esa esac

Camino bajo del Castillo, s/n
28692 Villanueva de la Cañada
Madrid
Spain

DOCUMENT

SOC-Provided Ancillary Data for Solar Orbiter

Prepared by	**Andrew Walsh**
Reference	**SOL-SGS-TN-0017**
Issue	**1**
Revision	**0**
Date of Issue	**16/12/2021**
Status	**Issued**
Document Type	**TN**
Distribution	**Instrument Teams; Project Scientists; SOC; MOC; ESAC SPICE Service**

APPROVAL

Title	
Issue 1	**Revision** 0
Author	**Date** 16/12/2021
Approved by	**Date**

CHANGE LOG

Reason for change	Issue	Revision	Date
Update for start of nominal phase	1	0	16/12/2021
Addition of science frame descriptions, updates to some filenames and discussion of kernel distribution	0	4	04/12/2019
Addition of dataset description for orbit digest CDF	0	3	22/08/2018
Update following first 'preview' release of SPICE Kernel dataset	0	2	18/09/2017

CHANGE RECORD

Issue 1	**Revision** 0		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Updated description of as-flown attitude products to reflect new handling of rolls and AEM processing	16/12/2021	11-12	
Added VSO and GSM frame descriptions to table	16/12/2021	19	
Clarified metakernel description for new attitude kernels and what constitutes a new SKD	16/12/2021	23-24	
Updated orbit cdf filename description to follow software implementation	16/12/2021	25	

Table of contents:

1 INTRODUCTION

Ancillary data products are those products that are not strictly science data, but are still helpful in scientific analysis or the preparation of higher-level science data products, for example an orbit file containing the position and velocity of the spacecraft. For Solar Orbiter, the Science Operations Centre (SOC), based at ESAC near Madrid has the responsibility of producing those ancillary data products that are not relevant to only a single instrument, and that are not based on instrument telemetry, but rather platform telemetry (e.g. AOCS parameters in housekeeping) or other data available on ground. This is not only to reduce duplication of effort, but also to ensure consistency in the ephemerides etc. that are used in producing the higher-level science data products on the ground, and therefore make multi-instrument data analysis as simple as possible.

The purpose of this document is to describe the content and format of the various ancillary data products that will be produced by the Solar Orbiter SOC, and explain how they will be distributed to the rest of the Science Ground Segment (SGS) and the broader scientific community.

1.1 Applicable Documents

AD1	SOL-SGS-TN-0009	Metadata Definition For Solar Orbiter Science Data	Iss 2. Rev 2.

1.2 Reference Documents

RD1	SPICE Toolkit user guide (https://naif.jpl.nasa.gov/naif/documentation.html)		N65
RD2	SOL-SGS-ICD-0004	Solar Orbiter Interface Control Document for Low Latency Data CDF Files	Iss 1. Rev 2.
RD3	SOL-SGS-ICD-0002	Solar Orbiter Data Producer to Archive ICD	Iss 0. Rev 2.
RD4	SOL.S.ASTR.TN.00079	Solar Orbiter TM-TC and Packet Structure ICD	Iss 7.
RD5	1.2.1.1.1.1 CDF Documentation (http://cdf.gsfc.nasa.gov/html/cdf_docs.html)		3.6.3
RD6	SOL-SGS-ICD-0009	File Transfer SOC – Instrument Teams ICD	Iss 0. Rev 3.
RD7	SOL-SGS-ICD-0005	Solar Orbiter Interface Control Document for Low Latency Data FITS Files	Iss 1. Rev 2.
RD8	SOL-SGS-TN-0015	The Effect of Solar Orbiter Spacecraft Attitude on EPD and SWA Science Return	Iss 1. Rev 0.

RD9	SOL.S.STR.TN.00099,	Solar Orbiter Coordinate System Document EN-14	Iss 5.

1.3 List of Acronyms and Abbreviations

AEM	Attitude Ephemeris Message
AOCS	Attitude and Orbit Control System
CDF	Common Data Format
DDS	Data Distribution System
ESA	European Space Agency
ESAC	European Space Astronomy Centre
ESOC	European Space Operations Centre
FECS	Flight Events and Communications Skeleton
GAM	Gravity Assist Manoeuvre
GFTS	Generic File Transfer Service
LL	Low Latency (Data)
LL02	Low Latency Level 2
MOC	Mission Operations Centre
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
OBT	On Board Time (used interchangeably with SCET)
OEM	Orbit Ephemeris Message
PTR	Pointing Request
SCET	Spacecraft Elapsed Time (used interchangeably with OBT)
SFTP	Secure File Transfer Protocol
SGS	Science Ground Segment (SOC + Instrument Teams)
SOAR	Solar Orbiter Archive
SOC	Science Operations Centre
UTC	Coordinated Universal Time

2 SUMMARY OF ANCILLARY DATA PRODUCTS

SOC will primarily produce ancillary data products for and using the SPICE toolkit, provided by NAIF. The SPICE toolkit and associated data files ('kernels') have been successfully used on many ESA and NASA solar system science missions in the past. Detailed documentation [RD1] for the SPICE software can be found on the NAIF Website[1] and is also accessible via the ESA SPICE Service[2]. SOC will produce SPICE kernels that will provide the following:

- The position and orbital velocity of the Solar Orbiter spacecraft.
- Spacecraft attitude.
- The conversion between SCET and UTC.
- A frames kernel that will allow for the transformation of data between spacecraft coordinates, instrument coordinates and various heliophysical coordinate systems.
- Operational misalignments for the remote sensing instruments.
- Instrument kernels that define fields of view.

SOC will distribute the operational misalignments for the remote sensing primarily as a record of the misalignments used in the production of LL02 data [RD7] and generation of PTRs. In situ instrument alignments are expected to be stable enough relative to the angular resolution of the data such that fixed rotation matrices, included in the LL data files themselves [RD2] will be used and a separate product will not be necessary.

SOC will not produce products that will allow for the transformation of data between instrument coordinates and spacecraft coordinates to a level of accuracy suitable for scientific analysis for the simple reason that we will not have the expertise to produce them. Furthermore, unlike for some other solar system missions, instrument teams are expected to provide Level 2 science data products to the SOAR that are already in either spacecraft or heliophysical coordinate systems, as appropriate, and that have the necessary metadata such that they are suitable for scientific analysis without reference to SPICE kernels [RD3, AD1]. Our SPICE-based ancillary products are primarily designed to aid in the production of L2 and higher-level science data by the instrument teams.

In addition to the ancillary data products in SPICE format, SOC will produce a more limited set of ancillary data in CDF format. These products will provide a summary or digest of the spacecraft orbit and roll angle, and are intended for use as a quick reference by the SGS during planning, and also for basic situational awareness as a complement to the low latency data.

[1] https://naif.jpl.nasa.gov

[2] https://www.cosmos.esa.int/web/spice

3 SPICE-BASED ANCILLARY DATA PRODUCTS

The primary format in which SOC will distribute ancillary data is the SPICE format. SPICE is a comprehensive toolkit and set of data files specifically designed for common tasks involving ancillary data for solar system missions [RD1]. The following describes the SPICE-based ancillary products produced by SOC. We do not attempt here to discuss the full capabilities of SPICE and the comprehensive uses of each type of SPICE data file (kernel), which are well documented elsewhere, but rather focus on the specific SPICE kernels that we will produce. It is recommended that for any operational software that requires SOC ancillary data, the SPICE kernels rather than the CDF products are used.

