Non-Toolkit Applications

March 2006

Summary

- NAIF developed and maintains a set of additional applications not included in the generic Toolkit. This set includes programs for:
  - making and modifying SPK files
    - » *pinpoint, bspidmod*
  - making and modifying CK files
    - » *msopck, prediCkt, ckslicer, ckspanit, dafcat, cksmrg*
  - computing derived quantities
    - » *orbnum, optics*
  - determining SPICE kernel type
    - » *archtype*
- Executables for these programs for a few selected environments are available from
  - ftp://naif.jpl.nasa.gov/pub/naif/utilities

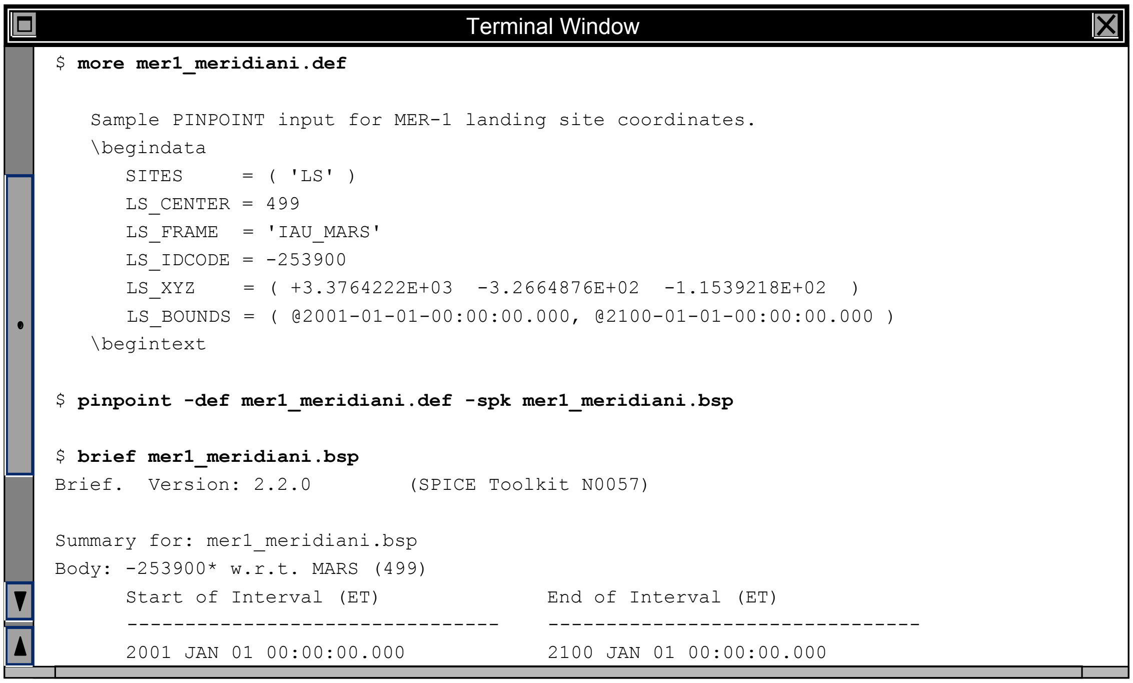
PINPOINT

- *pinpoint* is a program for creating SPK files for objects whose position is a constant offset with respect to another object
  - » Ground stations
  - » Landing sites, sites along a rover path
  - » Relative positions of manipulator joints, etc.
- *pinpoint* is a command line program with the following usage:

```
pinpoint -def deffile -spk spkfile [-pck tkfile]
```

- “deffile” is an input definitions file following text kernel file format and containing a set of keywords defining ID, center, reference frame, position (as XYZ or Gaussian Lat/Lon/R) and time coverage boundaries for one or more objects
- “spkfile” is an output SPK file containing a type 8 SPK segment for each of the objects specified in the “deffile”
- “tkfile” is optional PCK file (needed if posititons in the “deffile” are given as Lat/Lon/R) or FK file (needed if one or more of the frames specified in “deffile” is not one of the frames built into the Toolkit)
- The contents of “deffile” is included in the comment area

PINPOINT Example



```
$ more mer1_meridiani.def

   Sample PINPOINT input for MER-1 landing site coordinates.
   \begindata
      SITES     = ( 'LS' )
      LS_CENTER = 499
      LS_FRAME  = 'IAU_MARS'
      LS_IDCODE = -253900
      LS_XYZ    = ( +3.3764222E+03  -3.2664876E+02  -1.1539218E+02  )
      LS_BOUNDS = ( @2001-01-01-00:00:00.000, @2100-01-01-00:00:00.000 )
   \begintext

$ pinpoint -def mer1_meridiani.def -spk mer1_meridiani.bsp

$ brief mer1_meridiani.bsp
Brief.  Version: 2.2.0        (SPICE Toolkit N0057)

Summary for: mer1_meridiani.bsp
Body: -253900* w.r.t. MARS (499)
      Start of Interval (ET)              End of Interval (ET)
      --------------------------------    --------------------------------
      2001 JAN 01 00:00:00.000            2100 JAN 01 00:00:00.000
```

