CERES Software Bulletin 97-08

Revision 1 - January 13, 1998

How to Write Metadata for a Granule Using the Wrapper

Alice Fan (t.f.fan@larc.nasa.gov)
Maria Vallas Mitchum (m.v.mitchum@larc.nasa.gov)

1.0 Purpose

This bulletin describes how to use the CERESlib metadata wrapper subroutine, WriteMeta, to generate the required metadata for a Hierarchical Data Format (HDF) or Direct Access output product header record and to create the metadata load file (.met) for all output products. The term HDF used in this bulletin represents both HDF and HDF-EOS files. This discussion will include the requirements of the Process Control File (PCF) and Metadata Configuration File (MCF) for metadata creation as defined in the ECS Toolkit Version 5.2.1 and the CERES metadata requirements at the time of this writing. Please refer to the Computer Bulletin 97-06, CERES Metadata Approach, for a detailed description of the CERES metadata policy, approach, and metadata which is produced. This is a living document and will be updated if either the Toolkit requirements or the CERES metadata requirements change.

2.0 Metadata Wrapper

The metadata wrapper is a Fortran 90 module, named meta_util, that has three public subroutines: WriteMeta, ReadMeta, and ReadHeader. This bulletin describes how to use the subroutine WriteMeta, which is the access routine used to produce the metadata required by the Earth Observation System Data Information System Core System (ECS) and CERES. It produces three types of metadata: 1) Object Description Language (ODL) format metadata which reside on HDF and .met files, 2) ASCII format metadata which reside on Direct Access files, and 3) Vdata format metadata which reside on HDF files. Subroutine ReadMeta reads metadata in ODL format and subroutine ReadHeader reads metadata in ASCII format. Subroutines ReadMeta and ReadHeader will be described in a later computer bulletin.

Included in the meta_util module are eight public data types: Bounding_type, Bounding_null, GRing_type, GRing_null, Meta_type, Meta_null, Pointer_type, and Pointer_null. These data types will be described in more detail in later sections of this document.

3.0 Minimum Calling Sequence to the WriteMeta Subroutine

The WriteMeta subroutine must be called for all stored output products. At a minimum, the metadata will be written to the metadata load file (.met) for each output product. A header record (in ODL and Vdata format) is also written to all HDF files. A header record for non-HDF files is written using an optional parameter of the WriteMeta subroutine. This option should be used for Direct Access files. All other non-HDF file types, such as sequential binary files and ASCII files, should not use the header record option.

The minimum requirement to call the WriteMeta subroutine for a nonHDF file is:

```
CALL WriteMeta(MCF_LID, Output_LID,                                  & !required
               BeginDT, EndDT,                                       & !required*
               QAFlag, QAFlagExplain,                                & !required
               BoundingRect,                                         & !required**
               LoadFile_LID,                                         & !required
               WriteHead)                                              !optional
or
CALL WriteMeta(MCF_LID, Output_LID,                                  & !required
               BeginWholday, BeginFracday, EndWholday, EndFracday,   & !required*
               QAFlag, QAFlagExplain,                                & !required
               BoundingRect,                                         & !required**
               LoadFile_LID,                                         & !required
               WriteHead)                                              !optional
```

The minimum requirement to call the WriteMeta subroutine for an HDF file is:

```
CALL WriteMeta(MCF_LID, Output_LID,                                  & !required
               BeginDT, EndDT,                                       & !required*
               QAFlag, QAFlagExplain,                                & !required
               BoundingRect,                                         & !required**
               NumberOfRec)                                            !required
or
CALL WriteMeta(MCF_LID, Output_LID,                                  & !required
               BeginWholday, BeginFracday, EndWholday, EndFracday,   & !required*
               QAFlag, QAFlagExplain,                                & !required
               BoundingRect,                                         & !required**
               NumberOfRec)                                            !required
```

Note: required* - 'BeginDT, EndDT' or 'BeginWholday, BeginFracday, EndWholday, EndFracday' time notation is required.
required** - 'BoundingRect' or 'GRing' geometric description is required

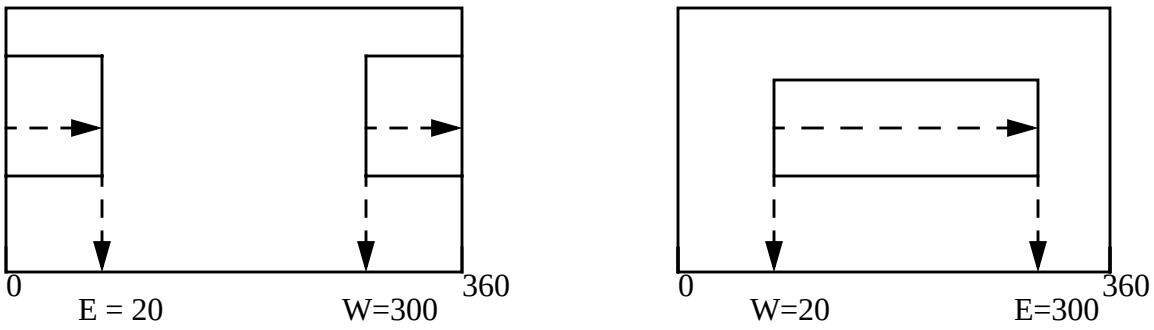
The input parameters are described below:

1. **MCF_LID**: the Logical ID (LID) for the MCF file, an integer.
2. **Output_LID**: the Logical ID for the output file, an integer.
3. **BeginDT** and **EndDT**: the beginning and ending data date and time in 27 character Consultative Committee Space Data Systems (CCSDS) format, which is YYYY-MM-DDTHH:MM: SS.xxxxxxZ. BeginDT and EndDT can be replaced by **BeginWholday**, **BeginFracday**, **End-Wholday**, and **EndFracday** in Julian format, where all values can be 4 byte (or all values can be 8 byte) real numbers. Although the user can input the datetime values in the Julian format, these values are converted to the CCSDS format in the wrapper before the metadata is produced.
4. **QAFlag**: Automatic Quality Flag, 64 character string. The valids are: Passed, Fail ed, and Suspect, which are defined as QA_PASS, QA_FAIL, and QA_SUSPECT in the 'ceres_status' mod-

ule of CERESlib.

5. **QAFlagExplain**: Automatic Quality Flag Explanation, 255 character string.
6. **BoundingRect**: the four corner CERES bounding coordinates for the produced data file.

