



UNIVERSITIES SPACE RESEARCH ASSOCIATION

STRATOSPHERIC OBSERVATORY FOR INFRARED ASTRONOMY
NASA AMES RESEARCH CENTER • MS N232-12 • MOFFETT FIELD, CA 94035
• 650-604-1984 (fax) • WWW.SOFIA.USRA.EDU •

14 October 2015
15-WFS-4763

Mr. Edward D. Harmon
NASA Ames Research Center
Mail Stop N232-12
Moffett Field, CA 94035-1000

Dear Mr. Harmon:

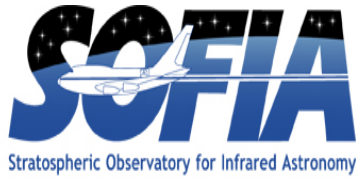
In accordance with the Document Requirements of Contract NAS2-97001, Stratospheric Observatory for Infrared Astronomy, Universities Space Research Association is pleased to provide SCI-US-ICD-SE03-2023, Revision F, Interface Control Document for the Data Cycle System(DCS), a type 1 document.

If you would like to discuss any of this material, please do not hesitate to call.

Respectfully,


William F. Szolnoki
USRA SOFIA Business Manager

WFS/ca

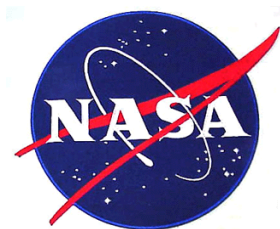


Interface Control Document for the Data Cycle System (DCS)

SCI-US-ICD-SE03-2023

Date: September 24, 2015

Revision: F



AFRC
Armstrong Flight Research Center
Edwards, CA 93523


ARC
Ames Research Center
Moffett Field, CA 94035



German Space Agency, DLR
Deutsches Zentrum für Luft und
Raumfahrt


ICD for the Data Cycle System SCI-US-ICD-SE03-2023, Rev. F

AUTHOR:




Dr. Lan Lin, USRA SOFIA Data Cycle System Manager Date 10/12/2015

APPROVAL:



Maureen Savage, USRA SOFIA Science Instrument
Development Manager Date 10/14/2015



for Dr. William Vacca, USRA SOFIA Assoc. Director for Data
Systems Date 10/14/15



William Szolnoki, USRA SOFIA Business Manager Date 10/14/15

REVISION HISTORY

REV	DATE	DESCRIPTION	AUTHOR
-	5/2/08	Initial Release	R.Y. Shuping
A	6/23/10	<ul style="list-style-type: none"> ▪ Updated keyword list for Early Science ▪ Added section on instrument configurations ▪ Updated section on instrument observation modes ▪ Added appendix describing requirements for Early Science. ▪ Updated sections 1 and 7 to be more clear on what applies to all SIs and what applies to only FSIs. ▪ Corrected spelling and MCS references ▪ Updated datamanifest section for new grammar. ▪ Added section on "Sources for Keyword values" ▪ Update to chop defs and HK locations ▪ Added ChopTip and ChopTilt ▪ Updated KW dictionary version definition (KWDICT) to refer to DCS_SI_01. ▪ Updated section 7 to include missing data values. ▪ Updated obs_id spec to match with new Mission ID spec (see JIRA MOPS-168) 	R.Y. Shuping
B	2/9/2012	<p>Transferred source document to MS Word.</p> <p>Updated AOT sections with new spec.</p> <p>Removed extraneous information from Data Manifest section; moved DM DTD to Appendix C.</p> <p>Moved pipeline spec grammar to Appendix F. Removed material from White Box pipelining section (now TBD).</p> <p>Removed appendix on interface requirements for Early Science (not applicable anymore, refer to Rev. A).</p> <p>Updated Section 2 to include specification of required keyword values for INSTCFG, INSTMODE, SPECTRAL, and SLIT.</p> <p>Added discussion of download limit to Section 7.1 (DCS RFI/RFA 109)</p> <p>Addressed DCS RFI/RFA 171</p>	R.Y. Shuping
C	7/18/2012	<p>Editorial mods to Sections 6 and 7.</p> <p>Section on SI specific keywords removed (7.2)</p>	R.Y. Shuping

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

		<p>Removed references to “Early Science.”</p> <p>Removed discussion of FILENAME keyword that was incorrect(7.3).</p> <p>Removed section on user defined coordinate systems (7.6)</p> <p>Added section 7.9 for Missing Keywords.</p> <p>Updated keyword dictionary in Appendix E (see detailed change log therein).</p> <p>Updated section 7.7 (WCS keywords).</p> <p>Updated Keyword Dictionary; see change log in Appendix E.</p> <p>Replaced Appendix C with current DM DTD and description.</p> <p>Corrected Section 3 to include SPECTEL1/2 and SLIT keywords and current spec for FORCAST.</p> <p>Editorial corrections.</p>	
D	10/16/2012	<p>Updated Keyword Dictionary; see change log in Appendix E.</p> <p>Replaced Appendix C with current DM DTD and description.</p> <p>Corrected Section 3 to include SPECTEL1/2 and SLIT keywords and current spec for FORCAST.</p> <p>Editorial corrections.</p>	R.Y. Shuping
E	3/5/2013	<p>Updated Section 7 to include requirements for Level 2, 3, and 4 (TBD) data, and re-titled.</p> <p>Removed Section 8 (Data Reduction) – this information is now part of the Generic Pipeline requirements.</p> <p>Removed Appendix on Pipeline Specification Grammar: this is now handled internally as part of the Data Processing System.</p> <p>Updated Appendix D (Keyword Dictionary) to include recommended values for Level 2 and 3 data.</p> <p>Added PIPELINE and PIPEVERS to keyword dictionary (no impact to SI teams).</p>	R.Y. Shuping
F	9/24/2015	<p>Removed pipeline reference. Pipeline functions have been moved to the Data Processing System.</p> <p>Added ObsPlan.dtd</p> <p>Added DCS interfaces for FMI and Cycle Scheduler</p> <p>Added ObsPlanObsBlockInfoList.dtd</p> <p>Updated FITS keyword dictionary with new and updated keywords requested by PIPELINE and SI teams</p>	L.Lin

ICD for the Data Cycle System (DCS_SI_01)

SCI-US-ICD-SE03-2023, Rev. F

Table of Contents

1. INTRODUCTION	7
1.1. SUPPORT SCIENCE INSTRUMENTS	7
1.2. SUPPORT FLIGHT PLANNING	8
2. SCOPE OF EXTERNALLY VISIBLE INTERFACES.....	8
3. INSTRUMENT CONFIGURATIONS	9
4. OBSERVING MODES.....	10
5. ASTRONOMICAL OBSERVATION TEMPLATES.....	11
6. OBSERVATION CONTROL.....	12
6.1. INSTRUMENT STATES	13
6.2. INSTRUMENT COMMANDS	13
6.2.1. <i>The Stop Command</i>	14
6.2.2. <i>Asynchronous Status Message from the Instrument</i>	14
6.2.3. <i>Instrument State Diagram</i>	14
7. DATA MANIFESTS.....	15
7.1. INTRODUCTION	15
7.2. LIFE CYCLE OF THE DATA MANIFEST	16
8. SCIENCE INSTRUMENT DATA PRODUCTS	18
8.1. DATA PRODUCT REQUIREMENTS	18
8.1.1. <i>Level 1 Data Product Requirements</i>	19
8.1.2. <i>Level 2 Data Product Requirements</i>	19
8.1.3. <i>Level 3 Data Product Requirements</i>	20
8.1.4. <i>Level 4 Data Product Requirements (TBD)</i>	20
8.2. SCIENCE INSTRUMENT METADATA (KEYWORDS)	20
8.2.1. <i>Overview</i>	20
8.2.2. <i>Science Instrument Observation Keywords</i>	21
8.2.3. <i>Unique Identifiers for SOFIA Observations</i>	21
8.2.4. <i>Data File Structure and Organization</i>	22
8.2.5. <i>Condition</i>	22
8.2.6. <i>Telescope Pointing Coordinate Systems</i>	22
8.2.7. <i>World (Physical) Coordinate Systems</i>	23
8.2.8. <i>Sources for Keyword Values</i>	23
8.2.9. <i>Missing Keywords</i>	23
8.2.10. <i>Common FSI Keywords</i>	24
9. OBSERVATION PLAN	25
9.1. DESCRIPTION	25
9.2. DOWNLOAD OBSPLAN USING THE DCS WEB HTTPS	25
9.3. DOWNLOAD PROPOSALS USING THE DCS WEB HTTPS	25
9.4. OBSPLAN XML DTD DEFINITION	26

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

9.5. OBSPLAN XML DTD DESCRIPTION	30
10. FLIGHTMANAGEMENT MODULE.....	42
10.1. FLIGHTMANAGMENT MODULE DEFINITION	42
10.1.1. Description	42
11. OBSERVATION PLAN / OBSERVATION BLOCK INFO LIST.....	43
11.1. CORBA INTERFACE	43
11.1.1. Retrieve ObsPlanObsBlockInfoList from DCS using CORBA call	43
11.2. HTTPS WEB SERVER PROTOCOL	44
11.2.1. Download ObsPlanObsBlockInfoList from DCS	44
11.3. OBSPLANOBSBLOCKINFOLIST DEFINITION.....	44
11.4. OBSPLANOBSBLOCKINFOLIST DESCRIPTION	46
11.5. OBSPLANOBSBLOCKINFOLIST SAMPLE XML	49
12. API TO UPLOAD A FLIGHT PLAN TO DCS.....	50
12.1. FLIGHT PLAN XML DEFINITION.....	50
12.1.1. Description	50
12.2. UPLOAD FLIGHT PLAN TO DCS WITH HTTPS WEB SERVER PROTOCOL	50
12.2.1. Upload Flight plan to DCS.....	51
13. API TO DONWLOAD A FLIGHT PLAN FROM DCS.....	51
13.1. RETRIEVE A FLIGHT PLAN THROUGH CORBA INTERFACE	51
13.1.1. Get all flight plan IDs for given flight series ID	51
13.1.2. Retrieve a Flight Plan	51
14. CONTRIBUTORS	52
APPENDIX A. ACRONYMS	53
APPENDIX B. SOFIA AOT TEMPLATE & EXAMPLES	54
APPENDIX C. SI KEYWORD DICTIONARY	55

ICD for the Data Cycle System (DCS_SI_01)

SCI-US-ICD-SE03-2023, Rev. F

1. INTRODUCTION

The Data Cycle System (DCS) supports SOFIA science instruments and SOFIA mission operations in the following ways:

1.1. Support Science Instruments

- Provides a common support framework to an instrument through the entire data lifecycle of proposal preparation, observation development and execution, data archiving and reduction, and data retrieval.
- Provides a common representation of instrument functionality (via instrument configurations, observing modes, and astronomical observation templates (AOT)) to the general investigator.
- Simulate planned SOFIA observations (known as astronomical observation requests (AOR)) in order to evaluate their correctness and readiness prior to flight.
- Convert planned SOFIA observations into whatever format is required by the FSI, and queue the observations to the instrument.
- ~~• Execute FSI reduction pipelines on behalf of the investigator, processing the raw data obtained during a flight.~~
- Provide long-term archival and retrieval functions for raw and reduced data.

This document describes and specifies the DCS/SI interfaces that provide these capabilities; the following list summarizes the efforts required of the FSI Teams for the adoption of an FSI into the DCS framework.

1. Define the instrument configurations (§2) and instrument observing modes (§3) to be supported for use by the general investigator.
2. Define any instrument unique structure and grammar of an AOT for each defined observing mode. Note that this task may be simplified to defining only a single AOT structure and grammar to be used across all observing modes.
3. Create any instrument-unique AOT content and all AOT parameters for each AOT.
4. Supply a simulator that generates data typical of the instrument for a given observation.
5. Provide an instrument command-level syntax or API that supports operations to abort, extend, pause, resume and stop an observation, and reports the instrument's status back to the observation queue (refer to §5).
6. Define the syntax translations that map data defined in an AOT to the instrument command-level syntax or API that set up the observation mode.
- ~~7. Create at least one data reduction pipeline for each instrument observing mode, and create~~

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

~~a data reduction pipeline specification §8 for each. This will include the specification of any special pipeline parameters, and pipeline environment variables, used by these pipelines.~~

8. Upon completion of an observation, provide a data manifest (§6, §C), with entries for all data to be archived and reduced.
9. Define any unique keywords (§7) that will be required by users searching for this instruments data in the DCS Archive.

Specification that is unique to an SI (e.g., the actual content of AOT for each of an instrument's modes, instrument-unique keywords) is necessarily contained in a separate SI-to-DCS ICD.

This ICD is intended primarily for facility instruments, but the sections on configurations and modes (§2,§3) apply to any SI that will be supported by the SOFIA Proposal Tool (SPT) and Time Estimators (SITE); and the section on database keywords (§7) applies to all SIs that will submit data to the DCS archive.

1.2. Support Flight Planning

The Data Cycle System (DCS) supports Flight Management Infrastructure (FMI) and Cycle Schedulers (CS) in the following ways:

- Provides interfaces for Flight Plan Editor (FPE) to upload flight plans to the DCS database server for storage, and searching.
- Provides interfaces for DCS to send ObsPlan, Flight Plans and ObsPlanObsBlockInfoList back to the FMI upon retrieval requests:
 - FPE, CS and Short Term Scheduler (STS) can get ObsPlans from DCS
 - CS can get ObservingProposals
 - FMI can get FlightPlan binary, get Flight ID list, and ObsPlanObsBlockInfoList

2. SCOPE OF EXTERNALLY VISIBLE INTERFACES

This document describes the externally visible interfaces provided by the DCS to Science Instruments, FMI and CSs. The DCS externally visible interface consists of Observation Plans (ObsPlans) and ObsPlan/Observation Block Info List definition which are read, written, and transmitted as XML data. The DCS generates and writes ObsPlans as XML data stored in (UTF-8) text files. Alternatively, when the FMI uploads or downloads Flight Plans (FP) to/from the Data Cycle System, XML data are transmitted as text streams. The same applies to the observation blocks info list (make sure we call this the same thing ALL the time - ObsPlanObsBlockInfoList definition. We present in this document the set of method to transmit data and the permissions associated with each of the method where applied.

The FMI to DCS and Cycle Scheduler to DCS ICDs are not part of the scope. The interfaces provided by FMI and CS are documented in FMI Interface Control Document (SCI-US-ICD-SE03-2005), and Interface Control Document for the Cycle Scheduler (SCI-US-ICD-SE03-2040).

3. INSTRUMENT CONFIGURATIONS

Each instrument must define at least one configuration for use and may define additional configurations if they desire. Configurations are **not** the same as observing modes: a configuration describes a particular way that the SI is set up for astronomical observing, whereas an observing mode defines one way to use the instrument in a particular configuration. The SI teams are free to define these configurations as they see fit, but in general, an instrument configuration should map to a unique datatype (IMAGE, SPECTRAL, OTHER), and/or a unique set of spectral elements (filters, gratings, etc...).

For each configuration, the SI-to-DCS ICD needs to specify:

- Datatype produced (IMAGE, SPECTRAL, OTHER; see Keyword Dictionary)
- The name of the configuration to be used in the AOT and FITS files.
- Applicable instrument observing modes (see next section)
- Spectral elements available, and any restrictions/rules governing filter/grating selection;
- Complete set of algorithms/look-up tables for calculating the sensitivity as a function of wavelength/filter.

Some configuration examples:

- **Single Channel Instruments:** An instrument with a single detector and filter set might only need a single configuration, e.g. IMAGING or GRISM. An integrated field spectrometer would probably define a single GRISM configuration.
- **Dual-Use Instruments:** Some instruments may be utilized both as imagers and spectrometers (with the use of grisms or similar implements). In this case, both IMAGING and GRISM configurations should be defined.
- **Multiple Channel Instruments:** Instruments with more than one detector or back-end providing different sets of filters for each channel should consider defining multiple configurations, one for each channel and perhaps one for dual simultaneous channels, e.g. IMAGING_DUAL, IMAGING_SWC, IMAGING_LWC.

The configuration specification will be used in a number of areas of the DCS including:

- the SOFIA Proposal Tool (SPT) and Instrument Time Estimators (SITE);
- AOT specification;
- observation and mission planning;
- archive search criteria.

Hence the SI teams must specify a consistent set of values for configurations, modes, and spectral elements. In particular, the SI must document in the SI-to-DCS ICD all the possible values of the following FITS keywords:

INSTCFG

INSTMODE

SPECTEL1/2

SLIT (if applicable)

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Keyword values should be strings with no whitespace (underscores are allowed); and since they will be used throughout the DCS, they should not be changed without notifying the DCS team and updating the relevant SI-to-DCS ICD. For the SPECTEL1/2 keywords, the SI should use standard names for OTS filters. For custom filters, we recommend using the following prescription:

INS_NNNN

Where “INS” is a three letter instrument code (e.g. “FOR” for FORCAST), and “NNNN” is a 4-numeral designation of the central wavelength or frequency. For example, a FORCAST 38.0 micron filter might be “FOR_3800”, while a 7.6 micron filter would be “FOR_0760”. For spectroscopic instruments that use a slit, all possible slits must be identified with a unique name, again a string with no whitespace (e.g. “SHORTSLITSS_N” for a short slit with fixed width; or “LONGSLITLS_N” for a narrow long slit).

For example, a set of keywords for FORCAST might be:

CONFIG	MODE	SPECTRAL 1	SPECTRAL 2	SLIT
IMAGING	C2N C2NC2	NONE FOR_F064 FOR_F066 FOR_F077 FOR_F111 FOR_F197 FOR_F242 (If NONE specified, single channel implied.)	NONE FOR_F315 FOR_F336 FOR_F348 FOR_F371 (If NONE specified, single channel implied.)	
GRISM_XD	C2N C2NC2	FOR_XG063 FOR_XG111	N/A	FOR_SS24
GRISM	C2N C2NC2	FOR_G063 FOR_G111 FOR_G227 FOR_G329	N/A	FOR_LS24 FOR_LS47

4. OBSERVING MODES

The observing mode is the key idea behind the adoption of FSI within SOFIA; all other aspects of DCS and FSI interfacing are ultimately derived from this common starting point. Additionally, there is an output intended not for the DCS, but for the general investigator: the definition of the mode itself. For these reasons, the observing mode is considered a deliverable in this ICD.

Each FSI is expected to define at least one “observing mode” for use by SOFIA general investigators. An observing mode is an overall method of using the instrument that non-experts might follow in order to obtain useful science results in their observations. The observing mode should capture the best practice of an instrument scientist, for use by other scientists. Though

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

entirely subjective in nature, a well-defined observing mode leaves as little room for error as possible in the use of the instrument by non-expert personnel; consequently, observing modes can be used to present a restricted subset of an instrument's functionality to the general investigator.

The definition of an FSI observation mode should include the name and description of the mode (what the mode accomplishes), the applicable instrument configuration, and the commands and interfacing needed by the instrument in this mode (communication requirements, command languages, API, and so forth). A set of algorithms for reducing the data that the instrument will collect during the observation can also be provided with an observing mode; the DCS team encourages FSI teams to consider data reduction as a part of their instrument deliverables.

Examples of existing observing modes include the HAWC “Stare And Nod” mode, the FORCAST “Two Position Chop” mode, and the FLITECAM “Nodding” mode. The definition of each of these modes would include descriptions suitable for the general investigator to decide which mode best suits his observational needs.

Once an observing mode is defined in such “human terms,” it can be refined into specific DCS inputs such as Astronomical Observation Templates; these are covered in subsequent sections. However, it is important not to overlook the audience of the observing mode: the general investigator. Without a well-defined observation mode, a general investigator would have no way of knowing what observations the instrument is capable of making.

Over the lifetime of the instrument, the initial set of observing modes will undoubtedly be expanded to capture new practice and procedures. The new modes can be added to the DCS later in the same way that at least one observing mode is provided now; the DCS is intended to support multiple observing modes per instrument.

There may be other ways of operating an instrument outside the scope of the DCS, bypassing the development of AOT and other deliverables. These “PI-like” operating models are outside the scope of this document. The FSI team are free to define as many of these other interfaces and models as they deem necessary, however none of them can be considered a deliverable supporting the SOFIA DCS operating model.

5. ASTRONOMICAL OBSERVATION TEMPLATES

Once an observing mode for an instrument has been defined, one will have a list of attributes, parameters, or other settings which need to be provided to the instrument when collecting data in that mode. The astronomical observation template (AOT) is the mechanism by which these settings are defined in the DCS. Once an AOT is defined, the DCS will use it as a guide in the development of observations for this instrument. An observation developed from the contents of an AOT is known as an astronomical observation request (AOR).

An AOT defines not only what parameters need to be supplied to an FSI to make an observation, but it also describes what values those parameters may take. In this way, an AOT may present a simple subset of the total functionality in an FSI to an investigator, or perhaps limit the range of values for a given parameter to a known set that will yield useful observations. An AOT also provides facilities to name the calibration observations that should be used in a target observation, and to supply information used by the observatory itself such as how the telescope and chopper will be used in the observation. It can also define an expected range of valid

observation duration times, providing bounds that reflect constraints set by the observatory. This observatory information includes telescope functionality such as tracking, mapping, dithering, and nodding.

Finally, an AOT indicates the required and optional reduction pipelines in the DCS that can process data collected in this mode. The DCS can provide SOFIA with a wide range of data reduction pipelines; this mechanism exists so that the instrument team can identify which pipelines are known to work with data collected from an instrument in a specific mode. Remember, the AOT contains information that is used to define observations from the observatory's point of view, not just the instrument.

AOTs will be defined internally within the DCS, but the specification of parameters, default values, data types, units, etc... must be provided by the FSI teams. Of special consideration are any dependencies among the parameters: for example, if the values of one parameter are restricted based on the value provided for another parameter. A common set of observatory level parameters will be available for all FSIs; modifications to these parameters should be discussed with the DCS Development and Science Operations teams. A template for defining an FSI AOT is given in B.

Once the AOT has been defined in the specific SI-DCS ICD, the DCS development team will cast the parameters into an XML formatted file for use with DCS AOR Editor (SOFIA Spot) and any other needed tools. The DCS development team must be notified of any proposed changes to the AOT, which then must be documented in the SI-DCS ICD.

