Multimission Software Interface Specification (SIS)

SPICE
Spacecraft and Planet Ephemeris Kernel

SPK

NAIF Document No. 367
Version 1.0

Prepared by: C. Acton

Navigation and Ancillary Information Facility (NAIF)
Jet Propulsion Laboratory
National Aeronautics and Space Administration

ABSTRACT: This Software Interface Specification outlines the format and content of SPICE SPK files containing ephemeris (trajectory) data for objects whose state (position and velocity) can be represented using the SPICE SPK subsystem. More importantly it

describes and gives examples of how to use SPICE Toolkit software to access the data in an SPK file. This is a multimission SIS. It can be supplemented with mission-specific additions where needed.

CHANGE LOG

Version	Date	Page Nos.	Reason
1.0	25 May 2000	All	New multimission version.

List of Acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee on Space Data Standards
CK	SPICE C-kernel
DAF	SPICE Double Precision Array File
JPL	Caltech/Jet Propulsion Laboratory
FTP	File Transfer Protocol
FTS	SFOC File Transfer Service
MGSO	Multimission Ground Systems Office
NAIF	Navigation and Ancillary Information Facility
PDB	Project Data Base
PDS	Planetary Data System
SDT	Galileo Science Data Team
SFDU	Standard Formatted Data Unit
SIS	Software Interface Specification
SPICE	S-, P-, I-, C- and E-kernels; the principal logical data components of a particular NASA ancillary information system
SPK for	S- and P-kernel file, the SPICE component that contains ephemeris data spacecraft and/or target bodies
VMS	Digital Equipment Corporation's Virtual Memory Operating System

Section 1
General Description

1.1   Purpose of Document

This Software Interface Specification (SIS) describes the use of SPICE Toolkit software for accessing (reading) the ephemeris data in SPICE S- and P-ephemeris kernel (SPK) files. It also outlines the structure and contents of SPK files, and it provides references to documents describing the details about structure and contents.

1.2   Scope

This is a multimission SIS, applicable to all projects and processes that use SPICE ephemeris products (SPK files).

1.3   Applicable Documents

The following reference documents are all available within each copy of the SPICE Toolkit.

No.	Name	NAIF Doc. No.
1	SPK Required Reading	168
2	DAF Required Reading	167
3	SPC Required Reading	211
4	NAIF Ids Required Reading	219
5	MKSPK User's Guide	357
6	BRIEF User's Guide	338
7	COMMNT User's Guide	313
8	SPACIT User's Guide	169
9	Convert User's Guide	333

Also of use are separate SPICE tutorials on reading and writing SPK files.

1.4   Functional Description

SPK files are the physical realization of two logical elements of the SPICE system—the S-kernel (spacecraft ephemeris) and the ephemeris portion of the P-kernel (planet, satellite, asteroid and comet ephemerides). When read using an appropriate subroutine from the SPICE Toolkit, an SPK file will yield state vectors—Cartesian position and velocity—of one

user-specified ephemeris object relative to another specified ephemeris object, at a specified epoch and in a specified reference frame.

In SPICE parlance an "ephemeris object" is most typically a spacecraft, planet, satellite, comet or asteroid. But it can also be any defined object whose location relative to some other object is defined with data in SPK format: additional examples are tracking station locations, landers, rovers, and appendages on space vehicles.

To facilitate rapid information access (whether sequential or random access) and to minimize the file size, SPK files are binary files. To allow porting of an SPK file between computers that use different binary data representations a "SPICE Transfer Format" is used. A transfer format SPK file consists of hexadecimal representations of the data, using ASCII characters.

An SPK file can be structured as a CCSDS Standard Format Data Unit (SFDU) where so required. In this case the SPK file is first converted to "SPICE Transfer Format" before the SFDU "wrapper" is attached. SPICE Toolkit utility programs that convert transfer format SPK files to binary format will ignore (not be impacted by) the presence of SFDU records.