CERES Data Envelope



The data type which describes the Bounding Rectangle, Bounding_type, is defined in the meta_util module as:

```
TYPE Bounding_type
        REAL        :: North
        REAL        :: South
        REAL        :: East
        REAL        :: West
END TYPE Bounding_type
```

The Bounding_NULL data type is defined as follows in the meta_util module:

```
TYPE (Bounding_type), PARAMETER:: &
    Bounding_NULL = Bounding_type(REAL4_DFLT, REAL4_DFLT, REAL4_DFLT, REAL4_DFLT)
```

Use the Bounding_NULL data type to initialize the Bounding_type data structure.

The CERES coordinate ranges are 0 degrees for North to 180 degrees for South, and 0 to 360 degrees for East and West. The North and South coordinates are converted to 90 to -90 degrees and the East and West coordinates are converted to -180 to 180 degrees in the wrapper in order to comply with the ECS valid ranges. The wrapper will expect the range of data longitude values to fall between the West to East longitude coordinates. See example below.

GRing: The parameter BoundingRect can be replaced by the parameter GRing, which is of

data type GRing_type. Its data structure, defined in the meta_util module, is listed below. The GRing polygon is currently defined as a five point GRing and may be changed later depending on the CERES GRing definition:

```
TYPE GRing_type
        REAL                    ::Latitude(5)
        REAL                    ::Longitude(5)
        INTEGER                 ::SeqNo(5)
        CHARACTER (LEN=1)       ::Flag
        CHARACTER (LEN=3)       ::dummy     !needed to pad record size to 4 bytes
END TYPE Gring_type
```

where:

the Latitude(5), Longitude(5) - represent a rectangular, 4 corner, polygon, using the CERES (colatitude, longitude) coordinate values (the 5th point finishes the connectivity of the ring)

SeqNo - sequence number of each point

Flag - Y (yes - Lat/Long starting point defined at inner (exclusion) GRing) or
- N (no - Lat/Long starting point defined at outer GRing)

Note: CERES typically will NOT have exclusion GRings

The GRing_NULL data type is defined as follows in the meta_util module:

```
TYPE (GRing_type), PARAMETER:: &
    GRing_NULL = GRing_type(REAL4_DFLT, REAL4_DFLT, INT4_DFLT,"N")
```

Use the GRing_NULL data type to initialize the GRing_type data structure.

7. **LoadFile_LID**: the .met file LID, an integer, required only for non-HDF output files.
8. **NumberOfRec**: Number of records, an integer, required for an HDF file as part of the automatically generated Vdata structure. NumberOfRec is an optional parameter for a non-HDF file. If provided, it is written to the archive group of the ODL formatted metadata (reside in HDF and .met files) and to the ASCII header of a Direct Access file.
9. **WriteHead:** an integer, optional parameter, where the valids are 1 or 0. The WriteHead parameter flag controls the creation of the CERES ASCII header record. The value 1 produces a header record on a non-HDF Direct Access file; the value 0 will not write the header record. The default value for this parameter is 0. It is recommended that all QC reports and sequential binary files use the default value of 0, no header record. Note: this parameter is not needed and it is ignored for an HDF file, where the header information is automatically written.

4.0 Examples of Minimum Calling Sequence

4.1 Example 1: for a Direct Access output file

```
USE ceres_status, ONLY      : QA_PASS
USE meta_util, ONLY         :  WriteMeta,  Bounding_type, Bounding_NULL
TYPE(Bounding_type)         :: Bounding
```

```
CHARACTER (LEN=27)      :: BeginDT, EndDT              ! 27 character CCSDS ASCII format
CHARACTER (LEN=64)      :: QAFlag
CHARACTER (LEN=255)     :: QAFlagExplain
INTEGER                 :: WriteHeader

Bounding = Bounding_NULL                               !initialization
BeginDT = "1996-01-15T15:00:00.000000Z"    EndDT    = "1996-01-15T15:59:59.423645Z"
QAFlag   = QA_PASS                         QAFlagExplain = "no error detected"
Bounding%East = 220.0                      Bounding%North = 0.0
Bounding%West = 60.0                       Bounding%South= 180.0
WriteHeader = 1

CALL WriteMeta(11101, 254,  BeginDT, EndDT, QAFlag, QAFlagExplain,    &
               BoundingRect= Bounding, LoadFile_LID = 2541, WriteHead = WriteHeader)
```

The three LIDs (11101, 254, and 2541) must be listed in the PCF. The first two LIDs indicate a direct access output file with LID = 254 and its corresponding MCF with LID = 11101. The LID = 2541 is the load file LID which is described in Section 6. The metadata created by the above example for the Bounding Rectangle is shown in Appendix A in ASCII format and Appendix B in ODL format.

4.2 Example 2: for an HDF output file

```
USE ceres_status, ONLY          : QA_PASS
USE meta_util, ONLY             :  WriteMeta, GRing_type, GRing_NULL
REAL(real4)                     :: Begin_Wday, Begin_Fday, End_Wday, End_Fday      !could be real8
TYPE(GRing_type)                :: GRing_SSF

GRing_SSF    =  GRing_NULL                                                         !initialization
Begin_Wday   =  2450098.0                  Begin_Fday   = 0.125
End_Wday     =  2450098.0                  End_Fday     = 0.1667
QAFlag         = QA_PASS                   QAFlagExplain = "no error detected"
GRing_SSF%Latitude(1)   = 50.0             GRing_SSF%Longitude(1) = 210.0
GRing_SSF%Latitude(2)   = 110.0            GRing_SSF%Longitude(2) = 210.0
GRing_SSF%Latitude(3)   = 110.0            GRing_SSF%Longitude(3) = 160.0
GRing_SSF%Latitude(4)   = 50.0             GRing_SSF%Longitude(4) = 160.0
GRing_SSF%Latitude(5)   = 50.0             GRing_SSF%Longitude(5) = 210.0
GRing_SSF%SeqNo(1)     = 1                 GRing_SSF%SeqNo(2)      = 2
GRing_SSF%SeqNo(3)     = 3                 GRing_SSF%SeqNo(4)      = 4
GRing_SSF%SeqNo(5)     = 5                 GRing_SSF%flag="N"

CALL WriteMeta(11102, 253,  Begin_Wday, Begin_Fday, End_Wday, End_Fday,
               QAFlag, QAFlagExplain,   GRing = GRing_SSF, NumberOfRec= 9876)
```

The two LIDs (11102, 253) must be listed in the PCF. The two LIDs indicate an HDF output file with LID = 253 and its corresponding MCF with LID = 11102. Note: for HDF files, the load file LID is not a required input to the WriteMeta subroutine. The .met file for an HDF file is an automatic output product from the Toolkit metadata functions. The metadata created by the above example for the GRing is shown in Appendix C in ASCII format and a partial listing in Appendix D in ODL format.