6. OBSERVATION CONTROL

This section documents the command and control protocol between the DCS and an FSI. The underlying nature of the interface (e.g. network object functions, socket bytestreams) can be developed between the instrument and DCS teams, but the functionality outlined in this section must be available via that interface. This interface will be used by the DCS Observation Queue (OQ).

On boot, the instrument should start listening for a connection from the Observation Queue. Once connected, the OQ can perform the following classes of actions:

- query for instrument status,
- send instrument-specific configuration commands, and
- command the instrument to start, pause, resume, truncate, and extend whatever data acquisition cycles have been configured.

The instrument responds to these OQ actions with status information as defined here, and with values of the current instrument configuration. The instrument also sends unsolicited (i.e. asynchronous) messages when the acquisition completes, when a problem arises, when a state change occurs due to events not initiated by the DCS, or when normal communication with the DCS is being affected (e.g. the imminent shutdown of a network connection).

6.1. Instrument States

It is understood that there are four instrument states. When the OQ requests the status of the instrument, one of the following results should be returned:

- **ready** The instrument is ready to acquire data. The number of acquisitions (cycles) currently configured is returned with the instrument's status.
- **running** The instrument is currently acquiring data. The number of cycles completed is returned with the instrument's status.
- **paused** Data acquisition by the instrument is currently suspended. The number of cycles completed prior to being paused should be returned with the status. Additionally, the instrument should indicate a reason for the current paused state; typical indications include:
 - The pause was requested via the DCS interface to the instrument.
 - The pause was initiated via a native instrument interface.
 - A problem in an indicated subsystem is responsible for the delay.
- **notready** The instrument is not prepared to acquire data. The reason for this state must be returned with the instrument's status; examples of possible reasons for this state include
 - The instrument has not been configured, or a required configuration parameter has not yet been set.
 - The instrument is not currently receptive to external commands.

6.2. Instrument Commands

There are eight specific commands to which the instrument is required to respond. These commands, and what they represent to the instrument, are presented here:

- **status** Send current instrument state and status.
- **itime** Send current integration time.
- **go** Begin acquisition as configured, then send status.
- **stop** This command immediately reconfigures the instrument to stop acquiring data after n cycles have completed, then send status. The number of cycles to complete, n , will be sent as a parameter with the stop command. The status from this command is further explained below. Response to this command may be delayed until the current cycle is complete, or some other consideration of efficiency or convenience is satisfied.
- **extend** Extend the current observation by a number of cycles n . The number of additional cycles is sent as a parameter with this command. If this number n is positive, the observation should be lengthened by n cycles. If this number n is negative, then the observation should be shortened by $-n$ cycles.
- **pause** Suspend data acquisition, then send status. Response to this command may be delayed until the current cycle is complete, or some other consideration of efficiency or convenience is satisfied.
- **resume** Resume data acquisition, then send status.

- **abort** Immediately end data acquisition, then send status. This command should be given a higher priority and immediacy in its processing than either the **stop** or **pause** commands.

In addition to the commands above, other commands specific to the instrument are expected. These commands are used to create or modify observation configuration information present in the instrument. They, and their results, must be documented by the FSI team along with the documentation that presents how the instrument state and command interfaces are presented to the DCS.

If the instrument receives a command it cannot interpret, it must return an error indicating such a condition. This error response must be obvious and distinct from all other FSI responses, and it must reproduce the objectionable content from the illegal command. This error should be provided in the simplest format matching the instrument's interface (e.g. a simple string for an FSI communicating over a byte stream, a structure for an FSI communicating over CORBA).

The remainder of this section provides additional information on specific aspects of the FSI/OQ command interface.

6.2.1. The Stop Command

The **stop** function immediately reconfigures the instrument to acquire data for only \$n\$ cycles in the current series. Instrument specific configuration commands are required to reconfigure the number of total cycles permanently. Because of this, **status** can indicate that it is "running cycle 3 of 2," indicating that after completing 3 cycles and while working on the 4th cycle, the **stop** command was received with a parameter of 2. After the 4th cycle, the instrument will respond to the new termination value and end the observation. Since the original configuration was to complete, say, 10 cycles, and this configuration is not affected by the **stop** command, the new status for this run can be "completed, 4 of 10."

6.2.2. Asynchronous Status Message from the Instrument

Although most communication in the OQ system is synchronous, there is one asynchronous (i.e. unsolicited) message sent by the instrument to the DCS. This is a `[[status]]` message, and is typically sent by the FSI as a reaction to one of the following events:

- Data acquisition has completed.
- Data acquisition has resumed.
- Instrument is entering the **notready** state.
- Instrument is back from the **notready** state.

6.2.3. Instrument State Diagram

Now that states and commands have been presented, a state diagram outlining the transitions between instrument states and the commands that cause them is presented in Figure 1. **Bold transition text** indicates the transition criteria, while the plain text underneath presents the expected results after the state change.

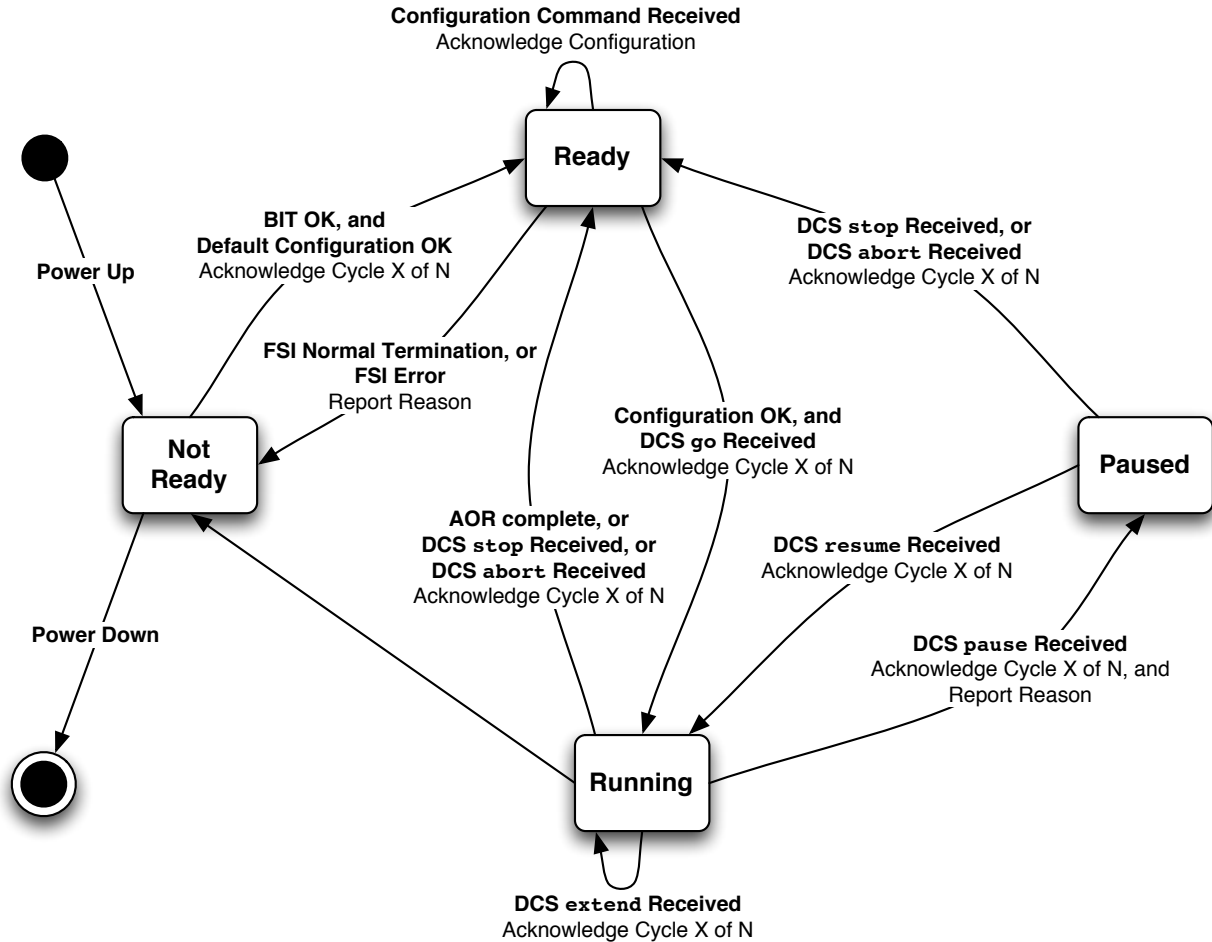


Figure 1: Instrument State Diagram

7. DATA MANIFESTS

7.1. Introduction

One of the first questions that arises when we consider archiving data from SOFIA is exactly what data need to be archived and where they are located. Before we can worry about database management and driver design we must first address this simple issue. The Archive Data Manifest Interface Format is designed to provide the necessary type and location information for any data destined for the archive.

A simple format for an archive data manifest would be an ASCII text file containing a list of Uniform Resource Locators (URL). Such a file would address the fundamental requirements of identifying the location of each file that needs to be archived. From the standpoint of data “producers”, the data manifest described in this document is really just a list of URIs that point to unique files. The DCS Core and archive subsystems, however, require a data interchange format that has greater semantic richness.

This document describes a simple eXtensible Markup Language (XML) based format for describing archive data manifest documents. The use of XML allows us to take advantage of the large base of commercially available software for reading, writing and validating Xml documents. Programmatic manipulation of XML documents is actually considerably easier than the manipulation of a simple text file.

The archive data manifest is a convenient format for exchanging information between the archiving system, the DCS core, and data producers (if desired). The data manifest was originally developed to provide a one-way flow of information from data “producers” to the archive (the primary data “consumer”). The need for a bi-directional means of communication between the DCS core and archive systems has expanded the role of the archive data manifest to include several functions beyond its original design goals. Data flows from the “producers” (e.g. instruments, MCCS, etc...) through the DCS Core to the archive by the creation of data manifest XML documents. The results of the archiving process, including any errors that are generated, are transmitted back to the DCS core via modified data manifests. The final version of a processed data manifest XML document (produced by the archiving software) includes all the information necessary to retrieve the archived data products from the SOFIA DBMS (Database Management System) and a record of when and how all of the data “ingestion” was carried out (times, software driver versions, operators, etc.).

The Data Manifest document type definition (DTD) and an example DM file are shown in Appendix C.

7.2. Life Cycle of the Data Manifest

The archive is the ultimate consumer of all data produced by the SOFIA observatory. Data files will be produced by a variety of sources including the MCCS, scientific instruments, astronomers and SMO personnel. In order to be archived, these files must be described by a **[[Datasource]]** element within a Data Manifest XML document. The Data Manifest file is the document that a producer provides to the archive in order to identify data products intended for the archive.

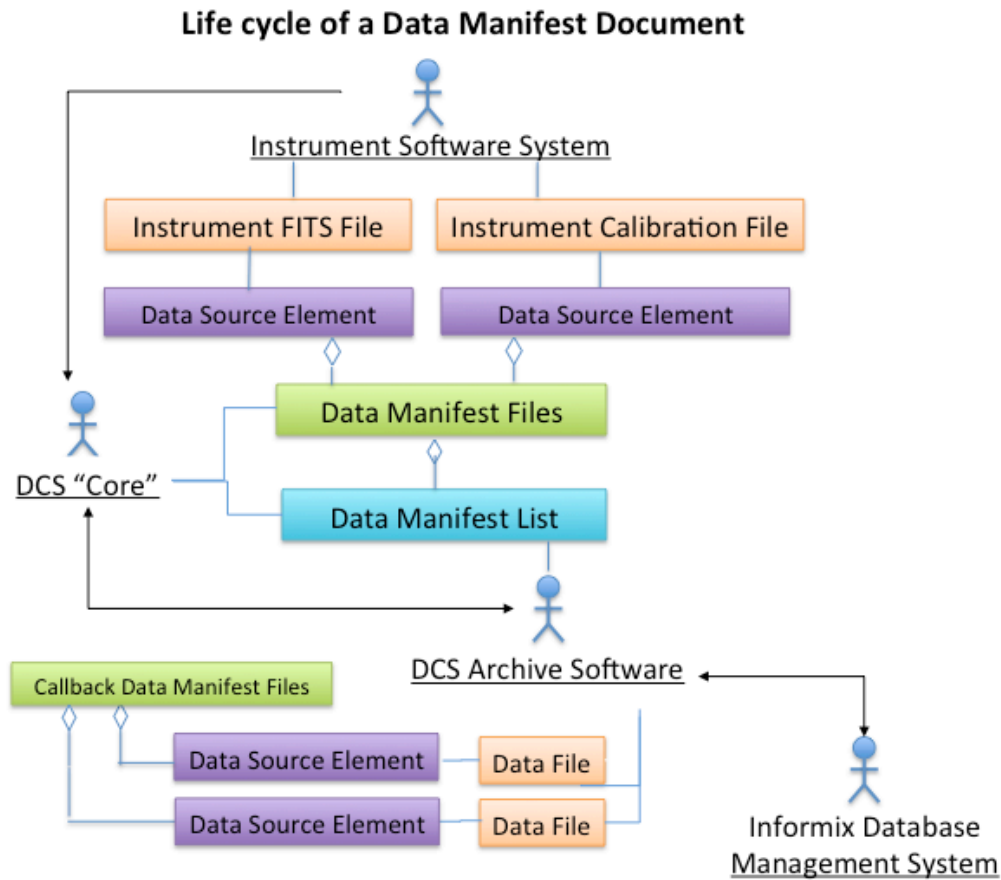


Figure 2: Life Cycle of a Data Manifest Document (Activity Diagram)

Who produces the Data Manifest file for a given set of documents? The instrument group or software system that produces a set of files is the unit with the necessary expertise to create a data manifest. The data manifest DTD is the first point of contact and one part of the interface between SOFIA data providers and the DCS Archive. Each data “producer” must provide a data manifest document that describes the data that will be submitted to the archive. The data manifest file created by each “producer” will be provided to the DCS “core” software system. The DCS “core” software system will collect and package all data manifest files produced during a given operation and will hand these files off to the Archive system. The Archive software will process the data manifest files (obtained from the DCS core) and populate the archive's DBMS. Alternately, the DCS core can also generate data manifest on a given set of mission data to be ingested by the archive system.

The Data Manifest XML file is also used to communicate the “status” and results of the archiving process back to the DCS Core system and ultimately to the “producers”. Errors that occur during archiving are documented by a combination of **[[Errorreport]]** and **[[Timestamp]]** elements that are individually associated with **[[Datasource]]** elements. These elements are added to the **[[SofiaDatamanifest]]** document during processing. After the processing of a Data Manifest document is completed, the modified XML document is submitted back to the DCS Core containing the “status” and results of the archiving process. The “success” or “failure” of

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

the DBMS transaction is indicated by the **[[Archivingstatus]]** element that may contain multiple **[[Archivekey]]** elements. The **[[Archivekey]]** elements can be used to retrieve the data from the SOFIA DBMS system using Structured Query Language (SQL). Since the Archive system has no direct knowledge of how to distribute the results of the archiving process, it is the responsibility of the DCS Core to assign responsibility for resolving errors to the appropriate actors. In some (most) cases, errors may be resolved by SOC personnel (e.g. DBA). In other circumstances, however, it may be necessary to refer the errors to the data “producers” (i.e. instrument groups) for resolution. After a data manifest file has been successfully processed it will be archived itself.

The DCS includes a tool called the Data Manifest Manager (DMM) for creating and modifying data manifest documents. The DMM allows documents conforming to the DTD to be created, written to, or read from a file, and validated against the DTD. The utility allows users to browse the local area network for files that are then added as **[[Datasource]]** elements to the data manifest file. **[[Datasource]]** elements may be added or removed from a document and existing elements may be modified. Existing data manifest documents may be edited and validated against the current DTD and new documents may be easily created.

MCCS House Keeping data are now extracted by a house keeping data reader and loaded to the house keeping data base. Ingestion of the MCCS HK data does not use the data manifest.

8. SCIENCE INSTRUMENT DATA PRODUCTS

8.1. Data Product Requirements

Four levels of data products are recognized by SSMO:

- Level 1 data are essentially raw from the instrument but in standardized format (FITS or CLASS);
- Level 2 data have been corrected for instrument artifacts (e.g., flats, darks, bad pixels);
- Level 3 data have been flux calibrated (e.g. BUNIT keyword takes image to MJy/sr);
- Level 4 data are high-order products possibly combining multiple exposures (e.g. mosaics, spectral cubes)

(see **Data Processing Plan for SOFIA Science Instruments** [SCI-US-PLA-PM17-2010] for detailed descriptions.) All SI data submitted to the DCS for archiving must be in FITS format (currently) and adhere to the metadata (keyword) requirements in Section 8.2. In addition, there are a small set of requirements for each Level.

Recommended values for all level data products:

- AOR_ID – This is the only unique keyword that is carried from the proposal to observing plan to flight plan and level 1/2/3/4 data products. This keyword allows DCS to provide an accurate status on the AORs through out the planning and the only keyword that ties the data files that are in the various processing state.

- **PLANID** – This is the only keyword that can tie a data file to its original owner of the data, e.g. the PI that submitted the proposals that generated these data. This keyword enables the archive search by PI names.

8.1.1. Level 1 Data Product Requirements

In order to facilitate archive searches, the **values** of the following keywords must be specified in the SI-DCS ICD according to Section 3 above:

INSTRUME *
 INSTCFG
 INSTMODE
 SPECTEL1/2 *
 SLIT

The values **must** match with the values specified for the instrument configurations, modes, and AOTs. Note that a single list of possible values can be submitted for both SPECTEL1 and 2. The values for these keywords will be used to construct database searches on the archive; values **not** listed in the SI-DCS ICD will be accepted by the archive, but will not be searchable (they will be collected up under the search term “OTHER”). Some important notes:

- Since this is a list for archival data, it is not restricted by current (or past) cycle offerings. The lists of values should be seen as comprehensive over time and can include filters, configs, modes, that are **not** offered to the general investigator.
- SPECTEL1/2 *: for instruments that only use one filter at a time, the one that is not taking data should be populated with “NONE”.
- We expect these lists to grow over time as SI teams add/change configs, filters, and modes. Anytime there are changes to the SI in these areas, the corresponding section of the SI-DCS ICD will need to be updated and the DCS team notified so that the archive search page can be updated to support the SI changes.

The SI team must provide specification for any instrument specific keywords that are required for data processing and calibration (including any keywords required for grouping data and what values they will need to have). Specification should include the keyword name, datatype, expected values (and/or range), and a short definition. These keywords are **not** parsed by the DCS archive, but will be used to validate Level 1 files in preparation for processing at the SSC.

8.1.2. Level 2 Data Product Requirements

Level 2 data products must also adhere to the SOFIA FITS keyword dictionary (Appendix E). In many cases the values of the keywords in the LEVEL_2 files can be carried forward from the LEVEL_1 files, but in some cases the pipeline will need to determine and update the values accordingly. Details listed in Appendix E.

In some cases, SI pipelines might produce different products as part of Level 2 processing. These will be distinguished using the PRODTYPE keyword; possible values must be specified in the SI-DCS ICD.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

The following information must be provided in the HISTORY keywords at the end of the header:

- List of input OBS_IDs for the input LEVEL_1 in the input manifest. This gives the GI or archive user a quick way of seeing what files were used to generate the LEVEL_2 product.
- List of any auxiliary data products used.
- List of processing steps completed and any errors encountered.

8.1.3. Level 3 Data Product Requirements

Level 3 data products must adhere to the same requirements as Level 2. In addition, the Level 3 products must:

- Provide imaging flux calibration information in the FITS header using standard conventions (e.g. BSCALE, BZERO, and BUNIT); pixel values in the image should not be changed.
- Provide spectroscopic flux calibration information by appending a calibration spectrum (cal factors vs. wavelength) to the Level 2 data; pixel values in the image should not be changed.
- Provide a reference to the mission calibration file used in the HISTORY keywords.

8.1.4. Level 4 Data Product Requirements (TBD)

TBD

8.2. Science Instrument Metadata (Keywords)

8.2.1. Overview

All science data files submitted by the SI teams to the DCS for archiving must be in FITS format and adhere to the FITS Standard (v3.0, 10 July 2008). In addition, the FITS header must contain the keywords specified in the FITS Keyword Dictionary (Appendix E). Note that not all of the keywords in the list are required to be present in every FITS file submitted for archiving (see the "Condition" column).

The SI-DCS ICD should contain a description of the SI data files that will be archived post flight, including any ancillary files. The description must also include specification of the values to be used for the INSTRUME, INSTCFG, INSTMODE, SPECTEL1/2, and SLIT (if applicable) keywords (see Section 2).

Also note that the SOFIA Archive has a download limit of no less than 2 GB (contact DCS development team for current value). Therefore any single file produced by the science instrument that exceeds this value can not be retrieved from the DCS archive. If the SI team expects to produce files larger than the download limit, they should notify the DCS development team so that the download limit can be increased (or other provisions made).

8.2.2. Science Instrument Observation Keywords

The primary interface between an SI and the DCS Archive is the metadata, or keywords, used to describe the science data produced by the SI. These keywords are controlled with a keyword dictionary which was initially developed by the SOFIA Archive Development Team at UCLA. This dictionary began through consultations with the HAWC, AIRES and FLITECAM instrument teams and borrows heavily from their early keyword lists. Respecting the instrumental perspective we have tailored the keyword dictionary to:

- be as transparent and user friendly as possible, for the general observer who will be referring to these keywords in data headers,
- facilitate straight-forward data reduction and, for the facility instruments, data pipelining,
- allow for efficient searches on the SOFIA database, in keeping with our evolving design of the SOFIA data archive, and
- institute commonality in the choice of keywords so that translations can be avoided when software is shared between instruments.

In offering a set of common keywords, we are aiming to unify the choices of parameter names so that users of multiple instruments on SOFIA will need to learn only one set, so that software can be shared between groups using different instruments, and so that we have a coherent set of database referents in the archive.