SPK files can be read with library routines provided in the SPICE Toolkit, which is available in both ANSI FORTRAN 77 and ANSI C for most popular platforms.

1.4.1   Data Source, Destination and Transfer Method

SPK files made for flight projects are usually produced on a Mission Design Team or Navigation Team computer. However, any person having access to the SPICE Toolkit or an SPK production program may produce an SPK file.

SPK files made by a flight project are typically placed in a project's general database or on a SPICE server to allow access by a wide assortment of customers. SPK files are transferred to the project's database or SPICE server using file transfer protocol (ftp) or direct copy.

Since SPK files can exist in both the binary format (often named as *.bsp) and SPICE "transfer format" (often named as *.xsp), care must be taken by the user to determine which format file is to be transferred and then use an appropriate transfer mechanism for that type. The user is also reminded that binary format SPK files cannot be ported between computers using different binary data standards; this is the reason SPK files are often stored in the project database in SPICE transfer format.

The SPICE Toolkit includes utilities for converting SPK files between the binary and transfer formats, and vice versa.

1.4.2   Labeling and Identification

SPK files include a "comment area" that may hold any amount of ASCII information pertinent to the SPK file. One such use is for metadata—information about the particular instance of an SPK file that is important to SPK users. Refer to Applicable Document No. 3 for a detailed description of the comment area, including Toolkit software for creating, adding, viewing, extracting or deleting "comments".

The comment area might be used, in part, to store label information that will subsequently be used in the construction of SFDU K-header labels or NASA Planetary Data System (PDS) labels. Where used, the specific content and structure of these labels will be determined by the flight project or the PDS.

The SPK file naming convention is selected by the responsible flight project team or other SPK producer. It is recommended that binary and transfer format SPK file names use "*.bsp" and "*.xsp" extensions respectively.

1.4.3   Assumptions and Constraints

SPK files must be read using the SPK "reader" routines provided in the SPICE Toolkit; no one need or should try writing their own SPK access routines. These "reader" routines are incorporated in a user's own program, written in a language that can access ANSI FORTRAN 77 or ANSI C routines. Refer to Applicable Document No. 1 for a detailed discussion of how to read SPK files.

SPK files must be written using routines provided in the SPICE Toolkit (Applicable Document No. 1) or by an existing application program designed for this purpose. One such application is the MKSPK program (Applicable Document No. 5). The user is advised to contact NAIF if interested in writing SPK files.

If a user decides to define a new SPK data type, the new type must be approved by and registered with the NAIF group at JPL. Also, at least two new subroutines must be written and added to the Toolkit. Contact NAIF for help.

A program constructed using Toolkit software may not simultaneously read ephemeris data from, and add new ephemeris data to, an SPK file.

Section 2
Environment

2.1   Hardware Characteristics and Limitations

        In excess of 650 Kbytes of memory is required to utilize the SPICE Toolkit on a CPU that does not provide virtual memory. A PC running DOS will not be able to use the Toolkit. With that one exception, SPICE software should work properly on any machine that correctly supports the ANSI FORTRAN 77 or ANSI C standard.

        The SPICE Toolkit (either FORTRAN or C) has been built and tested on numerous platforms. It normally is delivered with all included library and program modules built and ready to use. In some cases the delivery script provided with the Toolkit goes through a complete build process.

2.2   Interface Medium and Characteristics

        An SPK file must be in the local binary format to be accessed by SPICE Toolkit "reader" software.

        SPK files are sometimes converted into the SPICE transfer format, and where required, structured as SFDUs, before being transferred to another computer. Any text transfer mechanism may be used in this case, such as ASCII mode of ftp. If the file is to be transferred between two computers that use identical binary data representations then a binary transfer method may be used (an example is ftp, set to operate in binary mode). As an example, Sun, Hewlett Packard (HP), Silicon Graphics (SGI), Macintosh and NeXT computers all use the same (IEEE) binary standard, whereas PCs running Windows, PCs running Linux, DEC Alphas running VMS and DEC Alphas running Digital Unix each use different, incompatible binary standards.

