A File Format for Satellite Atmospheric Chemistry Data Based On Aura File Format Guidelines

Status of this Memo

This is a draft of the technical note on a file format for satellite atmospheric chemistry data.

Its distribution is unlimited.

Change Explanation

None

Copyright Notice

Abstract

This technical note presents a file format for satellite atmospheric chemistry data. It is based upon the file format guidelines successfully adopted for the EOS Aura mission. These guidelines make use of the HDF5 library with HDF-EOS5 extensions.

Table of Contents

1.0 Introduction

1.1 MOTIVATION BEHIND TECHNICAL NOTE

This technical note describes a file format for atmospheric chemistry data, which makes it easier for the end user to read product files from several different instruments. It was adopted by all four instruments on the EOS Aura satellite (HIRDLS, MLS, OMI and TES) and can be readily used by other atmospheric chemistry instruments.

The Aura teams realized that common data and file formats would greatly facilitate the sharing of data. The teams agreed to use the HDF5/HDF-EOS5 data format. In addition, it was important to concur upon a file format to further reduce the need for custom software. The teams agreed upon the names, data types and dimension order of fields. There was also agreement regarding the file-, group- and field-level attributes to include in each product file. A file-naming convention was also adopted.

Future atmospheric chemistry missions can use this technical note to enable their users to benefit from a common data and file format. They can either build upon the guidelines as described here, in which case they will benefit from being able to easily read Aura data files. Or future missions can simply use the techniques described below to adopt a common data and file format for their collection of instruments. There are many advantages to having multiple instruments follow the same data and file format conventions.

1.2 Overview of HDF and HDF-EOS

HDF (Hierarchical Data Format) is a data format library which provides self-describing, machine-transparent data storage for scientific data. Data are written and read with HDF library calls and can be accessed via C, Fortran and Java code. Numerous tools are available to allow viewing of HDF-based files. In addition, third party packages such as IDL, Mathematica and Matlab are able to manipulate HDF files. A few of the benefits of this data format are:

- Data files are easily transferable between machine platforms (ranging from PCs to high performance computers)
- Files are much more compact than equivalent ASCII files
- Information about the data can be stored along with the data (via HDF attributes)
- Files are self-describing and applications can determine the structure from data contained within the file
- Additional information about the data can be stored within the file with careful selection of names, groupings and attributes
- Data I/O is completely random access and any individual value or section of data can be quickly and independently manipulated
- Data files are easily subsettable

NASA recognized the usefulness of the HDF data format standard and adopted it for the EOS (Earth Observing System) Terra, Aqua and Aura missions. However, additional constraints were needed to ensure data compatibility between instruments and platforms. For this reason, HDF-EOS (HDF for EOS) was developed. HDF-EOS is built on top of the HDF library and requires that both libraries be present. HDF-EOS files must be written using the HDF-EOS library, but both the HDF and HDF-EOS libraries can read HDF-EOS files. While HDF-EOS was developed for EOS missions, its capabilities are generic, and can be used to store numerous kinds of Earth science data.

HDF-EOS contains four data structures which are named *Swath*, *Grid*, *Point* and *Zonal Average*. Each is accessed via its own set of HDF-EOS library calls. Swath is used for storing data that are time-ordered. It may or may not contain a cross-track dimension. Grid is used for storing data that have at least a 2-dimensional x-y grid. Point is used to store profiles of data that are not organized in time, longitude or latitude. Zonal Average is used to store mean data in latitudinal zones.

While HDF-EOS constrains HDF with its various implementations, it is still possible to create two files that contain the same information with completely different structures and which would require dramatically different readers. To ease use of each other's data sets, for purposes like validation and joint research, the Aura teams developed file format guidelines to make their files match as closely as reasonably possible. Such guidelines only make sense for Level 2 and above data files and are not intended to apply to Level 1 or other special product files.

2.0 Areas to Constrain in HDF/HDF-EOS

A number of HDF/HDF-EOS features must be agreed upon when creating uniform data sets. Items which are addressed in this technical note include:

- Major HDF-EOS version (HDF-EOS V2.x and HDF-EOS V5.x are not interchangeable)
- Organization of geolocation and data fields and attributes
- Dimension names
- Geolocation field names, data types and dimension ordering
- Data field names, data types and dimension ordering
- Units for geolocation and data fields
- Attribute names, values and units

The intended effect is to constrain the producers of data no more than is necessary. Data producers are free to shape their products in areas in which they are not specifically constrained in this technical note. For example, HDF-mediated compression is neither mandated nor forbidden. Because end users do not need to know whether or not the data has been compressed, as HDF handles any necessary uncompression automatically behind the scenes, the use of this feature has been left up to the individual data producers.

Please note that any substantial deviation from this set of guidelines will require that the user create special code for each case. A substantial deviation is something like the absence or misnaming of a mandatory field, or failure of its size or shape to match what this technical note prescribes. A non-substantial deviation would be something like including some extra fields or attributes in addition to what this technical note specifies. These are considered instrument-specific data, which are implicitly permitted by this technical note, and should not require the user to write special code.

Standardization of the type described within this document was created specifically for EOS Aura, a space-based atmospheric chemistry measurement platform. While this standard's specific usability is for the chemical species named in this document, it can be easily extended to other chemical species.

This technical note is not meant to be exhaustive; that is, it does not try to identify every item contained within a data file. Rather, it defines the major measurement fields within a file and identifies the specific requirements for storing these data. Individual instrument teams are allowed to identify additional data to include in their files and do not need to reach consensus with the rest of the community on how to store these. Information about these additional data is not included in the document.

3.0 LIST OF REQUIREMENTS

The following rules are required to be compatible with the Aura datasets. If a group of instruments are creating their own set of guidelines, then this list can be used as a template.

- Files will be created using HDF-EOS V5.x
- Data, such as profiles, which are time ordered, but not on a grid will use the HDF-EOS Swath structure.
- Data which are gridded will use the HDF-EOS Grid structure.
- Data which are zonally averaged will use the HDF-EOS Zonal Average structure.
- HDF-EOS Structure Names should adhere to the Valids list (in Section 5). It is important to note that in order for names to be uniform, they must match completely. Spacing as well as capitalization must be followed exactly.
- Profile data should be reported on a pressure grid and be ordered from the ground to space.
- Profile data fields which are multidimensional should be ordered so that the pressure coordinate is the fastest incrementing coordinate, and the number of times (profiles) is the slowest incrementing coordinate.
- Data fields should be stored in the units specified.
- HDF fill value and missing data values should have the same value. The actual missing data value will be specified in the MissingValue data field attribute.

The decision on whether to have a file contain only one species or multiple species is left to individual data providers.

4.0 Organization of data files

Standardization of HDF/HDF-EOS files requires adhering to a strict organization of the individual data file. Items which are standardized include organization of geolocation and data fields and attributes. Strict adherence to the names of dimensions, geolocation fields and data fields is required. In addition, the types, dimension ordering and units for each data and geolocation field are specified. Much the same detail is necessary for specifying attributes.

The organization of data files is graphically displayed in the following sections. The detailed specifics for each data field and attribute are described in detail in Section 5 and 6.

4.1 Swath geolocation and data fields and attributes

More than one species can be contained within a swath and more than one swath can exist within a file.

(Data File Structure)

File Level Attributes:
- InstrumentName
- ProcessLevel
- GranuleMonth
- GranuleDay
- GranuleYear
- TAI93At0zOfGranule
- PGEVersion

Swath Name: *Instrument Specific*

Swath Level Attributes:
- Pressure
- VerticalCoordinate

Dimensions:
- nTimes
- nLevels
- nWavel
- nXtrack
- nLayers

Geolocation Fields:
- Time
- Latitude
- Longitude
- Pressure
- Solar Zenith Angle
- Local Solar Time
- etc.

(See the valids for the complete listing of possible geolocation fields)

Geolocation Field Attributes:
- MissingValue
- Title
- Units
- UniqueFieldDefinition
- ScaleFactor (*only if applicable*)
- Offset (*only if applicable*)

Data Fields:
- Temperature
- O3
- etc.

(See the valids for the complete listing of possible Data fields)

Data Field Attributes:
- MissingValue
- Title
- Units
- UniqueFieldDefinition
- ScaleFactor (*only if applicable*)
- Offset (*only if applicable*)

Swath Name2: *Instrument Specific*

Additional swaths may occur in a file

4.2 Grid data fields and attributes

More than one species can be contained within a grid and more than one grid can exist within a file. While the HDF-EOS Grid structure does not distinguish grid (geolocation fields) from data fields, we have separated them for the sake of documentation.

(Data File Structure)

File Level Attributes:
- OrbitNumber
- OrbitPeriod
- InstrumentName
- ProcessLevel
- GranuleMonth
- GranuleDay
- GranuleYear
- GranuleDayOfYear
- TAI93At0zOfGranule
- PGEVersion
- StartUTC
- EndUTC
- Period (‘Daily’, ‘Monthly’, ‘8-day’, ‘Global Survey’)

Grid Name: *Instrument Specific*

Grid Level Attributes:
- Projection
- GridOrigin
- GridSpacing
- GridSpacingUnit
- GridSpan
- GridSpanUnit

Dimensions:
- XDim
- YDim
- nLevels

Grid Fields:
- Latitude
- Longitude
- Pressure

(See the valids for the complete listing of possible Grid Fields)

Grid Field Attributes:
- MissingValue
- Title
- Units
- UniqueFieldDefinition
- ScaleFactor (*only if applicable*)
- Offset (*only if applicable*)

Data Fields:
(Value, Precision, …) for the following:
- Temperature
- N2O
- Etc.

