value between **`1`** and **`999`**, both **`i`** and **`n`** stand for a number between **`1`** and **`9`**, and **`a`** stands for an *optional* letter **`A-Z`**.

Each header + data array taken together are referred to as a Header/Data Unit (HDU), as in formal FITS standards documents.

[2] E.g. a short form of the contents of **`PR_MODE`**, see Section 9.

All HDUs must be a *valid HDU* according to the most recent FITS Standard at the time of production (currently version 3.0). Otherwise, the file as a whole may be unreadable by standard FITS utilities and libraries.

All HDUs except auxiliary HDUs *must* contain **EXTNAME=<string>**. For simplicity, we will use notation like "in the **XYZ** HDU", "in HDU **XYZ**", or simply "in **XYZ**" to say "in the extension that has **EXTNAME=XYZ**".

In this document, HDUs fall into one of the categories below (as a quick reference, typical **EXTNAME** values are given in parentheses for each HDU category listed below):

- Observational HDUs, containing observational data (**He_I**)
- Tabulated-keyword HDUs[3] (**He_I:He_II;XPOSURE)**
- Index-value HDUs[4] (**He_I;SAT_IDX)**
- Auxiliary HDUs

In this example, the file contains two observational HDUs, **He_I** and **He_II**. The data have been taken using automated exposure control, and the exposure times (the **XPOSURE** values) as a function of time are tabulated in **He_I:He_II;XPOSURE** for both observational HDUs. The list of saturated pixels in the **He_I** HDU is given in the **He_I;SAT_IDX** HDU.

The characters comma, colon and semicolon have special functions in **EXTNAMEs** and lists of **EXTNAMEs**. Therefore, **EXTNAMEs** must not contain the characters comma, colon or semicolon except as prescribed in Appendix I: Tabulated-keyword convention, Appendix II: Meta-observation convention, and Appendix III: Index-value HDU convention.

In this document, will use the abbreviations Obs-HDU for observational HDUs and Tab-HDU for tabulated-keyword HDUs.

All types of HDUs may also appear in a meta form with "**;METAHDU**" appended to the **EXTNAME** (see Appendix II: Meta-observation convention).

4.2.1. Auxiliary HDUs

Auxiliary HDUs are meant to contain "additional data" that is required to describe the observations, to allow instrument-specific utilities to function correctly, to interpret the data correctly, or to enable further calibrations to be made.

Auxiliary HDUs that contain **SOLARNET=-1** may use the mechanisms described in Appendix I: Tabulated-keyword convention, Appendix II: Meta-observation convention and Appendix III: Index-value HDU convention, but then the rules relevant to these mechanisms will apply.

In all other respects, Auxiliary HDUs are not subject to any other SOLARNET conventions.

4.2.2. SOLARNET-compliant HDUs

All HDUs that contain all mandatory SOLARNET/FITS standard keywords, and no keywords with definitions that are in conflict with SOLARNET, are fully compliant with the SOLARNET conventions. They should have **SOLARNET=1.0**.

[3] See Appendix I: Tabulated-keyword convention
[4] See Appendix III: Index-value HDU convention

HDUs that do not contain all mandatory SOLARNET keywords, but have not used any SOLARNET/FITS standard keywords with a conflicting description, should have **SOLARNET=0.5**.

The keywords described below (**OBS_HDUS, AUX_HDUS, S**, and **IDX_HDUS)** are meant to be human-readable lists of *all* HDUs in the file, separated into one list for each of the possible categories.

All SOLARNET-compliant HDUs must contain the keyword **OBS_HDUS,** containing a comma-separated listing of the **EXTNAMEs** of *all* Obs-HDUs, in a format like "**He_I=0,He_II=1**", where the string before the equal sign is the HDU's **EXTNAME** value. The number after the equal sign represents the *HDU number*, starting with zero for the first HDU in the file[5].

If the file contains *one or more* Index-value HDUs, **IDX_HDUS** *must also* be present, listing all Index-value HDUs in a similar format to **OBS_HDUS**, e.g. "**He_I;SAT_IDX=4**". See Appendix II: Meta-observation convention for more details. Tab-HDUs must similarly be listed in **TAB_HDUS**. Auxiliary HDUs should be listed (for convenience) in **AUX_HDUS.**

4.3. Storing data in a single file or in separate files

From an SVO point of view, *each Obs-HDU represents a single "observation unit"*, and will be registered separately. Thus the choice of putting such observation units into separate files or not does not matter to an SVO in terms of searchability, but it does matter in terms of the file sizes of data that may be served. However, that issue will be discussed and dealt with later on, in the section "Meta-observations".

However, some "typical use" aspects should still be considered - including how most existing utilities[6] interact with observations of a particular type:

As a general rule, Obs-HDUs that would typically be analysed/used together and are seldom used as stand-alone products should be stored in the same file, whereas Obs-HDUs that are often analysed/used as stand-alone products should be stored in separate files. Furthermore, Obs-HDUs with data of fundamentally different types (e.g. filter images vs. spectra vs. Fabry-Pérot data vs. spectropolarimetry) should *not* be put in the same file.

Obviously, data processed by separate pipelines cannot be stored in a single file (unless they are combined at a later stage).

Examples and arguments in favour of a single file:

- Data from different Stokes parameters (in the same wavelength) are normally analysed together, and should be put together in a single file.

- Data from a spectrometer raster[7] are normally stored in a single file, even though the data may contain information from multiple detector readout windows. They have

[5] The HDU numbers are included because some FITS file reading utilities are not able to find an HDU based on the **EXTNAME** value.
[6] Some utilities may prefer different grouping of HDUs with respect to separate vs. single files, but that issue may be solved by a utility program that is able to join HDUs in separate files into a single file and vice versa.
[7] Rasters are observations (usually spectrometric) collected by stepping a slit across the observation area.

normally been acquired in a synchronous fashion, and they may be analysed together in order to have better estimates of continuum values when performing line fitting.

- Having too many files can lead to inefficiency both on a file system basis and on the level of utilities.

- Not least, having too many files will also be an inconvenience to users who want to look at file lists manually.

Examples and arguments in favour of separate files:

- Observations with different **POINT_ID** values (see section "Grouping") should not be stored in the same file.

- When solar rotation compensation or feature tracking creates jumps that would make it necessary to use a tabulated form of WCS in order to reach the *desired spatial coordinate accuracy* (cf. **CRDERi** in the section on WCS), the data before and after the jump should be stored in separate files. This is somewhat stricter than the **POINT_ID** rule, i.e. data spread over multiple files due to this "tabulated WCS rule" may have the same **POINT_ID**.

- Images/movies in different filters are often used as stand-alone products, even if a parallel observation in another filter exists. Thus observations from different filters should be put in separate files.

- In a similar fashion, Fabry-Pérot scans of *separate wavelength regions* should go in separate files. The same applies to similar observations such as from spectropolarimetry.

- Some observation series with very low cadence should be stored with each image in a separate file. The definition of "very low cadence", however, is somewhat dependent on the type of data, the resolution, and the variability time scale of resolved features. The "normal use" of the data also matters: If images are largely used as stationary context for other observations, they should definitely go into separate files. This is typically the case for synoptic observation series, which are also normally of indefinite length and therefore must be split up one way or another anyway.

- Observations with significantly different starting and/or end times should *not* be stored in a single file. "Significantly different" in this context means on the order of a few times the cadence/exposure time or larger - since there may be technical reasons for differences smaller than this.

An additional aspect is that grouping data into a single file makes it impossible to download "only the interesting part" of a data set for a given analysis purpose. However, given the guidelines above, we think this is unlikely to be an issue. Also, a future SVO could provide file splitting "services" to deal with this issue.

When an Obs-HDU (partially) overlaps in time and space with one or more HDUs stored in other files, **CCURRENT** (concurrent) should be set to a comma-separated list of its own file name plus the names of all files containing concurrent Obs-HDUs[8].

[8] This is of course on a "best effort" basis for the pipelines!

CCURRENT serves as a pointer to other concurrent (and probably relevant) observations, but it also serves a purpose in grouping search results (see section “Grouping”).

4.4. Obs-HDU content guidelines

In addition to guidelines determining how data should be stored in single vs. multiple files, we here give guidelines for what should be considered as single vs. multiple observational units - i.e. *what should be stored in a single vs. multiple Obs-HDUs within each file*.