5.0 Close Data File Requirement

Close each file before calling the WriteMeta subroutine. If an HDF file is left open, the last data record will be lost. In the case of a Direct Access file, it must be closed because the wrapper reopens the data file with a record length of 80 bytes for the metadata header record.

6.0 PCF Requirements

There are certain required entries in the Product Input Files, Product Output Files, and User Defined Runtime Parameters sections of the PCF.

1. PRODUCT INPUT FILES section
   1.a > The two entries (with LID 10252 and 10254) in the Product Input Files section are required by the Toolkit, as shown in the example below. They are scratch files used by the metadata Tools and should be placed under a scratch file directory.

   1.b> Each non-temporary file (data product or QC report) requires an MCF identification entry in the Product Input File section of the PCF. The last two entries in the following example show sample entries of two MCF files. The LIDs 11101 and 11102 are user-defined LID values, within the Toolkit constraints, where LID values {10000..11000} are reserved for the TK, and passed into the WriteMeta subroutine through the first required parameter MCF_LID.

   ```
   ? PRODUCT INPUT FILES
   10252|AttrGet.temp|/disk2/thunder/fan/Meta1/scr||||1          /*LID should not be changed*/
   10254|WriteMCF.temp|/disk2/thunder/fan/Meta1/scr||||1        /*LID should not be changed*/
   11101|cgflatab.mcf|/disk2/thunder/fan/Meta1/rcf||||1          /*your choice of LID for MCF*/
   11102|cgssf_ab.mcf|/disk2/thunder/fan/Meta1/rcf||||1         /*your choice of LID for MCF*/
   ```

2. PRODUCT OUTPUT FILES section
   Each output file requires an entry in the Product Output Files section of the PCF. The following are sample entries for two output files. The LID (253, or 254) is passed into the WriteMeta subroutine through the second required parameter Output_LID.

   ```
   ? PRODUCT OUTPUT FILES
   253|CER_SSF_TRMM-PFM-VIRS_PreFlight_00001.1996011515|||||1   /*output HDF file */
   254|CER_SSFB_TRMM-PFM-VIRS_PreFlight_00001.1996011515||||1 /*output Direct Access file*/
   ```

3. USER DEFINED RUNTIME PARAMETERS section
   There are several required entries in the User Defined Runtime Parameters section due to the Toolkit requirements and the CERES metadata approach. The entries listed below illustrate them:

   ```
   ?USER DEFINED RUNTIME PARAMETERS
   10220|Toolkit version string|SCF B.0 TK5.2.1                                    /*do not change*/
   2541|CER_SSFB_TRMM-PFM-VIRS_PreFlight_00001.1996011515.met|254:1     /*for nonHDF*/
   #
   141|PGEName|"4.6.1P1"                                          /* from Production request */
   142|SamplingStrategyOutput|"TRMM-PFM-VIRS"                    /* from Production Request */
   ```

```
143|CERHRofDay|”15”                                   /* from Production Request */
144|ProductionStrategyOutput|”AtLaunch”               /* from Production Request */
145|CERDataDateYear|”1996”                            /* from Production Request */
146|CERDataDateMonth|”01”                            /* from Production Request */
147|CERDataDateDay|”15”                               /* from Production Request */
148|CERHRofMonth|”352”                                /* from Production Request */
149|ConfigurationCode|”00001”                         /* from LaTIS database */
150|SWsccr|”00013”                                    /* from LaTIS database */.
151|DATAsccr|”00015”                                  /* from LaTIS database */
152|<ProductSpecificAttributeName>|<”User’s value”>   /* from Production Request*/
```

3.a> Toolkit requirements: The first entry, LID 10220, is a new required entry since TK5.2; it must exist on all PCF files and may not be changed.

3.b> Non-HDF output file requirement: The second entry, LID 2541, is required for each non-HDF output file, where the LID value is unique for each non-HDF Output Product. This entry is NOT needed for an HDF output file. The first field, 2541, is the user’s choice, within the TK constraints, and is passed into the wrapper through the parameter LoadFile_LID. It is used to generate the .met file for a non-HDF file. The second field is ignored by the Toolkit and can therefore contain any user’s comment. The third field, 254:1, is the necessary and essential element that connects the .met file with its associated output file specified in the PRODUCT OUTPUT FILES section. The value “254” is the LID and “:1” is the file_index or version number of the output non-HDF file. Restated, 254:1 is the first and the last field of a Product Output non-HDF file entry.

3.c> CERES metadata requirements: The entries (from LID 141 to 151) are required by the CERES metadata approach and not by the Toolkit. These LIDs are the user’s choices. The second field must be the same as shown above. If any of the above entries is not applicable (i.e. CERDataDateDay, CERHRofMonth) to a particular product, then the third field could be left blank as ““ or the entire entry may be eliminated.

3.d> Product Specific metatdata: A Production Generation Executable (PGE) may require additional Product Specific Attributes, which should be entered in locations 152 and beyond as Runtime Parameters, which may then be read into the wrapper for processing.

7.0 MCF Requirements

An MCF file template resides in the /CERES/lib/rcf directory on the thunder computer. The user should copy this template to the rcf directory and change the value of the ShortName object to the corresponding output product ShortName which is listed in the CERES internal document “CERES ESDT ShortName”. An example MCF structure of the ShortName object is shown below. The only line to be changed is the line starting with ‘Value = “XXXXXXXX”’, where XXXXXXXX is the 8 character CERES ShortName.

```
OBJECT = ShortName
    Data_Location = “MCF”
    NUM_VAL = 1
    TYPE = “STRING”
```

```
    Mandatory = "FALSE"
    Value = "XXXXXXXX"
  END_OBJECT = ShortName
```

An MCF has two major groups: Inventory and Archive. A user will not change or add objects to the Inventory group of the MCF. If a new entry is desired, the user must contact the authors of this bulletin for specific instructions. A user is allowed to add as many Product Specific Attributes (PSA) as desired to the Archive group of the MCF template. These attributes are written to the data file header record but are not entered into the Planning and Data Production System (PDPS) database.

Note: If there is a Toolkit upgrade that effects the MCF or a change in the CERES metadata requirements, a new version of the template may be required. If this occurs, it will be announced through proper channels, and users will be required to use the updated MCF template.

