



---

Navigation and Ancillary Information Facility

# Using the Frames Subsystem

April 2023



# What is the Power of Frames?

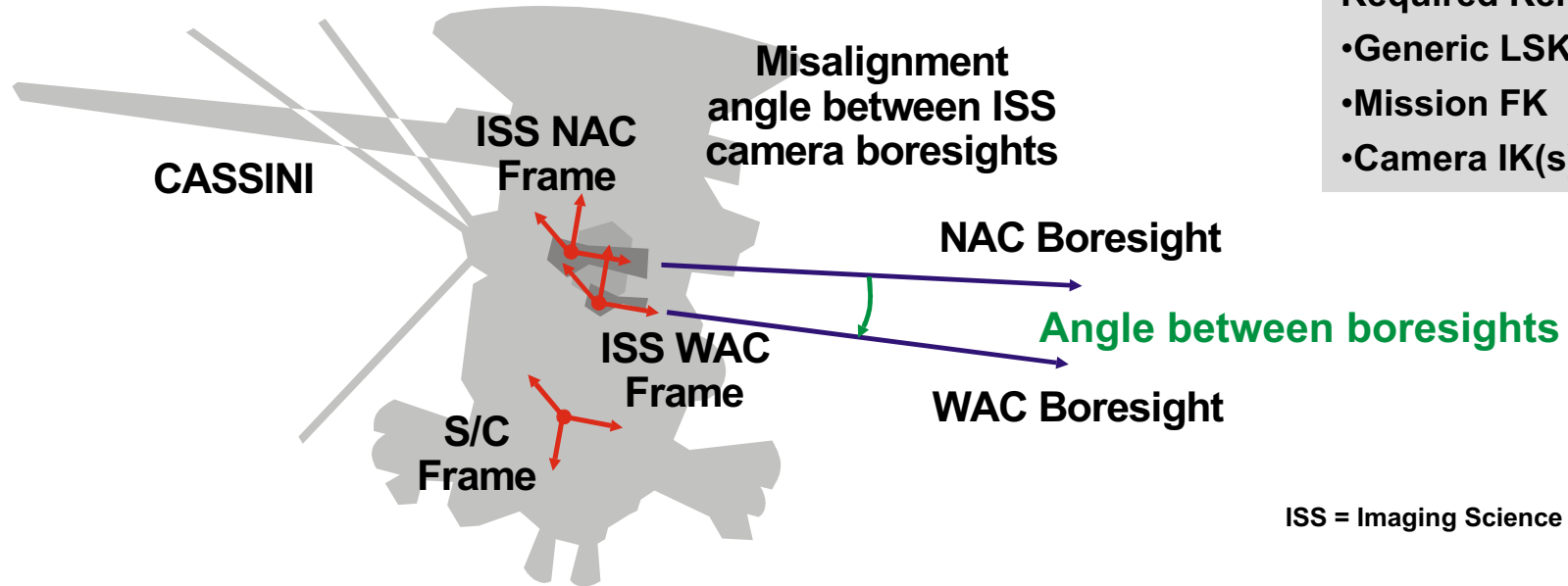
Navigation and Ancillary Information Facility

- The “power” of the Frames capability stems from the SPICE system’s ability to construct complex reference frame transformations with no programming effort required of you
  - But it’s crucial that you select and load the needed kernels
- The principal benefit from the Frames capability is obtained through the main SPK subsystem interfaces (SPKEZR and SPKPOS), the Frames subsystem interfaces (SXFORM, PXFORM, PXFRM2), and high-level derived geometry APIs (SINCPT, TANGPT, SUBPNT, ...).
- The remaining pages illustrate typical use of frames
- Several **VERY IMPORTANT** usage issues are mentioned in the Frames tutorial; be sure to also read that.



