DOCUMENT

Solar Orbiter Data Delivery Interface Control Document

Prepared by	**L. Michienzi**
Reference	**SOL-ESC-IF-05011**
Status	**Final**
Issue	**1**
Revision	**2**
Date of Issue	**26/06/2019**

European Space Agency
Agence spatiale européenne

APPROVAL

Paula Margarido Digitally signed by Paula Margarido
Date: 2019.06.26 15:15:18 +02'00'

Prepared by: ____
L. Michienzi and P. Margarido (OPS-GDS)
Solar Orbiter Data Systems Manager

Approved by: ____
Sylvain Lodiot (OPS-OPS)
Solar Orbiter Spacecraft Operations Manager

Approved by: ____
Alessandro Ercolani (OPS-GDS)
Head of Science Data Systems Section

Approved by: ____
Andrea Accomazzo (OPS-OP)
Solar Orbiter Ground Segment Manager

Approved by: ____
D. Mueller (SCI-S)
Solar Orbiter Project Scientist

Approved by: ____
L. Sanchez (SCI-ODS)
Solar Orbiter SGS Development Manager

CHANGE LOG

Reason for change	Issue	Revision	Date
First issue for external review	1	0	10/09/2013
Draft release 1 (internal only)	1	1	10/07/2015
Tidy up and alignment with BepiColombo DDID	1	2	26/06/2019

CHANGE RECORD

Date	Issue No.	Rev. No.	Description and Page No.		
10/09/2013	1	0	First issue for external review		
10/07/2015	1	1	DRAFT: - added section 7 Data Definition - resolved TCO TBC packets data		
11/02/2019	1	2	Removed the TBCs/TBWs throughout the document. Updated distribution list. Section 2 updated with clarification on the possibility of using ssh keys for file server delivery. DDS servers IP's to be added to the user's server whitelist. Section 4 updated – added known issues when accessing the DDS servers. Section 8.4.1 – Auxiliary File Types to be available through DDS servers reviewed. Appendix A – details on time correlation removed. This section now refers to A-8 the BepiColombo Time Correlation Approach and Accuracy document		

DISTRIBUTION LIST

Recipient	Organisation	Recipient	Organisation
C. Garcia	SCI-PS	P. Ferri	OPS-O
Y. Bonnefous	SCI-MC	P. Bargellini	OPS-ON
S. Strandmoe	SCI-PSS	M.Butkovic	OPS-ONO
C. Hass	SCI-PSV	T. Beck	OPS-ONX
A. Pacros	SCI-PS	Y. Doat	OPS-ONI
P. Pablos	SCI-PS	E. Vassallo	OPS-OW
L. Icardi	SCI-PS	O. Procope	OPS-ONC
Pierre Olivier	SCI-PSQ		
A. Oganessian	SCI-PSA	E. Dölling	FCI-ITC
J. Ramachandran	SCI-PSS	K.Robinson	OPS-ONC
K. Wirth	SCI-PSS	N. Bobrinsky	OPS-G
		M. Merri	OPS-GD
D. Mueller	SCI-S	A. Ercolani	OPS-GDS
L. Sanchez	SCI-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
C. Guerbuez	OPS-OPS	M. Lanucara	OPS-GSY
J. L. Pellon-Bailon	OPS-OPS	M.A. Larsen	OPS-GSC
M. Lauer	OPS-GFS	A. Mantineo	OPS-Q
V. Companys	OPS-GFS		
J. Schoenmaekers	OPS-GF	EILSservices@esa.int	
J.M. Sanchez Perez	OPS-GFA	A. Urwin-Phipps	SRE-PW
		I. Tanco	OPS-OPJ
G. Ravera	OPS-ONX	E. Montagnon	OPS-OPB
T. Anic	OPS-BA	P. Schmitz	OPS-OPE
M.H. Ferreira	OPS-Q	EOS_SOLARDOC@astrium.eads.net	
H. Sillack	OPS-OP		

Table of contents:

1 INTRODUCTION

1.1 Objective

The Solar Orbiter Data Delivery Interface Document is a mission-specific complement to the generic External User ICD (MDS-MCS-SW-ICD-1001-OPS-GD), referred to here as [A-1] which contains the details of the various data delivery interfaces between external users of the EDDS, such as the SGS and the Solar Orbiter MOC. These documents should be read together.