Such guidelines can only be given heuristically, due to the large diversity of possible data sets.

These guidelines will of course have implications for the sizes of Obs-HDUs, thus also for file sizes. As mentioned earlier, this issue will be discussed and dealt with in the section “Meta-observations” below.

Each Obs-HDU will be registered as an individual observation unit, with attributes such as duration, minimum/maximum wavelength etc. Such attributes may be important search terms, and this must be taken into account when considering what should be collected into a single Obs-HDU or not.

As with the guidelines for keeping data in single or multiple files, some “typical use” aspects must also be considered - including how most existing utilities[9]interact with observations of a particular type. In addition, any “user convenience” issues should be taken into account.

Some examples and arguments in favour of a single HDU:

- If a Fabry-Pérot scan is stored with each exposure (each wavelength) as a separate HDU, a search made for data covering a particular wavelength inside the scan’s min/max range will not locate any files unless there is an exact match between the requested and the reported wavelength in one of the HDUs.

- If an observation is repeated with a more or less fixed cadence (except for small cadence variations caused by e.g. technical issues/limitations), this will not be immediately apparent if each repetition is stored as a separate HDU.

- Observations are often visualised by displaying slices of a multi-dimensional data array, sometimes also scanning through one of the dimension in order to visualise it as a movie (though not necessarily in the time dimension). Data that are likely to be visualised in such ways should be put into a single HDU. In other words, all dimensions through which slicing or scanning may be desirable should be included in a single HDU[10].

- Pointing adjustments *in order to track* solar features by means of solar rotation compensation or feature tracking should not cause the data to be stored in separate HDUs. This is somewhat dependent upon the frequency and magnitude of the tracking movements relative to the cadence and the field of view, but we leave it up to the

[9] Some utilities may prefer different practices, but a relatively simple program that is able to split or join HDUs in specific dimensions would solve the problem.

[10] In particular, the time dimension should be included when observations are repeated and are suitable to be presented as a movie. Repeated rasters have traditionally not been collected into single HDUs, but that may be because of their low cadence - causing relatively few files to be created during a single observation run. This is changing, however, and we recommend that repeated rasters should be joined into single HDUs which include the time dimension, e.g. `(x,y,lambda,t)`.

discretion of pipeline designers to determine when it is appropriate to split image sequences in such cases: if the data are suitable for making a single movie, store them in a single HDU.

- Pointing changes in e.g. slit-jaw movies due to slit movements should not cause the movie to be split into separate HDUs, even if there are relatively large pointing changes associated with the starting of new rasters in a series.

- Uneven spatial sampling, e.g. dense in the centre and sparse in the periphery, should not cause the data to be separated into multiple HDUs, though note that a tabulated form of WCS coordinates must be used.

- Variable exposure times due to *Automatic Exposure Control (AEC)* should not cause the exposures to be stored in separate HDUs - as long as the settings for the AEC is constant.

Examples and arguments in favour of separate HDUs:

- Observation units stored in separate files *according to the guidelines in the previous section* must be stored as separate HDUs.

- If the readout of a spectrometer has gaps (i.e. only small portions of the spectrum are extracted, in "wavelength windows"), the different wavelength windows should *not* be stored in a single HDU, since that would falsely indicate that the observation unit covers the entire spectrum between the minimum and maximum wavelengths.

- Some observation series are made with alternating long and short exposure times. These should *not* be collected in a single Obs-HDU, because of the resulting difficulty in describing the exposure time[11] as well as the complexity that would be required in utilities in order to handle/display such data correctly. Instead, the data should be separated into one HDU with long exposures and one HDU with short exposures.

- Data that are often displayed side by side, such as images in different filters, or different Stokes parameters, should be split into separate HDUs.

For very closely connected, parallel observations, it is preferable to handle the grouping of data into Obs-HDUs in the same way for all of the observations, even if they are not of the same type (e.g. repeated rasters and corresponding slit-jaw movies).

Bearing all of the above in mind, observations that fit the following description should be collected into a single HDU:

An array of data that has (quasi-)uniform spacing in each physical dimension, e.g. x, y, lambda, and time, and also has (quasi-)constant attributes such as pointing, exposure times, gain, filter, and other relevant settings.

[11] The tabulated-keyword convention *could* be used for e.g. `XPOSURE` if such data are stored in a single HDU, e.g. `(x,y,t,2)`, with `XPOSURE` stored as `(1,1,1,2)`. But this is much less self-explanatory than having two separate HDUs, and it would require users to know about the convention, and implementation of the convention inside utilities. Since it is not absolutely necessary to do it this way, the tabulated-keyword convention should not be used.

5. Cadence

The guidelines for collecting data in a single (meta-)Obs-HDU gives the possibility of reporting attributes such as cadence, which may be a very important search term for certain uses.

The planned/commanded cadence should be reported in **CADENCE**. The average (actual) cadence should be reported in **CADAVG**.

The cadence *regularity* is also important: The keywords **CADMAX** and **CADMIN** should be set to the maximum and minimum frame-to-frame spacing, respectively. **CADMAXPC** should be set to **100*CADMAX/CADENCE**, and ditto for **CADMINPC**. **CADVAR** should be set to the variance of the frame-to-frame spacings, and **CADVARPC** to **100*CADVAR/CADAVG**.

Some instruments take observation series with a significant difference in cadence between different filters ("A" and "B"), e.g. AAABAAAB. For many instruments, this results in "*skipped"* frames for the A series of images.

CADDROPS should be set to the number of dropped/skipped/missing frames. Such frames should still be "present" in the HDU (filled with blank/missing values), at least in cases where this saves the trouble of using a tabulated form of WCS.

For e.g. synoptic observation series stored with single exposures in separate files, the **CADENCE** keyword should be set to the planned series' cadence. The rest of the keywords should be set as if the HDU had a single frame-to-frame spacing, equal to the time elapsed since the previous *planned* image in the series. **CADDROPS** should be set to zero.

6. Physical description of observational data

The physical description of observational data should *define the contents and boundaries of each measurement:* observation type (e.g. brightness, Stokes parameters, …), units, coordinates (spatial, spectral, temporal), location of the instrument, etc.

6.1. The World Coordinate System (WCS) and related keywords

The World Coordinate System (see the References section) should be used for the description of data coordinates. It is a very comprehensive standard that may describe data coordinates all the way down to the individual pixel level.

Note that the keywords **CROTAn** are *deprecated* by the FITS Standard in favour of the **PCi_j** and **CDi_j** notations, and even explicitly *forbidden* for HDUs that use the **PCi_j** matrix to describe coordinates. This means that any HDU coordinate system that is rotated relative to the data dimensions *should* use either **PCi_j** or **CDi_j**.

Note that all coordinates may be given in tabulated form (Sect. 6 in Paper III), which may be of particular use with e.g. Fabry-Pérot imaging spectroscopy scanning through line profiles with uneven steps in the wavelength direction. Also, WCS allows for specifications of distortions down to a pixel-by-pixel level basis if required (see Paper V).

In ground-based observations, image restoration techniques such as MOMFBD leave behind apparent local movements of image features[12]. Such residual effects represent local errors/distortions in the coordinate system specified by the HDU's WCS keywords.

In the FITS Standard Sect. 8.2, it is specified that a "representative average" of such random errors may be given in the keywords `CRDERi` (for axis number `i`).

If the exact distortions *can* be determined but are not rectified, they may be specified through a WCS convention that allows tabulation of local, random distortions. In normal cases, however, only the typical magnitude of the random distortions is known.

Likewise, representative averages for systematic errors in the coordinates may be given in the keywords `CSYERi` (for axis number `i`). Thus `CSYERi` should be used to represent the uncertainty in the pointing/position of the image as a whole, and uncertainties in the wavelength calibration for spectrometric data.

If a coordinate system has been determined or refined through the use of some external reference image(s) or other source(s), or even been adjusted manually, the keyword `COORDREF` should be used to give a comma-separated list of the images/sources/people. If it is not possible to give specific image names/references, the name of the instrument (and filter, etc., as appropriate) should be given. Since `COORDREF` images must obviously be (near) co-temporal with the data in the Obs-HDU, this should not introduce much ambiguity.