3.1 Time Conversion Products

While a UTC packet time is available in the SCOS2000 headers of packets retrieved from the DDS, this information is not available for any acquisition times (for example) held within the data field of the packet. In the case of some instrument teams these acquisition times may be significantly different from the time at which the packet was created and as such the conversion between UTC and OBT for the data within the packet may not be the same as that in the packet header. As such SOC will provide a means of conversion between SCET and UTC for use by all instrument teams in the production of their data: the time correlation. This will ensure conversion to UTC is applied consistently across all data and will aid in multi-instrument data analysis. SOC will provide this time correlation as a SPICE kernel.

In SPICE, the conversion between onboard time (SCET/OBT) and UTC is carried out via a spacecraft clock kernel (sclk). These kernels as produced for Solar Orbiter essentially contain pairs of correlated SCET and UTC times and the time conversion is linearly interpolated between these pairs. SOC will provide two spacecraft clock kernels, a so-called 'fictional' clock kernel for use before launch and for predictive products and the operational clock kernel, which will be based on empirical time correlation data determined at MOC. A reference implementation in Python explaining the use of the spacecraft clock kernel can be found in APPENDIX A.

3.1.1 Fictional Spacecraft Clock Kernel

In order for many of the SPICE APIs to function properly a spacecraft clock kernel needs to be loaded and available, even if a real time correlation is not available. The fictional spacecraft clock kernel fulfils this need and simply contains a 1:1 correlation between on-board coarse time ticks and UTC seconds, with zero seconds defined as 00:00:00 UTC on 1 January 2000 [RD4]. The fictional spacecraft clock kernel will be produced once and has the filename

```
solo_ANC_soc-sclk-fict_20000101_V01.tsc
```

3.1.2 Spacecraft Clock Kernel

The 'real' spacecraft clock kernel will be produced at SOC whenever a new time correlation is published by MOC. Each kernel will cover the entirety of the mission to date (see below). A new time correlation will be published when the instantaneous SCET-UTC conversion drifts by more than 5ms (TBC) from the currently published correlation. MOC time correlations are essentially straight-line segments (UTC = m * SCET + c) that have an earliest valid time and are considered to be valid until the next segment starts. This means that the MOC time correlation is discontinuous, which is not allowed by SPICE. As such, when producing the SPICE kernel, SOC will insert an extra segment, joining two MOC segments together. This is illustrated in **Error! Reference source not found.**. dopting this approach, made necessary by using SPICE, means that for short periods near the start of a new MOC time correlation (~60 s) SGS and MOC time correlations will be slightly different. The same technique has been used successfully for Rosetta.

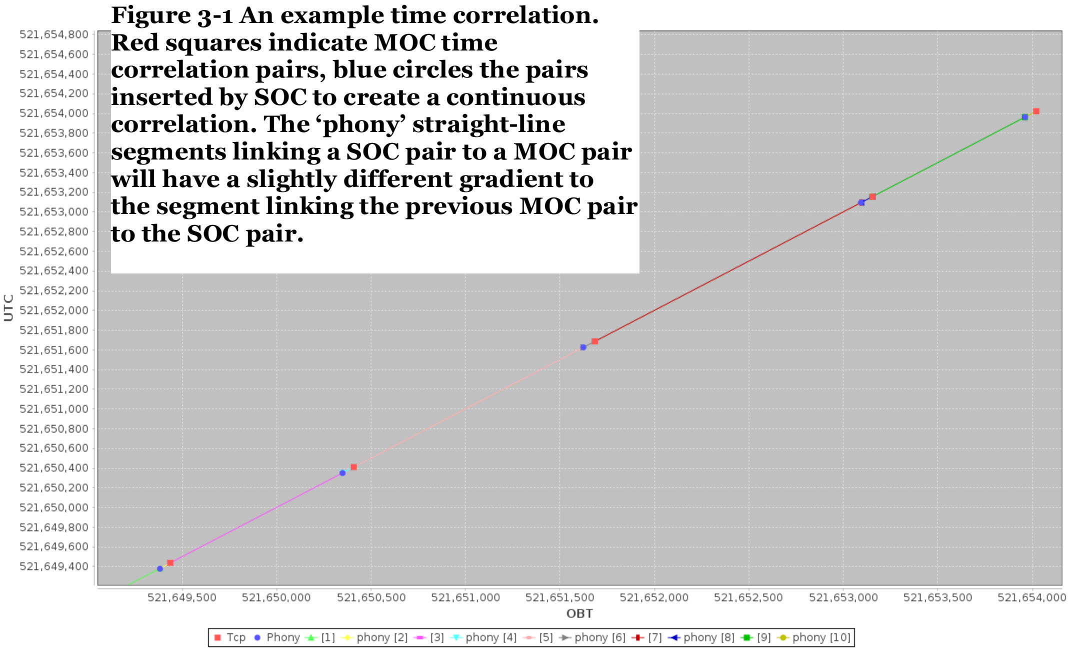

Figure 3-1 An example time correlation. Red squares indicate MOC time correlation pairs, blue circles the pairs inserted by SOC to create a continuous correlation. The 'phony' straight-line segments linking a SOC pair to a MOC pair will have a slightly different gradient to the segment linking the previous MOC pair to the SOC pair.

The spacecraft clock kernel will be named according to [AD1] as follows:

```
solo_ANC_soc-sclk_YYYYMMDD_V01.tsc
```

Here YYYYMMDD represents the the most recent time correlation pair in the file.

While SCET-UTC conversions for times after the most recent time correlation pair in the file are possible, for planning purposes these ought to be avoided. The relationship between SCET and UTC can change with new information (a new leap second kernel or new time-correlation data). Thus only historical conversions are stable. There should be no planning need to predict the relationship of OBT to UTC into the future, so the fictional clock kernel can be used when SPICE requires a sclk. It is recommended that any science data be reprocessed once a stable time correlation for their acquisition time is available.

3.2 Spacecraft Orbit Products

SOC will produce a single orbit SPICE kernel (spk) from ESOC-provided OEM files for the entire mission, unless the required time resolution makes this unmanageable. This will be updated whenever ESOC produces a new OEM file. Since Solar Orbiter's trajectory is essentially ballistic it is expected that this will be after launch and subsequently after each GAM. The spacecraft state vector (position and velocity) can be obtained from the spk at an arbitrary time resolution in any coordinate system known to SPICE.

The spk will be named as follows:

```
solo_ANC_soc-orbit_YYYYMMDD-YYYYMMDD_VOEM_VNN.bsp
```

Here YYYYMMDD-YYYYMMDD refers to the start and end validity of the file, VOEM a reference to the version of the OEM file used to create the kernel and VNN the version number of the kernel produced *from that source OEM*.

VOEM is formatted as follows:

```
L<xxx>_V<y>_<nnnnn>
```

Where <xxx> represents the LTP cycle number, <y> the version of the file and <nnnnn> is a counter. Thus the complete spk file name will be as follows:

```
solo_ANC_soc-orbit_YYYYMMDD-YYYYMMDD_L<xxx>_V<y>_<nnnnn>_VNN.bsp
```

Note that pre-launch orbit kernels derived from mission analysis OEM files omit the VOEM field.

3.3 Spacecraft Attitude Products

SOC will provide three different spacecraft attitude products, two as-planned attitude kernels and one as-flown kernel. An attitude SPICE kernel is known as a 'ck'.

3.3.1 As-Planned Attitude Kernels

The two predicted attitude kernels produced by SOC assume that the spacecraft will be pointed at the centre of the solar disk.

The first as-planned attitude kernel will cover the entire length of the mission and will also assume that the spacecraft will have its default roll-angle. This kernel will have 1-hour resolution and will be updated with the same frequency as the orbit kernel. It will be named as follows:

solo_ANC_soc-default-att_YYYYMMDD-YYYYMMDD_VOEM_VNN.bc

Here YYYYMMDD-YYYYMMDD, VOEM and VNN have the same meanings as for the orbit file..

