CERES Software Bulletin 98-01

July 20, 1998

Placing Attributes on CERES HDF Products

Carla B. Franklin (c.b.franklin@larc.nasa.gov)

1.0 Purpose

This bulletin describes the usage of the Hierarchical Data Format (HDF) functions in placing attributes on the Vdatas and Scientific Data Sets (SDS) and naming the SDS dimensions on CERES HDF output products. If the SDS contains parameters that have the same unit, range, and fill value or the SDS contains only one parameter, pre-defined attributes may be used. If the SDS contains parameters that have different units, ranges, or fill values, user-defined attributes must be used. Both types of attributes will be discussed. Only user-defined attributes are available for Vdatas. This bulletin references calls to the Fortran HDF functions. Output from sample "hdp" commands used to list an SDS and a Vdata from a sample file is given in Appendices A and B. Two tables containing the Fortran routines used to place attributes on SDSs and Vdatas, respectively, and their equivalent C routines are given in Appendix C.

2.0 Naming an SDS Dimension

When naming the dimensions of an SDS array, the dimension names must be set before attributes are assigned, and once the attributes are set, the names should not be changed. If the dimension name is not unique, it is assumed that both dimensions refer to the same object and changes to one will be reflected in the other. Dimensions that are not named explicitly by the user will be named by the HDF library.

Function **sfdimid** gets the dimension identification number for a specific dimension and function **sfsdmname** assigns a name to that dimension. The following is an example of how to open the HDF file, create the SDS, get the dimension id, and name the dimension:

```
dfacc_rdwr = 3
hdf_id = sfstart (hdf_file, dfacc_rdwr)
sds_id = sfcreate (hdf_id, sds_name, sds_numtype,  rank, dim)
dim_id = sfdimid(sds_id, dim_index)
retn = sfsdmname(dim_id, dim_name)
```

sfstart opens the HDF file, *hdf_file*, initializes the Scientific Data (SD) interface, and returns the identifier, *hdf_id*. Argument, *dfacc_rdwr*, indicates the file is to be opened with read_write access. **sfcreate** creates a new SDS and returns its identification number, *sds_id*. Arguments *sds_name*, *sds_numtype*, *rank*, and *dim* are the name, type, rank, and dimension, respectively, of the SDS being created. **sfdimid** returns a dimension identifier, *dim_id*, for the dimension specified by *dim_index*, a zero-based integer indicating the location of the dimension in the data set. **sfsdmname** assigns a dimension name, *dim_name*, to the dimension specified by argument, *dim_id*.

To obtain information about an SDS's dimensions, **sfgdinfo** can be used as follows:

```
dim_id = sfdimid (sds_id, dim_index)
retn = sfgdinfo (dim_id, name, count, data_type, nattrs)
```

The dimension name, size, data type of the dimension scales, and the number of dimension attributes are returned by sfgdinfo in the *name*, *count*, *data_type*, *and nattrs* arguments. (For more information on dimension scales, see Section 3.9.4 in the HDF User's Guide, Version 4.1r1 REVIEW, March 1997.)

3.0 Creating Attributes on an SDS Entity

3.1 Using User-Defined Attributes To Define Attributes on an SDS

User-defined attributes can be attached to three types of objects: files, data set arrays or SDSs, and dimensions. These attributes are called, respectively, file attributes, array attributes, and dimension attributes. They take the form label = value, where label is a character string name of the attribute and value contains one or more entries of the same data type. File attributes contain information about all data objects in the file. Array attributes describe individual SDS arrays and dimension attributes provide information applicable to an individual SDS dimension. File ids, SDS ids, and dimension ids are used to, respectively, access file attributes, SDS array attributes, and dimension attributes. To assign an attribute, the following steps must be taken:

1. Obtain the appropriate identifier.
2. Create the attribute.
3. Terminate access by disposing of the identifier.

An example of placing an attribute on an HDF object follows:

```
retn = sfscatt(sd_id, attr_name, data_type, count, value)
```

The arguments to the **sfscatt** routine are described below:

1) *sd_id*: the identifier for the HDF object to be assigned the attribute. (It can be a file id, SDS id, or dimension id.)
2) *attr_name*: an ASCII string containing the name of the attribute. If this is set to the name of an existing attribute, the value portion of the existing attribute will be overwritten. Therefore, for each parameter in the SDS, the *attr_name* must have a unique value.
3) *data_type*: describes the data type for the attribute values.
4) *count*: the total number of values in the attribute. For character string attributes, *count* is the number of characters in the character string.
5) *value*: one or more values of the same data type. For character string attributes, *value* is a character string of length, *count*.