8.0 Full Set of Parameters for the WriteMeta Subroutine:

There are three additional optional parameters: SpecificAttr, InputPointer, and Output_index available to a user in the WriteMeta subroutine. The calling sequence is as follows:

```
WriteMeta(MCF_LID, Output_LID,                  & !required
      BeginDT, EndDT,                           & !required*
      QAFlag, QAFlagExplain,                    & !required
      BoundingRect, Gring,                      & !required**
      Loadfile_LID, WriteHead,                  & !required (for non-HDF file)
      NumberOfRec,                              & !required (for Vdata - HDF)
      SpecificAttr,                             & !optional
      InputPointer,                             & !optional
      Output_index)                               !optional
```

Note: required* - 'BeginDT, EndDT' or 'BeginWholday, BeginFracday, EndWholday, EndFracday' time notation is required.
required** - 'BoundingRect' or 'GRing' geometric description is required

The usage of the three optional parameters is described in the following sections.

8.1 Parameter SpecificAttr

There are two groups of Product Specific Attributes (PSA): 1) CERES-Required PSAs as discussed in Section 8.1.2, and 2) Working Group selected PSAs. The parameter SpecificAttr is provided for the user to input any number of Working Group selected PSAs to be stored by the wrapper. It is a variable length array of data type meta_type which has five fields:

```
TYPE meta_type
  CHARACTER (LEN=PGSd_MET_NAME_L)   :: Name
  INTEGER                           :: IntVal
  REAL                              :: RealVal
```

```
    REAL(real8)                                   :: doubleVal
    CHARACTER (LEN= 256)                          :: StringVal
END TYPE meta_type
```

The meta_NULL data type is defined as follows in the meta_util module:

```
TYPE (meta_type), PARAMETER::  &
        meta_NULL = meta_type(“”,INT4_DFLT, REAL4_DFLT, REAL8_DFLT, “”)
```

Use the meta_NULL data type to initialize the meta_type data structure.

The data type “meta_type” has been structured to allow users the flexibility to store metadata of any data format (integer, 4 byte real, 8 byte real, and character string). In order to use this parameter properly, the user is required to:

1. Declare an array of any size with the data type as meta_type and initialize the array with the meta_NULL data type.
2. Enter the attribute name field with a recommended maximum length of 30 characters.
3. Enter one of the 4 data fields (IntVal, RealVal, DoubleVal, or StringVal) data value.

The following example shows the steps of initializing and assigning values to the array.

```
USE ceres_status, ONLY : QA_PASS
 USE meta_util, ONLY   : meta_type, meta_NULL
 TYPE (meta_type)      :: SSFFile_Attr(3)                    !could be any size
 INTEGER               :: WriteHeader
 DO i = 1, 3
    SSFFile_Attr(i)  = meta_NULL                             !initialization
 END DO
 SSFFile_Attr(1)%Name        = “PercentCrosstrackFOV”
 SSFFile_Attr(1)%RealVal     = 69.8
 SSFFile_Attr(2)%Name        = “PercentRapsFOV”
 SSFFile_Attr(2)%RealVal     = 30.2
 SSFFile_Attr(3)%Name        = “PercentOtherFOV”
 SSFFile_Attr(3)%RealVal     =  0.0
 Writeheader                 = 1
```

When calling the WriteMeta subroutine with the PSA parameter SSFFile_Attr, the PercentCrosstrackFOV, PercentRapsFOV, and PercentOtherFOV will be written in the metadata of the .met file and in the header record.

```
CALL WriteMeta(11102, 253,                                                  &
      BeginDT, EndDT, QAFlag= QA_PASS, QAFlagExplain=”no error detected”,   &
       BoundingRect= Bounding,  WriteHead = WriteHeader, NumberofRec = 9876,  &
      SpecificAttr=SSFFile_Attr)
```

8.1.1 Where do PSAs reside in the metadata?

Any number of PSAs can be passed into the wrapper through the optional parameter, “SpecificAttr”. These parameters may or may not have been predefined within the MCF template. The

wrapper will not have the knowledge of the location destination of these parameters on the MCF. There are three options or locations where the parameters could reside, 1) if the PSA has been registered through the metadata ECS registration process, then the PSA will exist as a unique field in the MCF template in the Inventory Group, 2) if the user chooses to predefine a PSA in the Archive Group of the MCF template, the destination of the PSA will be on the header record of the output product and the .met file for non-HDF files, and 3) otherwise the PSA will be placed in the AdditionalAttributes location of the Inventory Group. The wrapper will proceed through the following logical steps for each attribute:

1. The wrapper first tries to locate the PSA in the Inventory Group. If an attribute exists in the Inventory Group, then it will be set.
2. If the attempt to set the PSA in the Inventory Group fails, it tries to set the PSA under the Archive Group. If a PSA exists in the Archival Group, it will be set.
3. If both attempts fail and the PSA does not exist in either the Inventory or the Archival Group in the MCF, then the parameter will be set in the AdditionalAttributes location of the Inventory Group.

8.1.2 CERES Product Specific Attributes

There are seven CERES Product Specific attributes: PGEName, SamplingStrategy, ProductionStrategy, CERDataDateYear, CERDataDateMonth, CERDataDateDay, and CERHRofMonth. At the time of this writing, these parameters are not registered parameters in the ECS data model and do not exist on the MCF template. Therefore, they must be stored in the AdditionalAttribute location of the Inventory Group. The wrapper will read these parameters from the PCF, perform any special calculations and write them out. See the Appendix B for examples of AdditionalAttributes.

8.2 Parameter InputPointer

The attribute “InputPointer” in the MCF template is designed to store the input files used by a PGE during execution. The Object declaration on the MCF template is listed below:

```
GROUP = InputGranule
   OBJECT = InputPointer
      Data_Location = “PGE”
      NUM_VAL = 800
      TYPE = “STRING”
      Mandatory = “TRUE”
   END_OBJECT = InputPointer
END_GROUP = InputGranule
```

The NUM_VAL = 800 is the maximum number of possible input files at this time. This parameter is used to reserve space in memory and must be modified to a number which is proper for the output granule. Below is a sample of the InputPointer attribute in ODL format on a .met file:

```
OBJECT                  = INPUTPOINTER
  NUM_VAL               = 2
   VALUE  = (“/disk2/thunder/fan/Meta1/CER_SSFI_TRMM-PFM-VIRS_AtLaunch_00001. 1996011515”,
```

```
                "/disk2/thunder/fan/Meta1/CER_LWSM_TRMM-PFM_AtLaunch_00014.1996Winter")
END_OBJECT          = INPUTPOINTER
```

Below is a sample of the InputPointer attribute in ASCII format in the header record of a Direct Access file:

```
InputPointer.1 = /disk2/thunder/fan/Meta1/CER_SSFI_TRMM-PFM-VIRS_AtLaunch_00001.1996011515
InputPointer.2  = /disk2/thunder/fan/Meta1/CER_LWSM_TRMM-PFM_AtLaunch_00014.1996Winter
```

In order to address the usage of the InputPointer parameter of the WriteMeta subroutine, it is important to know the possible sources where this attribute may be obtained.