The second as-planned attitude kernel will assume disk-centre pointing but will also include spacecraft rolls for communications or calibration purposes. There will be one file for each three month planning period, based on the EFECS. They will be distributed approximately 3 months in advance of the start of that planning period, once the long term planning is complete. They will also be updated whenever the orbit kernel is updated. These kernels will nominally have 5-minute time resolution and will be named as follows

solo_ANC_soc-pred-roll_YYYYMMDD-YYYYMMDD_VOEM_VEFECS_VNN.bc

Here YYYYMMDD-YYYYMMDD, VOEM and VNN have the same meanings as for the orbit file, VEFECS identifies the source EFECS file from which the roll information was taken and is of the form MXX_VYY where XX represents the LTP planning cycle number and YY the version of the EFECS for that planning cycle.

The kernel will have coverage from the start of the planning period to the end of the coverage of the orbit kernel it is based upon

Rolls will be handled according to the following rules:

- A new data point will be inserted at the time corresponding to the start of any roll event with the attitude at the start of the roll.
- A new data point will be inserted at the time corresponding to the end of any roll event with the attitude at the end of the roll.
- In the case that the duration of the roll is less than five minutes, no further datapoints will be inserted.
- In the case that the duration of the roll is greater than five minutes, intermediate data points will be inserted at the usual five minute intervals (00:00, 00:05...), assuming a constant roll rate across the duration of the event.
- This is illustrated graphically below, for a 2 minute roll commencing at 12:06 and a fifteen minute roll commencing at 12:27.
- In the case that the spacecraft has non-nominal roll angle at the end of the planning period, it is assumed that roll angle persists into the future, i.e. it remains valid until the first roll event in a subsequent EFECS

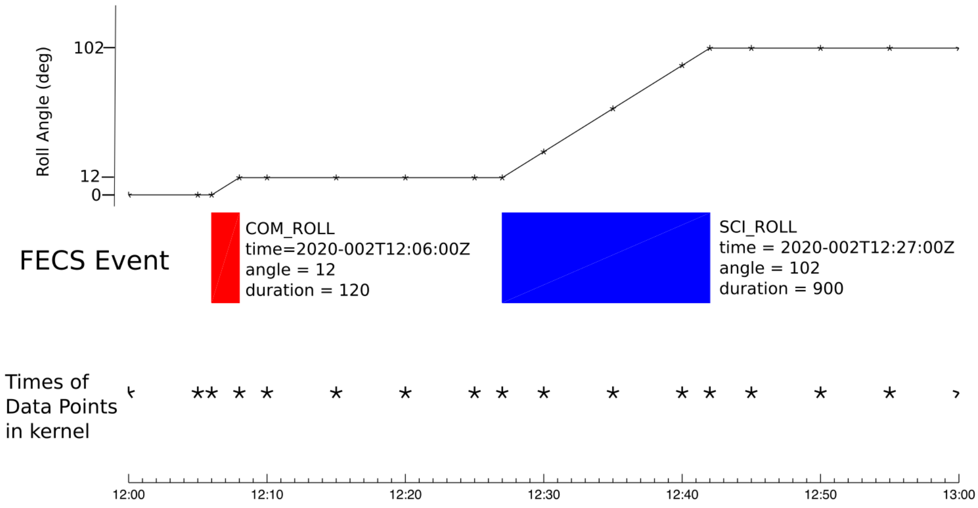

Figure 3-2 An example of how rolls will be handled in the predictive roll attitude kernel.

The third predictive attitude kernel will be based on Attitude Ephemeris Message files received by SOC from flight dynamics. This will include the full commanded attitude of the spacecraft, including rolls. This kernel will be produced once per STP when the input is provided by MOC. This is usually approximately weekly. Each kernel will include the predicted attitude for the entire mission to the end of the relevant STP and will be named as follows:

```
solo_ANC_soc-pred-att_YYYYMMDD-YYYYMMDD_SXXX_VY_ZZZZZ_VNN.bc
```

Here, YYYYMMDD-YYYYMMDD represents the start and end dates of coverage, SXXX_VY_ZZZZZ is inherited from the version of the AEM file, where XXX is the STP number of the latest STP included in the file, Y is the version of the AEM and ZZZZZ is a counter. NN represented the version of the kernel *produced from that AEM*.

3.3.2 As-Flown Attitude Kernel

SOC produce full as-flown attitude information based on quaternions returned in housekeeping data from the AOCS. SOC will produce the as-flown attitude kernel from these quaternions and will have their native time resolution, 300 s (Details TBC). The as-flown attitude kernel will be produced daily and will have the following file name:

```
solo_ANC_soc-flown-att_YYYYMMDDTHHMMSS-YYYYMMDDTHHMMSS_VNN.bc
```

Here YYYYMMDDTHHMMSS-YYYYMMDDTHHMMSS represents the coverage of the file, i.e. the earliest and latest quaternions downloaded as HK in a given communications pass. Since attitude TM will be downloaded from both the SSMM and OMM in a given pass, there may be some discontinuities in coverage of a single file, although nominally any missing data will be included in the next file, which would then overlap in coverage (this will be reflected in the file name).

Note some early as flown attitude kernels were named with only day resolution in the filenames.

Similarly to the predictive roll kernel, described above, additional data points will be introduced at the start times of manoeuvres in the as-flown kernel, although this will by necessity involve extrapolating the spacecraft attitude from the most recent real quaternion, based on the contemporaneously active pointing type.

3.3.3 A Note on Attitude Kernels and the Spacecraft Clock

Attitude data are stored internally to the kernels in terms of onboard time, and as such are associated with a particular spacecraft clock kernel. The predictive attitude kernels require a spacecraft clock kernel to be specified in their production since their definition is based on the spacecraft position and velocity, which is stored relative to ephemeris time in the orbit kernel.

Although the spacecraft clock kernels extrapolate into the future, and as such the predictive attitude kernels could be associated with the real spacecraft clock, this would necessitate their regeneration whenever the time correlation is updated. Instead we will produce the predictive attitude kernels based on the fictional clock, since any use cases involving them do not require knowledge of on board time.

SOC will maintain separate metakernels (see Section 3.5) for the predictive and as-flown datasets to aid switching between them.

3.4 Coordinate System and Reference Frame Products

SOC will produce SPICE kernels with instrument and spacecraft coordinate system and reference frame information, as well as scientific reference frame information.

3.4.1 Scientific Reference Frames Kernel

Relevant scientific reference frames will be provided as a text-based fk file that will be named as follows:

```
solo_ANC_soc-sci-fk_VNN.tf
```

And will contain the reference frames listed in the table below, where the 'Frame Id' column gives the string that can be passed to SPICE calls to use the frame (e.g. SOLO_HCI). More details of each frame can be found in the kernel itself. Note that the table reflects V08 of the kernel.