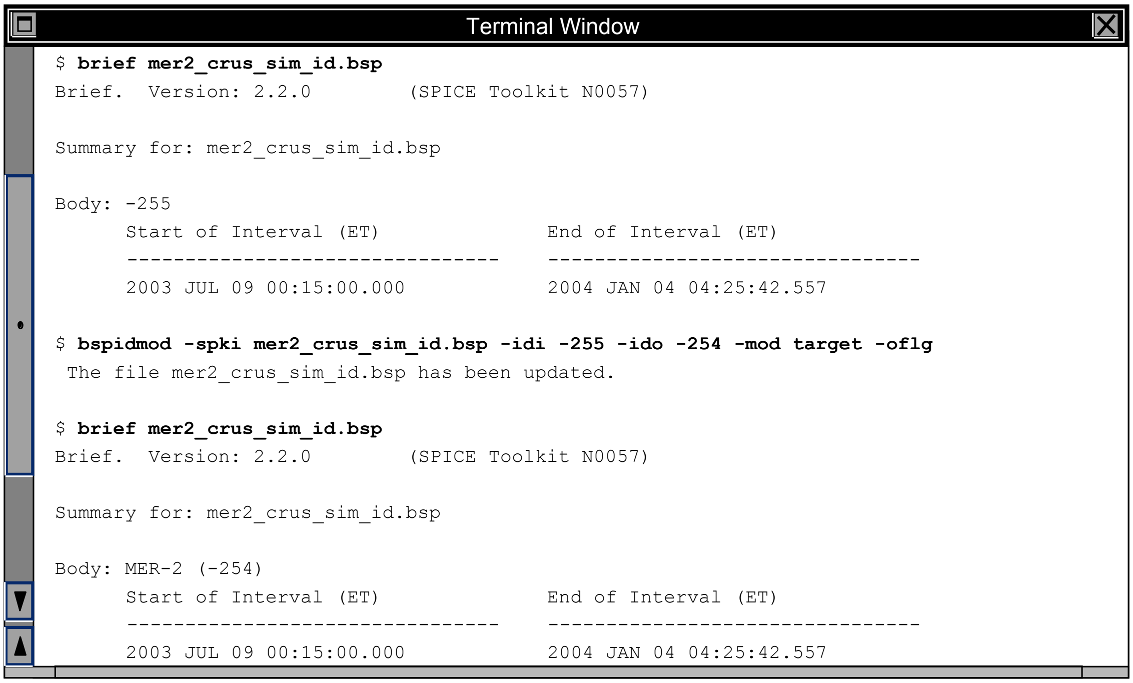
BSPIDMOD

- *bspidmod* is a program for altering the object IDs in a binary SPK file
  - » can be used to modify IDs in an SPK file(s) produced with “bogus” spacecraft ID (or SIM spacecraft ID)
  - » can be used to replace “good” IDs with “bogus” IDs if two different trajectories for the same object need to be used in the same program at the same time (for example for comparison)
- *bspidmod* is a command line program with the following usage:

```
bspidmod -spki inpspk -idi inpid -ido outid -mod item -oflg
```

- “inpspk” is input SPK; “inpid” and “outid” are current ID and new ID
- “item” indicates which IDs are to be replaced: TARGET -- only target IDs are replaced, CENTER -- only center IDs are replaced, or OBJECT -- both target and center IDs are replaced
  - » Replacements are made only when “inpid” matches an ID found in the input SPK
- “-oflg” flag indicating that change should be made directly in the input file; if not specified, the program produces output file with name that has “_out” appended to the name of the input file
- A note stating which IDs were modified is put in the comment area

BSPIDMOD Example



```
$ brief mer2_crus_sim_id.bsp
Brief.  Version: 2.2.0        (SPICE Toolkit N0057)

Summary for: mer2_crus_sim_id.bsp

Body: -255
      Start of Interval (ET)              End of Interval (ET)
      --------------------------------    --------------------------------
      2003 JUL 09 00:15:00.000            2004 JAN 04 04:25:42.557

$ bspidmod -spki mer2_crus_sim_id.bsp -idi -255 -ido -254 -mod target -oflg
 The file mer2_crus_sim_id.bsp has been updated.

$ brief mer2_crus_sim_id.bsp
Brief.  Version: 2.2.0        (SPICE Toolkit N0057)

Summary for: mer2_crus_sim_id.bsp

Body: MER-2 (-254)
      Start of Interval (ET)              End of Interval (ET)
      --------------------------------    --------------------------------
      2003 JUL 09 00:15:00.000            2004 JAN 04 04:25:42.557
```

MSOPCK and PREDICKT

- *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 or SCLK
  - » for more details see “Making a CK File” Tutorial
- *prediCkt* is a program for making CK files from a set of orientation specification rules, and schedules defining when these rules are to be applied
  - » has a simple command line interface
  - » requires orientation and schedule specification to be provided in a setup file that follows the SPICE text kernel syntax
  - » requires all supporting kernels -- SPK, PCK, etc -- to be provided in a meta kernel
  - » for more details see “Making CK Tutorial”

