Navigation Ancillary Information Facility

SPK: The SPICE Ephemeris Subsystem

A Tutorial

April 1998

Prologue

This document provides a brief tutorial for the SPICE ephemeris subsystem–the so-called SPK files and the SPK subroutine family found in the NAIF Toolkit SPICELIB library.

Caution: the examples used are not necessarily appropriate solutions to any particular application, and not all relevant application design issues are discussed. See the last page of this tutorial for a listing of references for further information.

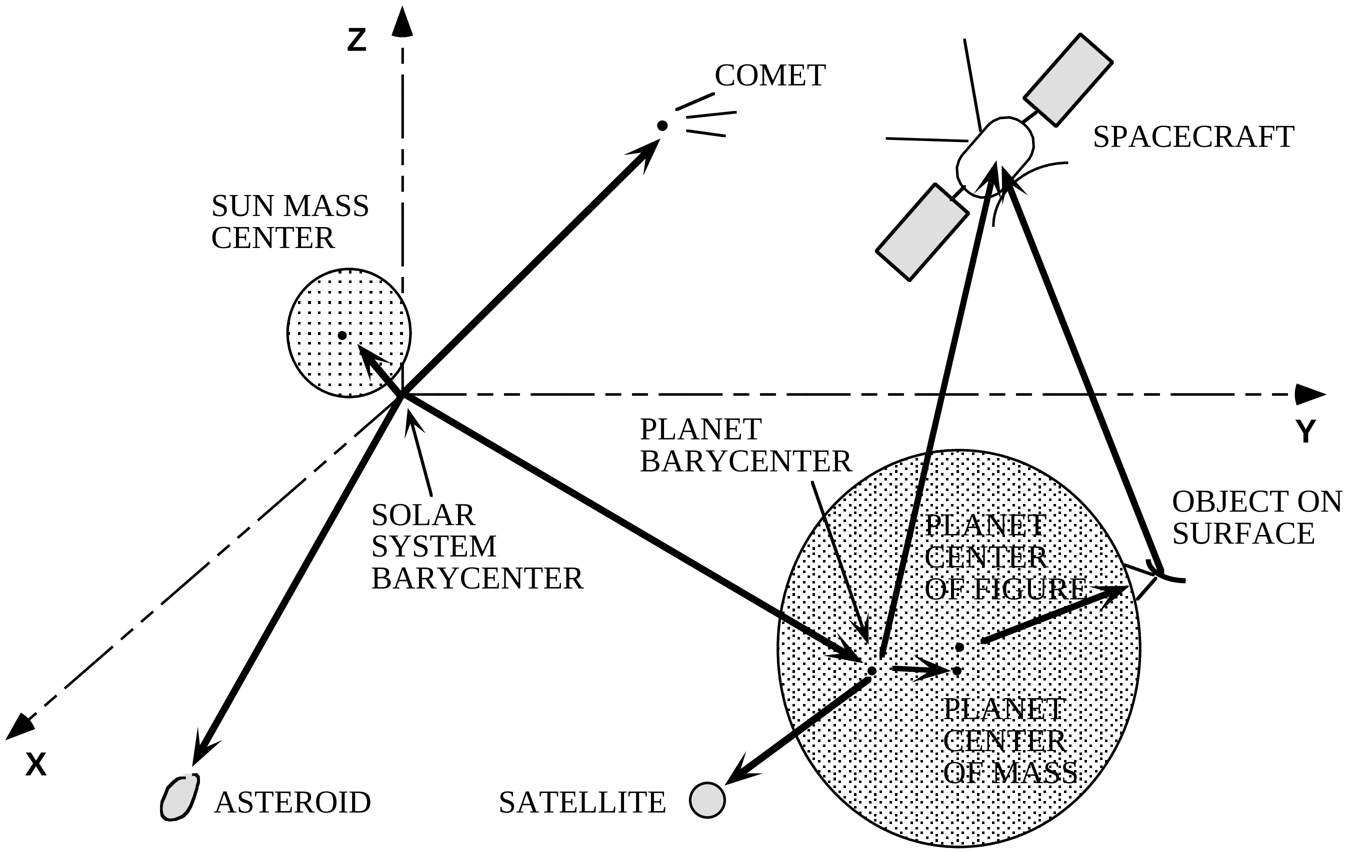
SPK File Contents

- **An SPK file holds ephemeris data for any number/types of solar system objects**
  - **“Ephemeris data” ⇒ position and velocity of one object relative to another**
  - **“Solar system object” ⇒ any spacecraft, planet (mass center or barycenter), satellite, comet or asteroid. Also, the sun and the solar system barycenter. Can also be a designated point on an object (e.g. a DSN station).**

- **A single SPK file can hold data for just one, or for any combination of objects**
  - **Examples:**
    - » **Cassini spacecraft**
    - » **MGS orbiter, M98 lander, Mars, Phobos and the sun**