3.2 Using Predefined Attributes To Define Attributes on an SDS

Predefined attributes are attributes that have reserved labels and, in some cases, predefined data types. They take the form of reserved_label=value. They can be assigned to two types of HDF objects: SDS data sets and dimensions. They can only be used to assign data to the SDS data sets that contain only one parameter or for those SDSs whose parameters share the same attribute values. Valid reserved labels include: label, unit, format, coordinate system, range, and fill value.

3.2.1 String Attributes

All predefined string attributes for an SDS array are assigned using one call to **sfsdtstr** as follows:

retn = **sfsdtstr**(sds_id, label, unit, format, coordsys)

sds_id is the identifier assigned by the **sfcreate** routine when the SDS was created. The arguments, *label*, *unit*, *format*, and *coordsys*, contain the values to be assigned to the corresponding attributes. Labels can be the independent variable names and dimension names used as primary search keys. Unit is the unit for the SDS data array. Formats should use standard Fortran-77 notation, for example, “F7.2”. Recommended coordinate system descriptions are “cartesian”, “polar”, and “spherical”. If no value is to be assigned to a particular attribute, a value of NULL should be assigned to that attribute argument.

All predefined string attributes for a dimension are assigned using one call to **sfsdmstr** as follows:

retn = **sfsdmstr** (dim_id, label, unit, format)

where *dim_id* is the identifier returned from a call to **sfdimid** and *label*, *unit*, and *format* contain the values to be assigned to the corresponding dimension attributes. Assign a value of NULL for those attributes that are not to be assigned a value.

3.2.2 Range Attributes

To define maximum and minimum values for the range attribute, use the following call:

retn = **sfsrange** (sds_id, max, min)

where *sds_id* is the SDS identifier, *max* is the range’s maximum value, and *min* is the range’s minimum value.

3.2.3 Fill Value Attributes

A fill value can be defined to fill the spaces between non-contiguous writes to SDS arrays using the fill value attribute. The fill value should be set prior to writing any data on the SDS. All locations not replaced by valid SDS data will contain the fill value.

For numeric fill data, use the following call to sfsfill:

retn = **sfsfill** (sds_id, fill_val)

For character fill data, use the following call to sfscfill:

```
retn = sfscfill (sds_id, fill_val)
```

3.3 Example of Putting Attributes on SDSs in an SSF HDF File

The attributes to be placed on the CERES output products are units, and optionally, range and fill_value. For multi-parameter SDSs, the names of the parameters on the SDSs are not displayed, so the full parameter names must be included as attributes on the SDSs.

For SDSs that contain only one parameter, adding attributes is fairly simple. Pre-defined attributes can be used to define the range, fill_value, and units. Since there is only one parameter on the SDS, the name of the SDS can match the name of the parameter.

For multi-parameter SDSs, if the ranges, fill_values, and units are the same for all parameters on the SDS, pre-defined attributes can be used.

An example, of using the pre-defined attributes, follows:

```
retn = sfsdtstr   (sds_id, label, unit, format, coordsys)
retn = sfsrange  (sds_id, max, min)
retn = sfsfill       (sds_id, fill_val)
```

For multi-parameter SDSs whose attributes do not have the same values, **sfscatt** must be called twice for each parameter on an SDS, once to define the name attribute and once to define the units attribute. **sfscatt** is used to set character string attributes and **sfsnatt** is used for numeric attributes. The value of the name attribute is the ASCII string containing the full name of the parameter. An example of a name for this attribute is “long_name of SDS Parameter n”, where “n” is the parameter number within the SDS. This attribute is important for the multiple-parameter SDSs, because neither EOSView nor Collage identifies the parameter names along a dimension on an SDS. An example of a name for the attribute that holds the parameter’s unit is, “units for SDS Parameter n”, where “n” is the SDS parameter number. Other possible attributes are the fill value and valid range. Since the SDSs on the SSF HDF contain parameters with like fill values, only one fill value attribute is required per SDS. Therefore, the predefined attribute for fill value is used to define the fill value to be stored on the SDS prior to data being written on the SDS.

