



Publication Year	2024
Acceptance in OA	2024-03-19T16:24:00Z
Title	SDHDF: A new file format for spectral-domain radio astronomy data
Authors	Toomey, L. J., Hobbs, G., Price, D. C., Dawson, J. R., Wenger, T., Lagoy, D., Staveley-Smith, L., Green, J. A., CARRETTI, ETTORE, Hafner, A., Huynh, M., Kaczmarek, J., Mader, S., McIntyre, V., Reynolds, J., Robishaw, T., Sarkissian, J., Thompson, A., Tremblay, C., Zic, A.
Publisher's version (DOI)	10.1016/j.ascom.2024.100804
Handle	http://hdl.handle.net/20.500.12386/34990
Journal	ASTRONOMY AND COMPUTING
Volume	47

SDHDF: A new file format for spectral-domain radio astronomy data

Toomey, L. J.^a, Hobbs, G.^a, Price, D. C.^b, Dawson, J. R.^{a,c}, Wenger, T.^{h,l}, Lagoy, D.^h,
 Staveley-Smith, L.^{d,e}, Green, J. A.^{f,j}, Carretti, E.^g, Hafner, A.^a, Huynh, M.^j, Kaczmarek, J.^{a,i},
 Mader, S.ⁱ, McIntyre, V.^a, Reynolds, J.^a, Robishaw, T.^h, Sarkissian, J.ⁱ, Thompson, A.^j,
 Tremblay, C.^{i,j,k}, Zic, A.^a

^a*CSIRO Space & Astronomy, P.O. Box 76, Epping, 1710, NSW, Australia*

^b*International Centre for Radio Astronomy Research, Curtin University, Bentley, 6102, WA, Australia*

^c*School of Mathematical and Physical Sciences and MQ Research Centre in Astronomy, Astrophysics, and
 Astrotechnology, Macquarie University, 2109, NSW, Australia*

^d*International Centre for Radio Astronomy Research (ICRAR), University of Western
 Australia, 6009, WA, Australia*

^e*ARC Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D), Australia*

^f*SKA Observatory, SKA-LOW Science Operations Centre, ARRC Building, 26 Dick Perry Avenue, Technology
 Park, Kensington, 6151, WA, Australia*

^g*INAF, Istituto di Radioastronomia, Via Gobetti 101, Bologna, 40129, Italy*

^h*Dominion Radio Astrophysical Observatory, Herzberg Astronomy and Astrophysics Research Centre, National
 Research Council Canada, PO Box 248, Penticton, BC V2A 6J9, Canada*

ⁱ*CSIRO Space & Astronomy, Parkes Observatory, Parkes, 2870, NSW, Australia*

^j*CSIRO Space & Astronomy, PO Box 1130, Bentley, 6102, WA, Australia*

^k*SETI Institute, Mountain View, CA 94043, USA*

^l*NSF Astronomy & Astrophysics Postdoctoral Fellow, Department of Astronomy, University of
 Wisconsin, Madison, WI 53706, USA*

Abstract

Radio astronomy file formats are now required to store wide frequency bandwidths and multiple simultaneous receiver beams and must be able to account for versatile observing modes and numerous calibration strategies. The need to capture and archive high-time and high frequency-resolution data, along with the comprehensive metadata that fully describe the data, implies that a new data format and new processing software are required. This requirement is suited to a well-defined, hierarchically-structured and flexible file format. In this paper we present the Spectral-Domain Hierarchical Data Format (‘SDHDF’) — a new file format for radio astronomy data, in particular for single dish or beam-formed data streams. Since 2018, SDHDF has been the primary format for data products from the spectral-line and continuum observing modes at Murrumbidgee, the CSIRO Parkes 64-m radio telescope, and we demonstrate that this data format can also be used to store observations of pulsars and fast radio bursts.

Keywords: Radio astronomy Astronomy data acquisition

1. INTRODUCTION

With increasing instantaneous bandwidth and higher data volumes output from modern receivers on radio telescopes, such as the Ultra-Wide-Bandwidth Low Frequency (UWL, [1]) receiver, and the

cryogenically-cooled phased-array feed (‘Cryo-PAF’) receiver at Murriyang¹ the CSIRO Parkes 64-metre radio telescope, there was a requirement for a new file format. This requirement is not unique to Parkes; there are many other observatories with instruments that produce multiple beams and wide bandwidths (such as the Five-hundred-meter Aperture Spherical Telescope ‘FAST’ in China, which utilises similar receiver technology, and has historically written data out with similar formats to Parkes). Here we present Spectral-Domain Hierarchical Data Format (‘SDHDF’) – a new file format for radio astronomy data based on the Hierarchical Data Format² (‘HDF³’).

Previously, spectral data from Murriyang were stored in ‘SDFITS’ format ([2]). This format for ‘Single Dish’ data uses the binary table data structure as specified in the Flexible Image Transport System (‘FITS’, [3], [4]). While the FITS format has enjoyed widespread usage over several decades, design decisions that were reasonable in 1980, now limit modern astronomical research; a summary of these issues is given by [5, 6], and we discuss these later on.

The use of HDF as a file format for Parkes spectral-line and continuum data was first introduced specifically for the HI-Pulsar ‘HIPSR’ digital backend ([7]) of the 21-cm multi-beam receiver, as an alternative to SDFITS. However, the HIPSR format was not intended for use outside the project, and does not implement features required for broader use cases. Amongst the wider astronomy community, HDF has been implemented at the Low Frequency Array (LOFAR, [8]) for a decade or so ([9]), albeit more recently for beam-formed data products⁴. The Laser Interferometer Gravitational-Wave Observatory (LIGO) Scientific Collaboration publish their data in HDF (for example, the O2 Data Release⁵), and the format is also supported for image cube visualisation of large-scale data from several telescopes such as the Karl G. Jansky Very Large Array (VLA), the Atacama Large Millimeter/sub-millimeter Array (ALMA), and SKA pathfinders, with the Cube Analysis and Rendering Tool for Astronomy (CARTA⁶) ([10]).

Consequently we explored the use of HDF for all spectral-line and continuum data products from Parkes receivers, converging on the SDHDF model as the formal convention based on the following considerations:

- The storage mechanism is based on HDF, a well-defined and documented format. The HDF Group states that they are “committed to always keeping HDF software free and open source”, and this is reflected in the extensive and up-to-date commit history of the repository.
- SDHDF is capable of storing comprehensive high time-resolution and wide-band high frequency-resolution data in a nested hierarchical structure suitable for telescope feeds with one beam (single pixel feed) or multiple beams, and with the potential to expand to support beam-formed data products from multiple single dishes, antennas or tiles.
- SDHDF data can be incorporated into HDF Virtual Datasets (VDS⁷), and the format is suitable for working with terabyte-scale data products in distributed computing infrastructure.
- The data and metadata are fully self-described by HDF ‘attributes’.

¹In the Wiradjuri Dreaming, Biyaami (Baiaame) is a prominent creator spirit and is represented in the sky by the stars which also portray the Orion constellation. Murriyang represents the ‘Skyworld’ where Biyaami lives.

²<https://www.hdfgroup.org/>

³Note: At the time of writing, any reference to HDF in this paper is to the data model of HDF version 5 (HDF5)

⁴<https://support.astron.nl/LOFARBeamformedCookbook/index.html>

⁵<https://doi.org/10.7935/CA75-FM95>

⁶<https://doi.org/10.5281/zenodo.3377984>

⁷<https://docs.h5py.org/en/stable/vds.html>

- The format is designed to be portable and open to new functionality and capability, with long-term development support funded by donation from within the community.
- Reduction of the data products can produce secondary data products that themselves meet the SDHDF definition.
- The format can be parsed by HDF libraries available for computer languages such as C/C++, MATLAB, IDL, Julia, Python, as well as by open source tools and graphical user interfaces (GUIs).
- SDHDF data products are suitable for storage in long-term archives because the metadata are described, the files are compressible, and the SDHDF definition has been allocated a Digital Object Identifier⁸ (DOI) for the life-cycle of the data.

SDHDF was successfully commissioned as the primary format for all spectral-line and continuum data products on completion of the Parkes UWL receiver project. High time-resolution pulsar observations are still produced in ‘PSRFITS’ format ([11]), but we demonstrate below that including pulsar data in SDHDF format would be possible in the future. Since late 2018, more than 80 TB of SDHDF data products have been archived in the Australia Telescope Online Archive⁹ (ATOA), becoming publicly available after an initial 18 month embargo period.

In this paper, we provide an overview of the SDHDF definition in Section 2, and present design features in Section 3. In Section 4 we provide information on working with SDHDF files, including the introduction of a new software package ‘INtegrated SDHDF Processing Engine in C for Telescope data Analysis’ (INSPECTA), developed specifically for working with this format and for which we provide some example use cases. In Section 5 we discuss the choice of HDF as the base for our format, further SDHDF development, and uptake within the community. We present our conclusion in Section 6. An overview of the definition (Appendix A), and an example of how the format stores some observation parameters for Parkes data (Appendix B), are available in the online supplementary material.

2. THE SDHDF DEFINITION

The SDHDF definition can be thought of as a tree-like structure of HDF ‘group’ and ‘dataset’ binary objects, containing the data, observation metadata, and time, frequency and polarisation information. A visual representation of the structure is shown in Figure 1.

SDHDF was designed for flexibility and to accommodate multi-dimensional channelised data in one or many frequency bands of any bandwidth in blocks of time of sub-second precision. A single file can contain one or many observations from different observing modes (for example a targeted pointing and/or scan) stored as multiple top-level ‘root_ID’ objects, each of which adhere to the SDHDF definition and can be a combination of many input data streams from one or many beams, associated metadata for both astronomy and noise source data products, and observation metadata. An SDHDF file may also be a product of post-processing from any number of other SDHDF data and metadata products structured according to the definition – these may contain more structure than the raw data. For example, we may optionally store datasets that have had mean or median

⁸<https://www.doi.org>

⁹<https://atoa.atnf.csiro.au/>

averaging applied, or store additional zoomed frequency bands. The SDHDF definition¹⁰ and a template data file are publicly available on CSIRO’s Data Access Portal¹¹.