8.2.1 Sources for the InputPointer Parameter

1) **Subroutine OpenFile**: (for nonHDF files) If a PGE uses the OpenFile (or C_OpenFile) CERESlib subroutine, available for opening nonHDF input files, the OpenFile subroutine records the opened input file in the I/O module of CERESlib. The wrapper takes this information and fills in the value for the InputPointer parameter, thus no direct interface with the wrapper is required by the user in this scenario.

2) **Subroutine AddInputPointer**: (for HDF files and nonHDF files) An HDF file cannot be opened by the OpenFile or C_OpenFile subroutines. The user can access the subroutine: AddInputFile(LID, version), from CERESlib, in order to log the Logical ID and version number for each of the opened HDF input files. The wrapper will then access this log at the time the metadata is written.

If the user does not wish to use the OpenFile subroutine to open nonHDF files, then the AddInputFile subroutine may also be used to record nonHDF files. The calling sequence is: Call AddInputFile(LID, version), where: LID is the Logical ID, and version is the file_index.

3) **Parameter InputPointer**: If the OpenFile and AddInputFile subroutines are not accessed, the user can enter the input file listing in the optional parameter "InputPointer" of the WriteMeta subroutine. This parameter contains an array of LIDs and corresponding file index numbers of the input files. The wrapper will then access the PCF with the Logical ID and file index parameters to retrieve the file path and filename. There are two data types: Pointer_type and Pointer_NULL, defined in the meta_util module. They are shown as below:

```
TYPE Pointer_Type
   INTEGER              :: LogicID
   INTEGER              :: File_index
END TYPE Pointer_Type
```

The Pointer_NULL data type is defined as follows in the meta_util module:

```
TYPE (Pointer_type), PARAMETER ::  Pointer_NULL = Pointer_type(INT4_DFLT, 1)
```

Use the Pointer_NULL data type to initialize the Pointer_type data structure.

The user is required to initialize the Pointer_type array with the Pointer_NULL default values before setting any values. The wrapper depends on checking for the default value in the LogicID field to stop the population of the InputPointer field. After the initialization, the only field that is required to fill is the LogicID field. The default value for the file_index is set to 1 at initialization, unless the user replaces this value. An sample calling sequence to the wrapper is shown below:

```
USE ceres_status, ONLY : QA_PASS
USE meta_util, ONLY :: Pointer_type, Pointer_NULL
  TYPE(Pointer_type)   :: SSF_input(10)                          !could be any size
  INTEGER              :: WriteHeader
  DO i = 1, 10
     SSF_input(i) = Pointer_NULL                                  !initialization
  END DO
  WriteHeader = 1
  SSF_input(1)%LogicID = 1112      !LID of an input data file
  SSF_input(1)%File_index = 2      !file index (=version) of an input data file
  SSF_input(2)%LogicID = 1110      !LID of another input data file

  CALL WriteMeta(11102, 253, &
     BeginDT, EndDT, QAFlag= QA_PASS, QAFlagExplain="no error detected",      &
     WriteHead = WriteHeader, NumberofRec= 9876,   BoundingRect= Bounding,  &
     SpecificAttr= SSFFile_Attr, InputPointer= SSF_input)
```

4) **Parameter SpecificAttr**: The optional parameter "SpecificAttr" may be used as an alternate method to provide the input file listing to the wrapper. Since there are no rules for the name field of the SpecificAttr array, it is possible for the user to enter 'InputPointer" in the name field. If this occurs, the wrapper will collect these values and build the InputPointer parameter. Although this alternate method is available, users are discouraged from using this parameter to provide Input file information.

8.3 Parameter Output_Index

Output_Index, an integer, optional parameter, is the index (or version) number for an output file. If not provided, the default value is 1. The parameter Output_LID, discussed in Section 3.0, and the parameter Output_Index comprise the required 'file identification entities' for the Toolkit output file functions.

9.0 Future Implementation: InputPointer for Multiple Output Files in One PGE

As stated in Section 8.1, the metadata wrapper concatenates the information from four sources. Then the wrapper calls the Toolkit function PGS_MET_SetAttr_S once to set the values for the InputPointer attribute. There are cases where not all the opened files are used to generate a particular output file in a PGE which produces more than one output product.

Although the optional parameters InputPointer and SpecificAttr can be used to provide the different input file information to the WriteMeta subroutine for different output products, the input files recorded in the I/O module have no way of discriminating the output source. There are cases where not all the input files recorded in the I/O module are used to generate a particular output

product. For now we will live with the fact that all the input files recorded are used for this PGE but not necessarily for this particular output product.

In order to remedy this kind of situation, a way to allow the user to remove input files from the input file list collected by the I/O module could be implemented. The suggested implementation is adding a field (i.e. inout) to the data type Pointer_type. Since the parameter InputPointer is of data type Pointer_type, it can be used to add or remove input files from the list collected in the I/O module. If this field is set to “I”, add this file to the value of the InputPointer attribute. If this field is set to “O”, omit, or remove, the file from the Input file list gathered by the I/O module and the wrapper will not include this file in the InputPointer parameter. The original input file list gathered by the I/O module should stay unchanged, so it can be used as the baseline information for the other output files. This option will not be implemented until there is a need.

