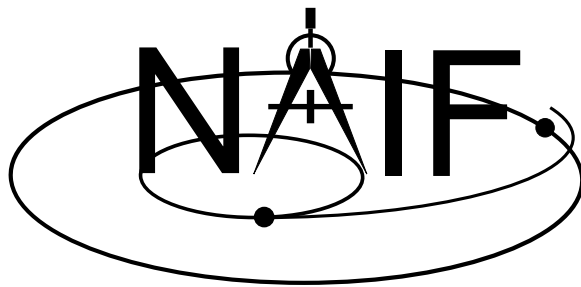


Optiks User's Guide



Navigation Ancillary Information Facility

Prepared by
F. S. Turner

NAIF Document No. 377.01
Last Modified: 17 JAN 2002

Table of Contents

OPTIKS User's Guide	1
Revisions	1
June 25, 2002	1
October 24, 2000	1
Summary	1
Usage	2
The OPTIKS Report	3
Header	3
Angular Extent Table	4
Boresight Vector Table	4
Controlling OPTIKS Report Output with Command Line Options	5
Customizing the Angular Extent Report	5
Customizing the Boresight Vector Report	6
Sample Command Lines and Their Outputs	7
Appendix A - FOV Definitions	11
Appendix B - Computing Angular Extents	13
Circular FOVs	14
Elliptical FOVs	15
Rectangular FOVs	17
Appendix C - Using C-Kernels and PC-Kernels with OPTIKS	18
Appendix D - Using Meta-Kernels	18
Appendix E - "Loading" Kernel Files	19

OPTIKS User's Guide

Revisions

June 25, 2002

Added documentation for the new command line keyword “-showfovframes” now understood by OPTIKS. When coupled with “-frame FRAMENAME”, OPTIKS displays the name of the FOV base frame in the boresight vector table.

Corrected discussion in Appendix D about meta-kernels. The samples utilize a keyword “PATH_NAMES”, and this has been changed to the proper keyword “PATH_VALUES”.

October 24, 2000

Initial release.

Summary

OPTIKS is a utility program that generates information about instrument fields of view from parameters present in SPICE Instrument and Frame Kernels (IK and FK).

The OPTIKS program uses a set of SPICE kernels specified on the command line. One or more may be 'FURNISH' meta-kernels (See Kernel Required Reading or Appendix D for details). The information provided is used to compute instrument field of view (FOV) specifications that are not explicitly called out in the FOV definitions found in instrument kernels.

The output data are organized in two tables. The first lists the angular extents (size) of circular, elliptical, and rectangular fields of view. The user

can select the unit of measure for the angular measurements, and whether half or full angular extents are listed.

The second table specifies, whenever possible, field of view boresights in a user specified frame at a particular epoch. Depending on the output frame requested by the OPTIKS user, one or more C-kernels and/or PC-kernels may be needed.

Usage

OPTIKS is invoked on the command line as shown below. Items appearing inside of square brackets are optional. The list of kernels and/or meta-kernels must follow any optional arguments.

Program usage:

```
> optiks [OPTIONS]... KERNEL ...
> optiks [OPTIONS]... META-KERNEL ...
```

The available options are listed below. The order of the options that precede the kernel list is unimportant.

-half	display the FOV angular extents as half angles, as opposed to the default full angles.
-units	specify the units used to display the angular quantities. The default is RADIANS; other common values include: DEGREES, MILLIRADIANS.
-frame	indicates the frame in which the boresight vectors are to be computed. Any frame known to SPICE after loading the specified kernels may be used. This keyword is required to display the boresight table.
-epoch	sets the epoch used to compute the instrument boresights in the requested frame. '01-JAN-2000-12:00UTC' is the default value. The characters specifying the epoch must consist of a continuous string of non-blank characters.
-showfovframes	includes the FOV base frame name in the list of boresight vectors that are computed. The base frame is the reference frame in which the the instrument boresight and boundary vectors are specified.

<code>-u, -usage</code>	display basic usage information.
<code>-h, -help</code>	display brief help text.
<code>-v, -version</code>	list information about the program version.