# Offset Between Instruments

Navigation and Ancillary Information Facility



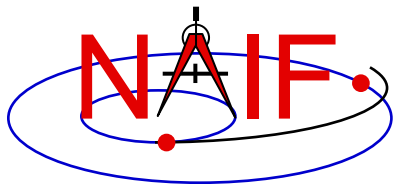
ISS = Imaging Science System

**Compute the angular separation between the Cassini ISS Narrow Angle Camera and Wide Angle Camera boresights:**

```
C      Retrieve the matrix that transforms vectors from NAC to WAC frame
      CALL PXFORM( 'CASSINI_ISS_NAC', 'CASSINI_ISS_WAC', ET, MAT )

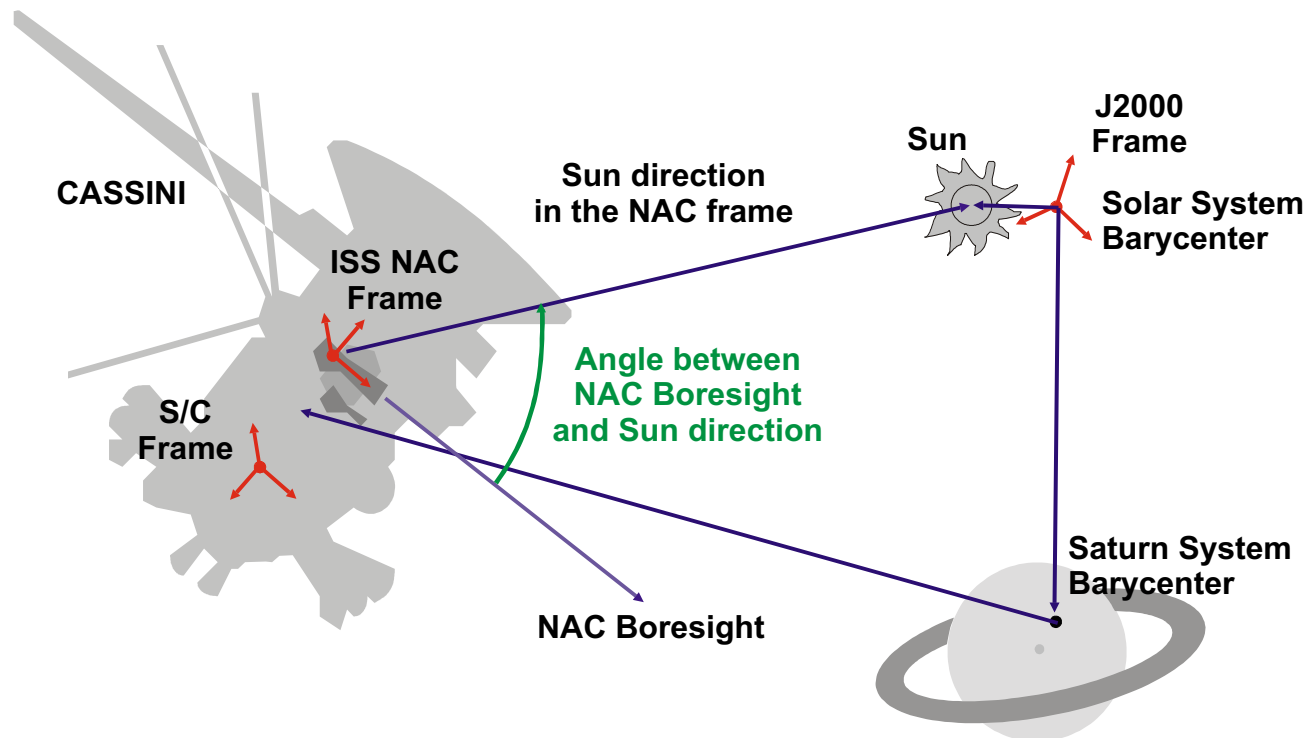
C      Transform NAC boresight to WAC frame and find separation angle
      CALL MXV ( MAT, NAC_BORESIGHT_nac, NAC_BORESIGHT_wac )
      ANGLE = VSEP( NAC_BORESIGHT_wac , WAC_BORESIGHT_wac )
```

Fortran  
example



# Angular Constraints

Navigation and Ancillary Information Facility



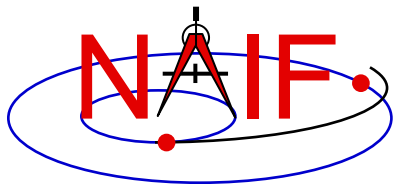
Required Kernels:

- Generic LSK
- Mission FK
- Spacecraft SCLK
- Camera IK
- Planetary Ephemeris SPK
- Spacecraft SPK
- Spacecraft CK

