esoc

European Space Operations Centre
Robert-Bosch-Strasse 5
D-64293 Darmstadt
Germany
T +49 (0)6151 900
F +49 (0)6151 90495
www.esa.int

DOCUMENT

Solar Orbiter Planning Interface Control Document

Prepared by **L. Michienzi & D. Werner**
Reference **SOL-ESC-IF-05010**
Issue **1**
Revision **2**
Date of Issue **04/09/2017**

European Space Agency
Agence spatiale européenne

APPROVAL

Prepared by: _____

Digitally signed by Luca Michienzi
DN: postalCode=75738, o=European Space Agency, street=8-10, Rue Mario Nikis, st=Cedex 15, l=Paris, c=FR, cn=Luca Michienzi, email=luca.michienzi@esa.int
Date: 2017.09.04 13:05:02 +02'00'

L. Michienzi (OPS-GDS)
Solar Orbiter Data Systems Engineer

Approved by: _____

Sylvain Lodiot
2017.09.04 15:17:37 +02'00'

S. Lodiot (OPS-OPS)
Solar Orbiter Spacecraft Operations Manager

Approved by: _____

Digitally signed by Mathias Lauer
Date: 2017.09.05 11:51:03 +02'00'

M. Lauer (OPS-GFS)
Solar Orbiter Flight Dynamics Manager

Approved by: _____

A. Accomazzo (OPS-OP)
Solar Orbiter Ground Segment Manager

Approved by: _____

D. Mueller (SRE-S)
Solar Orbiter Project Scientist

Approved by: _____

L. Sanchez (SRE-ODS)
Solar Orbiter SGS Development Manager

CHANGE LOG

Reason for change	Issue	Revision	Date
First issue for external review	1	0	21/05/2013
Numerous updates (see change record below)	1	2	04/09/2017

CHANGE RECORD

Date	Iss. No.	Rev. No.	Description and Section No.
21/05/2013	1	0	First issue for external review
	1	1	*Not formally released. Description merged within Issue 1.2 below.*
04/09/2017	1	2	Update of distribution list, reference documents and signature page Minor reformatting and editorial corrections, including renaming of HSO- to OPS-. Section 1.2: Clarification about the attached schema files and their relationship to this document's details Section 1.3.1: Added the MPS Planning Database ICD as applicable document Section 1.4: Updated list of Acronyms Section 3.2.1.1: Added type attribute to validityRange element, corrected editorials to avoid confusion and added an example. Section 3.2.1.2: - Changed passId from Mandatory to Optional - Updated instrument source list - Changed insertOrDeleteFlag from Mandatory to Optional - Added parameterList to the occurrence element table and listed the possible occurrence elements in the introduction of the section. - Added iVSTP complex element Section 3.2.1.2.1: - replaced the parameter type name (crfParameter) with the parameter element name (parameter) to avoid confusion with respect to the file format. - Clarified that CRF command and sequence occurrences shall list 'editable' parameters (excluding non-editable parameters). - Modified the definition of occurrence parameter list "count" - Clarified that for Octet Strings, the radix must be 'Hexadecimal'.

Date	Iss. No.	Rev. No.	Description and Section No.
			Section 3.2.2: replaced the occurrence type names with occurrence element names, including the Table 1, to avoid confusion Section 3.2.3.1: corrected left over from PF-ICD. For Solar-Orbiter the release time cannot be an absolute time. Section 3.2.4: - Removed confusing statement about uniqueID - Changed uniqueID last 16 characters to consist of alpha-numeric characters instead of numeric counter only. - Update Instruments/Source Rule and corrected the instrument source acronyms for EPD (SEPD), SOLOHI (SSHI) and STIX (SSTX). - Corrected instrument source identifiers - Added CRF Type vs Source table - Added explanation regarding 'MPSF' and 'RULE' as possible sources within UniqueID. Section 3.2.5: Added explanation about Data Systems behaviour, in case the field insertOrDeleteFlag is not present. Section 3.2.6: Added missing section Profiles (Z-Records) Section 4: Updated details about Manifest files Section 7: Removed statement about XML prolog, since this is covered by the applicable [PF-ICD] document [A-1] Section 8: updated FD to use sftp to a GFTS node Section 9.1 (CRF naming convention): - Added reference to applicable PLID Annex B (MOC-SOC ICD). - Added `PORG` to list of possible types. Section 10: - Removed the title Annex 1. - Expanded description of CRF whitelists, including file naming and format example. Updates to cover the support of Planned Operations Files (POF): Figure 2: added POF arrow Section 3.1: added POF to command request file types Section 3.1.2: added POF description - updated various field descriptions in sections 3.2.1.1, 3.2.1.2, 3.2.2, 3.2.4 and 3.2.5 in order to qualify the fields for the POF CRF type. Section 3.2.1.2: removed strikethrough formatting of 'subScheduleID' since it is expected for the POF CRF type. - Added POF in section 3.2.3.3 and to Table 1 Section 9.1 (CRF naming convention): Added `POF_` to list of possible types.

Date	Iss. No.	Rev. No.	Description and Section No.
			Attached (zipped) schema files: 2017-09-04_SolO_PLID_Schema_Files_PLID_1.2.zip This includes updates to the following schema files [1]: `solPlanningData.xsd`: - updated insertOrDeleteFlag to be optional - updated constrainedDeltaTime pattern to optionally accept days. I.e. new format is [-][DDD.]hh:mm:ss `solSpecificTypes.xsd`: - changed profileType enumerations to DR (for data-rate) and PW (for power) Attached Word document, that includes track-changes with respect to previous issue, to this PDF.

[1] Although the PLID was already updated to support optional insertOrDeleteFlag and days within the constrainedDeltaTime, the Mission Data Systems (MCS and MPS) will require updates that are tracked within Software Change Requests BEPIMCS-1184, BEPIMPS-787 and BEPIMPS-788

DISTRIBUTION LIST

Recipient	*Organisation*	*Recipient*	*Organisation*
P. Kletzkine	SRE-PS	P. Ferri	OPS-O
Y. Bonnefous	SRE-MC	M. Lugert	OPS-ON
S. Strandmoe	SRE-PSS	M.Butkovic	OPS-ONO
S. Thuerey	SRE-PSA	T. Beck	OPS-ONX
C. Hass	SRE-PSV	Y. Doat	OPS-ONI
F. Marliani	SRE-PSP	E. Vassallo	OPS-OW
K. Wirth	SRE-PS	O. Procope	OPS-ONC
S. Fahmy	SRE-PS		
L. Icardi	SRE-PS	E. Dölling	FCI-ITC
Pierre Olivier	SRE-PSQ	C. Vicente	FCI-ITC
A. Oganessian	SRE-PSA	J. Miro	OPS-G
Y. Yushtein	TEC-SWE	M. Merri	OPS-GD
		N. Peccia	OPS-GI
D. Mueller	SRE-S	A. Ercolani	OPS-GDS
L. Sanchez	SRE-ODS	L. Michienzi	OPS-GDS
		G. Gaudenzi	OPS-GDS
A. Accomazzo	OPS-OP	P. Margarido	OPS-GDS
S. Lodiot	OPS-OPS	D. Werner	OPS-GDS
M. Eiblmeier	OPS-OPS		
D. Lakey	OPS-OPS	K.J. Schulz	OPS-GS
M. Pantaleoni	OPS-OPS	M. Lanucara	OPS-GSY
J. L. Pellon-Bailon	OPS-OPS	M.A. Larsen	OPS-GSC
		A. Mantineo	OPS-Q
M. Lauer	OPS-GFS	A. Freter	OPS-GSY
V. Companys	OPS-GFS	M. Rossmann	OPS-ONI
J. Schoenmaekers	OPS-GF	EILSservices@esa.int	
J.M. Sanchez Perez	OPS-GFA	A. Urwin-Phipps	SRE-PW
G. Ravera	OPS-ONO	E. Montagnon	OPS-OPB

T. Anic	OPS-BA	P. Schmitz	OPS-OPE
M.H. Ferreira	OPS-Q	I. Tanco	OPS-OPJ
H. Sillack	OPS-OP		

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Objective

The Solar Orbiter Planning Interface Document (PLID) is a mission-specific tailoring of the multi-mission, generic ICD called Planning Files Interface Control Document, MDS-MCS-SW-ICD-1001-HSO-GD, referred to here as [A-1] which contains the specification of the structure and content of all planning files related to the Solar Orbiter Planning process, to be processed by the Science Ground Segment (SGS), the Mission Planning System (MPS) or the Mission Control System (MCS). This includes commanding requests, command request responses, command stack files and event files.