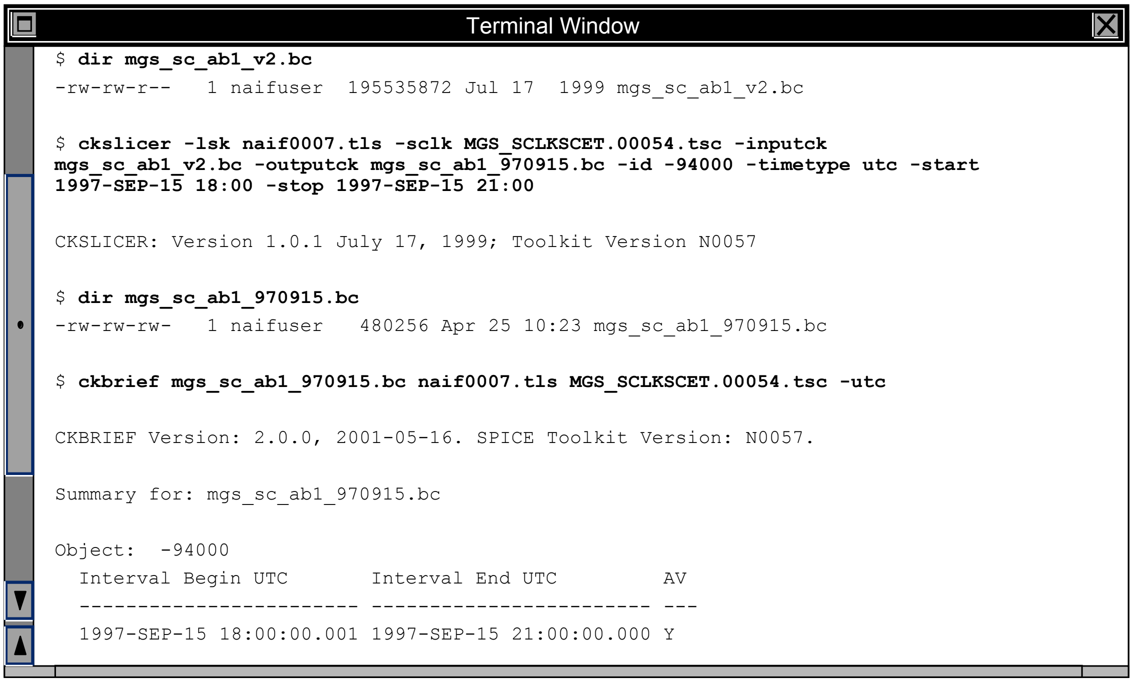
CKSLICER

- *ckslicer* is a program for subsetting a CK file
- *ckslicer* is a command line program with the following usage

```
ckslicer -lsk <lsk_file_name>
         -sclk <sclk_file_name(s)>
         -inputck <ck_file_name>
         -outputck <ck_file_name>
         -id <naif_id>
         -timetype <utc|sclk|ticks>
         -start <start_time>
         -stop <stop_time>
```

- *ckslicer* is useful in the situation when only a portion of a CK covering a short interval of time is needed (for example when the whole CK takes a lot of space) or to cut parts from a few CKs with the intent to merge them together (if reconstructed CKs from different sources have too much overlap to simply “cat” them together)
- A note stating which subset was extracted is put into the comment area of the output CK file

CKSLICER Example



```
$ dir mgs_sc_ab1_v2.bc
-rw-rw-r--   1 naifuser  195535872 Jul 17  1999 mgs_sc_ab1_v2.bc

$ ckslicer -lsk naif0007.tls -sclk MGS_SCLKSCET.00054.tsc -inputck
mgs_sc_ab1_v2.bc -outputck mgs_sc_ab1_970915.bc -id -94000 -timetype utc -start
1997-SEP-15 18:00 -stop 1997-SEP-15 21:00

CKSLICER: Version 1.0.1 July 17, 1999; Toolkit Version N0057

$ dir mgs_sc_ab1_970915.bc
-rw-rw-rw-   1 naifuser   480256 Apr 25 10:23 mgs_sc_ab1_970915.bc

$ ckbrief mgs_sc_ab1_970915.bc naif0007.tls MGS_SCLKSCET.00054.tsc -utc

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

Summary for: mgs_sc_ab1_970915.bc

Object:  -94000
  Interval Begin UTC       Interval End UTC         AV
  ------------------------ ------------------------ ---
  1997-SEP-15 18:00:00.001 1997-SEP-15 21:00:00.000 Y
```