SPICE Ephemeris Objects

Navigation Ancillary Information Facility



Examples of Generic SPK Files Obtained from NAIF

DExxx
Planetary Ephemeris

O	S.S. BC
1	Merc. BC
199	Merc.
2	Venus BC
299	Venus
3	Earth BC
301	Moon
399	Earth
4	Mars BC
499	Mars
5	Jupiter BC
6	Saturn BC
7	Uranus BC
8	Neptune BC
9	Pluto BC
10	Sun

Asteroid Ephemeris

2000253	Mathilde
2000433	Eros

URAxxx
Satellite Ephemeris

706	Cordelia
707	Ophelia
708	Bianca
•	
•	
•	
715	Puck
799	Uranus

JUPxxx + DE zzz
Merged Ephemeris

3	Earth BC
399	Earth
5	Jupiter BC
501	Io
502	Europa
503	Ganymede
504	Callisto
599	Jupiter
10	Sun

SPK Files Information Access

- User's need know very little about file format and content
  - NAIF provides well documented subroutines to read and evaluate the data in an SPK file
    - » Several access methods (subroutines) are provided
- NAIF software works on any computer that supports ANSI FORTRAN 77 or C
  - Examples: PC with Lahey or M.S. Powerstation compiler, Sun using OS or Solaris, HP, DEC Alpha, DEC VAX/VMS, Silicon Graphics, NeXT, Mac
  - NAIF subroutines can be called from C++
    - » Consult your computer’s language reference manuals

SPK Files: Data Types and Structure

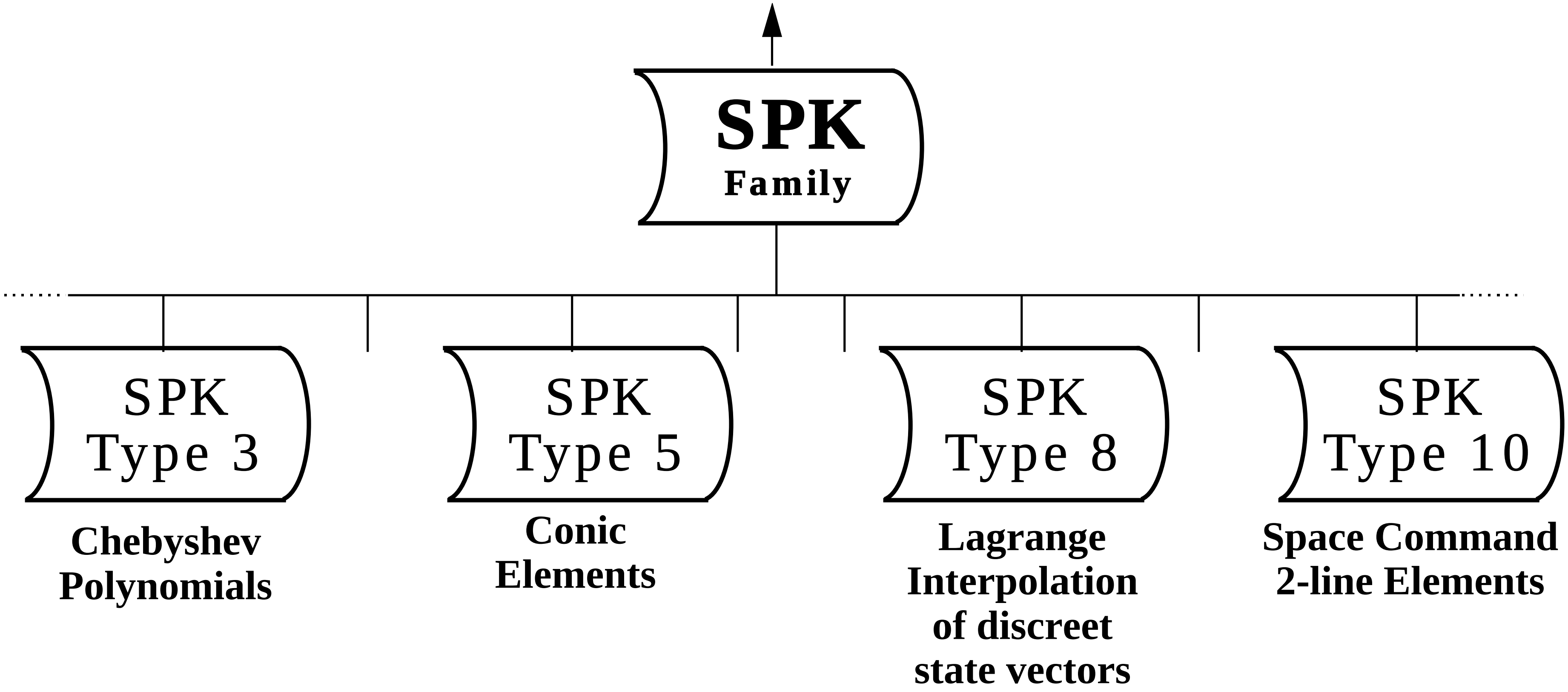
- **SPK File Data Types**
  - Many possibilities (now have ~12 different SPK data types)
    - » Examples: Chebyshev polynomials, sets of conic elements, Lagrange interpolation of discrete state vectors, analytic theories
  - A given SPK file can contain one or multiple SPK data types