The WCS allows *multiple* sets of coordinate systems to be specified for each data cube. In particular, this can be used to correctly describe data such as rasters, with one system describing the spatial-wavelength coordinate system (`x,y,lambda`), and another describing the temporal-spatial-wavelength coordinate system (`time,y,lambda`). Conversely, imaging observations scanning through the wavelength dimension could have a primary system describing (`x,y,lambda`) and a second coordinate system (`x,y,time`). The multiple coordinate systems are distinguished by a single letter at the end of the relevant coordinate system's keyword names.

The *really* important aspect is that the coordinate system(s) are *accurately described* in the header. Future pipelines should do this *only* through the *full, recommended WCS standard*, not using *deprecated features or other instrument- or mission-specific conventions*, as long as there is an appropriate WCS mechanism. Hopefully, this will in time eliminate the plethora of more or less *ad hoc* solutions for different projects.

Rotating FOVs in movies cannot (currently) be described in a very accurate manner in the WCS standard. The WCS recommendation is to use one coordinate system representing the

[12] In ground-based observations, some of the effects of the atmospheric disturbances may be removed by the use of e.g. MOMFBD methods. This yields a sharper image based on many rapid, subsequent exposures. However, such methods can usually only correct for atmospheric distortions that occur over short periods of time, since it requires that the observed region does not change. Furthermore, they are usually based on local correlations from one image to the other. Such methods also rely on the observed part of the Sun being static. Thus over longer time scales, it cannot distinguish between large-scale variations and changes on the Sun, and only short-timescale effects are corrected. This leaves a "long-timescale" residual in the image restoration - apparent local shifts of features that are in fact not moving on the Sun. Such residual effects represent local errors/distortions in the coordinate system by the WCS system, which are *not known*. Had the distortions been known, they could have been removed, or specified through a WCS convention that allows tabulation of local, random distortions.

position/angle of the FOV at the beginning of the movie and one representing the position/angle at the end of the movie, tacitly assuming a linear (in time) rotation between the two. See Paper IV, Sect. 6.2.5 for more details.

However, given the sometimes highly nonlinear rotation speeds during ground based solar observations, we *also* recommend that the WCS transformation matrix keywords are tabulated using the tabulated-keyword convention outlined in Appendix I: Tabulated-keyword convention when necessary.

For observations (instruments) where the plate scale/pointing is derived from measurements of the apparent solar radius versus the physical size, the keywords **RSUN_OBS** should be used to report the reference value for the physical radius used in the calculations (Paper VII Sect. 8).

WCS keywords explicitly or implicitly mentioned in this section should be specified using the tabulated-keyword convention whenever appropriate (and possible). Note that the tabulated-keyword convention includes specifying a scalar value as well.

6.2. WCS positional and velocity keywords

Ground based observatories should report their geographical location using the keywords **OBSGEO-X**, **OBSGEO-Y**, and **OBSGEO-Z**, implicitly stating that the observer is following Earth rotation (Paper VII Sect. 3; Paper III Sect. 7). In principle, the coordinates should be given in ITRF geocentric coordinates. However, for SOLARNET purposes, GPS coordinates are an acceptable proxy.

For space based observations, the radial velocity of the observer relative to the source cannot be derived from the positional keywords, and it should therefore be reported through **SPECSYSa**[13]**='HELIOCEN'** and the **VELOSYSa** keyword (see Sect. 8.4.1 in the FITS Standard; Paper III Sect. 7). When no absolute wavelength calibration exists, however, this may be skipped *if the velocity does not change significantly during the observation*. If it does, it should be specified using the tabulated-keyword convention (Appendix I: Tabulated-keyword convention).

Earth-orbiting satellites should in addition report their position through **GEOX_OBS**, **GEOY_OBS**, and **GEOZ_OBS**. Contrary to the **OBSGEO-X/Y/Z** keywords, these keywords do *not* implicitly imply that the coordinates are fixed w.r.t. Earth's rotation, but are otherwise identically defined (ITRF, but GPS is an acceptable proxy). For many observations, these keywords must be reported using the tabulated-keyword convention since the spacecraft might move considerably during the observation.

Satellites that are not in Earth orbit should use the keywords **DSUN_OBS** (distance from Sun in meters), **HGLN_OBS** (longitude), and **HGLT_OBS** (latitude) in the Stonyhurst Heliographic system (Paper VI Sect. 2.1 and 9.1).

Although the Solar B0 and P0 angles may be calculated from the date and the observer's position, they should be included in the keywords **SOLAR_B0** and **SOLAR_P0**, to enable easier searches for observations covering the actual north/south poles.

[13] Note that **HELIOCEN** is *not* to be interpreted as the solar system barycentre, but the physical centre of the Sun.

6.3. Data type/units (BTYPE/BUNIT)

It is of course important that each Obs-HDU has a description of *what* the data array itself represents. For this, the **BTYPE** keyword should be used, even though it is not mentioned in any FITS standard document. It is, however, a natural analogy to the **CTYPE/TTYPE** keywords used for the *coordinates* in image/binary table extensions. Similarly, the **BUNIT** keyword should be used to indicate the units used. The units should follow the rules in Sect. 4.3 of the FITS Standard version 3.0.

6.4. Exposure time, binning

The exposure time used in the acquisition of an Obs-HDU should be given in the keyword **XPOSURE** - not in **EXPTIME.** The reason why **EXPTIME** should not be used is that in *some cases* it has been used for individual exposure times in summed multi-exposure observations, introducing an ambiguity. According to the recommendation in Paper IV, **XPOSURE** should always contain the *accumulated* exposure time whether or not the data stems from single exposures or summed multiple exposures.

When the data are a result of multiple summed exposures with identical exposure times, the keywords **NEXPOSUR** and **TEXPOSUR** can be used to indicate the number of summed exposures and the single-exposure time, respectively.

When the **XPOSURE** or **TEXPOSUR** values vary as a function of time or any other of the Obs-HDU's dimension(s) the tabulated-keyword convention (see Appendix I) can be used to specify their exact values as a function of those dimensions. This would typically be the case when Automatic Exposure Control is used - both **XPOSURE** and **TEXPOSUR** could vary as a function of time.

In some situations, it is desirable to increase the effective dynamic range of observations beyond that of the detector or the analogue to digital converter by taking a combination of *alternating* long and short exposures. See Appendix IV: "Post-processing of multi-exposure data with alternating exposures" for a discussion of such cases.

Note that if the data has been binned, the **XPOSURE** keyword should reflect the *physical* exposure time, not the sum of exposure times of the binned pixels. Binning should be specified by the keywords **NBINn,** where n is the dimension number (analogous to the **NAXISn** keywords). E.g. for an observational array with dimensions (**x,y,lambda,t**) where 2x2 binning has been performed in the **y** and **lambda** directions (as is sometimes done with slit spectrometers), **NBIN2** and **NBIN3** should be set to 2. The default value for **NBINn** is 1, so **NBIN1** and **NBIN4** may be left unspecified.

In order to provide a simple way to determine the binning factor, the keyword **BINNING** should be set to the product of all specified **NBINn** keywords.

6.5. Instrument/data characteristics etc.

In order to characterise the spectral range covered by an Obs-HDU, the keywords **WAVEMIN, WAVEMAX, WAVELNTH,** and **WAVEFWHM** should be used.

For spectrometers, the **WAVEMIN/WAVEMAX** values represent the range of wavelengths covered by the Obs-HDU. For filter images, the definition is somewhat up to the discretion of the pipeline

constructor, since effective response curves[14] are never a perfect top-hat function. Bear in mind that these two keywords are primarily meant to be used for search purposes. E.g. if someone wants an observation covering a specific wavelength lambda, the search can be formulated as "**WAVEMIN < lambda < WAVEMAX**". In other words, it might be wise to include more than the "intended" or "nominal" min/max wavelengths of a filter (e.g. sometimes parts of an extended tail should be included if it covers a potentially interesting emission line that is normally very weak, but may be strong under certain conditions).

For a spectrometer, the **WAVELNTH** keyword value is simply the middle of the wavelength range of the HDU: **(WAVEMAX+WAVEMIN)/2**.

For filter images, the **WAVELNTH** keyword should be set to the response-averaged wavelength.

For spectrometric data, the **WAVEFWHM** keyword should be set to the width of the Obs-HDU's data array in the wavelength dimension. For filter images, the value should be set to the response curve's full width at half maximum.