During development of these keywords, we have tried to accommodate alternative ways of taking or storing data. For example, we have not made any choices about whether an image is to be stored as a normal FITS file or as a binary extension table in a FITS file. Time of sample for any keyword value is assumed to be at the start of data acquisition unless otherwise specified. The keyword dictionary and additional specifications are given in §D.1. We have tried to adopt standard FITS keywords whenever possible and to follow common usage as laid out in the NOAO Keyword Dictionary¹. We recommend instrument developers consult this dictionary for their instrument specific keywords as well. As always, any input regarding the keywords is welcome.

8.2.3. Unique Identifiers for SOFIA Observations

Currently there are two unique identifiers for every SOFIA science observation: AORUniqueID and ObservationID (FITS: AOR_ID, OBS_ID). The relationship and specification of each is discussed below.

The AORID is assigned by the DCS during the creation of an Astronomical Observation Request (AOR) in the observation planning process. If the current observation is not associated with an AOR, then this keyword is not required.

The Observation ID is a unique identifier for any given data file, meant to provide ready timing and sequencing information at a glance to the user. It is created at the same time that the data file is first written, in a format common to all instruments (and thus mandated by the observatory).

¹ <http://iraf.noao.edu/iraf/web/projects/ccdmosaic/imagedef/fitsdic.html>

The OBSID is assigned to saved datasets only. Some test data may be taken during a flight for immediate diagnostic purposes, but not saved (e.g., a snapshot to verify pointing), in which case there would be no point in assigning an OBSID, but we note that the default should almost always be to save the data and thus assign this keyword. See the Keyword dictionary listing for the specification of OBSID.

The OBSID number is distinct from the AORID. The AORID is the unique identifier for an AOR which can, in principle, generate multiple files to be saved. Each of these files would have a unique OBSID. The mapping from AORID to OBSID is one-to-many, i.e., one AOR can produce multiple data files. Note that the converse is not true—one data file cannot be the product of multiple AORs.

All recorded data files would have an OBSID, even if the data are judged to be bad. This allows for later second opinions, or for the resurrection of marginal, but time-critical data. Another keyword, DATAQUAL, is under development to describe data quality.

8.2.4. Data File Structure and Organization

Every science data file submitted to the SOFIA archive must include (among other things) the ObservationID keyword (FITS: OBS ID) with a value which adheres to the unique identifiers specification (§7.2). Each file may have multiple images (e.g., FITS Extensions), indexed using the ImageID keyword, which constitute a single “observation”. We leave it to the SI teams and investigators to define what an “observation” means. For the FSI, this is largely set via the AOR/AOT concept. Therefore, a single data file may contain, for example:

- All chop/nod pairs of an observation as separate images;
- The target observation and any calibration observations (e.g., fluxcalibrators, flatfields);
- All the images in a dithering/mapping observation.

We also acknowledge that in many cases groups of images will be stored individually in separate files. In this case, the data files can be associated using the FileGroupID (FITS: FILEGPID) keyword, where all related files would be given the same value. The FileGroupID will be especially useful for map/dithered images stored as separate files to be combined later.

8.2.5. Condition

Each keyword has a condition associated with it which indicates when a particular keyword is required for ingestion into the archive. For example, if the data were collected while chopping (isChopping = T), then the set of chopping-related keywords and values would need to be present. If no condition is listed, then the keywords must be present in every FITS file submitted to the SOFIA archive. They are necessary to (1) uniquely identify the origin of the data, (2) produce primary search indices for the archive database, and (3) satisfy minimum summary information about an observation.

8.2.6. Telescope Pointing Coordinate Systems

The position of the SI boresight as returned by the MCCS in equatorial coordinates (i.e., where the telescope thinks it is pointing) is specified using the TELRA/DEC and TELEQUI keywords (ICRS J2000 is the default for MCCS). The SI will also need to supply the **requested** RA and

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Dec from the AOR (if applicable) using the OBSRA, OBSDEC and EQUINOX keywords.

8.2.7. World (Physical) Coordinate Systems

Application of a world, or physical, coordinate system (WCS) to a dataset is usually accomplished by specifying for each axis; units, a reference pixel, a value for the reference pixel, the relative pixel separation in the specified units (which can be non-linear), and a rotation. In FITS headers these relations are usually specified with the keywords CTYPE_n, CRPIX_n, CRVAL_n, and the rotation matrix elements CD_{i,j} (or alternatively, the older convention keywords CDELT_n and CROTA_n), where *n* is the axis index. For WCS transformations, we recommend the use of these keyword conventions as discussed in A User's Guide for the Flexible Image Transport System (FITS)². All SOFIA data must contain valid WCS keywords: the SI team may choose to implement **either** the CD matrix form, **or** the CDELT and CROTA form, but **not** both. The FITS comment field must also carry an uncertainty estimate for the following WCS keywords:

- CRVAL_n: Uncertainty in the telescope absolute pointing (see TBD Document).
- CD_{i,j}: Uncertainty for each element in the rotation matrix. This is really a combination of the uncertainty in the telescope rotation of field (see TBD document) **and** the uncertainty in the platescale across the detector.
- CROTA₂: Uncertainty in the telescope rotation of field (see TBD Document).
- CDELT_n: Uncertainty in the platescale, as determined by the SI team. Should account for distortion and other optical effects. If the data are distortion corrected, then this should include any fit errors.

8.2.8. Sources for Keyword Values

The listing in the dictionary for each keyword contains a source field which indicates where the value for that keyword can be obtained. Keywords with no source listed are provided by the SI directly. For Early Science, there is no computational interface for the Observing or Mission plans, therefore any values that are sourced from those plans must be configured at run-time or queried from the user at the time the FITS file is written.

For keywords that are sourced from MCCA Housekeeping (HK) data, the MCCA SI 04 should be consulted for additional clarifications. If the HK datanode is not available, or returns "NotFound" (or any other error), then the following values should be used to populate the corresponding FITS keyword based on the FITS keyword data type (float, int, string, boolean):

FLOAT -9999

INT -9999

STRING UNKNOWN

BOOL No blanket value; must be defined for each keyword

In addition, a note should be inserted into the keyword comment line indicating that there was an error accessing the MCCA HK data item.

8.2.9. Missing Keywords

² http://fits.gsfc.nasa.gov/fits_wcs.html and see also <http://tdc-www.harvard.edu/wcstools/wcstools.fits.html>

If any of the required keywords are missing (or have the wrong datatype) the DCS archive will issue an error report indicating which files are non-compliant and which keyword(s) are missing. In most cases, the files will still be ingested into the archive and made available for download via the DCS webpages. Some keywords, however, are required for ingestion: if one of these keywords is missing, the host file will **not** be available until the header is corrected. The following keywords are absolutely required for archive ingestion:

- DATASRC
- OBS_ID
- MISSN-ID
- DATE-OBS
- INSTRUME
- SPECTEL1
- SPECTEL2

See Appendix for keyword requirements.

Non-compliant files can then be “checked-out” of the archive by SMO staff, updated, and then checked back in. On check-in, the database tables will be updated automatically with the new keyword values and the updated files made available for download.

8.2.10. Common FSI Keywords

Included here is list of common SI keywords and possible standardizations. These keywords have not been inducted into the dictionary, we are waiting for feedback from the SI teams.

- **FLATFILE**: Pointer to flatfield file. We might consider a whole group of calibration file pointers, e.g., FLATFILE, DARKFILE, LAMPFILE
- **GAIN** or **EPERADU**: Electrons per ADU (DN). Most instruments seem to have a specification for this so we should consider standardizing one.
- **CAMMODE**: Camera mode.
- **DETBIAS**: Detector bias voltage.
- **SMPLFREQ**: Detector sample frequency (alternative to COADDS).
- **SMPLPIMG**: Detector samples per chop image (again, as an alternative to COADDS).

9. OBSERVATION PLAN

9.1. Description

The DCS generates and writes ObsPlans as XML data stored in (UTF-8) text files. Once a proposal submitted by a Guest Investigator to the DCS is approved, the proposal is turned into an ObsPlan. The ObsPlan contains general proposal information and a list of Astronomical Observation Request (AOR). Each AOR contains key information such as instrument, instrument mode, target, exposure time on the target and the overhead required to set up the pointing of the telescope.

The AORs are grouped together by the Cycle Scheduler to form the Observation Blocks (ObsBlock) list. The Flight Management Infrastructure (FMI) then plans flights based on the list of ObsBlock.

Only privileged users or internal software are allowed to access the observation plan.

9.2. Download ObsPlan using the DCS Web HTTPS

To download an ObsPlan from DCS, the FMI opens an HTTPS connection and passes the ObsPlan ID string to the DCS. The server returns the status and the ObsPlan data if successful.

- URL: *https://[DCSWEBSITE]/observationPlanning/DbProxy/getObsPlan.jsp*
- Input parameters
 - *id* String – ObsPlan ID
- Output: XML string that contains the return value of the call. If it is a success, the xml contains the ObsPlan xml. If permission is designed, the user must log in first.
 - Success: `<RetVal><Success/></RetVal>`
 - Error: `<Error><MustLogin/></Error>`
- Permission: Only the following roles can get an ObsPlan
 - GI or Co-I who owns this ObsPlan
 - SMO

9.3. Download Proposals using the DCS Web HTTPS

Downloading proposals (not approved yet) from DCS is very similar to downloading the ObsPlans. The difference is that instead using *getObsPlan.jsp*, it calls *getPropCover.jsp* to get the proposal cover XML. To do this, the Cycle Scheduler (CS) or Short Term Cycle Scheduler (STS) opens an HTTPS connection, and passes the proposal ID string to the DCS. The server returns the status and the Proposal data if successful.

- URL: *https://[DCSWEBSITE]/observationPlanning/DbProxy/getPropCover.jsp*
- Input parameters
 - *id* String – Proposal ID

- Output: If it is a success, the xml contains the ProposalCover xml. The ProposalCover and ObsPlan XML definition are the same except the root element. If the caller does not have the right privilege, the server returns “Permission Denied” response. In this case, the user must log in with the right privilege first.
- Only the following roles can get a proposal.
 - DCSAdmin
 - TACLead
 - CSAdmin

9.4. Obsplan XML DTD Definition

```
<!-- Content for an ObservingPlan -->
<!-- add boolean flag IsLocked element to indicate ObsPlan status. If IsLocked is
set
    then the ObsPlan cannot be edited -->
<!-- add boolean flag MustDo element for SPR1635 -->
<!ELEMENT ObservingPlan (
    ObsPlanID, PropCycleID, Proposal, Investigator+, ProposedObservation+,
    SSMOCScientist?, Note*, IsLocked?, MustDo? ) >

<!-- add completionrate attribute that represents completion rate (0.00 - 1.00)
of AORs have been DONE i.e.
    completionrate = Sum(AORState.Done)/Sum(proposedobservation) -->
<!-- Add state attribute in ObservingPlan element to indicate state of an ObsPlan
-->
<!-- State can be either PENDING, PROBLEM, or APPROVED -->
<!ATTLIST ObservingPlan
    phase          ( Skeletal | Populated | FullySpecified | Refined ) #REQUIRED
    lastdatamod CDATA #REQUIRED
    lastphasemod CDATA #REQUIRED
    completionrate CDATA #IMPLIED
    state CDATA #IMPLIED >
<!-- comment out version CDATA #FIXED "2" for now -->

<!ELEMENT ObsPlanID ( #PCDATA ) >
<!ELEMENT PropCycleID ( #PCDATA ) >

<!-- Content for a ProposalCover -->
<!ELEMENT ProposalCover ( Proposal, Investigator+, ProposedObservation+ ) >

<!ELEMENT Proposal (PropID, Title, Category, ScienceKeywords, ProposalAbstract,
    TotalObsDuration, isQueue?, isService?, isTOO?, IsImpact?,
    isSurvey?, PropDocURI?, SpecialInstructions?, AwardedTime?) >
<!-- add SMOMember -->
<!ATTLIST Proposal
    TACQueue          CDATA #IMPLIED
    datesubmitted CDATA #IMPLIED
    TACGrade          CDATA #IMPLIED
    TACMember         CDATA #IMPLIED
    status            ( Pending | Approved | Declined ) #REQUIRED
    SMOMember         CDATA #IMPLIED >

<!ELEMENT PropID ( #PCDATA ) >
<!ELEMENT SpecialInstructions ( #PCDATA ) >
<!ELEMENT ScienceKeywords ( #PCDATA ) >
```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```

<!ELEMENT Title ( #PCDATA ) >
<!ELEMENT Category ( #PCDATA ) >
<!ELEMENT TotalObsDuration ( #PCDATA ) > <!-- Seconds -->
<!ELEMENT ProposalAbstract ( #PCDATA ) >
<!ELEMENT PropDocURI ( #PCDATA ) >
<!-- Observation scheduling modes. Boolean flags -->
<!ELEMENT isQueue EMPTY >
<!ELEMENT isService EMPTY >
<!ELEMENT isTOO EMPTY >
<!ELEMENT IsImpact EMPTY >
<!ELEMENT isSurvey EMPTY >

<!ELEMENT AwardedTime ( #PCDATA ) > <!-- Seconds -->
<!-- Each ObservingPlan must have one or more investigators, one of which should be
      flagged as Primary -->
<!ELEMENT Investigator (Primary?,Identity,Address?,Email?,Phone?,Website?)>
<!ELEMENT Primary EMPTY >
<!ELEMENT DCSUserName ( #PCDATA ) >
<!ELEMENT Identity (DCSUserName?)>
      <!ATTLIST Identity
        FirstName CDATA #REQUIRED
        LastName CDATA #REQUIRED
        MiddleInitial CDATA #IMPLIED
        Honorific CDATA #IMPLIED
        Suffix CDATA #IMPLIED
        JobTitle CDATA #IMPLIED
        Institution CDATA #REQUIRED>
      <!ELEMENT Address EMPTY>
        <!ATTLIST Address
          Street1 CDATA #IMPLIED
          Street2 CDATA #IMPLIED
          City CDATA #IMPLIED
          State CDATA #IMPLIED
          Country CDATA #IMPLIED
          Postcode CDATA #IMPLIED>
      <!ELEMENT Phone EMPTY>
        <!ATTLIST Phone
          HomePhone CDATA #IMPLIED
          OfficePhone CDATA #IMPLIED
          CellPhone CDATA #IMPLIED
          Fax CDATA #IMPLIED>
      <!ELEMENT Email ( #PCDATA ) >
      <!ELEMENT Website ( #PCDATA ) >

<!--
  An ObservingPlan must have one or more ProposedObservations.
  This information will be used for first AOR creation.

  Add AORState element to indicate AOR state. AORState can be either
  NEW, READY, PROBLEM or DONE

  Add boolean flag IsNewObservation element to indicate that the observation was
  added during the phase 2 planning

  isComplete indicate whether the ProposedObservation has been observed and
  archived
-->
<!ELEMENT ProposedObservation
  ( ObservationNumber, Instrument, Duration, Overhead, MapArea, AstroObject,
    Timing?, isCalibrator?, isTimeCritical?, AORID?, DependentObs?,
    Comments?, ObservationID?, FlightManagement?, isComplete?, Duplicate?,
    AORState?, IsNewObservation?, MinimumContiguousTime?, AORExeDates?) >

<!-- WaterVapor and watervapor_max mapping between SPT and Proposal Cover XML

```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```

    low = 5.0 microns
    medium = 10.0 microns
    otherwise = NaN

    altitude_min: feet
    elevation_min: degrees
-->
    <!ATTLIST ProposedObservation
        order          CDATA    #IMPLIED
        priority        CDATA    #IMPLIED
        watervapor_max  CDATA    #IMPLIED
        altitude_min    CDATA    #IMPLIED
        elevation_min    CDATA    #IMPLIED
        key              CDATA    #IMPLIED>

    <!ELEMENT ObservationNumber ( #PCDATA )> <!-- Serial number (by ObsPlan) for
                                                proposed observations. -->
    <!ELEMENT MapArea ( #PCDATA )> <!-- Square arcmin, 0 for single pointing -->
    <!ELEMENT Duration ( #PCDATA )> <!-- Seconds -->
    <!ELEMENT Overhead ( #PCDATA )> <!-- Seconds -->
    <!ELEMENT isCalibrator EMPTY> <!-- Is the proposed observation for
                                                calibration? Boolean flag. -->

    <!ELEMENT AORID ( #PCDATA )>
    <!ELEMENT ObservationID ( #PCDATA )>
    <!ELEMENT Comments ( #PCDATA )>
    <!-- Now the instrument specifications - an instrument must have a name
        attribute, a config, and a spectral element. A second spectral element
        is optional. -->
    <!ELEMENT Instrument ( Config, Mode, SpectralElement, SpectralElement2?, Slit?,
        Wavelength?, Wavelength2?, Frequency?, Frequency2?,
        DitherPattern?)>
    <!ATTLIST Instrument
        name CDATA #REQUIRED >
    <!ELEMENT Config ( #PCDATA )>
    <!ELEMENT Mode ( #PCDATA )>
    <!ELEMENT SpectralElement ( #PCDATA )>
    <!ELEMENT SpectralElement2 ( #PCDATA )>
    <!ELEMENT DitherPattern ( #PCDATA )>
    <!ELEMENT isTimeCritical EMPTY>
    <!-- Timing is defined with a positive convention,
        i.e. the time interval defined by Before and After is defined as
        the period during which the observations should be carried out. -->
    <!ELEMENT Timing ( After_UTDate?, Before_UTDate? ) * >
    <!ELEMENT Before_UTDate ( #PCDATA )>
    <!ELEMENT After_UTDate ( #PCDATA )>
    <!-- Each proposed observation must have an AstroObject associated with it. -->
    <!ELEMENT AstroObject ( AstroObjectName, AstroObjectType, ObservationCenter,
        ProperMotion?, Velocity?, Guidestar*)>
    <!ELEMENT AstroObjectType ( #PCDATA )>
    <!ELEMENT AstroObjectName ( #PCDATA )>

    <!-- ObservationCenter defines location on the sky, following AOR convention -->
    <!ELEMENT ObservationCenter ( Sidereal | NonSidereal )>
    <!ELEMENT Sidereal ( RA, Dec, Equinox )>
    <!ELEMENT RA ( #PCDATA )> <!-- decimal hours -->
    <!ELEMENT Dec ( #PCDATA )> <!-- decimal degrees -->

    <!ELEMENT NonSidereal ( NAIF_ID | OrbitalElements | (RA,Dec,Equinox))>
    <!ELEMENT OrbitalElements (Epoch, PerihelionDistance, Eccentricity,
        Inclination, littleOmega, BigOmega, PerihelionDay)>
    <!ELEMENT NAIF_ID ( #PCDATA )>
    <!ELEMENT PerihelionDistance ( #PCDATA )> <!-- Units: AU -->
    <!ELEMENT Eccentricity ( #PCDATA )>

```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE


```

<!ELEMENT Inclination      ( #PCDATA )> <!-- UnitS: degrees -->

<!-- Argument of Perihelion (DEG) with regard to J2000 ecliptic/equinox -->
<!ELEMENT littleOmega      ( #PCDATA )>

<!-- Longitude of Ascending Node (DEG) with regard to J2000 ecliptic/equinox -->
<!ELEMENT BigOmega         ( #PCDATA )>
<!ELEMENT PerihelionDay    ( #PCDATA )> <!-- Perihelion Julian Day Number -->
<!ELEMENT Epoch           ( #PCDATA )> <!-- Epoch of the osculating elements,
Julian date -->

<!ELEMENT Equinox ( J2000 )>
<!ELEMENT J2000 EMPTY >

<!ELEMENT ProperMotion ( PropMotnRA, PropMotnDec ) >
<!ELEMENT PropMotnRA ( #PCDATA ) > <!-- arcsec/yr -->
<!ELEMENT PropMotnDec ( #PCDATA ) > <!-- arcsec/yr -->

<!-- Rest frequency (GHz) for line of interest: Heterodyne instruments -->
<!ELEMENT Frequency ( #PCDATA ) >

<!-- Rest frequency (GHz) in the second
channel/backend for line of interest: Heterodyne instruments -->
<!ELEMENT Frequency2 ( #PCDATA ) >

<!ELEMENT Velocity ( #PCDATA ) > <!-- either km/s or z, helio and lrs km/s, z no
units -->
<!ATTLIST Velocity ref ( helio | lsr | z ) #REQUIRED >
<!ELEMENT Guidestar      ( GuideStarName?, GuideStarRA, GuideStarDec,
GuideStarMagnitude, GuideStarBminusV? ) >
<!ELEMENT GuideStarName  ( #PCDATA )>
<!ELEMENT GuideStarRA    ( #PCDATA )> <!-- decimal hours -->
<!ELEMENT GuideStarDec   ( #PCDATA )> <!-- decimal degrees -->
<!ELEMENT GuideStarMagnitude ( #PCDATA )>
<!ELEMENT GuideStarBminusV ( #PCDATA )>
<!ATTLIST Guidestar Imager ( FPI | FFI ) #IMPLIED >
<!-- Proposed observations may also have associated dependent observations. -->
<!ELEMENT DependentObs ( ObservationID?, isContiguous? )>
<!ATTLIST DependentObs key CDATA #IMPLIED>
<!ELEMENT isContiguous EMPTY>
<!ELEMENT FlightManagement (PlannedExecution? )>
<!ATTLIST FlightManagement InstRunID CDATA #REQUIRED >
<!ELEMENT PlannedExecution ( #PCDATA ) >
<!ELEMENT SSMOCScientist ( #PCDATA ) >
<!-- Each ObservingPlan can have 0 or more timestamped notes. -->
<!ELEMENT Note ( #PCDATA ) >
<!ATTLIST Note
    Doc_Link CDATA #IMPLIED
    timestamp CDATA #REQUIRED >

<!-- Add boolean flag IsLocked element to indicate status of an ObsPlan.
If IsLocked is set, then the ObsPlan cannot be edited -->
<!ELEMENT IsLocked EMPTY >

<!-- add boolean flag MustDo element for SPRI635 -->
<!ELEMENT MustDo EMPTY >

<!-- add a new Boolean flag to indicate completion status of proposed observation
-->
<!ELEMENT isComplete EMPTY >

<!-- add a new element Duplicate to indicate the ProposedObservation is duplicate -
->