- **SPK File Structure**
  - “Double Precision Array File” (DAF) architecture, designed by NAIF
    - » Uses direct access and data buffering
    - » Users need know nothing about this structure: user interface is a subroutine argument list

Kernel Data Type Concept

Underlying data are of varying types, but there is a single user view (interface) to each of these:

CALL SPKEZR ('target', time, 'frame', 'correction', 'observer', state, lite_time)



SPK
Family

SPK Type 3

Chebyshev Polynomials

SPK Type 5

Conic Elements

SPK Type 8

Lagrange Interpolation of discreet state vectors

SPK Type 10

Space Command 2-line Elements

Manipulating and Using SPK Files

- You can subset an SPK file
  - Based on objects, or time, or both
- You can merge two or more SPK files
  - Based on objects, or time, or both
- You can port SPK files between any two computers
- You can provide a label inside an SPK file
  - It's optional; it's free format; it's easily viewed/processed
  - Label can say when/why/how and for what purpose the file was made
- You can read data from just one, or many* SPK files in your application program

(* The allowed number of simultaneously open files depends on both system and NAIF limitations. “Out-of-the-box” NAIF settings restrict this to 20 DAF-based files at most, but this can be increased.)

Getting Positions of Objects - 1

- To get states (position and velocity) of "objects" (spacecraft, planets*, satellites, sun, comets, asteroids) one normally needs two SPICE file types
  - – Ephemeris file (SPK)
  - – Leapseconds file (LSK)
    - » Used to convert between UTC time and Ephemeris Time (ET)

* Position of planet mass center or planet system barycenter. To get planet mass center, one usually also needs a satellite ephemeris for that planet.

Getting Positions of Objects - 2

- Topocentric capability in SPICE
  - One can make an SPK file containing positions of tracking stations, observatories, roving vehicles, etc.
  - Uses non-inertial frames capability recently added to SPICE in version N0042 (January 1996)
    - Requires use of a SPICE PcK file to rotate vectors to an inertial frame such as J2000
      - Can use a “low precision” IAU model of the Earth– a tri-axial ellipsoid
      - Can use a high precision Earth model (“EOP”)
  - User reads this file as any other SPK file
    - Use antenna or observatory SPICE ID as the “target” or “observer” in an SPK reader argument list

Barycenters and Mass Centers

- Numbers in parenthesis are SPICE ID codes
- Planets and their satellites orbit the planet system barycenters
  - For example, Jupiter (599) orbits the Jupiter barycenter (5)
- Planet systems and the sun orbit the solar system barycenter (S.S.BC)

		Mass Offset	
Body Mass Center	Barycenter	from Barycenter*	Offset / Radius
Sun (10)	S. S. BC (0)	1,295,728 km	1.9
Mercury (199)	M. BC (1)	0	--
Venus (299)	V. BC (2)	0	--
Earth (399)	E. BC (3)	4874	0.8
Mars (499)	M. BC (4)	~0	~0
Jupiter (599)	J. BC (5)	165	0.002
Saturn (699)	S. BC (6)	296	0.005
Uranus (799)	U. BC (7)	17	0.0006
Neptune (899)	N. BC (8)	74	0.003
Pluto (999)	P. BC (9)	1500	1.25

* At epoch 1997 JAN 01

How To Read An SPK File

Initialization...typically once per program

- Tell your program which SPICE files to use (“loading” files)
  - CALL SPKLEF ('spk_file_name', handle) *
  - CALL LDPOOL ('leapseconds_file_name')

Loop... do as often as you need

- Convert UTC time to ephemeris time (TDB), if needed
  - CALL STR2ET ('utc_string', tdb)
- Get state vector from SPK file at requested time
  - CALL SPKEZ (target, tdb, 'ref_frame', 'correction', observer, state, lightime)

inputs

outputs

*** Repeat as needed: call SPKLEF once for each SPK file to be used**

Arguments of the SPKEZ Routine - 1

- TARGET* and OBSERVER*: Character names or SPICE IDs for the end point and origin of the state vector (Cartesian position and velocity vectors) to be returned. Planets, satellites, comets and asteroids all have positive IDs; spacecraft have negative IDs. See the document “NAIF_IDs Required Reading” for a listing of many of these IDs.
- TDB: The time at which the state vector is to be computed. The time system used is Ephemeris Time (ET), now generally called Time Barycentric Dynamical (TDB). The units are number of seconds past the year 2000.0. SPICE software is available to compute ET based on UTC (SCET) or Spacecraft Clock (SCLK) time.
- Ref_frame: The SPICE name for the reference frame in which your output state vector is to be given. Often ‘J2000’, but other choices–both inertial and non-inertial–are available. See the documents “NAIF_IDs Required Reading” and “Frames R.R.” for details. SPK software will automatically convert data to the frame you specify (if needed).