1.2 Scope

The PLID is the top-level ICD, and contains only the format specification for:

- Command Request Files (CRFs), and manifest files that shall accompany their submission,
- Command Request Response (CRR) file reporting on the processing of the CRFs by the receiving system,
- Command Stack Files generated by the Mission Control System and the Mission Planning System
- Event Files.

The detailed description of the contents of these files (e.g. PTRs, Events, SORs) is to be found in the PLID annexes and associated lower-level ICD. There are three main Annexes to the PLID:

- Annex A: Flight Dynamics ICD, describing all the planning products generated by Flight Dynamics
- Annex B: MOC-SOC ICD, describing all the planning products generated by MPS and delivered to the SOC and vice-versa.
- Annex C: ESTRACK ICD, describing the interface to the ESTRACK Management System (EMS)

This Interface Control Document (ICD) is applicable for all phases of the Solar Orbiter mission.

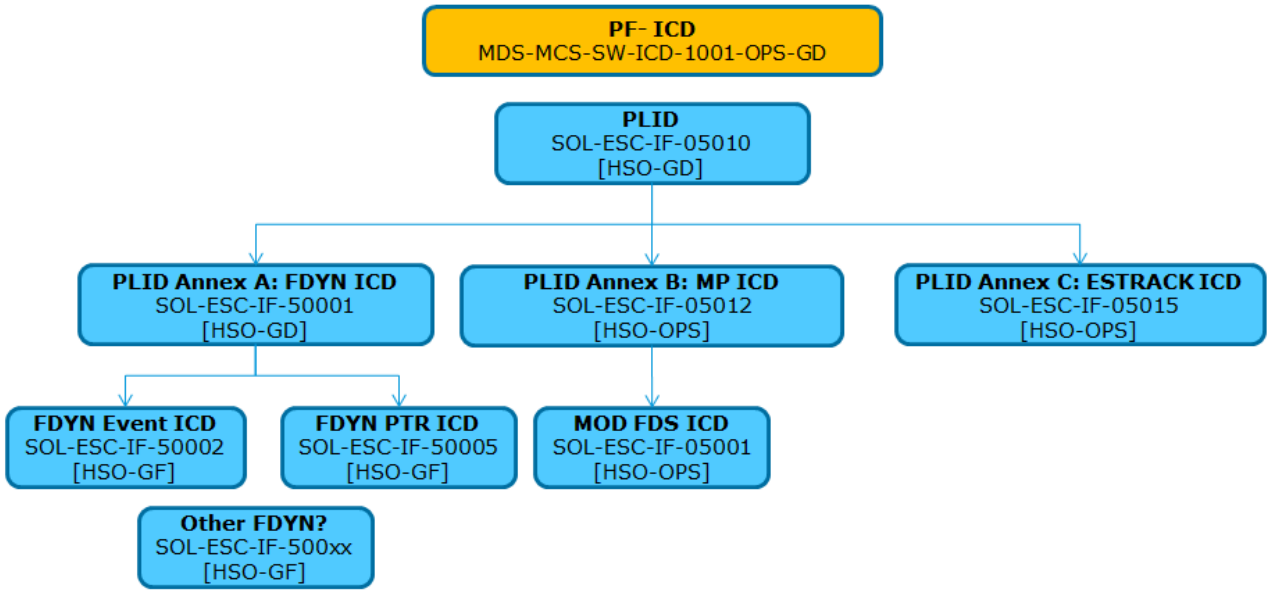

Figure 1: SOL PLID Structure

It should be noted that the formal definition of the file formats, defined in this document, is provided in the tables and text. The attached schema (and associated schema examples) should be in line with this ICD, but viewed as an auxiliary tool to help for validation purposes only.

1.3 Applicable and Reference Documents

1.3.1 Applicable Documents

ID	Document Title	Document Reference	Issue	Date
A-1	Planning Files ICD	MDS-MCS-SW-ICD-1001-OPS-GD	4.4	30/11/2016
A-2	Solar Orbiter GFTS Configuration Document		*TBW*	
A-3	Solar Orbiter Ground Segment System Requirements Document	SOL-ESC-RS-05000	1.1	
A-4	Solar Orbiter Science Operations Implementation Agreement	SOL-ESC-IA-05002	1.1	17/01/2014
A-5	*Deleted*			
A-6	Solar Orbiter MODFDS Interface Control Document	SOL-ESC-IF- 05001	*TBW*	
A-7	Solar Orbiter PLID Annex A – Flight Dynamics ICD	SOL-ESC-IF-50001	1.0	01/10/2013
A-8	Solar Orbiter PLID Annex B – MOC-SOC ICD	SOL-ESC-IF- 05012	*1.3*	31/08/2017
A-9	Solar Orbiter PLID Annex C – ESTRACK/FCT ICD	SOL-ESC-IF- 05015	*TBW*	
A-10	Rosetta/BepiColombo MPS Planning Database ICD	BC-ESC-IF-23002	*2.0*	09/12/20116

1.3.2 Reference Documents

ID	Document Title	Document Reference
R-1	*Deleted*	
R-2	*Deleted*	
R-3	*Deleted*	
R-4	SCOS-2000 Database Import ICD	EGOS-MCS-S2K_ICD-0001

1.4 Acronyms

CRF	Command Request File
CRFG	Command Request File Group (multiple CRF grouped in a ZIP file)
CRR	Command Request Response
CRRG	Command Request Response Group (ZIP File)
CSF	Command Stack File
DDS	Data Dissemination/Distribution/Disposition System
DOR	Direct Operation Request
EDDS	EGOS Data Disposition System
EMS	ESTRACK Management System
ESOC	European Space Operations Centre
EVF	Event File
FCT	Flight Control Team
FDR	Flight Dynamics Request
FDS	Flight Dynamics System
FOP	Flight Operations Procedure
GFTS	Generic File Transfer System
ICD	Interface Control Document
LAN	Local Area Network
MAES	Mission Automation Event Schedule
MAN	Manifest File
MAPS	Mission Automation Planning Schedule
MCS	Mission Control System
MDOR	Memory Direct Operation Request
MDS	Mission Data System
MIB	Mission Information Base
MOC	Mission Operations Centre
MPS	Mission Planning System
OGS	Operations Ground Segment
PDOR	Payload Direct Operation Request
POF	Planned Operations File
POR	Payload Operation Request
S/C	Spacecraft
SDE	Software Development Environment
SGS	Science Ground Segment
SOR	Spacecraft Operation Request
SRD	System Requirements Document
STP	Short Term Planning
SVF	Software Validation Facility
TC	Telecommand
UTC	Universal Time Coordinated
TBW	To be Written

2 CONTEXT FOR FILE EXCHANGE

The following diagram shows the context in which the files described in this document are used and between which entities they are sent. These files formats are defined in the next sections.

The entities in charge of generating and/or processing files within the Operational Ground Segment (OGS) at ESOC are:

- FDS – **F**light **D**ynamics **S**ystem
- MCS – **M**ission **C**ontrol **S**ystem (telecommanding system)
- MPS – **M**ission **P**lanning **S**ystem
- FCT – **F**light **C**ontrol team off-line **T**ools *(offline software tools that are in hands of the Flight Control Team to develop the CRF and Event files to be used on the MPS/MCS)*
- EMS – **E**STRACK **M**anagement **S**ystem

Outside ESOC, the following centres are set-up to generate and/or process command request files:

- SGS – **S**cience **G**round **S**egment at ESAC
- SOL PIs

Figure 2 provides a conceptual overview of the command file exchange between Solar Orbiter users/data systems. The command request file types listed on this figure are defined in section 3.

