Semper 6

READ/WRITE

file format

Synoptics

Semper 6 READ/WRITE file format

This document describes in complete detail the various Fortran file formats handled by Semper's READ and WRITE commands.

A file can be formatted or unformatted. Formatted files can be transferred from one Semper installation to another, but tend to be large and slow to access. Unformatted files are recommended for use on a given computer installation, for transferring picture data between Semper and other image processing packages. Note that when inputing an unformatted picture file with the READ command, the UNFORMATTED option must be specified, otherwise READ will assume that it is to read a Fortran formatted picture file.

The details for the file formats are presented in the form of example Fortran coding as this reflects exactly how the READ and WRITE commands operate.

A picture file can be broken down into four component parts:

(1) Header record

(2) Title record

(3) Label record

(4) Picture data

The header record and the picture data are always present in a picture file. The title record is present if the picture has a title. The label record is optional. The value of IFLAG specifies whether either record is present in the picture file. It also records the format in which the title record is stored. The WRITE command omits the label record if the option UNLABELLED is specified.

The label record contains all of the data that makes up the picture label. Some of this information is duplicated in the header record, which is why the READ command can manage without the label record: it constructs a picture label using the information in the header record.

The picture file is read or written in the following way, according to whether the file is formatted or unformatted:

(1) Header record

Contains picture size, class, data form, title string length and label record flag. For a Fortran formatted file, a format string for reading the pixel data is also included in the header record.

Formatted READ:

```
READ (IUNIT,'(6I6,1X,A20)') NCOL,NROW,NLAY,ICLASS,IFORM,IFLAG,FORMAT
```

Formatted WRITE:

```
WRITE (IUNIT,'(6I6,1X,A20)') NCOL,NROW,NLAY,ICLASS,IFORM,IFLAG,FORMAT
```

NB for unformatted READ/WRITE all these 2- variables must be declared INTEGER*2 (Default is INTEGER*4)

Unformatted READ:

```
READ (IUNIT) NCOL,NROW,NLAY,ICLASS,IFORM,IFLAG
```

Unformatted WRITE:

```
WRITE (IUNIT) NCOL,NROW,NLAY,ICLASS,IFORM,IFLAG
```

(2) Title record

Contains a string of up to 156 characters representing the picture title. This record is not present if the title string length NTITLE is zero. The form in which the characters are stored depends on the value of IVERSN. If IVERSN is zero, the string is stored as an array of integer Hollerith values. Otherwise, the string is stored in Fortran 77 character format.

Formatted READ:

```
IF (IVERSN.EQ.0) THEN
   READ (IUNIT,'(80A1)') (ITITLE(I),I=1,NTITLE)
ELSE
   READ (IUNIT,'(80A1)') (TITLE(I:I),I=1,NTITLE)
ENDIF
```

Formatted WRITE:

```
IF (IVERSN.EQ.0) THEN
   WRITE (IUNIT,'(80A1)') (ITITLE(I),I=1,NTITLE)
ELSE
   WRITE (IUNIT,'(80A1)') (TITLE(I:I),I=1,NTITLE)
ENDIF
```

Unformatted READ:

```
IF (IVERSN.EQ.0) THEN
   READ (IUNIT) (ITITLE(I),I=1,NTITLE)
ELSE
   READ (IUNIT) TITLE(1:NTITLE)
ENDIF
```

Unformatted WRITE:

```
IF (IVERSN.EQ.0) THEN
   WRITE (IUNIT) (ITITLE(I),I=1,NTITLE)
ELSE
   WRITE (IUNIT) TITLE(1:NTITLE)
ENDIF
```

(3) Label record

Contains 256 integer values that represent the contents of the Semper picture label. This record is not present if the label record flag ILABEL is set to zero.

Formatted READ:

```
READ (IUNIT,'(16I4)') (LABEL(I),I=1,256)
```

Formatted WRITE:

```
WRITE (IUNIT,'(16I4)') (LABEL(I),I=1,256)
```

Unformatted READ:

```
READ (IUNIT) (LABEL(I),I=1,256)
```

Unformatted WRITE:

```
WRITE (IUNIT) (LABEL(I),I=1,256)
```

(4) Picture data

The remaining data in the file records the pixel values for the picture. The number of picture columns must not exceed the maximum row length for a given Semper installation (the Semper command SHOW SYSTEM will display this information). The number of picture rows and layers is limited by the space available in Semper's picture disc. Pixel data in byte or integer form are recorded as integer values and pixel data in floating-point or complex form are recorded in floating-point form as follows:

Data in byte or integer form (IFORM = 0 or 1):

```
READ/WRITE ( ..... ) (IPIXEL(I),I=1,NCOL)
```

Data in floating-point form (IFORM = 2):

```
READ/WRITE ( ..... ) (RPIXEL(I),I=1,NCOL)
```

Data in complex form (IFORM = 3):

```
READ/WRITE ( ..... ) (RPIXEL(I),I=1,2*NCOL)
```

or

```
READ/WRITE ( ..... ) (CPIXEL(I),I=1,NCOL)
```

When writing pixel data in Fortran formatted form, care must be taken to ensure that a suitable format is used. The format string may be specified to the WRITE command by means of the FORMAT key, e.g.

```
WRITE NAME ' .... ' FORMAT '(1X,10F8.2)'
```