* Character names work for the target and observer inputs only if registered in subroutine ZZBODTRN or in a Frames Kernel. Otherwise use the SPICE numeric ID in quotes, as if it were a character string.

Arguments of the SPKEZ Routine - 2

- Correction: Specification of what kind af aberration correction(s) to apply in computing the output state vector. Choices are none, planetary (light time), planetary and stellar, or converged newtonian (a more accurate version of planetary aberration correction). See the header for subroutine SPKEZR, or the document SPK Required Reading, for details.
- State: This is the Cartesian state vector you requested, including corrections. Contains 6 components: three for position (x,y,z) and three for velocity (dx, dy, dz) of the target with respect to the observer.
- Light time: The one-way light time from the observer to the geometric position of the target at the specified epoch. This is *not* affected by the Correction selection.

Example of Reading SPK Files

initialization - do this just once

```
CALL LDPOOL  ('CAS0004.TLS')
CALL SPKLEF  ('HUY_27A2.BSP', H1)
CALL SPKLEF  ('CAS_STANDARD_EPH.BSP', H2)
```

repeat at different times as needed to solve your particular problem

```
CALL STR2ET  ('2004 SEP 12 09:45:21.3', TDB)
CALL SPKEZR  (‘TITAN’, TDB, 'J2000', 'LT+S',
              ‘CASSINI HUYGENS PROBE’, STATE, LT)
(Can use other SPICE code here to make derived computations such as
  spacecraft sub-latitude and longitude, lighting angles, etc.)
```

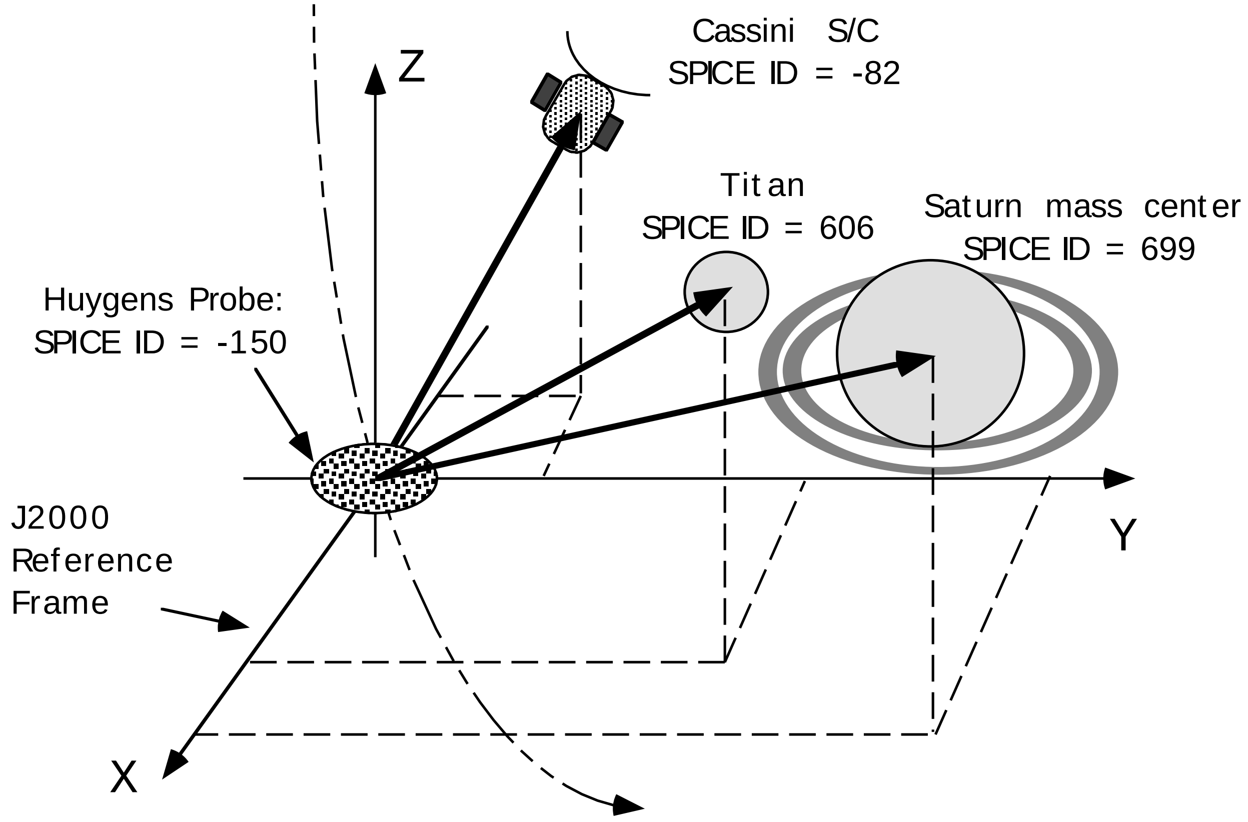
In this example we get the state (STATE, cartesian position and velocity) of Titan as seen from the Cassini Huygens Probe at the epoch 2004 SEP 12 09:45:21.3. The state vector is given in the J2000 inertial reference frame and has been corrected for both light time and stellar aberration (LT+S). The one-way light time (LT) is also returned.