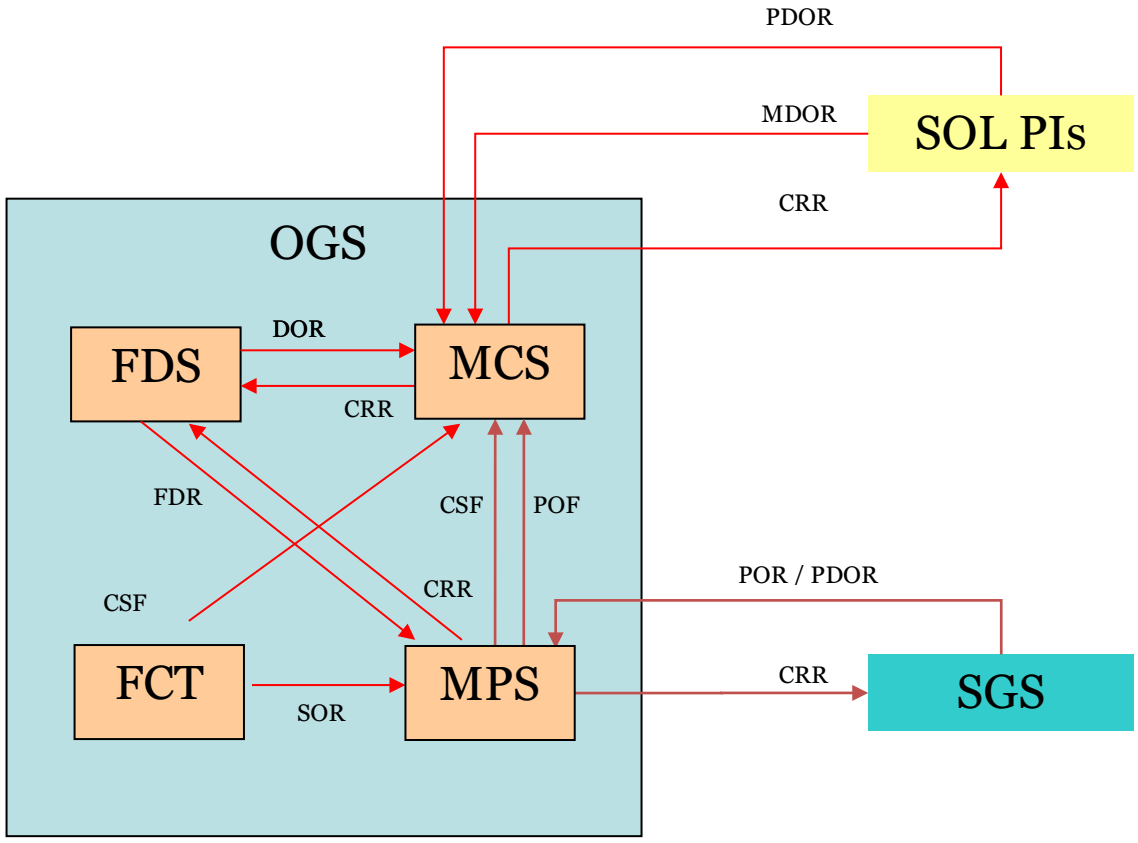

Figure 2: Overview of Command Files Exchange

Figure 3 provides a functional overview of the event file exchange between Solar Orbiter users/data systems:

- Link 1 refers to the exchange of orbital events between Flight Dynamics and EMS for the purpose of long-term ground station conflict analysis. This event file is defined in [A-7].
- Link 2 refers to the provision of event files by Flight Dynamics for mission planning. This event files are defined in [A-7].
- Link 3 refers to the consolidated mission event file provided by MPS to SGS for mission planning. This event file is defined in [A-8].
- Link 4 refers to the provision of an event file by MPS to EMS for the purpose of generation of ground station operations schedule. This event file is defined in [A-9].

All event files are made available to all external users via the DDS.

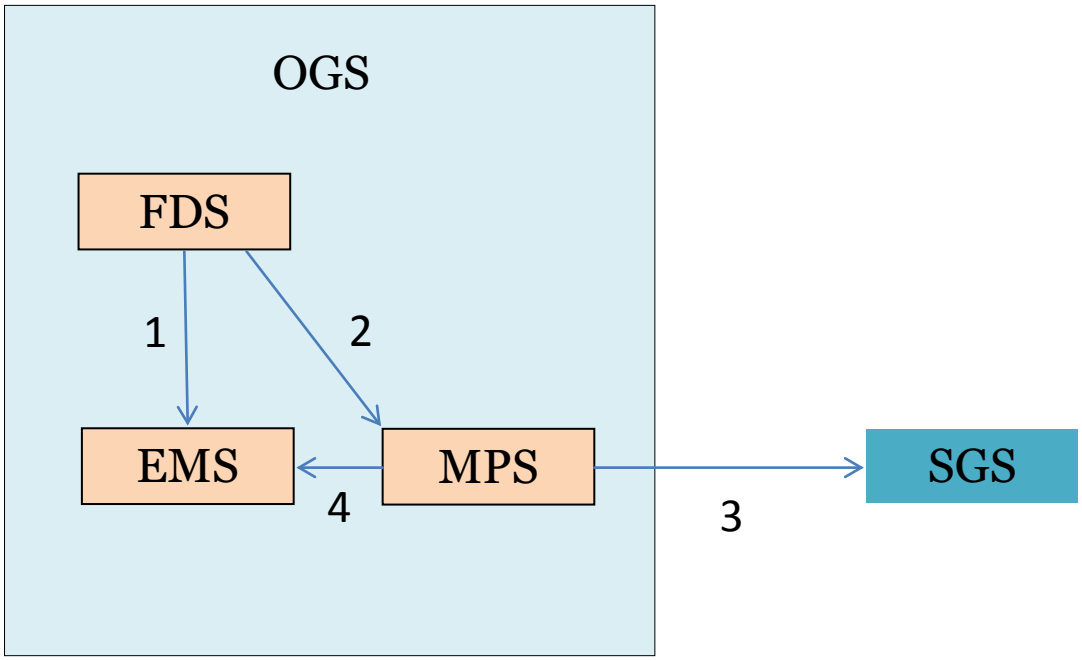

Figure 3: Overview of Event Files Exchange

3 COMMAND REQUEST FILES

The generic CRF format is defined in the PF-ICD [A-1], which includes an associated XML Schema.

The following section tailors the PF-ICD [A-1] by defining how CRFs are to be instantiated for use on Solar Orbiter. In particular this section describes any mission-specific constraints which should be applied to elements or attributes of the generic format, e.g. what the accepted values are and how optional fields should be populated etc.

The Solar Orbiter specific XML types to be included in the PF-ICD XML Schema used for validating CRFs (included as **`solSpecificTypes.xsd`**) is provided with this document.

3.1 Command Request File Types

There are 6 file types, or instances of a CRF to be used by the Solar Orbiter MCS TC chain or MPS, each of which is an enumeration of the **`crfFileType`** XML simple type, (used to populate the **`type`** attribute of the **`header`** complex element), as follows:

- SOR
- POR
- FDR
- DOR
- PDOR
- MDOR
- POF

The above listed file types can be grouped as:

- Inputs to MPS
- Inputs to MCS TC System

In addition to the above file types, there will be an instance of CRR file generated by the MCS (CRF Checker application) when receiving a CRF. This will validate the CRF content and pass the response back to the source. If it is valid, the CRF will be made available to the MPS or MCS as appropriate.

3.1.1 INPUTS TO MPS

1. Spacecraft Operation Requests (**SOR**): these may include any type of spacecraft (both platform and payload) or ground segment operations and related ground station and control system operations. These shall be prepared by the Flight Control Team (FCT), possibly with the assistance of an offline tool.

2. Payload Operation Requests (**POR**): these will contain inputs for routine science operations during Cruise Phase (In-situ instruments and Remote Sensing payload checkouts) and routine science operations during Normal Science Phase (full payload complement). These requests shall be generated only by the SGS.

3. Flight Dynamic Requests (**FDR**): these will contain inputs from the flight dynamics system and will contain requests pertaining to;

- Attitude and orbit manoeuvres
- Configuration parameters/tables (i.e. Gyro calibration)
- AOCS mode changes
- Other AOCS/RCS commands

3.1.2 INPUTS TO MCS TC SYSTEM

4. Direct Operation Requests (**DOR**): these also originate from the Flight Dynamics System and will contain the same request types as the FDR files but for direct input to the MCS TC system. DORs are typically used in non-routine scenarios, e.g. LEOP, contingency, safe mode recovery.

5. Payload Direct Operations Requests (**PDOR**): these will contain inputs from Js for payload engineering operations. These inputs shall be injected directly to the MCS TC system and they are for engineering purposes that cannot be pre-planned (e.g. troubleshooting, instrument maintenance, interactive instrument operations).

6. Memory Direct Operation Request (**MDOR**): these contain sequences for the purpose of patching or dumping instrument on-board memories, as well as for requesting checksums. They are also processed directly by the MCS TC system and are to be used by PI teams to specify memory maintenance operations. The format is similar to the PDOR format but utilising octet string parameter type to allow a memory patch to be specified as a string of hexadecimal values.

7. Planned Operations Files (POF): these contain planned operations in form of sequences, instead of single commands (CSF), output by the MPS towards the MCS. For the Solar Orbiter mission, it is intended to use the POF format instead of the CSF format on that MPS to MCS interface.

3.2 Command Request File Format

The following sections describe the Solar Orbiter specific population rules and constraints for elements and attributes of the CRF format, as defined in the PF-ICD XML Schema [A-1].