The table of boresights is displayed only if the `-frame` option is specified on the command line. In the absence of this option, the table output is suppressed.

Kernels are loaded in the order they are specified from left to right on the command line. (See Appendix E for a discussion on the data precedence order determined by the order in which files are loaded.)

At least one IK file and one FK file must be loaded to make full use of OPTIKS (unless instrument information normally distributed between these two kinds of files has been placed in a single text kernel).

OPTIKS requires a SPICE leapseconds kernel (LSK) be included in the set of kernels to load—otherwise necessary time conversions will fail.

The OPTIKS Report

The OPTIKS program generates a report containing three parts: the header, the angular extent table, and the boresight vector table. These tables are computed based on the contents of the kernels loaded. An entry for each properly defined and loaded FOV definition is made in both tables.

Header

The OPTIKS report header contains a time stamp and a list of loaded kernels. Only kernels that are not classified as meta-kernels are listed. Further, the paths to the kernel files are removed when the list is generated. A sample header follows:

```
Report Generated at: 10-24-2000 16:01
optiks 1.0.0 (SPICE Toolkit N0051)
```

```
Kernels Loaded:
```

```
naif0007.tls
msc_ins1_v00.ti
msc_ins2_v04.ti
msc_ins3_v10.ti
msc_ins4_v25.ti
msc_v41.tf
```

The kernels are listed top to bottom in the order that they are loaded (top = first loaded).

Angular Extent Table

The angular extent table follows the header. It provides information about each loaded FOV definition's shape and size. The shapes supported are:

- CIRCLE
- RECTANGLE
- ELLIPSE

FOVs that have other shapes supported by SPICE are listed, but no angles are computed. A sample table of the default output is shown below.

```

FOV full-angular extents computed in RADIANS

Field of View      Shape           Length          Width
-----
-100021            CIRCULAR        +0.24123410     +0.24123410
TEST_DET1          CIRCULAR        +1.57000000     +1.57000000
TEST_DET2          ELLIPTICAL      +3.20000000     +1.39999999
TEST_DET3          RECTANGULAR     +2.60000000     +1.40000000
TEST_DET4          POLYGONAL

```

The FOVs are listed in alphabetical order. In the event that an FOV does not possess a name, the associated ID code is listed in the table. The “Length” column lists the larger of the two angular extents of the field of view; the “Width” column lists the smaller dimension. The first line of the table indicates what units and extents are displayed.

Boresight Vector Table

The boresight vector table displays the FOV boresights computed in some frame at an epoch requested. This table is only displayed by OPTIKS if a frame is specified by the user on the command line with the -frame option. A sample table is shown below.

```

FOV boresights computed at epoch 2000-JAN-01 11:58
FOV boresights computed in frame TEST_SC_COORD

Field of View      Boresight Vector
-----
-100021            ( +1.0000000, +0.0000000, +0.0000000 )
TEST_DET1          Unable to compute boresight.
TEST_DET2          ( +0.0040698, -0.9999917, +0.0000000 )
TEST_DET3          ( -0.0004449, -0.9999999, +0.0000000 )
TEST_DET4          ( +0.0004449, -0.9999999, +0.0000000 )

```

Here we have selected the spacecraft frame (in this example we are using “TEST_SC_COORD”) which yields the table displayed above. In this case the detector TEST_DET1 is an instrument that articulates relative to the TEST_SC_COORD frame; an instrument C-kernel needs to be loaded to compute its boresight in the spacecraft frame.

Controlling OPTIKS Report Output with Command Line Options

This section focuses on customizing the OPTIKS output through the use of command line options. All options must precede the list of kernel files to load. Some options have an additional argument which is specified on the command line following the option. Every argument must be devoid of spaces. All options affecting the report output are displayed below. Those that take an argument have the default value listed with the option.

Customizing the Angular Extent Report

There are two command line options that impact the output generated for the angular extent table. They are:

<code>-half</code>	display the FOV angular extents as half angles, as opposed to the default full angles.
<code>-units RADIANS</code>	specify the units used to display the angular quantities. The default is RADIANS; other common values include: DEGREES, MILLIRADIANS.