In order to ensure that all SDHDF data products from a set of telescope observations meet the requirements of the definition, we have implemented a template methodology. With this, on completion of an observation, the data and metadata structure, descriptions, data units and values are first populated in an SDHDF template file built from the definition, where parameters are configured for a specific telescope and observing system. The HDF datasets for the astronomy data are initiated such that they are resizable. The template structure at this stage is just a placeholder – it is then populated by the raw data and metadata to form the final data product. Metadata are stored with HDF ‘attributes’ – objects that directly describe the dataset or group object to which they are attributed. All attributes are stored as native Python data types.

The benefits of a flexible and configurable template structure are two-fold — it facilitates uptake of the format by a range of radio astronomy institutions, and use of a template ensures that all data products from different institutions adheres to the SDHDF format definition, thereby enhancing data provenance in long-term archives.

2.1. SDHDF classes and attributes

The SDHDF definition contains HDF classes and attributes that are unique to an SDHDF format file – these are useful for grouping objects together when performing operations on the data. Classes are assigned to specific dataset types, identified by the ‘SDHDF_CLASS’ attribute. Table 1 lists some examples of these classes and their descriptions.

Two classes, `sdhdf_table` and `sdhdf_waterfall` are central to the SDHDF definition. We provide a brief summary below; further details about these and other SDHDF classes can be found in the SDHDF definition. Observation metadata are stored in tables with the SDHDF_CLASS `sdhdf_table`. This class is similar to the HDF Table Specification Version 1.0¹². A one-dimensional compound HDF dataset is used to store the data, with column names ascribed in the compound datatype.

The `sdhdf_waterfall` class is an N -dimensional array with dimension scales to describe each axis. Following the HDF dimension scale specification, the attribute ‘DIMENSION_LABELS’ provides a list of strings to describe each axis. Similarly, the attribute ‘DIMENSION_LIST’ provides references to any datasets that represent a dimension scale. While the `sdhdf_waterfall` class can be used for any N -dimensional dataset, in this context the dimensions are labelled as time, product type (default: polarisation), frequency and bin¹³, as defined in Section §3.1. Information about units (e.g. jansky, kelvin, counts) is stored in the ‘UNIT’ attribute.

Arrays in the `sdhdf_waterfall` datasets can be large and HDF provides the means to store multiple segments separately in the file (using a method known as ‘chunking’). The SDHDF definition does not preclude chunking any dataset in a way that will make reading and writing the data more efficient. For the Parkes observations we currently use an automatic chunking algorithm, but are currently exploring more efficient methods. We note that as long as HDF tools are used then the choice of chunking method does not affect how the data sets are loaded or processed.

¹⁰<https://doi.org/10.25919/880t-0m95>

¹¹<https://data.csiro.au/>

¹²https://docs.hdfgroup.org/hdf5/develop/_t_b_1.html

¹³In the `sdhdf_waterfall` class, the term ‘bin’ refers to the phase of a periodic signal, such as that of a pulsar or switched noise source

Table 1: Examples of SDHDF classes

Class name	Description
<code>sdhdf_file</code>	SDHDF file class
<code>sdhdf_band</code>	SDHDF frequency band class
<code>sdhdf_table</code>	SDHDF metadata table class
<code>sdhdf_frequency</code>	SDHDF frequency dataset
<code>sdhdf_data</code>	SDHDF astronomy dataset
<code>sdhdf_waterfall</code>	SDHDF frequency time dataset
<code>sdhdf_flags</code>	SDHDF RFI flags dataset
<code>sdhdf_weights</code>	SDHDF RFI weights dataset

Throughout the definition, other attributes are used to define special parameters — for example, the ‘FRAME’ attribute describes the spectral reference frame of the frequency dataset, set to ‘topocentric’ by default in the definition.

In Table 2 we show an example of the metadata including attributes, associated with a dataset that contains information about a telescope beam. The position of the dataset in the object hierarchy is shown in ‘Path Information’, with the data being stored as a compound entity of two different data types (‘Data Information’). Attributes of the example dataset include ‘NUMBER_OF_BANDS’ and ‘SOURCE’ entries – the definition allows for any number of beams, each with potentially different sources and associated metadata for any number of frequency bands (sub- or zoom-bands).

3. SDHDF DESIGN FEATURES

3.1. Flexible observing modes

The SDHDF structure is flexible enough to accommodate various observing modes. The telescope may scan across a sky region, the receiver may rotate to account for parallactic angle, or (with receivers such as phased-array feeds) individual beams may be steered independently. The top (or ‘root_ID’ level) in the SDHDF definition stores a description of each receiver beam (0 to $N-1$ beams, where ‘beam_0’ is the first beam¹⁴).

The tier down from the beam group is the ‘band’ group object. In this context a ‘band’ refers to a single range of frequencies, and may be colloquially referred to as a sub- or zoom-band. Note that we do not explicitly preclude overlapping bands; for instance a file could include a wide bandwidth and a separate band that is a sub-set of the full band. The information stored in the band group will have originally come from a signal processor, which will have formed spectra at specific time intervals.

¹⁴In SDHDF, all indices are zero-based.

Table 2: Example metadata from an SDHDF dataset object

Path Information			
Basename	beam_parameters		
Local	sdhdf_template.hdf::/metadata/beam_parameters		
Physical	sdhdf_template.hdf::/metadata/beam_parameters		
Data Information			
HDF type	COMPOUND		
Data types	LABEL (string) NUMBER_OF_BANDS (int64) SOURCE (string) RIGHT_ASCENSION (string) DECLINATION (string)		
Default Attributes			
Name	Description	Unit	Value
SDHDF_CLASS	SDHDF class name	None	sdhdf_table
SDHDF_DESCRIPTION	SDHDF object description	None	Metadata specific to the antenna beam
Dimension Attributes			
Name	Description	Unit	Value
LABEL	Beam label	None	beam_N
NUMBER_OF_BANDS	Number of frequency bands	None	band_N
SOURCE	Source name	None	source_name
RIGHT_ASCENSION	Coordinates in Right Ascension	HH:MM:SS.s	HH:MM:SS.s
DECLINATION	Coordinates in Declination	DD:MM:SS.s	DD:MM:SS.s
Additional Attributes			
Name	Description	Unit	Value
EQUINOX	Equinox of the coordinates	Julian year	NNNN

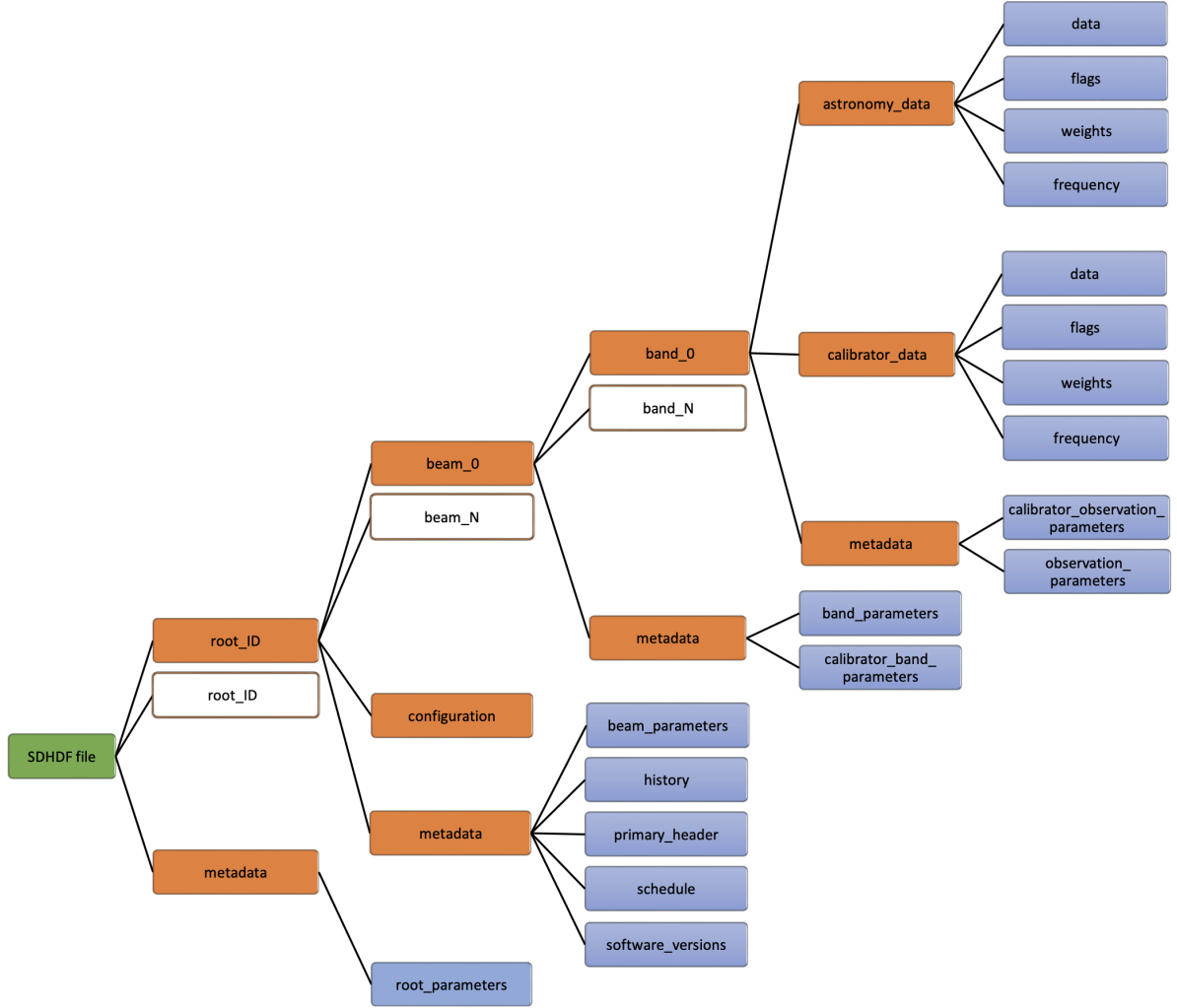


Figure 1: A visual representation of an example data file, showing the hierarchical association of HDF groups and datasets. An SDHDF data file (the solid green rectangle) may contain optionally none or any number of ‘root_ID’ objects each of which is of the SDHDF_CLASS `sdhdf_file` and each adhering to the SDHDF definition, where each root_ID object could be a separate observation and grouped together (for example one may wish to store multiple single scans in the same file). Information about the root_ID objects are stored in a ‘root_parameters’ dataset; this dataset and the parent metadata group are not required if the file does not contain a root_ID object. In this Figure, the solid orange and solid blue rectangles represent HDF group and dataset objects respectively. Any number of root_ID, beam and band groups per beam are supported, as illustrated by the unfilled rectangles with orange borders. For Parkes UWL data, we chose to record a switching noise source for calibration purposes, storing both on, off and a 32-bin time-averaged dataset in the ‘calibrator_data’ group (not shown in this Figure).