3.2.1 Solar Orbiter instantiation of the PFICD format

For convenience, this section reproduces the CRF template defined in the PFICD and illustrates how it is instantiated for Solar Orbiter. The elements modified on Solar Orbiter w.r.t to the PFICD are highlighted with a grey background. The elements which shall not be included in the Solar Orbiter files are striked-through.

The first column (**Field**) provides the name of the element (tag value) or attribute as seen in the resulting XML file. The second column (**E/A**) indicates whether the field is an XML element (using 'E'), or an attribute (using 'A') of an XML element. The third column (**Type**), describes the basic type of the field. The fourth field (**Description**) provides a textual definition of the field which may contain examples, format specifications and conditional statements about what the field may contain. The final column (**Need**) indicates whether the field's value is Mandatory (M), Optional (O), or Qualified (Q) – the latter requiring explanation in the Description column.

3.2.1.1 Header

Field	E/A	Type	Description	Need
type	A	string	Identifies the filetype. It is an enumerated as defined in section 3.1.	M
formatVersion	A	integer	A positive number identifying the version of the CRF format specification, allowing the processing software to take into account format changes. The value which will be applicable to this current version of the PFICD will be '1'	M
~~**fileVersion**~~	~~A~~	~~integer~~	~~A non negative number whose population can be freely defined per mission to allow the differentiation of different versions of this file~~	~~O~~
genTime	E	string	File Generation time in format: **YYYY-DDDThh:mm:ss**[.mmm]**Z**	M
validityRange	E	complex	Specifies the file's validity start and end times, which implies that any execution time of requests in the file must fall within this validity range. In some cases, it does not make sense to provide a validity time range, e.g. for requests which are treated as 'standing orders' and do not have an explicitly associated time, but refer to regular events. In this case the validityRange element can be omitted. For a CRF of type 'POF', this shall be set to the time range specified for the generation of this file. If used, for Solar Orbiter only the value "absoluteTime" is allowed.	O

When the **validityRange** element is present, then it shall be conform to the following complex type:

Field	E/A	Type	Description	Need
type	A	string	For Solar Orbiter, the validityRange element's attribute 'type' needs to be present, and is restricted to the value "absoluteTime"	M
startTime	E	string	Start of file's validity time which must be no later than the time of the first request in the file, expressed as follows: YYYY-DDD**T**hh:mm:ss[.mmm]**Z**	M
stopTime	E	string	End of file's validity and must be no earlier than the time of the last request in the file, expressed as follows: YYYY-DDD**T**hh:mm:ss[.mmm]**Z**	M

Example of a possible validityRange element:

```
<validityRange type="absoluteTime">
   <startTime>2022-001T00:00:00Z</startTime>
   <stopTime>2022-181T00:00:00Z</stopTime>
</validityRange>
```

3.2.1.2 Ocurrence List

The **occurrenceList** complex element (as described in 3.2.2) represents the set of operations requests contained in the file, each of which can be one of four elements:

- **command** which is of type **commandOccurrence**,
- **sequence** which is of type **sequenceOccurrence**,
- **procedure** which is of type **procedureOccurrence** and
- **event** which is of type **eventOccurrence**.

The **occurrenceList** complex element shall be constructed as follows:

Field	E/A	Type	Description	Need
count	A	integer	A count of the number of occurences	M
creationTime	A	string	The creation time of this occurrence list in absolute time format: YYYY-DDD**T**hh:mm:ss[.mmm]**Z**	M
author	A	string	Used to specify the user who created the occurrence list. For a CRF of type 'POF', this shall be set to 'MPS'.	M
~~**description**~~	~~A~~	~~string~~	~~Used only for visualisation purposes in order to better annotate nested occurrence lists~~	~~O~~

Occurrences consist of the following, but note that some elements do not apply to all occurrence types.

Field	E/A	Type	Description	Need
type	A	string	See section 3.2.2 The type is only mandatory for **eventOccurences** since command, sequence and procedure occurrences have only one possible type.	Q
name	A	string	The name of the occurrence which must match the	M

			name of the operation to be carried out exactly as defined in the relevant database – it is case sensitive. Note: this is restricted to 8 characters for command and sequence occurrences.	
passID	E	string	This field shall be left blank for all occurrences in all request file types, except in SORs where it can be used to specify a reference to a Ground Pass for execution of automation or ground station operations.	O
uniqueID	E	string	See section 3.2.4	M
insertOrDeleteFlag	E	string	See section 3.2.5	O
source	E	string	This field is used to be backward compatible with SCOS 2000 (source field in the TPF). It is used to identify the origin of the file. Allowed values are: 'SSGS' (Solar Orbiter Science Ground Segment) 'SFDR' (Solar Orbiter Flight Dynamics) 'SFCT' (Solar Orbiter FCT). Each instrument shall have its own source identifier as defined below: • 'SEPD' (EPD) • 'SEUI' (EUI) • 'SMAG' (MAG) • 'SMET' (METIS) • 'SPHI' (PHI) • 'SRPW' (RPW) • 'SSPI' (SPICE) • 'SSHI' (SOLOHI) • 'SSTX' (STIX) • 'SSWA' (SWA)	M
destination	E	string	This value is used to identify a target output. Two values shall be supported for Solar Orbiter: • C for Critical – will be used to segregate critical commands • R for Routine – will be used for all routine occurrences The use of value R shall be used for all occurrences in all request file types, except in SORs, where the value of C may be used.	M
subScheduleID	E	integer	This is applicable only if the occurrence is a command with an execution time or a sequence containing time-tagged commands. It is used for identifying groups of commands in the on-board queue that can be affected together.	O

			If there is no value or set to 0, the sub-schedule ID as defined in the relevant database will be used. I.e. in the SCOS-2000 MIB, CCF_SUBSCHEDID (for commands) or CSF_SUBSCHEDID (for sequences) External users shall not specify sub-schedule ID values as part of their operations requests. For CRF of type POF, the type is present to reflect the subschedule ID settings defined as part of the mission planning process.	
releaseTime	E	complex	see below and section 3.2.3. This field shall be omitted in CRFs of type 'POF' and in CRFs to the MPS (i.e. SOR, FDR, POR). In other CRFs to the MCS (i.e. DOR, PDOR, MDOR), it is exclusive with the **executionTime** field. For a CRF of type 'POF', this field shall be omitted.	O
executionTime	E	complex	see below and section 3.2.3. In CRFs to the MCS (i.e. DOR, PDOR, MDOR), this field is exclusive with the **releaseTime** field. For a CRF of type 'POF', this field is mandatory.	O
iVSTP	E	complex	see below. This field is optional and reserved for use within CRFs from SGS to the MPS (i.e. POR). The field serves mission planning and can be used during STP. The SGS can use it to define so called iVSTP windows, which will later be used to insert dedicated instrument commands.	O
~~**description**~~	~~E~~	~~string~~	~~Mission specific operational data (e.g. units selection) which are delivered for information/printout only.~~ ~~*Note: This field can be used as a field for miscellaneous data if required by a mission, in order to avoid mission specific extensions to the format.*~~	~~O~~
parameterList	E	complex	See section 3.2.1.2.1	O

The **releaseTime** complex element is composed of the following:

Field	E/A	Type	Description	Need
actionTime	E	string	The time at which the occurrence is to be released, defined as a constrained positive delta time (with respect to the previous operation) format: `hh:mm:ss` If this field is empty then it will imply an ASAP release time.	M
~~**eventID**~~	~~E~~	~~string~~	~~An alphanumeric event mnemonic (contained in either an Event file see section) relative to which the operation is to be released.~~ ~~Note: This field can only contain a value if there is no actionTime release time.~~	~~O~~

			If the eventCount field is specified, it will be combined with the eventID to identify a single instance of an event. If eventCount is not specified then all events with the eventID will be referenced – an operation will be created at the event time + specified delta for every occurrence of this event. If repeat is specified, then repeat operations will be created for each occurrence of the event.	
eventCount	E	integer	Only allowed if eventID is specified, identifying a specific instance of the event specified in eventID.	O
eventCount2	E	integer	Mandatory if eventCount is specified. It is used in order to specify a range of event instances to be triggered. If both eventCount and eventCount2 are not specified or set to zero, then trigger all instances of the event. If the both eventCount and eventCount2 have the same non zero value, then trigger only that instance of the event Otherwise, trigger only the event instances between eventCount and eventCount2 (inclusive) Must be equal to or greater than eventCount	O
delta	E	string	Only allowed if eventID is specified The offset time to the event specified in the eventID field (described above) at which the occurrence should be scheduled. If this is not specified then the event will be scheduled at exactly the time of the event If specified it must be in the following format: [ ][ddd.]hh:mm:ss[.mmm]	O
repeat	E	integer	The number of times that an operation is to be scheduled for the same instance of the event specified by eventID. If it is not specified then the occurrence will be scheduled only once for that event instance. It can be defined for any start time specification. If specified the operation will be created first at the start time and then (repeat – 1) times separated by the separation time. If a value is specified then it must be an integer number in the range 00002 – 99999.	O
separation	E	string	This is mandatory only if the value in the repeat field is specified, otherwise it is ignored. This is used to specify the separation between repeated occurrences of an operation. If a value is specified then it must be a delta time in the format: [ddd.]hh:mm:ss[.mmm]	O
earliestOffset	E	string	The offset of the start of the time window for the operation, which is applied to the actionTime or the eventID, if one of them has been specified. Specified as a delta time: [ ][ddd.]hh:mm:ss[.mmm] This field is mandatory if a latestOffset is provide and if	O