The following is a list of the angles supported by the `-units` argument:

RADIANS
MILLIRADIANS
MICRORADIANS
NANORADIANS
DEGREES
ARCSECONDS
ARCMINUTES
SECONDDANGLES

MINUTEANGLES

HOURANGLES

The value following the -units option is case-insensitive.

Customizing the Boresight Vector Report

There are two command line options that control the output generated for the boresight vector table. They are:

<code>-frame <frame_name></code>	indicates the frame in which the boresight vectors are to be computed. Any frame known to SPICE after loading the specified kernels may be used. This keyword is required to display the boresight table.
<code>-epoch 01-JAN-2000-12:00UTC</code>	set the epoch used to compute the instrument boresights in the requested frame. '01-JAN-2000-12:00UTC' is the default value. The characters specifying the epoch must consist of a continuous string of non-blank characters.
<code>-showfovframes</code>	includes the FOV base frame name in the list of boresight vectors that are computed. The base frame is the reference frame in which the the instrument boresight and boundary vectors are specified.

The -frame option specifies the frame in which the boresight vectors are computed. This may be any frame available to the SPICE system natively or defined through the loaded sets of kernels. In most cases, setting this to the root frame of the spacecraft frame tree is the appropriate choice. A frame must be specified on the command line for OPTIKS to compute the boresight table.

The -epoch option allows the user to control the epoch at which boresights are computed. Utilizing this argument is necessary when frame definitions that vary with time are required to compute boresight vectors in the desired frame. For a list of supported input time strings see the STR2ET section

in Time Required Reading. It is important to note that although spaces are permitted in the reference, this program requires no spaces are present in the input time string. A list of sample time strings is displayed below.

- 01-JAN-2000-12:00:00.00UTC
- 10-FEB-2000-8:14:15.23TDB
- 26-JUN-1999-18:59TDT
- 2005-100T19:10:12.00

The `-showfovframes` inserts a column between the “Field of View” and “Bore-sight Vector” columns entitled “Field of View Frame”. This column lists the name of the frame in which the FOV vectors are computed and stored in the instrument kernel.

Sample Command Lines and Their Outputs

This portion of the documentation provides examples using Cassini orbiter kernels to demonstrate a few OPTIKS scenarios.

In each of the examples that follow the meta-kernel named `meta.tk` is loaded. The file’s contents are as follows:

```
KPL/MK

This kernel is the FURNSH meta-kernel that loads the appropriate
kernel files for the OPTIKS examples.

\begindata

PATH_SYMBOLS = ( 'LSK_DIR',
                 'IK_DIR',
                 'FK_DIR' )

PATH_VALUES = ( '/kernels/gen/lsk',
                '/kernels/Cassini/ik',
                '/kernels/Cassini/fk' )

KERNELS_TO_LOAD = (
    '$LSK_DIR/naif0007.tls',
    '$IK_DIR/cas_cda_v00.ti',
    '$IK_DIR/cas_cirs_v05.ti',
    '$IK_DIR/cas_iss_v07.ti',
    '$FK_DIR/cas_v28.tf'
)

\begintext
```

17 JAN 2002

Due to the limits imposed by the documentation formats used in the SPICE toolkit, the output reports displayed below have been modified slightly from their original format.

Example One: Full Angular Extents in RADIANS, Frame is CASSINI_SC_COORD

```
> optiks -frame CASSINI_SC_COORD meta.tk
```

```
Report Generated at: 10-24-2000 14:51  
optiks 1.0.0 (SPICE Toolkit N0051)
```

Kernels Loaded:

```
naif0007.tls  
cas_cda_v00.ti  
cas_cirs_v05.ti  
cas_iss_v07.ti  
cas_v28.tf
```