Frame Id	Frame Name	Description

SOLO_SUN_RTN	Solar Orbiter/Sun Radial Tangential Normal	Origin: Solar Orbiter X: Sun-Spacecraft direction, positive antisunward. Y: Completes Right Handed Set Z: Projection of Solar North on plane perpendicular to X Notes: Geometric (i.e. no aberration corrections applied).
SOLO_SOLAR_MHP	S/C centred mirror helioprojective	Origin: Solar Orbiter X: Completes Right Handed Set (positive towards solar East) Y: Projection of Solar rotation axis on plane perpendicular to Z, positive North. Z: Parallel to the S/C-Sun apparent direction. Positive sunward. Notes: Corrected for light time and stellar aberration.
SOLO_IAU_SUN_2009	Sun Body-Fixed based on IAU 2009 report	Origin: Sun centre of mass X: Aligned with the ascending node of the Sun's orbital plane (around the solar system barycentre) on the solar equator plane Y: Completes the right-handed set Z: Parallel to the Sun's rotation axis, pointing toward the North side of the invariable plane Notes: Equivalent to IAU_SUN when using pck00009.tpc and pck00010.tpc
SOLO_IAU_SUN_2003	Sun Body-Fixed based on IAU 2003 report	Origin: Sun centre of mass X: Aligned with the ascending node of the Sun's orbital plane

		(around the solar system barycentre) on the solar equator plane Y: Completes the right-handed set Z: Parallel to the Sun's rotation axis, pointing toward the North side of the invariable plane. Notes: Equivalent to IAU_SUN when using pck00008.tpc
SOLO_GAE	Geocentric Aries Ecliptic at J2000	Origin: Earth centre of mass X: Positive towards the first point of Aries, i.e. the intersection of the Earth's mean orbit plane with the Earth's mean equatorial plane. Y: Completes the right handed set. Z: Towards to the ecliptic north pole. Notes: Uses ecliptic of J2000
SOLO_GSE	Geocentric Solar Ecliptic at J2000	Origin: Earth centre of mass X: Position of the Sun relative to the Earth, positive sunward. Y: Completes the right-handed set. Z: Towards ecliptic north. Notes: All vectors are geometric. Uses the ecliptic of J2000.
SOLO_HEE	Heliocentric Earth Ecliptic at J2000	Origin: Sun centre of mass X: Position of the Earth relative to the Sun, positive Earthward. Y: Completes the right-handed set. Z: Towards the ecliptic North pole. Notes: All vectors are geometric. Uses ecliptic of J2000

SOLO_ECLIPDATE	Mean Ecliptic of Date Frame	Origin: Earth centre of mass X: Positive towards the first point of Aries for the mean ecliptic of date. Y: Completes the right-handed set. Z: Positive towards ecliptic North. Notes: All vectors are geometric.
SOLO_HCI	Heliocentric Inertial Frame	Origin: Sun centre of mass X: Ascending node on the ecliptic of J2000 of the solar equator Y: Completes the right-handed set Z: Positive towards solar noth (Solar rotation axis of J2000) Notes: All vectors are geometric
SOLO_HEE_NASA	Heliocentric Earth Ecliptic Frame	Origin: Sun centre of mass X: Points from the Sun to the Earth Y: Completes the right-handed set. Z: The component orthogonal to the +X axis of the northern surface normal to the mean ecliptic of date Notes: All vectors are geometric. Similar to SOLO_HEE but for the ecliptic of date not the ecliptic of J2000.
SOLO_HEEQ	Heliocentric Earth Equatorial Frame	Origin: Sun centre of mass X: The component of the position vector of the Earth relative to the Sun that is orthogonal to +Z. Y: Completes the right-handed set. Z: Positive towards Solar North Notes: All vectors are geometric.

SOLO_GEORTN	Geocentric Radial Tangential Normal Frame	Origin: Earth centre of mass X: Sun-Earth direction, positive antisunward. Y: Completes the right-handed set. Z: The component of the Solar North direction perpendicular to +X Notes: All vectors are geometric.
SUN_ARIES_ECL*	Heliocentric Aries Ecliptic	Origin: Sun centre of mass X: Positive towards the first point of Aries. Y: Completes the right-handed set. Z: Aligned with the north-pointing vector normal to the mean orbital plane of the Earth.
SUN_EARTH_ECL*	Heliocentric Earth Equatorial	Origin: Sun centre of mass X: Component of the Sun-Earth vector orthogonal to Z. Y: Completes the right-handed set. Z: Normal vector to the ecliptic plane that points toward the north pole of date. Notes: Equivalent to SOLO_HEE_NASA
SUN_EARTH_CEQU*	Heliocentric Earth Ecliptic	Origin: Sun centre of mass X: Positive towards the component of the Sun-Earth vector that is orthogonal to the +Z axis. Y: Completes the right-handed set. Z: Positive towards the Sun's north pole of date. Notes: All vectors are geometric. This frame is equivalent to SOLO_HEEQ.

SUN_INERTIAL*	Heliocentric Inertial	Origin: Sun centre of mass X: Positive towards the ascending node of the Sun's equatorial plane on the ecliptic plane of J2000 Y: Completes the right-handed set Z: Parallel to the Sun's rotation axis of epoch J2000, pointing toward the Sun's north pole Notes: Equivalent to SOLO_HCI.
EARTH_SUN_ECL*	Geocentric Solar Ecliptic	Origin: Earth centre of mass X: Component of the Earth-Sun vector that is orthogonal to +Z Y: Completes the right-handed set Z: The north ecliptic pole of date. Notes: All vectors are geometric. This frame is similar to SOLO_GSE but uses to the ecliptic of date, not the ecliptic of J2000.
EARTH_MECL_MEQX*	Earth Mean Ecliptic and Equinox of date frame	Origin: Earth centre of mass X: Positive towards the mean equinox, which is defined as the intersection of the Earth's mean orbital plane with the Earth's mean equatorial plane. Y: Completes the right-handed set. Z: Positive along the north-pointing vector normal to the mean orbital plane of the Earth Notes: Equivalent to SOLO_ECLIPDATE.

SOLO_VSO	Venus Solar Orbital frame Introduced in V0	Origin: Venus centre of mass X: Positive towards the Sun Y: Completes the right-handed set. Z: Positive along the north-pointing vector normal to the mean orbital plane of the Earth
SOLO_GSM	Geocentric Solar Magnetic/Magnetospheric	Origin: Earth centre of mass X: Positive towards the Sun Y: Completes the right-handed set. Z: Positive northward in the plane containing X and the north geomagnetic pole Note that this is a ck-based frame that uses a time-varying position of the north geomagnetic pole derived from IGRF coefficients. The initial version has coverage up to 2025. As IGRF is updated, so too will the frame coverage.

Note that frames marked with a * may in the future be incorporated into the generic frames kernel distributed by NAIF. If so, they will be removed from the Solar Orbiter science frames kernel.

3.4.2 Spacecraft and Instrument Reference Frames

Spacecraft and instrument reference frames will also be provided through a variety of different kernel files, all frames are defined in text-based *fk* files as for the scientific coordinate systems. This is all that is required for some frames (so-called “fixed” and “dynamic” frames), while others will also need ck files, similar to the attitude kernels, because they are time-varying in nature . Note that, similarly for the attitude kernels (also cks) there will be two versions of each frame ck: one produced using the fictional spacecraft clock and one using the real spacecraft clock.

The basic frame hierarchy is explained diagrammatically below, and in complete detail in the top-level frame kernel:

```
solo_ANC_soc-sc-fk_V01.tf
```