Appendix A: ASCII header with Bounding Rectangle Example

```
BEGIN_HEADER
ShortName                          = CGFLATAB
VersionID                          =      1
CERPGEName                         = 4.6P1
SamplingStrategy                   = TRMM-PFM-VIRS
ProductionStrategy                 =  AtLaunch
CERDataDateYear                    = 1996
CERDataDateMonth                   = 01
CERDataDateDay                     = 15
CERHrOfMonth                       = 352
RangeBeginningDate                 = 1996-01-15
RangeBeginningTime                 = 15:00:00.000000
RangeEndingDate                    = 1996-01-15
RangeEndingTime                    = 16:00:00.000000
AssociatedPlatformShortName.1      = TRMM
AssociatedInstrumentShortName.1    = PFM
LocalGranuleID                     =  CER_SSFB_TRMM-PFM-VIRS_AtLaunch_00001.1996011515
PGEVersion                         = 00001
CERProductionDateTime              = 1998-01-08T12:32:21.000000Z
LocalVersionID                     =  Clib-19971230  TK5.2.1 HDF-4.1r1  HDFEOS- 2.0  SW00013  ANC00015
ProductGenerationLOC               = NASA Langley Research Center, HOST - thunder1-f  OS - IRIX64
NumberOfRecords                    =       9876
WestBoundingCoordinate             =   60.000000
NorthBoundingCoordinate            =   90.000000
EastBoundingCoordinate             = -140.000000
SouthBoundingCoordinate            =  -90.000000
CERWestBoundingCoordinate          =   60.000000
CERNorthBoundingCoordinate         =    0.000000
CEREastBoundingCoordinate          =  220.000000
CERSouthBoundingCoordinate         =  180.000000
AutomaticQualityFlag.1             = Passed
AutomaticQualityFlagExplanation.1  = no error detected
ImagerShortName                    = VIRS
CERHrOfDay                         =         15
InputPointer.1   =  /disk2/thunder/fan/Meta1/CER_SSFI_TRMM-PFM-VIRS_AtLaunch_00001.1996011515
InputPointer.2   =  /disk2/thunder/fan/Meta1/CER_LWSM_TRMM-PFM_ArLaunch_00014.1996Winter
END_HEADER
```