(See the valids for the complete listing of possible Data fields)

Data Field Attributes:
- MissingValue
- Title
- Units
- UniqueFieldDefinition
- ScaleFactor (*only if applicable*)
- Offset (*only if applicable*)

Grid Name2: *Instrument Specific*

Additional Grids may occur in a file

4.3 Zonal Average data fields and attributes

More than one species can be contained within a Zonal Average and more than one Zonal Average can exist within a file. While the HDF-EOS Zonal Average structure does not distinguish zonal (geolocation) fields from data fields, we have separated them for the sake of documentation.

(Data File Structure)

File Level Attributes:
- OrbitNumber
- OrbitPeriod
- InstrumentName
- ProcessLevel
- GranuleMonth
- GranuleDay
- GranuleYear
- GranuleDayOfYear
- TAI93At0zOfGranule
- PGEVersion
- StartUTC
- EndUTC
- Period ('Daily', 'Monthly', '8-day' 'Global Survey')

Zonal Name: *Instrument Specific*

Zonal Level Attributes:
- Pressure
- VerticalCoordinate
- ZonalSpacing
- ZonalSpacingUnit

Dimensions:
- nLats
- nLevels

Zonal Fields:
- Latitude
- Pressure
- Etc.

(See the valids for the complete listing of possible Zonal fields)

Zonal Field Attributes:
- MissingValue
- Title
- Units
- UniqueFieldDefinition
- ScaleFactor (*only if applicable*)
- Offset (*only if applicable*)

Data Fields:
(Value, StdDeviation , DataCount,…) for the following
- Temperature
- N2O
- Etc.

(See the valids for the complete listing of possible Data fields)

Data Field Attributes:
- MissingValue
- Title
- Units
- UniqueFieldDefinition
- ScaleFactor (*only if applicable*)
- Offset (*only if applicable*)

Zonal Name2: *Instrument Specific*

Additional zonals may occur in a file

5.0 File Specifics

This section lists the specific names which will be used within the file. These names must be used if they are applicable. HDF/HDF-EOS lends itself to creation of data files which may or may not have specific data fields. Each instrument team only writes out the data fields which pertain to their instrument. Every data field is explicitly defined with its name, dimension order, data type and units. All of these components must be followed exactly.

5.1 Swath Files

This section lists the specifics of each part of an HDF-EOS Swath file. Details are provided for the names of the swath(s), dimensions and data fields. Each field is then further specified with its exact name, dimensionality, data type and units.

5.1.1 Name of Swath

Varies by Instrument

The value for this name is not constrained. An end user can use HE5_SWinqswath to determine what the values are for each data set.

5.1.2 Dimensions in Swath

These are the actual dimensions of the geolocation and data field quantities. These dimensions must be used if applicable.

nLevels	Number of pressure levels
nTimes	Number of times (profiles) in data set (this may be unlimited)
nWavel	Number of wavelengths (OMI specific)
nXtrack	Number of pixels in the across track direction (OMI specific)
nLayers	Number of atmospheric layers (OMI specific)
nFreqs	Number of frequencies (MLS specific)
nCloudTypes	Number of cloud types (HIRDLS specific)
nFreq	Number of frequencies (TES specific)
nUARSlevels	87 standard UARS levels (TES specific)

5.1.3 Swath Geolocation Fields:

This is data which describe the scientific measured quantities. They provide information to aid in describing the data's "location". The name, dimension order, data type and units must be adhered to exactly, if applicable.

NOTE THE ORDER OF DIMENSIONS IS IN FORTRAN ORDER WITH THE FIRST DIMENSION BEING THE MOST RAPIDLY INCREMENTING. IN LANGUAGES WHERE THE LAST DIMENSION IS THE MOST RAPIDLY INCREMENTING DIMENSION, THE ORDER OF DIMENSIONS SHOULD BE REVERSED.

HDF-EOS Name	*Dimension*	*Data Type*	*Units*(1)	*Instrument*				*Notes*
				H	*M*	*O*	*T*	
Time	(nTimes)	REAL - 64 BIT	s	X	X	X	X	time in TAI units
Latitude	(nTimes)	REAL - 32 BIT	deg	X	X		X	Geodetic Latitude
Latitude	(nXtrack,nTimes)	REAL - 32 BIT	deg			X		Geodetic Latitude
Longitude	(nTimes)	REAL - 32 BIT	deg	X	X		X	range: [-180 to 180]
Longitude	(nXtrack,nTimes)	REAL - 32 BIT	deg			X		range: [-180 to 180]
Pressure+	(nLevels)	REAL - 32 BIT	hPa	X	X			Pressure will be a superset of the UARS pressure levels (ordered from ground to space)
Pressure+	(nLevels,nTimes)	REAL - 32 BIT	hPa				X	Pressure will be a superset of the UARS pressure levels (ordered from ground to space)
Altitude	(nLevels,nTimes)	REAL - 32 BIT	m	X			X	
SecondsInDay	(nTimes)	REAL - 32 BIT	s	X		O		Seconds from midnight of day listed in global attributes
SolarZenithAngle	(nTimes)	REAL - 32 BIT	deg	X	X		X	
SolarZenithAngle	(nXtrack,nTimes)	REAL - 32 BIT	deg			X		
LocalSolarTime	(nTimes)	REAL - 32 BIT	h	X	X		X	(hours)
SpacecraftLatitude	(nTimes)	REAL - 32 BIT	deg	X		X	A	Geodetic latitude above WGS84 ellipsoid
SpacecraftLongitude	(nTimes)	REAL - 32 BIT	deg	X		X	A	Longitude above WGS84 ellipsoid , range: [-180 to 180]
SpacecraftAltitude	(nTimes)	REAL - 32 BIT	m	X		X	A	Height above WGS84 ellipsoid
OrbitAscendingFlag	(nTimes)	INTEGER - 8 BIT	NoUnits	X			A	1= true then orbit is ascending

HDF-EOS Name	*Dimension*	*Data Type*	*Units*(1)	*Instrument* *H*	*M*	*O*	*T*	*Notes*
SurfaceElevStandardDeviation	(nTimes)	REAL - 32 BIT	m				X	
OrbitGeodeticAngle	(nTimes)	REAL - 32 BIT	deg		X			
Frequency	(nTimes)	REAL - 32 BIT	GHz		X			In file only when appropriate
LineOfSightAngle	(nTimes)	REAL - 32 BIT	deg(EastofNorth)		X			
SolarAzimuthAngle	(nXtrack,nTimes)	REAL - 32 BIT	deg(EastofNorth)			O		range: [-180 to 180]
SolarAzimuthAngle	(nTimes)	REAL - 32 BIT	deg(EastofNorth)				A	range:[0 to 360]
ViewingZenithAngle	(nXtrack,nTimes)	REAL - 32 BIT	deg			X		
ViewingAzimuthAngle	(nXtrack,nTimes)	REAL - 32 BIT	deg(EastofNorth)			O		range: [-180 to 180]
RelativeAzimuthAngle	(nXtrack,nTimes)	REAL - 32 BIT	deg(EastofNorth)			O		(SolarAzimuthAngle + 180) - ViewingAzimuthAngle
TerrainHeight	(nXtrack,nTimes)	UNSIGNED INT - 16 BIT	m			X		
GroundPixelQualityFlags	(nXtrack,nTimes)	UNSIGNED INT - 16 BIT	NoUnits			X		
Tgt_SpacecraftAzimuth	(nTimes)	REAL - 32 BIT	deg(EastofNorth)				X	range:[90 to- 90]
Tgt_SpacecraftZenith	(nTimes)	REAL - 32 BIT	deg				X	
ScienceScanMode	(nTimes)	INTEGER - 16 BIT	NoUnits	X				HIRDLS Science Scan Mode identifier (short integer)
ScanTable	(nTimes)	INTEGER - 16 BIT	NoUnits	X				HIRDLS Scan Table identifier
ScanUpFlag	(nTimes)	INTEGER - 8 BIT	NoUnits	X				HIRDLS Scan Up identifier 1=up (true)
ScanElevationAtNominalAltitude	(nTimes)	REAL - 32 BIT	deg	X				
ScanAzimuthAtNominalAltitude	(nTimes)	REAL - 32 BIT	deg	X				range: [-180 to 180]
TangentHeightAtNominalAltitude	(nTimes)	REAL - 32 BIT	m	X				
ViewDirectionAtNominalAltitude	(nTimes)	REAL - 32 BIT	deg(EastofNorth)	X				
ProfileID	(nTimes)	INTEGER - 32 BIT	NoUnits	X				HIRDLS identification number for that day's profiles
BoresightNadirAngle	(nTimes)	REAL - 64 BIT	deg				X	32 bits (accommodate N+N2 scans)
BoresightNadirAngleUnc	(nTimes)	REAL - 64 BIT	deg				X	
BoresightAzimuth	(nTimes)	REAL - 64 BIT	deg				X	range:[0 to 360]
BoresightTangentHeight	(nTimes)	REAL - 32 BIT	m				X	(Limb only)
BoresightTangentHeightUnc	(nTimes)	REAL - 32 BIT	m				X	(Limb only)
Latitude_Footprint_1	(nTimes)	REAL - 64 BIT	deg				X	
Latitude_Footprint_2	(nTimes)	REAL - 64 BIT	deg				X	
Latitude_Footprint_3	(nTimes)	REAL - 64 BIT	deg				X	
Latitude_Footprint_4	(nTimes)	REAL - 64 BIT	deg				X	
Longitude_Footprint_1	(nTimes)	REAL - 64 BIT	deg				X	