A SPICE leapseconds file (CAS0004.TLS) is used, as are two SPICE ephemeris files:

- **HUY_27A2.BSP, containing ephemeris for the Huygens Probe**
- **CAS_STANDARD_EPH.BSP, containing ephemeris for Saturn, Saturn's satellites and the sun**

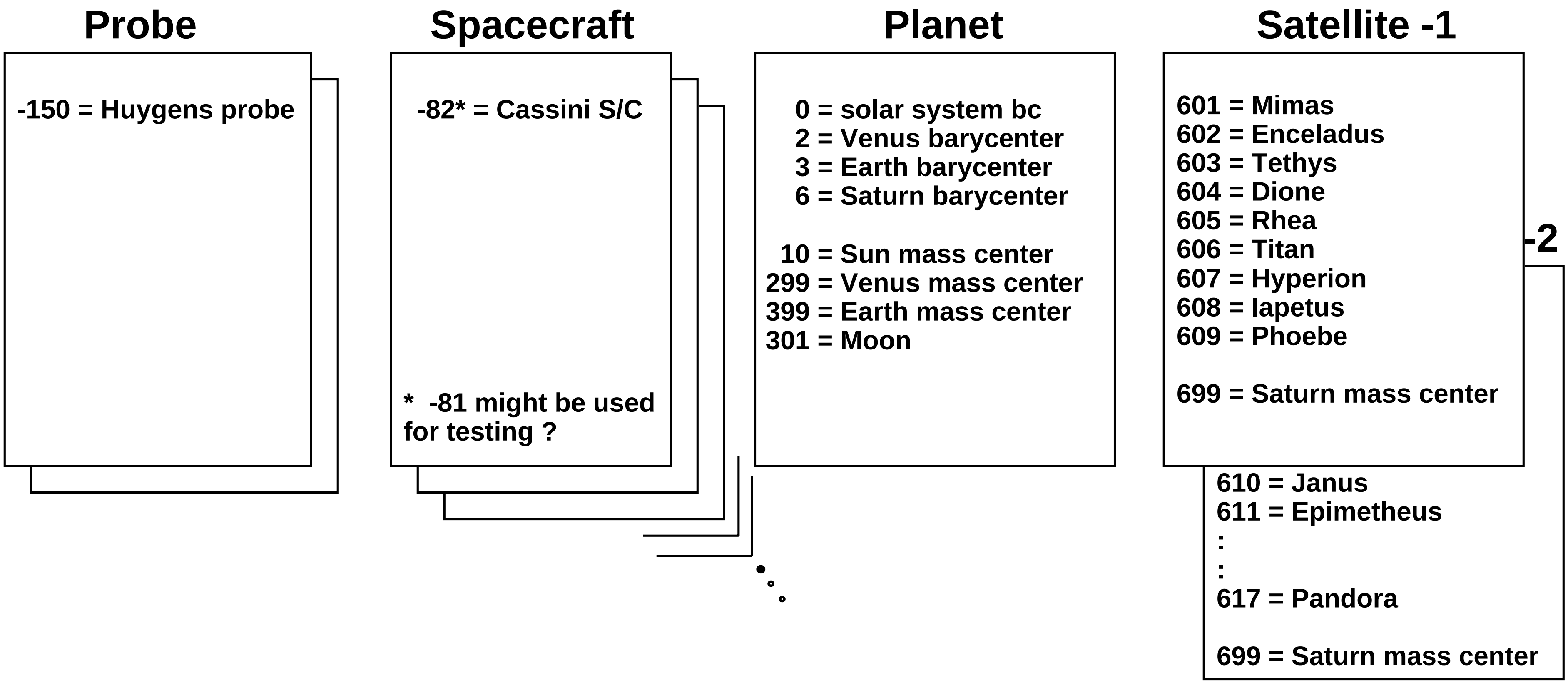
Note: all file names used in these examples are arbitrary choices; your project will specify its own file naming conventions

SPK Output: A Pictorial Example



Output: state (position and velocity) of Titan or Saturn or Cassini relative to the Huygens probe, given in the selected reference frame (e.g. J2000), at the time requested.