**Check whether the angle between the camera boresight and the direction to the Sun is within the allowed range:**

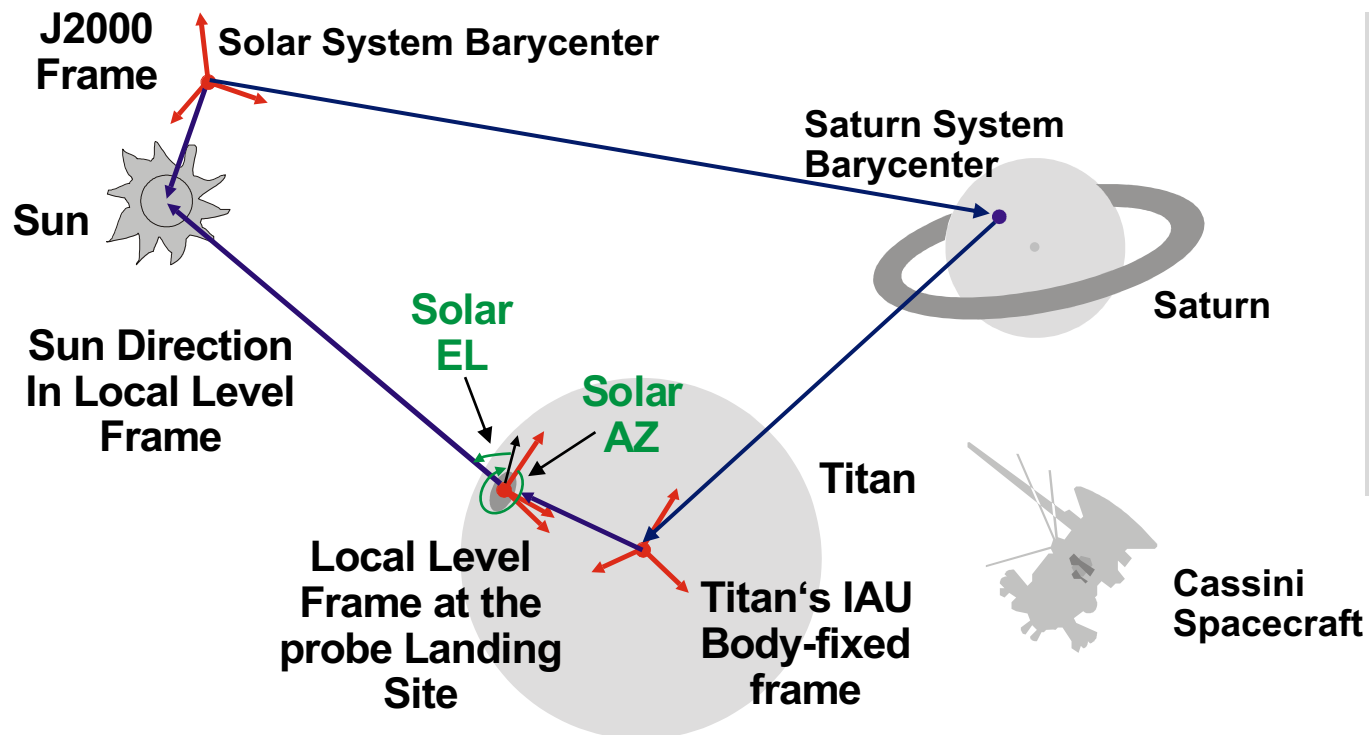
```
CALL SPKPOS( 'SUN', ET, 'CASSINI_ISS_NAC', 'LT+S', 'CASSINI', SUNVEC, LT )
ANGLE = VSEP( NAC_BORESIGHT_nac, SUNVEC )
IF ( ANGLE .LE. CONSTRAINT ) WRITE(*,*) 'WE ARE IN TROUBLE!'
```

Fortran  
example



# Angles at the Surface

Navigation and Ancillary Information Facility



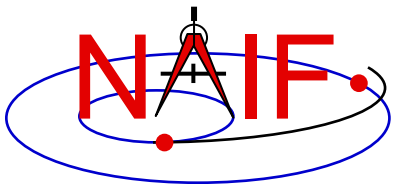
Required Kernels:

- Generic LSK
- Generic PCK
- Mission FK
- Planetary Ephemeris SPK
- Satellite Ephemeris SPK
- Landing Site SPK