CKSPANIT

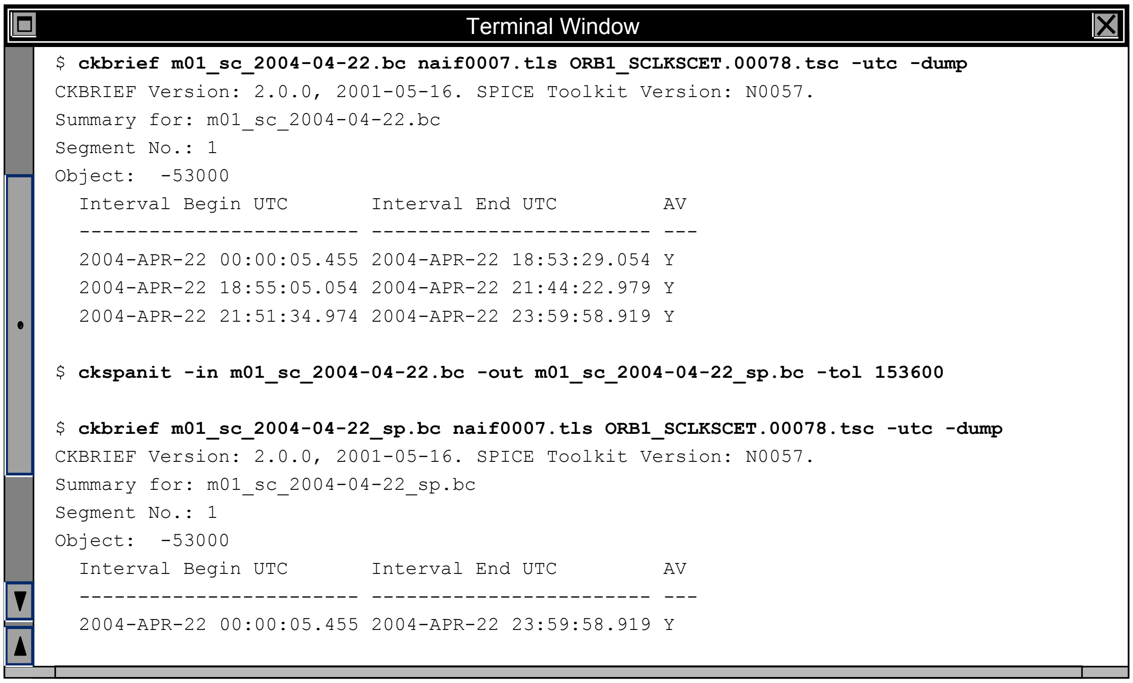
- *ckspanit* is a program for modifying interpolation interval information in type 3 CK segments
  - » it can also convert a type 1 CK to a type 3 CK
- *ckspanit* is used when one is dealing with a type 3 CK containing many small gaps within segments. It allows you to alter the CK in such a way that SPICE will interpolate over those gaps
- *ckspanit* is a command line program with the following usage

```
ckspanit -in inp_ck -out out_ck -tol threshold [-frm fk]
```

- “threshold” is the longest time interval over which interpolation is to be permitted in the output CK file
  - » must be specified in SCLK ticks
    - For example if 1 tick is 1/256 of a second and interpolation over 30 second intervals is needed, “threshold” must be set to 256*30=7680
- “fk” is optional FK file name, needed only if the base frame in the input CK is not one of the frames built into the Toolkit

CAUTION: before running *ckspanit*, make sure that interpolation over larger gaps is appropriate for the CK you are dealing with. You should add appropriate comments to the new CK file.

CKSPANIT Example



```
$ ckbrief m01_sc_2004-04-22.bc naif0007.tls ORB1_SCLKSCET.00078.tsc -utc -dump
CKBRIEF Version: 2.0.0, 2001-05-16. SPICE Toolkit Version: N0057.
Summary for: m01_sc_2004-04-22.bc
Segment No.: 1
Object:  -53000
  Interval Begin UTC       Interval End UTC         AV
  ------------------------ ------------------------ ---
  2004-APR-22 00:00:05.455 2004-APR-22 18:53:29.054 Y
  2004-APR-22 18:55:05.054 2004-APR-22 21:44:22.979 Y
  2004-APR-22 21:51:34.974 2004-APR-22 23:59:58.919 Y

$ ckspanit -in m01_sc_2004-04-22.bc -out m01_sc_2004-04-22_sp.bc -tol 153600

$ ckbrief m01_sc_2004-04-22_sp.bc naif0007.tls ORB1_SCLKSCET.00078.tsc -utc -dump
CKBRIEF Version: 2.0.0, 2001-05-16. SPICE Toolkit Version: N0057.
Summary for: m01_sc_2004-04-22_sp.bc
Segment No.: 1
Object:  -53000
  Interval Begin UTC       Interval End UTC         AV
  ------------------------ ------------------------ ---
  2004-APR-22 00:00:05.455 2004-APR-22 23:59:58.919 Y
```

DAFCAT

– *dafcat* is a program for concatenating binary DAF files by simply copying all data segments from all input files, in the order they are provided, into the output file

   » Works on SPKs, CKs, and binary PCKs

      • will not merge different types of kernels together, i.e. will not merge SPKs with CKs, CKs with PCKs, etc.

      • should not be used for merging SPKs because *spkmerge* provides a much more powerful and sophisticated capability

– *dafcat* is a command line program with the following usage

```
dafcat output_file
```