1.2 Scope

This document is primarily concerned with the specification of the request and delivery mechanisms and data products available to an external user of the Solar Orbiter Data Dissemination System (DDS), which is based on the generic EDDS infrastructure software.

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

This document only provides Solar Orbiter specific information and details any changes or additions with respect to what is provided by the generic EDDS.

The configuration and usage of the EDDS Client which is the interface to be used by external parties to the Solar Orbiter DDS, is described in [A-2].

1.3 Applicable and Reference Documents

1.3.1 Applicable Documents

ID	Document Title	Document Reference
A-1	(EDDS) External User ICD	EGOS-GEN-EDDS-ICD-1001
A-2	(EDDS) Software User Manual	EGOS-GEN-EDDS-SUM-1001
A-3	(Solar Orbiter/BepiColombo) Packetiser ICD	BC-ESC-IF-22002
A-4	Solar Orbiter Planning ICD (PLID)	SOL-ESC-IF-05010

A-5	DDID External User Details[1]	N/A
A-6	BepiColombo MCS S/W Design Document (SDD)	BC-ESC-DD-22001
A-7	Command Source Handler ICD	GAIA-ESC-ICD-0015.03
A-8	BepiColombo Time Correlation Approach and Accuracy	BC-ESC-TN-05004

1.3.2 Reference Documents

ID	Document Title	Document Reference
R-1	(EDDS) XFDU Formatting TN	EGOS-GEN-EDDS-TN-1003
R-2	EDDS Software Requirements Specification	EGOS-MDW-EDDS-SRS-1001
R-3	NIS/NCTRS Vol2 ICD for MCS	EGOS-NIS-NCTR-ICD-0002
R-4	SSMM TMTC ICD	SOL-A-TAS-ICD-00001
R-5	CSW TMTC ICD	SOL.S.ASTR.ICD.00030
R-6	Space-Ground ICD	SOL-ESC-IF-05002, (Space-Ground ICD)_
R-7	SOL PLID Annex A - Flight Dynamics ICD	SOL-ESC-IF-50001 PLID Annex A FD Interfaces
R-8	SOL PLID Annex B - Mission Planning System Outputs ICD	SOL-ESC-IF-05012 (PLID Annex B-MOC SOC ICD)

[1] This document is a password protected file containing sensitive information under OPS-GDS team responsibility.

R-9	SCOS-2000 Packet Management ADD	EGOS-MDS-S2K-ADD-0024
R-10	SCOS-2000 MIB ICD	EGOS-MCS-S2K-ICD-0001
R-11	SCOS-2000 Commanding ADD	EGOS-MCS-S2K-ADD-0002
R-12	MPS Planning Database ICD	BC-ESC-IF-23002
R-13	End User Needs for Spacecraft Telemetry Parameters	BC-SGS-TN-063
R-14	BepiColombo and Solar-Orbiter MCS SRS	BC-ESC-RS-22002
R-15	RAPID ICD	EGOS-MCS-S2K-ICD-1005

1.4 Acronyms

DARC	Parameter Archive
DDS	Data Dissemination System
EDDS	EGOS Data Dissemination System
ERT	Earth Reception Time
ESOC	European Space Operations Centre
FARC	File Archive
FCT	Flight Control Team
GFTS	Generic File Transfer System
GPB	Google ProtoBuf
HK	Housekeeping
ICD	Interface Control Document
LTA	Long Term Archive
MCS	Mission Control System
OGS	Operational Ground Segment
OBT	On-Board Time
PI	Principal Investigator
PTV	Pre-Transmission Verification
PUS	Packet Utilisation Standard
RAPID	Raw Archive Packet Interceptor Dumper
S/C	Spacecraft
SGS	Science Ground Segment
TC	Telecommand
TM	Telemetry
UTC	Universal Time Coordinated
VC	Virtual Channel

XFDU XML Formatted Data Units

1.5 Roles

SolarOrbiter DDS administrators	The administration tasks of the DDS will be handled by the software support and software coordinator teams. The email aliases bs_mcs_sws@esa.in can be used.

2 DATA EXCHANGE SUMMARY

Data will be distributed from the OGS to the SGS and/or Principal Investigators.

