170-WP-004-002

ECS Browse Granule Description

White Paper

November 2000

Prepared Under Contract NAS5-60000

RESPONSIBLE AUTHOR

Larry Klein /s/ 11/30/00

Larry Klein Date
Siri Jodha S. Khalsa
EOSDIS Core System Project

RESPONSIBLE OFFICE

Robert Plante /s/ 11/30/00

Robert Plante Date
EOSDIS Core System Project

Raytheon Company
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

This document is a specification for browse granules in the EOSDIS Core System (ECS). The basic browse service enabled by following these guidelines is the display of pre-computed images associated with science data granules that have been identified through queries against the ECS archives. Text is also supported as a browse product. The purpose of this document is to help data providers produce browse products that will be compatible with this browse service. This document contains specifications for the format, organization, and syntax of a browse product. Browse is one of the common services ECS makes available on science data products held in its archives. Because this service must be capable of handling all the browse products supplied to it by the various instrument teams, the availability of these services will be dependent on those browse products being in a specified format. Thus, conformance to the specifications in this document will enable ECS to support this basic browse services on the products it archives. This specification also describes optional extensions of the basic requirements and suggests formats and additional metadata.

Keywords: browse, metadata, services

This page intentionally left blank.

Contents

This page intentionally left blank.

1. Introduction

The Earth Observation System Data and Information System (EOSDIS) will provide browse services to assist earth scientists seeking appropriate data for research and analysis. Most EOSDIS Standard Data Products will have browse granules associated with the data granules making up that product. The basic browse service that the EOSDIS Core System (ECS) will provide is the retrieval of pre-computed images associated with science data granules that have been identified through an external search and order tool, such as the EOSDIS Data Gateway (EDG). Text and tables will also be supported as browse products. Normally, browse granules will be generated at the same time as the data granules they are associated with. However, browse granules may also be added to the system at a later time.

EOSDIS defines browse to be a *rapidly-accessible, on-line aid to ordering* science data products. Browse is the service used in determining whether a data product is suitable for a given purpose. What we call a "Browse granule" is an HDF file containing images, text or tables used in making this determination.

One browse granule may serve as the browse product for multiple science data granules. A single cloud cover mask, for example, may serve as appropriate browse for all Level 1 and Level 2 data granules covering the same temporal and spatial domain. Also, a science data granule may have more than one browse granule associated with it. This allows data providers to supply browse granules tailored to different disciplines. However, since a browse granule may contain multiple images it is recommended that all browse information for any given science granule be contained within a single browse granule. Also, the EDG at present can support only one browse granule per science granule

Currently, the EDG is used to search science data granule inventories, select candidate granules for ordering, and browse these candidate granules to determine if they are suitable for the analysis intended. The purpose of this document is to help data providers in producing browse granules that will be compatible with the browse services in the EDG and any future interfaces that may be created to access data in the ECS archives. This document contains specifications for the format, organization, and syntax of a browse granule. The specific science content of browse granules is the responsibility of the science team associated with the science data product. The Earth Science Data and Information System Project (ESDIS) has surveyed Distributed Active Archive Centers (DAACs), EOS Instrument Teams (ITs), and related earth observation projects to determine their requirements for browsing and browse data. The specifications defined in this document are intended to help ECS satisfy those needs.

This page intentionally left blank.

2. Browse Services Provided by EOSDIS

Browse is one of the generic services EOSDIS makes available on science data products held in the DAAC's archives. This service, currently provided through the EDG, must be capable of handling all the browse granules supplied to it by the various instrument teams. Thus, the availability of these services will be dependent on those browse granules conforming to the specifications in this document. This specification also describes optional extensions of the basic requirements and suggests formats and additional metadata. These extensions could be used by future ECS releases, or by other clients accessing the ECS archives through external gateways, to provide advanced browse services.

2.1 Core Metadata for Browse Granules

The ECS Data Model (420-TP-015-002) has in the core metadata four attributes related to associated browse granules:

- *BrowseDescription*, a description of the character and contents of the browse granule (255 characters maximum);
- *BrowseSize* the size of the browse granule, in Megabytes
- *BrowseProductionDateTime* the date and time a Browse was produced; and
- *BrowsePointer* the location in the system of the browse granule.

BrowseDescription, BrowseSize, and BrowseProductionDateTime chacterize the browse product and are held in the inventory database tables for the generic browse Earth Science Data Type (ESDT). BrowsePointer, on the other hand, links a science granule with the browse product that represents it, and is held in the inventory tables for the each particular ESDT that has a browse associated with it.

It is not necessary for there to be any metadata associated with a browse product when it is inserted into ECS. BrowseSize is supplied by the system. BrowseProductionDateTime is also supplied by the system when the product is made inside of ECS. An external data provider supplying browse can optionally populate this attribute, along with a value for BrowseDescription, in a metadata (.met) file accompanying the browse product. A Metadata Configuration File (MCF) in Appendix B provides a template for ECS browse metadata. This MCF is best used in conjunction with the Science Data Processing Toolkit. See the SDP Toolkit Users' Guide (333-CD-004-002) for instructions on how to populate the MCF with metadata values, or how an output ASCII metadata file looks in comparison to the input MCF.

A value for BrowsePointer is supplied by Science Data Server when a browse granule is inserted, and is subsequently entered into the inventory metadata for the science granule. This pointer enables browse services to retrieve browse granules associated with selected science granules. If a browse granule is inserted in a separate operation some time after the science granule it is

associated with has been inserted, the location (UR in ECS parlance) of the science granule must be supplied at that time. See the SIPS Interface Document for more details on the protocols for supplying browse products from outside of ECS.

