Multimission Software Interface Specification (SIS)

SPICE
Instrument Kernel

IK

NAIF Document No. 369
Version 1.0

Prepared by: C. Acton
Navigation and Ancillary Information Facility (NAIF)
Jet Propulsion Laboratory
National Aeronautics and Space Administration

PURPOSE: This SIS describes the format and content of SPICE Instrument Kernel (IK) file, used to establish instrument field-of-view geometric characteristics and other instrument ancillary data.

CHANGE LOG

Version	Date	Page Nos.	Reason
1.0	14 June 2000	All	New multimission version.

List of Acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee on Space Data Standards
CK	SPICE C-kernel
ET	Ephemeris Time
FK	SPICE Frames Kernel
FTP	File Transfer Protocol
FTS	SFOC File Transfer Service
IK	SPICE Instrument Kernel
JPL	Caltech/Jet Propulsion Laboratory
MSOO	Mars Surveyor Operations Office
NAIF	Navigation and Ancillary Information Facility
PDB	Project Data Base
PDS	Planetary Data System
SFDU	Standard Formatted Data Unit
SIS	Software Interface Specification
SPICE	S-, P-, I-, C- and E-kernels; the principal logical data components of a particular NASA ancillary information system
TMOD	Telecommunications and Mission Operations Directorate
VMS	Digital Equipment Corporation's Virtual Memory Operating System

Section 1
General Description

1.1 Purpose of Document

This Software Interface Specification (SIS) describes the contents and structure of a SPICE Instrument Kernel (IK) file.

1.2 Scope

This is a multimission SIS, applicable for all flight projects.

1.3 Applicable Documents

No.	**Document ID**	**Version**	**Title**
1.	NAIF Doc. No. 318	Latest release	Kernel Required Reading

Also useful as a reference is the I-Kernel tutorial, available from the NAIF server:

ftp://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/

1.4 Functional Description

An instrument kernel contains specifications for the size, shape and orientation of an instrument field(s)-of-view. It may also contain specifications for other instrument characteristics useful in constructing observation geometry or operating characteristics used in planning or interpreting observations.

Although not a technical requirement from a SPICE software perspective, normally a separate IK file is produced for each instrument.

1.4.1 Data Source, Destinations, and Transfer Method

Instrument Kernels are made available to flight projects through whatever mechanism is used for providing access to SPICE products, such as a Project Database (PDB), a File Interchange System (FIS) or a SPICE Server.

1.4.2 Labeling and Identification

IK files may include internal identification information, although there is no general requirement for such. IK file names may utilize any syntax picked by a flight project, although limiting the length to the "27.3" specification adopted by the Planetary Data System (PDS) is suggested. NAIF further suggests using the "*.ti" standard generally used by NAIF.

1.4.3 Assumptions and Constraints

Contents of an IK file must adhere to SPICE text kernel specifications as described in NAIF Document "Kernel Required Reading," Reference 1.

Section 2
Data Object Definition

2.1 Structure and Organization

An instrument kernel is a simple ASCII file containing data sections and descriptive text sections. The structure conforms to the specifications described in Reference 1.

Text sections of an IK are used to describe the data. They are preceded by the token:

```
\begintext
```

which must appear on a line by itself.

If it appears first in the file, before any data, the first text section does not need this delimiter—it is interpreted as a text section by default.

The initial text section may contain labels (metadata) providing provenance for the file. This labeling practice is highly recommended by NAIF, although it is not a SPICE requirement. Such labels may utilize the same "keyword = value" syntax used in data sections of the IK. In general the text sections are not restricted to a particular format other than each line must not exceed 79 characters.

All data sections start with the begin data delimiter,

```
\begindata
```

which must appear on a line by itself.

Data are provided using a "keyword = value" syntax. The data sections are parsed by SPICE kernel file readers and so must adhere to the format specified in the NAIF Document Kernel Required Reading (Reference 1).

2.2 Data Format and Definition

2.2.1 Metadata Description

At the beginning of the file is usually found an introduction to the instrument kernel. This may include version and date, references, author and general comments about using the IK file.

2.2.2 Data Description

Instrument Kernel data follow the specifications described in Reference 1 (Kernel Required Reading), using a KEYWORD = VALUE syntax. Each assignment must appear on a separate line and each line must not exceed 79 characters.

2.2.3 Keyword Syntax

All keywords in an instrument kernel are constructed using the template INS[ID]_[WORD] where [ID] is replaced with the NAIF instrument ID and [WORD] completes the keyword.