			~~specified, its value must be zero or negative.~~	
~~**latestOffset**~~	~~E~~	~~string~~	~~The offset of the end of the time window for the operation, which is applied to the actionTime or the eventID if one of them has been specified~~ ~~Specified as a delta time:~~ ~~[ddd.]hh:mm:ss[.mmm]~~ ~~This field is mandatory if an earliestOffset is provided and if specified, its value must be zero or positive.~~	~~O~~
~~**propagationFactor**~~	~~E~~	~~integer~~	~~The value of a factor by which to multiply the propagation delay. A positive number will add multiples of the delay to the time, a negative number will subtract.~~ ~~If no value is specified in this field, 0 will be used as the default, i.e. no delay applied~~ ~~This field can be used in conjunction with the event (eventID and delta) or the actionTime depending on which (if either) is specified.~~	~~O~~

The **executionTime** complex element is composed of the following:

Field	E/A	Type	Description	Need
actionTime	E	string	The time at which the occurrence is to be executed or started. Absolute time format: `YYYY-DDD`**`T`**`hh:mm:ss[.mmm]`**`Z`** Any time-tagged commands contained in a sequence are provided with execution times relative to this one. This field is mutually exclusive with the ***eventID***. On Solar Orbiter, this field is mandatory for CRFs to the MCS (i.e. DOR, PDOR, MDOR, POF), if the **executionTime** complex element is being used.	O
eventID	E	string	An alphanumeric event mnemonic as defined in the relevant Solar Orbiter ICDs [A-7] and [A-8], relative to which the operation is to be executed or started. *Note: This field can only contain a value if there is no* ***actionTime****, i.e. no absolute time is specified.* If the ***eventCount*** field is specified, it will be combined with the ***eventID*** to identify a single instance of an event. If ***eventCount*** is not specified then all events with the ***eventID*** will be referenced - an operation will be created at the event time + specified delta for every occurrence of this event. If ***repeat*** is specified, then ***repeat*** operations will be created for each occurrence of the event.	O
eventCount	E	integer	Only allowed if ***eventID*** is specified.	Q

			This field identifies a specific instance of the event specified in ***eventID***.	
eventCount2	E	integer	Mandatory if **eventCount** is specified. It is used in order to specify a range of event instances to be triggered. If both ***eventCount*** and ***eventCount2*** are not specified or set to zero, then trigger all instances of the event. If both ***eventCount*** and ***eventCount2*** have the same non zero value, then trigger only that instance of the event. Otherwise, trigger only the event instances between ***eventCount*** and ***eventCount2*** (inclusive). Must be equal to or greater than ***eventCount***.	Q
delta	E	string	Only allowed if ***eventID*** is specified An offset time to the event specified in the ***eventID*** field (described above) at which the occurrence should be scheduled. If this is not specified then the occurrence will be scheduled at exactly the time of the event If specified it must be in the following format: `[-][ddd.]hh:mm:ss[.mmm]`	Q
repeat	E	integer	The number of times that an operation is to be scheduled for the same instance of the event specified by ***eventID***. If it is not specified then the occurrence will be scheduled only once for that event instance. It can be defined for any start time specification. If specified the operation will be created first at the start time and then (***repeat*** – 1) times separated by the ***separation*** time. If a value is specified then it must be an integer number in the range 00002 – 99999.	O
separation	E	integer	This is mandatory only if the value in the ***repeat*** field is specified, otherwise it is ignored. This is used to specify the separation between repeated occurrences of an operation. If a value is specified then it must be a delta time in the format: `[ddd.]hh:mm:ss[.mmm]`	Q
~~**earliestOffset**~~	~~E~~	~~string~~	~~The offset of the start of the execution time window for the operation, which is only applied to the actionTime or the eventID, if one of them has been specified~~ ~~Specified as a delta time:~~ ~~`[-][ddd.]hh:mm:ss[.mmm]`~~ ~~This field is mandatory if a latestOffset is provided.~~ ~~If specified, its value must be zero or negative.~~	~~Q~~
~~**latestOffset**~~	~~E~~	~~string~~	~~The offset of the end of the execution time window for the occurrence, which is only applied to the actionTime or the eventID, if one of them has been specified.~~ ~~Specified as a delta time:~~	~~Q~~

			~~[ddd.]hh:mm:ss[.mmm]~~ ~~This field is mandatory if an earliestOffset is provided.~~ ~~If specified, its value must be zero or positive.~~	
propagationFactor	E	integer	The value of a factor by which to multiply the propagation delay. A positive number will add multiples of the delay to the time, a negative number will subtract. If no value is specified in this field, 0 will be used as the default, i.e. no delay applied Can be used in conjunction with the event (***eventID*** and ***delta***) or ***actionTime*** depending on which (if either) is specified. On Solar Orbiter, this field must be left blank or omitted in CRFs to the MCS (i.e. DOR, PDOR, MDOR, POF), since the use of this feature is not supported on the MCS.	O

The **iVSTP** complex element is composed of the following:

Field	E/A	Type	Description	Need
startTime	A	string	The time at which the iVSTP validity window will start. Absolute time format: YYYY-DDD**T**hh:mm:ss[.mmm]**Z**	M
endTime	A	string	The time at which the iVSTP validity window will end. Absolute time format: YYYY-DDD**T**hh:mm:ss[.mmm]**Z**	M
max_TCs	A	integer	A non-negative number	M
hex_onboard_filename	A	string	The name of the on-board filename (hex value)	M
POR_ground_filename	A	string	The name of the corresponding POR filename	M

Example of a possible iVSTP element:

```
<iVSTP starttime="2022-175T23:53:51Z" endtime="2022-175T23:58:51Z" max_TCs="10"
hex_onboard_filename="2160007B" POR_ground_filename="POR__SSGS_V123_IH_22001.SOL" />
```

3.2.1.2.1 Parameter List

The **parameterList** complex element represents the set of parameters associated with the occurrence. The number of parameters in this list ***must*** be the same as the number of parameters defined for that operation in the relevant database, i.e. all database defined parameters must be included. In addition, the **count** attribute of the **parameterList**, is optional but if it is specified, then its value must also be the same as the number of parameters in the list. It is used only for internal consistency checking.

Field	E/A	Type	Description	Need
count	A	integer	A count of the number of parameters	O

Ancillary data such as the parameter name and value qualifiers are also delivered in order to enable the processing software to perform syntactical and consistency checks. The **position** attribute of the parameter must match the position according to the ordering of the parameters as specified in the relevant database (e.g. using the SCOS-2000 MIB, sequences are ordered by CSP_FPNUM, and commands are ordered by CDF_BIT).

Note: For occurrences of type command, there must exist an entry for every defined editable parameter and not for fixed areas or fixed parameter. (I.e. all elements with a CDF_ELTYPE value of 'E'). In addition, if any parameter of the command is defined in the SCOS-2000 database as a group repeater parameter, and it has a value other than 1, then the total number of parameter elements in the list must match the expanded command structure. For example, taking a command that is defined as having a group repeater which controls a group of three parameters. If the group repeater has a value of 2, then the total number of parameters which should be supplied would be 7 (group repeater plus 2 repetitions of a group of 3).
The same applies to occurrences of type sequence, there must exist an entry for every defined editable parameter and not for fixed areas or fixed parameter. (I.e. all elements with a SDF_FTYPE value of 'E').

For a CRF of type 'POF', the parameter fields shall be populated as they are defined in the MPS.

In the tables below reference is made to SCOS-2000 database field names as specified in [R-4] when the occurrence is either command or sequence.