A Sample Set of SPK Files



Probe

-150 = Huygens probe

Spacecraft

-82* = Cassini S/C

* -81 might be used for testing ?

Planet

0 = solar system bc
2 = Venus barycenter
3 = Earth barycenter
6 = Saturn barycenter

10 = Sun mass center
299 = Venus mass center
399 = Earth mass center
301 = Moon

Satellite -1

601 = Mimas
602 = Enceladus
603 = Tethys
604 = Dione
605 = Rhea
606 = Titan
607 = Hyperion
608 = Iapetus
609 = Phoebe

699 = Saturn mass center

-2

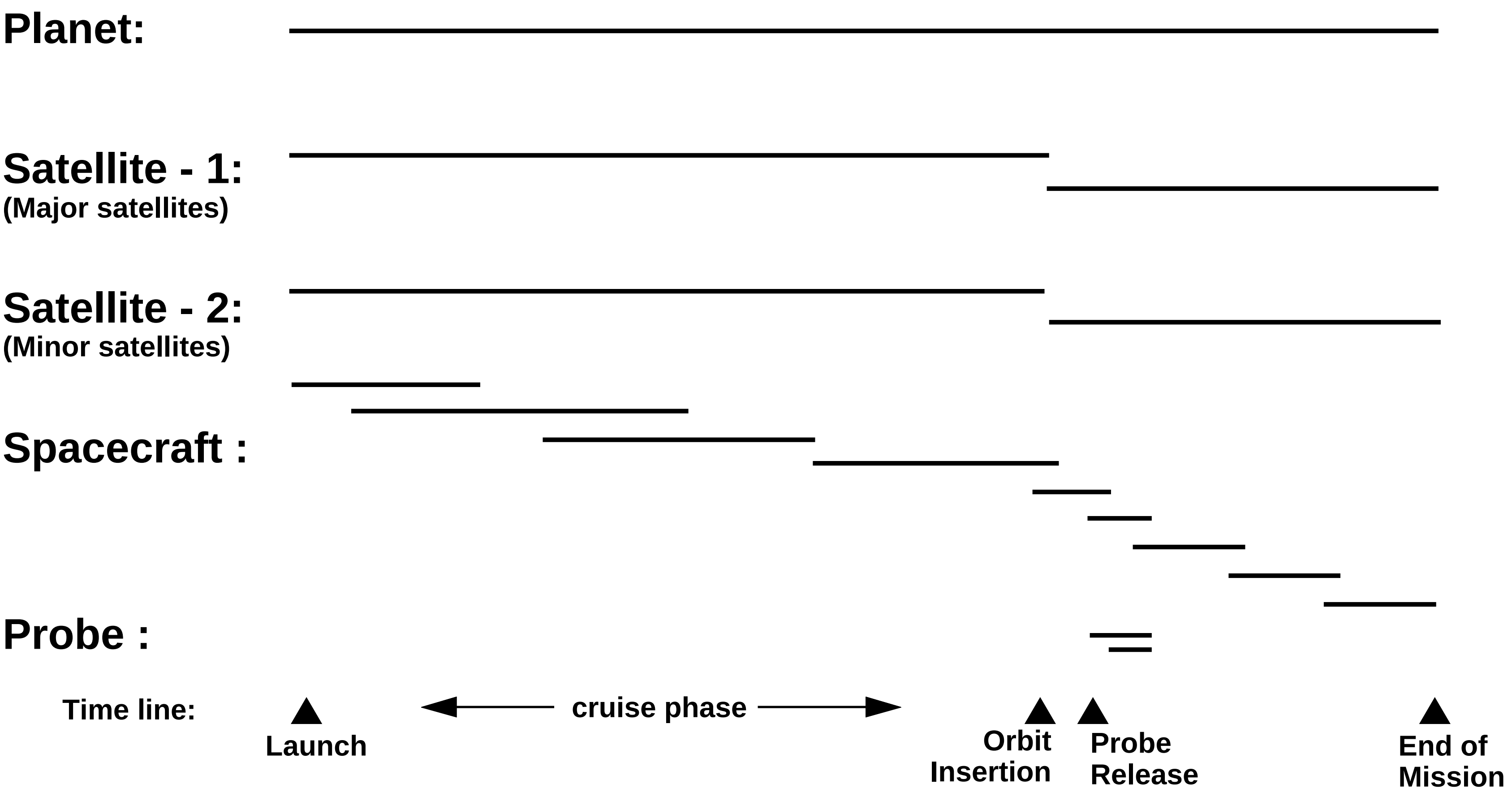
610 = Janus
611 = Epimetheus
:
:
617 = Pandora

699 = Saturn mass center

The user's program must be able to "load" as many of these files as needed to satisfy his requirements at a given time. It is strongly recommended that such programs have the flexibility to load a list of SPK files provided to the program at run time.

