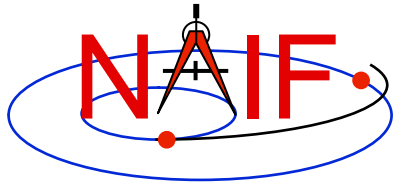


---

Navigation and Ancillary Information Facility

# Toolkit Applications

March 2010



# Toolkit Applications

---

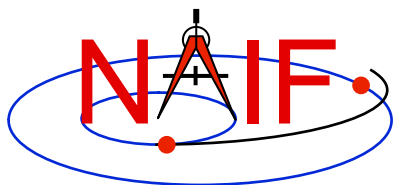
Navigation and Ancillary Information Facility

**Toolkit applications create or manipulate kernels, or perform other functions such as time conversion.**

**Each of these applications is included in the generic Toolkits.**

- Time conversion tool: *chronos*
- SPK generation tool: *mkspk*
- SPK merge and subset tool: *spkmerge*
- SPK comparison tool: *spkdiff*
- CK generation tool: *msopck*
- Frame comparison tool: *frmdiff*
- Kernel summary tools: *brief, ckbrief, spacit*
- Comments manipulation tools: *commnt, spacit*
- File format converters: *tobin, toxfr, and bingo\**

\* bingo is not included in generic Toolkits; it is available only from NAIF's webpages



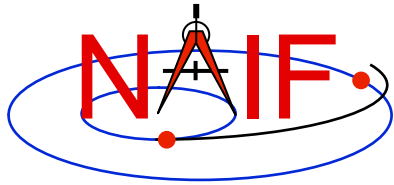
# CHRONOS

Navigation and Ancillary Information Facility

***chronos*** is an application that provides a flexible interface to the SPICE Toolkit time conversion capabilities.

***chronos*** supports time conversion between the following time systems/types:

<i>Supported Time Systems</i>	<i>--&gt;</i>	<i>Supported Time Types</i>
-----		-----
Universal Coord. Time (UTC)	-->	SCET, ERT, ETT, LT
Ephemeris Time (ET)	-->	SCET, ERT, ETT, SECONDS, LT
S/C On-board Clock Time (SCLK)	-->	SCLK, HEX, TICKS
Local Solar Time (LST)	-->	LST, LSUN



# CHRONOS - Input/Output Matrix

Navigation and Ancillary Information Facility

## *Input System/Type*

-----

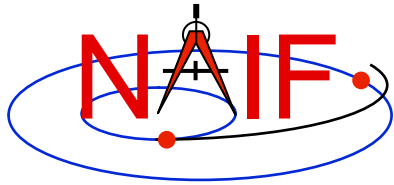
UTC / SCET (\*)  
 UTC / ERT  
 UTC / ETT  
 ET / SCET (\*)  
 ET / ERT  
 ET / ETT  
 ET / SECONDS  
 SCLK / SCLK (\*)  
 SCLK / HEX  
 SCLK / TICKS  
 LST / LST

## *Output System/Type*

-----

UTC / SCET (\*)  
 UTC / ERT  
 UTC / ETT  
 UTC / LT  
 ET / SCET (\*)  
 ET / ERT  
 ET / ETT  
 ET / SECONDS  
 ET / LT  
 SCLK / SCLK (\*)  
 SCLK / HEX  
 SCLK / TICKS  
 LST / LST (\*)  
 LST / LSUN

(\*) default input/output types

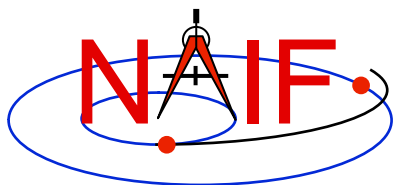


# CHRONOS - Miscellaneous

---

Navigation and Ancillary Information Facility

- ***chronos* normally converts one input time but can run in batch mode to speed up conversion for multiple input times.**
- **OS shell alias capabilities can be used to define shortcuts for commonly used time conversions.**
- ***chronos* has an extensive User's Guide.**



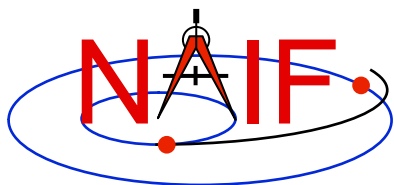
# CHRONOS - Usage

## Navigation and Ancillary Information Facility

```
Terminal Window
$ chronos
...
CHRONOS Usage
-----

To convert time from one supported system/type to another:

% CHRONOS -SETUP <setup file name OR kernel file name(s)>
  -FROM <"from" time system>
  [-FROMTYPE <"from" time type>]
  -TO <"to" time system>
  [-TOTYPE <"to" time type>]
  [-FORMAT <output time format picture>]
  -TIME <input time> | -BATCH
  [-SC <sc ID>]
  [-CENTER <cental body ID>]
  [-LANDINGTIME <UTC time of the landing>]
  [-SOL1INDEX <index of the first SOL>]
  [-NOLABEL]
  [-TRACE]
```