The **`parameter`** elements consist of the following:

Field	E/A	Type	Description	Need
`name`	A	string	The name of the operation parameter for which a value is supplied in this data record. Note: This is restricted to 8 characters for command and sequence parameters.	M
`position`	A	integer	Defines the position of the parameter in terms of the ordering specified in a database if applicable. This is so that there is no ambiguity in case a given occurrence has multiple instances of a given parameter. The first parameter shall have position 1.	M
`unit`	A	string	This field is free text and used to check for consistency with the SCOS-2000 database against a unit for operations only of type telecommand or sequence. If the ***unit*** for a value is specified in the database, then the CRF file must contain the same unit for that value. If the ***unit*** is not present in the database, then it must also be absent in the CRF file. For clarification on how ***representation***, ***unit*** and ***radix*** and ***value*** can be combined see the table above.	O
`description`	E	string	This field can be used to add remarks about the parameter and parameter value being supplied.	O
`value`	E	string	This is field is mandatory if ***representation*** is explicitly specified. If this field is omitted, the default value (specified in the relevant database) will be used for this parameter. If this field is left blank and there is no default in the database then the CRF file shall be rejected. The format to be used to specify the parameter value shall comply with the parameter type for raw values or with the calibration type for engineering values. • Unsigned Integer. This format is used to supply raw values for parameters of type Boolean Enumerated or Unsigned Integer. It can also be used in principle to supply engineering values for parameters which are associated to a calibration curve with an unsigned integer engineering format. The adopted radix in case of raw is specified in the ***radix*** attribute, see below.	Q

			• Signed Integer. This format is used to supply raw values for parameters of type Signed Integer. It can also be used to supply engineering values for parameters which are associated to a calibration curve with a signed integer engineering format. • Real. Double precision real values can be specified using the IEEE scientific or floating point notation e.g. 1.23E3 or 1230.00. This format is used to supply raw values for parameters of type Real. It can also be used to supply engineering values for parameters which are associated to a calibration curve with a real engineering format • ASCII String. This format is used to supply raw values for parameters of Character-string type or engineering values for parameters calibrated using text strings e.g. ON/OFF. • Octet String. This format is used to supply raw values for variable length octet string parameters (PUS PTC 7). Each octet or byte is represented by a pair of hexadecimal characters. *Note: In previous missions (e.g ROS/MEX/VEX) this would have been done using an 'R' record.* • ASCII Time format. Absolute time values shall be expressed in the format `YYYY-DDDThh:mm:ss[.mmm]Z` This format is used to supply values for parameters of type absolute time • Delta/Relative Time. Delta times shall be expressed in the format `[-][ddd.]hh:mm:ss[.mmm]`. This format is used to supply raw values for parameters of type delta time. For clarification on how ***representation***, ***unit*** and ***radix*** and ***value*** can be combined see the table above.	

The **`value`** element may contain the following optional attributes:

Field	E/A	Type	Description	Need
`representation`	A	string	This field specifies the type of the parameter value as follows: • '`Raw`' : the value is supplied in uncalibrated form i.e. no de-calibration has to take place in order to evaluate the value to be encoded for uplink. Note that this does not necessarily imply that an unsigned integer value is supplied. Depending on the parameter type, a raw value can in fact be e.g. a signed integer, a real or a time value. • '`Engineering`' : the value is supplied in calibrated form i.e.	O

			it has to be de-calibrated in order to evaluate the value to be encoded. Depending on the type of the parameter calibration, engineering values can be supplied in different formats e.g. a real value supplied in engineering units or a text string. For clarification on how ***representation***, ***unit*** and ***radix*** and ***value*** can be combined see the table above.	
radix	A	string	The radix must be specified for all raw parameters. It must be omitted for all engineering parameters. For unsigned integer raw parameters it can take one of the following values: • 'Decimal' • 'Hexadecimal' • 'Octal' For parameters of type Octet String it must be 'Hexadecimal' For all other raw parameter types it must be 'Decimal'. This field is used by the CRF processing software to make any relevant conversions. For clarification on how ***represenation***, ***unit*** and ***radix*** and ***value*** can be combined see the table above.	O

The parameter list may contain parameters in which the value field is left blank, indicating that the default value for that parameter should be used. However, this is illegal if there is no default value specified for that parameter in the database and the CRF will therefore be rejected. A value can be specified for a parameter which is the same as its default value defined in the relevant database.

For commands and sequences, only the following three possibilities of combinations are permitted.

Value	Representation	Unit	Radix	Comment
Specified	"Raw"	Irrelevant (specified or omitted)	Specified for unsigned integer values	Explicitly specify raw value; the specified radix (Decimal, Hexadecimal or Octal) is used to interpret an unsigned integer value (Note: All other parameter types will be implied decimal) The unit can be specified but would be disregarded by the CRF processing system. This relaxation of the acceptance rules for CRF files is in place for operational ease only. For clarity it is highly recommended to not specify a unit for a raw value.
Specified	"Engineering" (calibrated)	Specified or omitted	Omitted	Explicitly specify engineering value; the specified unit is checked against the unit in the database and an error is flagged in case of a mismatch. The radix must be omitted in this case. If the unit for a value is specified in the relevant database, then the CRF file must contain the same unit for that value, i.e. no conversion takes place. If the unit is not present in the database, then it must also be absent in the CRF file.
Omitted	Omitted	Omitted	Omitted	Take default parameter value and all necessary information from database. If a parameter value is not given and there is no default

Value	Representation	Unit	Radix	Comment
				parameter value in the database, then the CRF file shall be rejected.

3.2.1.2.2 Relative To Event Time ('E' Record)

The need to use E-Records has not been identified on Solar Orbiter and will therefore not be supported.

3.2.1.2.3 Profiles ('Z' Record)

In addition to a parameter list, the **sequence** complex element also contains an optional list of profiles **`profileList`**, which were previously referred to as Z-records. For a CRF of type 'POF', profiles shall however not be used. In this list, the number of profile elements (which can be null) must be the same, as the value specified in the **`count`** attribute of the **`profileList`**, if it is specified

If present, the **`profile`** element consists of the following:

Field	E/A	Type	Description	Need
`type`	A	string	An enumeration identifying the type of profile. See section 3.2.6 for the values to be used on Solar Orbiter.	M
`timeOffset`	E	string	Defined in the constrained delta time format: `[-][DDD.]hh:mm:ss.` The offset time is relative to the event defined in the respective ***eventID*** of the sequence. The time offset of the first profile associated to a sequence shall be • Non-negative in case the sequence has an ***actionTime*** specified (providing an absolute time value for the execution time). • Not lower than the field ***delta*** in case the sequence has an execution time relative to an event. The time offset of profiles associated to a sequence shall be strictly monotonically increasing (i.e. no profiles shall have the same time offset value).	M
`value`	E	real	See section 3.2.6.	M

3.2.2 Occurrence Types

Solar Orbiter will use the 4 occurrence types as defined by the PF-ICD XML Schema, i.e.:

- **`commandOccurrence`**
- **`sequenceOccurrence`**
- **`procedureOccurrence`**
- **`eventOccurrence`**

Solar Orbiter will not allow nested occurrences, i.e. an **`occurrenceList`** cannot itself contain a nested instance of **`occurrenceList`**.

Further, the following sub-categorisation of CRF occurrence types (defined as enumerations of the **type** attribute of the associated occurrence XML complex element) is expected:

`command` **(TC)**: there is no sub-categorisation for a **command** element (**commandOccurrence**)

`sequence` **(SQ)**: there is no sub-categorisation for a **sequence** element (**sequenceOccurrence**)

`procedure` **(AP)**: there is no sub-categorisation for a **procedure** element (**procedureOccurrence**)

`event`: allows the following sub-categorisation types for an **event** element (**eventOccurrence**)

- **'UserEvent' (UE)** – user events are used to inject events in the planning system.
- *Other sub-categories are TBD.*

For a CRF of type 'POF', the occurrence type shall be omitted.

The required occurrence types are allowed in the following file types:

		command	sequence	procedure	event
Inputs to	**CRF type**	**TC**	**SQ**	**AP**	**UE**
MPS	**SOR**		X	X	X
	POR		**X**		
	FDR		**X**		
MCS	**DOR**		**X**		
	PDOR	**X**	**X**		
	MDOR	**X**	**X**		
	POF		**X**		

Table 1 - Allowed occurrence types per CRF type

The whitelist mechanism described in Section 10 allows to define for each file type the occurrences which are allowed to be included.

3.2.3 Time Specification

Each operation request contains one or more time specification fields, which depending on the request will be an execution/start time and/or a release time. Alternatively a request may refer to an event to determine its execution/start or release time. Release and execution time specification are exclusive for CRFs to be processed by the MCS.

3.2.3.1 Release Time

The release time defines the time at which an occurrence should be released by the Mission Control System. It is only relevant for interactive operations, i.e. the release time elements may only be present for CRFs processed by the MCS. The release time elements shall not be included in CRFs to the MPS.