The primary mechanism for distributing data is via the use of the generic ESOC Infrastructure DDS software known as the EDDS. This application provides a secure HTTPS web-based interface for requesting Spacecraft Telemetry packet and parameter data, Telecommand history, event data (both Spacecraft and internal MCS events) and any configured auxiliary files stored in the MOC mission archive. It also provides catalogue information. Requests can either be to open a stream or to deliver a file containing the requested data submitted as batch requests, depending on user privilege and configured quota. These can be one off requests, which can be saved for re-use later, or they can be scheduled. The response data can then be transferred directly to the user's file server via SFTP or it can be stored on the DDS machine for later download.

When using the file server delivery, the user shall ensure that the IP addresses of mmaasweb01.esoc.ops.esa.int (195.74.163.52) and mmaasweb02.esoc.ops.esa.int (195.74.178.8) are in the sFTP server's whitelist. This avoids that the DDS servers get banned from the user's sFTP server, which would prevent the file delivery.

Furthermore, if the file server delivery option is used, and the user would like to make use of a SSH key instead of providing the file server password in the request file, then ESA can provide the user with a public SSH key. This public key shall be added to the .ssh/authorized_keys of the user's sFTP server.

Usage of this interface is expected to be performed via the EDDS Client application software described in [A-2], which can be downloaded and installed (using Java web-start technology) directly from the EDDS site (see section 5.1 for details). This application allows the saving of request templates which can then be re-used, but also provides functionality to set up scheduled requests. In addition, if required requests can be submitted in the form of XML files via SFTP directly to the OGS EDDS relay server.Alternatively, external users may implement their own EDDS client, which supports the EDDS interface as described in [A-1].

Users shall also be able to use the EDDS to subscribe to any auxiliary files that are received by the mission file archive. That is, whenever a new file or an update to an existing file is produced and submitted for storage to the MOC file archive (both from an external or an internal source) it can be configured to be delivered to a subscribed user, using the same mechanisms as are available for standard auxiliary file requests supported by the EDDS.

The secondary means for distributing data is via the use of the GFTS infrastructure software, which allows to forward RAPID binary files. RAPID files contain all packet data which is received processed and archived by the Mission Control System. This will be the nominal distribution method between OGS and the SGS.

2.1 DDS Requests from DDS User

Requests to the DDS can be generated by the EDDS client application which provides an interface to make data requests (both batch and streamed) and if the appropriate privilege is granted, user management requests, e.g. to create new user accounts etc. Requests are generated in the form of an XML file (see the schema definitions[2] in the appendices of []), which can also be sent directly to the DDS server via the EDDS Web server.

2.2 DDS Data Response to DDS User

The DDS supports several data types, including packets, parameters, reports and files, which are described in chapter 5 of the[A-1]. The responses to the requests made by the user to the DDS can be delivered in several ways, depending on the format of the data. This is detailed in chapters 6 and 7 of [A-1].

[2] *For convenience, EDDS version 2.3 schema definitions are also attached to the PDF version of this DDID*

3 OPERATIONAL ASSUMPTIONS AND CONSTRAINTS

3.1 Assumptions

- External access between the OGS and all external users has been authorised and the necessary firewall configurations applied.
- A platform with Web access (i.e. a suitable internet browser) and running Java Runtime Environment 6 or higher is installed wherever the EDDS client is expected to execute. (The baselines supported by Infrastructure today are SLES-11, SLES-12 and Windows XP, but will likely work on others)
- A suitably configured (i.e with external write access) repository containing directories and adequate disk capacity for storing requested data transferred by the EDDS to the requesting source, is available at each external location. The exact disk capacity required is dependent on the size of the data requested, the frequency of requests and the data management policy of each individual user.
- An SFTP server is available on the external machine expecting to receive and/or send auxiliary file data, to or from the OGS.

-

3.2 Constraints

In order to reduce the risk of overloading the DDS and affecting its performance and the quality of service it provides, a set of constraints may be applied to each user of the DDS.

- The DDS supports a quota allocation feature which allows the Solar Orbiter DDS administrator to restrict user access in terms of:the type of data that can be retrieved (i.e. filtering by APIDs, SPIDs, parameter names, file names and/or file types),
- the amount of data requested over a given period,
- a time range over which the data may be requested
- or the period of time the user has to access the data.

4 DDS INTERFACE CHARACTERISTICS

The DDS server which will offer the EDDS services described in [A-1] will be located on the DDS prime and backup servers. It is to be noted however, that only one of these servers will be available at a time since the redundant DDS server will be available in cold redundancy only.

The DDS server will host the EDDS delivery manager and web server processes and will be configured to access the mission telemetry, parameters and file archives of the MCS.