Sample* SPK File Time Coverages

Navigation Ancillary Information Facility



* Note: This is not a bona fide Cassini scenario; it is a highly simplified illustration of some of the possibilities for ephemeris delivery on a planetary mission.

Summarizing an SPK File - 1

- A brief summary can be made using the NAIF Toolkit utility “brief”
    - Summary => bodies and start/stop epochs
- At your command line prompt, type the program name (with path), followed by the name of the binary SPK file that is to be summarized.

```
% /SPICE/bin/brief  sat052.bsp

 BRIEF --- Brief SPK Summary Tool, Version 2.x
 Times are given as calendar format ephemeris time, not UTC

 SPK file sat052_nov2003_dec2014.bsp in brief:

   Bodies: 601, 602, 603, 604, 605, 606, 607, 608, 609, 699

   Begin Ephemeris             End Ephemeris
   -------------------------   -------------------------
   2003 NOV 17, 00:00:00.000   2014 DEC 19, 00:00:00.000
```

Summarizing an SPK File - 2

- A detailed summary can be made using the NAIF Toolkit utility “SPACIT.” See the SPACIT User’s Guide for details.

```
 Summary for SPK file: sat052_nov2003_dec2014.bsp

--------------------------------------------------------------------------------
Segment ID     : SAT052
UTC Start time: 2003 NOV 16 23:58:57.817
UTC Stop time : 2014 DEC 18 23:58:57.816
ET Start time :  1.2229920000000E+08
ET Stop time  :  4.7221920000000E+08
Target Body    : MIMAS (601)
Center Body    : SATURN BARYCENTER (6)
Inertial frame: DE-200 (14)
SPK Data Type : Chebyshev polynomials: position only (2)
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Segment ID     : SAT052
UTC Start time: 2003 NOV 16 23:58:57.817
UTC Stop time : 2014 DEC 18 23:58:57.816
ET Start time :  1.2229920000000E+08
ET Stop time  :  4.7221920000000E+08
Target Body    : ENCELADUS (602)
Center Body    : SATURN BARYCENTER (6)
Inertial frame: DE-200 (14)
SPK Data Type : Chebyshev polynomials: position only (2)
--------------------------------------------------------------------------------
                              :
                              :
```

(This is a partial output; not all data could be displayed on this chart)

How are SPK Files Made?

- Mission design teams and project navigation teams are the usual sources of such data
  - But NAIF/JPL does make generic SPK files for planets, satellites, comets and asteroids, based on standard JPL ephemeris sources
- An SPK file may be made directly as an output of the ephemeris or trajectory propagation process
- An SPK file may be made by a NAIF utility program that converts some other ephemeris representation into an SPK format
  - A Type 5 SPK file can be made from two or more sets of osculating elements or state vectors
  - A more general utility is under construction: it will handle a table of discrete state vectors, conic elements, or space command elements*

* Capability to be added to SPICE late in FY98

Problems Using SPK Files - 1

- You're trying to use an SPK file that is not properly formatted for your computer
  - You can read only a binary SPK file with the SPK subroutines; you CAN'T read a "transfer format" file
    - » Although not required, binary SPK files often have a name like "xxxxxx.bsp"
    - » Although not required, transfer format SPK files often have a name like "xxxxxx.xsp" (formerly "xxxxxx.tsp")
  - You must have the proper kind of binary file for your computer
    - » Sun, HP, SGI, MAC and NeXT all use the same (IEEE) binary standard
    - » DEC Alpha and PC use different binary standards
  - The NAIF Toolkit utility program SPACIT, or the pair of programs named TOBIN and TOXFR, can be used to port SPK files between computers with dissimilar binary standards

Problems Using SPK Files - 2

- The file, or files, you’ve loaded do not contain data for both your target and observer bodies.

- The file, or files, you’ve loaded do not cover the time for which you’ve requested a state vector.