Compute solar azimuth and elevation at the Huygens probe landing site

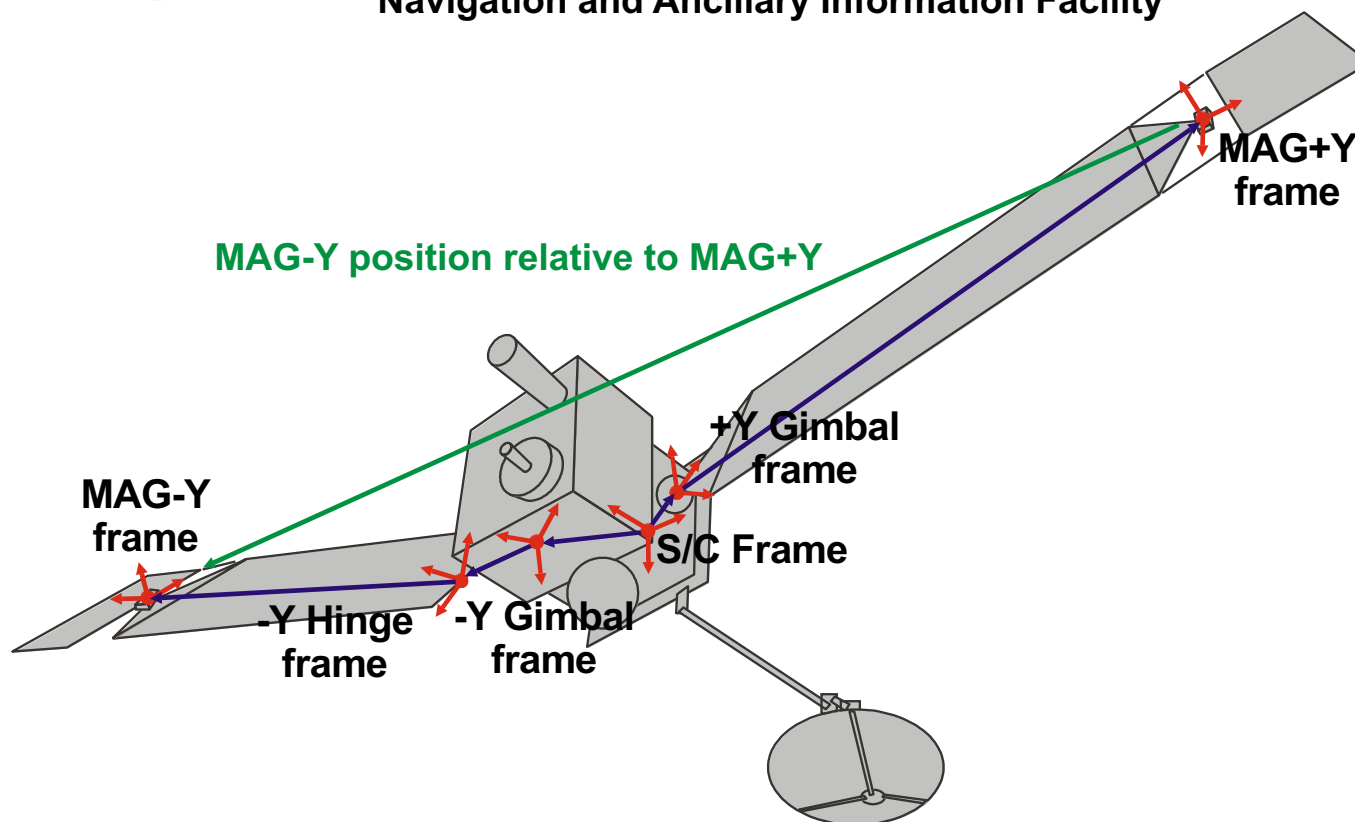
```
CALL SPKPOS('SUN',ET, 'HUYGENS_LOCAL_LEVEL', 'LT+S', 'HUYGENS_PROBE',SUNVEC,LT)
CALL RECAZL( SUNVEC, .TRUE., .FALSE., RANGE, AZ, EL )
```

Fortran  
example



# Relative Position of Sensors

Navigation and Ancillary Information Facility



Required Kernels:

- Generic LSK
- Mission FK
- Structure Locations SPK
- Spacecraft SCLK
- Solar Array CK

Find the position of one MGS MAG sensor with respect to the other in the MGS s/c frame. Also find the relative orientation of the sensors:

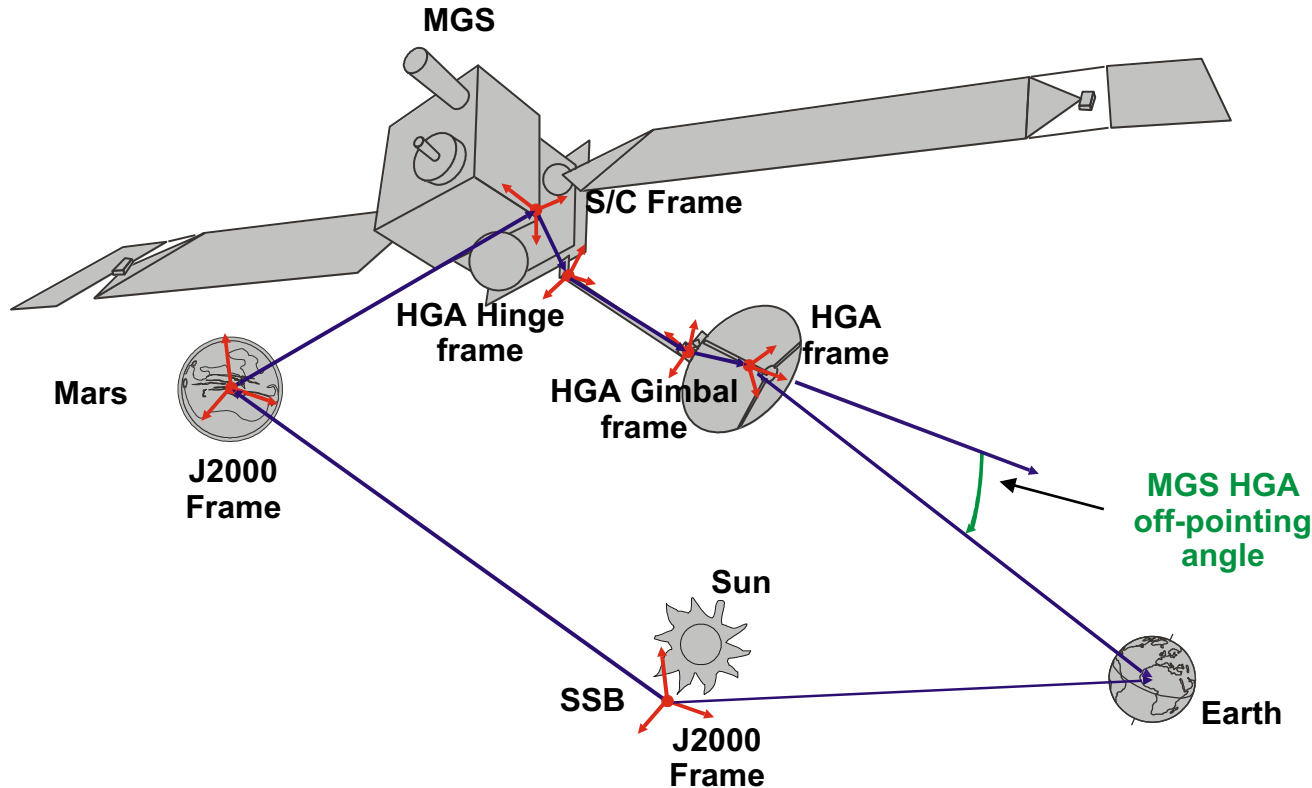
```
CALL SPKPOS('MGS_MAG-Y', ET, 'MGS_SPACECRAFT', 'NONE', 'MGS_MAG+Y', POSVEC, LT)
CALL PXFORM('MGS_MAG+Y_SENSOR', 'MGS_MAG-Y_SENSOR', ET, MAT)
```

Fortran  
example



# Manipulators - 1

## Navigation and Ancillary Information Facility



### Required Kernels:

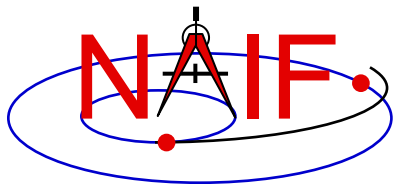
- Generic LSK
- Mission FK
- Spacecraft SCLK
- HGA IK
- Planetary Ephemeris SPK
- Spacecraft SPK
- Spacecraft CK
- HGA CK