Browse granules are essentially metadata for the science data granules they describe. For this reason, any metadata related to the science content of the browse should be stored as metadata attributes of the science granule, not the browse granule. Appendix A provides conventions for supplying optional metadata for describing the various objects that can be contained in a browse granule.

2.2 Browse Services

The EOS Data Gateway is currently the only mechanism by which users can search and order data from the ECS archives. When the results set from a data search are displayed in the EDG any granule that has a browse product associated with it has a button which when selected brings up a display showing the contents of that browse product. The first image contained in the browse file is displayed, with an option to view any additional images. The images can also be downloaded in GIF or JPEG formats. If there is any metadata contained in the browse product it can also be viewed. A user is also given the option of ordering the browse granule which will then be delivered via ftp. Generic browse products are in HDF and can be viewed with any tool that can open and display HDF files.

This browse specification is intended to provide a data structure that fulfills the basic needs of browse services. The specification is intended to be easily extensible to include additional data types and browse requirements.

3. Structure of the Browse Granule

3.1 Overview

Images are the preferred form of browse products in ECS because images can be compressed before storage and they are easily displayed and readily understood. Images conform with the intended purpose of browse which is to provide rapidly-accessible, on-line *representations* of data. However, tables (arrays) and text are also acceptable forms of browse. These are most useful as adjuncts to images. Currently, the EDG cannot utilize tables as part of Browse.

The browse service supplied by the EDG requires that browse granules be in HDF (Hierarchical Data Format). HDF was selected by NASA as the standard format for EOSDIS data products. It was developed at the National Center for Supercomputing Applications (NCSA) to facilitate data access in heterogeneous computing environments (http://hdf.ncsa.uiuc.edu/).

All browse granules will reside on rapid-access file storage. In order for the Browse service to satisfy its purpose of supporting on-line, rapid-access to browse, the overall size of a single browse granule is limited to *one megabyte*. HDF supports JPEG and other lossy and lossless compression methods. The one megabyte limit is on the object size *after* compression has been applied.

Browse functionality that extends beyond simply displaying the contents of an HDF file, for example geolocation on browse images, requires adherence to additional standards and inclusion of additional metadata within the product. The following sections discuss these two options separately: generic HDF browse granules for simple services, and HDF-EOS browse granules where extended browse services are required.

3.2 Generic HDF Browse Granules

HDF defines a file organization that supports a variety of data objects including:

- 8-bit and 24-bit raster images and associated palettes;
- Multidimensional arrays of Scientific Data Sets (SDS);
- Attributes, which are text annotations associated with individual SDSs or with the file as a whole;
- Binary tables (Vdata); and
- Vgroup structures that allow the user create groupings of objects.

Browse granules should be composed of any combination of the HDF data objects illustrated in Figure 3-1. As mentioned above, the use of 8-bit (RIS8) or 24-bit (RIS24) raster images best support the intended purpose of providing a real-time, on-line aid to ordering. Extensions to basic images are also possible. *Text* may be used to provide more lengthy annotations than provided for

in the core metadata attribute *BrowseDescription*, or in the extended browse metadata. *Science Data Tables* would be used to provide either data values or to generate overlays, scatter plots, or linegraphs. Note, however, that the EDG Client does not support tables. If used in browse products, they will not be assessable to a user. Further discussion of possible HDF objects in browse products, as well as recommended associated metadata, is contained in Appendix A.

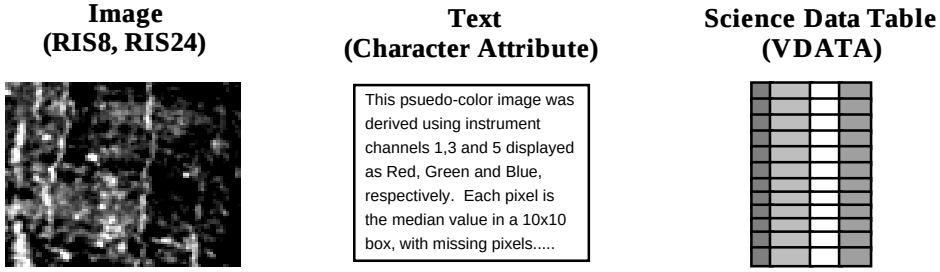

Figure 3-1. Browse Package Data Objects

The simplest form an ECS browse granule can take is an HDF file containing either an 8-bit (RIS8) or 24-bit (RIS24) raster image. Command-line utilities are available from NCSA for converting raw 8-bit or 24-bit images and JPEG files into generic HDF image files, with options to include an associated color palette. The recommended size is a 300 by 300 pixels. This pixel size is a good target for the size of browse images. Larger images are acceptable, within the 1Mb file size limit.

No geolocation services such as latitude-longitude under cursor location, overlay of coastlines, or a latitude-longitude graticule are provided on generic HDF browse granules. If overlays are desired in an image it is recommended that they be burned into the image itself. Appendix A suggests a method of encoding overlays using Vdatas in generic HDF files for use with possible future enhanced browse tools.

If the ECS Science Data Processing (SDP) Toolkit is being used to generate browse granules, the Toolkit's metadata routines can be used to write ECS core metadata attributes. In the Science Computing Facility (SCF) environment it can also be used to write any other product-specific metadata. The various kinds of metadata that can be written with the SDP Toolkit is described in the next section. A generic HDF file need not contain any ECS metadata. If a data producer wishes to include non-ECS metadata in browse granules we recommend that this information be attached as an HDF global text attribute using the SDsetattr function in HDF. This allows the EDG and the ECS-provided visualization tool EOSView to display the metadata written in this fashion.