In addition, the keyword **WAVEDESC** may be used for a human-readable description of the (expected) strongest emission/absorption line in HDUs containing spectral observations, or the most dominant line contributing to filter images.

For filter observations where a more thorough specification of the response curve is required for a proper analysis, the tabulated keyword **RESPONSE** may be used. In such cases, the **RESPONSE** Tab-HDU must contain an extra dimension compared to the Obs-HDU. E.g. a movie with dimensions **(x,y,t)** must have a **RESPONSE** Tab-HDU with dimensions **(1,1,1,lambda)**.

For spectrometric data, the resolving power R should be given in the keyword **RESOLVPW**. For slit spectrometers, the slit width should be given in **SLIT_WID**.

6.6. Quality aspects

Many quality aspects of ground-based observations change rapidly, even from one exposure to the next. Keywords that describe such quality aspects must therefore often use the tabulated-keyword convention to specify the time evolution of such values. When a keyword is tabulated, the scalar value should in most cases be the average value of the tabulated values.

Until now, there has been little effort in order to characterise quality aspects of ground-based observations in a manner that is *consistent* between different telescopes, perhaps even between different setups at the same telescope. The suggestions below must therefore be seen as quite temporary suggestions, subject to changes and amendments as the development of common pipelines progresses. With that caveat, we propose the following keywords for data quality aspects:

R0 should be used to specify the atmospheric coherence length r0 in cm. If the effective value varies within the field of view, the value for the best part of the image should be used.

R_SEEING (raw seeing, in arcsec) should be used to describe the seeing at the time of data acquisition (including the effects of the telescope and adaptive optics). This value should *not* include the effects of any post-processing such as MOMFBD. In principle, the value should be calculated as the square root of the normalised 2nd central moment of a point source image,

[14] A response curve takes into account the wavelength-dependent transmission of all optical elements and the sensitivity of the detector.

which is equal to sigma when applied to a standard Gaussian function. In practice, other methods will have to be used to estimate this value. When the seeing (as defined here) varies as a function of the field of view, the value for the best part of the image should be used.

If techniques such as MOMFBD have been applied to the data (i.e. the Obs-HDU does not contain the original data), the **R_SEEING** values for each of the underlying images may be specified by adding one or more dimensions compared to the Obs-HDU when using the tabulated-keyword convention. E.g. a movie **(x,y,t)** may have **R_SEEING** tabulated with dimensions **(1,1,t,n)**, where n is the number of images used to produce each **(x,y)** image.

F_SEEING ("final seeing" or "effective seeing") should be used to report the corresponding effective seeing of the data in the Obs-HDU (i.e. after post-processing).

The exact methods of measuring **R0** or calculating/estimating the **R_SEEING**, and **F_SEEING** values are not important - as long as the methods are *well-defined and used in a consistent manner across all observations of similar types*. We leave the practical implementation up to the pipeline developers. Conventions on how to determine the values will be subject to discussions later on in the SOLARNET project period.

ANG_ELEV: This keyword should be used to quote the telescope's angle of elevation at the time of data acquisition.

The keyword **AO_LOCK** should be used to indicate the status of any adaptive optics. The values should be 0 (no lock) or 1 (lock).

If automatic feature tracking is used, the keyword **FT_LOCK** should be set to 1 when the system is indeed tracking a feature, or 0 if not (i.e. no tracking lock achieved).

The keyword **ROT_COMP** should be set to 1 if automated solar rotation compensation is in effect, and to 0 if not. The keyword **ROT_ALG** should be set to the name of the algorithm used for rotation compensation.

LOG_LOC: Location of the log file that is relevant to this observation, if available. It may be a path that is relative to the directory in which the **INFO_URL** document is located (see Section 10) or it may be a full URL. E.g. if **INFO_URL="http://instr.site.org/info.html**", and the log files relevant to this observation is stored in "**http://instr.site.org/logs/2014/11/04/**", then **LOG_LOC** could either be "**2014/11/04/**" or "**http://instr.site.org/logs/2014/11/04/**".

COMMENT: May be used to include the relevant parts of the **OBS_LOG**, and any other relevant comments about the HDU that may be useful for the interpretation of the data.

6.7. Data statistics

It may be useful to have statistics about the contents of the Obs-HDUs in order to search for "particularly interesting" files (or to filter out particularly *uninteresting* files for that matter). **DATAMEAN** should contain the average data value, **DATARMS** the RMS deviation from mean, **DATASKEW** the skewness, **DATAKURT** the kurtosis, **DATAMAX** the maximum value, **DATAMIN** the minimum value, and **DATAMEDN** the median value. Also, the keywords **DATAPnn** (where nn represents a percentile: 01, 10, 25, 75, 90, 95, 98, and 99) should be present, as these may also be useful in order to scale large data sets for display purposes without first scanning through all pixels.

6.8. Missing and saturated pixels, spikes/cosmic rays

In some data sets, the data in the HDU may be affected by missing/lost telemetry, acquisition system glitches, cosmic rays/noise spikes, or saturation, etc. Some keywords are useful to find/exclude files based on how many such pixels there are. Consider an Obs-HDU containing a movie that pans left to right over an active region, with padding on the sides of the actual FOV to make the active region seem stationary when "playing back" the HDU array. The keyword **NTOTPIX** is set to the total number of *expected data pixels* in the HDU - not including pixels that stem from e.g. padding. The keyword **NDATAPIX** gives the *actual* number of usable data pixels in the HDU - excluding all pixels identified as missing, lost, contaminated, or otherwise deemed unusable. **NMISSPIX** is defined as **NTOTPIX-NDATAPIX.**

Since missing pixels due to missing/lost telemetry and acquisition system glitches typically occur in large chunks, it is normally impossible to fill them in with estimated values based on values from the surrounding pixels. Missing pixels due to cosmic rays/noise spikes, saturation or dust contamination, however, are usually more "benign" in this respect. Thus the keyword **NLOSTPIX** should be set to the number of lost pixels (telemetry/acquisition problems) and the percentage of these relative to **NTOTPIX,** respectively.

In addition, the number of saturated pixels may be of interest, reported in the keyword **NSATPIX.** Finally, the number of identified cosmic ray/noise spike pixels should be reported in the keyword **NSPIKEPIX.**

Corresponding percentages relative to **NTOTPIX** are given in **PCT_DATA, PCT_SATP, PCT_SPIK, PCT_LOST**, and **PCT_MISS.**

There are two principally different methods of handling any identified missing/blank or otherwise "unusable" pixels in FITS files to be distributed: The first is to flag them with a special value specified by the keyword **BLANK** in integer-valued HDUs or setting the pixels to **NaN** in a floating-point HDU (**BLANK** should not be set in a floating-point-valued HDU). Analysis programs must be able to recognise both of these FITS Standard conventions.

The second method is to fill the missing/blank pixels based on valid data in the surrounding pixels.

The **FILLED** keyword must be set to 0 if no filling has been performed or 1 if any form of filling has been performed. Filling algorithms are free to decide that a pixel should not be filled in because of technical reasons, and should then replace any original value with **NaN** or **BLANK.** The algorithm(s) used to fill in data should be apparent from the **PRxxxxnn** keywords in Section 9 (Pipeline processing applied to the data).

6.8.1. Optional listing of missing, saturated, and spike/cosmic ray pixels

For some purposes, it may be useful to list the location and original values of the pixels that have either been filled or are left **BLANK**/**NaN**.

The simplest method is to use a single Index-value HDU (Appendix III: Index-value HDU convention) with locations and original values, not distinguishing between the different types of pixels. This Index-value HDU name must end in "**;MISS_IDX**", and the colon-separated list of extension names preceding the semicolon must contain the name(s) of the HDUs to which the Index-value HDU applies – e.g. **He_I;MISS_IDX**. When original values are not available, **NaN** may be used, or no value may be given at all (**m=0** in Appendix III: Index-value HDU convention).

If it is desirable to specify which pixels are in the different categories, lost pixels may be listed in an Index-value HDU correspondingly named e.g. **He_I;LOSS_IDX**, saturated pixels in **He_I;SAT_IDX**, and spike/cosmic ray pixels in **He_I;SPIK_IDX**.