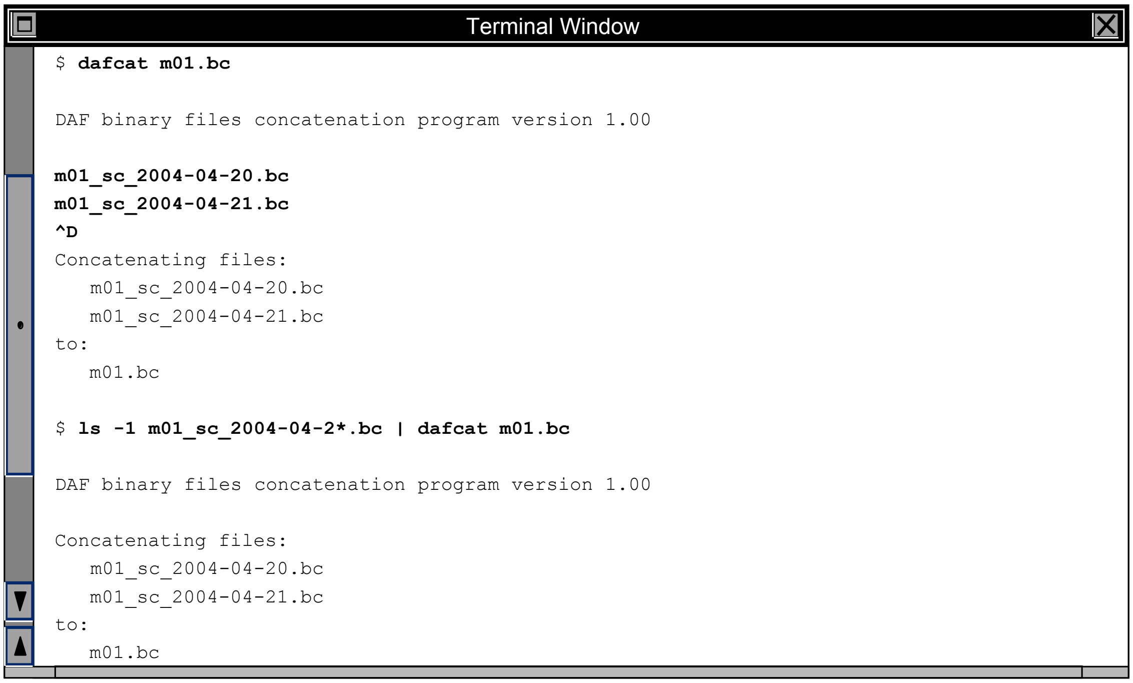
– “output_file” is the output file name and is the program’s only argument

– Input file names should be provided from standard input

   » this is very convenient for use with Unix shell pipes

– *dafcat* does not put any information into the comment area of the output file, leaving this responsibility to the user (use *commnt* to do so)

DAFCAT Example



```
$ dafcat m01.bc

DAF binary files concatenation program version 1.00

m01_sc_2004-04-20.bc
m01_sc_2004-04-21.bc
^D
Concatenating files:
   m01_sc_2004-04-20.bc
   m01_sc_2004-04-21.bc
to:
   m01.bc

$ ls -1 m01_sc_2004-04-2*.bc | dafcat m01.bc

DAF binary files concatenation program version 1.00

Concatenating files:
   m01_sc_2004-04-20.bc
   m01_sc_2004-04-21.bc
to:
   m01.bc
```