3.3 HDF-EOS Browse Granules

HDF-EOS has been developed as a part of the EOSDIS Core System, with the intent that it provide data producers with a standard format for earth science data archives within ECS. It is built upon the HDF libraries produced by NCSA, and is fully compatible with HDF.

All HDF-EOS granules must contain ECS core metadata, written as an HDF global text attribute. The ECS Science Data Processing (SDP) Toolkit contains the software necessary to place these attributes into HDF-EOS file. Also needed is the Metadata Configuration File (MCF) for browse, included in Appendix B of this document, but normally provided by ECS Data Server during data production.

Browse granules are best created at the same time as the data granules they are associated with. Within ECS this would mean the SDP Toolkit was already in use and browse would be just another output of the product generation executive (PGE). For example, a data array could be subsampled for every 10th pixel, the resultant values scaled to the 0 - 255 range, then converted to an HDF RIS-8 object. This file could be output along with the necessary metadata to become the browse granule.

In addition to carrying ECS core metadata, science data products in HDF-EOS may contain data objects in any of three specified formats. These data objects, which are composites of standard HDF objects, are: Swath, Grid, and Point. These data types were created in order to provide a consistent method of storing and accessing geolocation information associated with the data product. ECS has built application program interfaces (APIs) which allow data products to be created and manipulated in ways appropriate to each data type, without regard to the actual HDF objects and conventions underlying them. The sum of these new APIs comprise the HDF-EOS library.

Data products created in any of the HDF-EOS data types, swath, grid or point, contain geolocation information in an HDF text attribute called *Structural Metadata*. All ECS services requiring geolocation information (e.g. subsetting by geographic region, pixel tracking on images, geographic overlays) require Structural Metadata to be present in the product. Thus, to enable geolocation-based browse services the swath, grid, or point data types must be used. Because these data types contain numeric data rather than images, the browse tools must convert these data into images in order for these data types to serve as visual browse. Although use of HDF-EOS data types for browse may have a negative impact on the performance of browse tools, it enables current and future browse services requiring geolocation information.

More information on HDF-EOS, and the ECS conventions associated with HDF-EOS, are available in the *HDF-EOS Users Guide for the ECS Project*, 170-TP-005-002 (Vol. 2) and 170-TP-005-003 (Vol. 1).

3.4 Browse Granule Format and Restrictions

An ECS Browse Granule will always contain one or more of the browse data objects from the list shown in Figure 3-1. Optionally, associated with each object will be entries in the Browse Metadata described in Appendix A.

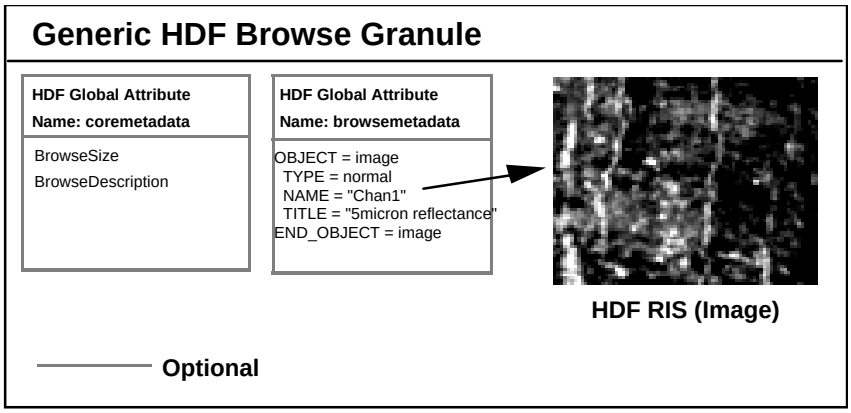

Figure 3-2. Generic Browse Granule Example

A schematic of a generic HDF browse granule is shown in Fig. 3-2. It consists of a single raster image in either RIS8 or RIS24. This example granule includes two optional HDF global attributes holding metadata associated with the browse granule. These are: *Coremetadata* containing ECS Core metadata (described more in the *SDP Toolkit Users Guide*), and *Browsemetadata* containing extended browse metadata (described more in Appendix A). The only core metadata that can be supplied for a browse granule are BrowseSize, BrowseProductionDateTime and BrowseDescription. Product-Specific Attributes that are searchable in the inventory database tables is not permitted for browse products since all browse are part of the same ECS collection, for which PSAs are not defined.

Extended Browse Metadata is optional. It can be written using the SDP Toolkit metadata routines, which will ensure that this metadata is in the proper ECS Object Description Language (ODL) format that ECS tools can read. At present, this is possible only with browse granules generated outside of ECS, since the Metadata Configuration File (MCF) supplied by ECS Data Server for browse granules can only contain the *Coremetadata* block in it. Currently, no ECS tools make explicit use is made of such extended browse metadata. Note that all metadata objects for generic HDF browse granules are optional.