An example of placing the name, units, and fill_value attributes on one SDS, Julian_Time, is shown below. This logic should be repeated for each SDS on the HDF file. It is assumed that “nfov” is already defined as the number of footprints on the binary SSF, REAL8_DFLT is set to the fill value to be used, DFNT_CHAR8 is set to 4 to indicate an 8-bit character type, and “hdf_file” is the name of the HDF file being created.

```
CHARACTER (LEN=80) :: Julian_Time_Names(2)
CHARACTER (LEN=50) :: Julian_Time_Units(2)
CHARACTER (LEN=80) :: Julian_Time_Name_Attr_Names(2)
CHARACTER (LEN=80) :: Julian_Time_Units_Attr_Names(2)
INTEGER                          :: Julian_Time_Numparams
```

```
      INTEGER                       :: Julian_Time_Names_Char(2)
      INTEGER                       :: Julian_Time_Units_Char(2)
      INTEGER                       :: j, sdsid, count, dimid, dfacc_rdwr, hdf_id, rank
      INTEGER                       :: sds_numtype,  dim_2d(2)
      CHARACTER (LEN=80), INTENT(OUT)   :: dimension_name(2)
      DATA   Julian_Time_Units/‘day’, ‘kilometer’ /
      DATA   Julian_Time_Units_Char/3, 9 /
      DATA   Julian_Time_Names /‘Time of observation’,                                  &
                                ‘Radius of satellite from center of Earth at observation’/
      DATA   Julian_Time_Names_Char/19, 55 /
      DATA   Julian_Time_Name_Attr_Names/‘long_name of SDS Parameter1’,                 &
                                         ‘long_name of SDS Parameter2’/
      DATA   Julian_Time_Units_Attr_Names/‘units for SDS Parameter1’,                   &
                                          ‘units for SDS Parameter2’/
DATA   Julian_Time_Numparams / 2 /
      dfacc_rdwr = 3
      hdf_id = sfstart(hdf_file, dfacc_rdwr)
      rank = 2
      sds_name = ‘Julian_Time’
      sds_numtype = 6
      dim_2d(1) = 2
      dim_2d(2) = nfov
      sdsid = sfcreate (hdf_id, sds_name, sds_numtype, rank, dim_2d)
!
! Use the predefined attribute, fill_value, to define a data set array attribute for the Julian_Time SDS
!
      retn = sfsfill(sdsid,REAL8_DFLT)
!
! Name each dimension
!
      dimension_name(1) = ‘Julian_Time Parameters’
      dimension_name(2) = ‘Footprints(FOV)’
!
! For each dimension, get the dimension id number and set the dimension name.
!
       DO i = 1, rank
         count  = i - 1
         dimid = sfdimid(sdsid, count)
         retn = sfsdmname(dimid, dimension_name(i))
       ENDDO
!
! Use user-defined attributes to set the name and units attributes for each parameter on the SDS
!
      DO j = 1, Julian_Time_Numparams
         retn = sfscatt(sdsid, Julian_Time_Name_Attr_Names(j), DFNT_CHAR8,              &
                        Julian_Time_Names_Char(j), Julian_Time_Names(j))
         retn = sfscatt(sdsid, Julian_Time_Units_Attr_Names(j), DFNT_CHAR8,
      &
                             Julian_Time_Units_Char(j), Julian_Time_Units(j))
      ENDDO
      istat = sfendacc (sdsid)
      retn = sfend(hdf_id)
```

4.0 Creating Attributes on Vdatas and Vdata Fields

Vdata and Vdata field attributes are similar to SDS attributes. Any number of attributes can be assigned to a Vdata or Vdata field. Each Vdata attribute name must be unique and each Vdata field attribute name must be unique.

To write attributes on a Vdata, the following calls are necessary to open the HDF file, start the Vdata interface, locate the Vdata, and attach the Vdata for writing:

```
   DATA dfacc_rdwr / 3 /
   hdf_id    = hopen(hdf_file, dfacc_rdwr, 0)
   CALL vfstart (hdf_id)
   vdata_ref = vsffnd(hdf_id,vdataname)
   vdata_id  = vsfatch(hdf_id,vdata_ref, 'w')
```

In the above calls, the value of *dfacc_rdwr* allows reading from and writing to the HDF file. *vdataname* is the name of the Vdata on the HDF file, and *vdata_id* is the identifier of the Vdata.

If a Vdata field attribute is being set, **vsffidx** is called to first get the index, *field_index*, of the specified field, *fieldname*, as shown in the following example. This call is not necessary when setting a Vdata attribute.

```
   CALL vsffidx (vdata_id, fieldname, field_index)
```

Next call **vsfscat** to set a character string attribute for the field:

```
   CALL vsfscat (vdata_id, field_index, attr_name, data_type, count, values)
```

or call **vsfsnat** to set a numeric field attribute:

```
   CALL vsfsnat (vdata_id, field_index, attr_name, data_type, count, values)
```

In the above examples, *vdata_id* is the Vdata identifier returned by the call to **vsfatch**. The argument, *values*, is the value of the attribute with an attribute name of *attr_name*. *count* is the number of values in the attribute of type, *data_type*. For calls to **vsfscat**, where the attribute value is a character string, *count* is the number of characters in the character string. If the attribute is a Vdata attribute, *field_index* should be set to -1 in the call to **vsfscat** or **vsfsnat**.

