



Toolkit Applications

October 2007



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*
- Kernel summary tools: *brief*, *ckbrief*, *spacit*
- Comments manipulation tools: *commnt*, *spacit*
- File format converters: *tobin*, *toxfr* and *bingo*



CHRONOS - Time Conversions

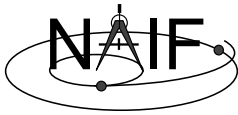
Navigation and Ancillary Information Facility

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:

Supported Time Systems --> Supported Time Types

```
-----  
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

Output System/Type

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

```
-----  
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
```

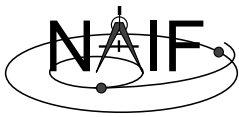
(*) default input/output types



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*** will eventually be offered as a subroutine as well.



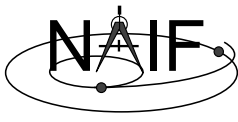
CHRONOS - Usage

Navigation and Ancillary Information Facility

```
$ 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 <central body ID>]
  [-LANDINGTIME <UTC time of the landing>]
  [-SOL1INDEX <index of the first SOL>]
  [-NOLABEL]
  [-TRACE]
```



CHRONOS - Example

Navigation and Ancillary Information Facility

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

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

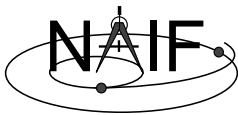
$ chronos -setup chronos.cas -from utc -to sclk -time 1999 JAN 12 12:00
1/1294833883.185                                        (SCLK/SCLK)

$ 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
```

Toolkit Applications

7



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.

Toolkit Applications

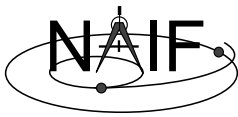
8



SPKMERGE

Navigation and Ancillary Information Facility

- The contents of an SPK file or set of SPK files may be merged or subsetting 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

```
$ 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

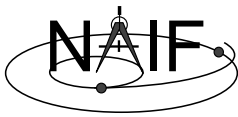
> spkmerge_cas_example.cmd

Creating output.bsp

$
```

Toolkit Applications

11



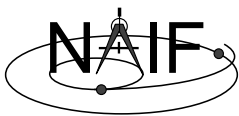
SPKDIFF

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 simple statistical analysis of the differences.

Toolkit Applications

12



SPKDIFF - Usage

Navigation and Ancillary Information Facility

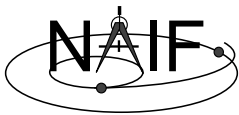
```
$ spkdiff

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 - Example 1

Navigation and Ancillary Information Facility

```
$ 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
```



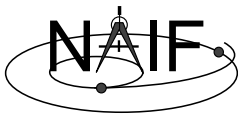
SPKDIFF - Example 2

Navigation and Ancillary Information Facility

```
$ 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--
```

Toolkit Applications

15



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

Toolkit Applications

16

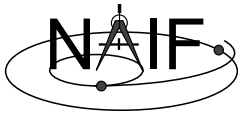


Kernel Summary

Navigation and Ancillary Information Facility

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

- Use *brief* to find the bodies and associated time coverage contained in an SPK file or set of SPK files.
- Use *ckbrief* to find the structure(s) and associated time coverage(s) stored in a C-Kernel or set of C-Kernel files.
- Use *spacit* for a segment by segment summary of the contents of a CK, SPK, binary PCK or EK/ESQ.
 - The program *spacit* also identifies the SPK or CK data type for each segment.



Using *brief* - 1

Navigation and Ancillary Information Facility

The utility *brief* provides a simple command line interface to summarize the contents of SPK files.

- Use *brief* to quickly determine what bodies are present in an SPK file or set of SPKs.

```
$ brief spk.bsp or $ brief *.bsp
```
- Find all the bodies in an SPK file for which there is coverage at an epoch or during an interval of time.