The release time may be specified as:

- as a delta-time, in which case it refers to the previous occurrence in the CRF,
- or as an empty string, in which case it means a release time as soon as possible.

In case there is no previous occurrence, the delta-release time will be applied from the moment the operator initiates the release of the command on the commanding stack.

Release time specification relative to event is not supported on Solar Orbiter. The use of the **`propagationFactor`** element to complement the definition of a release time is not supported on Solar Orbiter.

Solar Orbiter does not support the notion of time window, therefore the **`earliestOffset`** and **`latestOffset`** elements shall not be included in the CRFs. This implies that all operations will be released at the exact time specified.

3.2.3.2 Execution Time

The execution time defines the time at which the occurrence will be executed. This implies that TC and Sequence occurrences with a specified execution time are loaded on the on-board Mission Timeline for execution at the specified time. CRFs for processing by the MPS shall include the execution time element. CRFs for processing by the MCS may include it.

In CRFs for processing by the MPS, the execution time will be specified at sequence level, as absolute time or relative to an event. As defined in PF-ICD [A-1], the **`propagationFactor`** element can be used to complement the specified execution time.

For CRFs to be processed by the MCS, execution time shall only be specified as absolute time. In this case, the use of the **`propagationFactor`** element to complement the definition of an execution time is not supported.

For execution time specified against events, **`eventCount`** and **`eventCount2`** should be both set to non-zero values, except in SORs, where the use of zero is allowed. If these two fields have the same value, the occurrence is scheduled against a single instance of the event. If these two fields have different non-zero values, the occurrence is scheduled against a finite range of events. When these two fields are both set to zero, the occurrence should be scheduled against all instances of the specified event (standing order).

For execution time specified against events, the use of the repeat and separation fields is allowed in all request file types. This can be used to specify that an occurrence is to be repeated a finite number of times with fixed duration of time between two instances of the occurrence. The event reference is used to schedule the first occurrence instance.

Solar Orbiter does not support the notion of time window, therefore the **`earliestOffset`** and **`latestOffset`** elements shall not be included in the CRFs. This implies that all operations will be executed at the exact time specified.

3.2.3.3 Time specification summary

The following table summarises the time specification fields that can be used for operation requests as determined by the type of request file. Only those time specifications listed can be used with the combination of file type.

Time Specification Key
RT(abs) = Release Time of the request, defined as absolute time
RT(rel) = Release Time of the request, defined as relative time
ET(abs) = Execution Time of the request, defined as absolute time
+/- PD* = Optionally, the propagation delay complements the specified time
ET(EV) = Execution time of the request relative to the time of an event
DEL* = The optional offset time from the associated event – the **`delta`** element

Inputs to	Request File Type	Valid Time Specifications (* indicates optional value)	Notes
MPS	SOR POR FDR	• ET(abs) +/-PD* or • ET(EV) + DEL* +/-PD*	Use of zero for **`eventCount`** and **`eventCount2`** is allowed only for SORs.
MCS	DOR MDOR PDOR POF	• RT(abs) or • RT(rel) or • ET(abs)	An occurrence definition may not include both release and execution time.

3.2.4 Unique Identification of Occurrence Requests

Unique identification of requests is needed by the MPS and the MCS in order to implement the delete request mechanism and to eventually provide a means of tracing commands/sequences. This unique identifier will be maintained throughout the life-cycle of the occurrence and reported in telecommand history, from where it is made available via the DDS. This allows external users to check the status of commanding requests which have been submitted.

The unique identifier element **`uniqueID`** is used by the processing software, to uniquely identify a request. When deleting a request, this identifier is used to find the originally inserted request. The occurrence's **`name`** is cross-checked with the insert and delete requests to ensure consistency.

The following rules shall be followed for definition of the **`uniqueID`**:

- It shall be limited to 20 characters
- The first 4 characters shall identify the request source (list of values provided below)
- The last 16 characters shall consist of alpha-numeric characters always guaranteed not to be repeated by any other file of the same request source.

The following sources can provide requests with a **`uniqueID`**,

- 'SSGS' (Solar Orbiter Science Ground Segment)
- 'SFDR' (Solar Orbiter Flight Dynamics)
- 'SFCT' (Solar Orbiter FCT).

Each instrument shall have its own source identifier as defined below:

- 'SEPD' (EPD)
- 'SEUI' (EUI)
- 'SMAG' (MAG)
- 'SMET' (METIS)
- 'SPHI' (PHI)
- 'SRPW' (RPW)
- 'SSPI' (SPICE)
- 'SSHI' (SOLOHI)
- 'SSTX' (STIX)
- 'SSWA' (SWA)

The following table lists the allowed source for each CRF Type:

CRF Type	Allowed Source
SOR	SFCT, SSGS
POR	SSGS, SEPD, SEU, SMAG, SMET, SPHI, SRPW, SSPI, SSHI, SSTX, SSWA
FDR	SFDR
DOR	SFDR
MDOR	SEPD, SEU, SMAG, SMET, SPHI, SRPW, SSPI, SSHI, SSTX, SSWA
PDOR	SEPD, SEU, SMAG, SMET, SPHI, SRPW, SSGS, SSPI, SSHI, SSTX, SSWA

Table 2 - CRF Type vs Source

The ESOC sources ('SFDR' and 'SFCT') can make use of details derived from the local machine on which the CRF is generated in order to ensure uniqueness across different MCS clients.

For a CRF of type 'POF', the '`uniqueID`' field shall reflect the unique system ID value set in the MPS for the occurrences. In case of CRFs with several (repeating) instances of the same occurrence, the MPS ensures uniqueness of each instance, by assigning appropriate values. When having re-occuring facts or automatically created facts, then the first 4 characters will identify the MPS component that generated that fact, to avoid any possibility of duplication between the components:

- 'MPSF' (if the fact was created by the MPSF sub-system)
- 'RULE' (if the fact was created by the Rule engine)

Note: The source specified within the first 4 characters of the uniqueID does not need to match the source element field of the occurrence. For example, a POF will contain 'MPS_' within the source field, while the uniqueID could start with 'SSGS' as first 4 characters (maintained from the original uniqueID of the corresponding MPS input file).

3.2.5 Use of the delete flag

The insertOrDeleteFlag indicates whether the request is to insert or delete an occurrence.

The **insertOrDeleteFlag** shall normally take the value 'Insert'. If not present, the Data Systems will consider it as 'Insert'.

Use of this flag to request the deletion of previously inserted occurrences is only allowed for CRFs processed by the MPS (i.e. SOR, FDR, POR). All occurrences details specified for occurrence with **insertOrDeleteFlag** set to 'Delete' must be identical to the corresponding (previous) 'Insert' request.

For a CRF of type 'POF', the '`insertOrDeleteFlag`' field shall reflect the value set in the MPS.

3.2.6 Profiles (Z-Records)

For the Solar Orbiter, the supported profile types are defined in PLID Annex B, MOC-SOC ICD **[**A-8**].**

The use of these profiles is only foreseen for PORs.

4 MANIFEST FILES

The Manifest File contains the list of the delivered files and shall be used as specified in the PF-ICD [A-1]. The file type for manifest files is `MAN_` .

5 COMMAND REQUEST RESPONSE FILE (CRR)

Command Request Response Files will be generated as specified in the PF-ICD [A-1].

The only difference imposed by the Solar Orbiter tailoring concerns the time format to be used for the field genTime of the CRR header:

Field	E/A	Type	Description	Need
genTime	E	string	File Generation time, in the format: YYYY-DDD**T**hh:mm:ss[.mmm]**Z**	M

6 COMMAND STACK FILES (CSF)

Command Stack Files will be generated as specified in the PF-ICD [A-1].

Solar Orbiter extends the possible values for field CATEGORY of a CSF Base Header Record compared to what is specified in PF-ICD [A-1], s. 3.5.2.2, "Base Header Record" as follows:

Category Number	**Category Type**
0	*Not used in Solar Orbiter*
1	*Not used in Solar Orbiter*
2	*Not used in Solar Orbiter*
3	No constraints imposed on the command/sequence records.
4	All command/sequence records associated to an ASAP or delta release time. Each command is immediate (non time-tagged). This is termed '**Type 1 CSF**' in the Solar Orbiter documentation.
5	All command/sequence records associated to an ASAP release time. Each command is time-tagged and targets the MTL, or it is an immediate service 11 MTL management command. This is termed '**Type 2 CSF**' in the Solar Orbiter documentation.
6	All command/sequence records associated to an ASAP release time. Each command is time-tagged and targets the Backup MTL. This is termed '**Type 3 CSF**' in the Solar Orbiter documentation.