4.1 Example of Placing Vdata and Vdata Field Attributes on the SSF HDF File

In the following example, a Vdata attribute, named 'SSF_Header Default Attributes', contains the default values for the units and ranges. Individual field attributes are defined containing the name of the field and any range, unit, or type that is different from the default values. The names of the Vdata fields are stored in the fattr array. Arrays tempchar, attr1, attr2, and attr3 contain character strings that are concatenated on the end of the field attribute name, ‘Attributes for SSF-H’.

```
 INTEGER                          :: i, hdf_id, retn
 INTEGER                          :: vsfatch, vsffnd, vsfscat, vsfdtch, vfend
 INTEGER                          :: vdata_ref, stat, vdata_id, hclose, hopen
 INTEGER, PARAMETER   :: DFNT_CHAR8 =4
 INTEGER, PARAMETER   :: felem(23) =                                                        &
   (/43,60,53,53,77,59,121,96,106,105,104,103,91,73,62,62,62,62,62,62,55,55,55/)
 CHARACTER (LEN=255)  :: hdf_file           ! hdf file of SSF data
 CHARACTER (LEN=2)      :: tempchar(23)
 CHARACTER(LEN=22)     :: aname
 CHARACTER(LEN=56)     :: attrname
 CHARACTER(len=121)      :: fattr(23)
 CHARACTER(len=88)       :: vd_attr
 CHARACTER (LEN=1),   PARAMETER :: ssf_access = ‘r’     ! access flag to the ssf
 CHARACTER(LEN=20),  PARAMETER :: fattrname =  ‘Attributes for SSF-H’
 CHARACTER(LEN=29),  PARAMETER :: vdattrname = ‘SSF_Header Default Attributes’
 CHARACTER(LEN=34),  PARAMETER :: attr1 = ‘ - Name, Unit, Valid Range, & Type’
 CHARACTER(LEN=28),  PARAMETER :: attr2 = ‘ - Name, Valid Range, & Type’
 CHARACTER(LEN=16),  PARAMETER :: attr3 = ‘ - Name and Type’
 vd_attr = ‘Unless defined as a separate field attribute, the range is ASCII string; the unit is N/A’
 fattr(1)  = ‘SSF_ID : range is 112 - 200; 32-bit integer’
 fattr(2)  = ‘Character name of CERES instrument : 4-byte character string’
 fattr(3)  = ‘Day and Time at hour start : 28-byte character string’
 fattr(4)  = ‘Character name of satellite : 4-byte character string’
 fattr(5)  = ‘Character name of high resolution imager instrument : 8-byte character string’
 fattr(6)  = ‘Number of imager channels : range is 1 - 20; 32-bit integer’
 fattr(7)  = ‘Central wavelengths of imager channels : unit is micron; range is 0.4 - 15.0, 32-bit
float - up to 20 elements per record’
 fattr(8)  = ‘Earth-Sun distance at hour start : unit is Astronomical Unit; range is 0.98 - 1.02; 32-
bit float’
 fattr(9)  = ‘Colatitude of subsatellite point at surface at hour start : unit is degree; range is 0 -
180; 32-bit float’
 fattr(10) = ‘Longitude of subsatellite point at surface at hour start : unit is degree; range is 0 -
360; 32-bit float’
 fattr(11) = ‘Colatitude of subsatellite point at surface at hour end : unit is degree; range is 0 -
180; 32-bit float’
 fattr(12) = ‘Longitude of subsatellite point at surface at hour end : unit is degree; range
is 0 - 360; 32-bit float’
 fattr(13) = ‘Along-track angle of satellite at hour end : unit is degree; range is 0 - 330;
32-bit float’
 fattr(14)    = ‘Number of Footprints in SSF product : range is 0 - 360000; 32-bit integer’
 fattr(15)    = ‘Subsystem 4.1 identification string : 128-byte character string’
 fattr(16)    = ‘Subsystem 4.2 identification string : 128-byte character string’
```

```
  fattr(17)    = ‘Subsystem 4.3 identification string : 128-byte character string’
  fattr(18)    = ‘Subsystem 4.4 identification string : 128-byte character string’
  fattr(19)    = ‘Subsystem 4.5 identification string : 128-byte character string’
  fattr(20)    = ‘Subsystem 4.6 identification string : 128-byte character string’
  fattr(21)    = ‘IES production date and time : 24-byte character string’
  fattr(22)    = ‘MOA production date and time : 24-byte character string’
  fattr(23)    = ‘SSF production date and time : 24-byte character string’
  tempchar(1)  = ‘1’
  tempchar(2)  = ‘2’
  tempchar(3)  = ‘3’
  tempchar(4)  = ‘4’
  tempchar(5)  = ‘5’
  tempchar(6)  = ‘6’
  tempchar(7)  = ‘7’
  tempchar(8)  = ‘8’
  tempchar(9)  = ‘9’
  tempchar(10) = ‘10’
  tempchar(11) = ‘11’
  tempchar(12) = ‘12’
  tempchar(13) = ‘13’
  tempchar(14) = ‘14’
  tempchar(15) = ‘15’
  tempchar(16) = ‘16’
  tempchar(17) = ‘17’
  tempchar(18) = ‘18’
  tempchar(19) = ‘19’
  tempchar(20) = ‘20’
  tempchar(21) = ‘21’
  tempchar(22) = ‘22’
  tempchar(23) = ‘23’
!
!  The HDF file must be opened via the Vdata open routine, hopen, in order to place attributes
!  on Vdatas or Vdata fields.  In this example, the Vdata is opened with read/write access by
using
!  the dfacc_rdwr argument.
!
          dfacc_rdwr = 3
          hdf_id    = hopen(hdf_file, dfacc_rdwr, 0 )
          CALL vfstart (hdf_id)
          vdata_ref = vsffnd(hdf_id,’SSF_Header’)
          vdata_id  = vsfatch(hdf_id,vdata_ref, ‘w’)
          stat      = vsfscat(vdata_id,-1,vdattrname,DFNT_CHAR8, 88, vd_attr)
          DO i = 1,23
            aname = trim(fattrname) // tempchar(i)
            SELECT CASE(i)
              case (2:5,15:23)
```

```
         attrname = trim(aname) // attr3
     case (1,6,14)
         attrname = trim(aname) // attr2
     case (7:13)
         attrname = trim(aname) // attr1
   END SELECT
   stat      = vsfscat(vdata_id,i-1,attrname,DFNT_CHAR8,felem(i),fattr(i))
ENDDO
CALL vsfdtch (vdata_id)
CALL vfend(hdf_id)
stat = hclose(hdf_id)
```