We store time-dependent information (such as the timestamp of each spectrum, pointing information, weather conditions, etc.) independently for each telescope beam and observing band in the ‘observation_parameters’ dataset. For Parkes data we store this information with a default cadence corresponding to the rate that the time-averaged spectra are recorded, but the implementation is flexible enough to store information on any cadence. The ‘observation_parameters’ dataset is configurable based on the requirements of a particular telescope – an example of the parameters required for a Parkes observation is shown in Appendix B. Metadata information, such as the time corresponding to a specific spectrum, or the frequency of a given frequency channel, is set to the centre of the time interval or the frequency channel. The primary time information is stored as an integer Modified Julian Date (MJD) and the number of seconds (as a float) since the start of that MJD. Storing these as separate values allows the format to have sufficient time precision for pulsar data.

The spectra are stored for each beam and band in the ‘astronomy_data’ groups, as a 4-dimensional `sdhdf_waterfall` dataset with floating point precision. For the majority of continuum and spectral-line observations only a single phase bin is stored, however, for pulsar or calibration observations a large number of phase bins may be recorded. The frequency of each channel is stored in the frequency dataset. This is a two dimensional array with the dimensions being the number of integrations and the number of frequency channels. In many cases only a single integration is present (as the frequency of the channels do not change in time). However, if the frequency axis has been Doppler corrected to account for the Earth’s motion then the number of integrations likely will equal the number of spectra in the file.

The spectra recorded in the SDHDF file usually need to be calibrated. The calibration steps depend on the observing system and the particular science case. The SDHDF structure therefore needs to be sufficiently flexible to allow for different calibration methods. SDHDF was initially developed for the spectral-line and continuum output data from the Parkes telescope, in which the astronomy spectrum is recorded with high time and frequency resolution, and spectra with the noise source switched on and separately switched off are also recorded with lower time and frequency resolution. Such information is stored in datasets for each beam and band in the ‘calibrator_data’ group with `sdhdf_waterfall` classes being used to store the on and off spectra for Parkes data. In order to confirm that the signal processor was correctly synchronised with the noise source we also used the `sdhdf_waterfall` class with multiple phase bins to record spectra as measured in 32 phase bins across the noise source period. Details of the calibration data can also be stored as HDF dataset attributes. For instance, the switched noise source data may be normalised so that the counts in the noise source spectra are equivalent to counts in the astronomy spectrum.

Traditional spectral-line calibration strategies rely on switching between observation parameters. For instance, the telescope position may switch between an on-source and an off-source position (or swapping beam positions), or the signal-processing system may implement a frequency-switching process. The current Parkes signal processor produces separate SDHDF data files for different source positions (i.e., on- and off-source pointings are in different files), however as shown in Figure 1, an SDHDF file can have multiple ‘root_ID’ groups. These can be used to store multiple observations (or multiple observing setups) in a single SDHDF file. Frequency-switched observations typically shift the frequency of a local oscillator in order to facilitate spectral bandpass calibration. Within SDHDF, such observations could be stored as (1) individual SDHDF files for each change in frequency, (2) as separate ‘root_ID’ groups for each frequency, (3) by adding in an extra ‘scan’ hierarchy after the ‘band’ group, (4) by expanding the ‘observation_parameters’ metadata to include the local oscillator frequency or (5) with the inclusion of observatory-specific

calibration datasets to be included in the ‘calibrator_data’ group. See Section 5.1 on extending the format.

3.2. Additional datasets, flags and weights

The SDHDF definition includes scope for extra, optional datasets. For instance, these could be required as part of the data reduction process. Perhaps the most commonly used extra datasets relate to weighting and flagging the spectra. The output of manual, or automatic, flagging routines can be stored in the ‘flags’ datasets. This is recorded with a 0 to indicate a spectral channel is not flagged and a 1 otherwise. Unless specified explicitly we use unsigned 8-bit values to record these flags to enable simple reading and writing, but they can be written as 1-bit values if required. We allow for flags to be applied to astronomy or calibration spectral channels or integrations. The PSRFITS definition does not allow flags to be applied independently for polarisation channels. However, we provide the option that flags (and weights, as described below) can be written with the same dimensions as the raw data (including polarisation and bin information). However if, for instance, the polarisation dimension is set to 1 then the flags would be applied to all polarisations for a given time and frequency channel.

Weights can also be applied (usually to each frequency channel in each integration). The weight can represent the product of the channel bandwidth and integration time (or other weighting methods could be used such as [12]). SDHDF weights are stored as floating point values in the ‘weights’ datasets.

If the data are recorded with low-bit quantisation (for example, pulsar search-mode data is generally a 2-bit data stream), then the data type is stored in the ‘DATA_TYPE’ attribute of the `sdhdf.waterfall` class. It is often necessary to provide tables of scaling factors enabling the conversion of the stored bit depth back to the measured signal strength. Following the PSRFITS definition we allow for independent `sdhdf.waterfall` datasets containing offsets and scaling parameters where the measured value v is recovered from the stored unsigned integer data value d as:

$$v = (d - z)S + \zeta \quad (1)$$

where z is a ‘zero-offset’ value, S a frequency, polarisation and time-dependent scaling factor and ζ a frequency, polarisation and time-dependent offset.

4. WORKING WITH SDHDF DATA

Open source GUIs and Python modules are available for interrogating HDF files in general, and the Parkes spectral-line and continuum teams have also been independently developing their own SDHDF processing tools in Python. We have developed the comprehensive INSPECTA software package, included as part of the publicly available SDHDF_TOOLS¹⁵ repository. INSPECTA is written in C and contains tools to perform a multitude of processing tasks, from visual inspection of files to calibration and even the simulation of various observing parameters. The complete list of tools is too comprehensive to list here, but allows for spectra to be visualised, calibrated and analysed. It also allows for both automatic and manual radio-frequency interference (RFI) flagging. We provide examples below in which SDHDF files have been processed using the INSPECTA software

¹⁵https://bitbucket.csiro.au/scm/cpda/sdhdf_tools

to demonstrate the suitability of SDHDF for different types of observations. These observations have been carried out with Murriyang and the majority make use of the UWL receiver.

Here we show both the versatility of the SDHDF format and the INSPECTA software package by demonstrating how a diverse set of observations can be processed and visualised. An example is shown in the left panel of Figure 2 in which the telescope was tracking the limb of the Moon as it occulted quasar 3C 273 on 20th February 2022 (as a modern re-enactment of [13]). The SDHDF file for this observation was obtained from the ATOA¹⁶. We recorded observations with 400 Hz resolution producing over 7 million spectral channels over the entire UWL band, which were written to disk each second. In order to ensure we do not produce a single file that is too large to be archived, our online processing scripts split data files when they reach a specified size (currently ~ 10 GB) – the files can be split by beam, band or time, and the INSPECTA software can subsequently be used to join files back together as required. The total observation time for this specific data file was ~ 1160 seconds. The SDHDF file captures each of the 26 sub-bands formed by the UWL receiver system along with metadata recording the telescope pointing directions for each of the 1 s integrations. To form the output shown in Figure 3 we used the INSPECTA tools to average the same observation in time. This is then visualised as the three RF bands for both polarisation channels. Note that no calibration has been carried out in this stage and hence the Figure shows the frequency-dependent bandpass shape and strong RFI signals primarily from mobile communication systems.

This observation can also be used to demonstrate continuum observations in which a low frequency resolution is generally required, but the data are output with short time duration, showing that the SDHDF format can store pointing information at sufficient precision and temporal resolution. We first carried out an automatic removal of both persistent and transient RFI using the INSPECTA software. We then averaged the frequency channels, summed the polarisations and plotted the total signal strength as a function of time in the 128 MHz-wide sub-band centred at 1408 MHz. Of course, we can make similar plots in other sub-bands, and a full analysis of the recorded signal will be presented elsewhere.

The occultation observation was carried out by tracking the position of the source; it is more common to scan the telescope beam across a known bright source. In the right panel of Figure 2 we show a scan across the flux density calibrator source PKS 1934–638. Here we have written three frequency bands to the SDHDF file (centred on 1408, 1664 and 3328 MHz) and the traces have been normalised and offset from each other to show the beam widths. This demonstrates that the stored metadata provide sufficient time and positional information to recover the source position. The SDHDF file for this observation was obtained from the ATOA¹⁷.

