Time Conversion and Time Formats

**January 2017**

Time Systems and Kernels

- **Time inputs to and outputs from <u>user's programs</u> are usually strings representing epochs in these three time systems:**
  - Ephemeris Time (*ET*, also referred to as Barycentric Dynamical Time, *TDB*)
  - Coordinated Universal Time (*UTC*)
  - Spacecraft Clock (*SCLK*)

- **Time stamps in kernel files, and time inputs to and outputs from <u>SPICE routines</u> reading kernel data and computing derived geometry, are double precision numbers representing epochs in these two time systems:**
  - Numeric Ephemeris Time (TDB), expressed as ephemeris seconds past J2000
  - Encoded Spacecraft Clock, expressed as clock ticks since the clock start

- **SPICE provides routines to convert between these string and numeric representations.**

- **A time string used as an argument in a SPICE API must be provided in quotes.**
  - Fortran, Matlab and IDL: use single quotes
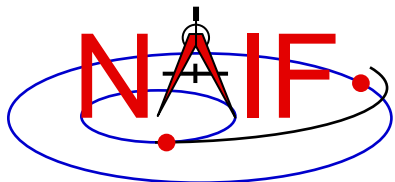 - C: use double quotes
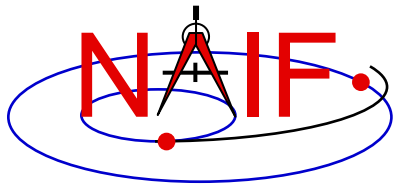
Converting Time Strings

- ## UTC, TDB, or TDT (TT) String to numeric Ephemeris Time
  - **STR2ET (*string*, ET)**
    - » Converts virtually any time string format known to the SPICE Time subsystem, excepting SCLK. For example:

      '1996-12-18T12:28:28'   '1978/03/12 23:28:59.29'   'Mar 2, 1993 11:18:17.287 p.m. PDT'
      '1995-008T18:28:12'   '1993-321//12:28:28.287'
      '2451515.2981 JD'   ' jd 2451700.05 TDB'
      '1988-08-13, 12:29:48 TDB'   '1992 June 13, 12:29:48 TDT'
    - » Requires the LSK kernel

- ## Spacecraft Clock String to numeric Ephemeris Time
  - **SCS2E (*scid*, *string*, ET)**
    - » Converts SCLK strings consistent with SCLK parameters. For example:
      '5/65439:18:513' (VGR1), '946814430.172' (MRO), '1/0344476949-27365' (MSL)
    - » Requires a SCLK kernel and the LSK kernel
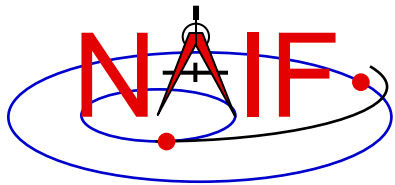
Converting Numeric Times

- **Numeric Ephemeris Time to Calendar, DOY or Julian Date *UTC*, *TDB*, or *TDT* String**
  - **TIMOUT (*et, fmtpic*, STRING)**
    - » ***fmtpic* is an output time string format specification, giving the user great flexibility in setting the appearance of the output time string and the time system used (*UTC*, *TDB*, *TDT*).**
      - **See the next slide for examples of format pictures to produce a variety of output time strings**
      - **See the TIMOUT header for complete format picture syntax**
      - **The module TPICTR may be useful in constructing a format picture specification from a sample time string**
    - » **Requires LSK Kernel**

- **Numeric Ephemeris Time to Spacecraft Clock String**
  - **SCE2S (*scid*, *et*, SCLKCH)**
    - » **Requires the LSK and a SCLK kernel**
    - » **Output SCLK string examples:**

      ```
      '1/05812:00:001'        (Voyager 1 and 2)

      '1/1487147147.203'      (Cassini, MRO)

      '1/0101519975.65186'    (MEX, VEX, Rosetta)
      ```

Example Time Strings and the Corresponding Format Pictures

Common Time Strings	Format Picture Used (*fmtpic*)
`1999-03-21T12:28:29.702`	`YYYY-MM-DDTHR:MN:SC.###`
`1999-283T12:29:33`	`YYYY-DOYTHR:MN:SC ::RND`
`1999-01-12, 12:00:01.342 TDB`	`YYYY-MM-DD, HR:MN:SC.### ::TDB TDB`
`2450297.19942145 JD TDB`	`JULIAND.####### ::TDB JD TDB`

Less Common Time Strings	Format Picture Used (*fmtpic*)
`465 B.C. Jan 12 03:15:23 p.m.`	`YYYY ERA Mon DD AP:MN:SC ampm`
`04:28:55 A.M. June 12, 1982`	`AP:MN:SC AMPM Month DD, YYYY`
`Thursday November 04, 1999`	`Weekday Month DD, YYYY`
`DEC 31, 15:59:60.12 1998 (PST)`	`MON DD, HR:MN:SC YYYY (PST)::UTC-8`
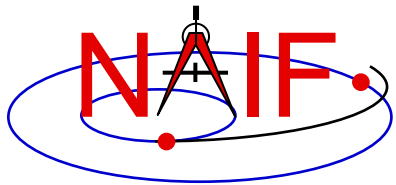
Additional Time Conversions

- **Numeric Ephemeris Time to Local Solar Time String**
    - **ET2LST(*et*, *body*, *long*, *type*, HR, MN, SC, TIME, AMPM)**
        - » **Requires SPK (to compute *body* position relative to the Sun) and PCK (to compute *body* rotation) kernels**

- **Numeric Ephemeris Time to planetocentric longitude of the Sun (*Ls*)**
    - **LS = LSPCN (*body*, *et*, *abcorr*)**
        - » **While *Ls* is not a time system, it is frequently used to determine *body* season for a given epoch**
            - **LS = 0° , Spring**
            - **LS = 90° , Summer**
            - **LS = 180° , Autumn**
            - **LS = 270° , Winter**

            For the northern hemisphere
        - » **The *Ls* calculation requires SPK and PCK kernels**
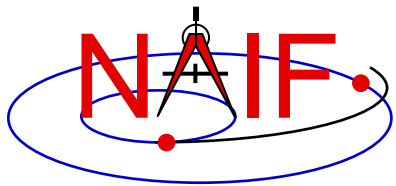
Navigation and Ancillary Information Facility

Kernels needed

- needs lsk
- needs sclk
- needs lsk and sclk
- needs pck and spk
- needs no kernels

= APIs mentioned in this tutorial