HDF-EOS Name	*Dimension*	*Data Type*	*Units*(1)	*Instrument* *H*	*M*	*O*	*T*	*Notes*
Longitude_Footprint_2	(nTimes)	REAL - 64 BIT	deg				X	
Longitude_Footprint_3	(nTimes)	REAL - 64 BIT	deg				X	
Longitude_Footprint_4	(nTimes)	REAL - 64 BIT	deg				X	
Sequence	(nTimes)	INTEGER - 16 BIT	NoUnits				X	
Scan	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
SurfaceTypeFootprint	(nTimes)	REAL - 32 BIT	deg				X	

1 - Aura used their own naming conventions for units. Future missions may want to consider using the SI conventions

X – Field in standard file
A – Data items which will be carried in an ancillary file that is created only once per set of standard products
O - Optional field

\+ *For HIRDLS* - nLevels is set to 145 (1000. * 10 ** (-i/24) i=0,144)
For TES Global Survey and Special Observation products - nLevels means 87 UARS levels + 1 for the surface. For the TES Summary product, nLevels means 15 UARS levels + 1 for the surface.

5.1.4 Swath Data Field Names:

This is the actual scientific data. The name, dimension order, data type and units must be adhered to exactly, if applicable.

NOTE THE ORDER OF DIMENSIONS IS IN FORTRAN ORDER WITH THE FIRST DIMENSION BEING THE MOST RAPIDLY INCREMENTING.* IN *LANGUAGES WHERE THE LAST DIMENSION IS THE MOST RAPIDLY INCREMENTING DIMENSION, THE ORDER OF DIMENSIONS SHOULD BE REVERSED.

HDF-EOS Name	*Dimension*	*Data Type*	*Units*(1)	*H*	*M*	*O*	*T*	*Notes*
Temperature	(nLevels,nTimes)	REAL - 32 BIT	K	X	X		X	
TemperaturePrecision	(nLevels,nTimes)	REAL - 32 BIT	K	X	X		X	
TemperatureNormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
TemperatureQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
O3	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X		X	
O3	(nLayers,nXtrack,nTimes)	REAL - 32 BIT	vmr			X		
O3Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X		X	
O3NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
O3Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
H2O	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X		X	
H2OPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X		X	
H2ONormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
H2OQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
ClONO2	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
ClONO2Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
ClONO2NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
ClONO2Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
N2O5	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
N2O5Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
N2O5NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
N2O5Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
N2O	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X			
N2OPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X			
N2ONormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
N2OQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
NO2	(nLevels,nTimes)	REAL - 32 BIT	vmr	X			X	TES limb only
NO2Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X			X	TES limb only

HDF-EOS Name	***Dimension***	***Data Type***	***Units***(1)	***H***	***M***	***O***	***T***	***Notes***
NO2NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
NO2Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
CH4	(nLevels,nTimes)	REAL - 32 BIT	vmr	X			X	
CH4Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X			X	
CH4NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
CH4Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
HNO3	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X		X	TES limb only
HNO3Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X	X		X	TES limb only
HNO3NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
HNO3Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
CFC11	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
CFC11Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
CFC11NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
CFC11Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
CFC12	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
CFC12Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr	X				
CFC12NormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
CFC12Quality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
OH	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
OHPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
HO2	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
HO2Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
CO	(nLevels,nTimes)	REAL - 32 BIT	vmr		X		X	
COPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X		X	
HCN	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
HCNPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
RHI	(nLevels,nTimes)	REAL - 32 BIT	%rhi		X			
RHIPrecision	(nLevels,nTimes)	REAL - 32 BIT	%rhi		X			
HCl	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
HClPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
HOCl	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
HOClPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
ClO	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
ClOPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
BrO	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
BrOPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
IWC	(nLevels,nTimes)	REAL - 32 BIT	g/m3		X			

HDF-EOS Name	*Dimension*	*Data Type*	*Units*[(1)]	*H*	*M*	*O*	*T*	*Notes*
IWCPrecision	(nLevels,nTimes)	REAL - 32 BIT	g/m3		X			
GPH	(nLevels,nTimes)	REAL - 32 BIT	m		X			
GPHPrecision	(nLevels,nTimes)	REAL - 32 BIT	m		X			
SO2	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
SO2Precision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
CH3CN	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
CH3CNPrecision	(nLevels,nTimes)	REAL - 32 BIT	vmr		X			
Status	(nTimes)	INTEGER - 32 BIT	NoUnits		X			
Quality	(nTimes)	REAL - 32 BIT	NoUnits		X			
7.1MicronCloudAerosolFlag	(nLevels,nTimes)	INTEGER - 8 BIT	see Note 1 below	X				Type of cloud identified
7.1MicronExtinction	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
7.1MicronExtinctionPrecision	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
7.1MicronExtinctionNormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
7.1MicronExtinctionQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
8.3MicronCloudAerosolFlag	(nLevels,nTimes)	INTEGER - 8 BIT	see Note 1 below	X				Type of cloud identified
8.3MicronExtinction	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
8.3MicronExtinctionPrecision	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
8.3MicronExtinctionNormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
8.3MicronExtinctionQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
10.8MicronCloudAerosolFlag	(nLevels,nTimes)	INTEGER - 8 BIT	see Note 1 below	X				Type of cloud identified
10.8MicronExtinction	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
10.8MicronExtinctionPrecision	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
10.8MicronExtinctionNormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
10.8MicronExtinctionQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
12.1MicronCloudAerosolFlag	(nLevels,nTimes)	INTEGER - 8 BIT	see Note 1 below	X				Type of cloud identified
12.1MicronExtinction	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
12.1MicronExtinctionPrecision	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
12.1MicronExtinctionNormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
12.1MicronExtinctionQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
17.4MicronCloudAerosolFlag	(nLevels,nTimes)	INTEGER - 8 BIT	see Note 1 below	X				Type of cloud identified
17.4MicronExtinction	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
17.4MicronExtinctionPrecision	(nLevels,nTimes)	REAL - 32 BIT	1/km	X				
17.4MicronExtinctionNormChiSq	(nTimes)	REAL - 32 BIT	NoUnits	X				
17.4MicronExtinctionQuality	(nTimes)	INTEGER - 8 BIT	NoUnits	X				
CloudTopPressure	(nCloudTypes,nTimes)	REAL - 32 BIT	hPa	X				
CloudTopPressure	(nTimes)	REAL - 32 BIT	hPa				X	
CloudTopPressureError	(nTimes)	REAL - 32 BIT	hPa				X	

HDF-EOS Name	*Dimension*	*Data Type*	*Units*[(1)]	*H*	*M*	*O*	*T*	*Notes*
CloudEffectiveOpticalDepth	(nTimes)	REAL - 32 BIT	NoUnits				X	
CloudEffectiveOpticalDepthError	(nTimes)	REAL - 32 BIT	NoUnits				X	
ColumnAmountO3	(nXtrack,nTimes)	REAL - 32 BIT	DU			X		
TropoColumnAmountO3	(nXtrack,nTimes)	REAL - 32 BIT	DU			X		
ColumnAmountNO2	(nXtrack,nTimes)	REAL - 32 BIT	molec/cm2			X		
CloudFraction	(nXtrack,nTimes)	REAL - 32 BIT	NoUnits			X		
AerosolOpticalDepth	(nWavel,nLayers, nXtrack,nTimes)	REAL - 32 BIT	NoUnits			X		
FinalAerosolOpticalDepth	(nWavel,nXtrack,nTimes)	REAL - 32 BIT	NoUnits			X		
UVAerosolIndex	(nXtrack,nTimes)	REAL - 32 BIT	NoUnits			X		
VISAerosolIndex	(nXtrack,nTimes)	REAL - 32 BIT	NoUnits			X		
InstrumentConfigurationId	(nTimes)	UNSIGNED INT - 8 BIT	NoUnits			X		OMI Instrument Configuration ID
MeasurementQualityFlags	(nTimes)	UNSIGNED INT - 8 BIT	NoUnits			X		OMI Measurement Quality Flags
ErythemalDoseRate	(nXtrack,nTimes)	REAL - 32 BIT	Wm^2			X		Erythemally weighted dose rate at local Solar noon
ErythemalDailyDose	(nXtrack,nTimes)	REAL - 32 BIT	Jm^2			X		Erythemally weighted daily dose
CSErythemalDoseRate	(nXtrack,nTimes)	REAL - 32 BIT	Wm^2			X		Clear sky erythemally weighted dose rate at local Solar noon
CSErythemalDailyDose	(nXtrack,nTimes)	REAL - 32 BIT	Jm^2			X		Clear sky erythemally weighted daily dose
Reflectivity331	(nXtrack,nTimes)	REAL - 32 BIT	%			X		Reflectivity at 331 nm
VerticalResolution	(nLevels,nTimes)	REAL - 32 BIT	km				X	range: [0-1000] km
SpeciesRetrievalConverged	(nTimes)	INTEGER - 8 BIT	NoUnits				X	Species retrieval converged, 1=T=success
SpeciesRetrievalQuality	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
NumIterPerformed	(nTimes)	INTEGER - 8 BIT	NoUnits				X	Number of model iterations performed for retrieval of the particular species, range: [0-8] (small int.)
MaxNumIterations	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
DeviationVsRetrievalCovariance	(nTimes)	REAL - 32 BIT	NoUnits				X	
RadianceResidualMean	(nTimes)	REAL - 32 BIT	NoUnits				X	
RadianceResidualRMS	(nTimes)	REAL - 32 BIT	NoUnits				X	