# CHRONOS - Example

Navigation and Ancillary Information Facility

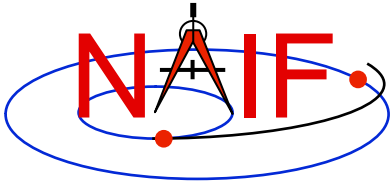
```
Terminal Window
$ cat chronos.cas
Sample CHRONOS setup file for Cassini
  \begindata
    KERNELS_TO_LOAD = ( 'naif0007.tls', 'cas00085.tsc' )
    SPACECRAFT_ID = -82
  \begintext

$ chronos -setup chronos.cas -from utc -to et -time 1999 JAN 12 12:00
1999-01-12, 12:01:04.184 (ET/SCET)

$ chronos -setup chronos.cas -from utc -to sclk -time 1999 JAN 12 12:00
1/1294833883.185 (SCLK/SCLK)

$ chronos -setup naif0007.tls cas00085.tsc -sc -82 -from sclk -to utc -time
1/1294833883.185
1999-01-12 11:59:59.998 (UTC/SCET)

$ chronos -setup naif0007.tls cas00085.tsc -sc -82 -from sclk -to utc -time
1/1294833883.185 -format 'YYYY-DOYTHR:MN:SC ::RND' -nolabel
1999-012T12:00:00
```



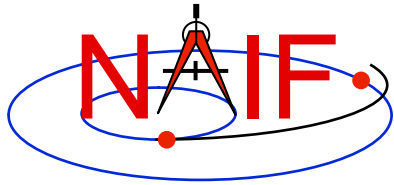
# MKSPK

---

Navigation and Ancillary Information Facility

- ***mkspk* may be used to generate an SPK file from any of several types of data, such as discrete states, classic elements, and two-line elements**
- **Use of this program is discussed in a separate tutorial about making SPK files, and in the *mkspk* User's Guide.**



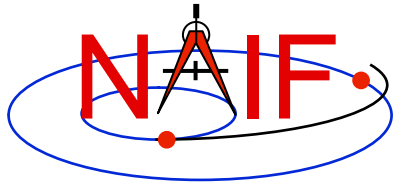


# SPKMERGE

---

Navigation and Ancillary Information Facility

- **The contents of an SPK file or set of SPK files may be merged or subsetted using *spkmerge***
  - Extract an interval of time of interest from a single SPK file or a set of SPK files.
  - Extract data for one or more objects from a single SPK file or a set of SPK files.
  - You can combine both the time and object selection mechanisms for the greatest flexibility.

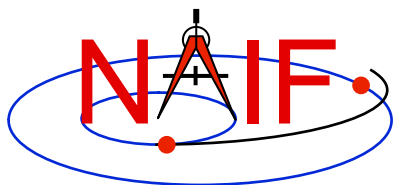


# SPKMERGE - Precedence Rule

---

Navigation and Ancillary Information Facility

- **SPK files created with *spkmerge* have no overlapping ephemeris data. The order in which the source files are specified determines precedence when sources have overlapping coverage for a body of interest.**
  - **IMPORTANT NOTE:** Data from an **earlier** specified source file take precedence over data from a later specified source file when the new (merged) file is created.



# SPKMERGE - Example

Navigation and Ancillary Information Facility

```
Terminal Window
$ cat spkmerge_cas_example.cmd
;This command file directs spkmerge to take data for
;Cassini, the Sun, the Earth, the Moon, and the Earth-
;Moon barycenter and place them into a single SPK.

leapseconds_kernel = naif0007.tls
spk_kernel          = output.bsp
bodies              = -82, 10, 301, 399, 3
source_spk_kernel   = de403s.bsp
source_spk_kernel   = 990825A_SCEPH_EM52_JP0.bsp

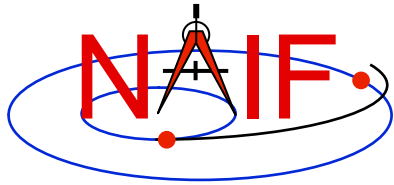
$ spkmerge
SPKMERGE -- SPK Merge Tool, Version 3.2, SPICE Toolkit N0057

Enter the name of the command file

> spkmerge_cas_example.cmd

Creating output.bsp

$
```

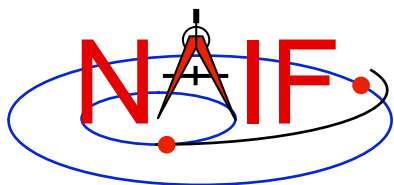


# SPKDIFF

---

Navigation and Ancillary Information Facility

- ***spkdiff*** is a command line program for comparing trajectories provided by two SPK files
- ***spkdiff*** compares SPKs by computing a set of geometric states for a specified body, center and frame over an interval of time with a fixed time step using one SPK file, then computing another set of geometric states for the same or different body, center, and frame at the same times using the other SPK file, and then subtracting the corresponding states from each other
- Depending of the requested output type ***spkdiff*** prints to the screen:
  - only the maximum differences,
  - a complete table of differences, or
  - a statistical analysis of the differences.