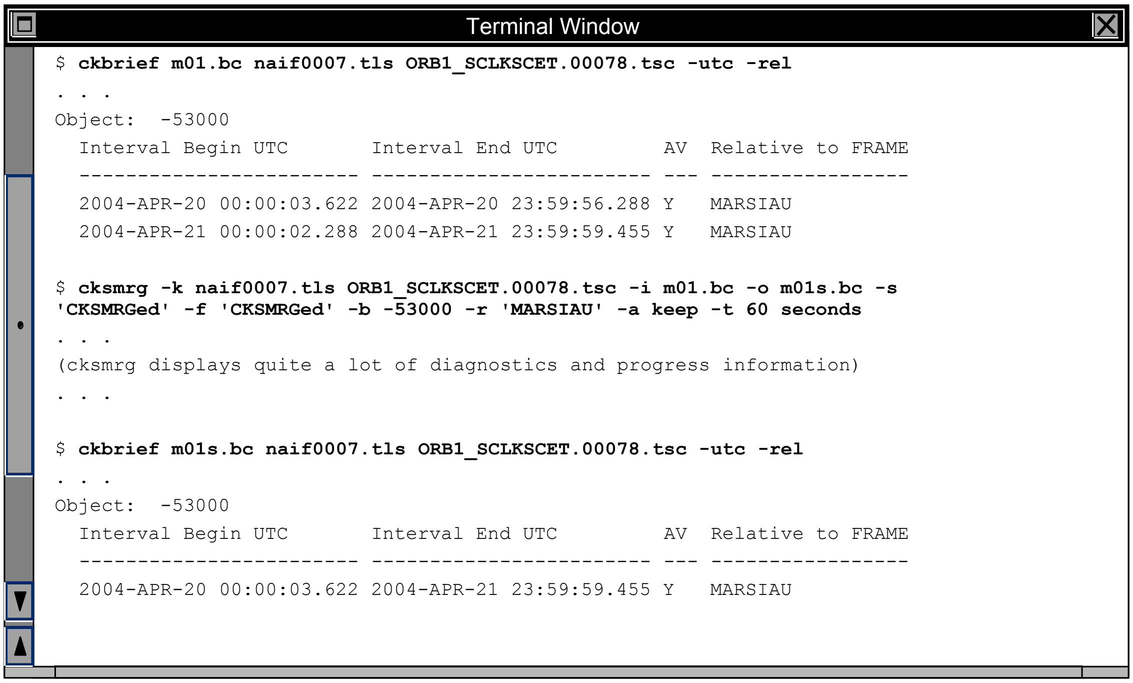
CKSMRG

- *cksmrg* is a program for merging data from two or more uniform CK segments (same ID, base frame and type) provided in a single CK file
- *cksmrg* is used for eliminating gaps between segments (that cannot be removed by *ckspanit*) and removing duplicate data points contained in different segments
- *cksmrg* is a command line program with the following usage

```
cksmrg -k|-kernels <meta kernel name|kernel file names>
       -i|-input <input ck file name>
       -o|-output <output ck file name>
       -s|-segid <output ck segment id string>
       -f|-fileid <output ck file id string>
       -b|-body <body id|name>
       -r|-reference <reference id|name>
       -a|-av <drop|keep|make|makeavrg>
       -t|-tolerance <tolerance (number units)>
       [-c|-correction <time delta|cor. table file>]
```

CAUTION: cksmrg should not be used to merge CKs from different sources, nor should it be used to merge overlapping predict CKs

CKSMRG Example



```
$ ckbrief m01.bc naif0007.tls ORB1_SCLKSCET.00078.tsc -utc -rel
. . .
Object:  -53000
  Interval Begin UTC       Interval End UTC         AV  Relative to FRAME
  ------------------------ ------------------------ --- -----------------
  2004-APR-20 00:00:03.622 2004-APR-20 23:59:56.288 Y   MARSIAU
  2004-APR-21 00:00:02.288 2004-APR-21 23:59:59.455 Y   MARSIAU

$ cksmrg -k naif0007.tls ORB1_SCLKSCET.00078.tsc -i m01.bc -o m01s.bc -s
'CKSMRGed' -f 'CKSMRGed' -b -53000 -r 'MARSIAU' -a keep -t 60 seconds
. . .
(cksmrg displays quite a lot of diagnostics and progress information)
. . .

$ ckbrief m01s.bc naif0007.tls ORB1_SCLKSCET.00078.tsc -utc -rel
. . .
Object:  -53000
  Interval Begin UTC       Interval End UTC         AV  Relative to FRAME
  ------------------------ ------------------------ --- -----------------
  2004-APR-20 00:00:03.622 2004-APR-21 23:59:59.455 Y   MARSIAU
```

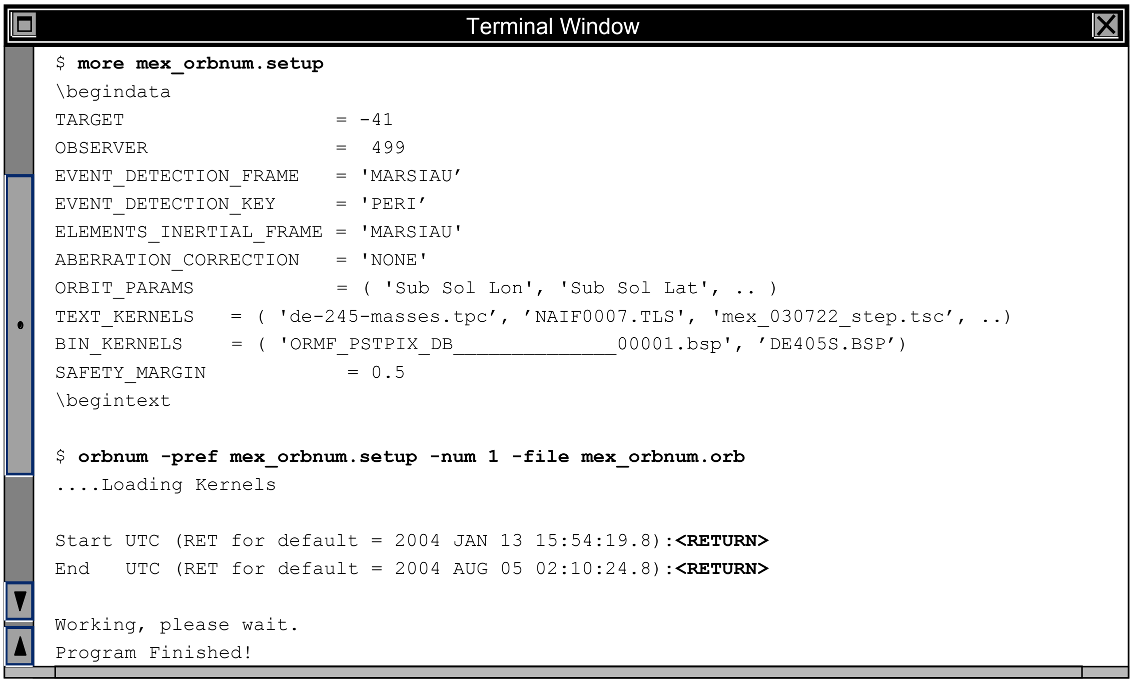
ORBNUM

- *orbnum* is a program for generating a so-called SPICE orbit number file containing orbit start/stop times and orbit numbers along with some additional derived quantities (orbital elements and coordinates of sub-spacecraft and sub-solar points)
  - » The orbit number increment can be specified as occuring at one of these events: periapsis, apoapsis, ascending equatorial node crossing, or descending equatorial node crossing
- *orbnum* is a command line program with the following usage

```
orbnum -pref pref_file -num init_orbit -file orbnum_file
```