Appendix B: ODL Metadata with Bounding Rectangle Example

```
GROUP       = INVENTORYMETADATA
 GROUPTYPE          = MASTERGROUP

 GROUP                 = ECSDATAGRANULE
                    OBJECT              = LOCALGRANULEID
                    NUM_VAL             = 1
                    VALUE               = “CER_SSFB_TRMM-PFM-VIRS_AtLaunch_00001.1996011515”
                    END_OBJECT          = LOCALGRANULEID

                    OBJECT              = LOCALVERSIONID
                    NUM_VAL             = 1
                    VALUE               = “Clib-19971230 TK5.2.1 HDF-4.1r1 HDFEOS- 2.0 SW00013 ANC00015”
                    END_OBJECT          = LOCALVERSIONID

                    OBJECT              = PRODUCTIONDATETIME
                    NUM_VAL             = 1
                    VALUE               = “1998-01-08T17:32:21.000Z”
                    END_OBJECT          = PRODUCTIONDATETIME
 END_GROUP             = ECSDATAGRANULE

 GROUP              = MEASUREDPARAMETER
  OBJECT               = MEASUREDPARAMETERCONTAINER
                    CLASS               = “1”

                    OBJECT              = PARAMETERNAME
                    CLASS               = “1”
                    NUM_VAL             = 1
                    VALUE               = “GranuleParameters”
                    END_OBJECT          = PARAMETERNAME

                    GROUP               = QAFLAGS
                    CLASS               = “1”
                    OBJECT              = AUTOMATICQUALITYFLAG
                    NUM_VAL             = 1
                    CLASS               = “1”
                    VALUE               = “Passed”
                    END_OBJECT          = AUTOMATICQUALITYFLAG

                    OBJECT              = AUTOMATICQUALITYFLAGEXPLANATION
                    NUM_VAL             = 1
                    CLASS               = “1”
                    VALUE               = “no error detected”
                    END_OBJECT          = AUTOMATICQUALITYFLAGEXPLANATION
                    END_GROUP           = QAFLAGS

  END_OBJECT          = MEASUREDPARAMETERCONTAINER
 END_GROUP           = MEASUREDPARAMETER


 GROUP                = COLLECTIONDESCRIPTIONCLASS

                    OBJECT              = SHORTNAME
                    NUM_VAL             = 1
                    VALUE               = “CGFLATAB”
                    END_OBJECT          = SHORTNAME
```

```
                OBJECT          = VERSIONID
                NUM_VAL         = 1
                VALUE           = 1
                END_OBJECT      = VERSIONID
END_GROUP           = COLLECTIONDESCRIPTIONCLASS

GROUP           = INPUTGRANULE
                OBJECT          = INPUTPOINTER
                NUM_VAL         = 800
                VALUE  = ("/disk2/thunder/fan/Meta1/CER_SSFI_TRMM-PFM-VIRS_AtLaunch_00001.1996011515",
                          "/disk2/thunder/fan/Meta1/CER_LWSM_TRMM-PFM_AtLaunch_00014.1996Winter")
                END_OBJECT      = INPUTPOINTER
END_GROUP       = INPUTGRANULE

GROUP           = SPATIALDOMAINCONTAINER
 GROUP          = HORIZONTALSPATIALDOMAINCONTAINER
  GROUP         = BOUNDINGRECTANGLE

   OBJECT               =WESTBOUNDINGCOORDINATE
    NUM_VAL             = 1
    VALUE               = 60.000000
   END_OBJECT           = WESTBOUNDINGCOORDINATE

   OBJECT               = NORTHBOUNDINGCOORDINATE
    NUM_VAL             = 1
    VALUE               = 90.000000
   END_OBJECT           = NORTHBOUNDINGCOORDINATE

   OBJECT               = EASTBOUNDINGCOORDINATE
    NUM_VAL             = 1
    VALUE               = -140.000000
   END_OBJECT           = EASTBOUNDINGCOORDINATE

   OBJECT               = SOUTHBOUNDINGCOORDINATE
    NUM_VAL             = 1
    VALUE               = -90.000000
   END_OBJECT           = SOUTHBOUNDINGCOORDINATE

  END_GROUP         = BOUNDINGRECTANGLE
 END_GROUP          = HORIZONTALSPATIALDOMAINCONTAINER
END_GROUP           = SPATIALDOMAINCONTAINER


GROUP           = RANGEDATETIME
                OBJECT          = RANGEBEGINNINGTIME
                NUM_VAL         = 1
                VALUE           = "15:00:00.000000"
                END_OBJECT      = RANGEBEGINNINGTIME

                OBJECT          = RANGEENDINGTIME
                NUM_VAL         = 1
                VALUE           = "16:00:00.000000"
                END_OBJECT      = RANGEENDINGTIME

                OBJECT          = RANGEBEGINNINGDATE
                NUM_VAL         = 1
                 VALUE          = "1996-01-15"
                END_OBJECT      = RANGEBEGINNINGDATE
```

```
                OBJECT          = RANGEENDINGDATE
                NUM_VAL         = 1
                VALUE           = “1996-01-15”
                END_OBJECT      = RANGEENDINGDATE
END_GROUP           = RANGEDATETIME

GROUP           = PGEVERSIONCLASS
                OBJECT          = PGEVERSION
                NUM_VAL         = 1
                VALUE           = “00001”
                END_OBJECT      = PGEVERSION
END_GROUP       = PGEVERSIONCLASS


GROUP           = ADDITIONALATTRIBUTES

 OBJECT         = ADDITIONALATTRIBUTESCONTAINER
  CLASS         = “1”

                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = “1”
                NUM_VAL         = 1
                VALUE           = “CERPGEName”
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                GROUP           = INFORMATIONCONTENT
                CLASS           = “1”
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = “1”
                        VALUE           = “4.6P1”
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
 END_OBJECT     = ADDITIONALATTRIBUTESCONTAINER

 OBJECT         = ADDITIONALATTRIBUTESCONTAINER
  CLASS         = “2”

                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = “2”
                NUM_VAL         = 1
                VALUE           = “SamplingStrategy”
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                GROUP           = INFORMATIONCONTENT
                CLASS           = “2”
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = “2”
                        VALUE           = “TRMM-PFM-VIRS”
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
 END_OBJECT         = ADDITIONALATTRIBUTESCONTAINER

 OBJECT         = ADDITIONALATTRIBUTESCONTAINER
  CLASS         = “3”

                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = “3”
```

```
                NUM_VAL         = 1
                VALUE           = “ProductionStrategy”
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                GROUP           = INFORMATIONCONTENT
                CLASS           = “3”
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = “3”
                        VALUE           = “AtLaunch”
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
END_OBJECT          = ADDITIONALATTRIBUTESCONTAINER

OBJECT            = ADDITIONALATTRIBUTESCONTAINER
 CLASS            = “4”
                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = “4”
                NUM_VAL         = 1
                VALUE           = “CERDataDateYear”
                 END_OBJECT     = ADDITIONALATTRIBUTENAME

                 GROUP          = INFORMATIONCONTENT
                 CLASS          = “4”
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = “4”
                        VALUE           = “1996”
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
END_OBJECT        = ADDITIONALATTRIBUTESCONTAINER

OBJECT            = ADDITIONALATTRIBUTESCONTAINER
 CLASS            = “5”
                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = “5”
                NUM_VAL         = 1
                VALUE           = “CERDataDateMonth”
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                 GROUP          = INFORMATIONCONTENT
                CLASS           = “5”
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = “5”
                        VALUE           = “01”
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
END_OBJECT       = ADDITIONALATTRIBUTESCONTAINER

OBJECT            = ADDITIONALATTRIBUTESCONTAINER
 CLASS            = “6”
                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = “6”
                NUM_VAL         = 1
                VALUE           = “CERDataDateDay”
                END_OBJECT      = ADDITIONALATTRIBUTENAME
                GROUP           = INFORMATIONCONTENT
```

```
                CLASS           = "6"
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = "6"
                        VALUE           = "15"
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
END_OBJECT      = ADDITIONALATTRIBUTESCONTAINER

OBJECT          = ADDITIONALATTRIBUTESCONTAINER
                CLASS           = "7"
                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = "7"
                NUM_VAL         = 1
                VALUE           = "CERHRofMonth"
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                GROUP           = INFORMATIONCONTENT
                CLASS           = "7"
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = "7"
                        VALUE           = "352"
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
END_OBJECT      = ADDITIONALATTRIBUTESCONTAINER

OBJECT          = ADDITIONALATTRIBUTESCONTAINER
                CLASS           = "8"
                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = "8"
                NUM_VAL         = 1
                VALUE           = "CERHRofDay"
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                GROUP           = INFORMATIONCONTENT
                CLASS           = "8"
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = "8"
                        VALUE           = "15"
                END_OBJECT      = PARAMETERVALUE
                END_GROUP       = INFORMATIONCONTENT
END_OBJECT      = ADDITIONALATTRIBUTESCONTAINER

OBJECT          = ADDITIONALATTRIBUTESCONTAINER
 CLASS          = "9"
                OBJECT          = ADDITIONALATTRIBUTENAME
                CLASS           = "9"
                NUM_VAL         = 1
                VALUE           = "ImagerShortName"
                END_OBJECT      = ADDITIONALATTRIBUTENAME

                GROUP           = INFORMATIONCONTENT
                CLASS           = "9"
                OBJECT          = PARAMETERVALUE
                        NUM_VAL         = 1
                        CLASS           = "9"
                        VALUE           = "VIRS"
```

```
                    END_OBJECT      = PARAMETERVALUE
                    END_GROUP       = INFORMATIONCONTENT
  END_OBJECT        = ADDITIONALATTRIBUTESCONTAINER

 END_GROUP          = ADDITIONALATTRIBUTES


 GROUP              = ASSOCIATEDPLATFORMINSTRUMENTSENSOR

  OBJECT            = ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER
   CLASS            = "1"

                    OBJECT          = ASSOCIATEDPLATFORMSHORTNAME
                    CLASS           = "1"
                    NUM_VAL         = 1
                    VALUE           = "TRMM"
                    END_OBJECT      = ASSOCIATEDPLATFORMSHORTNAME

                    OBJECT          = ASSOCIATEDINSTRUMENTSHORTNAME
                    CLASS           = "1"
                    NUM_VAL         = 1
                    VALUE           = "PFM"
                    END_OBJECT      = ASSOCIATEDINSTRUMENTSHORTNAME

                    OBJECT          = ASSOCIATEDSENSORSHORTNAME
                    CLASS           = "1"
                    NUM_VAL         = 1
                    VALUE           = "NOT SET"
                    END_OBJECT      = ASSOCIATEDSENSORSHORTNAME

   END_OBJECT       = ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER

 END_GROUP          = ASSOCIATEDPLATFORMINSTRUMENTSENSOR

END_GROUP           = INVENTORYMETADATA


GROUP               = ARCHIVEDMETADATA
 GROUPTYPE          = MASTERGROUP

 OBJECT             = CEREASTBOUNDINGCOORDINATE
                    NUM_VAL         = 1
                    VALUE           = 220.000000
 END_OBJECT         = CEREASTBOUNDINGCOORDINATE

 OBJECT             = CERNORTHBOUNDINGCOORDINATE
                    NUM_VAL         = 1
                    VALUE           = 0.000000
 END_OBJECT         = CERNORTHBOUNDINGCOORDINATE

 OBJECT             = CERSOUTHBOUNDINGCOORDINATE
                    NUM_VAL         = 1
                    VALUE           = 180.000000
 END_OBJECT         = CERSOUTHBOUNDINGCOORDINATE

 OBJECT             = CERWESTBOUNDINGCOORDINATE
                    NUM_VAL         = 1
                    VALUE           = 60.000000
 END_OBJECT         = CERWESTBOUNDINGCOORDINATE
```

```
  OBJECT              = CERPRODUCTIONDATETIME
                      NUM_VAL         = 1
                      VALUE           = “1998-01-08T12:32:21.000000Z”
  END_OBJECT          = CERPRODUCTIONDATETIME

  OBJECT              = NUMBEROFRECORDS
                      NUM_VAL         = 1
                      VALUE           = 9876
  END_OBJECT          = NUMBEROFRECORDS

  OBJECT              = PRODUCTGENERATIONLOC
                      NUM_VAL         = 1
                      VALUE           = “NASA Langley Research Center, HOST - thunder OS - IRIX64”
  END_OBJECT          = PRODUCTGENERATIONLOC

END_GROUP             = ARCHIVEDMETADATA

END
```