2.3   Deformatting

        If the SPK file is received in the SPICE transfer format it must first be converted to local binary format using a SPICE utility program (TOBIN or SPACIT, Applicable Document No. 9) or an appropriate SPICELIB or CSPICE routine, available in the SPICE Toolkit. If the transfer format file is wrapped with SFDU records, these will be ignored (skipped) by the SPICE Toolkit's TOBIN and SPACIT utilities starting with version N0050 of the SPICE Toolkit. (Earlier versions of TOBIN and SPACIT will not skip over SFDU wrappers; in this case the user must remove the SFDU using other means.)

2.4   Calling Sequence

SPK files are read using one of several "reader" routines provided in the SPICE Toolkit. See Attachment 2 for a full discussion of the readers available.

A program may access one or several SPK files at run time simply by using multiple SPK file load directives within the program. Some compilers restrict the total number of open files for a given process. SPICE software contains a parameter restricting the total number of open files to 20 to be within the lowest known bound on this constraint.

2.5 End-Of-File Convention

SPK files are terminated with the normal end of file convention of the computer on which they are produced.

2.6 Utility Programs

The TOXFR and TOBIN utilities may be used to convert a binary SPK to transfer format and transfer format SPK to binary, respectively. The SPACIT program (Applicable Document No. 8) may also be used to convert SPK files back and forth between binary and SPICE transfer formats, although TOXFR and TOBIN are generally easier to use.

The BRIEF (Applicable Document No. 6) and SPACIT utilities may be used to summarize the contents and time coverage of a binary format SPK file.

The COMMNT program (Applicable Document No. 7) may be used to add comments to, extract comments from, delete comments in or view comments in a binary format SPK file. SPACIT may also be used to view any comments.

Section 3
Data Flow Characteristics

3.1   Operational Characteristics

3.1.1   Generation Method and Frequency

        For JPL and some other flight projects NAIF provides a complete SPK generation program set to the Navigation Team or other group responsible for generating SPK files. SPK files are generated on a schedule determined by project requirements.

3.1.2   Time Span of Product

        The time span for each SPK file is determined by the producer of that file. Some degree of consistency may be maintained for SPK files produced by a particular flight project for a specific purpose. For example, a navigation team might regularly produce "archival" SPK files spanning a period of seven days and "predict" SPK files spanning a period of two months.

        While not a requirement, the start and stop times of each SPK file could be included with other pertinent descriptive information about the file in the comment area of the file. For those projects using SFDUs these times will also be included in the SFDU K-Header.

3.2   Data Volume

        The size of an SPK file is determined principally by the time span of the file, the number of ephemeris objects included in the file, and the SPK data type(s) used in the file. Estimated SPK file sizes for several example applications are shown in Table 3-1. The size of a transfer format SPK file is about 2.5 to 3 times the size of the corresponding binary SPK file.

Example Application	Estimated Size
Galileo cruise during the Earth-I flyby: Archive file of 380 days duration containing data for the Galileo spacecraft, Gaspra asteroid, Earth, Moon, Earth-moon barycenter and sun	0.6 Mbytes
MGS during mapping phase: Archive file of 7 days duration containing data for the MO spacecraft, Mars, Phobos, Deimos, Mars barycenter, Earth, Earth-moon barycenter and sun	3.7 Mbytes
Cassini tour: Advance planning file using data derived from patched conic element sets, covering 2004 Nov 8 to 2008 Jun 30 and containing data for only the Cassini spacecraft.	25 Mbytes

Note: sizes shown are for transfer format SPK files. Once received at a user's site and converted to the local binary format required by the SPICE Toolkit "readers", an SPK file will be reduced in size by a factor of approximately 2.5 to 3.