7 EVENT FILES

All event files defined for the Solar Orbiter mission are based on the template defined in the PF-ICD [A-1] for event files, instantiated as defined in this section.

There are two sources for event files on Solar Orbiter: Flight Dynamics and the Mission Planning System. The generation of event files by Flight Dynamics is ruled by the PLID Annex A (Flight Dynamics Interface ICD) [A-7]. The generation of event files by the Mission Planning System is ruled by PLID Annex B (MPS Output ICD) [A-8].

7.1 The Event File Root Element

Each event file shall have single root element named 'eventfile'. The root element is a container element containing the 'header' and the 'events' as specified below.

Field	E/A	Type	Description	Need
eventfile	E	container	The XML root element. Contains the 'header' and the 'events' elements.	M

7.2 Header

Each 'header' element shall be structured as follows:

Field	E/A	Type	Description	Need
gen_time	A	time	File Generation time, in the format: YYYY-DDD**T**hh:mm:ss.mmm**Z**	M
validity_start	A	time	The validity start time of the file shall correspond to the time of the earliest event contained in the file.	M
validity_end	A	time	The validity end time of the file shall correspond to the time of the latest event contained in the file.	M
spacecraft	A	spacecrafts	SOLO	M
icd_version	A	string	A reference to the specific document which is applicable for the events defined in this file.	M
format_version	A	string	The format version as defined by this ICD. Currently 1.	M
~~**purpose**~~	~~A~~	~~string~~	~~This wil lbe specified in the PLID section instatiating the single event file~~	~~O~~

7.3 Events

The 'events' element is a container for all events of the file.

Field	E/A	Type	Description	Need
events	E	container	Contains the events as child elements.	M

All events are included in the event file as child elements of the 'events' element. The chosen element name shall describe the nature of the event (e.g. aos/los).

These events are specified in dedicated parts of the PLID, but have to follow the structure and rules provided in this section. In addition, for some events (e.g. AOS/LOS), some characteristics (e.g. name, attributes) are defined which shall be followed by all missions as specified in PF-ICD.
Each '**event**' definition shall be based on the generic definition in the table below.

Field	E/A	Type	Description	Need
id	A	String of 4 characters	Defined in the dedicated ICDs	M
time	A	time	Time of the event in the format: YYYY-DDD**T**hh:mm:ss.mmm**Z**	M
count	A	unsigned long	integer counter for events with same ID, increasing by one for each new instance of the event.	M
Any attribute	A	N/A	Any attribute useful to describe the event can be added to the event. Each attribute has to be given a type and a name.	M

Note: the inclusion of a duration attribute is considered useful for most events. When used it shall be rounded to the closest integer number of seconds. In case of an event where the duration field is mandatory and the event file validity range does not allow its determination then the value -1 shall be used.

8 OPERATIONAL ASSUMPTIONS AND CONSTRAINTS

1. FCT will use GFTS [A-2] for the transfer of CRFs.
2. Flight Dynamics will use sftp to a GFTS node [A-2] for the transfer of CRFs and event files.
3. SGS and PIs shall deliver their inputs via sftp to a GFTS node for forwarding to their final destination, as specified in the SOIA [A-4].
4. For all received request files, an acknowledgement of receipt and validation, in the form of a CRR file, is sent back to the originating source. Acknowledgements of grouped requests (in a ZIP file) are sent back as a group.
5. Event files are made available to all external users via the DDS.

9 FILE DELIVERY PROCESS

9.1 CRF Naming Convention

It is imposed for Solar Orbiter, that **all** CRF submissions, and **all** resulting responses (I.e. CRR files) will make use of the ZIP file mechanism. A delivery of CRFs shall contain a manifest file, as described in section 3.2 of [A-1], even in the case where a single file is transferred.

Regarding files exchanged between MOC and SOC, the naming conventions are covered by the applicable PLID Annex B (MOC-SOC ICD), see [A-8].

In general, the file naming convention for deliverables is as follows:

`<type>_<source>_<plancycle>_<optional text>_<counter>.<EXT>`

Where:

`<type>` is a length string of 4 chars that defines the type of file. The allowed values are:

CRFG	ZIP file containing delivery of CRFs
PORG	ZIP file containing delivery of PORs
DOR_	Direct Operation Request
PDOR	Payload Direct Operation Request
MDOR	Memory Direct Operation Request
FDR_	Flight Dynamics Request
POR_	Payload Operation Request
SOR_	Spacecraft Operation Request
POF_	Planned Operations File
CRR_	CRF Response file
CRRG	ZIP file containing delivery of CRRs
MAN_	Manifest File

<source> identifies the entity that created the file, according to the allowed values specified in Sect. 3.2.4.

<plancycle> identifies the planning cycle for which the file is applicable. It shall conform to the pattern XYYY, where:
X identifies the plan cycle type as follows:

- S for STP
- M for MTP
- V for VSTP
- T for Test

YYY stands for the target plan cycle number.

<optional text> is a variable length string.
For files of type FDR and DOR, it shall contain the activity description as per MOD-FDS ICD [A-6].
For the POR file type, it shall conform to the pattern XX where XX is the short payload code as used in database mnemonics.
For other file types, this string can be used to provide more information about the contents of the file.

<counter> is defined as a 5-digit numerical string identifying the file instance. This number should start at 1 and increase by 1 every time a new instance of the same file type is produced by a given source. Gaps in the counters of submitted files are allowed, for example in case of file regeneration at the source.

<EXT> shall be `ZIP` for file types CRFG and CRRG, and `SOL` for all other file types.

Note: The naming convention for CRR files is to use the same name as the file to which it is responding, but with the `<type>` name at the start changed to `CRR`.

Examples of Valid Filenames

```
CRFG_SSGS_S123_All_PORs_00123.ZIP
POR__SSGS_S123_PH_01300.SOL
DOR__SFDR_M027_NMOCM_09871.SOL
```

9.2 Event File Naming Convention

The naming convention of event files defined by Flight Dynamics is defined in [A-7] for event files generated by Flight Dynamics and in [A-8] for event files generated by the Mission Planning System.

9.3 Failure Protection, Detection and Recovery Procedures

- *Data integrity checks:* An error free file transmission, ensured by the mechanisms and protocols used by sFTP and GFTS is assumed. In addition, a manifest file is always provided which will provide a means for the receiver to further check the integrity of the delivered products.

- *Recovery:* For deliveries from external users, if the sFTP server is temporarily not available, it is assumed to be brought back to operation within 1 working day.
  This delay is expected to be compatible with the planning cycles defined on Solar Orbiter. SGS inputs are expected to be provided at the end of the (working) week before STP starts. But MPS activities only start after provision of the FD STP inputs on Tuesdays, i.e. there is one working day of margin between provision of SGS inputs and the start of their processing by the MPS. Finally, there is also one day of margin in the cycle for uplink of the commanding products to the spacecraft. This shows that one day maximum outage of sFTP for provision of external products can be absorbed by the STP cycle.

  For the ESOC-internal nodes on which GFTS is installed, automatic recovery will be performed after a link drop.

- *Security Requirements:* The ESOC-external systems will be prevented from accessing ESOC's operational LAN by a Firewall. All transfer of data between the ESOC-external systems and the MCS/MPS will be via the mission specific EDDS, which shall be installed on a node located on ESOC's Front-end RelayLAN or the GFTS. The protocol used will be secure FTP (sFTP) which allows secure file access, file transfer, and file management functionality over a reliable and secure data stream such as SSH. The used sFTP account should be private (not public or guest) and protected by password.

9.4 Storage and File Deletion Requirements

Sufficient space must be available on each of the nodes to hold the relevant data. This means that each of the node used for exchange of CRF must be sized according to the volume expected to be exchanged.
Incoming files will be deleted from the agreed in-tray directory on the EDDS once they have been processed by the MCS/MPS.

10 CRF WHITELISTS

A CRF whitelist is a simple list of allowed command and sequences IDs. Multiple CRF whitelists can be defined such that there can be a whitelist file for each possible commanding source (see list within source field in section 3.2.1.2). A CRF whitelist file can contain multiple lines of comments marked at the beginning with the character "#".

The CRF whitelist files are expected to follow the naming convention:

cmdSeq<SOURCE>WhiteList.dat

where:

- <SOURCE> is a valid commanding source type as defined in section 3.2.1.2

A correctly formatted CRF whitelist file will simply contain one valid command sequence name per line. Wildcards are allowed within the command sequence name.

For example, file cmdSeqSFCTWhiteList.dat could contain:

```
AACC360A
AACC360B
AACF*
```

These whitelists will be handled by the Flight Control Team. It is not up to the 'sources' to provide the whitelists.