APPENDIX A: Sample SDS Viewed with the "hdp" Utility

To view the Julian_Time SDS on the HDF file, CER_SSF_TRMM-PFM-VIRS_Sample_000004.1998010500**,** using the hdp utility, use the following command:

```
hdp dumpsds -n Julian_Time CER_SSF_TRMM-PFM-VIRS_Sample_000004.1998010500
```

```
File name: CER_SSF_TRMM-PFM-VIRS_Sample_000004.1998010500

Variable Name = Julian_Time
        Index = 0
        Type= 64-bit floating point
        Ref. = 3
        Rank = 2
        Number of attributes = 5
        Dim0: Name=Footprints(FOV)
                Size = 5
                Type = number-type not set
                Number of attributes = 0
        Dim1: Name=Julian_Time Parameters
                Size = 2
                Type = number-type not set
                Number of attributes = 0
        Attr0: Name = _FillValue
                Type = 64-bit floating point
                Count= 1
                Value =
17976931348623157000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
0.000000

        Attr1: Name = long_name of SDS Parameter 1
                Type = 8-bit signed char
                Count= 19
                Value = Time of observation
        Attr2: Name = units for SDS Parameter 1
                Type = 8-bit signed char
                Count= 3
                Value = day
        Attr3: Name = long_name of SDS Parameter 2
                Type = 8-bit signed char
                Count= 55
                Value = Radius of satellite from center of Earth at ob
```

```
           servation
Attr4: Name = units for SDS Parameter 2
        Type = 8-bit signed char
        Count= 9
        Value = kilometer
Data :
       2450818.501440 6720.555481
       2450818.501426 6720.551019
       2450818.501440 6720.555520
       2450818.501426 6720.550981
       2450818.501440 6720.555558
```

