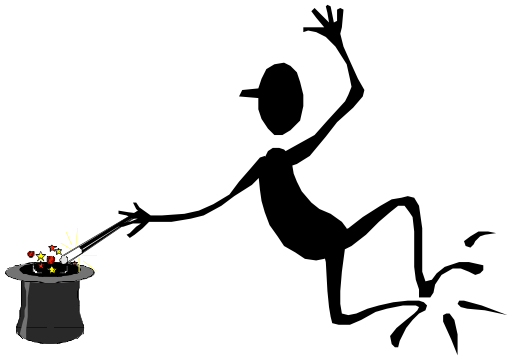
Using Module Headers

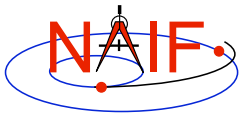


March 2006

Navigation and Ancillary Information Facility

Module Header Purpose

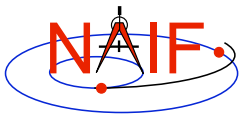
- Module“headers” are provided for all Toolkit modules, whether Fortran, C or IDL.
- This is the fundamental source for detailed user information and examples on how to use the module.
  - For some families of modules (e.g. SPK), additional information about the whole family is provided in a “required reading” document (e.g. …/doc/spk.req)
- Depending on the language, these “headers” may be found within the source code, or under the …/doc directory, or both. (See the next three pages.)



Fortran Module Header Locations

Navigation and Ancillary Information Facility

- For FORTRAN toolkits, the headers are provided within the source code modules.
  - YOURNAIFINSTALLATIONROOT/toolkit/src/spicelib/*.f or *.for
- In most cases there is a single “header” at the top of the source code.
- In a few cases, where a module has multiple entry points, there are additional “headers” at each entry point (e.g. “pool”).



C Module Header Locations

Navigation and Ancillary Information Facility

- For C toolkits the headers are provided in two formats, at two locations.
  - Plain text:
    - » YOURNAIFINSTALLATIONROOT/cspice/src/cspice/*.c
  - HTML:
    - » YOURNAIFINSTALLATIONROOT/cspice/doc/html/cspice/index.html

Icy Module Header Locations

Navigation and Ancillary Information Facility

- For IDL (“Icy”) toolkits, two kinds of headers are provided.
  - Icy wrappers in html format:
    - » YOURNAIFINSTALLATIONROOT/cspice/doc/html/icy/index.html
    - » This is a minimal set of information, and refers the user to the corresponding cspice header for details.
  - cspice headers, in text and html formats:
    - » YOURNAIFINSTALLATIONROOT/cspice/src/cspice/*.c
    - » YOURNAIFINSTALLATIONROOT/cspice/doc/html/cspice/index.html

CSPICE_SPKEZR Module Header I

Navigation and Ancillary Information Facility

- Let’s examine the information contained in an “Icy” module header using the SPKEZR module as example. Each module contains
  - Abstract
  - Inputs/Outputs (I/O)
  - Examples
  - Particulars
  - Required Reading
  - Version (History Information)
- Important: the information in an “Icy” wrapper is considered the bare minimum that might be needed by a user. More detail is provided in the corresponding cspice wrapper.

Abstract Information

Navigation and Ancillary Information Facility

-Abstract

CSPICE_SPKEZR returns the state (position and velocity) of a target body relative to an observing body, optionally corrected for light time (planetary aberration) and stellar aberration.

Input/Output Information - 1

Navigation and Ancillary Information Facility

-I/O

Given:

targ the scalar string name of a target body, optionally, you may supply the integer ID code for the object as an integer string, i.e. both "MOON" and "301" are legitimate strings that indicate the Moon is the target body,

The target and observer define a state vector whose position component points from the observer to the target.

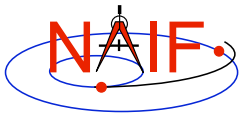
et the scalar or N-vector of double precision ephemeris time, expressed as seconds past J2000 TDB, at which the state of the target body relative to the observer is to be computed, 'et' refers to time at the observer's location

ref the scalar string name of the reference frame relative to which the output state vector should be expressed

abcorr the scalar string name of the aberration corrections to be applied to the state of the target body to account for one-way light time and stellar aberration

obs the scalar string name of a observing body, optionally, you may supply the integer ID code for the object as an integer string, i.e. both "MOON" and "301" are legitimate strings that indicate the moon is the observing body

continues



Input/Output Information - 2

Navigation and Ancillary Information Facility

cspice_spkezr, targ, et, ref, abcorr, obs, starg, ltime

returns:

starg

a double precision Cartesian 6-vector or 6xN array representing the position in kilometers and velocity in kilometers-per-second of the target body relative to the specified observer.

The first three components of 6-vector 'starg‘ represent the x-, y- and z-components of the target's position; the last three components form the corresponding velocity vector.

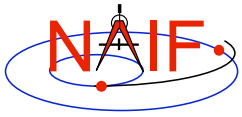
The 6xN form of 'starg' returns as starg[6,N]. Retrieve the ith state vector via:

state_i = state[*,i]

ltime

the double precision, scalar or N-vector, one-way light time between the observer and target in seconds; if the target state is corrected for aberrations, then 'ltime' is the one-way light time between the observer and the light time corrected target location

Please note, CSPICE documentation and source code uniformly uses the variable name 'lt' to designate the light-time between an observer and target. IDL uses "lt" as the less-than numeric comparison operator and so does not allow "lt" as a variable name.
Therefore, Icy documentation uses the name 'ltime’ for the light-time value.
'starg' and 'ltime' return with the same order (N) as 'et'.