To demonstrate that SDHDF records sufficient information to calibrate the flux density and polarisation properties of a spectral-line source we use a known, bright hydroxyl maser associated with high-mass star formation, G330.953–0.182 ([14], references therein). The SDHDF file was also obtained from the ATOA¹⁸. For this example we do not use an off-source pointing (although the archive does contain off-source pointings obtained for this observing project). During the observation the noise source was switched on and off at a frequency of 100 Hz. The SDHDF file contains the astronomical spectrum as originally recorded, along with the spectra at lower frequency resolution when the noise source is on and also when it is off, separated out using the known phase

¹⁶<https://atoa.atnf.csiro.au/>; Project ID: PX083; File: uwl.220219_143803_3.hdf

¹⁷<https://atoa.atnf.csiro.au/>; Project ID: P974; File: uwl.210914.053215.hdf

¹⁸<https://atoa.atnf.csiro.au/>; Project ID: P1073; File: uwl.220323_193627.hdf

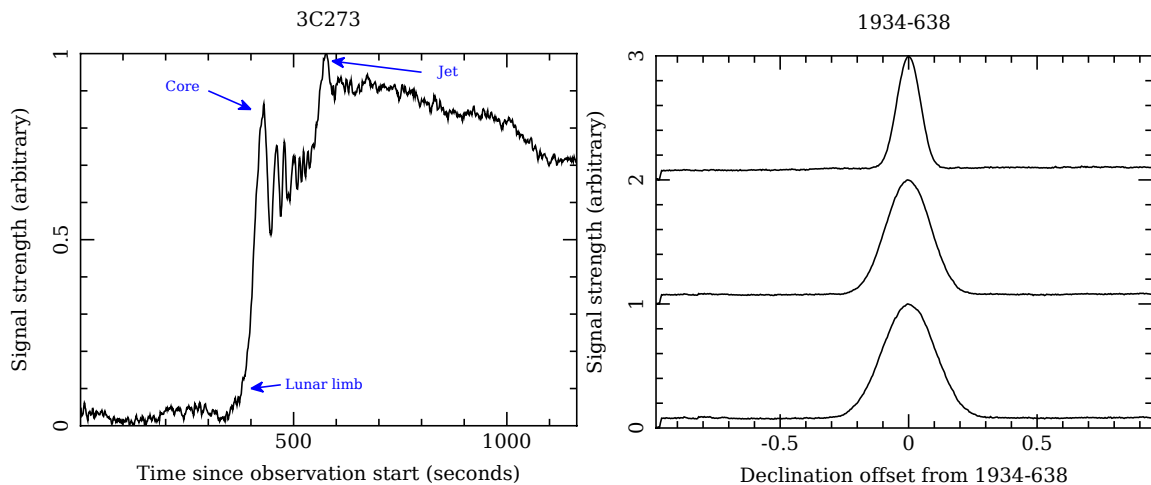


Figure 2: (Left panel) The signal strength in the UWL 1152 MHz band as a function of time during the lunar occultation of the quasar 3C 273. The quasar core and then a jet boost the flux as they emerge from behind the lunar limb. (Right panel) Representation of a scan across a primary flux calibrator source PKS 1934–638 in three observing bands centred on 3328 (top), 1664 (centre) and 1408 MHz (bottom), where the declination offset is in degrees.

of the noise source. We note that the integration time of the astronomy data can be different from that of the noise source. We use INSPECTA to average the data in time and to extract the part of the observing band that contains the hydroxyl lines of interest (the 1665.402 and 1667.359 MHz main-line ground-state transitions). We then use the switching noise source to calibrate the astronomy spectrum (both flux density and polarisation). This makes use of knowledge of the noise source amplitude measured through separate on- and off-source observations of PKS 1934–638. These flux calibration solutions for Murriyang are available online¹⁹.

The left panel in Figure 4 shows the flux and polarisation calibrated spectrum, using the time information stored in the SDHDF file to convert the observed topocentric frequencies to the local-standard-of-rest velocity. A detailed monitoring study of the variability of this source is being carried out using similar analysis methods and will be published elsewhere.

SDHDF has been designed to allow for multiple antenna beams. However, our test case with the Parkes UWL system only provides a single beam. In order to demonstrate that archival multi-beam data can be stored in SDHDF format, and that the format is adequate for the future (such as for phased-array feeds) we have used INSPECTA to simulate data of Galactic hydrogen emission and absorption in the vicinity of the Galactic Centre based on input spectra from the Parkes Galactic All-Sky Survey (GASS; [15]) – refer to the right panel of Figure 4. We have also demonstrated that the SDHDF format and the INSPECTA software can be used for storing many more beams (we have successfully simulated, stored and processed 72 beams in a single SDHDF file, and the scaling is linear).

¹⁹https://www.parkes.atnf.csiro.au/observing/Calibration_and_Data_Processing_Files.html

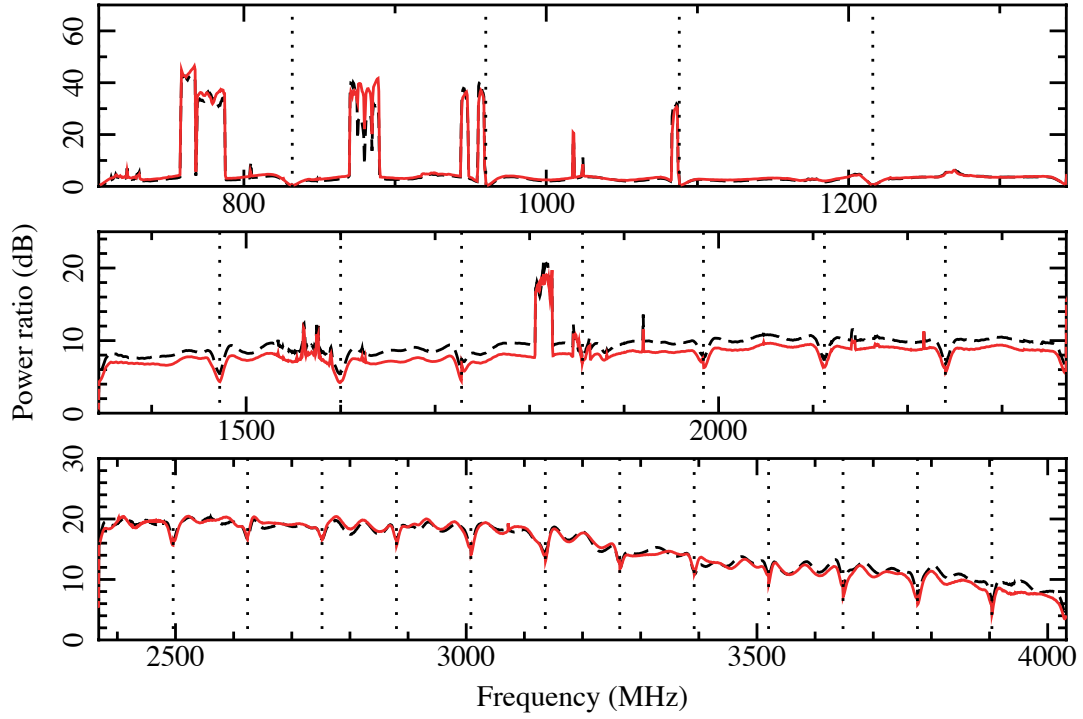


Figure 3: A wide-bandwidth observation demonstrating the ability of SDHDF to store, process and visualise such data. The three sub-panels represent the three RF bands recorded by the Parkes UWL receiver system (704 to 1344 MHz, 1344 to 2468 MHz and 2468 to 4032 MHz), showing polarisations A and B in red and black (dashed) respectively.

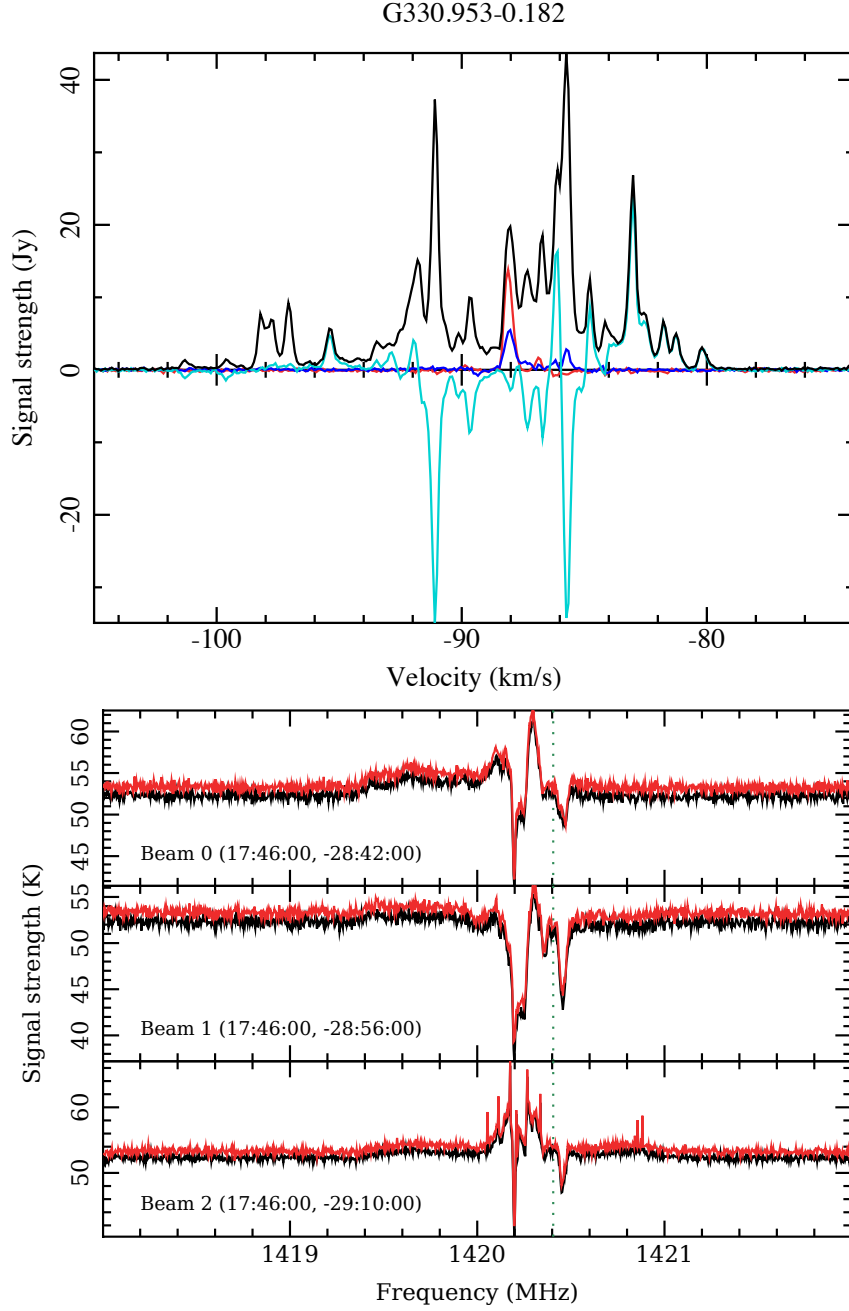


Figure 4: (Top panel) Fully calibrated spectrum of the known 1665-MHz hydroxyl maser G330.953–0.182 from a Parkes UWL observation, showing the Stokes parameters I, Q, U and V in black, red, blue and teal respectively. (Bottom panel) Simulated example multiple-beam observations of Galactic hydrogen making use of Parkes Galactic All-Sky Survey (GASS) observations of adjacent positions separated by 14 arc-minutes (the beam full width half maximum) in declination, showing polarisations A and B in red and black respectively. Note that all beams are stored within the same SDHDF file, which can account for both fixed and moving beam directions.