7. Where, how, who, why, what

The keywords in this section describe various pieces of metadata regarding the hierarchical origin of the data, the origin's affiliation organisation, structure and affiliation of the data. Although not generally required for the *use* of the data, such metadata can be useful w.r.t. searching and grouping/counting/reporting.

7.1. From where to how, PROJECT to SETTINGS/OBS_MODE

Some of the keywords below describing the hierarchical "physical affiliation and origin" of the data will not make sense for all data sets, because the depth, nature, and nomenclature of the hierarchies varies, e.g. between spacecraft and ground based observatories (GBOs). For some keywords, we therefore specify default values to be used.

PROJECT: If the "hardware origin" of the data has an overarching "hardware project affiliation" above the level of **MISSION/OBSRVTRY**, this keyword should be used to specify it - e.g. "Living With a Star". A **PROJECT** would typically be associated with more than one mission, observatory or spacecraft pursuing the same goal but not necessarily *identical* or even of the same type (e.g. one remote sensing, one *in situ)*, and they do not have to be closely coordinated. GONG and NSO qualify. Projects that have *not financed any specific hardware used* are not meant to be listed here. May consist of a comma-separated list of projects if necessary, and in that case in order of decreasing scope. Default: **MISSION.**

MISSION: Typically used for one or more closely coordinated or cooperating spacecraft, such as "STEREO" for STEREO A and B, CLUSTER, IRIS, etc. For ground based observations, GONG would qualify. Default: **OBSRVTRY** (note recursion).

OBSRVTRY: Might be best explained as "physical observation platform", i.e. the collective name for the "location" of *one or more* telescopes (telescope group) or other types of instruments/detectors. Examples are SST, SOHO, STEREO A or STEREO B, etc. The National Solar Observatory does *not qualify*, as it is a multi-observatory *PROJECT*! Default: **MISSION** (note recursion).

TELESCOP The name of the optical system used to capture photons – typically a value that can be inserted in the phrase "the <**TELESCOP**> main mirror". E.g. CDS/NIS and CDS/GIS, SST, IRIS. For non-telescope instruments/detectors like particle instruments, should be the collective name most applicable to the entry path of the particles.

TELECONF: Telescope configuration. A name that uniquely identifies the telescope configurations - which might easily change for GBOs.

INSTRUME: The word "instrument" has somewhat different meanings in space based and ground based observations: For ground based observations, there may be many different instruments for each telescope, e.g. SST telescope has instruments TRIPPEL and CRISP, and the suite of instruments may be "easily" changed. The opposite may be the case in space based observations, e.g. the "instrument" LASCO has three different telescopes (C1, C2 and C3). Thus the hierarchical ordering of **INSTRUME** vs. **TELESCOP** will differ in different situations, and the definition and interpretation of **INSTRUME** must be somewhat heuristic. However, since the

values of these keywords are anyhow project-specific, we will leave it up to the pipeline designers to determine the exact definition (matching the meaning of the word “instrument” within the project). Examples: CDS, LASCO, EIT, CRISP, TRIPPEL.

GRATING: When a spectrometer instrument has data from different gratings, the applicable grating name should be specified in this keyword.

CAMERA: A camera normally has a single detector, but it may have more than one detector. E.g. spectrometers with different detectors for separate regions of a spectrum such as CDS (NIS1/NIS2 and GIS1/2/3/4) has two cameras: NIS and GIS. The precise definition is left to the pipeline designers, although a typical description would be “the space between the last optical element (slit/mirror/grating/aperture) and the detector”.

DETECTOR: Uniquely identifying the detector(s) used to collect the data in the Obs-HDU. Note that the existence of this keyword implies that data from two detectors inside a single camera should not normally be combined in a single Obs-HDU. The only exception is when the HDU has been constructed such that the data appears to be from a contiguous detector, padding in pixels to remove coordinate jumps that may occur between the detectors. In addition, the detector specifications should be identical as well (e.g. no differences in coating etc). In such cases, **DETECTOR** should contain the name of a “virtual detector” that refers to all physical detectors in the Obs-HDU.

FILTERn: Name(s) of the filter(s) used during the observation (name of filter).

SLIT_WID: As described earlier, the width of any slit, in arc seconds.

OBS_MODE: A name that uniquely identifies a set of all settings used during the observations, from telescope optics down to the detector and *including* any acquisition processing of the data before it is recorded.

SETTINGS: All relevant settings, ideally everything *not described in other keywords above* needed to reproduce observation. The format should be a comma-separated list of parameter values in the form of “**PARAMETER1=xx,PARAMETER2=xx**” in *alphabetical order*. *For composite data* (if the HDU is a result of combining data from multiple detectors), all keywords from **PROJECT** down to **DETECTOR** *may* contain comma-separated lists.

7.2. Who and why? (And a note about “free text”)

First of all, we *strongly* recommend that all “free-text” keywords are filled in from a strictly controlled list of predefined texts. Experience has shown that free-text fields will be filled in incredibly inconsistently, even the writer’s own name. Nevertheless, for free-text searching purposes, something *may* be better than nothing - but keep this in mind: Finding N entries with **OBSERVER=”John Doe”** might lead to the conclusion that John Doe has been the observer for (only) N studies - not catching the fact that he has sometimes written his name as **”J. Doe”.** Furthermore, it is hard to say whether **”J. Doe”** refers to Joe or Jane Doe.

In order to track or attribute data taken or requested by specific people or organisations, the following keywords may be used:

OBSERVER: Comma separated list of operator(s) who acquired the data.

PLANNER: Comma separated list of observation planner(s).

REQUESTR: Comma separated list of who requested this particular observation instance - i.e. who requested the data in this HDU to be taken.

SREQUEST: Comma separated list of who requested the design of the observation sequence.

SCI_OBJ: Comma separated list of science objectives.

CAMPAIGN: Coordinated campaign name/number, including instance number, when applicable. If used consistently, can be used to tie together *all files from all sources* taking part in a coordinated campaign. In some cases, multiple "competing" campaign numbering schemes may apply, or a single observation might be part of multiple campaigns. In such cases, they should all be entered as a comma-separated list.

ORGANISA: Comma-separated list representing the hierarchy of organisations "in charge" of these observations, starting at the top of the hierarchy. The "principal responsible person" at each level *may* be included in parentheses behind the institution name(s). E.g. the PI of the experiment may be mentioned in parentheses behind his/her institution. The list should reflect the hierarchy from **PROJECT** to **DETECTOR**, as applicable, but should also include organisations that are responsible for planning and acquisition of the observations. I.e. it does not necessarily stay "within" a single top-level organisation (like ESA, which might be in charge of a mission, but not in charge of the instruments/sub-instruments). Where multiple institutions are in charge at the same level, separate them with a slash.

8. Grouping

As mentioned in the description of an ideal SVO, an important facet of how archive users interact with an archive is the ability to group the search results. E.g. if a search matches 1000 HDUs, but they are part of only 5 different observation series (e.g. different filters or wavelength regions), it makes sense to have an optional grouping mechanism to "collapse" the result listing into only 5 lines, showing some form of "summary" of the underlying files for each series.

It should be possible to group HDUs in different ways, e.g. whether HDUs in the same file should be reported on separate lines or not, or whether HDUs with observations from a single instrument in different filters should be reported on separate lines or not.

The logic behind grouping is that all members of the same group have identical values for *a particular set of keywords*. Thus the "graininess" of the grouping method depends on which keywords are included in the set.

In other words, the ability of an SVO to provide useful grouping choices depends on the availability of the proper keywords.

Note that in order to grasp the mechanics behind the system, these keywords should not be thought of as "grouping keywords", but rather "group splitting keywords" (normally concatenated into a "splitting string"). Using an *empty* group set, the splitting string is empty for *all* HDUs in the search result, thus they belong to a single group and will be summarised on a single line.

But when *adding* e.g. **PROJECT** to the group set (i.e. adding **PROJECT** to the splitting string), the results will be split into one line for each project (e.g. GONG, LWS, ISOON etc). When further adding e.g. **MISSION**, **OBSRVTRY**, **TELESCOP**, and **INSTRUME**, the results will be split into one line for each physical instrument with observations matching the search. By further adding the value(s) of **FILTERn** the result listing will be separated into observations in different filters/filter

combinations. Thus by adding more and more keywords to the splitting strings, more and more fine-grained grouping may be achieved, all the way down to meta-Obs-HDUs, and finally all individual constituent Obs-HDUs (i.e. no grouping at all).