HDF-EOS Name	***Dimension***	***Data Type***	***Units[(1)]***	***H***	***M***	***O***	***T***	***Notes***
RadianceResidualMax	(nTimes)	REAL - 32 BIT	NoUnits				X	
ConstraintVector	(nLevels,nTimes)	REAL - 32 BIT	VMR or K				X	
AveragingKernel	(nLevels,nLevels,nTimes)	REAL - 32 BIT	NoUnits				X	
TotalErrorCovariance	(nLevels,nLevels,nTimes)	REAL - 32 BIT	vmr2 or K2				X	
MeasurementErrorCovariance	(nLevels,nLevels,nTimes)	REAL - 32 BIT	vmr2 or K2				X	
ObservationErrorCovariance	(nLevels,nLevels,nTimes)	REAL - 32 BIT	vmr2 or K2				X	
ScanAveragedCount	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
DegreesOfFreedomForSignal	(nTimes)	REAL - 32 BIT	NoUnits				X	
InformationContent	(nTimes)	REAL - 32 BIT	NoUnits				X	
PixelsUsedFlag	(nTimes)	INTEGER - 64 BIT	NoUnits				A	
LandSurfaceEmissErrors	(nFreq,nTimes)	REAL - 32 BIT	NoUnits				A	TES Nadir only
RetrievedPointingAngle	(nTimes)	REAL - 32 BIT	deg				A	TES Limb only
RetrievedPointingAngleError	(nTimes)	REAL - 32 BIT	deg				A	TES Limb only
T_H2OCovariance	(nLevels,nLevels,nTimes)	REAL - 32 BIT	K*vmr				A	
T_H2OAveragingKernel	(nLevels,nLevels,nTimes)	REAL - 32 BIT	NoUnits				A	
H2O_TAveragingKernel	(nLevels,nLevels,nTimes)	REAL - 32 BIT	NoUnits				A	
Filter_Position_1A	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
Filter_Position_1B	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
Filter_Position_2A	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
Filter_Position_2B	(nTimes)	INTEGER - 8 BIT	NoUnits				X	
TotalError	(nLevels, nTimes)	REAL - 32 BIT	vmr or K				X	
Initial	(nLevels, nTimes)	REAL - 32 BIT	vmr or K				X	
AveragingKernelDiagonal	(nLevels, nTimes)	REAL - 32 BIT	NoUnits				X	
AirDensity	(nLevels, nTimes)	REAL - 32 BIT	Molec/m^3				X	

1 - Aura used their own naming conventions for units. Future missions may want to consider using the SI conventions

X – Field in standard file
A – Data items which will be carried in an ancillary file that is created only once per set of standard products

Note 1: 0=no contamination, 1=unknown cloud, 2=cirrus layer, 3=PSC, 4=saturated cloud, 99=anomaly detected

5.2 Grid Files

This section lists the specifics of each part of an HDF-EOS Grid file. Details are provided for the names of the grid(s), dimensions and data fields. Each field is then further specified with its exact name, dimensionality, data type and units.

5.2.1 Name of Grid

Varies by Instrument

The value for this name is not constrained. An end user can use HE5_SWinqgrid to determine what the values are for each grid structure.

5.2.2 Grid Structure, Projection and Dimensions:

Aura instruments used the U.S. Geological Survey General Cartographic Transformation Package (GCTP) Geographic projection to store their gridded data. The GCTP_GEO projection code is 0. In the Grid files, YDim is latitude and XDim is longitude.

TES and MLS gridded output will be on spacings of 2 degrees in latitude and 4 degrees in longitude. The grid will span +/- 82 degrees in latitude. MLS longitudes will be specified as east longitude and go from 0 to 360 degrees. TES longitudes will also be east longitude, but they will be +/- 180 degrees. For both teams, the center is the origin of the grids.

The OMI gridded output will be on ¼ or $^{1}/_{8}$ degree latitude and longitude spacings. The grid will span +/- 90 degrees in latitude and +/- 180 degrees in longitude.

These are the actual dimensions of the Grid field quantities:

XDim	Dimension created automatically with the Grid interface. Corresponds to longitude for Geographic projection
YDim	Dimension created automatically with the Grid interface. Corresponds to latitude for Geographic projection
nLevels	Number of pressure levels
nCandidate	Number of candidate Level 2 observations for each Level 2G grid cell
nLayers	Number of atmospheric layers (OMI specific)
nWavel	Number of wavelengths (OMI specific)

5.2.3 Grid Fields*:*

This is data which describe the scientific measured quantities. They provide information to aid in describing the data's "location". While the Grid structure does not specify the difference between geolocation fields and data fields, we have separated them for the sake of documentation.

NOTE THE ORDER OF DIMENSIONS IS IN FORTRAN ORDER WITH THE FIRST DIMENSION BEING THE MOST RAPIDLY INCREMENTING. IN LANGUAGES WHERE THE LAST DIMENSION IS THE MOST RAPIDLY INCREMENTING DIMENSION, THE ORDER OF DIMENSIONS SHOULD BE REVERSED.

				Instrument				
HDF-EOS Name	***Dimension***	***Data Type***	***Units***[(1)]	***H***	***M***	***O***	***T***	***Notes***
Latitude	(YDim)	REAL - 32 BIT	deg	T	X	X	X	Range: [-82 to 82] OMI range: [-90 to 90]
Latitude	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg			G		
Longitude	(XDim)	REAL - 32 BIT	deg	T	X	X	X	range: [-180 to 180] – OMI,TES range:[0 to 360] – MLS
Longitude	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg			G		range:[-180.0 to 180.0]
Pressure	(nLevels)	REAL - 32 BIT	hPa	T	X		X	Pressure will be a superset of the UARS pressure levels (ordered from ground to space)
GroundPixelQualityFlags	(XDim,YDim,nCandidate)	UNSIGNED INT - 16 BIT	NoUnits			G		
PathLength	(XDim,YDim,nCandidate)	REAL - 32 BIT	NoUnits			G		secant(SolarZenithAngle) + secant(ViewingZenithAngle)
RelativeAzimuthAngle	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg(EastofNorth)			G		SolarAzimuthAngle + 180 - ViewingZenithAngle
SecondsInDay	(XDim,YDim,nCandidate)	REAL - 32 BIT	s			G		Time in Seconds since UTC midnight
SolarAzimuthAngle	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg(EastofNorth)			G		range: [-180 to 180]
SolarZenithAngle	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg			G		
SpacecraftAltitude	(XDim,YDim,nCandidate)	REAL - 32 BIT	m			G		Spacecraft Altitude above WGS84 Ellipsoid
SpacecraftLatitude	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg			G		Spacecraft Geodetic Latitude above WGS84 Ellipsoid
SpacecraftLongitude	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg			G		Spacecraft Geodetic Longitude above WGS84 Ellipsoid
TerrainHeight	(XDim,YDim,nCandidate)	UNSIGNED INT - 16 BIT	m			G		
ViewingAzimuthAngle	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg(EastofNorth)			G		range: [-180 to 180]
ViewingZenithAngle	(XDim,YDim,nCandidate)	REAL - 32 BIT	deg			G		
Time	(XDim,YDim,nCandidate)	REAL - 64 BIT	s			G		Time in TAI93 Units

1 - Aura used their own naming conventions for units. Future missions may want to consider using the SI conventions

X **–** Field in standard file
G - In OMI L2 Grid Files
T - TBD - At the current time, HIRDLS is not producing Gridded data, but may at a future date

5.2.4 Grid Data Fields:

This is the actual scientific data. This list is not meant to be exhaustive and contain all fields in individual team's data files. Rather it lists the fields where overlap between instrument teams occurs as well as other primary data fields.

NOTE THE ORDER OF DIMENSIONS IS IN FORTRAN ORDER WITH THE FIRST DIMENSION BEING THE MOST RAPIDLY INCREMENTING. IN LANGUAGES WHERE THE LAST DIMENSION IS THE MOST RAPIDLY INCREMENTING DIMENSION, THE ORDER OF DIMENSIONS SHOULD BE REVERSED.