APPENDIX B: Sample Vdata Viewed with the “hdp” Utility

To view the SSF_Header Vdata on the HDF file, CER_SSF_TRMM-PFM-VIRS_Sample_000004.1998010500, using the hdp utility, use the following command:

```
hdp dumpvd -n  SSF_Header CER_SSF_TRMM-PFM-VIRS_Sample_000004.1998010500
```

```
File name: CER_SSF_TRMM-PFM-VIRS_Sample_000004.1998010500

Vdata: 0
   tag = 1962; reference = 2;
   number of records = 1; interlace = 0;
   fields = [SSF_ID, INST_NAME, DATA_DATE, SAT_NAM, IMAG_NAME,
          NUMCHANNELS, WAVELENGTH, EARTH_SUN,COLAT_SUBSAT_BEG,
              LONG_SUBSAT_BEG,COLAT_SUBSAT_END, ];
   record size (in bytes) = 1332;
   name = SSF_Header; class = SSF;
   number of attributes = 1
    attr0: name=SSF_Header Default Attributes type=4 count=88 size=88
        U n l e s s   d e f i n e d   a s   a   s e p a r a t e
          f i e l d   a t t r i b u t e ,   t h e   r a n g e
        i s   A S C I I   s t r i n g ;   t h e   u n i t   i s
          N / A
- field index 0: [SSF_ID], type=24, order=1
   number of attributes = 1
    attr0: name=Attributes for SSF-H1 - Name, Valid Range, & Type type=4 count=43 size=43
        S S F _ I D   :   r a n g e   i s   1 1 2   -   2 0 0 ;
          3 2 - b i t   i n t e g e r
- field index 1: [INST_NAME], type=4, order=32
   number of attributes = 1
    attr0: name=Attributes for SSF-H2 - Name and Type type=4 count=60 size=60
        C h a r a c t e r   n a m e   o f   C E R E S   i n s t
        r u m e n t   :   4 - b y t e   c h a r a c t e r   s t
        r i n g
- field index 2: [DATA_DATE], type=4, order=224
   number of attributes = 1
    attr0: name=Attributes for SSF-H3 - Name and Type type=4 count=53 size=53
        D a y   a n d   T i m e   a t   h o u r   s t a r t   :
          2 8 - b y t e   c h a r a c t e r   s t r i n g
- field index 3: [SAT_NAM], type=4, order=32
   number of attributes = 1
    attr0: name=Attributes for SSF-H4 - Name and Type type=4 count=53 size=53
        C h a r a c t e r   n a m e   o f   s a t e l l i t e
        :   4 - b y t e   c h a r a c t e r   s t r i n g
- field index 4: [IMAG_NAME], type=4, order=64
```

```
  number of attributes = 1
   attr0: name=Attributes for SSF-H5 - Name and Type type=4 count=77 size=77
      C h a r a c t e r   n a m e   o f   h i g h   r e s o l
      u t i o n   i m a g e r   i n s t r u m e n t   :   8 -
      b y t e   c h a r a c t e r   s t r i n g
- field index 5: [NUMCHANNELS], type=24, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H6 - Name, Valid Range, & Type type=4 count=59 size=59
      N u m b e r   o f   i m a g e r   c h a n n e l s   :
      r a n g e   i s   1   -   2 0 ;   3 2 - b i t   i n t e
      g e r
- field index 6: [WAVELENGTH], type=5, order=20
  number of attributes = 1
   attr0: name=Attributes for SSF-H7 - Name, Unit, Valid Range, & Type type=4 count=121
size=121
      C e n t r a l   w a v e l e n g t h s   o f   i m a g e
      r   c h a n n e l s   :   u n i t   i s   m i c r o n ;
       r a n g e   i s   0 . 4   -   1 5 . 0 ,   3 2 - b i t
       f l o a t   -   u p   t o   2 0   e l e m e n t s   p
      e r   r e c o r d
- field index 7: [EARTH_SUN], type=5, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H8 - Name, Unit, Valid Range, & Type type=4 count=96
size=96
      E a r t h - S u n   d i s t a n c e   a t   h o u r   s
      t a r t   :   u n i t   i s   A s t r o n o m i c a l
      U n i t ;   r a n g e   i s   0 . 9 8   -   1 . 0 2 ;
      3 2 - b i t   f l o a t
- field index 8: [COLAT_SUBSAT_BEG], type=5, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H9 - Name, Unit, Valid Range, & Type type=4 count=106
size=106
      C o l a t i t u d e   o f   s u b s a t e l l i t e   p
      o i n t   a t   s u r f a c e   a t   h o u r   s t a r
      t   :   u n i t   i s   d e g r e e ;   r a n g e   i s
       0   -   1 8 0 ;   3 2 - b i t   f l o a t
- field index 9: [LONG_SUBSAT_BEG], type=5, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H10 - Name, Unit, Valid Range, & Type type=4 count=105
size=105
      L o n g i t u d e   o f   s u b s a t e l l i t e   p o
      i n t   a t   s u r f a c e   a t   h o u r   s t a r t
       :   u n i t   i s   d e g r e e ;   r a n g e   i s
      0   -   3 6 0 ;   3 2 - b i t   f l o a t
- field index 10: [COLAT_SUBSAT_END], type=5, order=1
  number of attributes = 1
```

```
   attr0: name=Attributes for SSF-H11 - Name, Unit, Valid Range, & Type type=4 count=104
size=104
     C o l a t i t u d e   o f   s u b s a t e l l i t e   p
     o i n t   a t   s u r f a c e   a t   h o u r   e n d
     :   u n i t   i s   d e g r e e ;   r a n g e   i s   0
      -   1 8 0 ;   3 2 - b i t   f l o a t
- field index 11: [LONG_SUBSAT_END], type=5, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H12 - Name, Unit, Valid Range, & Type type=4 count=103