Note that with a splitting string of e.g. "**PROJECT,.....,INSTRUME,FILTERn**",*all* observations over the lifespan of an instrument with a given (set of) filter(s) will be lumped together in a single group!

We therefore introduce a "pointing id", to be reported in **POINT_ID,** which must be given a new, unique string value[15] every time the telescope is significantly *repointed* - not counting feature tracking or rotation compensation. In other words, successive files "interrupted" by sudden (not continuous) solar rotation compensation/feature tracking jumps may share the same **POINT_ID.** HDUs sharing the same **POINT_ID** *may* have quite different fields of view, e.g. a series of overview images of a sunspot and a simultaneous series with a much smaller field of view focussing on the edge of the sunspot.

An SVO may now choose to make available different grouping methods based on a number of different splitting string compositions. For the most part, the splitting strings may be constructed in a hierarchical sense, but there may also be some special cases where the hierarchy divides into different branches. In fact, it is entirely possible to create a system where the splitting string is entirely up to the user, by choosing freely among all the keywords stored in the database - even including those that are instrument-specific.

9. Pipeline processing applied to the data

The use of "data level" as a description of the processing applied to different data sets has proved to be extremely instrument-/mission-/pipeline-dependent, and is therefore not very useful in terms of selecting and understanding the data.

Instead, a mechanism that is as uniform as possible across all pipelines must be used. This can be achieved by specifying an exhaustive list of *each* well-defined processing step that has been applied to the data. The steps may be e.g. "FIXED-PATTERN", "FLATFIELDING", "CALIBRATION", "GEOMETRY", "DESPIKE", "FILL-MISSING", etc. It is also important to ensure repeatability of the processing applied. Each unique step must be specified in the following way (nn is a number between 1 and 99), e.g.:

PRSTEPnn = 'FLATFIELDING' ← Need input from pipeline people for an "exhaustive" list!
PRPROCnn = 'name_of_procedure' ← name of the procedure/function/routine invoked
PRLIBnn = 'CRISPRED' ← Name of software library used. See below for multiple libraries.
PRVERnn = 1.5 ← Numeric value specifying the library "version"
PRBRAnn = 'branch_name' ← if library stems from a version control branch (other than trunk)
PRPARAnn = 'ITERATIONS=5' / List of parameters/options for the **PRPROCnn** procedure.
PRxxxnn = anything else necessary to specify this step, with a descriptive comment

Since multiple libraries may be used in a single processing step, it is also necessary to specify these, including their versions and variations. E.g. a flat-fielding routine in an instrument-specific library may call a generic routine from the SolarSoft library or some IDL library routine. In such cases, these libraries must also be specified for this step:

[15] Typically, a string giving the date and time of the repointing.

PRLIB01A = 'SSW'
PRPKG01A = 'gen xrt ontology cds iris' ← $SSW_INSTR (packages, in path order)
PRVER01A = 56918.25 ← Modified Julian Date (MJD) for last update/modification
PRBRA01A = '' ← Not necessary, since SolarSoft does not (currently) have branches
PRxxx01A = anything else necessary to specify this step, w/descriptive comment
PRLIB01B = 'IDL'
PRVER01B = 8.2001 ← IDL version 8.2.1 (incl. standard library idl/lib)
PRPKG01B = 'astrolib coyote' ← IDL extra libraries (if applicable)

The order in which the different libraries occur in the path should be reflected in the numbering (lettering) of the libraries: In the example above, the CRISPRED library occurs first in the path, SSW is second, and IDL's libraries occur last. I.e. in case of naming conflicts, the CRISPRED library takes precedence over SSW, which again takes precedence over IDL's libraries.

"Umbrella" procedures that invoke multiple processing steps such as flatfielding *and* calibration should be mentioned with a **PRSTEPnn** value set to a comma separated list of the **PRSTEPnn** values for the steps invoked by the procedure (in the order they were executed). This makes it possible for users to see at a glance that some standard (multi-step) procedure has been used. The individual steps invoked must also be listed as separate **PRSTEPs**. The umbrella procedure must be referenced before the first individual processing step performed by the procedure. Note that each individual processing step must be identifiable through a separate procedure call that can be specified in the **PRPROCnn** keyword.

The **PRCVERnn** keyword must be numerically increasing with increasing "maturity" of the library in question. Note that we are talking about *production* pipelines - development versions should not be used to populate an SVO. Production versions should be assigned some form of numeric version number. When using libraries with no (numeric) version numbers, the Modified Julian Day (MJD) of the time the library was last mirrored/changed should be used as a version number.

In addition, **PR_MODE** should be set to a value that uniquely identifies the "processing mode", when appropriate: Some pipeline types may be run with different trade-offs between e.g. signal to noise ratio versus spatial resolution or contrast. This should already be apparent from the other keywords, but **PR_MODE** provides a much simpler way of identifying data processed in a particular way (e.g. "BALANCED" or "HIGH CONTRAST"). **PR_MODE** does not need to uniquely identify processing done with different library versions, unless this results in a significant shift in the results with regard to the trade-offs. Note that a single observation may be registered multiple times in an SVO with different values of **PR_MODE** - but then a **PR_MODE**-specific identifier must be part of the file name.

Also, **VERSION** should be set to the processing version, an integer that should be increased whenever a reprocessing is performed in order to improve the data set (e.g. with a better flatfield, better detection of cosmic rays, etc). The version numbers in files published through an SVO may increase by more than one for each new published "generation", allowing the use of intermediate values for internal/experimental use.

ORIGIN should be set to a character string identifying the organization or institution responsible for creating the FITS file. **DATE** should be set to the date of creation of the FITS file.

10. Fixity, integrity administrative information

When the final version of a file is produced, **FINAL** should be set to 1. Naturally, it is difficult to know whether this is true or not - there may be developments in the future that enables an improvement in the processing. The exception, of course, is whenever the pipeline input data is to be deleted after the processing.

The **DATASUM** and **CHECKSUM** keywords (see References) should be set in all HDUs to allow a check on whether the data file has been modified from the original or has become corrupted. However, their values in a meta-HDU will be recomputed when constituent HDUs have been combined into a single HDU (after checking the constituent HDUs **DATASUM** and **CHECKSUM)**.

INFO_URL should point to a human-readable web page describing "everything" about the data set: what it is, how to use it, links to e.g. user guides, instrument/site/telescope descriptions, descriptions of caveats, information about data rights, preferred acknowledgements, whom to contact if you have questions, and repositories of observing/engineering logs.

Upon ingestion of data into an SVO, the material pointed to by **INFO_URL** and **LOG_LOC** (Section 6.6) might be "harvested" and preserved in such a way that it is possible to retrieve a copy even if the original source is no longer available. It might be possible for an SVO to recursively harvest pages/documents and even auxiliary data such as flatfields being linked to from **INFO_URL**. The harvesting will have to be restricted somehow - presumably limited to links pointing beside or below **INFO_URL**[16] and **LOG_LOC**.

Any other administrative information pertaining to the file should also be included at the **INFO_URL.**

11. Reporting of any events detected by the pipeline

If there are feature/event detection algorithms that will only work on data that are less refined (rawer) than the final pipeline product, they should be performed inside the pipeline. The detected events/features should be reported to relevant event registries following the appropriate standards (e.g. VOEvents).

The reported events (or list of events) should refer to the pipeline output file. Events should be flagged in the file metadata by grouping them into each event/feature type (also identifying the algorithm/version used). The type and number of each type should be reported as:

```
EVIDnnnn = 'CME-875578'         / Event ID (from registry, when available)
EVTYnnnn = 'CME-DETECTED-BY-X' / Identifies event type, algorithm/version
EVnnnn_i = 2                    / Specifies i'th coordinate value for event
EVTYPmmm = 'CME-DETECTED-BY-X' / Single entry for each event type
EVTYNmmm = 10                   / Total number of such events
```

The **EVnnnn_i** keywords specify *where* in the HDU's data array the event has been detected, by giving the value for the **i**'th coordinate. E.g. if an event is detected centered on **x=100** and **y=500** in the 4th image of a series of images **(x,y,t)**, we might have **EVnnnn_1=100,**

[16] E.g. with **INFO_URL='http://some.site/this/guide.html'**, documents **http://some.site/this.manual.pdf** and **http://some.site/this/subdirectory/auxiliary.dat** might be harvested, but not **http://some.site/other/use.pdf**.