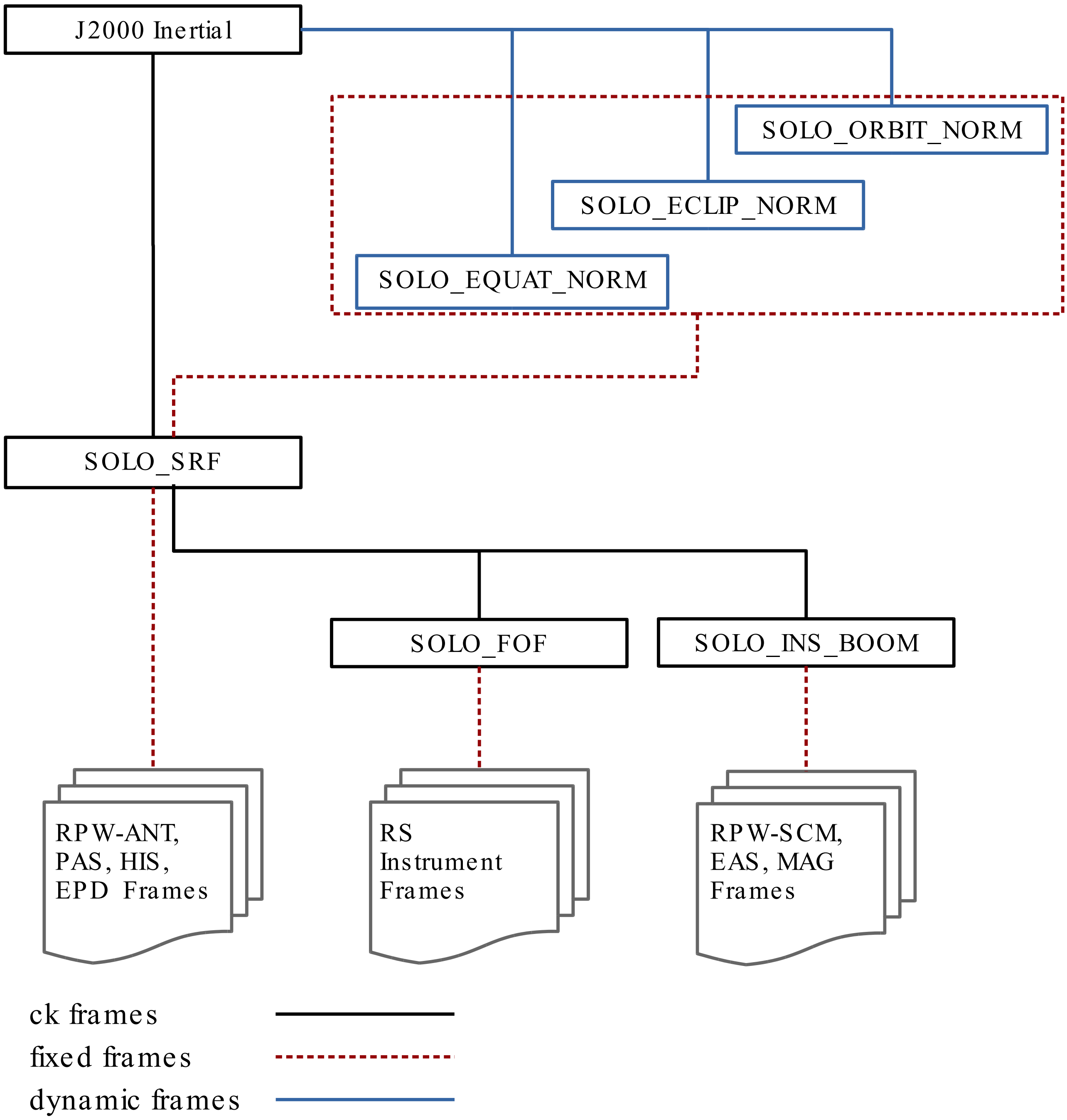

Figure 3-3 Simplified spacecraft and instrument frames hierarchy

Other frames (e.g. star tracker, HGA frames) are included in the file for completeness however these are not described here.

3.4.2.1 SOLO_SRF – "Spacecraft Coordinates"

SOLO_SRF (Solar Orbiter Spacecraft Reference Frame) is the name of the reference frame that is commonly referred to as spacecraft coordinates and is defined in [RD9]. It is a Cartesian frame centred on the point of intersection of the launcher longitudinal axis with the separation plane between the launcher and the composite, and has the following components:

- X points along the longitudinal axis of the spacecraft, positive towards the heat shield.
- Z points perpendicular to the launcher interface plane, positive towards the face of the spacecraft containing the MGA and one of the RPW antennae.
- Y completes the right handed set.

By default, assuming no off-points or rolls, SOLO_SRF_X will point towards the centre of the Sun; SOLO_SRF_Z will point normal to the plane of the orbit, positive towards solar north; SOLO_SRF_Y will complete the right-handed set, opposing the direction of motion of the spacecraft.

That said, the relationship between SOLO_SRF and J2000 can be specified in different ways depending on the desired application. **For most users, the link should be made by loading one of the spacecraft attitude kernels (see Section 3.3).** However, it is also possible to explore the characteristics of various possible default attitudes (see, for example [RD8]) through linking any of

SOLO_ORBIT_NORM
SOLO_ECLIP_NORM
SOLO_EQUAT_NORM

To SOLO_SRF.

SOLO_ORBIT_NORM represents the default attitude defined above. In the case of SOLO_ECLIP_NORM, SOLO_SRF_Z is held perpendicular to the plane of the ecliptic. In the case of SOLO_EQUAT_NORM, SOLO_SRF_Z is held perpendicular to the heliographic equatorial plane.

These three frames are defined in

```
solo_ANC_soc-ops-fk_V01.tf
```

Full instructions for their use can be found in that file.

3.4.2.2 SOLO_FOF and SOLO_INS_BOOM – "Alignment frames"

SOLO_FOF, the "flight optical frame" and SOLO_INS_BOOM the instrument boom frame are both ck-based frames that can vary in time and have been introduced so that any systematic misalignments of the remote sensing instruments (for SOLO_FOF) or imperfect deployment of the instrument boom[3] (SOLO_INS_BOOM) can be taken into account in a consistent way without the need to update several instrument frames directly.

The transformation between SOLO_SRF and SOLO_FOF will be the identity matrix at launch and will be updated in flight should a systematic misalignment of RS instrument bore sights from nominal be detected. Similarly, SOLO_INS_BOOM is currently as defined in [RD9] and will be updated if data analysis reveals the true orientation of the instrument boom post-deployment.

Note that these frames are designed as coarse adjustments primarily to be used internally by SOC in operations. Instrument teams are much better placed to determine fine misalignments of their instruments to higher accuracy, and take these into account in the production of their science data, but of course these can be used as a starting point.

3.4.2.3 Instrument Frames

A set of reference frames for each instrument, based on their respective EID-Bs, have also been defined. These are documented within the frames kernel so are not discussed in here, apart from the note that they are fixed relative to the most relevant of SOLO_SRF, SOLO_FOF and SOLO_INS_BOOM: SOLO_SRF for body-mounted in situ instruments (e.g. the EPD sensors, RPW-ANT); SOLO_FOF for remote sensing instruments mounted on the main optical bench and SOLO_INS_BOOM for boom-mounted instruments (e.g. MAG, SWA-EAS).

3.4.2.4 Instrument Fields-of-View

SOC will also produce simple instrument kernels containing field of view information only for use in SOC operational tools. It is expected that these remain will fixed throughout the mission. They should not become repositories of information about the instruments themselves, as is the case for some planetary missions.

3.5 Metakernels and Versioning

Metakernels are a way of loading a self-consistent dataset with a single command. They are essentially an ASCII file containing a list of other kernels to load, and are how we will manage version control: If a user has access to the entire Solar Orbiter SPICE kernel dataset and loads the most recent metakernel, they will automatically be using the most recent versions of each kernel.

---

[3] It is likely that SOLO_INS_BOOM will be replaced by two frames – one for each boom segment – in the future.

SOC will maintain two metakernels: One for the predictive dataset, and one for the as-flown dataset. This is necessary because of the different spacecraft clocks used in the production of the attitude kernels (Section 3.3.3). By default the metakernels will not map any of SOLO_ORBIT_NORM, SOLO_ECLIP_NORM and SOLO_EQUAT_NORM to SOLO_SRF (Section 3.4.2.1).

The predictive attitude metakernel will be named as follows:

```
solo_ANC_soc-pred-mk_VNNN_YYYYMMDD_XXX.tm
```

The as-flown metakernel will be named as follows:

```
solo_ANC_soc-flown-mk_VNNN_YYYYMMDD_XXX.tm
```

The three last fields in the file relate to versioning:

VNNN indicates the version of the *kernel dataset* (e.g V020). This is incremented whenever there are changes to “rarely produced” kernels. These include all of the ASCII kernels (aside from the metakernel itself) and instrument ck frame kernels that only change when new alignment information is available.