size=103
     L o n g i t u d e   o f   s u b s a t e l l i t e   p o
     i n t   a t   s u r f a c e   a t   h o u r   e n d   :
      u n i t   i s   d e g r e e ;   r a n g e   i s   0
     -   3 6 0 ;   3 2 - b i t   f l o a t
- field index 12: [MAX_ATRACK], type=5, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H13 - Name, Unit, Valid Range, & Type type=4 count=91
size=91
     A l o n g - t r a c k   a n g l e   o f   s a t e l l i
     t e   a t   h o u r   e n d   :   u n i t   i s   d e g
     r e e ;   r a n g e   i s   0   -   3 3 0 ;   3 2 - b i
     t   f l o a t
- field index 13: [NUMFOV], type=24, order=1
  number of attributes = 1
   attr0: name=Attributes for SSF-H14 - Name, Valid Range, & Type type=4 count=73 size=73
     N u m b e r   o f   F o o t p r i n t s   i n   S S F
     p r o d u c t   :   r a n g e   i s   0   -   3 6 0 0 0
     0 ;   3 2 - b i t   i n t e g e r
- field index 14: [SS4_1_IDSTR], type=4, order=128
  number of attributes = 1
   attr0: name=Attributes for SSF-H15 - Name and Type type=4 count=62 size=62
     S u b s y t e m   4 . 1   i d e n t i f i c a t i o n
     s t r i n g   :   1 2 8 - b y t e   c h a r a c t e r
     s t r i n g
- field index 15: [SS4_2_IDSTR], type=4, order=128
  number of attributes = 1
   attr0: name=Attributes for SSF-H16 - Name and Type type=4 count=62 size=62
     S u b s y t e m   4 . 2   i d e n t i f i c a t i o n
     s t r i n g   :   1 2 8 - b y t e   c h a r a c t e r
     s t r i n g
- field index 16: [SS4_3_IDSTR], type=4, order=128
  number of attributes = 1
   attr0: name=Attributes for SSF-H17 - Name and Type type=4 count=62 size=62
     S u b s y t e m   4 . 3   i d e n t i f i c a t i o n
     s t r i n g   :   1 2 8 - b y t e   c h a r a c t e r
     s t r i n g
```

```
- field index 17: [SS4_4_IDSTR], type=4, order=128
  number of attributes = 1
   attr0: name=Attributes for SSF-H18 - Name and Type type=4 count=62 size=62
      S u b s y t e m  4 . 4  i d e n t i f i c a t i o n
      s t r i n g  :  1 2 8 - b y t e  c h a r a c t e r
      s t r i n g
- field index 18: [SS4_5_IDSTR], type=4, order=128
  number of attributes = 1
   attr0: name=Attributes for SSF-H19 - Name and Type type=4 count=62 size=62
      S u b s y t e m  4 . 5  i d e n t i f i c a t i o n
      s t r i n g  :  1 2 8 - b y t e  c h a r a c t e r
      s t r i n g
- field index 19: [SS4_6_IDSTR], type=4, order=128
  number of attributes = 1
   attr0: name=Attributes for SSF-H20 - Name and Type type=4 count=62 size=62
      S u b s y t e m  4 . 6  i d e n t i f i c a t i o n
      s t r i n g  :  1 2 8 - b y t e  c h a r a c t e r
      s t r i n g
- field index 20: [IES_DATE], type=4, order=32
  number of attributes = 1
   attr0: name=Attributes for SSF-H21 - Name and Type type=4 count=55 size=55
      I E S  p r o d u c t i o n  d a t e  a n d  t i m e
       :  2 4 - b y t e  c h a r a c t e r  s t r i n g
- field index 21: [MOA_DATE], type=4, order=32
  number of attributes = 1
   attr0: name=Attributes for SSF-H22 - Name and Type type=4 count=55 size=55
      M O A  p r o d u c t i o n  d a t e  a n d  t i m e
       :  2 4 - b y t e  c h a r a c t e r  s t r i n g
- field index 22: [SSF_DATE], type=4, order=32
  number of attributes = 1
   attr0: name=Attributes for SSF-H23 - Name and Type type=4 count=55 size=55
      S S F  p r o d u c t i o n  d a t e  a n d  t i m e
       :  2 4 - b y t e  c h a r a c t e r  s t r i n g
Loc.    Data
0      112 P F M                                   1 9 9 8 - 0 1 - 0 5 T 0 0 : 0 0 : 0 0 . 0 0 0 0
0 0 Z
T R M M                              V I R S
5  0.630000 3.750000 10.800000 11.900000 1.600000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000  0.983293  58.834675  167.263962  94.638138  25.117634  236.194702  5  C
R H  A l b e d o  1 9 9 7 - 1 1 - 0 3 T 1 8 : 4 7 : 5 2 . 6 0 0 0 0 0 Z  C R H  B T e m p  1 9 9 7 -
1 1 - 0 3 T 1 8 : 4 7 : 5 2 . 6 0 0 0 0 0 Z  I m a g e r  1         5 . 1       1 9 9 8 - 0 4 - 0 1 T 1 7
: 4 1 : 5 0             E l v 0 0 0 0 3  H 2 0 0 0 0 0 3  S n o w 0 0 3 . 1  I c e 0 0 3 . 1  I G B P 0
0 0 0 3  T e r r 0 0 0 0 3  I D 0 0 0 0 3   E m 0 3 7 5 0 0 0 0 4   E m 1 0 8 0 0 0 0 0 4   E m 1 1
9 0 0 0 0 0 4                              D 0 0 0 0 3  E B i 0 0 0 0 3  B i 0 0 0 0 3  P h i 0 0
0 0 3  T 0 0 0 0 4  O D 0 0 0 0 3  0 0 0 0 3  V I N T r a y b r e f  b d n n r e f  m o d e l s n e w .
```

```
nnel3 nnel4 nnel5                    SCCR#   71 19980331 P
aram 19980120 CERES10 PSF 19980120 SARB 00003
98%RAPS SCC_TRM_19980401 CADM_SW_19971101 CADM_
WN_19971101
MOA Production Date = 1998-02-06T20:11:04\000“\000\000\000
1998-03-23T18:38:20.000Z       1998-02-06T20:11:04
1998-04-08T14:10:59           ;
```