HDF-EOS Name	***Dimension***	***Data Type***	***Units(1)***	***H***	***M***	***O***	***T***	***Notes***
Temperature	(XDim,YDim,nLevels)	REAL - 32 BIT	K	T	X		X	TES – possibility separate nadir obs by day/night, limb is not planned to be separated
TemperaturePrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	K	T	X			
O3	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X		X	
O3Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X			
O3Ascending	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
O3AscendingPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ColumnAmountO3	(XDim,YDim)	REAL - 32 BIT	DU			X		
H2O	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X		X	
H2OPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X			
HDO	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr				X	
N2O	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X			
N2OPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X			
HNO3	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X		X	TES – limb only
HNO3Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T	X			
ClONO2	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
ClONO2Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
N2O5	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
N2O5Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
NO2	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T			X	TES – limb only
NO2Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
CH4	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T			X	
CH4`Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
HNO3	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
HNO3Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				

HDF-EOS Name	***Dimension***	***Data Type***	***Units***[(1)]	***H***	***M***	***O***	***T***	***Notes***
CFC11	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
CFC11Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
CFC12	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
CFC12Precision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr	T				
OH	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
OHPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
OHAscending	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
OHAscendingPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
OHDescending	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
OHDescendingPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
CO	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X		X	
COPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
HCN	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
HCNPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
RHI	(XDim,YDim,nLevels)	REAL - 32 BIT	%rhi		X			
RHIPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	%rhi		X			
GPH	(XDim,YDim,nLevels)	REAL - 32 BIT	m		X			
GPHPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	m		X			
HCl	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
HClPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ClO	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ClOPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ClOAscending	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ClOAscendingPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ClODescending	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
ClODescendingPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	vmr		X			
IWC	(XDim,YDim,nLevels)	REAL - 32 BIT	g/m3		X			
IWCPrecision	(XDim,YDim,nLevels)	REAL - 32 BIT	g/m3		X			
UVAerosolIndex	(XDim,YDim)	REAL - 32 BIT	NoUnits			X		
Reflectivity331	(XDim,YDim)	REAL - 32 BIT	%			X		Effective Surface Reflectivity at 331 nm
AirMassFactor	(XDim,YDim,nCandidate)	REAL - 32 BIT	NoUnits			G		
CloudFraction	(XDim,YDim,nCandidate)	REAL - 32 BIT	NoUnits			G		
CloudPressure	(XDim,YDim,nCandidate)	REAL - 32 BIT	hPa			G		
ColumnAmountBrO	(XDim,YDim,nCandidate)	REAL - 32 BIT	molec/cm2			G		
ColumnAmountHCHO	(XDim,YDim,nCandidate)	REAL - 32 BIT	molec/cm2			G		
ColumnAmountOClO	(XDim,YDim,nCandidate)	REAL - 32 BIT	molec/cm2			G		

HDF-EOS Name	***Dimension***	***Data Type***	***Units*** (1)	***H***	***M***	***O***	***T***	***Notes***
ColumnAmountNO2	(XDim,YDim,nCandidate)	REAL - 32 BIT	molec/cm2			G		
ColumnAmountO3	(XDim,YDim,nCandidate)	REAL - 32 BIT	DU			G		
ColumnAmountSO2	(XDim,YDim,nCandidate)	REAL - 32 BIT	DU			G		
SO2Index	(XDim,YDim,nCandidate)	REAL - 32 BIT	NoUnits			G		
InstrumentConfigurationId	(XDim,YDim,nCandidate)	UNSIGNED INT - 8 BIT	NoUnits			G		OMI Instrument Configuration ID
MeasurementQualityFlags	(XDim,YDim,nCandidate)	UNSIGNED INT - 8 BIT	NoUnits			G		OMI Measurement Quality Flags
TerrainPressure	(XDim,YDim,nCandidate)	REAL - 32 BIT	hPa			G		
UVAerosolIndex	(XDim,YDim,nCandidate)	REAL - 32 BIT	NoUnits			G		

X – Field in standard file
G - In OMI L2G Grid Files
T - TBD - At the current time, HIRDLS is not producing Gridded data, but may at a future date

Just for information only - additional fields carried in the TES gridded product (but may be unique to TES and not part of the standard)

HDF-EOS Name	***Dimension***	***Data Type***	***Units***	***H***	***M***	***O***	***T***	***Notes***
TotalColumnDensity	(XDim,YDim)	REAL - 32 BIT	Molecules/cm3				X	TES – provided with each species w/ exception of Temperature
SurfaceTemperature_Day	(XDim, YDim)	REAL - 32 BIT	K				X	TES – used in Temperature daily product only
SurfaceTemperature_Night	(XDim,YDim)	REAL - 32 BIT	K				X	TES – used in Temperature daily product only
<species>DataCount	(XDim,YDim,nLevels)	INTEGER - 32 BIT	N/A				X	TES – used in 8 day, monthly products
<species>StdDeviation	(XDim,YDim,nLevels)	REAL - 32 BIT	N/A				X	TES – used in 8 day, monthly products
<species>Maximum	(XDim,YDim,nLevels)	REAL - 32 BIT	N/A				X	TES – used in 8 day, monthly products
<species> Minimum	(XDim,YDim,nLevels)	REAL - 32 BIT	N/A				X	TES – used in 8 day, monthly products
TotColDensDataCount	(XDim,YDim)	REAL - 32 BIT	N/A				X	TES – used in 8 day,

HDF-EOS Name	***Dimension***	***Data Type***	***Units***	***H***	***M***	***O***	***T***	***Notes***
								monthly products
TotColDensStdDeviation	(XDim,YDim)	REAL - 32 BIT	N/A				X	TES – used in 8 day, monthly products
TotColDensMaximum	(XDim,YDim)	REAL - 32 BIT	N/A				X	TES – used in 8 day, monthly products
TotColDensMinimum	(XDim,YDim)	REAL - 32 BIT	N/A				X	TES – used in 8 day, monthly products

1 - Aura used their own naming conventions for units. Future missions may want to consider using the SI conventions

5.3 Zonal Average Files

This section lists the specifics of each part of an HDF-EOS Zonal Average file. Details are provided for the names of the zonal average(s), dimensions and data fields. Each field is then further specified with its exact name, dimensionality, data type and units.

5.3.1 Names of Zonal Averages

Varies by Instrument
The value for this name is not constrained.

5.3.2 Structure and Dimensions*:*

MLS products have 2 degrees latitude band spacings. The OMI products have ¼ degree latitude spacings.

These are the actual dimensions of the Zonal Average field and data field quantities:

nLevels	Number of pressure levels
nLats	Number of latitudes

5.3.3 Zonal Average Fields:

This is data which describe the scientific measured quantities. They provide information to aid in describing the data's "location". While the Zonal Average structure does not specify the difference between geolocation fields and data fields, we have separated them for the sake of documentation. It should be pointed out that this structure is only available in HDF-EOS5.

NOTE THE ORDER OF DIMENSIONS IS IN FORTRAN ORDER WITH THE FIRST DIMENSION BEING THE MOST RAPIDLY INCREMENTING. IN LANGUAGES WHERE THE LAST DIMENSION IS THE MOST RAPIDLY INCREMENTING DIMENSION, THE ORDER OF DIMENSIONS SHOULD BE REVERSED.

				Instrument				
HDF-EOS Name	***Dimension***	***Data Type***	***Units[1]***	***H***	***M***	***O***	***T***	***Notes***
Latitude	(nLats)	REAL - 32 BIT	deg	T	X			
Pressure	(nLevels)	REAL - 32 BIT	hPa	T	X			Pressure will be a superset of the UARS pressure levels (ordered from ground to space)
SolarZenithAngle	(nLats)	REAL - 32 BIT	deg	T	X			
LocalSolarTime	(nLats)	REAL - 32 BIT	h	T	X			(hours)

1 - Aura used their own naming conventions for units. Future missions may want to consider using the SI conventions

X – Field in standard file
A – Data items which will be carried in an ancillary file that is created only once per set of standard products
T - TBD - At the current time, HIRDLS is not producing Gridded data, but may at a future date

.

5.3.4 Zonal Average Data Fields:

This is the actual scientific data. This list is not meant to be exhaustive and contain all fields in individual team's data files. Rather it lists the fields where overlap between instrument teams occurs as well as other primary data fields.

At least three data fields would be supplied for each product: Value, Standard Deviation and Data Count.

NOTE THE ORDER OF DIMENSIONS IS IN FORTRAN ORDER WITH THE FIRST DIMENSION BEING THE MOST RAPIDLY INCREMENTING. IN LANGUAGES WHERE THE LAST DIMENSION IS THE MOST RAPIDLY INCREMENTING DIMENSION, THE ORDER OF DIMENSIONS SHOULD BE REVERSED.