YYYYMMDD indicates the production date of the metakernel. This is incremented whenever a more rapidly varying kernel is updated or newly included in the metakernel, e.g. the spacecraft clock or as-flown attitude kernels.

XXX indicates the version of the metakernel produced *on that day*. This will almost always be 001, but could be incremented should an error in processing occur, or if there are two communications passes on a single calendar day, for example, and hence two updates to the as flown attitude kernels.

SOC will also provide the latest version of each metakernel with an “unversioned” file name, such that software you simply load this metakernel and always have access to the most up to date ancillary data. These are named as follows:

```
solo_ANC_soc-pred-mk.tm

solo_ANC_soc-flown-mk.tm
```

Older versions of the metakernel will also be kept in case users wish to choose an older set of SPICE kernels.

Note that in the case of the predictive metakernel, the default attitude kernel will be loaded first, then the rolls kernels, and finally the predictive attitude kernels, this will have the effect of ensuring the most up to date predictions of attitude are always used. In order to explicitly use default attitude, the science operations frames (SOLO_ORBIT_NORM etc.) can be used.

4 CDF ANCILLARY DATA PRODUCTS

The following describes the ancillary data products that SOC will provide in CDF format. Following the Solar Orbiter Metadata standard [AD1], CDF products will be of CDF version > 3.6 [RD5].

4.1 Spacecraft Orbit Digest

SOC will produce a single ‘orbit digest’ file for the entire mission from the Solar Orbiter SPK, which in turn will be based on the OEM orbit file produced by the Solar Orbiter MOC at ESOC. Since, between GAMs, Solar Orbiter’s orbit is essentially ballistic, it is anticipated that this file will not need to be updated with high cadence, and as such a new version will be nominally be issued after launch and after each GAM.

The orbit digest will follow the Solar Orbiter metadata standard and its filename will have the format:

```
solo_ANC_soc-orbit_YYYYMMDD-YYYYMMDD_VOEM_VNN.cdf
```

Where YYYYMMDD-YYYYMMDD represents the start date and end date of the coverage of the file, VOEM the version of the source OEM and VNN the version of the cdf. This is the same naming convention as for the orbit kernel.

The orbit digest will contain the following parameters at 1-hour resolution:

- HCI Position, XYZ, km.
- HCI Velocity, XYZ, km/s.
- HEE Position XYZ, km.
- Spacecraft Heliocentric distance, km.
- Spacecraft Heliographic latitude, degrees.
- Spacecraft ecliptic latitude, degrees.
- Spacecraft HCI longitude, degrees.
- Spacecraft Stonyhurst Heliographic longitude.

The ecliptic of J2000 will be used in defining the orbit digest file. See APPENDIX B for the definition of the HCI, HEE and Carrington Heliographic coordinate systems. Quantities will be entirely geometric, i.e. no light travel time or stellar aberration corrections will be applied.

4.1.1 *Orbit Digest CDF Variables and Attributes*

The Orbit Digest CDF will contain the following CDF variables and attributes.

Global Attributes

Name	Entry	Value
Project	1	"Solar Orbiter"
Project	2	"Cosmic Visions"
Source_name	1	"SOLO>Solar Orbiter"
Discipline	1	"Space Physics>Interplanetary Studies"
Data_type	1	"ANC>Ancillary Data"
Descriptor	1	"soc_orbit"
Data_version	1	"01"
Software_version	1	"01.00.00"
PI_name	1	"Solar Orbiter SOC"
PI_affiliation	1	"ESA/ESAC"
Text	1	"SOC-Provided Ancillary Data for Solar Orbiter"
Text	2	"SOL-SGS-TN-0017"
Instrument_type	1	"Ephemeris"
Mission_group	1	"Solar Orbiter"
Logical_source	1	"SOLO_ANC_SOC-ORBIT"
Logical_file_id	1	"solo_ANC_soc-orbit_YYYYMMDD-YYYYMMDD_VNN"
Logical_source_description	1	"Solar Orbiter SOC Ancillary Orbit Data"
Rules_of_use	1	"Free to Use"
Generated_by	1	"Solar Orbiter SOC, ESAC"
Generation_date	1	"YYYY-MM-DDTHH:MN:SS"
Mods	1	"V01 First Version"
Data_Product	1	"ORBIT>Orbit Digest"
Level	1	"ANC>Ancillary Data"
Instrument	1	"SOC>Science Operations Centre"
Parents	1	"solo_ANC_soc-orbit_YYYYMMDD-YYYYMMDD_VOEM_VNN"

Variables

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
EPOCH	CDF_TIME_TT2000	1	1	T	F

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	"Time"
CATDESC	CDF_CHAR	"Interval Centred Timetags"
FILLVAL	CDF_INT8	-9223372036854775807
FORMAT	CDF_CHAR	"I"
LABLAXIS	CDF_CHAR	"Time"
UNITS	CDF_CHAR	"nanoseconds"
VALIDMIN	CDF_INT8	-9223372036854775807
VALIDMAX	CDF_INT8	9223372036854775808
SCALEMIN	CDF_INT8	TBC #These will be the start and

SCALEMAX	CDF_INT8	TBC #end of the coverage of the file.
VAR_TYPE	CDF_CHAR	“support_data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
HCI_POS	CDF_REAL8	1	3	T	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HCI Position”
CATDESC	CDF_CHAR	“Spacecraft Position in Heliocentric Inertial Coordinates”
DEPEND_0	CDF_CHAR	“EPOCH”
DISPLAY_TYPE	CDF_CHAR	“time_series”
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	“f15.3”
LABL_PTR_1	CDF_CHAR	“LABL_HCI_POS”
UNITS	CDF_CHAR	“km”
SI_CONVERSION	CDF_CHAR	“1.0e3>m”
VALIDMIN	CDF_REAL8	-9999999999.999
VALIDMAX	CDF_REAL8	9999999999.999
SCALETYP	CDF_CHAR	“linear”
SCALEMIN	CDF_REAL8	-300000000.000
SCALEMAX	CDF_REAL8	-300000000.000
VAR_TYPE	CDF_CHAR	“data”
COORDINATE_SYSTEM	CDF_CHAR	“HCI”
TENSOR_ORDER	CDF_UINT1	1
REPRESENTATION_1	CDF_CHAR	“REP_HCI_POS”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
HCI_VEL	CDF_REAL8	1	3	T	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HCI Velocity”
CATDESC	CDF_CHAR	“Spacecraft Position in Heliocentric Inertial Coordinates”
DEPEND_0	CDF_CHAR	“EPOCH”
DISPLAY_TYPE	CDF_CHAR	“time_series”
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	“f8.3”
LABL_PTR_1	CDF_CHAR	“LABL_PTR_HCI_VEL”
UNITS	CDF_CHAR	“km s^-1”
SI_CONVERSION	CDF_CHAR	“1.0e3>m s^-1”
VALIDMIN	CDF_REAL8	-999.999
VALIDMAX	CDF_REAL8	999.999
SCALETYP	CDF_CHAR	“linear”
SCALEMIN	CDF_REAL8	-100.000
SCALEMAX	CDF_REAL8	100.000