4.1. Storing phase-resolved, high time-resolution data and data precision

SDHDF was originally developed for the spectral-line and continuum radio astronomy communities. However, SDHDF can store phase-resolved data and therefore can also be used for pulsar-style observations. Additionally SDHDF has the potential to store multiple modes of observations, such as those from commensal surveys.

For nearly two decades, PSRFITS has been the standard data format for phase-resolved and time-series data such as those from pulsar fold- and search-mode observations respectively. The format is well supported by pulsar processing software packages that are trusted by the community. However, we believe that pulsar data would also benefit from being stored in SDHDF for the following reasons (noting that this is not an exhaustive list):

- PSRFITS format is not hierarchically structured, and so has no concept of the beam dimension and therefore cannot support multi-beam data.
- I/O is significantly affected if a PSRFITS binary data unit is appended to, unlike the chunked storage in HDF which allows the file to be easily extended.
- For pulsar search-mode observations, there is generally a trade-off between splitting the time-series into small chunks each with its own metadata header (as in the PSRFITS format) and having a single header and an unbroken time series (as in SIGPROC's filterbank²⁰ format). The SDHDF format allows for an unbroken time-series, whilst storing the metadata at the required cadence.
- Pulsar fold-mode observations generally start and/or end with separate calibration observations of the injected noise source, that are currently stored in separate files — these could all be incorporated into the same SDHDF file for simplicity and ease of use.
- SDHDF is able to store auxiliary data products such as images of integrated pulse profiles or gain variations which would add value to the data once archived.
- The spectral-line and pulsar software packages are independent and yet apply similar methods, for example, for calibration. A single set of algorithms can be used to ensure consistency between different observing modes.

Pulsar fold-mode data are produced by folding the incoming signal at the known period of the pulsar and storing the pulse profile for each frequency and polarisation channel with a specified number of phase bins. Such data can be stored within the SDHDF format without modification as the data structure allows for binned data. An example pulsar profile is shown in Figure 5. The pulsar, PSR J1717–4054 was observed as part of the PULSE@Parkes outreach project ([16]) — the PSRFITS file can be obtained from the DAP²¹. The file was converted into SDHDF using the INSPECTA tools and then visualised. The upper panel shows the time, polarisation and frequency averaged de-dispersed pulse profile, the central panel shows the pulse amplitude as a function of time (noting that the pulsar switches off and on during the observation) and the bottom panel shows the dispersed pulse sweep as a function of observing frequency.

²⁰<https://sigproc.sourceforge.net>

²¹<https://data.csiro.au/>; Project ID: P595; File: s101109.000655.rf

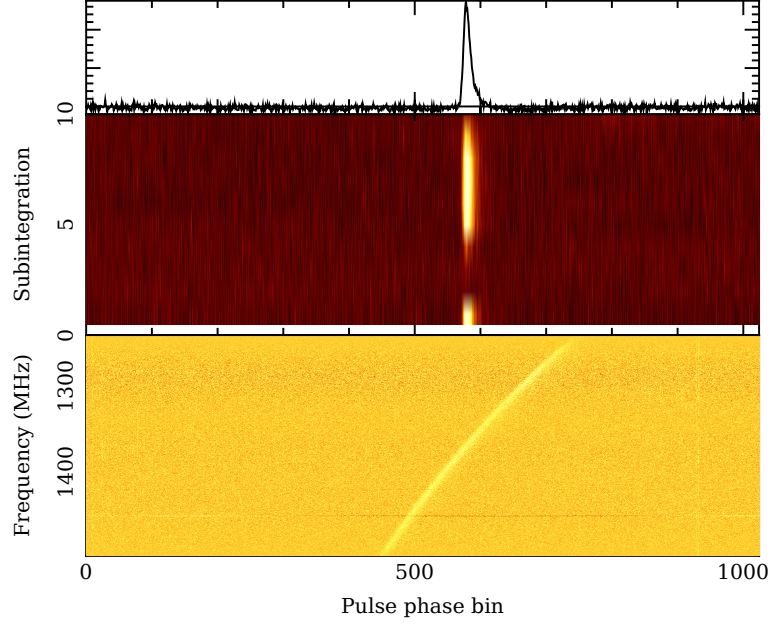


Figure 5: Visualisation of data requiring phase bins. This observation is of the intermittent pulsar PSR J1717–4054 and the data stream has been channelised and folded online at the known period of the pulsar. The top panel shows the de-dispersed and averaged pulse profile. The central panel represents the de-dispersed pulse signal strength as a function of sub-integration, and the bottom panel shows the dispersed pulse signal strength as a function of phase and frequency.

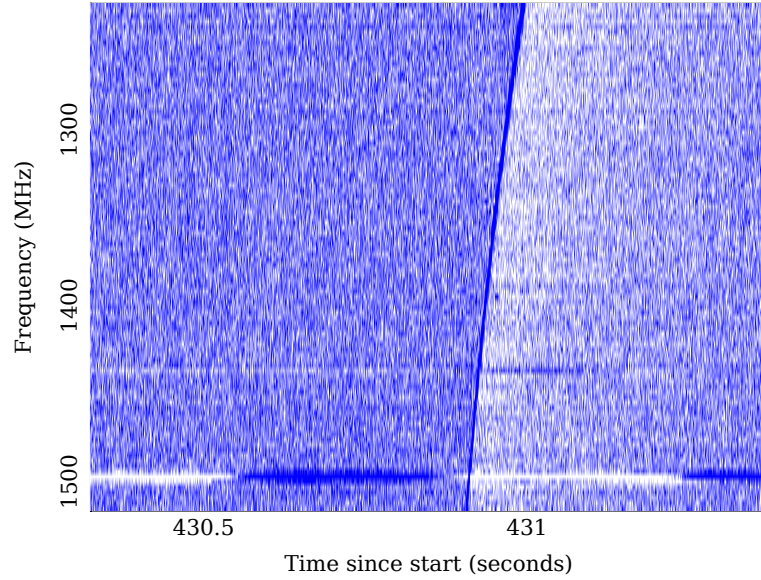


Figure 6: High time-resolution, multi-beam data converted into SDHDF format. We show 1 second of data around the first discovered fast radio burst (FRB) in beam 6 of the Parkes multi-beam receiver ([17]).

High time-resolution time-series data (used, for instance, to search for pulsars or fast radio bursts) cannot be stored with floating point precision. Typically such data streams are recorded with 1, 2, 4 or 8-bits of precision. In Figure 6 we show the first discovered fast radio burst (FRB; [17]) after converting the 13 PSRFITS data files corresponding to the 13 beams of the multi-beam receiver into a single SDHDF file. We note that the data volume in SDHDF is slightly smaller than the original PSRFITS files as we do not require the extra metadata information every 4096 samples.

5. DISCUSSION

A commonly-held criticism and limitation of HDF is the lack of accessibility and readability — HDF is an entirely binary data format and not human-readable. Astronomers have historically enjoyed the benefits of partially text-based formats like FITS. For instance, the metadata in those files can be inspected in a standard text editor. However, we believe that the ability to manually edit a data file goes against good provenance practices, and in the Big Data era it is intractable to open an entire file in a simple text editor. We therefore believe that HDF is a suitable format for storing radio astronomy data for the following reasons:

- **Overcoming the limitations of FITS** — Existing FITS derivatives have limitations, for example RPFITS²² did not completely adhere to the FITS standard and lacked comprehensive metadata, SDFITS suffered from structure variation across files from different observations, and was not suitable for storing or processing of high-volume wide-frequency band data. PSRFITS, although the primary data format for pulsar astronomy data, is not flexible enough for the storage of data and metadata from spectral-line or continuum observing modes. Unlike HDF, the FITS metadata also suffer from keyword truncation, the Data Units (‘extensions’) do not easily accommodate appending of large data, and there is no support for hierarchical grouping of data and metadata required by multi-beam observations.
- **No suitable alternative formats** — We found no other file formats that could meet our requirements. We investigated Advanced Scientific Data Format (ASDF; [18]), a format designed to find a middle ground between FITS and hierarchically structured content, however whilst overcoming some of the limitations of FITS, there were restrictions on data and metadata storage and it was deemed too generic for storing high-volume radio astronomy data from wide-bandwidths and multiple beams.
- **Improved I/O performance and compression** — The HDF libraries support ‘chunked’ over ‘contiguous’ storage to improve I/O performance. SDHDF files can be cloud-optimised for distributed computing by writing out with increased chunk sizes. With regards to compression, historically metadata were stored sparsely, as storing multi-dimensional arrays of information such as flags would dramatically increase file size. However, the ability to compress these metadata arrays natively in HDF encourages their inclusion in the file. [19] demonstrated this improved I/O and compression of HDF over FITS during bench-marking of a FITS to HDF converter.

²²<https://www.atnf.csiro.au/computing/software/rpfits.html>

Table 3: SDHDF naming scheme changes

SDHDF <3.0	SDHDF 3.0 (Released: April 2023)	SDHDF 4.0 (Under development)	HDF type
config	config	configuration	Group
cal_data_on	cal_on	calibrator_data_on	Dataset
cal_data_off	cal_off	calibrator_data_off	Dataset
cal32_data	cal.binned	calibrator_data_binned	Dataset
cal_frequency	frequency	frequency	Dataset
data_weights	weights	weights	Dataset
data_flags	flags	flags	Dataset
obs_params	obs_params	observation_parameters	Dataset
cal_obs_params	cal_obs_params	calibrator_observation_parameters	Dataset
band_params	band_params	band_parameters	Dataset
cal_band_params	cal_band_params	calibrator_band_parameters	Dataset
beam_params	beam_params	beam_parameters	Dataset
backend_config	backend_config	instrument_configuration	Dataset
cal_backend_config	cal_backend_config	instrument_configuration	Dataset

- **Accessibility** — Unlike FITS, HDF supports Unicode characters and is therefore suitable for situations where text in languages other than English is required. For instance, we have successfully demonstrated that observer names can be stored in HDF format in Chinese characters.

5.1. Extending the SDHDF format

The spectral-line and continuum astronomy communities who use Parkes for their observations are adopting the new file format and processing tools. Some teams use their own processing tools, others are now making use of INSPECTA. Developers of external tools are also adopting SDHDF. In particular AOFlagger²³, an RFI flagging package, can now parse and flag SDHDF files.