APPENDIX C: HDF Routines Used in the Examples

Table 1: Equivalent Fortran and C HDF Routines for SDSs

Fortran Routines	C Routines	Purpose
sfcreate	SDcreate	Creates a new SDS.
sfgdinfo	SDdiminfo	Gets information about an SDS dimension.
sfendacc	SDendaccess	Disposes of an SDS identifier.
sfdimid	SDgetdimid	Gets the dimension identifier associated with a dimension in an SDS.
sfscatt/sfsnatt	SDsetattr	Defines a user-defined attribute for an SDS variable.
sfsdtstr	SDsetdatastrs	Sets the pre-defined attributes (label, unit, format, and coordinate system strings) for an SDS.
sfsdmname	SDsetdimname	Assigns a name to an SDS dimension.
sfsdmstr	SDsetdimstrs	Sets the predefined attributes (label, unit, and format) for a particular SDS dimension.
sfsfill/sfscfill	SDsetfillvalue	Sets the fill value for an SDS.
sfsrange	SDsetrange	Sets the valid maximum and minimum values for an SDS.
sfstart	SDstart	Opens the HDF file and initializes the SD interface.

Table 2: Equivalent Fortran and C HDF Routines for Vdatas (1 of 2)

Fortran Routines	C Routines	Purpose
hclose	Hclose	Closes the HDF file.
hopen	Hopen	Opens an existing HDF file or creates a new HDF file.
vfend	Vend	Releases the internal data structures for an HDF file.
vfstart	Vstart	Initializes internal data structures for an HDF file.
vsfatch	VSattach	Attaches an existing Vdata or creates a new Vdata.
vsfdtch	VSdetach	Detaches the Vdata from further access.
vsffnd	VSfind	Searches an HDF file for a particular Vdata.

Table 2: Equivalent Fortran and C HDF Routines for Vdatas (2 of 2)

Fortran Routines	C Routines	Purpose
vsffidx	VSfindex	Retrieves the index of a field on a Vdata.
vsfsnat/vsfscat	VSsetattr	Sets an attribute for a Vdata or Vdata field.