FOV full-angular extents computed in RADIANS

Field of View -----	Shape -----	Length -----	Width -----
CASSINI_CDA	CIRCULAR	+1.57079632	+1.57079632
CASSINI_CIRS_FP1	CIRCULAR	+0.00389208	+0.00389208
CASSINI_CIRS_FP3	RECTANGULAR	+0.00293215	+0.00029321
CASSINI_CIRS_FP4	RECTANGULAR	+0.00293215	+0.00029321
CASSINI_CIRS_RAD	CIRCULAR	+3.14159265	+3.14159265
CASSINI_ISS_NAC	RECTANGULAR	+0.00610865	+0.00610865
CASSINI_ISS_NAC_RAD	CIRCULAR	+3.14159265	+3.14159265
CASSINI_ISS_WAC	RECTANGULAR	+0.06073745	+0.06073745
CASSINI_ISS_WAC_RAD	CIRCULAR	+3.14159265	+3.14159265

FOV boresights computed at epoch 2000-JAN-01 12:00
FOV boresights computed in frame CASSINI_SC_COORD

Field of View -----	Boresight Vector -----
CASSINI_CDA	Unable to compute boresight.
CASSINI_CIRS_FP1	(+0.00406998, -0.99999171, +0.00000000)
CASSINI_CIRS_FP3	(-0.00044499, -0.99999990, +0.00000000)
CASSINI_CIRS_FP4	(+0.00044499, -0.99999990, +0.00000000)
CASSINI_CIRS_RAD	(+1.00000000, -0.00000000, +0.00000000)
CASSINI_ISS_NAC	(+0.00057595, -0.99999981, -0.00017097)
CASSINI_ISS_NAC_RAD	(+1.00000000, -0.00000000, +0.00000000)
CASSINI_ISS_WAC	(+0.00121834, -0.99999922, +0.00025445)
CASSINI_ISS_WAC_RAD	(+1.00000000, -0.00000000, +0.00000000)

Example Two: Half Angular Extents in MILLIRADIANS, Frame is CASSINI_SC_COORD

17 JAN 2002

```
> optiks -frame CASSINI_SC_COORD -half -units MILLIRADIANS meta.tk
```

Report Generated at: 10-24-2000 15:04
optiks 1.0.0 (SPICE Toolkit N0051)

Kernels Loaded:

```
naif0007.tls  
cas_cda_v00.ti  
cas_cirs_v05.ti  
cas_iss_v07.ti  
cas_v28.tf
```

FOV half-angular extents computed in MILLIRADIANS

Field of View	Shape	Length	Width
-----	-----	-----	-----
CASSINI_CDA	CIRCULAR	+7.8539E+02	+7.8539E+02
CASSINI_CIRS_FP1	CIRCULAR	+1.94604211	+1.94604211
CASSINI_CIRS_FP3	RECTANGULAR	+1.46607657	+0.14660765
CASSINI_CIRS_FP4	RECTANGULAR	+1.46607657	+0.14660765
CASSINI_CIRS_RAD	CIRCULAR	+1.5707E+03	+1.5707E+03
CASSINI_ISS_NAC	RECTANGULAR	+3.05432619	+3.05432619
CASSINI_ISS_NAC_RAD	CIRCULAR	+1.5707E+03	+1.5707E+03
CASSINI_ISS_WAC	RECTANGULAR	+3.0368E+01	+3.0368E+01
CASSINI_ISS_WAC_RAD	CIRCULAR	+1.5707E+03	+1.5707E+03

FOV boresights computed at epoch 2000-JAN-01 12:00
FOV boresights computed in frame CASSINI_SC_COORD

Field of View	Boresight Vector
-----	-----
CASSINI_CDA	Unable to compute boresight.
CASSINI_CIRS_FP1	(+0.00406998, -0.99999171, +0.00000000)
CASSINI_CIRS_FP3	(-0.00044499, -0.99999990, +0.00000000)
CASSINI_CIRS_FP4	(+0.00044499, -0.99999990, +0.00000000)
CASSINI_CIRS_RAD	(+1.00000000, -0.00000000, +0.00000000)
CASSINI_ISS_NAC	(+0.00057595, -0.99999981, -0.00017097)
CASSINI_ISS_NAC_RAD	(+1.00000000, -0.00000000, +0.00000000)
CASSINI_ISS_WAC	(+0.00121834, -0.99999922, +0.00025445)
CASSINI_ISS_WAC_RAD	(+1.00000000, -0.00000000, +0.00000000)