The SDHDF definition is managed in an open source repository and we encourage feedback on its development. We are currently developing SDHDF version 4.0²⁴, for which we have made improvements to the naming scheme, dataset dimensions and attributes based on user feedback, while the underlying structure remains unchanged — important changes that apply to archival data prior to version 4.0 are shown in Tables 3 and 4 respectively.

We do expect further updates as new observing systems are commissioned and more observatories make use of this format. New datasets and attributes will be added in the future. For example, the ability to store statistics from digitisers and a flag for data quality would both be useful additions

²³<https://aoflagger.readthedocs.io/en/latest/>

²⁴We note that for data written in SDHDF <4.0 format, users who wish to work with INSPECTA will need to checkout the ‘sdhdfv3’ branch of the software.

Table 4: SDHDF dataset dimension changes

HDF dataset	SDHDF <3.0	SDHDF ≥3.0	Reason
frequency	1-dimensional	2-dimensional	Allow storage of time-dependent frequency information
flags	2-dimensional	4-dimensional	To match the dimensions of the data, allowing flags to be applied to any dimension of the data
weights	2-dimensional	4-dimensional	To match the dimensions of the data, allowing weights to be applied to any dimension of the data

to the definition and planned for future versions. We note that SDHDF files do not require that all parameters are included. For instance, the observation metadata includes parameters such as the wind speed, which may not be relevant (or even known) for some observations. Similarly, there will always be cases in which a particular observing system requires metadata that is unique to that system. The SDHDF format does not preclude the addition of observatory-specific information.

5.2. Virtual Observatory Support

For the benefit of all users of astronomical data, we would like to see SDHDF become accessible and interoperable with the Virtual Observatory (VO). This can be achieved by following protocols of The International Virtual Observatory Alliance²⁵ (IVOA), an organisation that “debates and agrees the technical standards that are needed to make the VO possible”. The ATOA already has VO interoperability in place for RPFITS format data — we believe that the variety and extensive metadata included in the SDHDF format provides a significant improvement on this.

6. CONCLUSION

In this paper we have presented a new data format for radio astronomy data; the Spectral-Domain Hierarchical Data Format (SDHDF). The format is flexible enough to store high-time and high frequency-resolution data and metadata products from the era of large telescopes with single pixel or multi-beam receivers.

Murriyang, the CSIRO Parkes 64-metre radio telescope, is the first instrument to use the format natively for all spectral-line and continuum observations. The Ultra-Wide-Bandwidth Low Frequency receiver, a single pixel feed, has produced SDHDF files at Parkes since 2018, and the cryogenically-cooled phased array feed, to be commissioned in 2024, is expected to use this format for storing the data products from multiple beams.

We have developed the INSPECTA software package for the interrogation, processing and calibration of SDHDF data, and have demonstrated some of its features when working with radio astronomy data.

²⁵<https://ivoa.net/>

We envisage other institutions adopting SDHDF as their primary format because of the flexibility to store multiple data streams and comprehensive metadata, and its adherence to a formal definition suitable for long-term archiving.

Acknowledgements

The Parkes radio telescope is part of the Australia Telescope National Facility (grid.421683.a) which is funded by the Australian Government for operation as a National Facility managed by CSIRO. We acknowledge the Wiradjuri people as the Traditional Owners of the Observatory site. This paper includes archived data obtained through the Australia Telescope Online Archive (<http://atoa.atnf.csiro.au>). T.V.W. is supported by an NSF Astronomy and Astrophysics Postdoctoral Fellowship under award AST-2202340. We thank Dr. Philip Edwards for his invaluable comments on this paper.

References

- [1] G. Hobbs, R. N. Manchester, A. Dunning, A. Jameson, P. Roberts, D. George, J. A. Green, J. Tuthill, L. Toomey, J. F. Kaczmarek, S. Mader, M. Marquarding, A. Ahmed, S. W. Amy, M. Bailes, R. Beresford, N. D. R. Bhat, D. C. J. Bock, M. Bourne, M. Bowen, M. Brothers, A. D. Cameron, E. Carretti, N. Carter, S. Castillo, R. Chekkala, W. Cheng, Y. Chung, D. A. Craig, S. Dai, J. Dawson, J. Dempsey, P. Doherty, B. Dong, P. Edwards, T. Ergesh, X. Gao, J. Han, D. Hayman, B. Indermuehle, K. Jeganathan, S. Johnston, H. Kanoniuk, M. Kesteven, M. Kramer, M. Leach, V. McIntyre, V. Moss, S. Osłowski, C. Phillips, N. Pope, B. Preisig, D. Price, K. Reeves, L. Reilly, J. Reynolds, T. Robishaw, P. Roush, T. Ruckley, E. Sadler, J. Sarkissian, S. Severs, R. Shannon, K. Smart, M. Smith, S. Smith, C. Sobey, L. Staveley-Smith, A. Tzioumis, W. van Straten, N. Wang, L. Wen, M. Whiting, An ultra-wide bandwidth (704 to 4 032 MHz) receiver for the Parkes radio telescope, PASA37 (2020) e012. [arXiv:1911.00656](#), doi:10.1017/pasa.2020.2.
- [2] R. W. Garwood, SDFITS: A Standard for Storage and Interchange of Single Dish Data, in: N. Manset, C. Veillet, D. Crabtree (Eds.), *Astronomical Data Analysis Software and Systems IX*, Vol. 216 of *Astronomical Society of the Pacific Conference Series*, 2000, p. 243.
- [3] W. D. Pence, L. Chiappetti, C. G. Page, R. A. Shaw, E. Stobie, Definition of the Flexible Image Transport System (FITS), version 3.0, A&A524 (2010) A42. doi:10.1051/0004-6361/201015362.
- [4] D. C. Wells, E. W. Greisen, FITS - a Flexible Image Transport System, in: G. Sedmak, M. Capaccioli, R. J. Allen (Eds.), *Image Processing in Astronomy*, 1979, p. 445.
- [5] B. Thomas, T. Jenness, F. Economou, P. Greenfield, P. Hirst, D. S. Berry, E. M. Bray, N. Gray, D. Muna, J. Turner, M. de Val-Borro, J. Santander-Vela, D. Shupe, J. Good, G. B. Berriman, Significant Problems in FITS Limit Its Use in Modern Astronomical Research, in: N. Manset, P. Forshay (Eds.), *Astronomical Data Analysis Software and Systems XXIII*, Vol. 485 of *Astronomical Society of the Pacific Conference Series*, 2014, p. 351. [arXiv:1502.05958](#), doi:10.48550/arXiv.1502.05958.

- [6] B. Thomas, T. Jenness, F. Economou, P. Greenfield, P. Hirst, D. S. Berry, E. Bray, N. Gray, D. Muna, J. Turner, M. de Val-Borro, J. Santander-Vela, D. Shupe, J. Good, G. B. Berriman, S. Kitaeff, J. Fay, O. Laurino, A. Alexov, W. Landry, J. Masters, A. Brazier, R. Schaaf, K. Edwards, R. O. Redman, T. R. Marsh, O. Streicher, P. Norris, S. Pascual, M. Davie, M. Droettboom, T. Robitaille, R. Campana, A. Hagen, P. Hartogh, D. Klaes, M. W. Craig, D. Homeier, Learning from FITS: Limitations in use in modern astronomical research, *Astronomy and Computing* 12 (2015) 133–145. [arXiv:1502.00996](#), [doi:10.1016/j.ascom.2015.01.009](#).
- [7] D. C. Price, L. Staveley-Smith, M. Bailes, E. Carretti, A. Jameson, M. E. Jones, W. van Straten, S. W. Schediwy, HIPSR: A Digital Signal Processor for the Parkes 21-cm Multibeam Receiver, *Journal of Astronomical Instrumentation* 5 (4) (2016) 1641007. [arXiv:1702.00443](#), [doi:10.1142/S2251171716410075](#).
- [8] M. P. van Haarlem, M. W. Wise, A. W. Gunst, G. Heald, J. P. McKean, J. W. T. Hessels, A. G. de Bruyn, R. Nijboer, J. Swinbank, R. Fallows, M. Brentjens, A. Nelles, R. Beck, H. Falcke, R. Fender, J. Hörandel, L. V. E. Koopmans, G. Mann, G. Miley, H. Röttgering, B. W. Stappers, R. A. M. J. Wijers, S. Zaroubi, M. van den Akker, A. Alexov, J. Anderson, K. Anderson, A. van Ardenne, M. Arts, A. Asgekar, I. M. Avruch, F. Batejat, L. Bähren, M. E. Bell, M. R. Bell, I. van Bemmelen, P. Bennema, M. J. Bentum, G. Bernardi, P. Best, L. Birzan, A. Bonafede, A. J. Boonstra, R. Braun, J. Bregman, F. Breitling, R. H. van de Brink, J. Broderick, P. C. Broekema, W. N. Brouw, M. Brüggen, H. R. Butcher, W. van Cappellen, B. Ciardi, T. Coenen, J. Conway, A. Coolen, A. Corstanje, S. Damstra, O. Davies, A. T. Deller, R. J. Dettmar, G. van Diepen, K. Dijkstra, P. Donker, A. Doorduyn, J. Dromer, M. Drost, A. van Duin, J. Eislöffel, J. van Enst, C. Ferrari, W. Frieswijk, H. Gankema, M. A. Garrett, F. de Gasperin, M. Gerbers, E. de Geus, J. M. Grießmeier, T. Grit, P. Gruppen, J. P. Hamaker, T. Hassall, M. Hoeft, H. A. Holties, A. Horneffer, A. van der Horst, A. van Houwelingen, A. Huijgen, M. Iacobelli, H. Intema, N. Jackson, V. Jelic, A. de Jong, E. Juette, D. Kant, A. Karastergiou, A. Koers, H. Kollen, V. I. Kondratiev, E. Kooistra, Y. Koopman, A. Koster, M. Kuniyoshi, M. Kramer, G. Kuper, P. Lambropoulos, C. Law, J. van Leeuwen, J. Lemaitre, M. Loose, P. Maat, G. Macario, S. Markoff, J. Masters, R. A. McFadden, D. McKay-Bukowski, H. Meijering, H. Meulman, M. Mevius, E. Middelberg, R. Millenaar, J. C. A. Miller-Jones, R. N. Mohan, J. D. Mol, J. Morawietz, R. Morganti, D. D. Mulcahy, E. Mulder, H. Munk, L. Nieuwenhuis, R. van Nieuwpoort, J. E. Noordam, M. Norden, A. Noutsos, A. R. Offringa, H. Olofsson, A. Omar, E. Orrú, R. Overeem, H. Paas, M. Pandey-Pommier, V. N. Pandey, R. Pizzo, A. Polatidis, D. Rafferty, S. Rawlings, W. Reich, J. P. de Reijer, J. Reitsma, G. A. Renting, P. Riemers, E. Rol, J. W. Romein, J. Roosjen, M. Ruiter, A. Scaife, K. van der Schaaf, B. Scheers, P. Schellart, A. Schoenmakers, G. Schoonderbeek, M. Serylak, A. Shulevski, J. Sluman, O. Smirnov, C. Sobey, H. Spreeuw, M. Steinmetz, C. G. M. Sterks, H. J. Stiepel, K. Stuurwold, M. Tagger, Y. Tang, C. Tasse, I. Thomas, S. Thoudam, M. C. Toribio, B. van der Tol, O. Usov, M. van Veelen, A. J. van der Veen, S. ter Veen, J. P. W. Verbiest, R. Vermeulen, N. Vermaas, C. Vocks, C. Vogt, M. de Vos, E. van der Wal, R. van Weeren, H. Weggemans, P. Weltevrede, S. White, S. J. Wijnholds, T. Wilhelmsson, O. Wucknitz, S. Yatawatta, P. Zarka, A. Zensus, J. van Zwieten, LOFAR: The LOW-Frequency ARray, *A&A* 556 (2013) A2. [arXiv:1305.3550](#), [doi:10.1051/0004-6361/201220873](#).
- [9] K. Anderson, A. Alexov, L. Bähren, J. M. Grießmeier, M. Wise, G. A. Renting, LOFAR and HDF5: Toward a New Radio Data Standard, in: I. N. Evans, A. Accomazzi, D. J. Mink, A. H.