Appendix C: ASCII header with GRing Example

```
BEGIN_HEADER
ShortName                          = CGFLATAB
VersionID                          = 1
CERPGEName                         = 4.6P1
SamplingStrategy                   = TRMM-PFM-VIRS
ProductionStrategy                 = AtLaunch
CERDataDateYear                    = 1996
CERDataDateMonth                   = 01
CERDataDateDay                     = 15
CERHrOfMonth                       = 352
RangeBeginningDate                 = 1996-01-15
RangeBeginningTime                 = 15:00:00.000000
RangeEndingDate                    = 1996-01-15
RangeEndingTime                    = 16:00:00.000000
AssociatedPlatformShortName.1      = TRMM
AssociatedInstrumentShortName.1    = PFM
LocalGranuleID                     =  CER_SSFB_TRMM-PFM-VIRS_AtLaunch_00001.1996011515
PGEVersion                         = 00001
CERProductionDateTime              = 1998-01-08T14:08:52.000000Z
LocalVersionID                     =  Clib-19971230  TK5.2.1  HDF-4.1r1  HDFEOS-2.0  SW00013  DATA00015
ProductGenerationLOC               = NASA Langley Research Center, HOST-thunder1-f  OS-IRIX64
NumberOfRecords                    =      9876
GRingPointLatitude.1               =    40.00  -20.00  -20.00   40.00   40.00
GRingPointLongitude.1              =  -150.00 -150.00  160.00  160.00 -150.00
GRingPointSequenceNo.1             =    1    2    3    4    5
ExclusionGRingFlag.1               =  N
CERGRingPointLatitude.1            =    50.00  110.00  110.00   50.00   50.00
CERGRingPointLongitude.1           =   210.00  210.00  160.00  160.00  210.00
CERGRingPointSequenceNo.1          =    1    2    3    4    5
 CERExclusionGRingFlag.1           = N
AutomaticQualityFlag.1             = Passed
AutomaticQualityFlagExplanation.1  = no error detected
ImagerShortName                    = VIRS
CERHrOfDay                         =        15
InputPointer.1   =  /disk2/thunder/fan/Meta1/CER_SSFI_TRMM-PFM-VIRS_AtLaunch_00001.1996011515
InputPointer.2   =  /disk2/thunder/fan/Meta1/CER_LWSM_TRMM-PFM_AtLaunch_00014.1996Winter
END_HEADER
```

Appendix D: ODL metadata with GRing Example

Note: This is a partial listing of a .met file - reference Appendix B, the only difference shown here is the SpatialDomainContainer.

```
        GROUP               = SPATIALDOMAINCONTAINER
            GROUP               = HORIZONTALSPATIALDOMAINCONTAINER
                GROUP               = GPOLYGON
                        OBJECT              = GPOLYGONCONTAINER
                                CLASS               = “1”
                                GROUP               = GRING
                                CLASS               =  “1”

                                OBJECT              = EXCLUSIONGRINGFLAG
                                        NUM_VAL         1
                                        CLASS               = “1”
                                        VALUE               = “N”
                                END_OBJECT          = EXCLUSIONGRINGFLAG
                                END_GROUP           = GRING

                                GROUP               = GRINGPOINT
                                CLASS               = “1”

                                OBJECT              = GRINGPOINTLATITUDE
                                        NUM_VAL         = 5
                                        CLASS               = “1”
                                        VALUE    = (40.000000, -20.000000, -20.000000, 40.000000, 40.000000)
                                END_OBJECT          = GRINGPOINTLATITUDE

                                OBJECT              = GRINGPOINTLONGITUDE
                                        NUM_VAL         = 5
                                        CLASS               = “1”
                                        VALUE  = (-150.000000, -150.000000, 160.000000, 160.000000, -150.000000)
                                END_OBJECT          = GRINGPOINTLONGITUDE

                                OBJECT              = GRINGPOINTSEQUENCENO
                                        NUM_VAL         = 5
                                        CLASS               = “1”
                                        VALUE               = (1, 2, 3, 4, 5)
                                END_OBJECT          = GRINGPOINTSEQUENCENO
                                END_GROUP           = GRINGPOINT
                        END_OBJECT          = GPOLYGONCONTAINER
                END_GROUP           = GPOLYGON
            END_GROUP           = HORIZONTALSPATIALDOMAINCONTAINER
        END_GROUP           = SPATIALDOMAINCONTAINER

                .
                .
GROUP               = ARCHIVEDMETADATA
  GROUPTYPE         = MASTERGROUP
        GROUP               = CERGPOLYGON
            OBJECT              = CERGPOLYGONCONTAINER
                CLASS               = “1”

                OBJECT              = CERGRINGPOINTLATITUDE
                        CLASS               = “1”
```

```
                    NUM_VAL          = 5
                    VALUE            = (50.000000, 110.000000, 110.000000, 50.000000, 50.000000)
                END_OBJECT           = CERGRINGPOINTLATITUDE

                OBJECT               = CERGRINGPOINTLONGITUDE
                    CLASS            = “1”
                    NUM_VAL          = 5

                    VALUE            = (210.000000, 210.000000, 160.000000, 160.000000, 210.000000)
                END_OBJECT           = CERGRINGPOINTLONGITUDE
            END_OBJECT           = CERGPOLYGONCONTAINER
        END_GROUP            = CERGPOLYGON
                    .
                    .
                    .
  END_GROUP              = ARCHIVEDMETADATA
END
```