# SPKDIFF - Usage

## Navigation and Ancillary Information Facility

```
Terminal Window

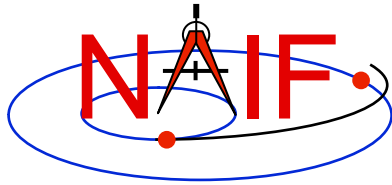
$ spkdiff

spkdiff computes differences between geometric states obtained from
two SPK files and either displays these differences or shows statistics
about them (see the User's Guide for more details.) The program usage is:

% spkdiff [options] <first SPK file> <second SPK file>

Options are shown below. Order and case of keys are not significant.
Values must be space-separated from keys, i.e. '-n 10', not '-n10'.

-b1 <first body name or ID>
-c1 <first center name or ID>
-r1 <first reference frame name>
-b2 <second body name or ID>
-c2 <second center name or ID>
-r2 <second reference frame name>
-k <other kernel file name(s)>
-b <interval start time>
-e <interval stop time>
-s <time step in seconds>
-n <number of states: 2 to 1000000 (default: 1000)>
-f <output time format (default: TDB seconds past J2000)>
-t <report type: basic|stats|dump|dumpvf (default: basic)>
```

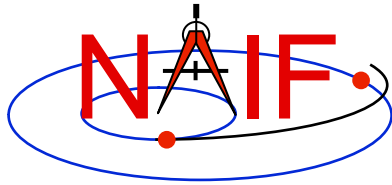


# SPKDIFF – Basic Output Example

Navigation and Ancillary Information Facility

```
Terminal Window
$ spkdiff mro_psp.bsp mro_psp_rec.bsp
# Comparison of 1000 'J2000'-referenced geometric states
#
#   of 'MARS RECON ORBITER' (-74) relative to 'MARS BARYCENTER' (4)
#   from SPK 'mro_psp.bsp'
#
# with 1000 'J2000'-referenced geometric states
#
#   of 'MARS RECON ORBITER' (-74) relative to 'MARS BARYCENTER' (4)
#   from SPK 'mro_psp_rec.bsp'
#
# evenly-spaced with 2617.6524668123 second (0d 0h 43m 37.652467s) step size
# within the time interval
#
#   from '2007 APR 01 00:01:05.185 TDB' (228657665.18565 TDB seconds)
#   to   '2007 MAY 01 06:25:00.000 TDB' (231272700.00000 TDB seconds)
#
Relative differences in state vectors:
                                     maximum           average
Position:                          8.4872836561757E-05  1.2312974450656E-05
Velocity:                           8.5232570159796E-05  1.2314285182022E-05

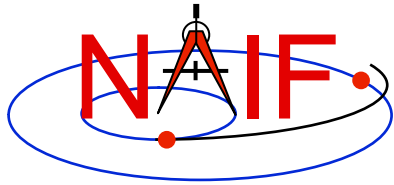
Absolute differences in state vectors:
                                     maximum           average
Position (km):                      3.1341344106404E-01  4.5090516995222E-02
Velocity (km/s):                   2.8848827480682E-04  4.2085874877127E-05
```



# SPKDIFF – Dump Output Example

Navigation and Ancillary Information Facility

```
Terminal Window
$ spkdiff -t dumpvf mro_psp.bsp mro_psp_rec.bsp | more
# Comparison of 1000 'J2000'-referenced geometric states
#
#   of 'MARS RECON ORBITER' (-74) relative to 'MARS BARYCENTER' (4)
#   from SPK 'mro_psp.bsp'
#
# with 1000 'J2000'-referenced geometric states
#
#   of 'MARS RECON ORBITER' (-74) relative to 'MARS BARYCENTER' (4)
#   from SPK 'mro_psp_rec.bsp'
#
# evenly-spaced with 2617.6524668123 second (0d 0h 43m 37.652467s) step size
# within the time interval
#
#   from '2007 APR 01 00:01:05.185 TDB' (228657665.18565 TDB seconds)
#   to   '2007 MAY 01 06:25:00.000 TDB' (231272700.00000 TDB seconds)
#
# time, down_track_p_diff, normal_to_plane_p_diff, in_plane_p_diff, down_track_v
#_diff, normal_to_plane_v_diff, in_plane_v_diff
2.2865766518565E+08 +4.2593079332056E-02 -9.0540866105197E-05 -3.9705894066565E-04 -8.0803561182349E-08
-1.0394439243989E-07 -3.9614350816493E-05
2.2866028283812E+08 +4.2172435702119E-02 +2.3672255851626E-06 -1.1475679619731E-04 +1.3970238250217E-07
+1.4080506259574E-07 -3.9250157214024E-05
2.2866290049059E+08 +4.4830247467488E-02 +9.1590974014175E-05 -7.3802870365833E-04 +5.7800410436763E-07
-1.1724240528272E-07 -4.2099832045985E-05
2.2866551814305E+08 +4.5968515669515E-02 -1.3529652839857E-04 -7.5686845133612E-05 -4.7565892258325E-07
+3.4127364997784E-08 -4.2529268294482E-05
--More--
```



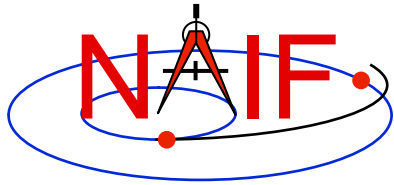
# MSOPCK

---

Navigation and Ancillary Information Facility

- ***msopck* is a program for making CK files from orientation provided in a text file as a time tagged, space-delimited table**
  - has a simple command line interface
  - requires all setups to be provided in a setup file that follows the SPICE text kernel syntax
  - can process quaternions (SPICE and non-SPICE flavors), Euler angles, or matrixes, tagged with UTC, SCLK, or ET
  - for more details see the “Making a CK File” Tutorial