HDF-EOS Name	*Dimension*	*Data Type*	*Units*[(1)]	*H*	*M*	*O*	*T*	*Notes*
TemperatureAscending	(nLevels,nLats)	REAL - 32 BIT	K	T	X			
TemperatureDescending	(nLevels,nLats)	REAL - 32 BIT	K	T	X			
O3Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
O3Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
H2OAscending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
H2ODescending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
N2OAscending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
N2ODescending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
HNO3Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
HNO3Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T	X			
ClONO2Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
ClONO2Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
N2O5Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
N2O5Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
NO2Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
NO2Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
CH4Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
CH4Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
CFC11Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
CFC11Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
CFC12Ascending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
CFC12Descending	(nLevels,nLats)	REAL - 32 BIT	vmr	T				
OHAscending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
OHDescending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			

HDF-EOS Name	***Dimension***	***Data Type***	***Units***[(1)]	***H***	***M***	***O***	***T***	***Notes***
COAscending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
CODescending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
GPHAscending	(nLevels,nLats)	REAL - 32 BIT	m		X			
GPHDescending	(nLevels,nLats)	REAL - 32 BIT	m		X			
HCNAscending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
HCNDescending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
RHIAscending	(nLevels,nLats)	REAL - 32 BIT	%rhi		X			
RHIDescending	(nLevels,nLats)	REAL - 32 BIT	%rhi		X			
HClAscending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
HClDescending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
ClOAscending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
ClODescending	(nLevels,nLats)	REAL - 32 BIT	vmr		X			
IWCAscending	(nLevels,nLats)	REAL - 32 BIT	g/m3		X			
IWCDescending	(nLevels,nLats)	REAL - 32 BIT	g/m3		X			
<species>AscendingDataCount	(nLevels,nLats)	INTEGER - 32 BIT	N/A		X			
<species>DescendingDataCount	(nLevels,nLats)	INTEGER - 32 BIT	N/A		X			
<species>AscendingStdDeviation	(nLevels,nLats)	REAL - 32 BIT	N/A		X			
<species>DescendingStdDeviation	(nLevels,nLats)	REAL - 32 BIT	N/A		X			

1 - Aura used their own naming conventions for units. Future missions may want to consider using the SI conventions

X – Field in standard file
A – Data items which will be carried in an ancillary file that is created only once per set of standard products
T - TBD - At the current time, HIRDLS is not producing Gridded data, but may at a future date

6.0 Definition of Mandatory Attributes in File

The following attributes are ones that are mandatory to appear in the file. They are meant to provide additional information or to ease use of the data. For instance, while the date is provided in the attached metadata, the GranuleMonth/Day/Year are provided as a simpler interface to this information. Instrument teams may have additional attributes that they have specified.

6.1 File Level Attributes (HDF-EOS Global File Attributes)

This is information that helps to describe this particular data set. It can be useful in labeling plots, calculating dates, etc... These will be set via calls to he5_ehwrglatt (HE5_EHwriteglbattr for C users).

Note that HE5T_NATIVE_CHAR and HE5T_NATIVE_SCHAR can be used interchangeably for character strings. End users should not see a difference in accessing attributes of either of these types.

Attribute Name	*Data Type*	*Attribute Description*
InstrumentName	CHARACTER STRING	“HIRDLS”, “MLS”, “TES” or “OMI”
ProcessLevel	CHARACTER STRING	Processing Level --- “L2”, “L3” etc.
GranuleMonth	INTEGER - 32 BIT	Month of start of granule --- 1-12
GranuleDay	INTEGER - 32 BIT	Day of start of granule ---- 1-31
GranuleYear	INTEGER - 32 BIT	Year of start of granule ---- e.g... 2004
TAI93At0zOfGranule	REAL - 64 BIT	TAI time of 0z of granule
PGEVersion	CHARACTER STRING	Processing version

Level 3 products shall include the above attribute names and the additional following attribute names:

Attribute Name	*Data Type*	*Attribute Description*
OrbitNumber	INTEGER - 32 BIT	For ‘Daily’ and ‘Global Survey’ product: an array containing 16 orbit numbers per day For ‘Monthly’ product: an array containing first and last orbit numbers of that month
OrbitPeriod	REAL - 64 BIT	For daily product: an array containing 16 orbit periods For monthly product: an array containing first and last orbit numbers of that month
Period	CHARACTER STRING	Type of product ---- “Daily”, “ Monthly”, “8-day”, “Global Survey”

6.2 HDF-EOS Group Attributes for Each Data Type

6.2.1 Swath Level Attributes

This is information which helps describe the swath to which it is attached. These will be set via calls to he5_swwrattr (HE5_SWwriteattr for C users).

Attribute Name	*Attribute Type*	*Attribute Description*
Pressure*	REAL - 32 BIT	pressure levels
VerticalCoordinate	CHARACTER STRING	“Pressure”, “Altitude”, “Potential Temperature”, “Total Column”, "Slant Column"

* This attribute is an exact duplicate of the Pressure geolocation field. Writing the pressure data in two locations was agreed upon as a compromise between instrument teams. This attribute is only mandatory if VerticalCoordinate is "Pressure".

6.2.2 Grid Level Attributes

This is information which helps describe the grid to which it is attached. These will be set via calls to he5_gdwrattr (HE5_GDwriteattr for C users).

Attribute Name	***Attribute Type***	***Attribute Description***
Projection	CHARACTER STRING	"Geographic", "Simple Cylindrical"
GridOrigin*	CHARACTER STRING	"Center"
GridSpacing	CHARACTER STRING	"(4,2)","(0.25,0.25)"
GridSpacingUnit	CHARACTER STRING	"deg"
GridSpan	CHARACTER STRING	"(0,360,-82,+82)","(-180,180,-90,90)"
GridSpanUnit	CHARACTER STRING	"deg"

* GridOrigin refers to the location where the grid cell's coordinates are located.

6.2.3 Zonal Average Level Attributes

This is information which helps describe the zonal average to which it is attached. These will be set via calls to he5_zawrattr (HE5_ZAwriteattr for C users).

Attribute Name	***Attribute Type***	***Attribute Description***
Pressure*	REAL - 32 BIT	pressure levels
VerticalCoordinate	CHARACTER STRING	"Pressure", "Altitude", "Potential Temperature", "Total Column", "Slant Column"
ZonalSpacing	CHARACTER STRING	"2"
ZonalSpacingUnit	CHARACTER STRING	"Degree"

* This attribute is an exact duplicate of the Pressure zonal average field. Writing the pressure data in two locations was agreed upon as a compromise between instrument teams. This attribute is only mandatory if VerticalCoordinate is "Pressure".

6.3 Geolocation and Data Field Attributes (HDF-EOS Local Attributes)

This is information that helps to describe the individual data fields. Data field attributes are a feature which can be useful in annotating plots as well as describing the data product to input routines. If ScaleFactor and Offset are not applicable they may be omitted. These attributes will be set via calls to he5_swwrlattr (HE5_SWwritelocattr for C users) if using swath format, he5_gdwrlattr (HE5_GDwritelocattr for C users) if using grid format, and he5_zawrlattr (HE5_ZAwritelocattr for C users) if using Zonal Average format.

Attribute Name	***Attribute type***	***Attribute Description***
MissingValue	Same type as Data Field	Contains the value for missing data
Title	CHARACTER STRING	For labeling a plot or axis
Units	CHARACTER STRING	Labeling units (for labeling color bars, converting between units, etc). After applying scale and offset, if applicable. The units are specified in Tables 5.3 and 5.4.
UniqueFieldDefinition[1]	CHARACTER STRING	Describes if definition of field is shared with other Aura Instruments ("Aura-Shared", "X-Specific", where X=Instrument Name, "X-Y[-Z]-Shared" where X,Y, and optional Z are instrument names (in alphabetical order)
ScaleFactor[2]	REAL - 64 BIT	Factor for scaling data (mandatory only if applicable)

Offset[2]	REAL - 64 BIT	Value to add to the data (mandatory only if applicable)

In addition to the attributes listed above, the *FillValue* attribute is recommended. It is created via the optional call to he5_SWsetfill (HE5_SWfillvalue for C users) if using swath format, he5_GDsetfill (HE5_GDfillvalue for C users) if using grid format, he5_ZAsetfill (HE5_ZAfillvalue for C users) if using zonal average format. Its value can be recovered by a call to he5_SWgetfill (HE5_SWgetfillvalue for C users) if using swath format, he5_GDgetfill (HE5_GDfillvalue for C users) if using grid format, he5_ZAgetfill (HE5_ZAfillvalue for C users) if using zonal average format. If it is used, its attribute type and value must be the same as the MissingValue attribute. Its literal name is set automatically and is not under the control of the instrument teams. In the form implemented by the HDFEOS library at the time of this writing that literal name is *_FillValue*.

[1]UniqueFieldDefinition is used to indicate to end-users if data from different instruments can be considered to have the same definition. If X-Specific is set, then instrument X has a unique definition of this field. If X-Y-Shared is set, then Instruments X and Y are using the same definition for this field. "Aura-Shared" indicates the same definition is used for all Aura instruments. Note that definitions can be shared even if dimensionalities are different.

[2]ScaleFactor and Offset utilize the following formulas:

When storing the data: DATA_FILE_VALUE = (SCIENCE_VALUE – Offset) / ScaleFactor
When reading the data: SCIENCE_VALUE = DATA_FILE_VALUE * ScaleFactor + Offset

7.0 Authors' Addresses

Cheryl Craig
National Center for Atmospheric Research
1850 Table Mesa Dr.
Boulder, CO 80303 USA
cacraig@ucar.edu

Pepijn Veefkind
KNMI
P.O. Box 201
NL-3730 AE De Bilt The Netherlands
veefkind@knmi.nl

Peter Leonard
ADNET Systems, Inc.
7515 Mission Dr., Suite A1C1
Lanham, MD 20706 USA
pleonard@sesda2.com

Paul Wagner
Mail Stop 183-701, Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91109 USA
Paul.A.Wagner@jpl.nasa.gov

Christina Vuu
Raytheon Information Solutions
299 N. Euclid Ave, Suite 500
Pasadena, CA 91733 USA
cvuu@mls.jpl.nasa.gov

Doug Shepard
Mail Stop 169-315, Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91733 USA
Douglas.H.Shepard@jpl.nasa.gov

A Appendix - ESDT LongNames and Metadata

ESDT LongNames are the most visible identifiers of science data collections in the archives, and so it is very useful if the LongNames carry a lot of identifying information. Aura LongNames will include the following information where applicable. The Sensor/Platform field will be first and instrument teams will determine the ordering of the other fields.