Example Three: Half Angular Extents in DEGREES, Frame is CASSINI_ISS_NAC

```
> optiks -frame CASSINI_ISS_NAC -half -units DEGREES meta.tk
```

Report Generated at: 10-30-2000 11:08
optiks 1.0.0 (SPICE Toolkit N0051)

Kernels Loaded:

```
naif0007.tls
```

17 JAN 2002

cas_cda_v00.ti
cas_cirs_v05.ti
cas_iss_v07.ti
cas_v28.tf

FOV half-angular extents computed in DEGREES

Field of View	Shape	Length	Width
-----	-----	-----	-----
CASSINI_CDA	CIRCULAR	+4.5000E+01	+4.5000E+01
CASSINI_CIRS_FP1	CIRCULAR	+0.11150000	+0.11150000
CASSINI_CIRS_FP3	RECTANGULAR	+0.08400000	+0.00840000
CASSINI_CIRS_FP4	RECTANGULAR	+0.08400000	+0.00840000
CASSINI_CIRS_RAD	CIRCULAR	+9.0000E+01	+9.0000E+01
CASSINI_ISS_NAC	RECTANGULAR	+0.17500000	+0.17500000
CASSINI_ISS_NAC_RAD	CIRCULAR	+9.0000E+01	+9.0000E+01
CASSINI_ISS_WAC	RECTANGULAR	+1.74000000	+1.74000000
CASSINI_ISS_WAC_RAD	CIRCULAR	+9.0000E+01	+9.0000E+01

FOV boresights computed at epoch 2000-JAN-01 12:00
FOV boresights computed in frame CASSINI_ISS_NAC

Field of View	Boresight Vector
-----	-----
CASSINI_CDA	Unable to compute boresight.
CASSINI_CIRS_FP1	(-0.00016577, +0.00349428, +0.99999388)
CASSINI_CIRS_FP3	(-0.00017249, -0.00102070, +0.99999946)
CASSINI_CIRS_FP4	(-0.00017116, -0.00013070, +0.99999997)
CASSINI_CIRS_RAD	(+0.00148701, +0.99999872, +0.00057595)
CASSINI_ISS_NAC	(0.00000000, 0.00000000, +1.00000000)
CASSINI_ISS_NAC_RAD	(+0.00148701, +0.99999872, +0.00057595)
CASSINI_ISS_WAC	(-0.00042446, +0.00064301, +0.99999970)
CASSINI_ISS_WAC_RAD	(+0.00148701, +0.99999872, +0.00057595)

Example Four: Full Angular Extents in RADIANS, Show FOV Frames,
Frame is CASSINI_SC_COORD.

```
> optiks -frame CASSINI_SC_COORD -showfovframes meta.tk
```

Report Generated at: 06-25-2002 17:51
optiks 2.0.0 (SPICE Toolkit N0052a)

Kernels Loaded:

cas_cda_v01.ti
cas_cirs_v09.ti
cas_iss_v08.ti
cas_v34.tf
naif0007.tls

FOV full-angular extents computed in RADIANS

Field of View	Shape	Length	Width
-----	-----	-----	-----

CASSINI_CDA	CIRCULAR	+1.57079632	+1.57079632
CASSINI_CIRS_FP1	CIRCULAR	+0.00390000	+0.00390000
CASSINI_CIRS_FP3	RECTANGULAR	+0.00300000	+0.00030000
CASSINI_CIRS_FP4	RECTANGULAR	+0.00300000	+0.00030000
CASSINI_CIRS_FPB	RECTANGULAR	+0.00300000	+0.00124000
CASSINI_CIRS_RAD	CIRCULAR	+3.14159265	+3.14159265
CASSINI_ISS_NAC	RECTANGULAR	+0.00610865	+0.00610865
CASSINI_ISS_NAC_RAD	CIRCULAR	+3.14159265	+3.14159265
CASSINI_ISS_WAC	RECTANGULAR	+0.06073745	+0.06073745
CASSINI_ISS_WAC_RAD	CIRCULAR	+3.14159265	+3.14159265