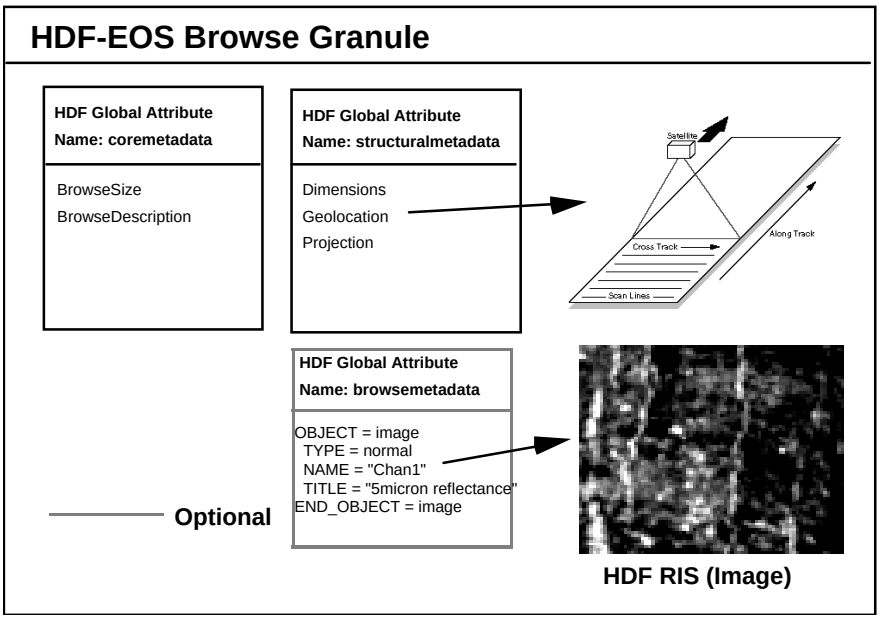

Figure 3-3. HDF-EOS Browse Granule Example

Figure 3-3 is a graphical representation of a Browse Granule implemented as an HDF-EOS file. This example granule includes three kinds of HDF global attributes holding metadata associated with the browse granule. In addition to *Coremetadata* and *Browsemetadata* described in the previous example, there is also *Structuralmetadata* containing the metadata describing the dimensions, components and geolocation information for the swath, grid and point data types. This metadata block is described in more detail the *HDF-EOS Users Guide*. Structural Metadata is required only if the HDF-EOS Swath, Grid, or Point data types are used. It is generated automatically by the HDF-EOS API.

This page intentionally left blank.

Appendix A - Extended HDF Browse

This appendix describes options for including additional objects and metadata within ECS Browse Granules (generic HDF or HDF-EOS). Currently, EOSDIS does not provide any functionality associated with these objects other than opening the HDF file and displaying them. They are included in this specification to allow EOSDIS, the DAACs, or third-party vendors to build future functionality upon them.

A.1 Browse Metadata

Browse granules are essentially metadata attributes themselves, used to characterize the science granule they are associated with. For this reason, any metadata related to the science content of the browse is more properly stored as metadata attributes of the science granule, not the browse granule. What we discuss in this section is the metadata that is needed to describe the browse objects that can be contained in a browse granule.

This section suggests a content and format for optional metadata associated with browse data objects. The browse information will consist of text in Object Description Language (ODL) written to an HDF global text attribute named *browsemetadata*. Each browse object (image, text, or table) within the browse granule should have a corresponding ODL object description. For example, the browse metadata for a granule containing nothing but a single raster image might look like:

```
GROUP = BROWSEMETADATA

GROUPTYPE = MASTERGROUP

OBJECT = image
 TYPE = normal
 NAME = "Earth"
 TITLE = "A Picture of the Earth"
END_OBJECT = image

END_GROUP = INVENTORYMETADATA
```

The `GROUP/END_GROUP`, and `GROUPTYPE` statements are ECS ODL conventions which are described in more detail in the SDP Toolkit Users Guide, Appendix J. The `OBJECT` name refers to the category of browse component. We suggest three: `image`, `text`, and `table`. These component categories are described in subsequent sections. The three parameters inside the `OBJECT/END_OBJECT` block in the above example are described are:

`TYPE` — This defines the subclass of object. For example, the type for an `image` object may be: `normal`, `legend`, or `animation`. We discuss these types in Section A.2.

NAME — This is the name of the HDF object that the metadata is referring to. For tables, this refers to the name of the Vdata that holds the table of values. For images and text, it refers to an attribute named "name" associated with that component.

TITLE — This is a presented title associated with all browse objects.

In addition, we specify four optional parameters:

DESCRIPTION — A more detailed text description of the browse object.

DEFAULT_RESOLUTION — For images, it refers to the resolution (in pixels) the image should first be displayed at, which is usually smaller than the stored pixel size of the image. Note that the actual pixel dimensions are not part of the browse metadata standard, since that information is already stored in the HDF file. It is up to the program displaying the image to scale the stored image appropriately.

LEGEND_IMAGE — Defines the image legend for the component. The image legend is a browse image of type "legend" that is meant to be displayed along with the component. The image legend is typically used for annotated color bars and the like. The field of this parameter refers to the name of the image legend.

ANIMATION_ORDER — Defines the sequence in which images will be displayed in animation. The lowest sequence number will be displayed first.

VECTOR_OVERLAY — Defines the vector overlay for the component. The vector overlay is a table of type "overlay" that contains columns of X-Y pairs. These values define vector outlines that are used to overlay the component image.