2.2.4 Data Items

The number of assignments included in an instrument kernel is not specifically limited, but the total number of assignments provided from all text kernels used at run time must not exceed 10,000. Each instrument team is free to request assignments if they do not violate SPICE text kernel requirements and are generally consistent with existing data and SPICE usage.

Frequently used keywords are described below, but not every IK will use every keyword. **The reader is advised to examine the specific IK files created for a specific project to see exactly what assignments are being used.**

2.2.4.1 Frequently Used Keywords

FOV_FRAME	The name of the frame in which the field of view boundary vectors and boresight are defined.
FOV_BOUNDARY_CORNERS	An array of vectors that point to the ``corners" of the instrument field of view. (See the discussion of FOV_SHAPE below for an expansion of the term ``corners of the field of view." Note that the vectors stored in FOV_BOUNDARY_CORNERS are not necessarily unit vectors.)
FOV_SHAPE	A character string that describes the ``shape" of the field of view. The only acceptable values are: 'POLYGON', 'RECTANGLE', 'CIRCLE' and 'ELLIPSE'. If the value of FOV_SHAPE is 'POLYGON' the field of view of the instrument is a pyramidal polyhedra. The vertex of the pyramid is at the instrument and the rays along the edges of the pyramid are parallel to the vectors stored in FOV_BOUNDARY_CORNERS. If the value of FOV_SHAPE is 'RECTANGLE' the field of view of the instrument is pyramid with a rectangular base centered about the boresight vector. Four vectors are stored in FOV_BOUNDARY_CORNERS. These vectors are parallel to the edges of the rectangular cone. If the value of FOV_SHAPE is 'CIRCLE' the field of view of the instrument is a circular cone about the boresite vector. A single vector is stored in FOV_BOUNDARY_CORNERS. This vector is parallel to a ray that lies in the cone that makes up the boundary of the field of view. If the value of FOV_SHAPE is 'ELLIPSE' the field of view of the instrument is a elliptical cone with the boresite vector as the axis of the cone. In this case two vectors are stored in FOV_BOUNDARY_CORNERS. One of the vectors stored in FOV_BOUNDARY_CORNERS points to the end of the semi-major axis of a cross section whose normal is parallel to the axis of the elliptic cone. The other vector points to the end of the semi-minor axis in the same cross section.
BORESIGHT	A vector in the field of view that points in the direction to the FOV center. The length of BORESIGHT is not specified

	other than being non-zero. For circular, elliptical, and rectangular fields of view, the boresight should be the central axis of the field of view.

2.2.4.2 Examples of Frequently Used Keywords

In the "ASCII art" drawings used below the "*" (asterisk) is used to represent corners of a figure, the "x" character represents a vector pointing into the page, and the "o" character (lower case "oh") is used to represent vectors pointing up out of the page, and also to signify units of degrees.

2.2.4.2.1 Circular FOV

In this example we will be considering an instrument whose field of view is circular and whose boresight lies along the Z-axis in its instrument frame. For the purposes of this example, the instrument ID is -11111 and the angular separation of the field of view is 30 degrees.

Since the FOV is circular we need a single boundary corner vector that lies along the edge of the FOV cone. For simplicities sake, consider a vector that points into the plane Z=1 in the instrument frame. So we have this, when viewed from the side.

```
  ^ X
  |  ins
  |
  |       /|
  |     /  |
  |   /    |
  | /   o  |
  |/ 15.0  |
  o--------------->
Y  \       |     Z
 ins \     |      ins
      \    |
        \  |
          \|

  |-- 1.0 --|
               Plane Y = 0
```

where the Y axis of the instrument frame points out of the page or screen, and the 15.0 degree angle of the triangle is half angle of interest. So to lay out the boundary vector that lies along the top half of the diagram we have:

Vector = (1.0 * tan(15.0 degrees), 0.0, 1.0)

since the point lies in the planes Z=1.0 and Y=0.0. The keywords and the values that define this field of view would be:

```
INS-11111_FOV_FRAME            = 'FRAME_FOR_INS-11111'
INS-11111_FOV_BOUNDARY_CORNERS = ( 0.267949192431  0.0  1.0 )
INS-11111_FOV_SHAPE            = 'CIRCLE'
INS-11111_BORESIGHT            = ( 0.0  0.0  1.0 )
```

2.2.4.2.2 Elliptical FOV

Now consider an instrument with an elliptical field of view centered on the instrument boresight which is parallel to the Z-axis of the instrument frame. The semi-major axis of the ellipse is parallel to the X-axis in the instrument frame, and the semi-minor axis is parallel to the Y-axis. For the purposes of this example assign the ID code -22222 to this instrument.