- “pref_file” is a preferences file using text kernel syntax, specifying setup parameters along with the kernels containing data to be used to search for orbit start and stop events -- spacecraft trajectory SPKs, center body PCK, spacecraft SCLK, etc.
- “init_orbit” is the number to be assigned to the first orbit determined using the kernels provided; subsequent orbits are assigned by incrementing “init_orbit” by 1
- If any of the command line arguments and some of the setup file parameters have not been provided, the program will prompt for them
- The program will also prompt for additional information such as the time span within which the search for orbit events is to be performed

ORBNUM Example



```
$ more mex_orbnum.setup
\begindata
TARGET                  = -41
OBSERVER                =  499
EVENT_DETECTION_FRAME   = 'MARSIAU’
EVENT_DETECTION_KEY     = 'PERI’
ELEMENTS_INERTIAL_FRAME = 'MARSIAU'
ABERRATION_CORRECTION   = 'NONE'
ORBIT_PARAMS            = ( 'Sub Sol Lon', 'Sub Sol Lat', .. )
TEXT_KERNELS   = ( 'de-245-masses.tpc’, ’NAIF0007.TLS', 'mex_030722_step.tsc’, ..)
BIN_KERNELS    = ( 'ORMF_PSTPIX_DB______________00001.bsp', ’DE405S.BSP’)
SAFETY_MARGIN            = 0.5
\begintext

$ orbnum -pref mex_orbnum.setup -num 1 -file mex_orbnum.orb
....Loading Kernels

Start UTC (RET for default = 2004 JAN 13 15:54:19.8):<RETURN>
End   UTC (RET for default = 2004 AUG 05 02:10:24.8):<RETURN>

Working, please wait.
Program Finished!
```

OPTIKS

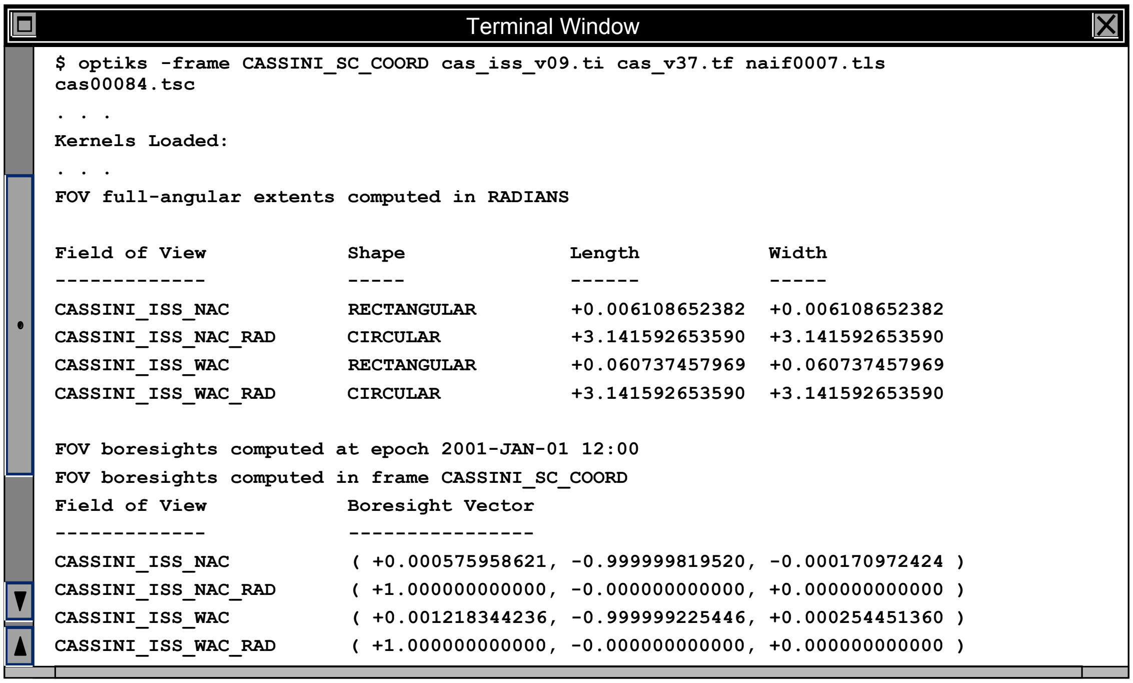
- ***optiks*** **is a utility program that generates information about instrument fields of view (FOV) from parameters present in IK and FK files**
  - **» FOVs must be defined using the keywords required by the GETFOV routine**
- ***optiks*** **is a command line program in one of two ways**

```
optiks [options]... kernel ...
optiks [options]... meta-kernel ...
```

- ***optiks*** **uses a set of SPICE kernels specified on the command line; one or more of these kernels may be a meta-kernel**
- **The output data are organized in two tables**
  - **» The first table lists the angular extents (size) of circular, elliptical, and rectangular FOVs. Using command line options “-units” and “-half” the user can select the unit of measure for the angular measurements, and whether half or full FOV angular extents are listed.**
  - **» The second table contains FOV boresights in a user specified frame at a particular epoch, specified using the “-epoch” option**