FOV boresights computed at epoch 2000-JAN-01 12:00

FOV boresights computed in frame CASSINI_SC_COORD

Field of View	Field of View Frame	Boresight Vector
-----	-----	-----
CASSINI_CDA	CASSINI_CDA	Unable to com ...
CASSINI_CIRS_FP1	CASSINI_CIRS_FP1	(+0.00567996, ...
CASSINI_CIRS_FP3	CASSINI_CIRS_FP3	(+0.00122999, ...
CASSINI_CIRS_FP4	CASSINI_CIRS_FP4	(+0.00216999, ...
CASSINI_CIRS_FPB	CASSINI_CIRS_FPB	(+0.00169999, ...
CASSINI_CIRS_RAD	CASSINI_CIRS_RAD	(+1.00000000, ...
CASSINI_ISS_NAC	CASSINI_ISS_NAC	(+0.00057595, ...
CASSINI_ISS_NAC_RAD	CASSINI_ISS_NAC_RAD	(+1.00000000, ...
CASSINI_ISS_WAC	CASSINI_ISS_WAC	(+0.00121834, ...
CASSINI_ISS_WAC_RAD	CASSINI_ISS_WAC_RAD	(+1.00000000, ...

The boresight vector table did not fit properly into all documentation formats. The “...” indicates continuation.

Appendix A - FOV Definitions

The following section describes in detail the keywords that are used to define fields of view in instrument kernels. As with other instrument kernel keywords, they are constructed using this template:

INS[ID]_[WORD]

where [ID] is replaced with the instrument ID and [WORD] is one of the following:

FOV_FRAME is the name of the frame in which the field of view boundary vectors and boresight are defined.

FOV_BOUNDARY_CORNERS is 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 is a character string that describes the “shape” of the field of view. The only acceptable values are:

'POLYGON'
'RECTANGLE'
'CIRCLE'
'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 a pyramid with a rectangular base centered about the bore-sight 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 bore-sight 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 boresight vector as the axis of the cone. In this case two vectors are stored in FOV_BOUNDARY_CORNERS. One of those vectors 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

is 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. No such restriction exists for arbitrary polygonal fields of view, but the boresight must be inside the FOV.

All of the four keywords must be present to properly define a FOV.

Appendix B - Computing Angular Extents

The following section describes how the angular extents are computed from the FOV definitions. The calculations discussed below explain how to obtain the half-angular extents—doubling them produces the full angles.

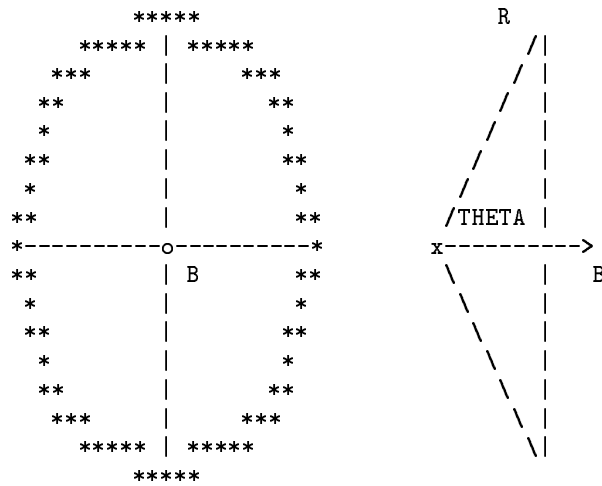
Circular FOVs

For the circular FOV case the following assignments are needed:

```

INS-11111_FOV_FRAME           = 'FRAME_FOR_INS-11111'
INS-11111_FOV_BOUNDARY_CORNERS = ( 0.000000 +0.939693 +0.342020 )
INS-11111_FOV_SHAPE           = 'CIRCLE'
INS-11111_FOV_BORESIGHT       = ( 0.000000 0.000000 1.000000 )
    
```

The keywords provide the boresight vector, B, and a single boundary corner vector, R. This boundary corner vector lies along the edge of the FOV cone. Looking in the plane the boresight and this radial component lie, we have a diagram similar to the one below:



From this diagram, the angle we are interested in is the angle between R and B.