```
$ brief -at '7/25/99' spk.bsp or  
$ brief -from '7/25/99' -to '8/1/99' spk.bsp
```
- Find out which SPK files contain coverage for a particular body.

```
$ brief -CASSINI spk.bsp
```

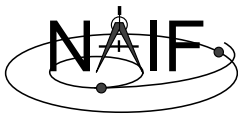
(Note: *brief* will only display body *names* that are hard-coded within SPICE library software because BRIEF can't load a text kernel containing a new name-to-ID mapping)



Using *brief* - 2

Navigation and Ancillary Information Facility

- To show the central body around which each target moves, use the '-c' option.
- If you want to treat a group of SPK files as an ephemeris set, rather than individual files, use the '-a' option.
- To display results in a tabular format use the '-t' option.
- Options may be provided in any order on the command line.
- The times displayed by *brief* are given in calendar format ET (TDB), which is not the same as UTC
 - ET currently differs from UTC by approximately 65 seconds.



Summarizing a Set of SPKs

Navigation and Ancillary Information Facility

```
$ brief *.bsp
Brief. Version: 2.1          (SPICE Toolkit N0050)

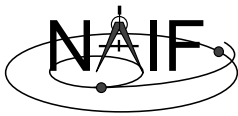
Summary for: 990825A_SCEPH_EM52_JP0.bsp

Body: CAS (-82)
  Start of Interval (ET)      End of Interval (ET)
  -----
1999 JUN 26 14:00:00.000      2000 DEC 30 11:00:00.000

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

Note: The wild character "*" does not work as part of a command
line input in the PC/Windows environment.
```



Using *ckbrief* - 1

Navigation and Ancillary Information Facility

***ckbrief* is a simple command line program for summarizing the contents of CK files.**

- Use *ckbrief* with just file names to quickly list the structures whose pointing is stored in one or more C-kernels.

```
$ ckbrief ck.bc or $ ckbrief *.bc
```

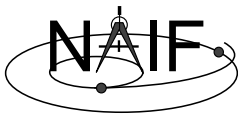
- If you want to view coverage using SCLK times you need to also provide an SCLK kernel and specify the '-sclk' option.

```
$ ckbrief sclk.ker ck.bc -sclk
```

- If you want to view coverage using UTC times you need to provide both SCLK and LSK kernels and specify the '-utc' option.

```
$ ckbrief sclk.ker naif0007.tls ck.bc -utc
```

- The names of the kernel files and the options used may be provided in any order on the command line.



Using *ckbrief* - 2

Navigation and Ancillary Information Facility

- Use the NAIF instrument ID to find out which files in a set of C-kernels contain pointing for the instrument of interest.

```
$ ckbrief -82000 *.bc
```

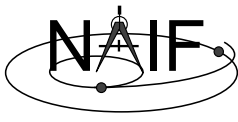
- Use the '-rel' option to find out to which frames the pointing is relative.