HGA = High Gain Antenna

**Compute the angle between the direction to Earth and the MGS HGA boresight:**

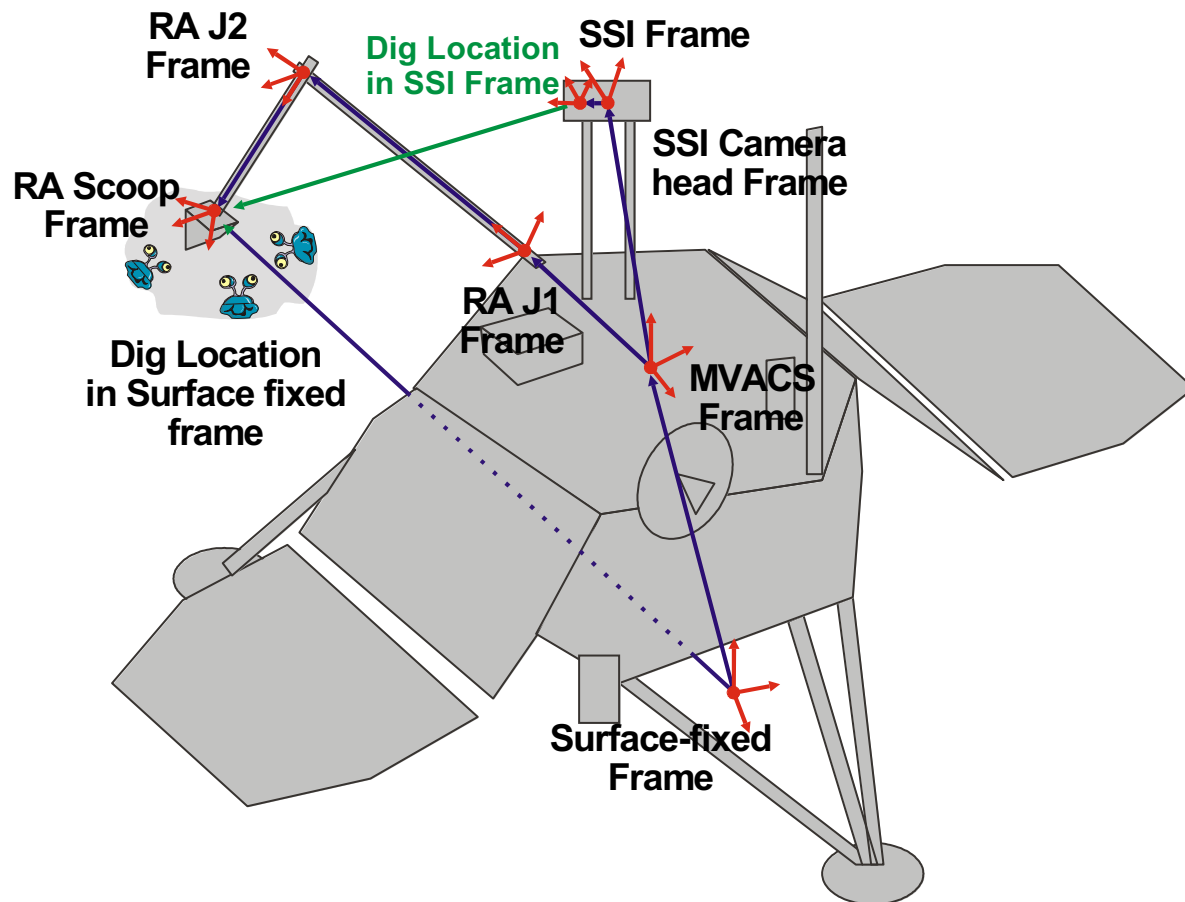
```
CALL SPKEZR( 'EARTH', ET, 'MGS_HGA', 'LT+S', 'MGS', EARTH_STATE, LT )
ANGLE = VSEP( HGA_BORESIGHT, EARTH_STATE )
```

Fortran  
example



# Manipulators - 2

## Navigation and Ancillary Information Facility



### Required Kernels:

- Generic LSK
- Mission FK
- Lander SCLK
- Structure Locations SPK
- Lander SPK
- Lander CK
- SSI CK
- RA CK

**Compute the soil digging location in the MPL surface-fixed and camera left eye frames:**

```
CALL SPKPOS ( 'MPL_RA_SCOOP', ET, 'MPL_SURFACE_FIXED', 'NONE', 'MPL_SURF', POS1, LT )
CALL SPKPOS ( 'MPL_RA_SCOOP', ET, 'MPL_SSI_LEFT', 'NONE', 'MPL_SSI', POS2, LT )
```

*Fortran  
example*