Elliptical FOVs

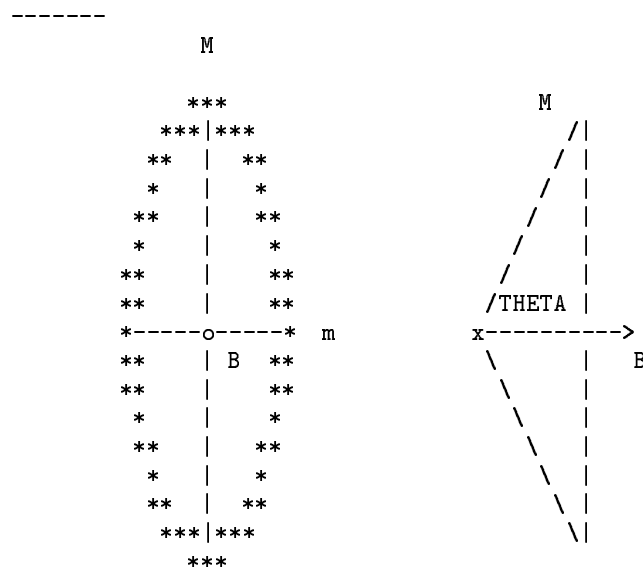
For the elliptical FOV case we have a set of keywords:

```

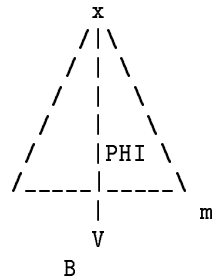
INS-22222_FOV_FRAME           = 'FRAME_FOR_INS-22222'
INS-22222_FOV_BOUNDARY_CORNERS = (
                                +0.939692  0.000000  +0.342020
                                0.000000  -0.500000  +0.866025
                                )
INS-22222_FOV_SHAPE           = 'POLYGON'
INS-22222_FOV_BORESIGHT       = ( 0.000000  0.000000  1.000000 )
    
```

The keywords provide the boresight vector, B, and two boundary corner vectors, M and m, the vectors that lie along the cone that intercept the semi-major and semi-minor axes of the elliptical cross section. Looking at the planes defined by the boresight and M and the boresight and m we have:

B and M:



B and m:



17 JAN 2002

From this diagram, the angles we are interested in are the angle between M and B, and the angle between m and B. The larger of these two angles is the length of the field of view, and the smaller is the width.

Rectangular FOVs

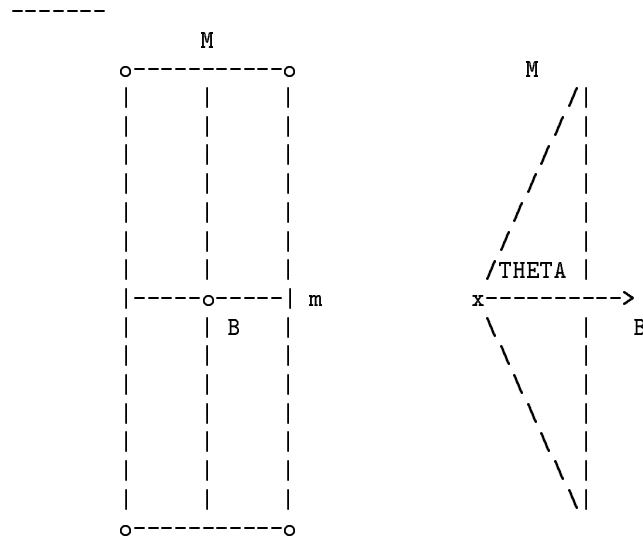
For the rectangular FOV case the following assignments are needed:

```

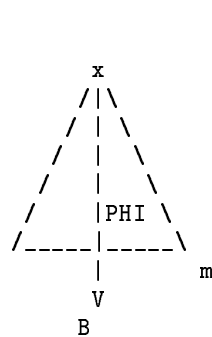
INS-33333_FOV_FRAME           = 'FRAME_FOR_INS-33333'
INS-33333_FOV_BOUNDARY_CORNERS = (
                                -0.193725 +0.921891 +0.335541
                                -0.193725 -0.921891 +0.335541
                                +0.193725 -0.921891 +0.335541
                                +0.193725 +0.921891 +0.335541
                                )
INS-33333_FOV_SHAPE           = 'RECTANGLE'
INS-33333_FOV_BORESIGHT      = ( 0.000000  0.000000  1.000000 )
    
```