Other *user-defined* parameters of arbitrary names can also be included in the metadata. They may be useful for product-specific applications. If used inside an OBJECT = image, text or table block, these user-define parameters should refer just to that specific browse component. An example is shown below.

```
GROUP = BROWSEMETADATA

GROUPTYPE = MASTERGROUP

OBJECT = image
 TYPE = normal
 NAME = "Earth"
 TITLE = "A Picture of the Earth"
 IMAGE_QUALITY = "good"   /* User-defined parameter */
END_OBJECT = image

END_GROUP = INVENTORYMETADATA
```

The `name`, `title`, and `description` keywords can *also* be used to describe the *entire* browse granule, not associated with any component. An example is shown below of a browse structure with two raster images, and a `name` and `title` that is associated with the entire structure, not a particular component.

```
GROUP = BROWSEMETADATA

GROUPTYPE = MASTERGROUP

NAME = "Planet Browse" /* refers to entire granule */
TITLE = "Browse for Planet Pictures"

OBJECT = image
 TYPE = normal
 NAME = "Earth"
 TITLE = "A Picture of the Earth"
END_OBJECT = image

OBJECT = image
 TYPE = normal
 NAME = "Mars"
 TITLE = "A Picture of Mars"
END_OBJECT = image

END_GROUP = INVENTORYMETADATA
```

Note that ODL treats single quotes (') and double quotes ("). as equivalent. In fact, the quotes themselves are optional, and are used here primarily for fields that may contain spaces.

A.2 Browse Objects

This section discusses the three types of browse objects, `image`, `table`, and `text`, plus the classes and related sub-classes of metadata that can be used to describe them.

A.2.1 Images

The image class of browse data (`OBJECT = image`) is a raster image of either 8-bit pseudo-color (RIS8 in HDF terms) or 24-bit true color (RIS24 in HDF terms). We expect these images to be compressed, using the standard features of the HDF libraries. We have defined three kinds of images: `normal`, `legend`, and `animation`. They are designated by the "`type=`" keyword. The

only difference between the three types is the metadata associated with each: the data format and organization of the images themselves is identical.

We expect the producer of the browse granule to use standard HDF calls to associate an image palette with each 8-bit raster image. Since these browse images may be displayed on systems with less than 256 colors available for display, we suggest the following guidelines for the construction of these palettes. Also, the RIS interface provides for palettes, and lossy or lossless compression.

- Since the palette may be compressed to a smaller color range, it is best if the palette has no sharp transitions in it. This is because as a palette is compressed, the actual palette entry used for a particular raster image value may move by an entry or two. This is not a problem for palettes with smooth transitions such as grayscale or rainbow, but may effect the image appearance for palettes with sharp transitions.
- Some raster images may need to guarantee certain colors that are used for say burned in overlays or text. We therefore will guarantee that EOS developed display programs will always maintain the colors stored in entries 0 and 255 of the palette.

A `normal` image is just an 8-bit or 24-bit raster image. The associated metadata is just that described above.

A `legend` image is identical to a normal image. It is meant to be displayed along with other images contained within the browse file. These other images use the "`legend_image=`" parameter to establish the relationship with the legend image. This feature is typically used for annotated color bars and the like.

An `animation` image is actually one in a series of images which if desired can be viewed as an animation. The `animation_order` is specified as an integer. We put no limitations on the images contained within the animation grouping, although the displaying applications may have their own limitations, such as demanding that the images all be of the same type and size.

```
GROUP = BROWSEMETADATA

GROUPTYPE = MASTERGROUP

OBJECT = animation
 TYPE = normal
 NAME = "Earth1"
 TITLE = "A Picture of the Earth"
 ANIMATION_ORDER = 1
END_OBJECT = image

OBJECT = animation
 TYPE = normal
 NAME = "Earth2"
 TITLE = "Another Picture of the Earth"
 ANIMATION_ORDER = 2
END_OBJECT = image

END_GROUP = INVENTORYMETADATA
```

In the example above, the displaying program would show the Earth1 image first, then Earth2, in succession. Note that the sizes of each of these images is not listed here, as those sizes are stored inside the HDF information for each image.

A.2.2 Tables

A table is an HDF Vdata with associated browse metadata. An HDF Vdata structure consists of a collection of one dimensional arrays. Each array has its own name and number type, but they must all be of the same number of entries.

We define three kinds of tables: `normal`, `overlay`, and `plot`. As before, the type of table is designated by the "`type`" keyword. These three types of tables are described below.

A `normal` table is just a table of science values, stored as a Vdata. It may contain any number of named columns (1D arrays), each of a different number type. Note that the names, number types, and sizes of the fields in the normal table are not listed in the metadata. This is because that information is inherent in the corresponding Vdata structure.

An `overlay` table contains at least two columns (fields) of name "`ordinate`" and "`abscissa`". This series of X-Y points can be used to generate a vector overlay on an image that has used a "`vector_overlay=`" parameter to specify this table. The (1,1) value will be assigned to the lower left point of the displayed image. These columns must be of number type `integer`. You may optionally include a third column of name "`pen`". A value of "`1`" in this column is interpreted as a "Pen-Up" command for that X-Y pair. Again, the field names are not in the metadata description, because they are stored with the Vdata.

A `plot` table is meant to be displayed as a lineplot. At least one column (field) must be named "`X`". The displaying program will then plot the other columns in the plot table as a function of this first column.

A.2.3 Text

Text components of browse granules are meant to contain descriptive ASCII text that is too large to fit comfortably in the description keyword of the metadata. The text should be stored as a separate global text HDF attribute (the same way that the core, browse, and structural metadata are stored). If the text block is larger than 64K characters (the HDF limit on attributes), then additional global character attributes should be created with names "`NameofText.0`", "`NameofText.1`", and so on (where "`NameofText`" is arbitrary: it is the name that the text component will be referred to with the metadata). Note that if the SDP Toolkit is used to create this metadata attribute, the creation of additional blocks to accommodate text beyond the 64K limit is handled automatically.