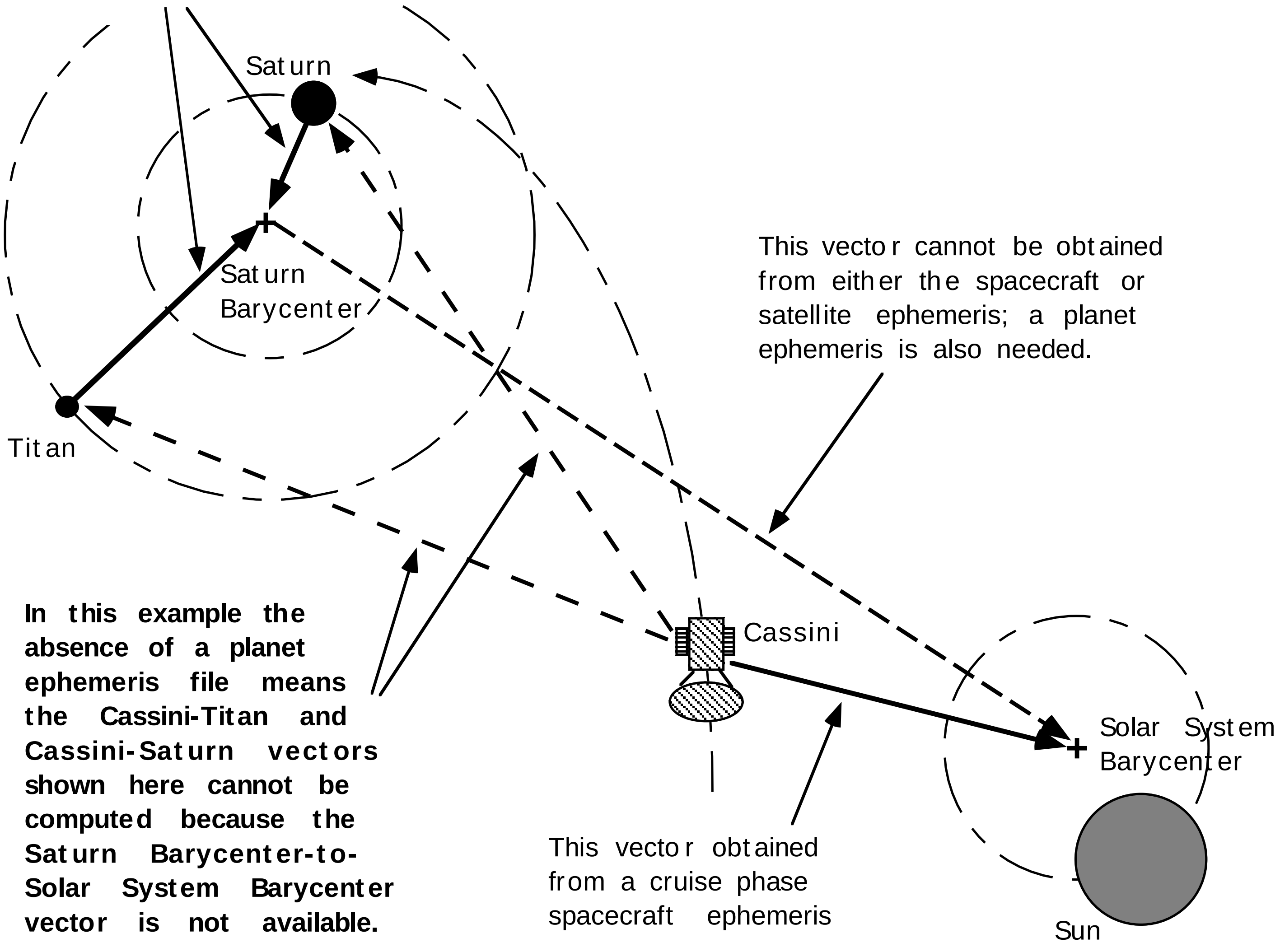
In both of these cases you’ll get an error message like the following:

```
SPICE(SPKINSUFFDATA) - -
Insufficient ephemeris data has been loaded to
compute the state of 602 relative to -82.
```

Problems Using SPK Files - 3

- Another problem, not often encountered, can arise when your files contain data for both target and observer, and cover the needed time period, but ephemeris data for an intermediate body needed to link together both target and observer is missing.
  - Example: You load a spacecraft SPK containing ephemeris for Cassini (-82) relative to the sun (10), and you load a satellite SPK containing the ephemeris for Titan (606) and Saturn (699) relative to the Saturn barycenter (6). But you forgot to load a planet SPK file that contains data for both the Saturn barycenter and the sun relative to the solar system barycenter (0). The SPK software cannot “connect” Cassini to Titan or to Saturn. See attached drawing.
  - In this case, knowing what is the “Center Body” of movement for each target body is important; this is shown in SPACIT summaries but is not shown in BRIEF summaries.

Problems Using SPK Files - 3 (drawing)



These vectors obtained from satellite ephemeris

This vector cannot be obtained from either the spacecraft or satellite ephemeris; a planet ephemeris is also needed.

In this example the absence of a planet ephemeris file means the Cassini-Titan and Cassini-Saturn vectors shown here cannot be computed because the Saturn Barycenter-to-Solar System Barycenter vector is not available.

This vector obtained from a cruise phase spacecraft ephemeris

Additional Information on SPK

- For more information about SPK, look at the following documents:
  - STATES cookbook program and User's Guide
  - Most Useful SPICELIB Subroutines
  - SPK Required Reading
  - Headers of the SPKEZ, SPKEZR and SPKLEF subroutines
  - BRIEF and SPACIT User's Guides

- Related documents:
  - NAIF_IDS Required Reading
  - Frames Required Reading
  - Time Required Reading
  - Pool Required Reading

Note: All code and documents mentioned above are included as files in every NAIF Toolkit delivery.