Sensor/Platform	(HIRDLS/Aura | MLS/Aura |OMI/Aura |TES/Aura)
File identification	(free text that identifies the file, indicating the Geophysical Parameter or type of data, may include vertical resolution: Total Column | Profile | TOA)
Operating Mode	(Zoomed | Science Mode|SpecialObservation)
Temporal Resolution	(5-Min | 1-Orbit | Daily | 8-Day | Monthly)
Processing Level	(L1B | L2 | L2G| L3 | L3e| L4 | Level 0 | Level 1 | Level2)
Spatial Coverage	(Swath | Global)
Spatial Resolution	(13x24km | 52x48km | 500km | 0.25deg | 1deg | 2x4deg)
Grid Scheme	(TBD when L3 and L4 products defined)

Examples of Aura ESDT LongNames (ShortNames are included for clarity):

ShortName	LongName
HIRDLS1	HIRDLS/Aura L1 Radiance Scans Science Mode Daily Global
HIRDLS2	HIRDLS/Aura L2 Vertical Profiles Science Mode Daily Global
HIRDLS3	HIRDLS/Aura L3 Lat Long Press Gridded Parameters Science Mode Daily Global
MLS0SCI1	MLS/Aura Level 0 Science Data APID=1744
ML1RADD	MLS/Aura Level 1 Radiances from DAACS Channels
ML2CLO	MLS/Aura Level 2 Chlorine Monoxide (ClO) Mixing Ratio
ML3DCLO	MLS/Aura L3 Daily Map of Chlorine Monoxide (ClO) Mixing Ratio Global 2x4 deg
ML3MMAPS	MLS/Aura L3 Monthly Maps for Standard Products Global 2x4 deg
OMBRO	OMI/Aura BrO Total Column 1-Orbit L2 Swath 52x48km
OML1BCAL	OMI/Aura Calibration Coefficients 1-Orbit L1B Swath
OMCLDO2	OMI/Aura Cloud Top Pressure (O2-O2 Absorption) 1-Orbit L2 Swath 13x24km
OML1BRUG	OMI/Aura Geolocated Earth Radiance UV 1-Orbit L1B Swath 13x24km
OML1BRUZ	OMI/Aura Geolocated Earth Radiance UV Zoomed 1-Orbit L1B Swath 13x12km
OML1BOPF	OMI/Aura Operational Parameters L1B
OMPROO3	OMI/Aura Ozone Profile 1-Orbit L2 Swath 13x24km
OMDOAO3	OMI/Aura Ozone Total Column DOAS 1-Orbit L2 Swath 13x24km
OMTROO3	OMI/Aura Ozone Tropospheric 1-Orbit L2 Swath 500x500km
OMSO2	OMI/Aura SO2 Total Column 1-Orbit L2 Swath 52x48km
OML1BIRR	OMI/Aura Solar Irradiance Daily L1B Swath 13x24km
OMTO3G	OMI/Aura Ozone (O3) Total Column Daily L2 Global 0.25deg Lat/Lon Grid
OMTO3d	OMI/Aura Ozone (O3) Total Column Daily L3 Global 1x1.25deg Lat/Lon Grid
OMTO3e	OMI/Aura Ozone (O3) Total Column Daily L3e Global 0.25deg Lat/Lon Grid
TL1BL	TES/Aura L1B Spectra Limb
TL1BN	TES/Aura L1B Spectra Nadir
TL2CH4L	TES/Aura L2 CH4 Limb
TL2CH4NS	TES/Aura L2 CH4 Nadir Special Observation
TL2O3N	TES/Aura L2 O3 Nadir
TL2ANC	TES/Aura L2 Ancillary Product
TL2SUM	TES/Aura L2 Summary Product

ESDT LongNames cannot exceed 80 characters because of the database hard limit. If more than 80 characters are needed for a LongName, then other metadata attributes can be used as an "extension" of the LongName, both at the Collection as well as at the Inventory and Archived metadata levels.

It is recommended that the ESDT LongName be saved in both the Collection and Archived metadata. The latter causes the LongName to appear in each product file. Setting Data_Location = MCF in both the Collection and Archived parts of the descriptor file is instrumental in making the two LongNames match.

B Appendix - Aura File Names

B.1 General Rules

File names traditionally have two parts – basis and suffix. The basis part of the file name is that which precedes the *period* before the suffix and will be discussed in more detail below. The suffix follows a *period* and identifies the type of file, and some software will assume certain behavior for a given suffix name. Examples: “txt” indicates a text file with only printable ASCII characters, “exe” indicates an executable, “doc” indicates a Microsoft Word document, and “jpg” indicates a particular kind of displayable image.

In most cases, file names are equivalent to LocalGranuleID. Ideally, the LocalGranuleID should be equal to the basis of the file name, but that is not what has been implemented. As an example, the Toolkit will add “.met” to the name that has been used for the data file name. For example, if the file name was “Granule1234.dat”, then the Toolkit will name the associated metadata file with the name “Granule1234.dat.met”. What should have been implemented was to name the metadata file “Granule1234.met”, but changing this is beyond the scope of this document. For all other Aura cases, we hope that there will only be one period and one suffix.

The file names should do several things. They should:

1) uniquely identify products
2) make it easy to write scripts/programs to parse the names.
3) use a scheme that will list things in the order that follows data organization or identification.
4) make it people friendly – shorter names may satisfy the first three, but longer names will make the names more readable by people, especially for those less familiar with NASA, EOS, and Aura.

File names must be compatible with Unix, Linux, and Windows file names. Aura will use only alpha-numeric characters plus *underscore*, *dash*, and *period*. *Underscores* will be used to delimit sections and the *period* only to delimit the suffix. *Dashes* can be used to delimit sub-fields within a section. Note that Unix and Linux are case sensitive, however Windows is not. Therefore case should not be used to create unique file names because when files are transferred to a Windows platform any uniqueness due to case will disappear. For consistency and continued readability, once a file is given its name, it should maintain its original case (character by character) for as long as it exists. That is, when a file is moved to the Windows environment, the case should not be changed. More importantly, the archive ingest and export processes should not change the case of any character when the file name is used as part of the LocalGranuleID.

For those characters that stand alone amongst numbers, it is recommended that those letters be kept in lower case except for the letter “L” that looks very much like the number “1”.

File names must not be longer than the most restrictive limits set by the operating systems (Linux, Unix, and Windows) and by the archive center. According to this, a string composed of the path, file name, and suffix ".met" must not exceed 256 characters in length.

B.2 Specific Rules

Aura file names will have 4 sections within the basis of the file name. Each section will be delimited by an *underscore*. The suffix will follow the basis and be delimited by a period. The four sections in the basis are Instrument ID, Data Type, Version, and Data ID. The order of the four sections has some flexibility, but all four instrument teams have agreed that the first two (Instrument ID and Data Type) should always be the first two. Version and DataID can be interchangeable and are easily identifiable by the beginning character. Version field will always begin with the character “v” and DataID can begin with any other alpha-numeric character.

```
<InstrumentID>_<DataType>_<Version>_<DataID>.<Suffix> or
```

```
<InstrumentID>_<DataType>_<DataID>_<Version>.<Suffix>
```

Instrument ID:

Instrument ID should contain the name or acronym of the instrument followed by a *dash* and the name of the platform. The instrument name precedes the platform name to make the instrument name more noticeable in the string – this is a people friendly feature. The platform name, although not necessary, helps in identifying this particular mission against past and future missions. Example: MLS was also on UARS and the hope is that there is a next generation MLS on a future platform. Without the "-Aura", some might confuse TES for a more famous cousin that traveled to Mars.

<Instrument>-<Platform>

Examples

HIRDLS-Aura
MLS-Aura
OMI-Aura
TES-Aura

Data Type:

Data Type will identify level of processing, species, look angle (nadir or limb), and other subgroups. To allow flexibility for the various instrument parameters, we propose that the names can be composed of the primary data type followed by optional sub-types that can be repeated as needed and separated by a *dash.*

`<Primary>[&-<subtype>]` & indicates this field can be repeated.

Examples:

For HIRDLS Level 2, there is only one type, so they will choose to use "L2"
For MLS Level 2, there are two Level 2 Primary types (L2AUX and L2GP) and multiple subtypes that specify the various species.
For OMI, there are multiple Level 2 subtypes that specify the various species.
For TES Level 2 products, there are three levels of information for Data Type – Level, species, and look. An example is L2-O3-Limb.