With the elliptical case, we need two boundary corner vectors. One that points along the edge of the elliptical cone that intercepts the semi-major axis of the ellipse, and the other the semi-minor axis.

For simplicities sake, compute the boundary vectors that point to the plane Z=1 in the instrument frame. So we have the two following diagrams to consider:

Semi-Major Axis Boundary Vector:

```
  ^ X
  |  ins
  |
  |         /|
  |       /  |
  |     /    |
  |   /   o  |
  |/ 45.0    |
  o---------------->
Y  \         |     Z
 ins \       |      ins
       \     |
         \   |
           \ |

  |-- 1.0 --|
               Plane Y = 0
```

Taking the 45 degree half angle we compute the boundary corner vector must have the following components:

Vector = (1.0 * tan(45.0 degrees), 0.0, 1.0)

Now repeat the process for the semi-minor axis. This time the diagram will be looking into the X=0 plane, where the X-axis points into the page or screen.

Semi-Minor Axis Boundary Vector:

```
  ^ Y
  |  ins
  |
  |        /|
  |      /  |
  |    /    |
  |  /   o  |
  |/ 30.0   |
  x---------------->
X  \        |    Z
 ins \      |     ins
       \    |
         \  |
           \|

  |-- 1.0 --|
               Plane X = 0
```

Here the 30 degree half angle gives the second boundary corner vector components:

Vector = (0.0, 1.0 * tan(30.0 degrees), 1.0)

Putting all of this together yields the following keywords and values:

```
INS-22222_FOV_FRAME            = 'FRAME_FOR_INS-22222'
INS-22222_FOV_BOUNDARY_CORNERS = (
                          1.0  0.0            1.0
                          0.0  0.57735026919  1.0
                                 )
INS-22222_FOV_SHAPE            = 'POLYGON'
INS-22222_BORESIGHT            = ( 0.0  0.0  1.0 )
```

2.2.4.2.3 Rectangular FOV

For the following example consider an instrument with a rectangular field of view whose boresight axis lies along the Z-axis in the instrument frame. Let the instrument ID code be -33333. The diagram below illustrates the field of view:

```
                     ^ Y
                     |  ins
                     |
                     |
                     |
  -  *--------------------------------*
  |  |                                |
  |  |                                |
  | o|                                |
20.0 |               o                |------>
  |  |                Z               |       X
  |  |                 ins            |        ins
  |  |                                |
  -  *--------------------------------*
                       o
     |-------------45.0---------------|
```

As with the other cases, consider the boundary vectors that lie in the Z=1 plane. Then looking up the Y-axis of the instrument frame we have:

```
  ^ X
  |  ins
  |
  |        /|
  |      /  |
  |    /    |
  |  /   o  |
  |/ 10.0   |
  o---------------->
Y  \        |     Z
 ins \      |      ins
       \    |
         \  |
           \|

  |-- 1.0 --|
           Plane Y = 0
```

where the 10.0 degree angle of the triangle is the half angle of interest that yields the magnitude of the X components according to the following relation:

$$X = 1.0 * \tan(10.0 \text{ degrees})$$

Similarly for Y, looking down the X-axis of the instrument frame we have:

```
  ^ Y
  |  ins
  |
  |        /|
  |      /  |
  |    /    |
  |  /   o  |
  |/ 22.5   |
  x--------------->
X  \        |    Z
 ins \      |     ins
       \    |
         \  |
           \|

  |-- 1.0 --|
               Plane X = 0
```

where the 22.5 degree angle of the triangle is the half angle of interest that gives the magnitude of the Y components according to the following relation:

Y = 1.0 * tan(22.5 degrees)

Now collecting all of this together we obtain the following keywords and values to define this FOV:

```
INS-33333_FOV_FRAME            = 'FRAME_FOR_INS-33333'
INS-33333_FOV_BOUNDARY_CORNERS = (

             0.176326980708   0.414213562373  1.0
            -0.176326980708   0.414213562373  1.0
            -0.176326980708  -0.414213562373  1.0
             0.176326980708  -0.414213562373  1.0
                                   )
INS-33333_FOV_SHAPE            = 'RECTANGLE'
INS-33333_BORESIGHT            = ( 0.0  0.0  1.0 )
```

In the case of the rectangular fields of view, the boundary corner vectors must be listed in the order they would be encountered traversing the edges of a cross section of the FOV cone.