VAR_TYPE	CDF_CHAR	“data”
COORDINATE_SYSTEM	CDF_CHAR	“HCI”
TENSOR_ORDER	CDF_UINT1	1
REPRESENTATION_1	CDF_CHAR	“REP_HCI_VEL”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
HEE_POS	CDF_REAL8	1	3	T	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HEE Position”
CATDESC	CDF_CHAR	“Spacecraft Position in Heliocentric Earth Ecliptic Coordinates”
DEPEND_0	CDF_CHAR	“EPOCH”
DISPLAY_TYPE	CDF_CHAR	“time_series”
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	“f15.3”
LABL_PTR_1	CDF_CHAR	“LABL_HEE_POS”
UNITS	CDF_CHAR	“km”
SI_CONVERSION	CDF_CHAR	“1.0e3>m”
VALIDMIN	CDF_REAL8	-9999999999.999
VALIDMAX	CDF_REAL8	9999999999.999
SCALETYP	CDF_CHAR	“linear”
SCALEMIN	CDF_REAL8	-300000000.000
SCALEMAX	CDF_REAL8	-300000000.000
VAR_TYPE	CDF_CHAR	“data”
COORDINATE_SYSTEM	CDF_CHAR	“HEE”
TENSOR_ORDER	CDF_UINT1	1
REPRESENTATION_1	CDF_CHAR	“REP_HEE_POS”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
HCENTRIC_DIST	CDF_REAL8	1	1	T	F

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“Heliocentric Distance”
CATDESC	CDF_CHAR	“Spacecraft Heliocentric Distance”
DEPEND_0	CDF_CHAR	“EPOCH”
DISPLAY_TYPE	CDF_CHAR	“time_series”
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	“f15.3”
LABLAXIS	CDF_CHAR	“R”
UNITS	CDF_CHAR	“km”
SI_CONVERSION	CDF_CHAR	“1.0e3>m”
VALIDMIN	CDF_REAL8	0
VALIDMAX	CDF_REAL8	9999999999.999
SCALETYP	CDF_CHAR	“linear”
SCALEMIN	CDF_REAL8	0

SCALEMAX	CDF_REAL8	-300000000.000
VAR_TYPE	CDF_CHAR	“data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
HGRAPH_LAT	CDF_REAL8	1	1	T	F

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“Heliographic Latitude”
CATDESC	CDF_CHAR	“Spacecraft Heliographic Latitude”
DEPEND_0	CDF_CHAR	“EPOCH”
DISPLAY_TYPE	CDF_CHAR	“time_series”
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	“f8.4”
LABLAXIS	CDF_CHAR	“H. Lat.”
UNITS	CDF_CHAR	“degrees”
SI_CONVERSION	CDF_CHAR	“0.0174532928>rad”
VALIDMIN	CDF_REAL8	-90.0
VALIDMAX	CDF_REAL8	90.0
SCALETYP	CDF_CHAR	“linear”
SCALEMIN	CDF_REAL8	-45.0
SCALEMAX	CDF_REAL8	45.0
VAR_TYPE	CDF_CHAR	“data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
ECLIP_LAT	CDF_REAL8	1	1	T	F

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“Ecliptic Latitude”
CATDESC	CDF_CHAR	“Spacecraft Ecliptic Latitude”
DEPEND_0	CDF_CHAR	“EPOCH”
DISPLAY_TYPE	CDF_CHAR	“time_series”
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	“f8.4”
LABLAXIS	CDF_CHAR	“H. Lat.”
UNITS	CDF_CHAR	“degrees”
SI_CONVERSION	CDF_CHAR	“0.0174532928>rad”
VALIDMIN	CDF_REAL8	-90.0
VALIDMAX	CDF_REAL8	90.0
SCALETYP	CDF_CHAR	“linear”
SCALEMIN	CDF_REAL8	-50.0
SCALEMAX	CDF_REAL8	50.0
VAR_TYPE	CDF_CHAR	“data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY

HCI_LON	CDF_REAL8	1	1	T	F

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	"HCI Longitude"
CATDESC	CDF_CHAR	"Spacecraft Heliocentric Inertial Longitude"
DEPEND_0	CDF_CHAR	"EPOCH"
DISPLAY_TYPE	CDF_CHAR	"time_series"
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	"f8.4"
LABLAXIS	CDF_CHAR	"HCI. Lon."
UNITS	CDF_CHAR	"degrees"
SI_CONVERSION	CDF_CHAR	"0.0174532928>rad"
VALIDMIN	CDF_REAL8	-180.0
VALIDMAX	CDF_REAL8	180.0
SCALETYP	CDF_CHAR	"linear"
SCALEMIN	CDF_REAL8	-180.0
SCALEMAX	CDF_REAL8	180.0
VAR_TYPE	CDF_CHAR	"data"

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
SHURST_LON	CDF_REAL8	1	1	T	F

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	"Stonyhurst Longitude"
CATDESC	CDF_CHAR	"Spacecraft Stonyhurst Longitude"
DEPEND_0	CDF_CHAR	"EPOCH"
DISPLAY_TYPE	CDF_CHAR	"time_series"
FILLVAL	CDF_REAL8	-1.0e31
FORMAT	CDF_CHAR	"f8.4"
LABLAXIS	CDF_CHAR	"Shurst. Lon."
UNITS	CDF_CHAR	"degrees"
SI_CONVERSION	CDF_CHAR	"0.0174532928>rad"
VALIDMIN	CDF_REAL8	-180.0
VALIDMAX	CDF_REAL8	180.0
SCALETYP	CDF_CHAR	"linear"
SCALEMIN	CDF_REAL8	-180.0
SCALEMAX	CDF_REAL8	180.0
VAR_TYPE	CDF_CHAR	"data"

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
LABL_HCI_POS	CDF_CHAR	1	3	F	T

ATTRIBUTES

Attribute_Name	Data_type	Value

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HCI POS Axis Label”
CATDESC	CDF_CHAR	“Axis Label for Spacecraft HCI Position”
FORMAT	CDF_CHAR	“A5”
VAR_TYPE	CDF_CHAR	“support_data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
REP_HCI_POS	CDF_CHAR	1	3	F	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HCI POS Representation”
CATDESC	CDF_CHAR	“Vector Representation for Spacecraft HCI Position”
FORMAT	CDF_CHAR	“A1”
VAR_TYPE	CDF_CHAR	“support_data”

LABL_HCI_VEL	CDF_CHAR	1	3	F	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HCI VEL Axis Label”
CATDESC	CDF_CHAR	“Axis Label for Spacecraft HCI Velocity”
FORMAT	CDF_CHAR	“A6”
VAR_TYPE	CDF_CHAR	“support_data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
REP_HCI_VEL	CDF_CHAR	1	3	F	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HCI VEL Representation”
CATDESC	CDF_CHAR	“Vector Representation for Spacecraft HCI Velocity”
FORMAT	CDF_CHAR	“A1”
VAR_TYPE	CDF_CHAR	“support_data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
LABL_HEE_POS	CDF_CHAR	1	3	F	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HEE POS Axis Label”
CATDESC	CDF_CHAR	“Axis Label for Spacecraft HEE Position”
FORMAT	CDF_CHAR	“A5”
VAR_TYPE	CDF_CHAR	“support_data”

Variable_Name	Data_type	DIMS	SIZES	R_VARY	D_VARY
REP_HEE_POS	CDF_CHAR	1	3	F	T

ATTRIBUTES

Attribute_Name	Data_type	Value
FIELDNAM	CDF_CHAR	“HEE POS Representation”
CATDESC	CDF_CHAR	“Vector Representation for Spacecraft HEE Position”
FORMAT	CDF_CHAR	“A1”
VAR_TYPE	CDF_CHAR	“support_data”

4.2 Spacecraft Attitude Products

SOC will produce two attitude products in CDF format. These are intended to provide simple information about the predicted and as-flown spacecraft roll angle. Here rolls are around spacecraft X and roll angle is defined as the angle between the spacecraft Z-axis and the zero roll reference. The zero roll reference is currently spacecraft Z parallel to the normal of the spacecraft orbital plane. If the spacecraft baseline attitude is not changed such that the spacecraft XY plane is held parallel to the ecliptic or solar equatorial plane [RD8] we will add the projection of solar north on the spacecraft YZ plane to the roll digest files (TBC). Rolls are defined as positive around spacecraft +X. Spacecraft offpoints will not be included in the CDF attitude products since these will nominally be less than 1 degree, and LL02 data and L2 science data should have taken already these into account. For applications that require detailed pointing information, the SPICE kernels should be used. If possible, a quality flag showing periods of non-nominal pointing (i.e. contingency cases) will be included in the as-flown roll angle file (TBC).