```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```
<!ELEMENT Duplicate (#PCDATA) >

<!-- Add AORState element to indicate AOR state. AORState can be either
      NEW, READY, PROBLEM or DONE -->
<!ELEMENT AORState ( #PCDATA )>

<!-- Add AORExeDates element to record AOR execution dates (separated by a comma)
-->
<!ELEMENT AORExeDates ( #PCDATA )>

<!-- add boolean flag IsNewObservation element to indicate that the observation was
      added during the phase 2 planning -->
<!ELEMENT IsNewObservation EMPTY >

<!-- SPR:1272 Add display of AOR minimum contiguous time to AOR display page -->
<!ELEMENT MinimumContiguousTime ( #PCDATA )>
<!-- SPR:1177 Last minute updates to SI configs/modes for Cycle 2 -->
<!ELEMENT Slit ( #PCDATA )>
<!--Added in Cycle 4 -->
<!ELEMENT Wavelength ( #PCDATA )>
<!ATTLIST Wavelength range CDATA #IMPLIED> <!-- units: microns -->
<!-- SPR:1666 add Wavelength2 -->
<!ELEMENT Wavelength2 ( #PCDATA )>
<!ATTLIST Wavelength2 range CDATA #IMPLIED> <!-- units: microns -->
```

9.5. Obsplan XML DTD Description

		Class	Type
ObservingPlan	phase	Attribute	Text
	lastdatamod	Attribute	Text
	lastphasemod	Attribute	Text
	completionrate	Attribute	Text
	state	Attribute	Text
	ObsPlanID	Element{1}	Text
	PropCycleID	Element{1}	Text
	Proposal	Element{1}	See Below
	Investigator	Element{1,N}	See Below
	ProposedObservation	Element{1,N}	See Below
	SSMOCScientist	Element{Opt}	Text
	Note	Element{0,N}	Text, See Below
	IsLocked	Element{Opt}	EMPTY
	MustDo	Element{Opt}	EMPTY

phase: The phase the proposal is at. It has one of the following values:
Skeletal

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Populated
FullySpecified
Refined

lastdatamod: The last time the ObsPlan was changed, e.g. 2012-09-03 22:47:19.0

lastphasemod: The last time the phase was changed, e.g. 2012-09-03 22:47:19.0

completionrate: The rate of completion, fraction of number of AORs that are in DONE state with regard to the total number of AORs. This value ranges from 0.00-1.00.

state: The state of this ObsPlan, The list of states are described in the state diagram (in the DCS Software Architectural Design Document SCI-US-SPE-SW02-2008)

ObsPlanID: The ID of the ObsPlan, it is the same as the proposal ID and has the form of CC_XXXX. CC is the SOFIA proposal cycle ID, XXXX is a serial number assigned to a single proposal.

PropCycleID: ID of a proposal Cycle (01, 02, etc.).

Proposal: The contents of the proposal.

Investigator: The investigator of this proposal. A proposal can have more than one Investigator.

ProposedObservation: The observation in the proposal. There can be any number of observations in a proposal. Each ProposedObservation is one AOR.

SSMOCScientist: SSMOC scientists assigned to review this proposal. It is a comma delimited string that includes multiple users.

Note: Proposal review notes.

IsLocked: An empty element indicating whether this ObsPlan is locked or not. If locked, then the GIs will not be allowed to upload their AORs.

MustDo: An empty element indicating that this ObsPlan must be planned for flight.

		Class	Type
ProposalCover	Proposal	Element{1}	See Below
	Investigator	Element{1,N}	See Below
	ProposedObservation	Element{1,N}	See Below

The ProposalCover element is used to contain information about the proposal before a proposal is approved.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Proposal: see Proposal under ObservingPlan.

Investigator: see Investigator under ObservingPlan.

ProposedObservation: see ProposedObservation under ObservingPlan.

		Class	Type
Proposal	TACQueue	Attribute	Text
	datesubmitted	Attribute	Text
	TACGrade	Attribute	Text
	TACMember	Attribute	Text
	status	Attribute	Text
	SMOMember	Attribute	Text
	PropID	Element{1}	Text
	SpecialInstructions	Element{1}	Text
	ScienceKeywords	Element{1}	Text
	Title	Element{1}	Text
	Category	Element{1}	Text
	TotalObsDuration	Element{1}	Text
	ProposalAbstract	Element{1}	Text
	PropDocURI	Element{Opt}	Text
	isQueue	Element{Opt}	EMPTY
	isService	Element{Opt}	EMPTY
	isTOO	Element{Opt}	EMPTY
	IsImpact	Element{Opt}	EMPTY
	isSurvey	Element{Opt}	EMPTY
	AwardedTime	Element{Opt}	Text

TACQueue: Which TACQueue this proposal belongs to, it can be “US” or “DE”.

datesubmitted: The date when the proposal was submitted.

TACGrade: The grade that the TAC assigns to the proposal. (Numeric number between 0 and 5)

TACMember: The TAC member that is assigned to rate this proposal.

status: The status of this proposal. The status can be Pending or Approved or Declined.

SMOMember: SMO member that are assigned to review this proposal.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

PropID: The ID of this proposal. It is the same as the ObsPlanID.

SpecialInstructions: Special instructions on this proposal.

ScienceKeywords: The science keywords for this proposal.

Title: The title of this proposal.

Category: The category of the targets.

TotalObsDuration: The total duration of this proposal, including exposure time and overhead.

ProposalAbstract: The abstract of this proposal.

PropDocURI: The location of the proposal file stored on the server.

isQueue: An empty element indicating whether this proposal is a Queued proposal.

isService: An empty element indicating whether this proposal is a Service proposal.

isTOO: An empty element indicating whether this proposal is a Target of Opportunity proposal.

IsImpact: An empty element indicating whether this proposal is an Impact proposal, these proposals are usually large and require more than one cycle)

isSurvey: An empty element indicating whether this proposal is a Survey proposal.

AwardedTime: The total time awarded to this proposal, in minutes.

		Class	Type
Investigator	Primary	Element{Opt}	EMPTY
	Identity	Element{Opt}	See Below
	Address	Element{Opt}	See Below
	Email	Element{Opt}	Text
	Phone	Element{Opt}	See Below
	Website	Element{Opt}	Text

Primary: An empty element indicating whether this is a primary investigator. Only one investigator can be the primary, the rest are co-investigators.

Identity: The identity of this investigator.

Address: The address of this investigator.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Email: The email provided by this investigator for contacting purpose. This can be the same email address the user uses to register with the DCS, but not necessary.

Phone: The phone number of this investigator.

Website: The web site provided by this investigator.

		Class	Type
Identity	FirstName	Attribute	Text
	LastName	Attribute	Text
	MiddleInitial	Attribute	Text
	Honorific	Attribute	Text
	Suffix	Attribute	Text
	JobTitle	Attribute	Text
	Institution	Attribute	Text
	DCSUserName	Element{1}	Text

FirstName: The first name of the investigator. It can be used to search ObsPlan and archives on the DCS web.

LastName: The last name of the investigator. It can be used to search ObsPlan and archives on the DCS web.

MiddleInitial: The middle initial of the investigator.

Honorific: The honorific of the investigator.

Suffix: The suffix of the investigator.

JobTitle: The job title of the investigator.

Institution: The institution of the investigator.

DCSUserName: The account user name that the user has registered with the DCS. This user name is used to determine access to various web pages, such as searching ObsPlan, AORs, and archives.

		Class	Type
Address	Street1	Attribute	Text
	Street2	Attribute	Text
	City	Attribute	Text
	State	Attribute	Text

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

	Country	Attribute	Text
	Postcode	Attribute	Text

		Class	Type
Phone	HomePhone	Attribute	Text
	OfficePhone	Attribute	Text
	CellPhone	Attribute	Text
	Fax	Attribute	Text

		Class	Type
ProposedObservation	order	Attribute	Text
	priority	Attribute	Text
	watervapor_max	Attribute	Text
	altitude_min	Attribute	Text
	elevation_min	Attribute	Text
	key	Attribute	Text
	ObservationNumber	Element{1}	Text
	Instrument	Element{1}	See Below
	Duration	Element{1}	Text
	Overhead	Element{1}	Text
	MapArea	Element{1}	Text
	AstroObject	Element{1}	See Below
	Timing	Element{Opt}	See Below
	isCalibrator	Element{Opt}	Text
	isTimeCritical	Element{Opt}	See Below
	AORID	Element{Opt}	Text
	DependentObs	Element{Opt}	See Below
	Comments	Element{Opt}	Text
	ObservationID	Element{Opt}	Text
	FlightManagement	Element{Opt}	See Below
	isComplete	Element{Opt}	EMPTY
	Duplicate	Element{Opt}	Text
	AORState	Element{Opt}	Text
	IsNewObservation	Element{Opt}	EMPTY
	MinimumContiguous	Element{Opt}	Text

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

	Time		
	AORExeDates	Element{Opt}	Text

ProposedObservation: A proposed observation is an AOR.

order: The order of this AOR. It is an integer. Multiple AORs can have the same order, which means orders are not important.

priority: The priority of this AOR. It can be 1 (high) or 2 (medium) or 3 (low).

watervapor_max: The maximum water vapor tolerance for this AOR. It can be high (> 10 micron), medium (between 5 and 10 micron), and low (<5 micron).

altitude_min: The minimum altitude this AOR should be observed at, in feet.

elevation_min: The minimum elevation this AOR should be observed at, in degree.

key: The observation key assigned to this AOR. A unique id assigned to this AOR by the DCS server database. It is the same as an AOR ID.

ObservationNumber: A serial number assigned to this AOR.

Instrument: The instrument to be used for this observation.

Duration: Duration of this observation, including integration time and overhead.

Overhead: The overhead of this observation.

MapArea: The map area of this observation if this is a mapping AOR. For single pointing, the value is set to 0.

AstroObject: The target of this observation.

Timing: An id of the timing constraint.

isCalibrator: An empty element indicating if this is a calibration AOR.

isTimeCritical: Contains one or more Timing definitions indicating the time constraints required for this observation. The Timing defines the time period in which this observation should be carried out.

AORID: The ID assigned to the AOR. It has the form of [ObsPlanID]_[ObservationNumber].

DependentObs: Other AOR this observation depends on.

Comments: The comments regarding this observation.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

ObservationID: A unique id assigned to this observation, it is not the same as AOR ID.

FlightManagement: Contains the flight plan info for this AOR.

isComplete: An empty element indicating whether or not this AOR has been completed.

Duplicate: The AOR ID of an AOR that this AOR is duplicate of.

AORState: AORState can be NEW, READY, PROBLEM or DONE.

IsNewObservation: An empty element indicating that the observation was added during the phase 2 planning.

MinimumContiguousTime: The minimum contiguous time to maintain for observation if this AOR must be split into multiple flights.

AORExeDates: The dates when this AOR was observed. This data is populated after the archive receives the data. It can be multiple dates, e.g. 2014-05-01 2014-05-08

		Class	Type
Instrument	name	Attribute	Text
	Config	Element{1}	Text
	Mode	Element{1}	Text
	SpectralElement	Element{1}	Text
	SpectralElement2	Element{Opt}	Text
	Slit	Element{Opt}	Text
	Wavelength	Element{Opt}	Text
	Wavelength2	Element{Opt}	Text
	Frequency	Element{Opt}	Text
	Frequency2	Element{Opt}	Text

Name: The name of the instrument.

Config: The configuration of the instrument.

Mode: The observing mode of the telescope, e.g. C2NC2, STARE.

SpectralElement: The spectral element 1 to use for this AOR.

SpectralElement2: The spectral element 2 to use for this AOR.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Slit: The slit to use for this AOR.

Wavelength: The wavelength to use for spectral element 1.

Wavelength2: The wavelength to use for spectral element 2.

		Class	Type
Wavelength	range	Attribute	Text
Wavelength2	range	Attribute	Text

range: Range of the wavelength , in microns.

Frequency: The frequency to use for spectral element 1.

Frequency2: The frequency to use for spectral element 2.

		Class	Type
isTimeCritical	Timing	Attribute	Text

Timing: The timing constraints are defined in the Timing element.

		Class	Type
Timing	Before_UTDate	Element{1}	Text
	After_UTDate	Element{1}	Text

Before_UTDate: The observation should happen before this date, in YYYY-MM-DD

After_UTDate: The observation should happen after this date, in YYYY-MM-DD

		Class	Type
AstroObject	AstroObjectName	Element{1}	Text
	AstroObjectType	Element{1}	Text
	ObservationCenter	Element{1}	See Below
	ProperMotion	Element{Opt}	See Below
	Velocity	Element{Opt}	Text

AstroObjectName: The name of the observation target.

AstroObjectType: The type of the observation target.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

ObservationCenter: The center of the observation.

ProperMotion: The proper motion of the target.

Velocity: The velocity of the target, in km/s if ref is helio or lrs, no units if ref is z

		Class	Type
Velocity	ref	Attribute	Text

ref: Reference of the velocity, it can be either helio or lsr or z.

		Class	Type
ObservationCenter	Sidereal	Element{1}	Text
	NonSidereal	Element{1}	Text

Sidereal: Sidereal target.

NonSidereal: Non-Sidereal target, including planets, asteroid, orbital elements.

		Class	Type
Sidereal	RA	Element{1}	Text
	Dec	Element{1}	Text
	Equinox	Element{1}	See Below

RA: Right ascension, in decimal hours.

Dec: Declination, in decimal degree.

Equinox: The equinox of the object. Defined as J2000

		Class	Type
NonSidereal	NAIF_ID	Element{1}	Text
	OrbitalElements	Element{1}	See Below
	RA	Element{1}	Text
	Dec	Element{1}	Text
	Equinox	Element{1}	See Below

NonSidereal can be defined as either a NAIF_ID or orbital element.

NAIF_ID: Navigation and Ancillary Information Facility(NAIF) assigned ID for an object.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

OrbitalElements: are used to describe the motion of satellites within an orbit.

RA: Right ascension, in decimal hours.

Dec: Declination, in decimal hours.

Equinox: The equinox of the object. Defined as J2000.

		Class	Type
OrbitalElements	Epoch	Element{1}	Text
	PerihelionDistance	Element{1}	Text
	Eccentricity	Element{1}	Text
	Inclination	Element{1}	Text
	littleOmega	Element{1}	Text
	BigOmega	Element{1}	Text
	PerihelionDay	Element{1}	Text

OrbitalElements is defined by 7 parameters:

Epoch: Epoch of the osculating elements, Julian date.

PerihelionDistance: Perihelion distance (AU).

Eccentricity: The eccentricity of the orbit.

Inclination: Inclination of orbit plane (DEG) with regard to J2000 ecliptic plane.

littleOmega: Argument of Perihelion (DEG) with regard to J2000 ecliptic/equinox. It is the angle (in the body's orbit plane) between the ascending node line and perihelion measured in the direction of the body's orbit. This angle is often denoted as lower-case omega (ω).

BigOmega: Longitude of Ascending Node (DEG) with regard to J2000 ecliptic/equinox. It is the angle between the reference X-direction (typically the vernal equinox) and the point at which the body passes up (north) through the reference plane. This angle is often denoted as capital omega (Ω).

PerihelionDay: Perihelion Julian Day Number.

		Class	Type
Equinox	J2000	Element{1}	EMPTY

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

		Class	Type
ProperMotion	PropMotnRA	Element{1}	Text
	PropMotnDec	Element{1}	Text

PropMotnRA: Rate of angular change in RA over time, as observed from the center of mass of the Solar system, measured in arcsec/yr.

PropMotnDec: Rate of angular change in Dec over time, as observed from the center of mass of the Solar system, measured in arcsec/yr.

		Class	Type
GuideStar	GuideStarName	Element{Opt}	Text
	GuideStarRA	Element{1}	Text
	GuideStarDec	Element{1}	Text
	GuideStarMagnitude	Element{1}	Text
	GuideStarBminusV	Element{1}	Text
	Imager	Attribute	Text

GuideStars are results from querying a public database for guide stars within a 0.5 degree radius around the current observation position and (if possible) brighter than 13th magnitude. From the database list, select up to the five brightest stars within 4 arcminutes of the current observation position which are brighter than 13.0 magnitude (these are stars suitable for use in the FPI). If there are no stars found, then select up to the five brightest stars within 0.5 degrees of the current observation position which are brighter than 9.0 magnitude (these are stars suitable for the FFI).

Each guide star is defined by the following parameters:

GuideStarName: Name of the guide star from catalog.

GuideStarRA: RA of the guide star, in decimal hours.

GuideStarDec: Dec of the guide star, decimal degrees.

GuideStarMagnitude: Magnitude of the guide star.

GuideStarBminusV: B-V is a value computed from B and V magnitudes retrieved from querying the database. If B and/or V is missing, then B-V is left blank.

Imager: FPI or FFI is used for guide stars.

Class	Type
-------	------

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

DependentObs	key	Attribute	Text
	ObservationID	Element{Opt}	Text
	isContiguous	Element{Opt}	EMPTY

key: The AOR ID of the AOR that this AOR depends on.

ObservationID: The ObsPlan ID of the dependent AOR.

isContiguous: An empty element indicating if this AOR and the dependent AOR are contiguous.

FlightManagement	InstRunID	Class	Type
		Attribute	Text
	PlannedExecution	Element{Opt}	Text

InstRunID: The instrument run ID of a flight plan for this AOR. (Obsolete)

PlannedExecution: The planned execution time for this AOR.

Note	Doc_Link	Class	Type
		Attribute	Text
	timestamp	Attribute	Text

Doc_Link: The file path on the server that stores this ObsPlan.

timestamp: Time stamp of the note. Format: UT YYYY-MM-DD HH:MM:SS.xxx

10. FLIGHTMANAGEMENT MODULE

The FlightManagment Module defines a list of services provided by the DCS to search, upload, and retrieve flight planning data.

10.1. FlightManagment module Definition

10.1.1. Description

The CORBA module of FlightManagement contains the definition of FMView Object and the services this object provides.

```
module FlightManagement {
```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```
interface FMView : ObsPlanDB::BaseAPI {
    typedef sequence<string> FlightPlanIDSeq;
    typedef sequence<octet> FlightPlanBinary;

    PDRetVal getObsBlockInfoList(
        in string obsPlanId,
        out string xml );

    PDRetVal getFlightPlanIDs (
        in string flightSeriesID,
        out FlightPlanIDSeq ids);

    PDRetVal getFlightPlanBinary (
        in string flightPlanID,
        out FlightPlanBinary fpdata );
};
```

getObsBlockInfoList: Get ObsPlan Observation Blocks from the server for a particular ObsPlan ID. The return value is in the form of XML and is described in section “ObsPlanObsBlockInfoList Definition” below.

getFlightPlanIDs: Get a list of Flight Plan IDs for a specific FlightSeries ID.

getFlightPlanBinary: Get a Flight Plan binary file for a specific Flight Plan ID. The binary format is defined in FMI-MCCS ICD and is not part of the scope of this document.

11. OBSERVATION PLAN / OBSERVATION BLOCK INFO LIST

This section documents the retrieval and the contents of the ObsPlanObsBlockInfoList. There are two ways of getting a ObsBlockInfoList. One is through a CORBA call using the FlightManagerment Module defined above. The other is to using a web interface. Both methods return XML that contains the ObsBlockInfoList.

The definition of ObservationBlock is defined in the Cycle Scheduler ICD (SCI-US-ICD-SE03-2040) and is not part of the scope of this document.