The keywords provide a boresight vector, B, and four boundary corner vectors. Let M be the bisector of the first and second boundary corner vector and m be the bisector of the second and third boundary corner vectors. This yields a figure as shown below:

B and M:



B and m:



From the above diagrams the angles we are interested in are:

THETA = Separation between B and M

PHI = Separation between B and m

As in the elliptical case, the larger of these two angles is labelled length, and the smaller width.

Appendix C - Using C-Kernels and PC-Kernels with OPTIKS

To make use of frames that are tied to C-kernels or body definitions provided by PC-kernels, one simply needs to add the necessary kernels containing coverage for the epoch of interest to the loaded kernel list. In the case of C-kernels a corresponding SCLK kernel must be loaded to support the internal time conversion from spacecraft clock to ephemeris time.

Appendix D - Using Meta-Kernels

The following is an excerpt from Kernel Required Reading (kernel.req) provided in the /doc subdirectory with each SPICE Toolkit:

One prefixes with path symbols the kernel names specified in the KERNELS_TO_LOAD variable. Each symbol is prefixed with a dollar sign to indicate that it is in fact a symbol.

Suppose a set of MGS kernels reside in the path

```
/flight_projects/mgs/SPICE_kernels
```

and the other kernels reside in the path

```
/generic/SPICE_kernels
```

Then we can add paths to our metakernel as follows:

```
\begindata
PATH_VALUES = ( '/flight_projects/mgs/SPICE_kernels',
                '/generic/SPICE_kernels' )
PATH_SYMBOLS = ( 'MGS',
                 'GEN' )
```

```

KERNELS_TO_LOAD = ( '$GEN/leapseconds.ker',
                    '$MGS/mgs.tsc',
                    '$GEN/generic.bsp',
                    '$MGS/mgs.bc',
                    '$GEN/earth.bpc',
                    '$MGS/mgs.bes' )

```

It is not required that paths be abbreviated using path symbols; it's simply a convenience.

Note the symbols defined here are not related to the symbols supported by a host shell or any other operating system interface.

Appendix E - "Loading" Kernel Files

In SPICE terminology, "loading" a file means making the data in a file available to a SPICE application. In the case of text kernel files, such as IK, FK, PCK, SCLK and LSK, when a file is "loaded" the data in the text kernel are parsed and read into memory. In the case of binary kernel files, such as a CK or SPK, a pointer to the file is established, but data are not actually read into memory until a specific pointing request is made.

The data in SPICE kernel files used by a SPICE-based application are prioritized based on the order that the kernel files are "loaded" into the program, which is the order in which the kernel file names are specified on the command line (left to right) or in a command file (top to bottom). Data in files loaded earlier take precedence over data loaded later. This often makes no difference since one usually does not provide a program with two files containing "competing data" (see below), but when there are competing data these precedence rules must be understood.

What are "competing data"?

SPICE text kernels use a "keyword = value" syntax: this is called an assignment. In text kernels two assignments are "competing" if the keywords are identical. This is analogous to assigning two different values to a single variable in a piece of code, such as:

```

....
NOBJ = 3
NOBJ = 7
....

```

In both the code and the text kernel cases the last assignment has precedence (overrides all prior assignments).

What are competing data in SPICE binary kernels depends on the type of kernel in question. For C-kernel files (CK) that provide orientation information for some object, data are competing if the two data items (or data sets) provide orientation for the same object (e.g. a spacecraft, or an instrument's scanning mirror) at a given time. For SPK files that provide trajectory (ephemeris) information for a spacecraft or natural body, data are competing if the two data items (or data sets) provide state information for the same object (the "target") at a given time, independent of what is the observer and what is the reference frame.