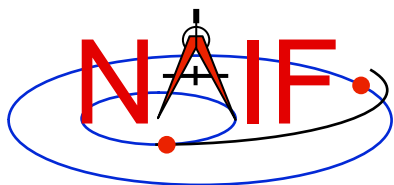




# FRMDIFF

Navigation and Ancillary Information Facility

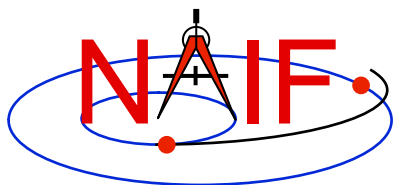
- ***frmdiff* is a command line program for sampling the orientation of a reference frame or for computing the difference between orientations of two reference frames based on provided set(s) of SPICE kernels**
- **In sampling mode, *frmdiff* computes a set of transformations from one frame to another frame over a specified interval with a specified step**
- **In comparison mode, *frmdiff* computes two sets of transformations for two pairs of “from”-“to” frames and then computes the difference in rotation and angular velocity between these transformations over a specified interval with a specified step**
- **Depending on the execution mode and the requested output type *frmdiff* prints to the screen:**
  - only the maximum rotation or the maximum rotation difference,
  - a complete table of rotations or differences (as angle and axis, SPICE- or engineering-style quaternions, matrixes, or Euler angles), or
  - a statistical analysis of rotations or differences.



# FRMDIFF - Usage

## Navigation and Ancillary Information Facility

```
Terminal Window
$ frmdiff
  % frmdiff [options] <first kernel name> <second kernel name>
  % frmdiff [options] <kernel name>
  % frmdiff [options]
where kernel can be a CK, an FK, or a PCK. Options are shown below.
-k <supporting kernel(s) name(s)>
-f1 <first ``from'' frame, name or ID>
-t1 <first ``to'' frame, name or ID>
-c1 <first frame for coverage look up, name or ID>
-k1 <additional supporting kernel(s) for first file>
-f2 <second ``from'' frame, name or ID>
-t2 <second ``to'' frame, name or ID>
-c2 <second frame for coverage look up, name or ID>
-k2 <additional supporting kernel(s) for second file>
-a <compare angular velocities: yes|no>
-m <frame for angular velocities: from|to>
-b <interval start time>
-e <interval stop time>
-n <number of points: 1 to 1000000 (default: 1000)>
-s <time step in seconds>
-f <time format: et|sclk|sclkd|ticks|picture_for_TIMEOUT>
-t <report: basic|stats|dumpaa|dumpm|dumpqs|dumpqo|dumpea|dumpc|dumpg>
-o <rotation axes order (default: z y x)>
-x <units for output angles> (only for -t dumpaa and -t dumpea)
```

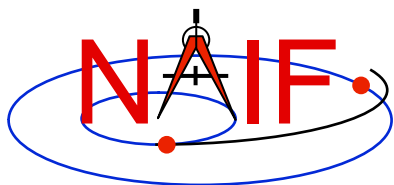


# FRMDIFF – Sampling Example

Navigation and Ancillary Information Facility

```
Terminal Window
$ frmdiff -k naif0009.tls DIF_SCLKSCET.00036.tsc di_v17.tf -s 5 -t dumpqo -f sclkd -a yes -m to
dif_sc_2009-01-27.bc > output.txt

$ cat output.txt
#
# Sampling of 16864 rotations
#
#   from 'J2000' (1) to 'DIF_SPACECRAFT' (-140000)
#   computed using
#
#       naif0009.tls DIF_SCLKSCET.00036.tsc di_v17.tf
#       dif_sc_2009-01-27.bc
#
# with a 5.000000000000000 second (0:00:00:05.000000) step size
# within the non-continuous (with 2 gaps) time period
#
#   from '2009 JAN 27 00:01:06.713' TDB (286286466.71354 TDB seco...
#   to   '2009 JAN 28 00:01:05.346' TDB (286372865.34683 TDB seco...
#
# including angular velocities relative to 'to' frame.
#
# Times are decimal SCLKs computed using SCLK ID -140.
#
# time, q_sin1, q_sin2, q_sin3, q_cos, av_x, av_y, av_z
2.8628543276953E+08 +6.9350853049532E-01 +3.7594179111024E-01 -6.1...
2.8628543776953E+08 +6.9350851552324E-01 +3.7594215798843E-01 -6.1...
```