- Rots (Eds.), *Astronomical Data Analysis Software and Systems XX*, Vol. 442 of *Astronomical Society of the Pacific Conference Series*, 2011, p. 53. [arXiv:1012.2266](#).
- [10] A. Comrie, A. Pińska, R. Simmonds, A. R. Taylor, Development and application of an HDF5 schema for SKA-scale image cube visualization, *Astronomy and Computing* 32 (2020) 100389. [doi:10.1016/j.ascom.2020.100389](#).
 - [11] A. W. Hotan, W. van Straten, R. N. Manchester, psrchive and psrfits: An open approach to radio pulsar data storage and analysis, *Publications of the Astronomical Society of Australia* 21 (3) (2004) 302–309. [doi:10.1071/AS04022](#).
 - [12] L. D. Anderson, B. Liu, D. S. Balser, T. M. Bania, L. M. Haffner, D. J. Linville, M. Luisi, T. V. Wenger, Methods for averaging spectral line data (2023). [arXiv:2310.09076](#).
 - [13] C. Hazard, M. B. Mackey, A. J. Shimmins, Investigation of the Radio Source 3C 273 By The Method of Lunar Occultations, *Nature* 197 (4872) (1963) 1037–1039. [doi:10.1038/1971037a0](#).
 - [14] J. L. Caswell, J. A. Green, C. J. Phillips, Parkes full polarization spectra of OH masers - II. Galactic longitudes 240° to 350°, *MNRAS* 439 (2) (2014) 1680–1739. [arXiv:1402.2726](#), [doi:10.1093/mnras/stu046](#).
 - [15] N. M. McClure-Griffiths, D. J. Pisano, M. R. Calabretta, H. A. Ford, F. J. Lockman, L. Staveley-Smith, P. M. W. Kalberla, J. Bailin, L. Dedes, S. Janowiecki, B. K. Gibson, T. Murphy, H. Nakanishi, K. Newton-McGee, Gass: The Parkes Galactic All-Sky Survey. I. Survey Description, Goals, and Initial Data Release, *ApJS* 181 (2) (2009) 398–412. [arXiv:0901.1159](#), [doi:10.1088/0067-0049/181/2/398](#).
 - [16] G. Hobbs, R. Hollow, D. Champion, J. Khoo, D. Yardley, M. Carr, M. Keith, F. Jenet, S. Amy, M. Burgay, S. Burke-Spolaor, J. Chapman, L. Danaia, B. Homewood, A. Kovacevic, M. Mao, D. McKinnon, M. Mulcahy, S. Osłowski, W. van Straten, The PULSE@Parkes Project: a New Observing Technique for Long-Term Pulsar Monitoring, *PASA* 26 (4) (2009) 468–475. [arXiv:0907.4847](#), [doi:10.1071/AS09021](#).
 - [17] D. R. Lorimer, M. Bailes, M. A. McLaughlin, D. J. Narkevic, F. Crawford, A bright millisecond radio burst of extragalactic origin, *Science* 318 (5851) (2007) 777–780. [arXiv:https://www.science.org/doi/pdf/10.1126/science.1147532](#), [doi:10.1126/science.1147532](#). URL <https://www.science.org/doi/abs/10.1126/science.1147532>
 - [18] P. Greenfield, M. Droettboom, E. Bray, ASDF: A new data format for astronomy, *Astronomy and Computing* 12 (2015) 240–251. [doi:10.1016/j.ascom.2015.06.004](#).
 - [19] D. C. Price, B. R. Barsdell, L. J. Greenhill, HDFITS: Porting the FITS data model to HDF5, *Astronomy and Computing* 12 (2015) 212–220. [arXiv:1505.06421](#), [doi:10.1016/j.ascom.2015.05.001](#).

Appendix A. THE SDHDF DEFINITION

SDHDF Definition Overview				
SDHDF Definition Version	4.0			
Author	Lawrence Toomey			
Copyright	CSIRO, 2024			

SDHDF File Overview				
HDF.Object.Name	HDF.Object.Type	Description	Unit	Value
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.file
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N	Group	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.beam
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N/beam_0	Group	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.band
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N/beam_N/astronomy_data	Group	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.data
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N/beam_N/astronomy_data/data	Dataset	-	-	-
DATA_DESCRIPTION	Attribute	Data description (Antenna Temperature, Brightness Temperature, Equivalent Point Source Flux Density, other	None	None
DATA_TYPE	Attribute	Data type	None	float32
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FREQUENCY	Attribute	Frequency, data array dimension 2	None	frequency
NORMALISATION_FACTOR	Attribute	The factor by which the number of samples integrated into each output sample are normalised	None	0.0
NUMBER_OF_BINS	Attribute	Number of phase bins	None	1
PHASE_BIN	Attribute	Phase bins, data array dimension 3	None	bin
PRODUCT_TYPE	Attribute	Product type, data array dimension 1	None	None
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.waterfall
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
TIME	Attribute	Integrations, data array dimension 0	None	time
UNIT	Attribute	Data unit, the number of native time samples that are integrated into each output time sample	None	counts
/beam_N/beam_N/astronomy_data/flags	Dataset	-	-	-
DATA_TYPE	Attribute	Data type	None	boolean
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FLAG_TYPE	Attribute	Type of flagging method applied	None	Default: 0=not flagged, 1=flagged)
FREQUENCY	Attribute	Frequency, data array dimension 2	None	frequency
PHASE_BIN	Attribute	Phase bins, data array dimension 3	None	bin
PRODUCT_TYPE	Attribute	Product type, data array dimension 1	None	None
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.flags
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Flags dataset
TIME	Attribute	Integrations, data array dimension 0	None	time
/beam_N/beam_N/astronomy_data/frequency	Dataset	-	-	-
DATA_TYPE	Attribute	Data type	None	float32
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FRAME	Attribute	Spectral reference frame (barycentric, topocentric, LSR, other)	None	topocentric
FREQUENCY	Attribute	Frequency, data array dimension 1	None	frequency
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.frequency
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Channel centre frequencies
TIME	Attribute	Integrations, data array dimension 0	None	time
UNIT	Attribute	Data unit	None	MHz
/beam_N/beam_N/astronomy_data/weights	Dataset	-	-	-
DATA_TYPE	Attribute	Data type	None	float32
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FREQUENCY	Attribute	Frequency, data array dimension 2	None	frequency
PHASE_BIN	Attribute	Phase bins, data array dimension 3	None	bin
PRODUCT_TYPE	Attribute	Product type, data array dimension 1	None	None
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.weights
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Weights dataset
TIME	Attribute	Integrations, data array dimension 0	None	time
UNIT	Attribute	Data unit	None	dimensionless
WEIGHT_TYPE	Attribute	Type of weighting method applied	None	Product of the channel bandwidth (Hz) and integration time (s)
/beam_N/beam_N/calibrator_data	Group	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf.data
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N/beam_N/calibrator_data/data	Dataset	-	-	-