Requests can be made from the external user machines directly via the EDDS client application, or submitted as XML request files via SFTP to the DDS server directly.

Solar Orbiter and Bepicolombo will share the edds backup server while the edds prime server will be hosted in the Multi Mission as a server machine.
The EDDS client web interfaces can be accessed using the following URL (from which the User can either download[3] or start directly through the web page the Client Application) :

- https://mmaasweb01.esoc.ops.esa.int/edds for the prime server
- https://mmaasweb02.esoc.ops.esa.int/edds for the backup server

The SFTP interface can be accessed using the following link:

- sftp://mmaasweb01.esoc.ops.esa.int/ (for the prime server)
- sftp:// mmaasweb02.esoc.ops.esa.int/ (for the backup server)

The response data will be delivered either using https to the requestor's client display or file server or via e-mail depending on what delivery mechanism was requested.

Data delivered by the DDS via SFTP will come from one of the following IP addresses:

- SolarOrbiter DDS server A: 195.74.163.52
- SolarOrbiter DDS server B: 195.74.178.8

The EDDS also supports the capability to store data temporarily on the DDS server, to be downloaded at a later date. However, this is not foreseen to be used as a baseline mechanism on SolarOrbiter as regular disk space maintenance will be performed on the machine to ensure that a disk does not fill up and prevent the DDS server from servicing further requests. Users interested in this storage option shall contact the SolarOrbiter DDS administrator.

Nodes, device addressing and authorised machines will be defined and maintained in [A-5].

It is to be noted that, when accessing the DDS servers, certain known issues could occur. In this case, the users are requested to take the following actions:

- **SSL Certificate** – While the SSL certificate is not installed on the DDS machines or in case this certificate expires before its renovation, the users will have to accept the security warning issued by the browser in order to proceed to the EDDS web page.

- **Issues with the DNS propagation** – Some users reported difficulties to access ESOC EDDS based on the machine name. As a work-around, to be able to access the DDS machines using the name, it may help to add an entry in the /etc/hosts file on the source machine, with the full hostname and the IP address of the DDS machines.

[3] Please note that a version for MacOS is also available, but officially NOT supported.

#MMAASWEB01:
mmaasweb01.esoc.ops.esa.int 195.74.163.52

MMAASWEB02:
mmaasweb02.esoc.ops.esa.int 195.74.178.8

If for some reason the /etc/hosts files cannot be edited, as an alternative, the users will still have the possibility to access the DDS servers using the IP address and the complete path to start the EDDS MMI:

Prime DDS – MMAASWEB01:
https://195.74.163.52/eddsweb/eddsMmi

Backup DDS - MMAASWEB02:
https:// 195.74.178.8/eddsweb/eddsMmi

5 DDS ACCESS

5.1 Interface Utility Software

For access to the DDS, an EDDS Client application is provided as a downloadable executable from the DDS server at the following URL:

- https://mmaasweb01.esoc.ops.esa.int:8443/edds/edds_runtimes.jsp

5.2 Failure protection and Detection

- For DDS see [A-1] .

5.3 Recovery Procedures

See [A-1] for details of the DDS recovery procedures in case of errors to create response data or in delivering the data back to the requestor.
The software recovery procedures, to be performed by the Software Support Team, will be documented in the SolarOrbiter MCS Software Support Procedures document.

5.4 File Naming Convention

The file naming convention for each delivered auxiliary file requested from the mission File Archive by the DDS is described in its appropriate ICD. E.g. for PORs see [A-4].

The file naming conventions for the EDDS request and response files is covered in chapter 7 of [].

5.5 Storage Requirements

The disk capacity of the DDS at the MOC shall be sized in order to hold a reasonable amount of data, including auxiliary files, housekeeping and payload telemetry, event and commanding data.

Regarding the DDS relay server, it should be noted that it is not foreseen as a data repository, but rather as a gateway to access data from the mission archives on the LTA. The disk capacity of the DDS at the OGS is only to store the data retrieved from the mission archives in response to requests, if the user has not requested it to be sent directly to them (the latter being the nominal case as explained in section 4.
Regarding the storage of RAPID files (see chapter 10), two weeks of RAPID data shall be kept in order to be able to recover in case there is a problem with the file forwarding to SGS using GFTS.

5.6 Security Requirements

The MCS LTA server is located on the operational LAN protected by the OPSLAN firewall. The DDS server is located on the Relay-LAN which is protected from the internet by the corporate firewall.