The current specification supports only ASCII text as a browse component. This ASCII text can be flagged as being formatted one of three ways: `normal` (straight ASCII text), `ps` (PostScript), or `rtf` (Rich Text Format). The formatting method is designated by the "`type=`" keyword in the browse metadata. An example of the text metadata is shown below. Note that the name listed here must match up exactly with the name of the global character attribute containing the text, less the "`.0`", "`.1`" extensions.

```
GROUP = BROWSEMETADATA

GROUPTYPE = MASTERGROUP

OBJECT = text
 TYPE = rtf     /* rich text format */
 NAME = "NameofText"
 TITLE = "Title of Description"
END_OBJECT = text

END_GROUP = INVENTORYMETADATA
```

Appendix B - Browse Metadata Configuration File (MCF)

```
GROUP = INVENTORYMETADATA

GROUPTYPE = MASTERGROUP

  GROUP = CollectionDescriptionClass

   OBJECT = ShortName
    Data_Location = "MCF"
    NUM_VAL = 1
    TYPE = "STRING"
    Mandatory = "TRUE"
    Value = "BROWSE"
   END_OBJECT = ShortName

   OBJECT = VersionID
    Data_Location = "MCF"
    NUM_VAL = 1
    TYPE = "STRING"
    Mandatory = "TRUE"
    Value = "1"
   END_OBJECT = VersionID

  END_GROUP = CollectionDescriptionClass

  OBJECT = BrowseSize
   Data_Location = "MCF"
   NUM_VAL = 1
   TYPE = "FLOAT"
   Mandatory = "TRUE"
  END_OBJECT = BrowseSize

  OBJECT = BrowseProductionDateTime
   Data_Location = "TK"
   NUM_VAL = 1
   TYPE = "DATETIME"
   Mandatory = "TRUE"
  END_OBJECT = BrowseProductionDateTime

  OBJECT = BrowseDescription
   Data_Location = "MCF"
   NUM_VAL = 1
   TYPE = "STRING"
   Mandatory = "TRUE"
  END_OBJECT = BrowseDescription

END_GROUP = INVENTORYMETADATA
```

This page intentionally left blank.

Appendix C - AIRS Summary Browse Product

This appendix describes a customized browse product developed for the Atmospheric Infrared Sounder (AIRS) instrument package to be flown on the EOS *Aqua* platform. The product differs from the generic browse product described in the main body of this Specification in that the AIRS browse products are stored as a science granules rather than belonging to the generic browse ESDT. The AIRS Summary Browse Product will combine the output of many input data granules and even though it is a reduction of the original data set, it is treated as a standard product by ECS. The metadata associated with the product is therefore the same as standard ECS product metadata. Users will employ the AIRS Summary Browse Products in the ECS search and order interface (EDG Client) as an aid in locating the science granules they need. They will be able to perform spatial, temporal and parameter searches using the summary browse product. These searches will be made against the parent collection named in a Product Specific Attribute held in the AIRS Summary Browse Product.

C.1 Overview

The AIRS Summary Browse Product will consist of five Summary Browse Package types, for ascending or descending orbits giving a total of ten AIRS Summary Browse Package files produced once a day. The five AIRS Browse Package types (Table C-1) will be associated with AIRS processing levels and instrument outputs of the AIRS science granules. Each AIRS Summary Browse Package file will contain multiple images for display within the EOS Data Gateway (EDG).

The AIRS Summary Browse Packages will be produced daily in order to provide global coverage images. It is assumed that the AIRS Summary Browse Package PGEs will execute when processing of the AIRS science granules is complete for a given day. Each separate AIRS Summary Browse Package will be split into DAY (ascending) and NIGHT (descending) for a total of ten AIRS Summary Browse Package files a day.

The AIRS Summary Browse Package files will be stored as HDF files containing ECS metadata. The AIRS Summary Browse Package files will use the 8-bit raster image data model defined by the HDF format. Each of the raster images within an AIRS Browse Package files will have an associated color palette. Additional metadata will be provided at the file and object level in text format using the HDF AN API. Each AIRS Summary Browse Package file will contain Core Metadata generated via the SDP Toolkit. The Core Metadata will be that which is required as a minimum for science data products, plus one additional Product Specific Attribute (PSA). The PSA will contain the ShortName and VersionID of Collection of the parent science granule represented by the browse image. This PSA is required for ordering science granules using the AIRS Summary Browse paradigm within the EOS Data Gateway.

Table C-1. AIRS Summary Browse Product Description

AIRS Summary Browse Package Type	*Description*	*Number of Images*	*Image Parameters*
L1B_HSB	Raster images of brightness temperatures for selected HSB channels.	4	HSB Channel #1 , HSB Channel #2, HSB Channel #3, HSB Channel #4
L1B_AMSU	Raster images of brightness temperatures for selected AMSU channels.	8	AMSU Channel #1, AMSU Channel #2, AMSU Channel #3, AMSU Channel #4, AMSU Channel #5, AMSU Channel #6, AMSU Channel #7, AMSU Channel #15
L1B_AIRS	Raster images of AIRS radiances for selected AIRS channels	5	AIRS temperature sensing channel (200 mb), AIRS Ozone sensing channel, AIRS window channel, AIRS methane channel, AIRS water vapor sensing channel.
L2_CC	Raster images of AIRS Level 2 cloud cleared radiances for selected AIRS channels (same channels as L1B_AIRS)	5	AIRS temperature sensing channel (200 mb), AIRS Ozone sensing channel, AIRS window channel, AIRS methane channel, AIRS water vapor sensing channel.
L2_Retrievals	Raster images of selected AIRS Level 2 retrieved parameters.	9	cld_pct1, cld_pct2, SST_ret, H2OVapor_Burden, O3_Burden, VisPctClr, VisVarIdx, MW_FirstGuess_Tot_H2O, MW_Rain_Rate

The AIRS Summary Browse Package files will consist of 2-D 8-bit pseudo-color raster images whose pixels provide a spatial resolution of 1° latitude by 1° longitude. An uncompressed image of this size (360 pixels in longitude by 180 pixels in latitude) and color depth amounts to 65 KB. For uniformity of Browse, all images will be represented at this resolution.