HIRDLS Names for Data Type	
Primary	**Subtype**
L1	
L2	
L3	

MLS Names for Data Type	
Primary	**Subtype**
L1BRADD	
L1BRADG	
L1BRADT	
L1BOA	
L2GP	BrO
L2GP	CH3CN
L2GP	ClO
L2GP	CO
L2GP	GPH
L2GP	H2O

L2GP	HCl
L2GP	HCN
L2GP	HNO3
L2GP	HO2
L2GP	HOCl
L2GP	N2O
L2GP	O3
L2GP	OH
L2GP	RHI
L2GP	SO2
L2GP	Temperature
L2GP	DGG
L2GP	IWC
L2AUX	DGM
L3DM	ClO
L3DM	CO
L3DM	GPH
L3DM	H2O
L3DM	HCl
L3DM	HCN
L3DM	HNO3
L3DM	N2O
L3DM	O3
L3DM	OH
L3DM	RHI
L3DM	Temperature
L3DM	IWC
L3DZ	Diagnostics
L3DZ	Standard
L3MM	Diagnostic
L3MM	Standard
L3MZ	Diagnostic
L3MZ	Standard

OMI Names for Data Type	
Primary	**Subtype**
L1	OML1BCAL
L1	OML1BOPF
L1	OML1BRUG
L1	OML1BRVG
L1	OML1BRUZ
L1	OML1BRVZ
L1	OML1BIRR
L2	OMDOAO3
L2	OMTO3
L2	OMPROO3
L2	OMTROO3
L2	OMAERO
L2	OMCLDRR
L2	OMCLDO2
L2	OMNO2
L2	OMBRO

L2	OMOCLO
L2	OMHCHO
L2	OMSO2
L2	OMTO3Z
L2	OMDOAO3Z
L2	OMPROO3Z
L2	OMTROO3Z
L2	OMAEROZ
L2	OMCLDRRZ
L2	OMCLCO2Z
L2	OMNO2Z
L2	OMBROZ
L2	OMOCLOZ
L2	OMHCHOZ
L2	OMSO2Z
L2G	OMTO3G
L3	OMTO3d
L3	OMTO3e

TES Names for Data Type		
Primary	**Subtype**	**Look**
L1B		Nadir
L1B		Limb
L2	H2O	Nadir
L2	H2O	Limb
L2	O3	Nadir
L2	O3	Limb
L2	CH4	Nadir
L2	CH4	Limb
L2	CO	Nadir
L2	CO	Limb
L2	NO2	Limb
L2	NO	Limb
L2	HNO3	Limb
L2	ATM-TEMP	Nadir
L2	ATM-TEMP	Limb
L2	ANCILLARY	
L2	SO	
L2	SUMMARY	
L3	ATM-TEMP	
L3	H2O	
L3	O3	
L3	CO	
L3	CH4	
L3	HDO	
L3	HNO3	
L3	NO2	8D
L3	ATM-TEMP	M
L3	H2O	M
L3	O3	M
L3	CO	M
L3	CH4	M

L3	NO2	M

Version:

The Version section allows the ability to identify the operational environment of the data. The parameters that are often used to identify this include software version, production identification (cycle number or production date-time), and/or a change in input configurations. Cycle is an identifier that helps to uniquely distinguish a granule. The uniqueness may be due to input data change or anything that causes the data to be re-processed using the same software version. In the case of OMI, the value for cycle uses the production date-time. For MLS and HIRDLS, the value for cycle is the letter c followed by a two digit number, while for TES it is a two digit number.

If the change is due to data format such that change is required of viewer and other software, flagging this information in the file name could be accomplished in several ways. One is to increment the major portion of the identifier. Another is to use an optional field as indicated below. The format field might be more clearly identifiable if it were prefixed with the character “f”. If the change in format is radical as changing from HDF4 to HDF5, this information is encoded in the suffix and would be redundant in the file name.

v<major>[&-<minor>][-<cycle>][-<format>]

HIRDLS plans to use software version with the following rule:

v<major>-<minor>-<InputDataChange>-c<cycle> with two digits for each.
E.g. v01-02-01-c01

MLS plans to use software version with the following rule:

v<major>-<minor>-c<cycle> with two digits for each.
E.g. v01-02-c01

OMI plans to use the archive center collection version plus production date-time.

v<major>-<ProductionDateTime>
E.g. v007-2002m0503t015937

TES plans to use the archive center collection version plus two digits specifying the format version followed by two digits specifying the content version.

F<major>-<minor>
E.g. F01_02

Data ID:

Data ID is the section that uniquely identifies the data based on attributes of the data such as date, time, location, orbit, frame, run identifier, sequence count, or some other changing parameter that distinguishes it from its neighboring data granules. For several of the instruments on Aura, the granularity of the files is daily, so using date is adequate. If date and/or time is used, please see the section below about rules for specifying date and time. If the Data ID requires a range of date and time, two date and time specifiers separated by a dash can indicate a time range. In some cases two or more specifiers may be required to uniquely identify the data. For TES and OMI, there are other parameters that are used in addition to or in place of date/time. They are run identifier and orbit. We propose that they are distinguished from time by using the following rule:

r<RunID> - where RunID is a sequence of alpha numeric characters, for TES this is 10 characters in length and zero padded

o<orbit> - where orbit is a 5 digit number that counts from begin of mission

tile<tile> - where tile is an alphanumeric ID of a static geographic area. Tiles that cover Earth will be defined when L3 products are defined.

zone<zone> - where zone is an alphanumeric ID of a static geographic area. Zones are bigger than tiles.

Rules about Date and Time:

<yyyy> - year, four digits, no leading Y
d<ddd> - day of year, three digits
m<mm[dd]> - month, two digits and optional 2 digits for day of month
t<hhmmssfffff> - time with 2 digits for hour, 2 for minutes, 2 for seconds and trailing digits for fractional seconds. Any trailing digits that are missing are assumed to be set to zero. Obviously, leading digits cannot be missing.

E.g.
- t01 – specifies 01:00:00.00000
- t0001 – specifies 00:01:00.00000
- t01020399999 – specifies 01:02:03.99999
- 2002d123 – specifies day 123 of year 2002
- 2002m01 – specifies January of 2002
- 2002m0101 – specifies January 1, 2002
- 2002d123t0102 – specifies year 2002, day of year 123, time 01:02:00.00000
- 2002d123-2002d127 – specifies a range of days including day 123 through 127

Suffix:

The following are the basic suffixes for Aura.
- h5 – HDF5
- he5 – HDF-EOS5
- met – metadata
- h4 – HDF4
- he4 – HDF-EOS2
- txt – ASCII text file
- dat – generic data file, recommend that this not be used for standard products

Examples:
- TES-Aura_L2-O3-Nadir_rnnnnnnnnnn_Fff_cc.he5
- TES-Aura_L2-O3-SO-Nadir_rnnnnnnnnnn_Fff_cc.he5
- TES-Aura_L2-ANCILLARY_rnnnnnnnnnn_Fff_cc.he5
- TES-Aura_L2-SUMMARY_rnnnnnnnnnn_Fff_cc.he5
- HIRDLS-Aura_L2_v01-02-01_2002d253.he5

Examples of MLS file names:
- MLS-Aura_L1BRADD_v01-03-c01_2002d123.h5
- MLS-Aura_L2GP-O3_v01-00-c01_2004d253.he5
- MLS-Aura_L3DM-Temperature_v01-02-c01_2002d123.he5
- MLS-Aura_L3MM-Standard_v01-02-c01_2002m02.he5

Examples of OMI file names:
- OMI-Aura_L1-OML1BRVG_2004m0523t0732-o01696_v002-2004m0526t123012.he4
- OMI-Aura_L2-OMTO3_2004m0523t0732-o01696_v002-2004m0526t123259.he5
- OMI-Aura_L2-OMTO3_2004m0523t0732-o01696_v002-2004m0526t123259.he5.met
- OMI-Aura_L2-OMPROO3_2004m0523t0732-o01696_v002-2004m0526t124354.he5
- OMI-Aura_L2-OMDOAO3_2004m0523t0732-o01696_v002-2004m0526t124139.he5
- OMI-Aura_L2-OMPROO3_2004m0523t0732-o01696_v002-2004m0526t023411.he5
- OMI-Aura_L2-OMPROO3_2004m0523t0732-o01696_v003-2004m0526t232034.he5.met
- OMI-Aura_L2-OMTROO3_2004m0523t0732-o01696_v002-2004m0526t124700.he5
- OMI-Aura_L2-OMTROO3Z_2004m0523t0732-o01696_v002-2004m0526t124724.he5
- OMI-Aura_L2-OMPROO3Z_2004m0523t0732-o01696_v002-2004m0526t125901.he5
- OMI-Aura_L2G-OMTO3G_2006m0831_v002-2006m0901t181039.he5
- OMI-Aura_L3-OMTO3d_2006m0831_v002-2006m0902t151849.he5
- OMI-Aura_L3-OMTO3dtoz_2006m0831_v002-2006m0902t151849.txt

OMI-Aura_L3-OMTO3e_2006m0831_v002-2006m0902t152105.he5
OMI-Aura_L3-OMTO3etoz_2006m0831_v002-2006m0902t152105.txt

C Appendix - Glossary of acronyms

Acronym	Description
EOS	Earth Observing System
EOSDIS	Earth Observing Data and Information System
ESDT	Earth Science Data Type
HDF	Hierarchical Data Format
HDF5	Hierarchical Data Format Version 5.X
HDF-EOS	Hierarchical Data Format for the EOS mission
HDF-EOS5	Hierarchical Data Format for the EOS mission Version 5.X
HIRDLS	High Resolution Dynamics Limb Sounder
MCF	Metadata Configuration File
MLS	Microwave Limb Sounder
NASA	National Aeronautics and Space Administration
OMI	Ozone Monitoring Instrument
TES	Tropospheric Emission Spectrometer

D Appendix - References

Details on HDF5 can be found by visiting the web site: http://hdfgroup.com/
Details on HDF-EOS5 can be found by visiting the web site: http://www.hdfeos.org/