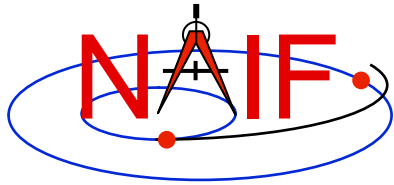


# FRMDIFF – Comparison Example

Navigation and Ancillary Information Facility

```
Terminal Window
$ frmdiff -k naif0009.tls cas00130.tsc cas_v40.tf -s 10 -b 2009-JAN-09 00:00 -e 2009-JAN-10 00:00 -t
dumpaa 09009_09025pa_fsiv_lud2.bc 09006_09011ra.bc > output.txt

$ cat output.txt
#
# Comparison of 3143 rotations
#   from 'J2000' (1) to 'CASSINI_SC_COORD' (-82000)
#   computed using
#     naif0009.tls cas00130.tsc cas_v40.tf
#     09009_09025pa_fsiv_lud2.bc
#
# with 3143 rotations
#   from 'J2000' (1) to 'CASSINI_SC_COORD' (-82000)
#   computed using
#     naif0009.tls cas00130.tsc cas_v40.tf
#     09006_09011ra.bc
#
# with a 10.000000000000 second (0:00:00:10.000000) step size
# within the non-continuous (with 1 gaps) time period
#
#   from '2009 JAN 09 15:17:06.359' TDB (284786226.35996 TDB seco...
#   to   '2009 JAN 10 00:01:06.184' TDB (284817666.18419 TDB seco...
#
# Times are TDB seconds past J2000.
# angle is shown in radians.
#
# time, angle, axis_x, axis_y, axis_z
+2.8478622635996E+08 +5.4958832051797E-05 +8.2101753099566E-01 +4....
+2.8478623635996E+08 +5.4931030131424E-05 +8.2046010733260E-01 +4....
```



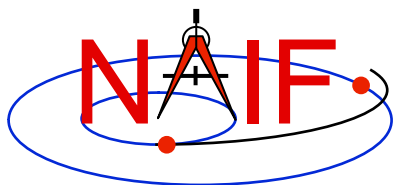
# Kernel Summary Applications

---

Navigation and Ancillary Information Facility

The contents of binary kernels can be summarized with the kernel summary tools.

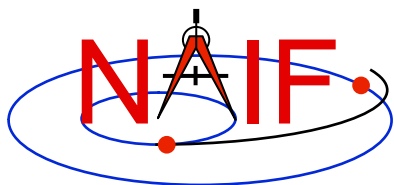
- ***brief*** displays the bodies and associated time coverage in an SPK file or set of SPK files.
  - *brief* also works on binary PCK files
- ***ckbrief*** displays the structure(s) and associated time coverage in a CK file or set of CK files.
- ***spacit*** displays a segment by segment summary of the contents of a CK, SPK, binary PCK, or EK/ESQ file.
  - *spacit* also identifies the SPK or CK data type present in each segment.



# BRIEF

Navigation and Ancillary Information Facility

- ***brief* is a simple command line program for summarizing the contents of SPK or binary PCK files**
- **the files to be summarized can listed on the command line, given in a meta-kernel provided on the command line, or provided in a list file**
- ***brief* provides command line options for**
  - displaying coverage boundaries as date UTC, DOY UTC, or ET seconds past J2000 (default time format is calendar ET)
    - » to display time as UTC an LSK file must be provided on the command line
  - displaying centers of motions along with the bodies
  - treating all input files as if they were a single file
  - displaying summary only for files covering a specified time or time range or containing data for a specified body
  - displaying summary in tabular format or grouped by coverage
  - and many others ...



# BRIEF - Usage

## Navigation and Ancillary Information Facility

```
Terminal Window
$ brief

BRIEF -- Version 3.0.0, January 14, 2008 -- Toolkit Version N0063

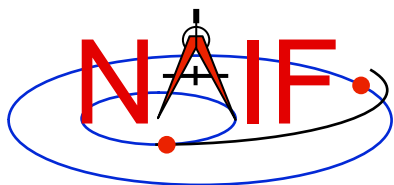
BRIEF is a command-line utility program that displays a summary for
one or more binary SPK or binary PCK files. The program usage is:

% brief [-options] file [file ...]

The most useful options are shown below. For the complete set of
options, run BRIEF with the -h option. The order of options is not
significant. The case of option keys is significant: they must be
lowercase as shown below.

-c          display centers of motion/relative-to frames
-t          display summary in a tabular format
-a          treat all files as a single file
-utc        display times in UTC calendar date format (needs LSK)
-utcdoy     display times in UTC day-of-year format (needs LSK)
-etsec      display times as ET seconds past J2000

An LSK file must be provided on the command line to display times in
UTC formats. FK file(s) must be provided on the command line to
display names of any frames that are not built into the Toolkit.
```



# BRIEF - Example

## Navigation and Ancillary Information Facility

```
Terminal Window
$ brief de405s.bsp m01_cruise.bsp

BRIEF -- Version 3.0.0, January 14, 2008 -- Toolkit Version N0063

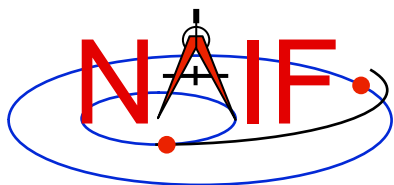
Summary for: de405s.bsp

Bodies: MERCURY BARYCENTER (1)  SATURN BARYCENTER (6)  MERCURY (199)
        VENUS BARYCENTER (2)    URANUS BARYCENTER (7)  VENUS (299)
        EARTH BARYCENTER (3)    NEPTUNE BARYCENTER (8) MOON (301)
        MARS BARYCENTER (4)     PLUTO BARYCENTER (9)  EARTH (399)
        JUPITER BARYCENTER (5)  SUN (10)              MARS (499)
Start of Interval (ET)                End of Interval (ET)
-----
1997 JAN 01 00:01:02.183              2010 JAN 02 00:01:03.183

Summary for: m01_cruise.bsp

Body: MARS SURVEYOR 01 ORBITER (-53)
Start of Interval (ET)                End of Interval (ET)
-----
2001 APR 07 16:25:00.000              2001 OCT 24 05:00:00.000
```