C.2 Example Metadata Configuration File

In this section, we show the Metadata Configuration File for the AIRS L2 daily summary browse product (Cleared Radiances). Metadata configuration files for the other products in Table C.1 are similar.

```
GROUP = INVENTORYMETADATA
GROUPTYPE = MASTERGROUP
  GROUP = ECSDataGranule
    OBJECT = LocalGranuleID
      Mandatory = "TRUE"
```

```
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = LocalGranuleID
    OBJECT = SizeMBECSDataGranule
      Mandatory = "FALSE"
      Data_Location = "DSS"
      NUM_VAL = 1
      TYPE = "DOUBLE"
    END_OBJECT = SizeMBECSDataGranule
    OBJECT = ProductionDateTime
      Mandatory = "TRUE"
      Data_Location = "TK"
      NUM_VAL = 1
      TYPE = "DATETIME"
    END_OBJECT = ProductionDateTime
    OBJECT = DayNightFlag
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = DayNightFlag
    OBJECT = ReprocessingActual
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = ReprocessingActual
    OBJECT = LocalVersionID
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = LocalVersionID
  END_GROUP = ECSDataGranule
  GROUP = MeasuredParameter
    OBJECT = MeasuredParameterContainer
      Data_Location = "NONE"
      Mandatory = "TRUE"
      CLASS = "M"
      GROUP = QAFlags
        CLASS = "M"
        OBJECT = ScienceQualityFlag
          Mandatory = "FALSE"
          Data_Location = "DP"
          NUM_VAL = 1
          TYPE = "STRING"
        END_OBJECT = ScienceQualityFlag
        OBJECT = AutomaticQualityFlagExplanation
          Mandatory = "TRUE"
          Data_Location = "PGE"
          NUM_VAL = 1
          TYPE = "STRING"
        END_OBJECT = AutomaticQualityFlagExplanation
        OBJECT = AutomaticQualityFlag
          Mandatory = "TRUE"
          Data_Location = "PGE"
```

```
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = AutomaticQualityFlag
    OBJECT = OperationalQualityFlagExplanation
      Mandatory = "FALSE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = OperationalQualityFlagExplanation
    OBJECT = OperationalQualityFlag
      Mandatory = "FALSE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = OperationalQualityFlag
    OBJECT = ScienceQualityFlagExplanation
      Mandatory = "FALSE"
      Data_Location = "DP"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = ScienceQualityFlagExplanation
  END_GROUP = QAFlags
  GROUP = QAStats
    CLASS = "M"
    OBJECT = QAPercentMissingData
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = QAPercentMissingData
    OBJECT = QAPercentOutofBoundsData
      Mandatory = "FALSE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = QAPercentOutofBoundsData
    OBJECT = QAPercentCloudCover
      Mandatory = "FALSE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = QAPercentCloudCover
    OBJECT = QAPercentInterpolatedData
      Mandatory = "FALSE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = QAPercentInterpolatedData
  END_GROUP = QAStats
  OBJECT = ParameterName
    Mandatory = "TRUE"
    CLASS = "M"
    Data_Location = "PGE"
    NUM_VAL = 1
    TYPE = "STRING"
  END_OBJECT = ParameterName
END_OBJECT = MeasuredParameterContainer
```

```
END_GROUP = MeasuredParameter
GROUP = OrbitCalculatedSpatialDomain
  OBJECT = OrbitCalculatedSpatialDomainContainer
    Data_Location = "NONE"
    Mandatory = "TRUE"
    CLASS = "M"
    OBJECT = StartOrbitNumber
      Mandatory = "TRUE"
      CLASS = "M"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = StartOrbitNumber
    OBJECT = EquatorCrossingDate
      Mandatory = "TRUE"
      CLASS = "M"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "DATE"
    END_OBJECT = EquatorCrossingDate
    OBJECT = EquatorCrossingTime
      Mandatory = "TRUE"
      CLASS = "M"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "TIME"
    END_OBJECT = EquatorCrossingTime
    OBJECT = OrbitNumber
      Mandatory = "FALSE"
      CLASS = "M"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = OrbitNumber
    OBJECT = EquatorCrossingLongitude
      Mandatory = "TRUE"
      CLASS = "M"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "DOUBLE"
    END_OBJECT = EquatorCrossingLongitude
    OBJECT = StopOrbitNumber
      Mandatory = "TRUE"
      CLASS = "M"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "INTEGER"
    END_OBJECT = StopOrbitNumber
  END_OBJECT = OrbitCalculatedSpatialDomainContainer
END_GROUP = OrbitCalculatedSpatialDomain
GROUP = CollectionDescriptionClass
  OBJECT = VersionID
    Mandatory = "TRUE"
    Data_Location = "MCF"
    NUM_VAL = 1
    TYPE = "INTEGER"
    Value = 1
```

```