EVnnnn_2=500, and **EVnnnn_3=4.** If x and y coordinates are not determined (e.g. the event is not located in a specific place in the image), **EVnnnn_1** and **EVnnnn_2** are simply not present.

Information about events/features that can be detected in the final data set should not be included in the output file, because an improved (or just debugged) algorithm could later reprocess this data, thereby making the information obsolete. They should, however, be reported to the relevant event registry.

12. Other keywords/rules

DATE-BEG must be given, referring to the *start* of the observations contained in the HDU. However, if for technical reasons the cadence of a time series is uneven, it is recommended to also give this keyword using the tabulated-keyword convention, specifying the start of each repetition as a function of the time dimension.

DATE-END must be given, referring to the *end* of the observations. See **DATE-BEG** for the recommended handling of uneven cadences.

FILENAME must contain name of the original file containing the HDU - even if the HDU has been extracted from this file and put into a file of its own.

Keyword values *must* - unless otherwise specified - be given in units following the rules in Sect. 4.3 of the FITS Standard version 3.0, and the unit *must* be given in the keyword comment section (see Sect. 4.3.2 of the FITS Standard). Dimensionless keywords are exempt, of course.

Keywords should have an appropriate short description in the comment field.

LONGSTRN='OGIP 1.0' should be used, in order to allow keywords to contain strings longer than 68 characters, see OGIP 1.0. E.g.:

```
LONGSTRN= 'OGIP 1.0'           / The OGIP Long String Convention may be used.
STRKEY  = 'This is a very long string keyword&'  / Optional Comment
CONTINUE  ' value that is continued over 3 keywords in the &  '
CONTINUE  'FITS header.' / This is another optional comment.
```

13. Appendix I: Tabulated-keyword convention

In some cases, values given by keywords will be a function of time or some other dimension(s). E.g. if automatic exposure control (AEC) is used, the exposure time will be a function of time, and in a Fabry-Pérot scan, the exposure time may be a function of wavelength.

Consider an Obs-HDU containing an image sequence with dimensions **(x,y,t)=(120,120,100)**, where the exposure time varies as a function of time. To cover such cases, we define a generic convention to specify how keywords vary as a function of the data cube dimensions:

The keyword **TAB_HDUS** should set to a comma-separated list declaring keyword names and HDU names where the corresponding values are tabulated. E.g., the header of the above Obs-HDU might contain the following entries:

```
TAB_HDUS = 'He_I:He_II;XPOSURE=4' / Tabulated keywords, HDU names
XPOSURE  =  5.41                  / [s] Average exposure time
```

This means that the tabulated values of the keyword **XPOSURE** applying to Obs-HDUs **He_I** *and* **He_II** are to be found in a Tab-HDU with **EXTNAME = He_I:He_II;XPOSURE**, and that this Tab-HDU is HDU number 4.

The HDU with tabulated values *must* have an *equal or larger number of dimensions* as the HDU that refers to it, and the *order* of the first N dimensions must be the same, where N is the number of dimensions in the referring HDU.

But the Tab-HDU dimensions do not need to have the same *sizes* as the referring HDU. E.g. **XPOSURE's** Tab-HDU would typically have dimensions **(1,1,100)** to indicate that the **XPOSURE** time is constant over the entire image. It is even possible to shrink non-singular dimensions: **XPOSURE's** Tab-HDU may e.g. have dimensions **(1,1,15),** which means that linear interpolation will be used to assign values to the **XPOSURE** keyword for each time step.

When the HDU containing tabulated keyword values contains *more* dimensions than the HDU that refers to it, the extra dimension(s) must be *appended* to the referring HDU's dimensions. For an example of such uses of the convention, see the discussion of accumulated vs. individual exposure times (**XPOSURE** vs. **TEXPOSUR**) in Section 6.4 (Exposure time, binning).

Note that multiple Obs-HDUs may refer to the same Tab-HDU using this convention.

In order to keep backwards compatibility with utilities that do not follow this convention, scalar values for tabulated keywords *must* also be given. In most cases, the most sensible scalar value will be the average of the tabulated values, but in some cases other values may make more sense. We leave this decision to the discretion of pipeline developers, with the *recommendation* that the average value be used whenever possible. When such a decision is to be made, the case should be disseminated to the community for discussion in order to ensure consistency.

Functions that extract keyword values from FITS headers may be extended to accept an *option* indicating that the caller is aware of this convention and that the tabulated values should be returned, after any applicable (optional) linear interpolation has been applied.

14. Appendix II: Meta-observation convention

Most users expect to be able to analyse at least one file at a time on a laptop, preferably with all of the data loaded into memory. Thus at some point, files become too large for comfort[17] when following the guidelines for what to store in a single file/single Obs-HDU in a strict sense.

An obvious solution to this problem for a file that contains multiple Obs-HDUs would be to split it into multiple files containing only a single HDU each. However, high-cadence, high-resolution observations often produce extremely large amounts of data, and at some point this strategy will not be enough to keep file sizes reasonable. Thus the issue of prohibitively large files *must* be dealt with in a more generic way while preserving the "spirit" of the guidelines for what should be stored together.

We do this by providing this convention to logically connect HDUs stemming from a single observation series that *ought* to be put in a single HDU according to the guidelines. It allows HDUs to be split into smaller constituent HDUs, stored in separate files and individually recorded in an SVO as separately retrievable observation units, whilst *also* recording the entire

[17] Processing easily doubles the size of data to be kept in memory. Data cubes prepared for immediate visualisation are often accessed using memory-mapped files, though, so files of "arbitrary size" are ok.

observation as a meta-observation unit, without duplicating the data. The meta-observation unit reflects the observation's global attributes like duration, data statistics etc. for searchability reasons, but the data are only retrievable as a collection of constituent files.

In this appendix we only discuss splitting observations in the *time dimension*. Although it is possible to use the convention to split observational data in any other dimension, this seems likely to be confusing for users, and we therefore do not recommend it.

To give an example: A series of 10000 images might have to be split over 10 files, each having a single Obs-HDU containing 1000 images, with any accompanying Tab-HDUs, Index-value HDUs, or Auxiliary HDUs. All HDUs in those 10 constituent files should be self-consistent, and it should be possible to analyse each file independently. In fact, each file should be created just as if it were not part of such a meta-observation[18], except for some additional information and some additional rules/metadata.

The convention is designed to make it possible to construct a *stitching utility* that can collect constituent HDUs into "ideal" HDUs that *would* have been created if it were not for file size considerations. Here we describe the additional rules and metadata required to make such a stitching utility work.

First of all, constituent HDUs must have the same **EXTNAME** in order to be stitched together. I.e. **He_I** HDUs inside the constituent files will be stitched together into a new **He_I** HDU, and **He_II** HDUs will be stitched together into a new **He_II** HDU.

In addition, *all* HDUs that are part of a meta-observation *must have the following additional keywords:*

- **METAFIL** must be set to "<**filename1**>", where <**filename1**> is the name of the *first* file that contains parts of the observation unit, and **EXTNAME** is the value of the HDUs' **EXTNAME** keyword[19].

- **METAFILS** set to a comma-separated list of all files that contain data for this meta-observation unit, in sequential order.

- **METADIM** set to the dimension that has been split. E.g. when splitting an array **(x,y,t)** into time chunks, **METADIM=3**. Note than an accompanying Auxiliary HDU with dimensions **(t,y)** would set **METADIM=1**. **METADIM=0** should be used for e.g. Auxiliary HDUs whose data array dimensions does not contain the split dimension. It is allowed to have **METADIM=NAXIS+1**. E.g. if constituent HDUs have dimensions **(x,y,lambda)** and **METADIM=4**, the output meta-HDU will have dimensions **(x,y,lambda,t)**. Note that the meta-HDU in the constituent files must have a set of WCS keywords that correctly describe the resulting array, including any added dimensions.

- **METANUM** set to 1 for the first file containing parts of the meta-observation unit, 2 for the second file, etc.

[18] The file names should also reflect this - e.g. the date/time part should reflect the start of the observations contained in each constituent file.
[19] Since file names are unique and **EXTNAMEs** are unique within a file, the resulting string is unique for all meta-observations.