If no format string is specified, the following default formats are used:

Data in byte form (IFORM = 0):

```
FORMAT = '(1X,24I3)'
```

Data in integer form (IFORM = 1):

```
FORMAT = '(1X,12I6)'
```

Data in floating-point or complex form (IFORM = 2 or 3):

```
FORMAT = '(1X,1P6E12.5)'
```

The pixel values are recorded in the file with the picture row data arranged in the following way:

Formatted READ:

```
      DO 200 K=1,NLAY
         DO 100 J=1,NROW
            READ (IUNIT,FORMAT) ( ......... )
100      CONTINUE
200   CONTINUE
```

Formatted WRITE:

```
      DO 200 K=1,NLAY
         DO 100 J=1,NROW
            WRITE (IUNIT,FORMAT) ( ......... )
100      CONTINUE
200   CONTINUE
```

Unformatted READ:

```
      DO 200 K=1,NLAY
         DO 100 J=1,NROW
            READ (IUNIT) ( ......... )
100      CONTINUE
200   CONTINUE
```

Unformatted WRITE:

```
      DO 200 K=1,NLAY
         DO 100 J=1,NROW
            WRITE (IUNIT) ( ......... )
100      CONTINUE
200   CONTINUE
```

The Fortran variables and arrays used in the coding examples above are defined as follows:

```
INTEGER IUNIT
INTEGER NCOL,NROW,NLAY
INTEGER ICLASS,IFORM
INTEGER IFLAG,NTITLE,ILABEL,IVERSN
INTEGER ITITLE(156),LABEL(256)
INTEGER ICCOLN,ICROWN,ICLAYN
INTEGER IWP
INTEGER IYEAR,IMONTH,IDAY
INTEGER IHOUR,IMINUT,ISEC
INTEGER NCRANG
INTEGER IPLTYP
INTEGER IPIXEL( .... )
REAL    RPIXEL( .... )
COMLPEX CPIXEL( .... )
CHARACTER*20 FORMAT
CHARACTER*156 TITLE

EQUIVALENCE (RPIXEL,CPIXEL)
```

IUNIT - Fortran unit for accessing picture file

NCOL - number of picture columns
NROW - " " " rows
NLAY - " " " layers

ICLASS - picture class
= 1, image
= 2, macro
= 3, fourier
= 4, spectrum
= 5, correlation
= 6, undefined
= 7, walsh
= 8, position list
= 9, histogram
= 10, display look-up table

IFORM - picture data form
= 0, byte
= 1, integer
= 2, floating-point
= 3, complex

NTITLE - number of characters in picture title
= 0, title record not present in picture file
ILABEL - label record flag
= 0, label record not present in picture file
= 1, label record present in picture file
IVERSN - version number
= 0, title stored in Hollerith form
= 1, title stored in character form
IFLAG - picture title and label flag
= 10000*IVERSN + 1000*ILABEL + NTITLE

ITITLE - array of NTITLE integer ASCII codes making up the picture title string (not used if NTITLE = 0)

LABEL - array of 256 integer values making up the picture label (not used if ILABEL = 0)

```
ICCOLN - column number of picture origin
ICROWN - row      "    "      "       "
ICLAYN - layer    "    "      "       "

IWP    - write protect flag
         = 0, picture is not write-protected
         = 1,    "    "  write-protected

IYEAR,IMONTH,IDAY - creation date of picture

IHOUR,IMINUT,ISEC - creation time of picture

NCRANG - number of characters that encode the range of intensity
         values in the picture data
         = 0, picture range is not recorded

IPLTYP - position list type
         = 0, picture is not a position list
         = 1, position list
         = 2, open curve
         = 3, closed curve

FORMAT - character variable containing the format string used to
         write the picture data in formatted picture file
```

If complete information about a picture is desired, the data in the picture label must be arranged in the following way:

```
LABEL(1 to 6) - integer ASCII codes for the string 'Semper'
                = 83, 101, 109, 112, 101 and 114 always

NCOL = 256*LABEL(7) + LABEL(8)
NROW = 256*LABEL(9) + LABEL(10)
NLAY = 256*LABEL(11) + LABEL(12)

ICCOLN = 256*LABEL(13) + LABEL(14)
ICROWN = 256*LABEL(15) + LABEL(16)
ICLAYN = 256*LABEL(17) + LABEL(18)

ICLASS = LABEL(19)

IFORM = LABEL(20)

IWP = LABEL(21)

IYEAR = LABEL(22) + 1900
IMONTH = LABEL(23)
IDAY = LABEL(24)

IHOUR = LABEL(25)
IMINUT = LABEL(26)
ISEC = LABEL(27)

NCRANG = LABEL(28)

LABEL(29 to NCRANG+28) - integer ASCII codes for string encoding the
                         picture range as two floating-point decimal
                         strings separated by a comma ','.
```

LABEL(NCRANG+29 to 55) - not used (set to zero)

IPLTYP = LABEL(56)

LABEL(57 to 99) - not used (set to zero), but reserved for later use

NTITLE = LABEL(100)

LABEL(101 to NTITLE+100) - integer ASCII codes making up the picture title string

LABEL(NTITLE+101 to 256) - not used (set to zero)