11.1. CORBA interface

11.1.1. Retrieve ObsPlanObsBlockInfoList from DCS using CORBA call

```
module FlightManagement {
    VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE
```



```
        PDRetVal getObsBlockInfoList(  
            in string obsPlanId,  
            out string xml );  
  
    }
```

11.2. HTTPS web server protocol

11.2.1. Download ObsPlanObsBlockInfoList from DCS

- To download ObsPlan Observation Blocks from the DCS, use URL:
<https://dcs.sofia.usra.edu/observationPlanning/DbProxy/GetObsPlanObsBlockInfoList.jsp>
- Input parameter name: obsplanid
- Privilege: SMO, DCSAdmin, TACLead, FPAdmin and CSAdmin only. Flight Planner can log in as FPAdmin to upload Flight Plans. CSAdmin can upload Cycle Schedules. A GI does not have privileges to see Flight Plan related data.
- Output:
 - ObsPlanObsBlockInfoList XML – Success
 - Error Message
 - Permission Denied – GI or Public user
 - ObsPlanID cannot be empty – Bad input
 - No Matching Record – the given ObsPlanID doesn't exist in the database
 - ObsPlan exists, but it does not have Observation Block
 - DCS system error, and please try again later – Internal error occurs, e.g. DB connection failure

11.3. ObsPlanObsBlockInfoList Definition

- The ObsPlanObsBlockInfoList are not flight plans, they are data extracted from the Observing Plans packaged in a specific way to be used by the flight planner during flight planning.
- ObsPlanObsBlockInfoList DTD

```
<?xml version='1.0' encoding='UTF-8'?>  
  
<!ELEMENT ObsPlanObsBlockInfoList (RetVal, ObsBlockInfo*,  
Proposal?)>  
<!ELEMENT RetVal (#PCDATA)>  
<!ELEMENT Proposal (TACGrade, Title, PropID, MustDo?,  
PIEmail, PILastName, PIFirstName)>  
<!ELEMENT TACGrade (#PCDATA)>  
<!ELEMENT Title (#PCDATA)>  
<!ELEMENT PropID (#PCDATA)>  
<!ELEMENT PEmail (#PCDATA)>  
<!ELEMENT PILastName (#PCDATA)>  
<!ELEMENT PIFirstName (#PCDATA)>
```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```
<!-- add boolean flag MustDo element for SPR1635 -->
<!ELEMENT MustDo EMPTY >

<!ELEMENT ObsBlockInfo (AORList, ExecutedDuration,
PlannedDuration, RequestedDuration,
ObservationCenter, Instrument, ObsBlockName, ObsBlockID)>

<!ELEMENT AORList (AORID)*>
<!ELEMENT AORID (#PCDATA)>

<!ELEMENT PlannedDuration (#PCDATA)>
<!ELEMENT ExecutedDuration (#PCDATA)>

<!ELEMENT RequestedDuration (#PCDATA)>

<!ELEMENT Instrument (Config)>
<!ATTLIST Instrument name CDATA #IMPLIED >
<!ELEMENT Config (#PCDATA)>

<!ELEMENT ObsBlockName (#PCDATA)>
<!ELEMENT ObsBlockID (#PCDATA)>

<!ELEMENT ObservationCenter (Sidereal | NonSidereal)>

<!ELEMENT Sidereal ( RA, Dec, Equinox ) >
<!ELEMENT RA (#PCDATA)>
<!ELEMENT Dec (#PCDATA)>
<!ELEMENT Equinox (#PCDATA)>

<!ELEMENT NonSidereal ( NAIFID | OrbitalElements)>
<!ELEMENT NAIFID (#PCDATA)>
<!ELEMENT OrbitalElements (Epoch, PerihelionDistance,
Eccentricity,
                                Inclination, LittleOmega,
BigOmega, PerihelionDay)>
<!ELEMENT Epoch (#PCDATA)>
<!ELEMENT PerihelionDistance (#PCDATA)>
<!ELEMENT Eccentricity (#PCDATA)>
<!ELEMENT Inclination (#PCDATA)>
<!ELEMENT LittleOmega (#PCDATA)>
<!ELEMENT BigOmega (#PCDATA)>
<!ELEMENT PerihelionDay (#PCDATA)>
```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

- ObsBlockName, Instrument, and ObservationCenter – from one of AORs in the block (for now)
- Duration – the sum of all the AOR durations in seconds
- PercentExecuted and PercentPlanned – set to 0.0 for now. These are reserved for future use. (I think the FMI would like to be able to use PercentPlanned now)

11.4. ObsPlanObsBlockInfoList Description

		Class	Type
ObsPlanObsBlockInfoList	Retval	Element{1}	Text
	ObsBlockInfo	Element{0,N}	See Below
	Proposal	Element{Opt}	See Below

Retval: The status of the **getObsBlockInfoList** method. If using a web interface, a login is required before making the call. The values can be one of the following:

- **Permission Denied**
<Retval>Permission Denied.</Retval>
- **No Matching Record**
<Retval>No Matching Record Found.</Retval>
- **ObsPlan exists, but it does not have observation block**
<Retval>No Observation Block is found for ObsPlan 01_0072</Retval>
- **Success**
<Retval>SUCCESS</Retval>

ObsBlockInfo: Information about a specific ObsBlock. The ObsBlock is a group of AORs that have similar pointing.

Proposal: Information about a proposal.

		Class	Type
Proposal	TACGrade	Element{1}	Text
	Title	Element{1}	Text
	PropID	Element{1}	Text
	MustDo	Element{Opt}	EMPTY
	PIEmail	Element{1}	Text

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

	PILastName	Element{1}	Text
	PIFirstName	Element{1}	Text

TACGrade: TAC grading of this proposal, a numeric number between 0 and 5.

Title: Proposal Title.

PropID: Proposal ID.

MustDo: A Flag indicating this proposal must be done.

PIEmail, PILastName, PIFirstName, PI contact information.

		Class	Type
ObsBlockInfo	AORList	Element{1}	See Below
	PercentExecuted	Element{1}	Text
	PercentPlanned	Element{1}	Text
	Duration	Element{1}	Text
	ObservationCenter	Element{1}	See Below
	Instrument	Element{1}	See Below
	ObsBlockName	Element{1}	Text
	ObsBlockID	Element{1}	Text

AORList: List of AOR ID.

PercentExecuted: Percent of AORs executed.

PercentPlanned: Percent of AORs flight planned.

Duration: The duration of this AOR.

ObservationCenter: The target of the observation. It could be a fixed target (sidereal) or a moving target (non-sidereal).

Instrument: The instrument used for this observation.

ObsBlockName: Name of this ObsBlock.

ObsBlockID: ID of this Observation ObsBlock.

		Class	Type
AORList	AORID	Element{0,N}	Text

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

AORList contains of a list of AOR ID.

		Class	Type
Instrument	name	Attribute	Text
	Config	Element{1}	Text

name: Name of the instrument

Config: Instrument Configuration

		Class	Type
ObservationCenter	Sidereal NonSidereal		
	Sidereal	Element{1}	Text
	NonSidereal	Element{1}	Text

ObservationCenter can be either a Sidereal target or a NonSidereal target.

		Class	Type
Sidereal	RA	Element{1}	Text
	Dec	Element{1}	Text
	Equinox	Element{1}	Text

Sidereal target is defined by RA, Dec and Equinox.

		Class	Type
NonSidereal	NAIF_ID OrbitalElements		
	NAIF_ID	Element{1}	Text
	OrbitalElements	Element{1}	See Below

A NonSidereal target can be defined as either a NAIF_ID or an OrbitalElements.

		Class	Type
OrbitalElements	Epoch	Element{1}	Text
	PerihelionDistance	Element{1}	Text
	Eccentricity	Element{1}	Text
	Inclination	Element{1}	Text
	LittleOmega	Element{1}	Text
	BigOmega	Element{1}	Text
	PerihelionDay	Element{1}	Text

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

OrbitalElements is defined by 7 parameters:

Epoch: Epoch of the osculating elements, Julian date

PerihelionDistance: Perihelion distance (AU)

Eccentricity: The eccentricity of the orbit

Inclination: Inclination of orbit plane (DEG) with regard to J2000 ecliptic plane

LittleOmega: Argument of Perihelion (DEG) with regard to J2000 ecliptic/equinox. It is the angle (in the body's orbit plane) between the ascending node line and perihelion measured in the direction of the body's orbit. This angle is often denoted as lower-case omega (ω).

BigOmega: Longitude of Ascending Node (DEG) with regard to J2000 ecliptic/equinox. It is the angle between the reference X-direction (typically the vernal equinox) and the point at which the body passes up (north) through the reference plane. This angle is often denoted as capital omega (Ω).

PerihelionDay: Perihelion Julian Day Number

11.5. ObsPlanObsBlockInfoList Sample xml

- **Permission Denied**

```
<ObsPlanObsBlockInfoList>
  <Retval>Permission Denied.</Retval>
</ObsPlanObsBlockInfoList>
```
- **No Matching Record**

```
<ObsPlanObsBlockInfoList>
  <Retval>No Matching Record Found.</Retval>
</ObsPlanObsBlockInfoList>
```
- **ObsPlan exists, but it does not have observation block**

```
<ObsPlanObsBlockInfoList>
  <Proposal>
    <PropID>01_0072</PropID>
    <Title>[CII] in the Magellanic Clouds: sampling low-
metallicity ISM physics</Title>
    <TACGrade>4.7</TACGrade>
  </Proposal>
  <Retval>No Observation Block is found for ObsPlan
01_0072</Retval>
</ObsPlanObsBlockInfoList>
```
- **Success**

```
<ObsPlanObsBlockInfoList>
  <Proposal>
    <PropID>01_0051</PropID>
    <Title>Mapping [CII] emission in the NGC2023
```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```

reflection nebula.</Title>
  <TACGrade>3.9</TACGrade>
</Proposal>
<ObsBlockInfo>
  <ObsBlockID>OB_01_0051_01</ObsBlockID>
  <ObsBlockName>HD37903-1</ObsBlockName>
  <Instrument name="GREAT">
    <Config>DUAL-CHANNEL</Config>
  </Instrument>
  <ObservationCenter>
    <Sidereal>
      <RA>5.693997222222222</RA>
      <Dec>-2.259027777777778</Dec>
      <Equinox>2000.0</Equinox>
    </Sidereal>
  </ObservationCenter>
  <Duration>2637.864</Duration>
  <PercentPlanned>0.0</PercentPlanned>
  <PercentExecuted>0.0</PercentExecuted>
  <AORList>
    <AORID>01_0051_1</AORID>
    <AORID>01_0051_2</AORID>
  </AORList>
</ObsBlockInfo>
...
  <Retval>SUCCESS</Retval>
</ObsPlanObsBlockInfoList>

```

12. API TO UPLOAD A FLIGHT PLAN TO DCS

This section documents the message and protocol for the FMI to use to upload a flight plan to DCS using HTTPS protocol. There is no CORBA interface for uploading function.

12.1. Flight Plan XML Definition

12.1.1. Description

Flight Plan XML string containing the summary information of a flight plan. The schema of the Flight Plan XML is defined in FMI ICD (**SCI-US-ICD-SE03-2005**) thus is not part of the scope of this document. - (example should be in the FMI ICD, not here or will have to update this document whenever the FMI ICD updates).

12.2. Upload Flight plan to DCS with HTTPS web server protocol

This section describes the upload protocol for FMI to upload flight plan and its related data to the DCS via a web server service method.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

12.2.1. Upload Flight plan to DCS

To upload a flight plan to DCS, the FMI opens a secure HTTPS connection, and passes the flight plan xml string to the DCS, along with the flight plan image and a flight plan summary. The [DCSWEBHOST] is the host name of the DCS server where the flight plan will be uploaded to.

- URL:
`https://[DCSWEBHOST]/observationPlanning/DbProxy/UploadFlightPlanFiles.jsp`
- Input parameters
 - *fpFile* File – Individual flight plan .fp file
 - *xmlFile* File – Individual flight plan .xml
 - *imgFile* File – Individual flight plan image .png
 - *misFile* File – Individual flight plan summary .mis
 - *flightSeriesID* String
- Output:
 - String – Status returned by DCS, always starts with either "SUCCESS:" or "ERROR:"
- Permission: Only the following roles can upload a flight plan
 - TACLead
 - DCSAdmin
 - FPAdmin

13. API TO DOWNLOAD A FLIGHT PLAN FROM DCS

13.1. Retrieve a flight plan through CORBA interface

To retrieve a flight plan stored in the DCS, a CORBA interface is provided. There is no web service interface for this function. The CORBA object FMView is defined in Section “FlightManagment Module”.

13.1.1. Get all flight plan IDs for given flight series ID

Use this method to get a list of flight plan IDs associated with a flight series ID.

- Method definition:


```
typedef sequence<string> FlightPlanIDSeq;
PDRetVal getFlightPlanIDs (
    in string flightSeriesID,
    out FlightPlanIDSeq ids);
```

The flight plan ID list is returned inside the “out” parameter as an array of string.

- Return Flag
 - *PDRetVal.Success*
 - *PDRetVal.NotFound* -- *ids.value* = new String[0]
 - *PDRetVal.InternalError* -- *ids.value* = new String[0]

13.1.2. Retrieve a Flight Plan

To retrieve a single flight plan based on a flight plan ID, use the method

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE


```
typedef sequence<octet> FlightPlanBinary;  
PDRetVal getFlightPlanBinary (  
    in string flightPlanID,  
    out FlightPlanBinary fpdata );
```

The flight plan is returned inside the “out” parameter as an array of byte. The Flight Plan binary was uploaded to the DCS and stored in the database without any parsing, and is returned in its original form here.

- Return Flag
 - *PDRetVal.Success*
 - *PDRetVal.NotFound*
 - *PDRetVal.InternalError*

14. CONTRIBUTORS

The following people were original authors or contributed significantly to this document.

Dr. Ralph Shuping, Space Science Institute
Dr. Elizabeth Moore, USRA SOFIA
Bob Krzaczek, RIT Data Cycle System Software Engineer
Li Sun, USRA SOFIA
Tom Kilsdonk, USRA/SOFIA
Thomas Civeit
Melanie Clarke, USRA SOFIA

APPENDIX A. ACRONYMS

AOR	Astronomical Observation Request
API	Application Programming Interface
ARC	Ames Research Center
CCB	Configuration Control Board
CM	Configuration Management
CORBA	Common Object Request Broker Architecture
CS	Cycle Scheduler
CSCI	Computer Software Configuration Item
CVS	Concurrent Versioning System
DCS	Data Cycle System
FPE	Flight Plan Editor
FMI	Flight Management Infrastructure
GI	Guest Investigator
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
JSP	Java Server Page
NAIF_ID	Navigation and Ancillary Information Facility (NAIF) assigned ID for an object
ObsPlan	Observing Plan
ObsBlock	Observation Block
SPR	Software Problem Report
SMO	SOFIA Missions Operations
SSC	SOFIA Science Center
SSMO	SOFIA Science Missions Operations
SSMOC	SOFIA Science Missions Operations Center
STS	Short Term Scheduler
TAC	Time Allocation Committee
USRA	University Space Research Association
XML	Extensible Markup Language

APPENDIX B. SOFIA AOT TEMPLATE & EXAMPLES

To help the SOFIA scientists and the instrument teams define an AOT without the need to work in an XML file format, these tables will collect the similar information needed to define an AOT. The DCS SSPOT team will translate this table into the AOT XML file used by the DCS. For each of the observing mode, (or AOT), there is a set of parameters that need to be defined, either by the user (for stand alone parameters) or set because of another parameter is set by the user (dependency). FSI teams can use the template table below as a guide:

AOT Definition Template

Parameter	Preferred Name	FITS Keyword	Value(s) Default	Data Type	Units	Hi	Low	Stand Alone

Notes:

- Preferred parameter name is the label that is going to be displayed to the user. This is necessary because different instruments may call the same parameter differently. For example, chop throw for FORCAST may be called chop amplitude for another instrument.
- If a parameter is a stand alone parameter, it means it does not have dependencies on the other parameters, the user can enter any value within the limit. If a parameter is not a stand alone one, the user is not allowed to enter a value at will, the value of this parameter will be determined by another parameter. An example of how to define the dependencies between parameters is given below.
- Dependency definition is where the scientists and the instrument team should spend effort in. This will translate into “if this one is selected”, what else should be set according to this”. Be sure to spend some time to define this part clearly. The dependencies in the AOT_FORCAST_TPCD are very cleanly defined by Jim De Buizer and Ralph Shuping. It should serve as a template to define dependencies.
- Default and limits save user work and prevent unnecessary mistakes, so be sure to spend time defining these as well.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

APPENDIX C. SI KEYWORD DICTIONARY

Changes to Keyword Dictionary from Rev E to Rev F (September 24, 2015)

- Added Data Processing keywords DATAQUAL, N_SPEC, ASSC_AOR
- Added Instrument keyword DETCHAN, TOTINT
- Added Dither keywords DTHCRSYS, DTHXOFF, DTHYOFF
- Updated MCCS TELEL source to coord.pos.sibs.alt from coord.pos.sibs.el
- Updated Chopping/Nodding
 - Increased angle range to +/-360 from +/-180
 - Added nodding pattern BA to NODPATT
 - Added SIRF as one of the CHPCRSYS.

Changes to Keyword Dictionary from Rev D to Rev E (Feb 5, 2013)

- Added PIPELINE and PIPEVERS keywords to be populated by pipeline or other processing software.

Changes to Keyword Dictionary from Rev C to Rev D (10-12-2012)

- Included changes for C1 and C2:
 - updated MCCS locations for LASTREW and CHOPSYM
 - updated ALTI_STA/END to refer to **pressure** altitude and changed MCCS location.
 - reverted back to TEMPPRI1/2/3 and TEMPSEC1
 - Removed TEMPPRIC/L/R/F
- Reinstated DITHER keyword
- Changed DCSREV to FILEREV (no impact to SI teams)
- Added "FOCUS_LOOP" to possible values for OBSTYPE.
- Changed units in heterodyne area to MHz (J. Stutzki).
- Added value "UNKNOWN" for source type.
- Added relevant MCCS HK locations for some Nodding keywords. Updated values for NODCRSYS keyword to match MCCS values.
- Updated FLIGHTLG MCCS source to "fltexec.leg_data.leg_seq".
- Clarified TSC-STAT: MCCS location is "ta_state.tsc_status"

Notes:

- Files with DATASRC = "CALIBRATION" are made public immediately upon archive ingest.

Rules of “requirements”:

The 7 absolutely required keywords are the minimum set that will allow the data files to be archived in the DCS database (Section 8.1.1). Without these 7 keywords, the ingestion process will fail.

Beyond the above keywords, other required keywords are important for data processing (reduction) and provide values for searching.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Besides the absolutely required and required keywords, each instrument can provide instrument specific data in the FITS files as well. These keywords will be defined in the individual [Instrument]-DCS ICD thus is outside the scope of this document.

Recommended values for all level data products:

Observation keywords:

- AOR_ID – This is the only unique keyword that is carried from the proposal to observing plan to flight plan and data products. This keyword allows DCS to provide an accurate status on the AORs through out the planning and the only keyword that ties the data files that are in various processing state.

Mission Management Keywords:

- PLANID – This is the only keyword that can tie a data file to the owner of the data, e.g. the PI that submitted the proposals that generated these data. This keyword enables the archive search by PI names.

Recommended values for Level 2 and 3 data products:

Observation Keywords:

- OBS_ID – If there is a one-to-one mapping between the LEVEL 1 file and the LEVEL 2 file, then the original OBS_ID should be used with a “P_” pre-pended to indicate that the data has been processed. If there is **not** a one-to-one mapping, then an appropriate representative OBS_ID should be chosen from the input files and pre-pended with a “P_”.

Data Processing Keywords: Must be updated by the pipeline to appropriate values.

- PROCSTAT = ‘LEVEL_2’
- PRODTYPE: In some cases, SI pipelines might produce different products as part of Level 2 processing. These will be distinguished using the PRODTYPE keyword; possible values must be specified in the SI-DCS ICD.
- PIPELINE: Name of pipeline/processing software used to generate file (e.g. “FLITECAM Data Reduction Pipeline v1.0.0”)
- PIPEVERS: Set to full release tag for pipeline that produced the file (e.g. “FDRP_1_0_0_UT2013_4_1”)

Mission Keywords: Values can be carried forward from the LEVEL_1 input data.

Origination Keywords:

- ORIGIN: Set to organization that is operating the pipeline. For example, if the pipeline is being run in the SOFIA Science Center as part of SMO, then set to ‘SOFIA Science and Mission Ops’ (or similar). This will need to be configured in the deployed pipeline.
- OBSERVER: Carried from LEVEL_1 data.
- CREATOR: Name and version of the processing software: e.g. ‘FORCAST DRIP v1.0’.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

- OPERATOR: Carried from LEVEL_1 data.
- FILENAME: Name of host file.

Date and Time Keywords:

- DATE: date of file creation by pipeline
- DATE-OBS, UTCSTART/END: use earliest and latest values from LEVEL_1 input files.
-

Environmental Keywords:

- WVZ_STA/END: use earliest and latest values from LEVEL_1 input files.
- All others: pipeline should set to values for **earliest** input LEVEL_1 file.

Aircraft Keywords:

- All start/end keywords: use earliest and latest values from LEVEL_1 input files.
- All others: pipeline should set to values for **earliest** input LEVEL_1 file.

Telescope Keywords:

- TELESCOP/TELCONF: carried forward from LEVEL_1 input data.
- TELRA/TELDEC/TELVPA/TELEQUI: RA and Dec of the **central** pixel of the array, or center pixel of slit – computed from WCS. VPA to be calculated from WCS as well.
 - **EXCEPTION:** For some processed data (e.g. FORCAST NMC and NPC modes), it will not be possible to produce a meaningful WCS solution across the final image. In these cases, the TELRA and TELDEC should just be carried forward from a representative LEVEL 1 file.
- LASTREW: Value can be carried forward from the LEVEL_1 input data
- FOCUS_ST/EN: use earliest and latest values from LEVEL_1 input files.
- TELEX/XEL/LOS: pipeline should set to values for **earliest** input LEVEL_1 file.
- TSC-STAT/FBC-STAT: pipeline should set to values for **latest** input LEVEL_1 file.
- OBSRA/DEC/EQUINOX: Requested RA/Dec from AOR. Should be the same for all input files, and hence value can be carried forward into Level 2 files.
- ZA_START/END: use earliest and latest values from LEVEL_1 input files.
- TRACMODE: Value can be carried forward from the LEVEL_1 input data.
- TRACERR: If **all** LEVEL_1 files have TRACERR = F, then LEVEL_2 TRACERR = F; else TRACERR = T.

Data Collection Keywords: can be carried forward from the LEVEL_1 input data.

Instrument Keywords:

- EXPTIME: recalculated by the pipeline and set appropriately.
- All others: can be carried forward from the LEVEL_1 input data.

Array Detector Keywords:

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

- SIBS_X/Y: For data products with a one-to-one mapping with the original LEVEL_1 files, values can be carried forward from the LEVEL_1 data; otherwise, the keywords should be set to the default unknown value (Integers: -9999) or removed.
- WCS Keywords: Must be calculated and set by the pipeline.
- All others: can be carried forward from the LEVEL_1 input data.

Heterodyne Keywords: TBD, but will likely need to be updated by the pipeline.

Nodding/Chopping Keywords: can be carried forward from the LEVEL_1 input data.

Dithering Keywords:

- DTHINDEX: If no meaning for combined dataset, set to the default unknown value (Integers: -9999) or remove.
- All others: can be carried forward from the LEVEL_1 input data.

Mapping/Scanning Keywords: can be carried forward from the LEVEL_1 input data.

History Keywords: In addition to the usual keyword values, the following information must be provided in the HISTORY keywords at the end of the header:

- List of input OBS_IDs for the input LEVEL_1 in the input manifest. This gives the GI or archive user a quick way of seeing what files were used to generate the LEVEL_2 product.
- List of any auxiliary data products used.
- List of processing steps completed and any errors encountered.

(attach external PDF documents)

SOFIA Keywords Dictionary

[**Version:**REV F] [**Date:** 09-24-2015]

FITS Keywords Table

All FITS files submitted to the DCS for archiving must adhere to the FITS standard (v3.0, 2008 July 10)

WCS Keywords (see Array Detector Keywords section) should adhere to standard conventions (see http://fits.gsfc.nasa.gov/fits_wcs.html and <http://tdc-www.harvard.edu/wcstools/wcstools.fits.html> for discussion and references).

FITS Name: Keyword name - generally not the same as the abstract title.

Comment: Short description of keyword - suitable for FITS comment fields. Long descriptions can be found in the detailed descriptions. Comment text should include units as well.

HDU: header data unit - where the keyword can be used in the FITS file.