OPTIKS Example

Navigation and Ancillary Information Facility



```
$ optiks -frame CASSINI_SC_COORD cas_iss_v09.ti cas_v37.tf naif0007.tls
cas00084.tsc
. . .
Kernels Loaded:
. . .
FOV full-angular extents computed in RADIANS

Field of View            Shape              Length           Width
-------------            -----              ------           -----
CASSINI_ISS_NAC          RECTANGULAR        +0.006108652382  +0.006108652382
CASSINI_ISS_NAC_RAD      CIRCULAR           +3.141592653590  +3.141592653590
CASSINI_ISS_WAC          RECTANGULAR        +0.060737457969  +0.060737457969
CASSINI_ISS_WAC_RAD      CIRCULAR           +3.141592653590  +3.141592653590

FOV boresights computed at epoch 2001-JAN-01 12:00
FOV boresights computed in frame CASSINI_SC_COORD
Field of View            Boresight Vector
-------------            ----------------
CASSINI_ISS_NAC          ( +0.000575958621, -0.999999819520, -0.000170972424 )
CASSINI_ISS_NAC_RAD      ( +1.000000000000, -0.000000000000, +0.000000000000 )
CASSINI_ISS_WAC          ( +0.001218344236, -0.999999225446, +0.000254451360 )
CASSINI_ISS_WAC_RAD      ( +1.000000000000, -0.000000000000, +0.000000000000 )
```

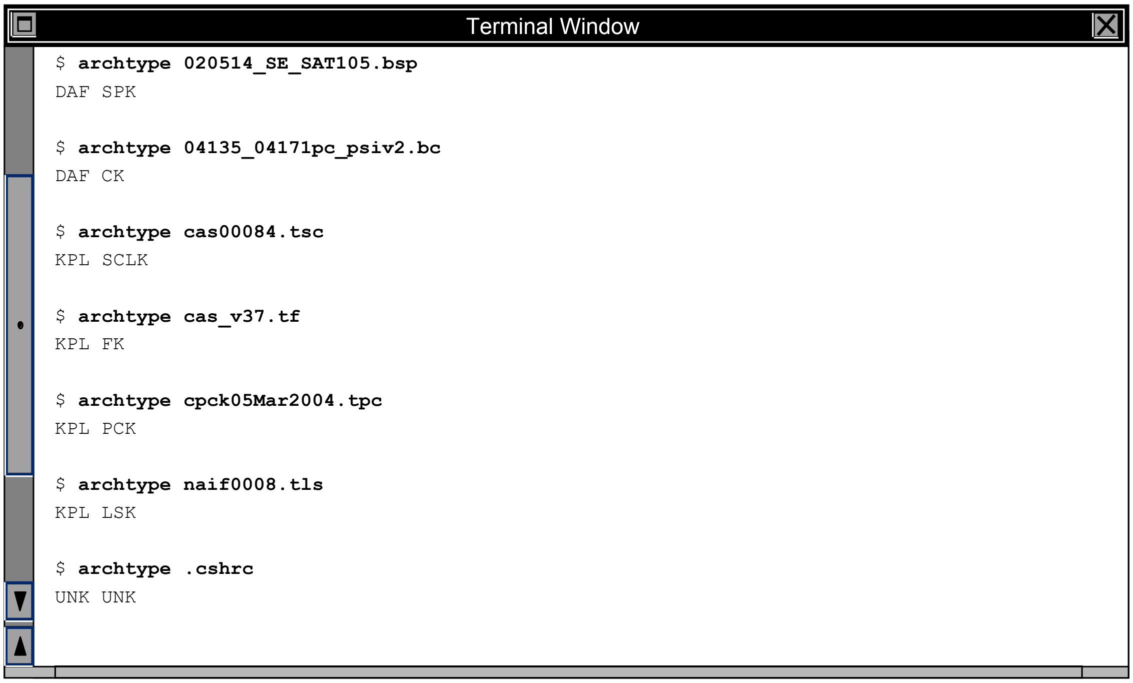
ARCHTYPE

Navigation and Ancillary Information Facility

- *archtype* is a program that displays the file architecture and type of a SPICE kernel; it is useful for scripting applications
  - » To identify the architecture and type *archtype* uses the same mechanism as the FURNSH routine
- *archtype* has a simple command line interface and requires only one argument -- the name of a kernel file:

  ```
  archtype kernel_name
  ```
- Archtype prints architecture and type to standard output as two space delimited acronyms
  - » Architecture can be:
    - 'DAF' or 'DAS' for binary kernels
    - 'KPL' for text kernels
  - » Type can be 'SPK', 'PCK', 'CK', 'EK', 'LSK', 'IK', 'FK'
- If architecture and/or type cannot be determined, the program displays 'UNK'
- In order for text kernels to be recognized, the first few characters of the file must contain 'KPL/<type>' (i.e. 'KPL/IK', 'KPL/FK', etc.)

ARCHTYPE Example



```
$ archtype 020514_SE_SAT105.bsp
DAF SPK

$ archtype 04135_04171pc_psiv2.bc
DAF CK

$ archtype cas00084.tsc
KPL SCLK

$ archtype cas_v37.tf
KPL FK

$ archtype cpck05Mar2004.tpc
KPL PCK

$ archtype naif0008.tls
KPL LSK

$ archtype .cshrc
UNK UNK
```