Input/Output Information - 3

Navigation and Ancillary Information Facility

The cspice_spkezr function performs the same operation as

- cspice_spkez with the exception that cspice_spkezr expects arguments 'targ' and 'obs' to pass as strings, whereas
- cspice_spkez expects the arguments as integers. cspice_spkezr also accepts the string form of the integer NAIF IDs as inputs to 'targ' and 'obs', e.g.

```
targ = "Mars"
obs  = "Earth"
```

or (remember, string representation of the integers)

```
targ = "499"
obs  = "399"
```

Examples - 1

- Examples

Any numerical results shown for this example may differ between platforms as the results depend on the SPICE kernels used as input and the machine specific arithmetic implementation.

```
;;
;; Load a set of kernels: an SPK file, a PCK file and a leapseconds file. Use a meta
;; kernel for convenience.
;;
cspice_furnsh, "standard.ker"

;;
;; Define parameters for a state lookup:
;;
;; Return the state vector of Mars (499) as seen from Earth (399) in the J2000 frame
;; using aberration correction LT+S (light time plus stellar aberration) at the epoch
;; July 4, 2003 11:00 AM PST.
;;
target   = "Mars"
epoch    = 'July 4, 2003 11:00 AM PST'
frame    = "J2000"
abcorr   = "LT+S"
observer = "Earth"

;;
;; Convert the epoch to ephemeris time.
;;
cspice_str2et, epoch, et
```


Examples - 2


```
;; Look-up the state for the defined parameters.
;;
cspice_spkezr, target, et, frame, abcorr, observer, state, ltime
;;
;; Output...
print, 'The position of    : ', target
print, 'As observed from   : ', observer
print, 'In reference frame : ', frame
print, 'At epoch           : ', epoch
;; The first three entries of state contain the X, Y, Z position components. The final
;; three contain the Vx, Vy, Vz velocity components.
;;
print, 'Scalar'
print, 'R (kilometers)     : ', state[0:2]
print, 'V (kilometers/sec) : ', state[3:5]
print, 'Light time (secs)  : ', ltime
print, ' between observer'
print, ' and target'
;;
;; Create a vector of et's, starting at 'epoch‘  in steps of 100000 ephemeris seconds.
;;
SIZE   = 5
vec_et = dindgen( SIZE )*100000. + et
;; Look up the 'state' vectors and light time values 'lt'  corresonding to the vector of
;; input ephemeris time 'vec_et'.
cspice_spkezr, target  , vec_et, frame, abcorr, observer, state , ltime
print, 'Vector'
cspice_et2utc, vec_et, 'C', 3, vec_epoch
```

Examples - 3

Navigation and Ancillary Information Facility

```
;;
;; When called with a vector 'et', cspice_spkezr returns
;; 'state' as an 6xN matrix. Extract the ith state from the
;; matrix as:
;;
;;   state_i = state[*,i]
;;
for i=0, 4 do begin

   print, 'At epoch (UTC)     : ', vec_epoch[i]
   print, 'R (kilometers)     : ', (state[*,i])[0:2]
   print, 'V (kilometers/sec) : ', (state[*,i])[3:5]
   print, 'Light time (secs)  : ', ltime[i]
   print, ' between observer'
   print, ' and target'
   print

endfor

;;
;; It's always good form to unload kernels after use,
;; particularly in IDL due to data persistence.
;;
cspice_unload, "standard.ker"
```


Examples: Printout

Navigation and Ancillary Information Facility

```
IDL outputs:

The position of    : Mars
As observed from   : Earth
In reference frame : J2000
At epoch           : July 4, 2003 11:00 AM PST

Scalar
R (kilometers)     :       73822235.      -27127919.      -18741306.
     V (kilometers/sec) :      -6.8090923       7.5131823       3.0009890
Light time (secs)  :       269.68988
 between observer
 and target

Vector
At epoch (UTC)     : 2003 JUL 04 19:00:00.000
R (kilometers)     :       73822235.      -27127919.      -18741306.
V (kilometers/sec) :      -6.8090923       7.5131823       3.0009890
Light time (secs)  :       269.68988
 between observer
 and target

At epoch (UTC)     : 2003 JUL 05 22:46:40.000
R (kilometers)     :       73140185.      -26390525.      -18446763.
V (kilometers/sec) :      -6.8317855       7.2333512       2.8893940
Light time (secs)  :       266.56404
 between observer
 and target

... (etc.)
```

Particulars

Navigation and Ancillary Information Facility

-Particulars

None.

Note: this is an example of where important details are provided in the corresponding cspice module header (spkezr_c.c). There, the "Particulars" section is several pages long! It is not replicated here (in "Icy") just to reduce maintenance problems.

Required Reading

Navigation and Ancillary Information Facility

For important details concerning this module's function, please refer to the CSPICE routine spkezr_c.

ICY.REQ
SPK.REQ
FRAMES.REQ

A statement similar to this one appears in each "Icy" header.

-Icy Version 1.1.0, 01-AUG-2004, EDW (JPL)

Added capability to process vector 'et' input
returning a matrix 'starg' and vector 'ltime'
on output.

-Icy Version 1.0.0, 16-JUN-2003, EDW (JPL)