2.2.4.2.4 Polygonal FOV

For the following example consider an instrument with a trapezoidal field of view whose boresight axis lies along the Z-axis in the instrument frame. Let the instrument ID code be -44444. The diagram below illustrates the field of view:

```
                               ^ Y
                               |  ins
                               |
                               |
                 (-1,1)        |        (1,1)
                      *-----------------*
                     /                   \
                    /                     \
                   /     (0,0)             \
                  /            o            \---------->
                 /              Z            \          X
                /                ins          \          ins
               /                               \
              *---------------------------------*
        (-2,-1)                                  (2,-1)

                                                      Plane Z =
1
```

As with the other cases, consider the boundary vectors that lie in the Z = 1 plane. Since the above diagram illustrates a cross-section of the FOV in this plane, we need only to collect the coordinates of the corner vectors where they intersect the plane. Collecting all of this together we obtain the following keywords and values to define this FOV:

```
INS-44444_FOV_FRAME             = 'FRAME_FOR_INS-44444'
INS-44444_FOV_BOUNDARY_CORNERS = (

       1.0        1.0       1.0
      -1.0        1.0       1.0
      -2.0       -1.0       1.0
       2.0       -1.0       1.0
                                  )
INS-44444_FOV_SHAPE             = 'POLYGON'
INS-44444_BORESIGHT             = ( 0.0  0.0  1.0 )
```

In the case of the polygonal fields of view, the boundary corner vectors must be listed in the order they would be encountered traversing the edges of a cross section of the FOV cone.

2.3 Access Routines

2.3.2 Loading and I-Kernel

Before you can access data in an I-kernel it is necessary to “load” that file (or files). As for all SPICE text-format kernels, the act of “loading” the kernel transfers all of the keyword = value assignments contained in the data sections of the text kernel into computer memory. Loading is accomplished with the FURNSH module, found in both the SPICELIB and CSPICE toolkit libraries.

```
CALL FURNSH ( <IK file name>)          FORTRAN version
furnish_c   (<IK file name>);          C version
```

2.3.2 High-level Access Routine

A single subroutine named GETFOV (or getfov_c, for CSPICE) provides a common interface to retrieving the geometric characteristics of an instrument field of view for a wide variety of remote sensing instruments across many different space missions.

Given the NAIF instrument ID, this routine returns the bore-sight of the instrument, the "shape" of the field of view, a collection of vectors that point along the edges of the field of view, and the name of the reference frame in which these vectors are defined.

It is expected that users of this routine will be familiar with the SPICE frames subsystem and will be comfortable writing software to further manipulate the vectors retrieved by this routine.

GETFOV has the following input and output arguments.

```
CALL GETFOV ( INSTID, ROOM, SHAPE, FRAME, BSIGHT, N, BOUNDS )
```

The input arguments are:

INSTID NAIF ID of an instrument
ROOM Maximum number of FOV boundary vectors that can be returned in the BOUNDS output argument

The output arguments are:

SHAPE Instrument FOV shape, selected from one of the following allowed values: ‘POLYGON’, ‘RECTANGLE’, ‘CIRCLE’, ‘ELIPSE’
FRAME Name of the reference frame used for specifying the FOV vectors

BSIGHT   Boresight vector, specifying the “look direction” of the center of the FOV, given in the FRAME reference frame

N   Number of FOV boundary vectors returned

BOUNDS   A 3 x N array containing the N returned FOV boundary vectors

Further discussion about the operation of and restrictions on GETFOV, and an example of its use solving a typical space science problem (determine the planetocentric latitude and longitude of the points where the corners of a rectangular field of view intersect the surface of an object) may be found in the module’s SPICELIB or CSPICE source code header.

2.3.3 Low-level Access Routines

As for all SPICE text kernels, the SPICE Toolkit contains low level routines to retrieve keywords and the value field(s) assigned to them. The FORTRAN SPICELIB subroutines GCPOOL, GDPOOL, and GIPOOL are available for this purpose. The modules gcpool_c, gdpool_c and gipool_c are the analogs for the C version of the SPICE Toolkit (CSPICE).