Table 3-1
Examples of SPK File Size
Section 4
Detailed Data Object Definition

4.1   Structure and Organization Overview

SPK files may be delivered to a project database in transfer format or in binary format. If transfer format is used, the file may or may not be encased in SFDU wrappers. If binary format is used, which binary form must be known to the end user.

If an SFDU version of an SPK file has been received, use an SFDU stripper tool to strip off leading and trailing SFDU-specific records, leaving a SPICE transfer format SPK file. If using SPICE Toolkit version N0050 or later the TOBIN and SPACIT tools ignore the SFDU wrappers when converting a transfer format SPK file to binary format.

If a SPICE transfer format SPK file has been received, or once SFDU structures have been removed as described above, use the NAIF Toolkit utility program TOBIN or SPACIT to convert the transfer format SPK file to the local binary format. This is a straight forward process requiring no knowledge of SPK file structures or host computer conventions for text or binary files.

4.2   Data Format and Definition

The next sections provide an introduction to the structure of SPK files. Specific structure details and specification of the contents are found in the appendices as noted.

Under most circumstances those who intend only to read an SPK file need understand few if any structure details: the user's interface to an SPK file is through SPICE Toolkit subroutine argument lists. Examples of this interface are found in Applicable Document No. 1.

4.2.1   Transfer Format SPK Format Description

A description of the transfer format SPK file is not provided since this format is used only to facilitate file porting. Use the TOBIN or SPACIT utility program to convert a transfer format SPK file to local binary format for use in your application programs.

4.2.3   Binary SPK Format Description

          Ignoring host-specific file control components and architecture details, a
binary SPK file is structured as shown in Figure 4-2.

```
+------------------------+
|                        |
|      Comment area       |
|------------------------|
|                        |
|      Ephemeris data     |
|                        |
+------------------------+
```
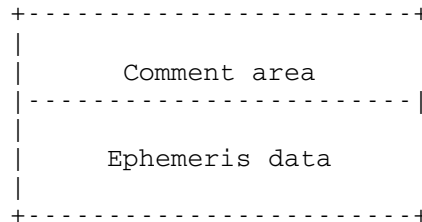Figure 4-2
Binary SPK File Structure

          Details about this structure, and examples of using the Toolkit "readers"
(subroutines) to access the data in an SPK file, are found in Applicable Document No. 1. The
size of this document may be intimidating. However, it is not particularly difficult reading and it
is filled with practical examples. Novice SPK users may model their code after these examples.
Several SPICE Toolkit "cookbook" programs (STATES or SIMPLE) provide working examples
of code designed to read SPK files.

          The comment area may be used to hold metadata describing how and why the
particular SPK file was created. Details about the comment area, including how to use Toolkit
software to read it, write to it, extract it, delete it or view it are found in Applicable Document
Nos. 3 and 7.

          The ephemeris data found in SPK files can be provided in any of a variety of
different forms—called SPK Data Types. Some examples of these SPK data types are:

          • Modified difference arrays
          • Chebyshev polynomials
          • Two-body conic elements
          • Space Command "Two-line elements"
          • Lagrange interpolation of discrete state vectors
          • Hermite interpolation of discrete state vectors
          • Precessing ellipse conic elements

The common element in all of these SPK data types is that, with the appropriate evaluator from
SPICELIB or CSPICE, they all yield a state vector for an ephemeris object at a time requested by
the user.

          SPK ephemeris data, such as the polynomials or conic elements mentioned
above, are stored in a data structure called a Double Precision Array File (DAF). Complete

11

specifications for DAFs are found in Applicable Document No. 2. Anyone needing a detailed understanding of the SPK file structure and the most fundamental (low level) Toolkit subroutines for working with SPK files should read this document.

SPK files use a numeric code to identify ephemeris objects—spacecraft, planets, satellites, comets and asteroids. A listing of these codes and the objects to which they pertain is found in Applicable Document No. 4.