Representation : How the value of the keyword should be represented. In simple cases this may just be "string" or "float", but more complicated formats can be specified here (e.g. date and time)

Type: Specific FITS type - integer, float, string, or logical (boolean).

Units: Required units for keyword, if applicable.

Range: Possible keyword values, including enumerated types.

Example: Value example.

Is Required : Condition for which the keyword is required

-- If "Absolutely required", then the keywords are required for archive ingestion: If any of these is missing, the host file must be corrected and then re-ingested.

-- If it is "Yes", then the keywords are needed for data processing or archive search: If any of these is missing, the files can still be ingested into the archive, but it will impact pipeline and searching.

-- Conditionally required keywords are those that are only required if the stated condition applies, (e.g. DETCHAN is required if the instrument is FORCAST or FIFI-LS)

Source: Provider and location, if blank then data provided by SI. Known pre-defined aliases for some of the MCCS HK data items are included. We recommend the SI developers assign custom aliases to the others as well for ease of reference.

Missing Data Sources: If the HK datanode is not available, or returns NotFound (or any other error), then the following values should be used to populate the corresponding FITS keyword based on the FITS keyword datatype (float, int, str, bool): FLOAT = -9999.0; INT = -9999; STRING = UNKNOWN; BOOL = defined on keyword by keyword basis. For missing RA and Dec values, use the string representation with "UNKNOWN".

Observation Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
Datasource	DATASRC	Data Source	primary	[string]	[str]		enum [ASTRO, CALIBRATION, LAB, TEST, OTHER, FIRSTPOINT]	[ASTRO]	Yes (Absolutely)*	:
ObservationType	OBSTYPE	Observation type	any	[string]	[str]		enum [OBJECT, STANDARD_FLUX, STANDARD_TELLURIC, LAMP, FLAT, DARK, BIAS, SKY, BB, GASCELL, LASER, FOCUS_LOOP]	['OBJECT']	Yes	:
SourceType	SRCTYPE	Source type	any	[string]	[str]		enum [POINT_SOURCE, EXTENDED_SOURCE, OTHER, UNKNOWN]	['POINT_SOURCE']	Yes	:
KeywordDictionaryVersion	KWDICT	SOFIA Keyword dictionary version, DCS ICD rev.	primary	[string]	[str]		[]	['DCS_SI_01_A']	Yes	:
ObservationID	OBS_ID	SOFIA Observation Identification	any	[MMMMMM_[C]nnn]	[str]		[]	['2011-06-08_FO_F064B0187']	Yes (Absolutely)*	:

ImageID	IMAGEID	Image identification index	extension	[integer]	[int]			SCI-US-ICD-SE03-2023 Rev. F [1]	Multiple images in a single file for an observation, e.g. dithering, mapping, etc...	:
ObjectName	OBJECT	Object Name	any	[string]	[str]			['Orion Nebula (M42)']	Yes	:
AOTUniqueID <i>(Updated)</i>	AOT_ID	Astronomical Observation Template Identifier, most SI already use the correct format, clarified in revF	any	[string]	[str]			['FLITECAM_Imaging AOT SCI-US-ICD-SE03-2044, revB, AOT_FORCAST_GRISM SCI-US-ICD-SE03-2001 Rev. D']	If AOT in use.	:
AORUniqueID <i>(Updated)</i>	AOR_ID	Astronomical Observation Request Identifier. Clarified format in revF to be [PLANID]_n, n is an integer from 1 to 9999	any	[PLANID_n]	[str]			['02_0103_128']	If observation associated with DCS AOR	DCS : Observing Plan

Data Processing Related Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
ProcessingStatus <i>(Updated)</i>	PROCSTAT	Processing status, required for all SI starting revF	primary	[string]	[str]		enum [LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3, LEVEL_4]	['LEVEL_1']	Required for all SI, FLITECAM already has	SI Pipeline :
HeaderStatus <i>(Updated)</i>	HEADSTAT	Header status, added MODIFIED in revF	primary	[string]	[str]		enum [ORIGINAL, UNKNOWN, CORRECTED, ERROR, MODIFIED]	['ORIGINAL']	Yes	SI DCS :
DataQuality <i>(New)</i>	DATAQUAL	Result of data processing, replaced the old DataQualityAssessment	primary	[string]	[str]		enum [NOMINAL, USABLE, TEST, PROBLEM, FAIL]	['NOMINAL']	LEVEL 2/3/4 Product.	Pipeline :
NumberOfSpectral <i>(New)</i>	N_SPEC	Number of spectra included in file.	primary	[integer]	[int]				Yes	Pipeline :
AssociatedAORIDs <i>(New)</i>	ASSC_AOR	List of Astronomical Observation Request Identifiers.	any	[PLANID_n,PLANID_n]	[str]			['02_0103_1, 03_0098_127']	LEVEL 2/3/4 data, if product is associated with multiple DCS AORs	DCS : Observing Plan
PipelineName	PIPELINE	Pipeline/Processing Software	primary	[string]	[str]			['FDRP v1.0.0']	LEVEL 2/3/4 Product.	Pipeline :

PipelineVersion	PIPEVERS	Pipeline Version, full tag.	primary	[string]	[str]			[]	['FDRP_1_0_0_UT2013_4_1']	LEVEL 2/3/4 Product.	Pipeline :
ProductType	PRODTYPE	Product type.	primary	[string]	[str]			[]	['DRIP-COADEDED']	LEVEL 2/3/4 Product.	Pipeline :
DCSFileRevision	FILEREV	File revision identifier.	primary	[string]	[str]			[]	['r2']	If file changed in post-processing.	DCS only :

Mission Management Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
ObservingPlanUniqueID	PLANID	Observing plan identification	primary	[string]	[str]		[]	['81_0131']	If observation associated with an observing plan.	DCS : Observing Plan
AircraftDeployment	DEPLOY	Site deployment	primary	[string]	[str]		[]	['DAOFF']	Yes	SSC : Mission Plan
MissionID	MISSN-ID	Mission ID	primary	[string]	[str]		[]	[2011-06-08_FO_F06]	Yes (Absolutely)*	MCCS : session.user_environment.mission_id
FlightLeg (<i>Updated</i>)	FLIGHTLG	Flight leg, updated to fltexes_data from leg_data	primary	[integer]	[int]		[]	[4]	Yes	MCCS : fltexec.fltexec_data.leg_seq

Origination Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
Origin	ORIGIN	Origin of FITS file.	primary	[string]	[str]		[]	['FORCAST -- Cornell Univ.']	Yes	:
Observers	OBSERVER	Observer(s)	primary	[string]	[str]		[]	['Jane Astro']	Yes	:
FileCreator	CREATOR	File creation task (with version info).	primary	[string]	[str]		[]	['Mauna Kea IR']	Yes	:
TelescopeOperator	OPERATOR	Telescope operator	primary	[string]	[str]		[]	['Joe Astro']	Yes	SSC : Mission Plan
Filename	FILENAME	Name of host file	primary	[string]	[str]		[]	['040101_077_05FL001.fits']	Yes	:

Date and Time Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
CreationDate	DATE	Date of file creation	any	[yyyy-mm-ddThh:mm:ss[.sss]]	[str]		[]	['2004-01-01T13:45:45.2']	Yes	:
ObservationDate	DATE-OBS	UTC Date of exposure start	any	[yyyy-mm-ddThh:mm:ss[.sss]]	[str]		[]	['2004-01-01T12:11:10.5']	Yes (Absolutely)*	:
ObservationStartUTC	UTCSTART	UTC of exposure start	any	[hh:mm:ss.s]	[str]		[]	['09:30:01.00']	Yes	:
ObservationEndUTC	UTCEND	UTC of exposure end	any	[hh:mm:ss.s]	[str]		[]	['09:30:01.00']	Yes	:

Environmental Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
WaterVaporZenith_Start	WVZ_STA	Water vapor, integrated to zenith, observation start.	any	[general]	[flt]	microns	gt [0.0]	[1.503]	Yes	MCCS : wvm_if.wvmdata.water_vapor

WaterVaporZenith_End	WVZ_END	Water vapor, integrated to zenith, observation end.	any	[general]	[flt]	microns	gt [0.0]	[1.634]	Yes	MCCS : wvm_if.wvmdata.water_vapor
static_air_temp	TEMP_OUT	Static air temperature outside aircraft. (For Early Science: Value=NotSet)	any	[general]	[flt]	C	[]	[-10.5]	Yes	MCCS : das.ic1080_2hz.static_air_temp \$static_air_temp
PrimaryMirrorTemperature_1	TEMPPRI1	Temperature of primary mirror	any	[general]	[flt]	C	gt [-273.0]	[-10.5]	Yes	MCCS : ta_mcp.mcp_hk_pms.pms_temp_1
PrimaryMirrorTemperature_2	TEMPPRI2	Temperature of primary mirror	any	[general]	[flt]	C	gt [-273.0]	[-10.5]	Yes	MCCS : ta_mcp.mcp_hk_pms.pms_temp_2
PrimaryMirrorTemperature_3	TEMPPRI3	Temperature of primary mirror	any	[general]	[flt]	C	gt [-273.0]	[-10.5]	Yes	MCCS : ta_mcp.mcp_hk_pms.pms_temp_3
SecondaryMirrorTemperature_1	TEMPSEC1	Temperature of secondary	any	[general]	[flt]	C	[]	[-15.2]	Yes	MCCS : ta_mcp.mcp_hk_pms.sma_temp_1

Aircraft Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
Altitude_Start	ALTL_STA	Aircraft pressure altitude, start of observation.	any	[general]	[flt]	Feet	[]	[35229]	Yes	MCCS : das.ic1080_15hz.press_alt
Altitude_End	ALTL_END	Aircraft pressure altitude, end of observation.	any	[general]	[flt]	Feet	[]	[35128]	Yes	MCCS : das.ic1080_15hz.press_alt
Airspeed	AIRSPEED	True aircraft airspeed.	any	[general]	[flt]	Knots	[]	[375.0]	Yes	MCCS : das.ic1080_10hz.true_airspeed \$true_airspeed
GroundSpeed	GRDSPEED	Aircraft ground speed.	any	[general]	[flt]	Knots	[]	[350.0]	Yes	MCCS : das.ic1080_2hz.ground_speed \$ground_speed
Latitude_Start	LAT_STA	Aircraft latitude, start of observation.	any	[general]	[flt]	Degrees	[]	[35.2567]	Yes	MCCS : das.ic1080_2hz.lat_fms_1 \$latitude
Longitude_Start	LON_STA	Aircraft longitude, start of observation.	any	[general]	[flt]	Degrees	[]	[35.2567]	Yes	MCCS : das.ic1080_2hz.lon_fms_1 \$longitude
Latitude_End	LAT_END	Aircraft latitude, end of observation.	any	[general]	[flt]	Degrees	[]	[35.2567]	Yes	MCCS : das.ic1080_2hz.lat_fms_1 \$latitude
Longitude_End	LON_END	Aircraft longitude, end of observation.	any	[general]	[flt]	Degrees	[]	[35.2567]	Yes	MCCS : das.ic1080_2hz.lon_fms_1 \$longitude
Heading	HEADING	Aircraft true heading.	any	[general]	[flt]	Degrees	[]	[10.7892]	Yes	MCCS : das.ic1080_2hz.true_heading \$heading
TrackAngle	TRACKANG	Aircraft track angle.	any	[general]	[flt]	Degrees	[]	[10.7892]	Yes	MCCS : das.ic1080_2hz.true_track_angle \$track_angle

Telescope Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
Telescope	TELESCOP	Telescope name	primary	[string]	[str]		[]	['SOFIA 2.5m']	Yes	:
TelescopeConfig	TELCONF	Telescope configuration	primary	[string]	[str]		[]	[]	Yes	SSC : Mission Plan
TelescopeRA	TELRA	SI Bore sight RA (J2000) - as returned by MCCS, , removed type str in revF	any	[general] [hh:mm:ss.s]	[flt]	Hours	interval [0,24]	[9.023456] [5:35:17.3]	Yes	MCCS : coord.pos.sibs.ra
TelescopeDec	TELDEC	SI Bore sight Dec (J2000) - as returned by MCCS, removed type str in revF	any	[general] [dd:mm:ss]	[flt]	decimal degrees	interval [-90,90]	[47.345789] [-5:23:28]	Yes	MCCS : coord.pos.sibs.dec

TelescopeVPA	TELVPA	SI Bore sight VPA (ICRS J2000) - as returned by MCCS	any	[general]	[flt]	decimal degrees	interval [0,360]	[255.05]	Yes	MCCS : coord.pos.sibs.vpa
TelescopeEquinox	TELEQUI	Equinox of ERF coords (RA/Dec/VPA).	any	[general]	[str]		[]	[J2000]	Yes	MCCS : coord.pos.sibs.equinox
LastRewindUTC	LASTREW	Time of last rewind (UTC).	any	[yyyy-mm-ddThh:mm:ss.sssZ]	[str]		[]	['2012-09-13T21:30:28.317Z']	Yes	MCCS : ta_pos.time_of_last_rewind
TelescopeFocus_Start	FOCUS_ST	Telescope focus - SMA FCM t position (microns), observation start.	any	[general]	[flt]	microns	interval [-5000,5000]	[1245]	Yes	MCCS : ta_scs.fcm_status.fcm_act_t
TelescopeFocus_End	FOCUS_EN	Telescope focus - SMA FCM t position (microns), observation end.	any	[general]	[flt]	microns	interval [-5000,5000]	[1322]	Yes	MCCS : ta_scs.fcm_status.fcm_act_t
TelescopeElevation (Updated)	TELEL	Telescope elevation above the horizon at observation start - as returned by MCCS. Changed source from coord.pos.sibs.el to coord.pos.sibs.alt in revF.	any	[general]	[flt]	decimal degrees	interval [0,90]	[47.345789]	Yes	MCCS : coord.pos.sibs.alt
TelescopeCrossElevation	TELXEL	Telescope cross elevation at observation start - as returned by MCCS	any	[general]	[flt]	decimal degrees	interval [-90,90]	[0.543]	Yes	MCCS : coord.pos.sibs.xel
TelescopeLineOfSight	TELLOS	Telescope LOS at observation start - as returned by MCCS	any	[general]	[flt]	decimal degrees	interval [-180,180]	[0.543]	Yes	MCCS : coord.pos.sibs.los
TascuStatus	TSC-STAT	TASCU Status at observation end.	any	[string (TBC)]	[str]		[]	[STAB_INERTIAL_ONGOING]	Yes	MCCS : ta_state.tsc_status
TascuFBCStatus	FBC-STAT	FBC Status at observation end.	any	[string]	[str]		enum [FBC_OFF (1), FBC_QS (2), FBC_DY (3), FBC_ON (4)]	[FBC_ON]	Yes	MCCS : ta_tsc.tsc_mcs_hk.fbc_status
ObservationRequestRA	OBSRA	RA - requested	any	[general] [hh:mm:ss.s]	[flt] [str]	Hours	interval [0,24]	[9.0230] [5:35:17.3]	Yes	DCS : AOR
ObservationRequestDec	OBSDEC	Dec - requested	any	[general] [dd:mm:ss]	[flt] [str]	Degrees	interval [-90,90]	[47.3465] [-5:23:28]	Yes	DCS : AOR
ObservationRequestEquinox	EQUINOX	Coordinate equinox for OBSRA and	any	[general]	[flt]	yr	[]	[2000.0]	Yes	DCS : AOR

		OBSDEC								
ZenithAngle_Start	ZA_START	Telescope zenith angle, start of observation.	any	[general]	[flt]	decimal degrees	interval [0,90]	[42.359]	Yes	MCCS : (90 - coord.pos.sibs.alt)
ZenithAngle_End	ZA_END	Telescope zenith angle, end of observation.	any	[general]	[flt]	decimal degrees	interval [0,90]	[39.285]	Yes	MCCS : (90 - coord.pos.sibs.alt)
TrackingMode	TRACMODE	SOFIA tracking mode	any	[string]	[str]		enum [OFF, CENTROID, ROF, LIMB, OFFSET,ROF+OFFSET, CENTROID+INERTIAL, ROF+INERTIAL, OFFSET+INERTIAL, ROF+OFFSET+INERTIAL]	['CENTROID']	Tracking	MCCS : ta_pos.track.state
TrackingError	TRACERR	Tracking error flag.	any	[boolean]	[log]		[]	[F]	Tracking	: ta_trc.trc_status_table.main_op_mode_id

Data Collection Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
isChopping	CHOPPING	Chopping flag	primary	[boolean]	[log]		[]	[T]	If mode in use.	:
isNodding	NODDING	Nodding flag	primary	[boolean]	[log]		[]	[T]	If mode in use.	:
isDithering	DITHER	Dithering flag	primary	[boolean]	[log]		[]	[T]	If mode in use.	:
isMapping	MAPPING	Mapping flag.	primary	[boolean]	[log]		[]	[T]	If mode in use.	:
isScanning	SCANNING	Scanning flag.	primary	[boolean]	[log]		[]	[T]	If mode in use.	:

Annotation Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
-----------	--------------	---------	-----	----------------	------	-------	-------	---------	-------------	--------

Instrument Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
Instrument	INSTRUME	Instrument	primary	[string]	[str]		[]	['FLITECAM']	Yes (Absolutely)*	:
Datatype	DATATYPE	Data type	primary	[string]	[str]		enum [IMAGE, SPECTRAL, OTHER]	[IMAGE]	Yes	:
InstrumentConfiguration	INSTCFG	Instrument configuration	primary	[string]	[str]		[]	['IMAGING']	Yes	:
InstrumentMode	INSTMODE	Instrument observing mode	primary	[string]	[str]		[]	['C2N']	Yes	:
MCCSMode	MCCSMODE	MCCS SI Mode (MCCS_SI_04).	primary	[string]	[str]		[]	['flitecam_imaging']	Yes	MCCS :

										(instrument).si_config.current_mode
ExposureTime	EXPTIME	On-source exposure time	any	[general]	[flt]	s	[]	[600]	Yes	:
SpectralElement1 (Updated)	SPECTEL1	First spectral element in use -- Clarified in revF, set to "NONE" if not in use.	any	[string]	[str]		[]	['FLT_J']	Yes (Absolutely)*	:
SpectralElement2	SPECTEL2	Second spectral element in use -- Set to "NONE" if not in use.	any	[string]	[str]		[]	['FOR_XG063']	Yes (Absolutely)*	:
InstrumentSlit	SLIT	Instrument slit in use.	any	[string]	[str]		[]	['FOR_SS24']	Spectroscopy configs: if slit in use.	:
WavelengthCentral	WAVECENT	Central wavelength of observation.	primary	[general]	[flt]	Microns	[]	[2.2]	Imaging modes only.	:
Resolution	RESOLUN	Spectral resolution of observation.	primary	[general]	[flt]		[]	[2500]	Spectroscopy modes only.	:
DetectorChannel (New)	DETCAN	The values are FORCAST: SW LW , FIFI-LS: BLUE RED	primary	[string]	[str]		enum [SW, LW, BLUE, RED]	[LW]	FORCAST and FIFI-LS should populate these.	:
TotalIntegrationTime (New)	TOTINT	Total integration time (s), for FORCAST, If SKYMODE is C2NC2 or NXCAC, then TOTINT = DETTIME * 0.5, otherwise TOTINT = DETTIME*2.0	any	[float]	[flt]	s	[]	[600]	FORCAST	:

Array Detector Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
Detector	DETECTOR	Detector name	primary	[string]	[str]		[]	['SBRC InSb']	Array instruments only.	:
DetectorSize	DETSIZE	Detector size	primary	[vector]	[str]		[]	['(1024,1024)']	Array instruments only.	:
PixelScale	PIXSCAL	Pixel scale	primary	[general]	[flt]	arcsec	[]	[0.32]	Array instruments only.	:
Subarrays	SUBARRNO	Number of subarrays used	primary	[integer]	[int]		[]	[2]	If subarrays in use.	:
SubarraySize	SUBARR%2d	Sub array size	primary	[section]	[str]		[]	['[0:255,0:255]']	If subarrays in use.	:
ScienceInstrumentBoresightX (Updated)	SIBS_X	SI Boresight (x) - as returned by MCCS, typo in name fixed	any	[float]	[flt]		[]	[255]	Array instruments only.	MCCS : coord.pos.sibs.xsi
ScienceInstrumentBoresightY (Updated)	SIBS_Y	SI Boresight (y) - as returned by MCCS, typo in name fixed	any	[float]	[flt]		[]	[255]	Array instruments only.	MCCS : coord.pos.sibs.ysi
WCS_CTYPE	CTYPEn	Axis type (8 characters)	any	[string]	[str]		[]	[RA--TAN]	Imaging only.	:

WCS_CRPIX <i>(Updated)</i>	CRPIXn	Array location of the reference point in pixels for the n-th axis. Changed from int to float in revF	any	[float]	[flt]		[]	[511]	Array instruments only.	:
WCS_CRVAL	CRVALn	Coordinate value at reference point for the n-th axis.	any	[flt]	[flt]		[]	[82.345690]	Imaging only.	:
WCS_CDELT	CDELTn	Plate scale for the n-th axis at reference point (deg/pixel).	any	[flt]	[flt]		[]	[1.3852E-4]	Imaging only.	:
WCS_CROTA2	CROTA2	Rotation of axes in degrees.	any	[flt]	[flt]		[]	[113.45]	Imaging only.	:
WCS_RotMatrix	CDi_j	WCS Rotation Matrix elements.	any	[flt]	[flt]		[]	[1.3852E-4]	Imaging only.	:

Heterodyne Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
FrontendDevice	FRONTEND	Name of frontend device.	primary	[string]	[str]		[]	['GREAT LOW 1']	Heterodyne instruments only.	:
BackendDevice	BACKEND	Name of backend device.	primary	[string]	[str]		[]	['AOS']	Heterodyne instruments only.	:
BackendBandwidth	BANDWID	Total bandwidth of heterodyne backend	primary	[general]	[flt]	MHz	[]	[300.547]	Heterodyne instruments only.	:
SystemTemperature	TSYS	System temperature.	primary	[general]	[flt]	K	[]	[10.5]	Heterodyne instruments only.	:
FrequencyResolution	FREQRES	Nominal frequency resolution -- may differ from channel spacing.	primary	[general]	[flt]	MHz	[]	[100.0]	Heterodyne instruments only.	:
ReferenceFrequency	OBSFREQ	Reference frequency.	primary	[general]	[flt]	MHz	[]	[]	Heterodyne instruments only.	:
SidebandFrequency	IMAGFREQ	Image sideband frequency.	primary	[general]	[flt]	MHz	[]	[]	Heterodyne instruments only.	:
RestFrequency	RESTFREQ	Rest frequency.	primary	[general]	[flt]	MHz	[]	[]	Heterodyne instruments only.	:
VelocityDefinition	VELDEF	Velocity Definition	primary	[string]	[str]		[]	[RADI-LSR]	Heterodyne instruments only.	:
VelocityFrame	VFRAME	Radial velocity of reference frame.	primary	[general]	[flt]	km/s	[]	[250.3]	Heterodyne instruments only.	:
RadialVelocity	RVSYS	Radial velocity.	primary	[general]	[flt]	km/s	[]	[22.3]	Heterodyne instruments only.	:

Data Structure Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
-----------	--------------	---------	-----	----------------	------	-------	-------	---------	-------------	--------

Chopping Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
ChopFrequency	CHPFREQ	Chop frequency	any	[general]	[flt]	MHz	interval [0.0, 20.0]	[15.0]	Chopping	MCCS : sma.chop.frequency
ChopProfile	CHPPROF	Chopping profile: 2 or 3 point	any	[string]	[str]		enum [2-POINT, 3-POINT]	['2-POINT']	Chopping	MCCS : sma.chop.profile
ChopSymmetry	CHPSYM	Chopping symmetry: symmetric or asymmetric	any	[string]	[str]		[]	['on-axis']	Chopping	MCCS : sma.chop_symmetry

ChopAmplitude_1	CHPAMP1	Chop amplitude 1	any	[general]	[flt]	arcsec	interval [-1125.0, 1125.0]	[300.0]	Chopping	MCCS : sma.sky_amplitude
ChopAmplitude_2	CHPAMP2	Chop amplitude 2	any	[general]	[flt]	arcsec	interval [-1125.0, 1125.0]	[300.0]	Chopping	MCCS : sma.sky_amp2
ChopCoordSys (Updated)	CHPCRSYS	MCCS Coordinate system for sky tip, tilt, and angle. Added SIRF in revF	any	[string]	[str]		enum [TARF, ERF, SIRF]	[ERF]	Chopping	MCCS : sma.sky_coord_sys
ChopAngle (Updated)	CHPANGLE	Calculated angle in the sky_coord_sys reference frame. Range increased in revF from -/+180 to -/+360	any	[general]	[flt]	decimal degrees	interval [-360,360]	[45.0]	Chopping	MCCS : sma.sky_angle
ChopTip	CHPTIP	Calculated tip in the sky_coord_sys reference frame.	any	[general]	[flt]	arcsec	interval [-301,301]	[15]	Chopping	MCCS : sma.sky_tip
ChopTilt	CHPTILT	Calculated tilt in the sky_coord_sys reference frame.	any	[general]	[flt]	arcsec	interval [-301,301]	[10]	Chopping	MCCS : sma.sky_tilt
ChopPhase	CHPPHASE	Chop phase	any	[int]	[int]	millisec	interval [0,1000]	[200]	Chopping	MCCS : sma.chop.phase

Nodding Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
NodDwellTime	NODTIME	Nod time	any	[general]	[flt]	s	[]	[300.0]	Nodding	:
NodCycles	NODN	Nod cycles.	any	[integer]	[int]		[]	[30]	Nodding	:
NodSettleTime	NODSETL	Nod settle time	any	[general]	[flt]	s	[]	[0.0556]	Nodding	:
NodAmplitude	NODAMP	Nod amplitude on sky.	any	[general]	[flt]	arcsec	[]	[30.0]	Nodding	MCCS : (e.g. nod.amplitude)
NodBeam	NODBEAM	Current nod beam position	any	[string]	[str]		[]	['A']	Nodding	MCCS : (e.g. nod.current)
NodPattern (Updated)	NODPATT	Nodding pattern, one cycle. Added BA in revF	any	[string]	[str]		enum [ABBA, AB, BA, CUSTOM]	['ABBA']	Nodding	:
NodStyle	NODSTYLE	Chop/nod Style.	any	[string]	[str]		enum [NMC, NPC, C2NC2, CUSTOM]	['NPC']	Nodding/Chopping	:
NodCoordSys	NODCRSYS	Coordinate system for Nod angle	any	[string]	[str]		enum [ERF, ECRF, GALRF, TARF, FPIRF, FFIRF, WFIRF, SIRF, USER]	['ERF']	Nodding	MCCS : (e.g. nod.coord_sys)
NodAngle (Updated)	NODANGLE	Nod angle, range increased in revF from - /+180 to -/+360	any	[general]	[flt]	decimal degrees	interval [-360,360]	[45.0]	Nodding	MCCS : (e.g. nod.pos_angle)

Dithering Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
DitherCoordiante (New)	DTHCRSYS	Dither coordinate, needed by DPS for FORCAST, FLITECAM	any	[str]	[str]		enum [SIRF,TARF,ERF]	[SIRF]	Dithering	SI DCS, not from MCCS :
DitherXOffset (New)	DTHXOFF	Dither offset in X axis (arcseconds).	any	[float]	[flt]	arcsec	[]	[2.5]	Dithering	:
DitherYOffset (New)	DTHYOFF	Dither offset in Y axis (arcseconds).	any	[float]	[flt]	arcsec	[]	[2.5]	Dithering	:
DitherPattern	DTHPATT	Dither pattern, added NONE in revF	any	[string]	[str]		enum [NONE, 3-POINT, 5-POINT, 9-POINT,	['9-	Dithering	:

<i>(Updated)</i>								CUSTOM]	POINT']		
DitherPositions	DTHNPOS	Number of dither positions.	any	[integer]	[int]			[]	[9]	Dithering	:
DitherPositionIndex	DTHINDEX	Dither position index.	any	[integer]	[int]			[]	[5]	Dithering	:
DitherOffset	DTHOFFS	Dither offset (arcseconds).	any	[float]	[flt]	arcsec		[]	[2.5]	Dithering	:

Mapping Keywords

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
MapCoordSys	MAPCRSYS	Coordinate system for mapping/scanning.	any	[string]	[str]		enum [EQUATORIAL, GALACTIC, ECLIPTIC, USER]	['GALACTIC']	Mapping	:
MapPositionsX	MAPNXPOS	Number of map positions in X	any	[integer]	[int]		[]	[4]	Mapping	:
MapPositionsY	MAPNYPOS	Number of map positions in Y	any	[integer]	[int]		[]	[4]	Mapping	:
MapIntervalX	MAPINTX	Mapping step interval in X	primary	[general]	[flt]	arcmin	[]	[8.5]	Mapping	:
MapIntervalY	MAPINTY	Mapping step interval in Y	any	[general]	[flt]	arcmin	[]	[8.5]	Mapping	:

Scanning Keywords (Constant Velocity)

Parameter	FITS Keyword	Comment	HDU	Representation	Type	Units	Range	Example	Is Required	Source
ScanStartRA	SCNRA0	Start of scan - RA.	any	[general] [hh:mm:ss.s]	[flt] [str]	Hours	interval [0,24]	[9.0230] [5:35:17.3]	Scanning	:
ScanStartDec	SCNDEC0	Start of scan - Dec.	any	[general] [dd:mm:ss]	[flt] [str]	Degrees	interval [-90,90]	[47.3465] [-5:23:28]	Scanning	:
ScanEndRA	SCNRAF	End of scan - RA.	any	[general] [hh:mm:ss.s]	[flt] [str]	Hours	interval [0,24]	[9.0305] [5:40:32.5]	Scanning	:
ScanEndDec	SCNDECf	End of scan - Dec.	any	[general] [dd:mm:ss]	[flt] [str]	Degrees	interval [-90,90]	[47.3465] [-5:23:28]	Scanning	:
ScanRate	SCNRATE	Scan rate	any	[general]	[flt]	arcsec/s	[]	[10.0]	Scanning	:
ScanDirection	SCNDIR	Scan direction	any	[general]	[flt]	degrees	[]	[-35.5]	Scanning	:

Keyword Descriptions revF September 2015

Observation Keywords

Datasource

FITS Name:DATASRC

FITS Type:str

Description: Overall source/type of data: ASTRO = Astronomical observation; CALIBRATION = calibration data; LAB = Laboratory data; TEST = Test data; FIRSTPOINT = SOFIA first pointing observation; OTHER = Any other source not listed here. If datasource is set to "CALIBRATION", the host file will be made public immediately upon ingestion.

Requirement:Absolutely Required for Archive Ingestion*

ObservationType

FITS Name:OBSTYPE

FITS Type:str

Description: The type of observation such as an astronomical exposure or a particular type of calibration exposure: OBJECT = astronomical object; STANDARD = Astronomical flux standard for calibration; LAMP = Calibration lamp; FLAT = Flat-field exposure for calibration; DARK = Dark current exposure for calibration.

Requirement: Required

SourceType

FITS Name:SRCTYPE

FITS Type:str

Description:Source type. Maps to AOR SourceType. Needed for reduction of slit spectra, but useful for other data types as well.

Requirement: Required

KeywordDictionaryVersion

FITS Name:KWDICT

FITS Type:str

Description: SOFIA keyword dictionary version. Since the keyword dictionary is now a part of the DCS ICD (DCS_SI_01, this keyword should reference which version of the ICD is being used. The document rev (A, B, C, etc...) should be appended to the string 'DCS_SI_01' with an underscore (see example).

Requirement: Required

ObservationID

FITS Name:OBS_ID

FITS Type:str

Description:The unique identifier for any given data file, meant to provide ready timing and sequencing information at a glance to the user. It is created at the same time that the data file is first written, in a format common to all instruments (and thus mandated by the observatory). The OBS_ID is assigned to saved datasets only. Some test data may be taken during a flight for immediate diagnostic purposes, but not saved (e.g., a snapshot to verify pointing), in which case there would be no point in assigning an OBS_ID, but we note that the default should almost always be to save the data and thus assign this keyword. The OBS_ID number is distinct from the AOR_ID. The AOR_ID is the unique identifier for an AOR which can, in principle, generate multiple files to be saved. Each of these files would have a unique OBS_ID. The mapping from AOR_ID to OBS_ID is one-to-many, i.e., one AOR can produce multiple data files. Note that the converse is not true -- one data file cannot be the product of multiple AORs. We suggest the following as a format for the OBS_ID keyword: MMMMMM_[C]nnn. Where: MMM = Mission ID (see MissionID) -- Data not taken as part of a SOFIA mission should set MMM to current date or some other suitable value; C = Channel identifier (optional) -- preferably a letter (B for Blue, S for short, etc...); nnn = Observation sequence number -- The sequence should be reset at mission start and advanced all the way through mission close-out (as defined by the MCCS), which should allow for capture of pre- and post-flight data -- This field will have a minimum of three digits (typical expected length), but more digits will be allowed as necessary.

Requirement:Absolutely Required for Archive Ingestion*

ImageID

FITS Name:IMAGEID

FITS Type:int

Description:The image identification when there are multiple images for an observation within the SAME file (e.g. images stored as FITS extensions).

Requirement:Multiple images in a single file for an observation, e.g. dithering, mapping, etc...

ObjectName**FITS Name:**OBJECT**FITS Type:**str**Description:** The object name as given by the observer, or as specified by the flight plan.**Requirement:** Required**AOTUniqueID -- Updated!****FITS Name:**AOT_ID**FITS Type:**str**Description:** Unique Astronomical Observation Template (AOT) identifier as defined in SI-DCS ICD. AORs generated from SOFIA SSpot are based on the [SI]-DCS ICD which defines AOTs, the combination of the AOT name, ICD document number, and the ICD version number can uniquely identify an AOT version. Note that the ICD version number is important, AOTs in different versions of ICD can sometimes have conflict values.**Requirement:**If AOT in use.**AORUniqueID -- Updated!****FITS Name:**AOR_ID**FITS Type:**str**Description:** Unique Astronomical Observation Request (AOR) identifier. The AOR_ID is assigned during the creation of the final version of an AOR to be used in flight planning. The value of this keyword should be equivalent to the AOR/ID tag in the AOR document. During Early Science, the AOR ID will be equivalent to the proposed observation ID and will be documented in the observing plan. The AOR_ID should not be reused once data is taken for this AOR to define a different observation. For example, 90_0004_1 has been observed for SA 114-656, it should not be reused to observe SA 114-670 even if all other parameters are exactly the same.**Requirement:**If observation associated with DCS AOR

Data Processing Related Keywords

ProcessingStatus -- Updated!**FITS Name:**PROCSTAT**FITS Type:**str**Description:** Status of any processing applied to the data, as defined in the SOFIA Project Data Management Plan: LEVEL_0 = Raw engineering data for diagnostic purposes, generally not intended for archiving; LEVEL_1 = Raw, uncalibrated science data in FITS or SDFITS format with complete header adhering to the SOFIA Keyword Dictionary; LEVEL_2 = Processed science data corrected for instrument artifacts; LEVEL_3 = Flux-calibrated science data (e.g. BUNIT keyword takes image to MJy/sr); LEVEL_4 = Higher order products.**Requirement:**Required for all SI, FLITECAM already has**HeaderStatus -- Updated!****FITS Name:**HEADSTAT**FITS Type:**str**Description:** Status of FITS header data (updated during post-processing). ORIGINAL: Header values are from original raw data file and have not been modified. UNKNOWN: header values have not been reviewed/verified. CORRECTED: header values have been corrected as part of post-processing, see HISTORY records for details. ERROR: There is a problem with the header values that has not been fixed, see COMMENT records for details. SI would set HEADSTAT = 'ORIGINAL' for raw data acquired on the AC. MODIFIED: Headers have been changed but are not yet fully CORRECTED.**Requirement:** Required**DataQuality -- New!****FITS Name:**DATAQUAL**FITS Type:**str**Description:** Indicates overall data quality; indicator of scientific reliability for the dataset. Values are: NOMINAL: no outstanding issues with processing/calibration/observing conditions. USABLE: minor issue(s) with processing/calibration/conditions but should still be scientifically valid (perhaps with larger than usual uncertainties); see HISTORY records for details. PROBLEM: significant issue(s) encountered with processing, calibration, or observing conditions; may not be scientifically useful (depending on application); see HISTORY records for details. In general, these cases are addressed through manual re-processing before archiving and distribution. FAIL: data could not be processed successfully for some reason. These cases are rare and generally not archived or distributed to the GI.**Requirement:**LEVEL 2/3/4 Product.

NumberOfSpectral -- New!**FITS Name:**N_SPEC**FITS Type:**int**Description:** Number of spectral, optional for spectroscopic modes.**Requirement:** Required**AssociatedAORIDs -- New!****FITS Name:**ASSC_AOR**FITS Type:**str**Description:**List of all unique input AOR_IDs used in generating a combined output file.**Requirement:**LEVEL 2/3/4 data, if product is associated with multiple DCS AORs**PipelineName****FITS Name:**PIPELINE**FITS Type:**str**Description:**Name of pipeline/processing software used to generate file (e.g. "FDRP v1.0.0"). LEVEL 2/3/4 data only.**Requirement:**LEVEL 2/3/4 Product.**PipelineVersion****FITS Name:**PIPEVERS**FITS Type:**str**Description:**Full release tag for pipeline that produced the file (e.g. "FDRP_1_0_0_UT2013_4_1"). LEVEL 2/3/4 data only.**Requirement:**LEVEL 2/3/4 Product.**ProductType****FITS Name:**PRODTYPE**FITS Type:**str**Description:**Type of product produced by the processing software or pipeline, as defined in SI-DCS ICD. Should be simple identifier that the GI can use to look up in the processing handbook or ICD. ID should include an identifier for the software that was used to produce the file (e.g. "DRIP_XXX"). For example, the FORCAST product types are: DRIP-UNDISTORTED, DRIP-MERGED, DRIP-COADEDED, DRIP-REDALL. Generally, the SI will not need to set this keyword for raw data.**Requirement:**LEVEL 2/3/4 Product.**DCSFileRevision****FITS Name:**FILEREV**FITS Type:**str**Description:** File revision identifier, to be inserted by archive ingestion tasks if file was modified as part of post-processing. Change details to be documented using HISTORY records at the end of the header. String value, typically a lower case 'r' (ASCII 114) immediately followed by a integer greater than zero in decimal format with no padding or leading zeros. E.g., "r1" not "r01". The SI will not need to set this keyword for raw data; included here for completeness.**Requirement:**If file changed in post-processing.

Mission Management Keywords

ObservingPlanUniqueID**FITS Name:**PLANID**FITS Type:**str**Description:** The observing plan which contains all the AORs. The value of this keyword should be equivalent to <AOR/Reference/ObservingPlan> in the AOR document.**Requirement:**If observation associated with an observing plan.**AircraftDeployment****FITS Name:**DEPLOY**FITS Type:**str**Description:** Aircraft base of operations for current instrument run.**Requirement:** Required**MissionID****FITS Name:**MISSN-ID

FITS Type:str

Description: Unique mission identifier, as specified in the Mission Plan and returned by the MCCA. Current spec for mission ID includes a date stamp, instrument ID, and mission type indicators.

Requirement: Absolutely Required for Archive Ingestion*

FlightLeg -- *Updated!*

FITS Name:FLIGHTLG

FITS Type:int

Description: Flight leg identifier for given mission ID.

Requirement: Required

Origination Keywords

Origin

FITS Name:ORIGIN

FITS Type:str

Description: Organization or institution responsible for creation of FITS file.

Requirement: Required

Observers

FITS Name:OBSERVER

FITS Type:str

Description: Observer name(s).

Requirement: Required

FileCreator

FITS Name:CREATOR

FITS Type:str

Description: Software task which wrote the FITS file (including version information).

Requirement: Required

TelescopeOperator

FITS Name:OPERATOR

FITS Type:str

Description: The telescope operator for the mission. Can be obtained from list of active MCCA sessions (get list=active_session) if needed.

Requirement: Required

Filename

FITS Name:FILENAME

FITS Type:str

Description: Name of host file. The FILENAME keyword allows for different stages in the treatment of a dataset: raw, calibrated, custom reduced, pipelined, or reduced at an intermediate stage. Files resulting from actions taken upon a given raw dataset will all have the same OBS_ID, but could have different values of the FILENAME keyword. For the FILENAME format, we recommend using OBSID as a prefix, and attach qualifiers denoting the stage of treatment and format.

Requirement: Required

Date and Time Keywords

CreationDate

FITS Name:DATE

FITS Type:str

Description: UTC date of file creation in date/time format (yyyy-mm-ddThh:mm:ss.ssss); see FITS standard for additional detail.

Requirement: Required

ObservationDate

FITS Name:DATE-OBS

FITS Type:str

Description: UTC date of observation at the start of the exposure in date/time format (yyyy-mm-ddThh:mm:ss.ssss); see FITS standard for additional detail.

Requirement: Absolutely Required for Archive Ingestion*

ObservationStartUTC

FITS Name:UTCSTART

FITS Type:str

Description: UTC time at the start of the exposure.

Requirement: Required

ObservationEndUTC

FITS Name:UTCEND

FITS Type:str

Description: UTC time at the end of the exposure.

Requirement: Required

Environmental Keywords

WaterVaporZenith_Start

FITS Name:WVZ_STA

FITS Type:flt

Description: Integrated precipitable water vapor to the zenith, running average of previous 60 seconds. Start of observation.

Requirement: Required

WaterVaporZenith_End

FITS Name:WVZ_END

FITS Type:flt

Description: Integrated precipitable water vapor to the zenith, running average of previous 60 seconds. End of observation.

Requirement: Required

static_air_temp

FITS Name:TEMP_OUT

FITS Type:flt

Description: Static air temperature, as returned by the MCCA at start of observation.

Requirement: Required

PrimaryMirrorTemperature_1

FITS Name:TEMPPRI1

FITS Type:flt

Description: Primary mirror temp #1, at start of observation.

Requirement: Required

PrimaryMirrorTemperature_2

FITS Name:TEMPPRI2

FITS Type:flt

Description: Primary mirror temp #2, at start of observation.

Requirement: Required

PrimaryMirrorTemperature_3

FITS Name:TEMPPRI3

FITS Type:flt

Description: Primary mirror temp #3, at start of observation.

Requirement: Required

SecondaryMirrorTemperature_1

FITS Name:TEMPSEC1

FITS Type:flt

Description: Temperature of secondary mirror, at start of observation.

Requirement: Required

Aircraft Keywords

Altitude_Start

FITS Name:ALTI_STA

FITS Type:flt

Description:Aircraft altitude from mean sea level according to MCCA (baro corrected), at start of observation.

Requirement: Required

Altitude_End

FITS Name:ALTI_END

FITS Type:flt

Description:Aircraft altitude from mean sea level according to MCCA (baro corrected), at end of observation.

Requirement: Required

Airspeed

FITS Name:AIRSPEED

FITS Type:flt

Description:True aircraft airspeed, as returned by the MCCA at start of observation.

Requirement: Required

GroundSpeed

FITS Name:GRDSPEED

FITS Type:flt

Description:Current ground speed of aircraft, as returned by the MCCA at start of observation.

Requirement: Required

Latitude_Start

FITS Name:LAT_STA

FITS Type:flt

Description:Current aircraft latitude, as returned by the MCCA at start of observation.

Requirement: Required

Longitude_Start

FITS Name:LON_STA

FITS Type:flt

Description:Current aircraft longitude, as returned by the MCCA at start of observation. W is negative, E is positive, from prime meridian.

Requirement: Required

Latitude_End

FITS Name:LAT_END

FITS Type:flt

Description:Current aircraft latitude, as returned by the MCCA at end of observation.

Requirement: Required

Longitude_End

FITS Name:LON_END

FITS Type:flt

Description:Current aircraft longitude, as returned by the MCCA at end of observation. W is negative, E is positive, from prime meridian.

Requirement: Required

Heading

FITS Name:HEADING

FITS Type:flt

Description:True aircraft heading, as returned by the MCCA at start of observation.