4.2.1 Predicted Roll Angle File

The predicted roll angle file will be issued per medium term planning period (~6 months: Jan-Jun, Jul-Dec), approximately 6 months in advance of the beginning of that planning period. It will contain the expected spacecraft roll angle that results from the long-term planning process. It will be produced based on the predicted roll angle CK, which is itself produced from the FECS issued by the Solar Orbiter MOC at ESOC. It will have 5-minute resolution.

The predicted roll angle file will follow the Solar Orbiter metadata standard and will be named as follows:

```
solo_ANC_soc-pred-roll_YYYYMMDD-YYYYMMDD_V01.cdf
```

Where YYYYMMDD-YYYYMMDD represents the start date and end date of the coverage of the file. This is equivalent to the predicted roll SPICE kernel.

4.2.1.1 Predicted Roll Angle CDF Variables and Attributes

TBW

4.2.2 As-Flown Roll Angle File

One as-flown roll angle file will be issued per calendar day, and will be produced a few hours after the end of each communications pass. It will contain the actual spacecraft roll angle that was flown. It will be produced based on the flown-attitude CK, which is itself produced from spacecraft housekeeping. It will have the native resolution of the housekeeping data from which it is derived, 60-seconds (TBC).

The as-flown roll angle file will follow the Solar Orbiter metadata standard and will be named as follows:

```
solo_ANC_soc-flown-roll_YYYYMMDD_V01.cdf
```

Where YYYYMMDD represents the date of the coverage of the file.

4.2.2.1 As-Flown Roll Angle CDF Variables and Attributes

TBW

5 DISTRIBUTION OF ANCILLARY DATA PRODUCTS

The primary distribution channel of SOC ancillary products to the instrument teams will be via GFTS [RD6], whereby as files are produced they will be pushed from SOC to the GFTS nodes located at instrument team premises. The files will be distributed via GFTS first and the most recent files will always be available via this mechanism. As such, any operational workflows that require SOC ancillary data should be designed with this in mind.

SPICE kernels will be distributed in zipped archives, either as a complete set, named

```
solo_ANC_soc-kernels-complete_Vxxx_YYYYMMDD_nnn.zip
```

or a so-called "delta" distribution, containing only files that have been newly produced since the last delivery of kernels, as well as a complete set of metakernels. These are named

```
solo_ANC_soc-kernels-delta_Vxxx_YYYYMMDD_nnn.zip
```

Here Vxxx refers to the SPICE kernel dataset version, YYYYMMDD the date of production of the zip and nnn a counter signifying how many times on a given day the zip has been produced.

The complete kernel set is expected to be produced and distributed at least once per planning period when an updated predictive roll kernel and/or orbit is available. The delta set will be primarily used to distribute the latest as-flown attitude and time correlation, as well as updated metakernels, after each communications pass. During the early part of the mission, however, instead of the delta set, the complete set may be distributed after each pass, until the size of the complete set becomes too large for this to be practical.

Both archives will contain the directory "kernels" and all necessary subdirectories ("spk", "ck“ etc.) depending on the contents of the archive.

A complete archive can safely be used to replace the entirety of an existing kernel set.

In the case of the delta archive, kernels/mk should *replace* your existing metakernel subdirectory since we always distribute the complete metakernel set. The contents of any other subdirectories should be *added* to your existing subdirectories since they will only contain the newly-produced files.

Solar Orbiter SPICE kernels will also be made available to the broader scientific community via the ESA SPICE Service SFTP server, although not necessarily immediately after their production, and will be usable via the ESAC WebGeoCalc instance.

All SOC ancillary data products will also be archived and publicly available in the SOAR.

APPENDIX A

The code sample provided here is a reference implementation for the use of the spacecraft clock kernel written in Python - specifically, Python 2.7. The most recent version of this reference implementation can be found at https://issues.cosmos.esa.int/solarorbiterwiki/x/bwXz. NAIF does not provide an official Python API for the toolkit, so here we make use of the *spiceypy* library. This library can be found at https://pypi.python.org/pypi/spiceypy.

We recommend the use of Anaconda data science platform (https://www.continuum.io/downloads), which will allow you to easily create a Python 2.7 environment with the needed libraries (*spiceypy* in this case).

To perform the time translation two SPICE kernels are needed:

- Leap seconds kernel: currently *naif0012.tls*, which can be found here.

  The file is provided by NAIF via the ESA SPICE service: ftp://spiftp.esac.esa.int/data/SPICE/SOLAR-ORBITER/kernels/lsk and will be occasionally updated when new leap seconds are known.

- Spacecraft clock kernel: solo_ANC_soc-sclk_20000101_20160712_V01.tsc

  This is file is provided by the Solar Orbiter SOC and it will be frequently updated. The version linked above is the one used currently for SOC testing

For both the above kernels, it is instrument team responsibility to ensure provision of the up-to-date kernels to the software.

Python Code

```
import spiceypy

# Provided by Solar Orbiter SOC
# Version: 1.0
# Date: 09-Aug-2016
class SpiceManager:

    # SOLAR ORBITER naif identifier
    solar_orbiter_naif_id = -144

    def __init__(self, tls_filename, sclk_filename):
       spiceypy.furnsh(tls_filename)
       spiceypy.furnsh(sclk_filename)

    def obt2utc(self, obt_string):
        # Obt to Ephemeris time (seconds past J2000)
        ephemeris_time = spiceypy.scs2e(self.solar_orbiter_naif_id, obt_string)
        # Ephemeris time to Utc
        # Format of output epoch: ISOC (ISO Calendar format, UTC)
        # Digits of precision in fractional seconds: 3
        return spiceypy.et2utc(ephemeris_time, "ISOC", 3)

    def utc2obt(self, utc_string):
        # Utc to Ephemeris time (seconds past J2000)
        ephemeris_time = spiceypy.utc2et(utc_string)
        # Ephemeris time to Obt
        return spiceypy.sce2s(self.solar_orbiter_naif_id,ephemeris_time)
```

Examples Using the Above Code

```
# Load spice kernels files
tls_filename = "<<your_path>>\naif0012.tls"
sclk_filename = "<<your_path>>\solo_ANC_soc-sclk_20000101_20160712_V01.tsc"

# Initialise class
spicemanager = SpiceManager(tls_filename, sclk_filename)

# Execute conversions

obt = "0"
utc = "2000-01-01T00:00:00.000"
print "OBT {0} -> UTC {1}".format(obt, spicemanager.obt2utc(obt))
# Returns OBT 0 -> UTC 2000-01-01T00:00:00.000
print "UTC {0} -> OBT {1}".format(utc, spicemanager.utc2obt(utc))
# Returns UTC 2000-01-01T00:00:00.000 -> OBT 1/0000000000:00000

obt = "521651623:37539"
utc = "2016-194T15:13:46.381"
print "OBT {0} -> UTC {1}".format(obt, spicemanager.obt2utc(obt))
# Returns: OBT 521651623:37539 -> UTC 2016-07-12T15:13:46.381
print "UTC {0} -> OBT {1}".format(utc, spicemanager.utc2obt(utc))
# Returns: UTC 2016-194T15:13:46.381 -> OBT 1/0521651623:3753
```