```
$ ckbrief -rel *.bc
```

- Frame kernel(s) may also be provided on the command line to make *ckbrief* display frame names rather than IDs in the 'relative to' column.

- A number of other options are also available

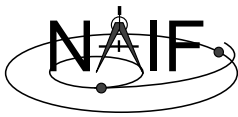
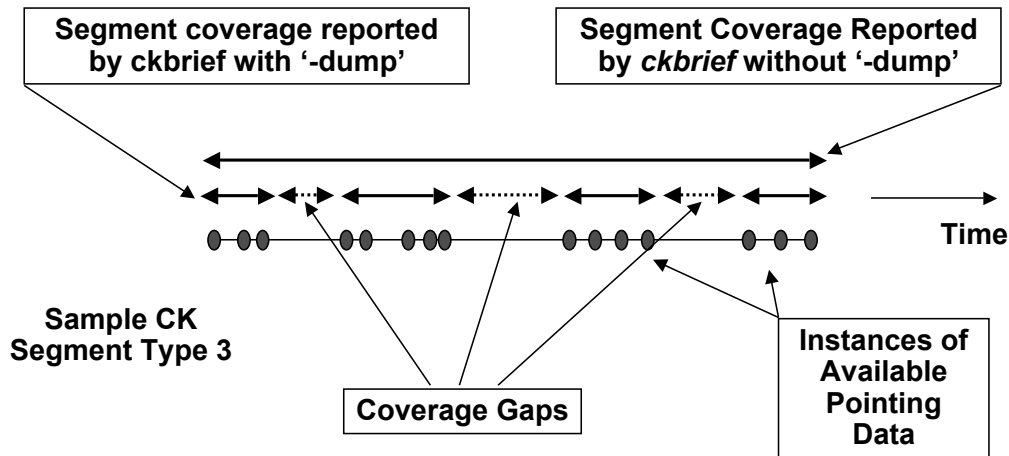
- Type *ckbrief* with no arguments to see a help display.



Using *ckbrief* - 3

Navigation and Ancillary Information Facility

- There often are coverage gaps within a CK segment
- To get a complete list of continuous coverage intervals for each segment, use the '-dump' option.



Summarizing CKs with UTC and SCLK

Navigation and Ancillary Information Facility

```
$ ckbrief -sclk 981116_981228pa.bc sclk.ker

CKBRIEF Version: 1.0.0, 1999-09-09. SPICE Toolkit Version: N0050.

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: 1.0.0, 1999-09-09. SPICE Toolkit Version: N0050.

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
```



Summarizing CKs with '-dump'

Navigation and Ancillary Information Facility

```
$ ckbriefer mgs_spice_c_kernel_2004-099.bc MGS_SCLKSCET.00053.tsc naif0007.tls -dump
-rel -utc

CKBRIEF Version: 2.0.0, 2001-05-16. SPICE Toolkit Version: N0057.

Summary for: mgs_spice_c_kernel_2004-099.bc

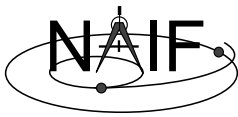
Segment No.: 1

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
```

Toolkit Applications

25



Using *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**

Toolkit Applications

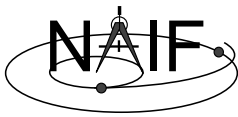
26



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.”
- To read, extract, add or delete comments stored in the comment area, use the Toolkit utility *commnt*.
 - 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).



Using *commnt* - 1

Navigation and Ancillary Information Facility

- *commnt* is both a command line utility and an interactive menu-driven program.
- To dump the comments from a binary kernel to the terminal using the command line use the ‘-r’ option:

```
$ commnt -r kernel_file
```
- To extract the comments from a binary kernel to a text file use the ‘-e’ option:

```
$ commnt -e kernel_file text_file
```
- To add or append new comments to a binary kernel, place them in a text file and use the ‘-a’ option:

```
$ commnt -a kernel_file comment_file
```

 - These are appended to any previously added comments



Using *commnt* - 2

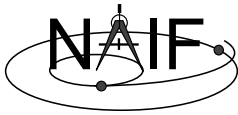
Navigation and Ancillary Information Facility

- To delete comments from a binary kernel use the '-d' option:

```
$ commnt -d kernel_file
```

- Comments should only be deleted when they are to be replaced with updated information.
- All of the command line features of *commnt* are supported by the interactive invocation.
- To run *commnt* interactively, type the utility with no command line arguments:

```
$ commnt
```



Using *commnt* Interactively

Navigation and Ancillary Information Facility

```
$ 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.
```



Reading Comments with *commnt*

Navigation and Ancillary Information Facility

```
$ 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 avail
able from NAIF or from the author: Dr. Myles Standish of JPL's Solar System Dy
namics Group. Because this memo was produced using the TeX processor and inclu
des numerous equations

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

-More-- (19%)
```

Toolkit Applications

31



File Format Conversion

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

Toolkit Applications

32