Requirement: Required

TrackAngle

FITS Name:TRACKANG

FITS Type:flt

Description:Aircraft track angle, as returned by the MCCA at start of observation.

Requirement: Required

Telescope Keywords

Telescope

FITS Name:TELESCOP

FITS Type:str

Description: Telescope used for the observation. Usually SOFIA but can also be telescope an another observatory.

Requirement: Required

TelescopeConfig

FITS Name:TELCONF

FITS Type:str

Description: Telescope configuration. The configuration defines the mirrors, correctors, light paths, etc... On SOFIA, this should be controlled by the MCCA.

Requirement: Required

TelescopeRA

FITS Name:TELRA

FITS Type:flt

Description:Right ascension of SI boresight (SIBS), as returned by the telescope control system (J2000). Representation may be either decimal hours or HH:MM:SS.s.

Requirement: Required

TelescopeDec

FITS Name:TELDEC

FITS Type:flt

Description: Declination of SI boresight, as returned by the telescope control system (J2000). Representation may be either decimal degrees or DD:MM:SS.

Requirement: Required

TelescopeVPA

FITS Name:TELVPA

FITS Type:flt

Description: Vertical Position Angle (VPA) of SI boresight, as returned by the telescope control system in ICRS (J2000) -- Is this really VPA or ROF? -- TBC.

Requirement: Required

TelescopeEquinox

FITS Name:TELEQUI

FITS Type:str

Description: Equinox of returned RA/Dec/VPA.

Requirement: Required

LastRewindUTC

FITS Name:LASTREW

FITS Type:str

Description: UTC time of last telescope rewind.

Requirement: Required

TelescopeFocus_Start

FITS Name:FOCUS_ST

FITS Type:flt

Description:Telescope focus: Measured position of the FCM focus mechanism in the T direction -- as returned by the MCCA (microns) at observation start.

Requirement: Required

TelescopeFocus_End**FITS Name:**FOCUS_EN**FITS Type:**flt**Description:**Telescope focus: Measured position of the FCM focus mechanism in the T direction -- as returned by the MCCS (microns) at observation end.**Requirement:** Required**TelescopeElevation -- [Updated!](#)****FITS Name:**TELEL**FITS Type:**flt**Description:** Telescope elevation above the horizon. At start of observation.**Requirement:** Required**TelescopeCrossElevation****FITS Name:**TELXEL**FITS Type:**flt**Description:** Telescope cross elevation in the cavity reference frame. At start of observation.**Requirement:** Required**TelescopeLineOfSight****FITS Name:**TELLOS**FITS Type:**flt**Description:** Telescope line of sight angle in the cavity reference frame. At start of observation.**Requirement:** Required**TascuStatus****FITS Name:**TSC-STAT**FITS Type:**str**Description:** State of the TASCU system at the end of the current observation as returned by the MCCS. See specification in MCCS_SI_04.**Requirement:** Required**TascuFBCStatus****FITS Name:**FBC-STAT**FITS Type:**str**Description:** State of the flexible body compensation system at the end of the current observation as returned by the MCCS. See specification in MCCS_SI_04.**Requirement:** Required**ObservationRequestRA****FITS Name:**OBSRA**FITS Type:**flt**Description:**Requested right ascension for the observation before any manual "tweaking", either from the instrument control software, an AOR, or the flight executor. This may be different from the actual object coordinates and/or telescope coordinates. Representation may be either decimal hours or HH:MM:SS.s. Reference frame can be specified in the comment field (ICRS recommended) and equinox specified using the Equinox keyword.**Requirement:** Required**ObservationRequestDec****FITS Name:**OBSDEC**FITS Type:**flt**Description:**Requested declination for the observation before any manual "tweaking", either from the instrument control software, an AOR, or the flight executor. This may be different from the actual object coordinates and/or telescope coordinates. Representation may be either decimal degrees or DD:MM:SS. Reference frame can be specified in comment field (ICRS recommended) and equinox specified using the Equinox keyword.**Requirement:** Required**ObservationRequestEquinox****FITS Name:**EQUINOX**FITS Type:**flt

Description: Equinox for ObservationRA and ObservationDec. Does not apply to TelescopeRA/Dec which are fixed to ICRS (J2000). See FITS standard for additional detail. If not specified, equinox is assumed to be year 2000.

Requirement: Required

ZenithAngle_Start

FITS Name:ZA_START

FITS Type:flt

Description: Zenith angle of telescope pointing at start of acquisition. Use telescope elevation to calculate ZA.

Requirement: Required

ZenithAngle_End

FITS Name:ZA_END

FITS Type:flt

Description: Zenith angle of telescope pointing at end of acquisition. Use telescope elevation to calculate ZA.

Requirement: Required

TrackingMode

FITS Name:TRACMODE

FITS Type:str

Description: SOFIA Tracking mode (last commanded). See MCCS_SI_04 for description of states and modes.

Requirement:Tracking

TrackingError

FITS Name:TRACERR

FITS Type:log

Description:Flag to indicate if there was a tracking error during the observation. Ideally the SI software would monitor the tracking mode (ta_trc.trc_status_table.main_op_mode_id) for any error/abnormalities. If an error does occur, the TRACERR would then be set to T. (

Requirement:Tracking

Data Collection Keywords

isChopping

FITS Name:CHOPPING

FITS Type:log

Description: Chopping flag

Requirement:If mode in use.

isNodding

FITS Name:NODDING

FITS Type:log

Description: Nodding flag -- this should be set if the SI is executing a repeated nod pattern, for example (ABBA)(ABBA), etc...

Requirement:If mode in use.

isDithering

FITS Name:DITHER

FITS Type:log

Description: Dithering flag

Requirement:If mode in use.

isMapping

FITS Name:MAPPING

FITS Type:log

Description: Mapping flag.

Requirement:If mode in use.

isScanning

FITS Name:SCANNING

FITS Type:log

Description: Scanning flag.
Requirement: If mode in use.

Annotation Keywords

Instrument Keywords

Instrument

FITS Name: INSTRUME

FITS Type: str

Description: Instrument name, as specified in the SI-DCS ICD.

Requirement: Absolutely Required for Archive Ingestion*

Datatype

FITS Name: DATATYPE

FITS Type: str

Description: Type of observation data: Image, Spectral, or Other.

Requirement: Required

InstrumentConfiguration

FITS Name: INSTCFG

FITS Type: str

Description: Instrument configuration - simple description, as specified in the SI-DCS ICD. e.g. IMAGING, GRISM, SPECTROSCOPY, etc...

Requirement: Required

InstrumentMode

FITS Name: INSTMODE

FITS Type: str

Description: Instrument observing mode - simple description, as specified in the SI-DCS ICD. e.g. C2D, MAPPING, etc...

Requirement: Required

MCCSMode

FITS Name: MCCSMODE

FITS Type: str

Description: SI mode as defined in the MCCS_SI_04.

Requirement: Required

ExposureTime

FITS Name: EXPTIME

FITS Type: flt

Description: Total effective on-source exposure time of the observation. This is the total time during which photons from the object of interest are collected by the detector. It includes any shutter corrections (which may not apply for most (all?) SOFIA instruments), and nodding/chopping corrections, and should match the algorithm(s) supplied for any time estimation tools (e.g. SITE). For FITS, EXPTIME should be used instead of the FITS specification EXPOSURE. ExposureTime should be specified for all images in a data file.

Requirement: Required

SpectralElement1 -- [Updated!](#)

FITS Name: SPECTEL1

FITS Type: str

Description: First spectral element (filter, grism, etc...) as specified in SI-DCS ICD. Need only contain the unique identifier from the SI-DCS ICD; more detailed filter/grism/mixer can be stored in instrument-specific keywords.

Requirement: Absolutely Required for Archive Ingestion*

SpectralElement2

FITS Name: SPECTEL2

FITS Type: str

Description: Second spectral element (filter, grism, etc...) as specified in SI-DCS ICD. Need only contain the unique identifier from the SI-DCS ICD; more detailed filter/grism/mixer can be stored in instrument-specific keywords. Set to "NONE" if no second element in use.

Requirement: Absolutely Required for Archive Ingestion*

InstrumentSlit

FITS Name: SLIT

FITS Type: str

Description: Slit identifier, as specified in the SI-DCS ICD. Need only contain the unique identifier; more detailed info can be stored in instrument-specific keywords.

Requirement: Spectroscopy configs: if slit in use.

WavelengthCentral

FITS Name: WAVECENT

FITS Type: flt

Description: Central wavelength of observation for imaging modes. This is a rough figure only, intended to be used for archive searches across all SOFIA instruments.

Requirement: Imaging modes only.

Resolution

FITS Name: RESOLUN

FITS Type: flt

Description: Approximate spectral resolution of observation for spectroscopy modes, expressed as $R = c/\Delta V = \lambda/\Delta\lambda$. This is a rough figure only, intended to be used for archive searches across all SOFIA instruments.

Requirement: Spectroscopy modes only.

DetectorChannel -- *New!*

FITS Name: DETCHAN

FITS Type: str

Description: Detector Channel as specified in the SI-DCS ICD.

Requirement: FORCAST and FIFI-LS should populate these.

TotalIntegrationTime -- *New!*

FITS Name: TOTINT

FITS Type: flt

Description: Total integration time (s)

Requirement: FORCAST

Array Detector Keywords

Detector

FITS Name: DETECTOR

FITS Type: str

Description: Detector name.

Requirement: Array instruments only.

DetectorSize

FITS Name: DETSIZE

FITS Type: str

Description: Unbinned detector size in pixels.

Requirement: Array instruments only.

PixelScale

FITS Name: PIXSCAL

FITS Type: flt

Description: Projected pixel scale on the sky.

Requirement: Array instruments only.

Subarrays

FITS Name:SUBARRNO**FITS Type:**int**Description:** Number of sub arrays used in data acquisition. Full array assumed if absent.**Requirement:**If subarrays in use.**SubarraySize****FITS Name:**SUBARR%2d**FITS Type:**str**Description:** The log unbinned size of the n-th subarray in section notation.**Requirement:**If subarrays in use.**ScienceInstrumentBoresightX -- *Updated!*****FITS Name:**SIBS_X**FITS Type:**flt**Description:**Location of SI boresight in pixel space -- x_si.**Requirement:**Array instruments only.**ScienceInstrumentBoresightY -- *Updated!*****FITS Name:**SIBS_Y**FITS Type:**flt**Description:**Location of SI boresight in pixel space -- y_si.**Requirement:**Array instruments only.**WCS_CTYPE****FITS Name:**CTYPEn**FITS Type:**str**Description:**WCS: Coordinate type and projection for n-th axis. See list of supported projections at <http://tdc-www.harvard.edu/wcstools/wcstools.fits.html>.**Requirement:**Imaging only.**WCS_CRPIX -- *Updated!*****FITS Name:**CRPIXn**FITS Type:**flt**Description:**WCS: pixel coordinates of the reference point of the n-th axis to which the projection and the rotation refer.**Requirement:**Array instruments only.**WCS_CRVAL****FITS Name:**CRVALn**FITS Type:**flt**Description:**WCS: reference pixel coordinate for n-th axis as right ascension and declination or longitude and latitude in decimal degrees. FITS Comment field should include estimate of uncertainty in absolute pointing (See TBD Document).**Requirement:**Imaging only.**WCS_CDELT****FITS Name:**CDELTn**FITS Type:**flt**Description:**WCS: Plate scale in degrees per pixel for the n-th axis at the reference pixel. Either the CDi_j *or* the CDELT/CROTA keywords should be used, but not both. FITS comment field should include estimate of uncertainty based on known distortion or other optical effects.**Requirement:**Imaging only.**WCS_CROTA2****FITS Name:**CROTA2**FITS Type:**flt**Description:**WCS: Rotation of axes in degrees. WCS FITS convention is to use CROTA2. Either the CDi_j *or* the CDELT/CROTA keywords should be used, but not both. FITS comment field should include estimate of uncertainty in TA rotation of field (see TBD Document).**Requirement:**Imaging only.**WCS_RotMatrix**

FITS Name:CDi_j**FITS Type:**flt**Description:**WCS: Rotation matrix for WCS -- CD1_1, CD1_2, CD2_1, and CD2_2. Either the CDi_j *or* the CDELT/CROTA keywords should be used, but not both. FITS comment field should include estimate of uncertainty for each matrix element.**Requirement:**Imaging only.

Heterodyne Keywords

FrontendDevice

FITS Name:FRONTEND**FITS Type:**str**Description:**Name of frontend device.**Requirement:**Heterodyne instruments only.

BackendDevice

FITS Name:BACKEND**FITS Type:**str**Description:**Name of backend device.**Requirement:**Heterodyne instruments only.

BackendBandwidth

FITS Name:BANDWID**FITS Type:**flt**Description:**Total bandwidth of heterodyne backend (Hz).**Requirement:**Heterodyne instruments only.

SystemTemperature

FITS Name:TSYS**FITS Type:**flt**Description:**Heterodyne system temperature.**Requirement:**Heterodyne instruments only.

FrequencyResolution

FITS Name:FREQRES**FITS Type:**flt**Description:**Nominal frequency resolution -- may differ from channel spacing.**Requirement:**Heterodyne instruments only.

ReferenceFrequency

FITS Name:OBSFREQ**FITS Type:**flt**Description:**The observed frequency (Hz) at the reference pixel of the frequency-like axis.**Requirement:**Heterodyne instruments only.

SidebandFrequency

FITS Name:IMAGFREQ**FITS Type:**flt**Description:**The image sideband freq (Hz) corresponding to ReferenceFrequency.**Requirement:**Heterodyne instruments only.

RestFrequency

FITS Name:RESTFREQ**FITS Type:**flt**Description:**Rest frequency.**Requirement:**Heterodyne instruments only.

VelocityDefinition

FITS Name:VELDEF**FITS Type:**str

Description: The velocity definition and frame (8 characters). The first 4 characters describe the velocity definition. Possible definitions include: RADI (radio); OPTI (optical); RELA (relativistic). The second 4 characters describe the reference frame (e.g. ``-LSR'', ``-HEL'', ``-OBS''). If the frequency-like axis gives a frame, then the frame in VELDEF only applies to any velocities given as columns or keywords (virtual columns).

Requirement:Heterodyne instruments only.

VelocityFrame

FITS Name:VFRAME

FITS Type:flt

Description: The radial velocity of the reference frame wrt the observer. $V_{\text{frame}} - V_{\text{telescope}}$.

Requirement:Heterodyne instruments only.

RadialVelocity

FITS Name:RVSYS

FITS Type:flt

Description: The radial velocity, $V_{\text{source}} - V_{\text{telescope}}$.

Requirement:Heterodyne instruments only.

Data Structure Keywords

Chopping Keywords

ChopFrequency

FITS Name:CHPFREQ

FITS Type:flt

Description: Measured TCM chop frequency

Requirement:Chopping

ChopProfile

FITS Name:CHPPROF

FITS Type:str

Description:Indicates whether 2 or 3 point chopping profile is being used. For 3-point chopping, the center position usually contains the object of interest. MCCS returns '0' for 2 point and '1' for 3-point.

Requirement:Chopping

ChopSymmetry

FITS Name:CHPSYM

FITS Type:str

Description:Indicates whether symmetric or asymmetric chopping is being used.

Requirement:Chopping

ChopAmplitude 1

FITS Name:CHPAMP1

FITS Type:flt

Description: Calculated amplitude on the sky. MCCS calculates the amplitude on the sky based on actual SMA data.

Requirement:Chopping

ChopAmplitude 2

FITS Name:CHPAMP2

FITS Type:flt

Description: Calculated second amplitude on the sky. MCCS calculates the amplitude on the sky based on actual SMA data.

Requirement:Chopping

ChopCoordSys -- [Updated!](#)

FITS Name:CHPCRSYS

FITS Type:str

Description: Reference frame for which MCCS computes SMA parameters. MCCS calculates sky_tip, sky_tilt, and sky_angle differently depending on which reference frame was last used in the sma.chop command. This value defaults to TARF if sma.chop has

not been sent previously.

Requirement:Chopping

ChopAngle -- *Updated!*

FITS Name:CHPANGLE

FITS Type:flt

Description: Calculated angle in the sky_coord_sys reference frame. MCCA calculates the angle in the sky_coord_sys reference frame based on actual SMA data. The angle is the orientation of the chop throw with up equals zero.

Requirement:Chopping

ChopTip

FITS Name:CHPTIP

FITS Type:flt

Description: Calculated tip in the sky_coord_sys reference frame. MCCA calculates the tip in the sky_coord_sys reference frame based on actual SMA data.

Requirement:Chopping

ChopTilt

FITS Name:CHPTILT

FITS Type:flt

Description: Calculated tilt in the sky_coord_sys reference frame. MCCA calculates the tilt in the sky_coord_sys reference frame based on actual SMA data.

Requirement:Chopping

ChopPhase

FITS Name:CHPPHASE

FITS Type:int

Description: Chopper phase as defined by MCCA. Time delay between the synch signal and the start of the setpoint which has the positive tilt increment with respect to the commanded offset.

Requirement:Chopping

Nodding Keywords

NodDwellTime

FITS Name:NODTIME

FITS Type:flt

Description: Total time per nod position (dwell time) -- not including nod slew time and nod settle time (see NodSettleTime).

Requirement:Nodding

NodCycles

FITS Name:NODN

FITS Type:int

Description: Number of nod cycles.

Requirement:Nodding

NodSettleTime

FITS Name:NODSETL

FITS Type:flt

Description:Time required for telescope to settle after nod slew is complete. Amount of time to wait between when telescope arrives at nod destination and when to begin integrating.

Requirement:Nodding

NodAmplitude

FITS Name:NODAMP

FITS Type:flt

Description: Nod amplitude on sky.

Requirement:Nodding

NodBeam

FITS Name:NODBEAM**FITS Type:**str**Description:** Current nod beam position.**Requirement:**Nodding**NodPattern -- *Updated!*****FITS Name:**NODPATT**FITS Type:**str**Description:** Pointing sequence pattern for one nod cycle (there could be many nod cycles in an observation). Beam A is usually assumed to contain the object of interest.**Requirement:**Nodding**NodStyle****FITS Name:**NODSTYLE**FITS Type:**str**Description:**Nodding style for coordinated chopping/nodding. e.g. nod-matched-chop, nod-perpendicular-chop, etc...**Requirement:**Nodding/Chopping**NodCoordSys****FITS Name:**NODCRSYS**FITS Type:**str**Description:** Coordinate system in which nod positions (NODPOSX,Y) and rotations are defined.**Requirement:**Nodding**NodAngle -- *Updated!*****FITS Name:**NODANGLE**FITS Type:**flt**Description:** Nod angle, clockwise from y axis defined by NODCRSYS.**Requirement:**Nodding

Dithering Keywords

DitherCoordiante -- *New!***FITS Name:**DTHCRSYS**FITS Type:**str**Description:** Coordinate system for dither offsets.**Requirement:**Dithering**DitherXOffset -- *New!*****FITS Name:**DTHXOFF**FITS Type:**flt**Description:** Specified dither offset X for each subsequent frame, in arcseconds.**Requirement:**Dithering**DitherYOffset -- *New!*****FITS Name:**DTHYOFF**FITS Type:**flt**Description:** Specified dither offset Y for each subsequent frame, in arcseconds.**Requirement:**Dithering**DitherPattern -- *Updated!*****FITS Name:**DTHPATT**FITS Type:**str**Description:** Approximate shape of dither pattern.**Requirement:**Dithering**DitherPositions****FITS Name:**DTHNPOS**FITS Type:**int

Description: Number of dither positions.

Requirement:Dithering

DitherPositionIndex

FITS Name:DTHINDEX

FITS Type:int

Description: Dither position index.

Requirement:Dithering

DitherOffset

FITS Name:DTHOFFS

FITS Type:flt

Description: Specified dither offset for each subsequent frame, in arcseconds.

Requirement:Dithering

Mapping Keywords

MapCoordSys

FITS Name:MAPCRSYS

FITS Type:str

Description: Coordinate system in which ES map positions are defined. OBSRA/DEC are assumed to describe position of Map Center.

Requirement:Mapping

MapPositionsX

FITS Name:MAPNXPOS

FITS Type:int

Description: Number of map positions in X coordinate as defined by MAPCRSYS.

Requirement:Mapping

MapPositionsY

FITS Name:MAPNYPOS

FITS Type:int

Description: Number of map positions in Y coordinate as defined by MAPCRSYS.

Requirement:Mapping

MapIntervalX

FITS Name:MAPINTX

FITS Type:flt

Description: Mapping step interval in X coordinate as defined by MAPCRSYS. OBSRA/DEC are assumed to describe position of Map Center.

Requirement:Mapping

MapIntervalY

FITS Name:MAPINTY

FITS Type:flt

Description: Mapping step interval in Y coordinate as defined by MAPCRSYS. OBSRA/DEC are assumed to describe position of Map Center.

Requirement:Mapping

Scanning Keywords (Constant Velocity)

ScanStartRA

FITS Name:SCNRA0

FITS Type:flt

Description:Start of scan - RA, coordinate system specified in keyword comment filed and EQUINOX.

Requirement:Scanning

ScanStartDec**FITS Name:**SCNDEC0**FITS Type:**flt**Description:** Start of scan - Dec , coordinate system specified in keyword comment filed and EQUINOX.**Requirement:**Scanning**ScanEndRA****FITS Name:**SCNRAF**FITS Type:**flt**Description:**End of scan - RA, coordinate system specified in keyword comment filed and EQUINOX.**Requirement:**Scanning**ScanEndDec****FITS Name:**SCNDECf**FITS Type:**flt**Description:** End of scan - Dec, coordinate system specified in keyword comment filed and EQUINOX.**Requirement:**Scanning**ScanRate****FITS Name:**SCNRATE**FITS Type:**flt**Description:** Commanded slew rate in arcsec/sec along path.**Requirement:**Scanning**ScanDirection****FITS Name:**SCNDIR**FITS Type:**flt**Description:** Angle on sky in some coordinate system to scan from ScanStartRA/Dec, alternative specification to ScanEndRA/Dec.**Requirement:**Scanning