- **METASTRI** set to the "stride" in the split dimension between each constituent HDU, i.e. 100 in the case above. Should be set to 0 for Meta-HDUs.

However, in order to correctly register a meta-observation unit, the global values of all keywords (including instrument-specific) must also be available.

This is of course only a problem for those keywords that are not constant among all the constituent HDUs. E.g. **DATE-BEG**, **DATE-END** and other keywords that vary as a function of the split dimension, or as a function of the data itself (e.g. **DATAMAX** and **DATAMIN**) must be given explicitly for the meta-observation. Also, the global scalar value of any tabulated keywords must be given.

This is achieved through a *Meta-Obs-HDU* containing the meta-observation unit's keyword values - i.e. the values that would have resulted from storing the meta-observation in a single HDU. The **NAXISn** keywords are exempt, since the Meta-Obs-HDUs should contain a singular data array. This Meta-HDU should occur in each file containing any constituent HDUs[20].

The **EXTNAME** of such a Meta-Obs-HDU *must* be the same as the (common) **EXTNAME** of the constituent HDUs, with the string "***;METAHDU***" appended.

Using the keywords given above and the Meta-Obs-HDU, it is now possible to reconstruct/stitch together constituent Obs-HDUs into an ideal Obs-HDU with a correct header. It is also possible to reconstruct their corresponding Tab-HDUs and Index-value HDUs.

The logic behind the stitching is as follows, given a collection of files that make up a meta-observation:

The name of the output file containing the stitched meta-observation will be "**<filename1>_META.fits**", where **<filename1>** is the name of the first constituent file. Below is a pseudo-code description of the subsequent steps in the stitching:

```
FOR each file (in the order given):
      FOR each HDU/EXTNAME:
            IF SOLARNET=0 OR HDU is a META-HDU:
                  Keep HDU, replacing any earlier HDU with same EXTNAME
            ENDIF ELSE:
                  IF HDU is an Index-value HDU:
                        Adjust index in the split dimension using METASTRIDE
                  ENDIF
                  Collect/aggregate HDU along split dimension (given by
                  METADIM) with earlier HDUs with same EXTNAME
            ENDELSE
      ENDFOR
ENDFOR
```

For all types of HDUs, only the *last* encountered *header* is preserved, but the **NAXIS** and **NAXISn** keywords are recomputed to match the stitched (aggregated) data array.

When finished, there is one HDU for each distinct **EXTNAME** that occurred in the file collection.

[20] Although from an SVO standpoint, such (identical) meta-HDUs are only required in one of the files, they do not take up much storage space since they only contain keyword values, and they might be useful to people who are analysing constituent files one file at a time.

The final step is to go through all Meta-HDUs, i.e. HDUs with **EXTNAMEs** ending in "**;METAHDU**". The "**;METAHDU**" string is clipped off the **EXTNAME.** If one of the processed HDUs has the resulting string as its **EXTNAME,** its *keywords* will be *replaced* by the keywords in the Meta-HDU. However, FITS keywords in the aggregated HDU describing the data array size (**NAXIS, NAXISn)** are *not* replaced, since they must represent the size of the data array in the target HDU (not the size of the singular data array in the Meta-HDU).

In other words, the guidelines for what to store together in a single file/single Obs-HDU may be followed strictly, interpreting them as guidelines for what to store together in a single meta-file/single meta-Obs-HDU.

15. Appendix III: Index-value HDU convention

Index-value HDUs are used to store lists of pixel indices and (optionally) one or more data values applying to specific pixels within the data array of another HDU ("the referring HDU").

An Index-value HDU's **EXTNAME** must be of the form "**He_I:He_II;MISS_IDX**". The first part (**He_I:He_II**) is a colon-separated list of the referring HDUs' **EXTNAMEs**.

The second part (**MISS_IDX**) is freely chosen seen from the Index-value convention's viewpoint, but not from an "application viewpoint" (see section 6.8.1). We recommend that the **EXTNAMEs** of all Index-value HDUs end in "**_IDX**".

If an Index-value HDU applies to an HDU with **N** dimensions, the Index-value HDU should be a table of size **(N+m,n)**, where **n** is the number of pixels to which the Index-value HDU applies, and **m** is the number of values to be associated with each of those pixels. For each pixel **i=1...n** the first **N** values in the row are pixel indices, and the remaining entries are the associated pixel value(s). If no value is associated with the pixels, **m=0**.

As an example, consider an image contained in Obs-HDU **He_I** with two identified and filled-in noise spike pixels, one in location **(5,10)** whose original value was **5.5**, and another pixel at **(100,2)** whose original value was **10.3**.

This would result in an Index-value HDU named **He_I;SPIK_IDX**, whose contents may now be represented as a table with three columns **(N+1)** and two rows (the number of noise spike pixels **n=2)**:

```
  5,      10,       5.5      (entries [1:3, 1])
100,       2,      10.3      (entries [1:3, 2])
```

Note that such Index-value HDUs *must* have the **METAxxx** keywords set in order to function correctly within the meta-HDU framework (see Appendix II: Meta-observation convention for details).

16. Appendix IV: Post-processing of multi-exposure data with alternating exposures

In some situations, it is desirable to increase the effective dynamic range of observations beyond that of the detector by taking a combination of *alternating* long and short exposures.

However, recording both exposures individually would double the data rate to be recorded or transmitted. It would also increase the complexity of the data set, either by adding one dimension to the Obs-HDU, or separating the time series into separate Obs-HDUs.

This can be avoided by producing a combined image. The processing may be done either on-board/on-site (which decreases the amount of data to be transferred/recorded) or in a later pipeline function - in order to produce a simpler data set for the end user.

In such cases, the **TEXPOSUR** keyword must be stored in a Tab-HDU which has an extra dimension compared to the Obs-HDU. E.g. if the Obs-HDU is a movie **(x,y,t)**, the individual exposure times (**TEXPOSUR**) must be recorded in a Tab-HDU with dimensions **(1,1,1,2)**. If the **TEXPOSUR** values also change with *time* (e.g. due to AEC), the Tab-HDU must have dimensions **(1,1,t,2)**.

In order to construct the non-overexposed image with higher effective dynamic range, the saturated pixels from the long exposure must be replaced with appropriately scaled-up values from the short-exposure data[21].

Note that if accurate signal to noise ratios are required, all pixels where such a replacement has occurred will have a lower signal to noise ratio than what the numbers indicate. This issue can be (mostly) dealt with by including a keyword called **PIXSAT** (the saturation value). The original short-exposure pixels may then be identified (values greater than the **PIXSAT** value), and their SNR values reconstructed by subtracting **PIXSAT** and downscaling, basing the SNR calculation on the resulting values.

Similar cases also occur when alternating detector settings such as gain are used.

17. References

- The FITS Standard, version 3.0
- Paper I: Representations of World Coordinates in FITS (Greisen & Calabretta)
- Paper II: Representations of celestial coordinates in FITS (Calabretta & Greisen)
- Paper III: Representations of spectral coordinates in FITS (Greisen, Calabretta, Valdes & Allen)
  - Authors' web sites, supplemental background: Eric Greisen, Mark Calabretta, http://www.atnf.csiro.au/people/mcalabre/WCS/index.html
  - An unofficial errata for Papers I, II, and II (Calabretta & Greisen)
- Paper IV: Representations of Time Coordinates in FITS (Rots), under development and review.
- Paper V: Representations of distortions in FITS WCS Calabretta, Valdes, Greisen, Allen
- Paper VI: Coordinate systems for solar image data (Thompson)
- Paper VII: Precision effects for solar image coordinates within the FITS world coordinate system (Thompson).
- The SolarSoft WCS Routines: A Tutorial (Thompson)
- The "Green Bank convention"
- The CHECKSUM/DATASUM convention
- Space Physics Archive Search and Extract (SPASE) instrument types
- Recommendations for Data & Software Citation in Solar Physics (2012AAS...22020127H)
- Best Practices for FITS Headers (2012AAS...22020128H)
- http://virtualsolar.org/checklists

[21] Scaled-up values that end up *below* the saturation limit should be set to the value of the saturation limit.

- http://docs.virtualsolar.org/wiki/MinimumInformation
- http://fits.gsfc.nasa.gov/registry/checksum.html