    END_OBJECT = VersionID
    OBJECT = ShortName
      Mandatory = "TRUE"
      Data_Location = "MCF"
      NUM_VAL = 1
      TYPE = "STRING"
      Value = "AIRI2DBR"
    END_OBJECT = ShortName
  END_GROUP = CollectionDescriptionClass
  GROUP = InputGranule
    OBJECT = InputPointer
      Mandatory = "FALSE"
      Data_Location = "PGE"
      NUM_VAL = 240
      TYPE = "STRING"
    END_OBJECT = InputPointer
  END_GROUP = InputGranule
  GROUP = SpatialDomainContainer
    GROUP = HorizontalSpatialDomainContainer
      GROUP = BoundingRectangle
        OBJECT = EastBoundingCoordinate
          Mandatory = "TRUE"
          Data_Location = "PGE"
          NUM_VAL = 1
          TYPE = "DOUBLE"
        END_OBJECT = EastBoundingCoordinate
        OBJECT = WestBoundingCoordinate
          Mandatory = "TRUE"
          Data_Location = "PGE"
          NUM_VAL = 1
          TYPE = "DOUBLE"
        END_OBJECT = WestBoundingCoordinate
        OBJECT = SouthBoundingCoordinate
          Mandatory = "TRUE"
          Data_Location = "PGE"
          NUM_VAL = 1
          TYPE = "DOUBLE"
        END_OBJECT = SouthBoundingCoordinate
        OBJECT = NorthBoundingCoordinate
          Mandatory = "TRUE"
          Data_Location = "PGE"
          NUM_VAL = 1
          TYPE = "DOUBLE"
        END_OBJECT = NorthBoundingCoordinate
      END_GROUP = BoundingRectangle
      GROUP = ZoneIdentifierClass
        OBJECT = ZoneIdentifier
          Mandatory = "FALSE"
          Data_Location = "MCF"
          NUM_VAL = 1
          TYPE = "STRING"
          Value = "Other Grid System"
        END_OBJECT = ZoneIdentifier
      END_GROUP = ZoneIdentifierClass
    END_GROUP = HorizontalSpatialDomainContainer
    GROUP = GranuleLocality
      OBJECT = LocalityValue
```

```
        Mandatory = "FALSE"
        Data_Location = "MCF"
        NUM_VAL = 1
        TYPE = "STRING"
        Value = "Global"
      END_OBJECT = LocalityValue
    END_GROUP = GranuleLocality
  END_GROUP = SpatialDomainContainer
  GROUP = RangeDateTime
    OBJECT = RangeEndingDate
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "DATE"
    END_OBJECT = RangeEndingDate
    OBJECT = RangeEndingTime
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "TIME"
    END_OBJECT = RangeEndingTime
    OBJECT = RangeBeginningDate
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "DATE"
    END_OBJECT = RangeBeginningDate
    OBJECT = RangeBeginningTime
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "TIME"
    END_OBJECT = RangeBeginningTime
  END_GROUP = RangeDateTime
  GROUP = PGEVersionClass
    OBJECT = PGEVersion
      Mandatory = "TRUE"
      Data_Location = "PGE"
      NUM_VAL = 1
      TYPE = "STRING"
    END_OBJECT = PGEVersion
  END_GROUP = PGEVersionClass
  GROUP = AssociatedPlatformInstrumentSensor
    OBJECT = AssociatedPlatformInstrumentSensorContainer
      Data_Location = "NONE"
      Mandatory = "TRUE"
      CLASS = "M"
      OBJECT = AssociatedSensorShortName
        Mandatory = "TRUE"
        CLASS = "M"
        Data_Location = "PGE"
        NUM_VAL = 1
        TYPE = "STRING"
      END_OBJECT = AssociatedSensorShortName
      OBJECT = AssociatedPlatformShortName
        Mandatory = "TRUE"
        CLASS = "M"
```

```
        Data_Location = "PGE"
        NUM_VAL = 1
        TYPE = "STRING"
      END_OBJECT = AssociatedPlatformShortName
      OBJECT = OperationMode
        Mandatory = "FALSE"
        CLASS = "M"
        Data_Location = "PGE"
        NUM_VAL = 1
        TYPE = "STRING"
      END_OBJECT = OperationMode
      OBJECT = AssociatedInstrumentShortName
        Mandatory = "TRUE"
        CLASS = "M"
        Data_Location = "PGE"
        NUM_VAL = 1
        TYPE = "STRING"
      END_OBJECT = AssociatedInstrumentShortName
    END_OBJECT = AssociatedPlatformInstrumentSensorContainer
  END_GROUP = AssociatedPlatformInstrumentSensor
  GROUP = AdditionalAttributes
    OBJECT = AdditionalAttributesContainer
      Data_Location = "NONE"
      Mandatory = "FALSE"
      CLASS = "M"
      OBJECT = AdditionalAttributeName
        Mandatory = "FALSE"
        CLASS = "M"
        Data_Location = "PGE"
        NUM_VAL = 1
        TYPE = "STRING"
      END_OBJECT = AdditionalAttributeName
      GROUP = InformationContent
        CLASS = "M"
        OBJECT = ParameterValue
          Mandatory = "FALSE"
          Data_Location = "PGE"
          NUM_VAL = 1
          TYPE = "STRING"
        END_OBJECT = ParameterValue
      END_GROUP = InformationContent
    END_OBJECT = AdditionalAttributesContainer
  END_GROUP = AdditionalAttributes
END_GROUP = INVENTORYMETADATA
END
```