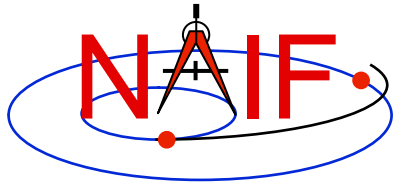




# CKBRIEF

Navigation and Ancillary Information Facility

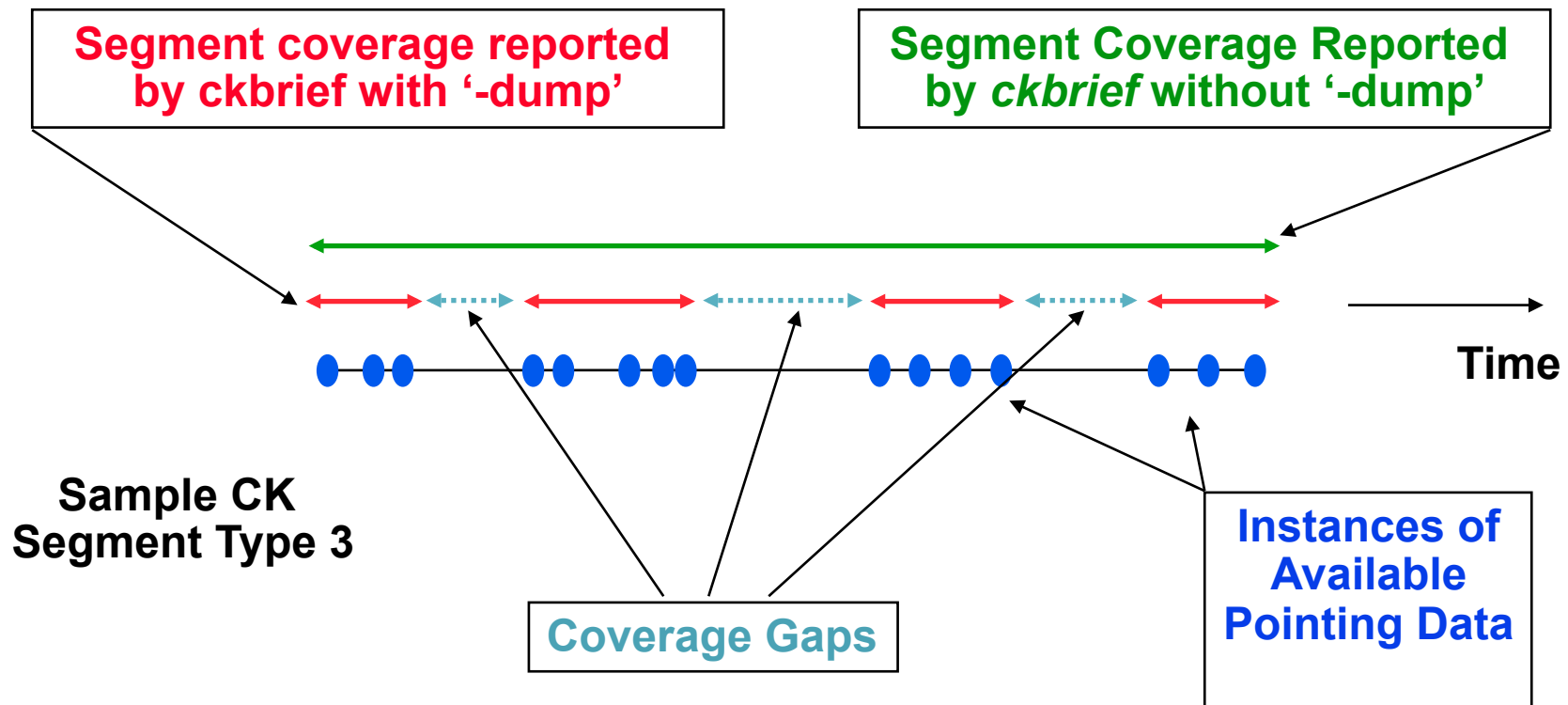
- ***ckbrief* is a simple command line program for summarizing the contents of CK files**
- **the files to be summarized can listed on the command line, given in a meta-kernel provided on the command line, or provided in a list file**
- ***ckbrief* provides command line options for**
  - displaying coverage at interpolation interval level
  - displaying coverage boundaries as date UTC, DOY UTC, SCLK, or encoded SCLK (default time format is calendar ET)
    - » to display times as ET, UTC, or SCLK, an LSK file and SCLK file(s) must be provided on the command line
  - Displaying frames with respect to which orientation is provided
  - Displaying the names of the frames associated with CK IDs
    - » an FK file(s) defining these frames must be provided on the command line
  - displaying summary only for files with data for a given CK ID
  - and many others ...