DATA_DESCRIPTION	Attribute	Data description (Antenna Temperature, Brightness Temperature, Equivalent Point Source Flux Density, other	None	None
DATA_TYPE	Attribute	Data type	None	float32
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FREQUENCY	Attribute	Frequency, data array dimension 2	None	frequency
NORMALISATION_FACTOR	Attribute	The factor by which the number of samples integrated into each output bin are normalised	None	0.0
NUMBER_OF_BINS	Attribute	Number of phase bins	None	1
NUMBER_OF_CHANNELS	Attribute	Number of frequency channels	None	0
PHASE_BIN	Attribute	Phase bins, data array dimension 3	None	bin
PRODUCT_TYPE	Attribute	Product type, data array dimension 1	None	None
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_waterfall
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
TIME	Attribute	Integrations, data array dimension 0	None	time
UNIT	Attribute	Data unit, the number of native time samples that are integrated into each output time sample	None	counts
/beam_N/beam_N/calibrator_data/flags	Dataset	-	-	-
DATA_TYPE	Attribute	Data type	None	boolean
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FLAG_TYPE	Attribute	Type of flagging method applied	None	Default: 0=not flagged, 1=flagged)
FREQUENCY	Attribute	Frequency, data array dimension 2	None	frequency
PHASE_BIN	Attribute	Phase bins, data array dimension 3	None	bin
PRODUCT_TYPE	Attribute	Product type, data array dimension 1	None	None
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_flags
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Flags dataset
TIME	Attribute	Integrations, data array dimension 0	None	time
/beam_N/beam_N/calibrator_data/frequency	Dataset	-	-	-
DATA_TYPE	Attribute	Data type	None	float32
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FRAME	Attribute	Spectral reference frame (barycentric, topocentric, LSR, other)	None	topocentric
FREQUENCY	Attribute	Frequency, data array dimension 1	None	frequency
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_frequency
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Channel centre frequencies
TIME	Attribute	Integrations, data array dimension 0	None	time
UNIT	Attribute	Data unit	None	MHz
/beam_N/beam_N/calibrator_data/weights	Dataset	-	-	-
DATA_TYPE	Attribute	Data type	None	float32
DIMENSION_LABELS	Attribute	HDF dimension labels	None	string
FREQUENCY	Attribute	Frequency, data array dimension 2	None	frequency
PHASE_BIN	Attribute	Phase bins, data array dimension 3	None	bin
PRODUCT_TYPE	Attribute	Product type, data array dimension 1	None	None
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_weights
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Weights dataset
TIME	Attribute	Integrations, data array dimension 0	None	time
UNIT	Attribute	Data unit	None	dimensionless
WEIGHT_TYPE	Attribute	Type of weighting method applied	None	Product of the channel bandwidth (Hz) and integration time (s)
/beam_N/beam_N/metadata	Group	-	-	-
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_metadata
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N/beam_N/metadata/calibrator_observation_parameters	Dataset	-	-	-
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Metadata specific to the observation integrations
/beam_N/beam_N/metadata/observation_parameters	Dataset	-	-	-
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Metadata specific to the observation integrations
/beam_N/metadata	Group	-	-	-
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_metadata
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
/beam_N/metadata/band_parameters	Dataset	-	-	-
CENTRE_FREQUENCY	Attribute	Band centre frequency	MHz	0.0
HIGH_FREQUENCY	Attribute	Band range high frequency	MHz	0.0
LABEL	Attribute	Band label	None	None
LOW_FREQUENCY	Attribute	Band range low frequency	MHz	0.0
NUMBER_OF_BINS	Attribute	Number of phase bins	None	1
NUMBER_OF_CHANNELS	Attribute	Number of channels in band	None	0
NUMBER_OF_INTEGRATIONS	Attribute	Number of spectral data integrations in band	None	0
NUMBER_OF_POLARISATIONS	Attribute	Number of polarisations (1, 2, 4)	None	0
PARTIAL_NUMBER_OF_INTEGRATIONS	Attribute	Number of partial or incomplete spectral data integrations in band	None	0

POLARISATION.TYPE	Attribute	Polarisation type (I, Q, U, V, RR, LL RL LR, XX, YY, XY, YX, POLI, POLA, AABBCRCI (for 4 pol coherence data, where AA and BB are the direct products of the two input channels A and B, and CR and CI are the real and imaginary parts of the cross product A* B), AA+BB (for 1 pol data with summed orthogonal products), AABB (for 2 pol data))	None	None
REQUESTED_INTEGRATION.TIME	Attribute	Requested integration time	s	0.0
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Metadata specific to the frequency bands of the antenna beam
/beam_N/metadata/calibrator_band_parameters	Dataset	-	-	-
CENTRE.FREQUENCY	Attribute	Band centre frequency	MHz	0.0
HIGH.FREQUENCY	Attribute	Band range high frequency	MHz	0.0
LABEL	Attribute	Band label	None	None
LOW.FREQUENCY	Attribute	Band range low frequency	MHz	0.0
NUMBER.OF.BINS	Attribute	Number of phase bins	None	1
NUMBER.OF.CHANNELS	Attribute	Number of channels in band	None	0
NUMBER.OF.INTEGRATIONS	Attribute	Number of spectral data integrations in band	None	0
NUMBER.OF.POLARISATIONS	Attribute	Number of polarisations (1, 2, 4)	None	0
PARTIAL.NUMBER.OF.INTEGRATIONS	Attribute	Number of partial or incomplete spectral data integrations in band	None	0
POLARISATION.TYPE	Attribute	Polarisation type (I, Q, U, V, RR, LL RL LR, XX, YY, XY, YX, POLI, POLA, AABBCRCI (for 4 pol coherence data, where AA and BB are the direct products of the two input channels A and B, and CR and CI are the real and imaginary parts of the cross product A* B), AA+BB (for 1 pol data with summed orthogonal products), AABB (for 2 pol data))	None	None
REQUESTED_INTEGRATION.TIME	Attribute	Requested integration time	s	0.0
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Metadata specific to the frequency bands of the antenna beam
/configuration	Group	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_configuration
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	string
/configuration/instrument_configuration	Dataset	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Digital backend instrument configuration
/configuration/receiver_configuration	Dataset	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Receiver instrument configuration
/configuration/telescope_configuration	Dataset	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Telescope configuration
/metadata	Group	-	-	-
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_metadata
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	string
/metadata/beam_parameters	Dataset	-	-	-
DECLINATION	Attribute	Coordinates in Declination	DD:MM:SS.s	None
EQUINOX	Attribute	Equinox of the coordinates	Julian year	2000
LABEL	Attribute	Beam label	None	None
NUMBER.OF.BANDS	Attribute	Number of frequency bands	None	0
RIGHT.ASCENSION	Attribute	Coordinates in Right Ascension	HH:MM:SS.s	None
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Metadata specific to the antenna beam
SOURCE	Attribute	Source name	None	None
/metadata/history	Dataset	-	-	-
DATE	Attribute	Date (UTC YYYY-MM-DD-hh:mm:ss)	None	None
PROCESS	Attribute	Process name	None	None
PROCESSING.HOST	Attribute	Host machine running the process	None	None
PROCESS.ARGUMENTS	Attribute	Process command arguments	None	None
PROCESS.DESCRPTION	Attribute	Process description	None	None
PROCESS.LOG	Attribute	Logged output from the process	None	None
SDHDF.CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF.DESCRPTION	Attribute	SDHDF object description	None	Metadata specific to the processing history of the file
/metadata/primary_header	Dataset	-	-	-
CALIBRATION.MODE	Attribute	Calibration mode (ON: calibrator data recorded, OFF: no calibrator data recorded)	None	None
DATE	Attribute	File creation date (UTC YYYY-MM-DD-hh:mm:ss)	YYYY-MM-DD-hh:mm:ss	None
FILE.FORMAT	Attribute	File format	None	HDF

FILE_FORMAT_VERSION	Attribute	File format version	None	5.0
HEADER_DEFINITION	Attribute	File format definition	None	string
HEADER_DEFINITION_VERSION	Attribute	File format definition version	None	4.0
INSTRUMENT	Attribute	Backend instrument name	None	None
NUMBER_OF_BEAMS	Attribute	Number of beams	None	0
OBSERVATION_TYPE	Attribute	Observation type (TRACK, SCAN)	None	None
OBSERVER	Attribute	Observer name	None	None
PROJECT_ID	Attribute	Project identifier	None	None
RECEIVER	Attribute	Receiver name	None	None
SCHEDULE_ID	Attribute	Observing schedule identifier	None	0
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	General observation metadata
TELESCOPE	Attribute	Telescope name	None	None
UTC_START	Attribute	Observation start (UTC YYYY-MM-DD-hh:mm:ss)	None YYYY-MM-DD-hh:mm:ss	None
/metadata/schedule	Dataset	-	-	-
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Metadata specific to the scheduling of the observation
/metadata/software_versions	Dataset	-	-	-
PROCESS	Attribute	Process name	None	None
SDHDF_CLASS	Attribute	SDHDF class name	None	sdhdf_table
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	Metadata specific to software packages used for creating or processing the file
SOFTWARE	Attribute	Software package	None	None
SOFTWARE_DESCRIPTION	Attribute	Software package description	None	None
SOFTWARE_VERSION	Attribute	Software package version	None	None

Appendix B. Example Observation Parameters

HDF_Object_Name	HDF_Object_Type	Description	Unit	Data_Type
/beam_N/band_N/observation_parameters	Dataset	-	-	-
EQUINOX	Attribute	Equinox of the coordinates	Julian year	string
SDHDF_CLASS	Attribute	SDHDF class name	None	string
SDHDF_DESCRIPTION	Attribute	SDHDF object description	None	string
SOURCE_LABEL	Attribute	Source label for multiple sources, e.g. for scans	None	string
AZIMUTH_ANGLE	Attribute	Antenna azimuth angle	degrees	float64
DECLINATION	Attribute	Declination	DD:MM:SS.s	string
ELEVATION_ANGLE	Attribute	Antenna elevation angle	degrees	float64
ELAPSED_TIME	Attribute	Elapsed time from UTC_START, at integration centre	s	float64
DRIVE_STATUS	Attribute	Telescope drive status (STATIONARY = SLEWING = STOPPING = PARKING = PARKED = STOWED = 0, TRACKING = 1, RAMPING UP = 2 RAMPING DOWN = 2, SCANNING = 3)	NULL	int64
GALACTIC_LATITUDE	Attribute	Galactic latitude	degrees	float64
GALACTIC_LONGITUDE	Attribute	Galactic longitude	degrees	float64
HOURL_ANGLE	Attribute	Hour angle	degrees	float64
INTEGRATION_TIME	Attribute	Actual integration time	s	float64
LOCAL_TIME	Attribute	Local time at the observatory	HH:MM:SS.s	string
MJD	Attribute	Timestamp at integration centre (Modified Julian Date)	days	int64
FRACTIONAL_MJD	Attribute	Time at integration centre since start of the MJD	seconds	float64
PARALLACTIC_ANGLE	Attribute	Parallactic angle	degrees	float64
PRESSURE	Attribute	Atmospheric pressure	hPa	float64
PRESSURE_MSL	Attribute	Atmospheric pressure at mean sea level	hPa	float64
RIGHT_ASCENSION	Attribute	Right ascension	HH:MM:SS.s	string
RELATIVE_HUMIDITY	Attribute	Outside relative humidity	%	float64
TEMPERATURE	Attribute	Outside temperature	degrees C	float64
UTC	Attribute	Timestamp at integration centre (UTC)	HH:MM:SS.s	string
WIND_DIRECTION	Attribute	Wind direction	degrees	float64
WIND_SPEED	Attribute	Wind speed	km/hr	float64
ZENITH_ANGLE	Attribute	Antenna zenith angle	degrees	float64