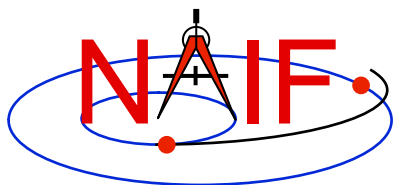


# CKBRIEF – Interval Summary

Navigation and Ancillary Information Facility

- There often are coverage gaps within a CK segment
- Using the ‘-dump’ option allows to get a complete list of continuous coverage intervals for each segment





# CKBRIEF – Usage

## Navigation and Ancillary Information Facility

```
Terminal Window
$ ckbrief
CKBRIEF -- Version 5.0.0, February 11, 2009 -- Toolkit Version N0063

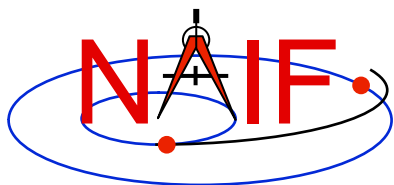
CKBRIEF is a command-line utility program that displays a summary for
one or more binary CK files. The program usage is:

% ckbrief [-options] file [file ...]

The most useful options are shown below. For the complete set of
options, run CKBRIEF with the -h option. The order of options is not
significant. The option keys must be lowercase as shown below.

-dump      display interpolation intervals
-rel       display relative-to frames (may need FK)
-n         display frames associated with CK IDs (may need FK)
-t         display summary in a tabular format
-utc       display times in UTC calendar date format (needs LSK&SCLK)
-utcday    display times in UTC day-of-year format (needs LSK&SCLK)
-sclk      display times as SCLK strings (needs SCLK)

LSK and SCLK files must be provided on the command line to display times
in UTC, ET, or SCLK formats. FK file(s) must be provided on the command
line to display names of any frames that are not built into the Toolkit.
```



# CKBRIEF – Example

## Navigation and Ancillary Information Facility

```
Terminal Window

$ ckbrief -sclk 981116_981228pa.bc sclk.ker

CKBRIEF -- Version 5.0.0, February 11, 2009 -- Toolkit Version N0063

Summary for: 981116_981228pa.bc

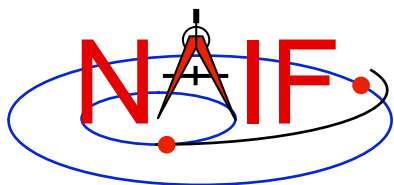
Object: -82000
Interval Begin SCLK          Interval End SCLK          AV
-----
1/1289865849.116           1/1293514473.118          N

$ ckbrief -utc sclk.ker naif0007.tls 990817_990818ra.bc

CKBRIEF -- Version 5.0.0, February 11, 2009 -- Toolkit Version N0063

Summary for: 990817_990818ra.bc

Object: -82000
Interval Begin UTC          Interval End UTC          AV
-----
1999-AUG-17 17:30:01.418 1999-AUG-17 23:05:42.039 N
1999-AUG-17 23:05:45.289 1999-AUG-18 06:06:05.874 N
1999-AUG-18 06:06:09.124 1999-AUG-18 11:52:17.741 N
1999-AUG-18 11:52:20.991 1999-AUG-18 13:30:00.953 N
```



# CKBRIEF - '-dump' Example

Navigation and Ancillary Information Facility

```
Terminal Window
$ ckbrief mgs_spice_c_kernel_2004-099.bc MGS_SCLKSCET.00053.tsc naif0007.tls -dump
-rel -utc

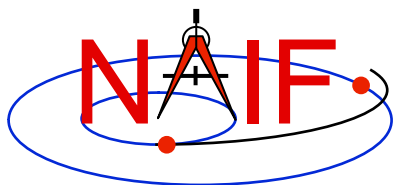
CKBRIEF -- Version 5.0.0, February 11, 2009 -- Toolkit Version N0063

Summary for: mgs_spice_c_kernel_2004-099.bc

Segment No.: 1

Object: -94000

Interval Begin UTC      Interval End UTC      AV  Relative to FRAME
-----
2004-APR-08 00:00:59.809 2004-APR-08 06:53:47.805 Y  J2000
2004-APR-08 06:54:07.805 2004-APR-08 06:54:07.805 Y  J2000
2004-APR-08 06:54:19.805 2004-APR-08 06:54:35.805 Y  J2000
2004-APR-08 06:54:51.805 2004-APR-08 06:54:55.805 Y  J2000
2004-APR-08 06:55:07.805 2004-APR-08 06:55:07.805 Y  J2000
2004-APR-08 06:55:23.805 2004-APR-08 06:55:23.805 Y  J2000
2004-APR-08 06:55:35.805 2004-APR-08 11:59:55.802 Y  J2000
2004-APR-08 12:00:55.802 2004-APR-08 23:59:55.795 Y  J2000
```

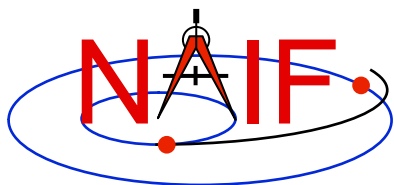


# SPACIT

---

Navigation and Ancillary Information Facility

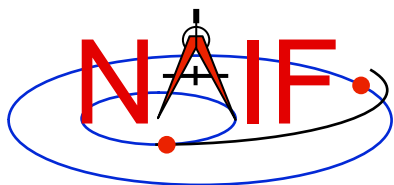
- ***spacit*** may be used to obtain a more detailed summary of an SPK or CK file than that offered by *brief* or *ckbrief*, respectively
  - ***spacit*** may also be used to summarize a binary PCK or an EK/ESQ.
  - ***spacit*** is an interactive program
    - » It will prompt you for all needed inputs
    - » It displays short menus where you choose the action desired
- ***spacit*** may also be used to manage comments, and to convert between binary and transfer format



# Comment Manipulation Tools

Navigation and Ancillary Information Facility

- Every kernel should contain metadata – called “comments” – describing the file contents, intended usage, etc.
- In binary kernels – SPKs, CKs, binary PCKs, and EKs – comments are stored in a special area of the file called the “comment area.”
- *commnt* can read, extract, add, or delete comments stored in the comment area
  - **Caution:** you cannot add or delete comments if the kernel file is not in native format for the machine on which you’re working.
    - » You can convert a non-native binary format file to native binary format by converting the file to “transfer format” using *toxfr* and then converting it back to binary format using *tobin*.
    - » Or use the *bingo* utility (available only from the NAIF website).



# COMMNT

Navigation and Ancillary Information Facility

- ***commnt* is both a command line utility and an interactive menu-driven program**
- **In command line mode, *commnt* provides options to**
  - print comments to the screen  

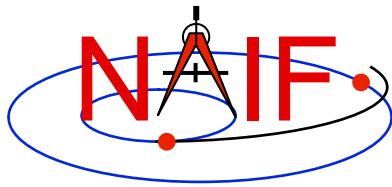
```
$ commnt -r kernel_file
```
  - extract comments to a text file  

```
$ commnt -e kernel_file text_file
```
  - add comments from a text file  

```
$ commnt -a kernel_file comment_file
```
  - delete comments  

```
$ commnt -d kernel_file
```
- **Important to note that**
  - When comments are added, they are appended at the end of the existing comments
  - Comments should be deleted **ONLY** to be replaced with better comments





# COMMNT - Command Line Example

Navigation and Ancillary Information Facility

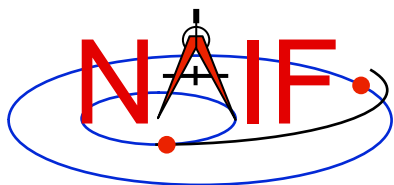
```
Terminal Window
$ commnt -r de405.bsp | more
; de405.bsp LOG FILE
;
; Created 1999-10-03/14:31:58.00.
;
; BEGIN NIOSPK COMMANDS

LEAPSECONDS_FILE      = /kernels/gen/lsk/naif0007.tls
SPK_FILE              = de405.bsp
SOURCE_NIO_FILE       = /usr2/nio/gen/de405.nio
  BODIES               = 1 2 3 4 5 6 7 8 9 10 301 399 199 299 499
  BEGIN_TIME          = CAL-ET 1950 JAN 01 00:00:41.183
  END_TIME            = CAL-ET 2050 JAN 01 00:01:04.183

; END NIOSPK COMMANDS

A memo describing the creation of the DE405 generic planet ephemeris is available from NAIF or from the author: Dr. Myles Standish of JPL's Solar System Dynamics Group. Because this memo was produced using the TeX processor and includes numerous equations

>>> Beginning of extract from Standish's DE405 memo <<
...
```



# COMMNT – Interactive Example

Navigation and Ancillary Information Facility

```
Terminal Window

$ commnt

Welcome to COMMNT Version: 6.0.0
(Spice Toolkit N0050)

COMMNT Options

( Q ) Quit.
( A ) Add comments to a binary file.
( R ) Read the comments in a binary file.
( E ) Extract comments from a binary file.
( D ) Delete the comments in a binary file.

Option: E

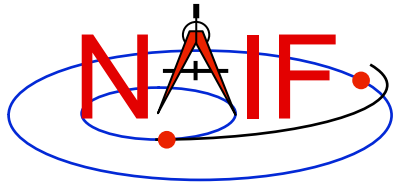
Enter the name of the binary file.

Filename? de405.bsp

Enter the name of the comment file to be created.

Filename? de405_comments.txt

The comments were successfully extracted.
```



# File Format Conversion Tools

---

Navigation and Ancillary Information Facility

- **With modern Toolkits (N0052 and later) the porting of DAF-based binary kernels\* between computers having dissimilar binary standards is usually not necessary.**
  - The advent of binary kernel readers that detect the binary style and do run-time translation if needed generally makes porting unnecessary for DAF-based types.
  - Refer to the “Porting Kernels” tutorial for more on this topic.
- **If true porting is needed (because you must modify or append to a kernel):**
  - use *toxfr* on the source computer and *tobin* on the destination computer
  - or use *bingo* on the destination computer
    - » **NOTE: bingo is NOT available in Toolkits; it must be downloaded from the NAIF website**

\* DAF-based binary kernels are SPK, CK and binary PCK