For details concerning security issues applicable to the DDS, see [R-2].

5.7 Data Integrity Check

No data integrity checks over and above those provided by the EDDS are foreseen. These are supported by the EDDS client MMI (e.g. only allowing certain values from drop-down menus etc.) and by validating the resulting request XML file against a schema.

5.8 Backup Requirements

The DDS server, shall have a prime and backup node in case of hardware failure. The backup will be a cold redundant machine, which means that in case of a failure with the prime DDS server, the backup will be switched on, configured as appropriate and the DDS software started.
No data will copied from the prime, and any requested data which was temporarily stored would need to be re-requested if needed. In most cases non-scheduled requests submitted prior to the failure will still be delivered, but this depends on the exact status in the process the request was in when the prime stops working. Scheduled requests would need to be re-submitted. There is no requirement for a standby backup or for automatic switch over.
Due to the fact that the web server for the DDS runs on the DDS server, the user would need to re-configure their EDDS client to point to the backup machine after such switch, i.e. changing the URL in their browser (see section 4).

In case of problems with the back-end, i.e. the data providers on the MCS, then the redundancy features of the MCS will be used, which also consist of re-configuring the DDS to point to the appropriate backup data source.
Note: The requirement for the availability of the EDDS is to be at least 95% over a period of one month (including maintenance downtimes).

Although, EDDS provides a service to list the DDS status, availability and planned downtimes via the EDDS website, this feature will not be used, since there is no automatic update mechanism between MCS and DDS to maintain this status page. Notification about downtimes and maintenance will therefore be done via a mailing list specified in [A-5].

5.9 Error Handling

No error handling above those provided by the EDDS are foreseen. The various error cases and the mechanism by which the EDDS logs them and/or informs the users, are described in [A-2] under various sections depending on the functional area causing the error.

5.10 Firewall Information

Information required in order to configure firewalls on both sides (External users and OGS) are specified in [A-5].

6

6 DETAILED DDS INTERFACE SPECIFICATIONS

6.1 Generation Method

The DDS data responses are generated by retrieving data from the Solar Orbiter MCS LTA server. In case the data request is for packet telemetry, MCS generated data (e.g. time correlation packets) events or telecommand data from the packet archive (PARC) on the LTA server will be interrogated. Requests for auxiliary files will be made on the File Archive (FARC) of the LTA server. Finally, any telemetry parameter will be retrieved from the LTA's parameter archive (DARC). The DDS will then process the retrieved data and convert it into the format requested.

Auxiliary files generated by various sources in the SolarOrbiter Ground Segment will be filed in the mission File Archive. A selected sub-set of file types is configured to be pushed automatically to each external user. This will be setup using the generic file transfer system (GFTS). Either the external user can retrieve the files from ESOC's relay lan via (s)ftp, or the files can be pushed directly onto an (s)ftp at the external user's premises.

In the case of spacecraft telemetry data, the data will be available for retrieval as soon as stored at the MCS LTA, which will be normally within seconds of reception at the ground station.

6.2 Data passed through the Interface

The data, which is passed across the EDDS interface, is summarised in the table below.
Note that the Telemetry packets that are downlinked via service 13 will be available as TM files (see also **Error! Reference source not found.** below).

Data Type		**Method**	**Source**	**Described in**
Packet	TM packets[5]	DDS	LTA PARC	7 and 8.1
	TC packets	DDS	LTA PARC	7 and 8.1
	Event packets	DDS	LTA PARC	7, 8.1 & 8.4.2.2
	TM, TC, Event Statistics	DDS	LTA PARC	[A-2] 5.2.3
Parameter	TM parameters	DDS	LTA DARC	8.2
	Parameter Statistics	DDS	LTA DARC	[A-2] 5.2.2
Report	Command History Report	DDS	LTA PARC	8.3.1
	TM Gap Report	DDS	LTA PARC	8.3.2
	TM, TC, Event Reports	DDS	LTA PARC	[A-2] 5.2.3
Archived Files	Auxiliary Files	DDS	LTA FARC	8.4.1
	TM Files	DDS	LTA FARC	8.4.2
	Archive Catalogues	DDS	LTA FARC	[A-1]6.2.4 [A-2] 5.2.1

The EDDS provides access to data via two types of services: the batch service and the stream service.

[5] This includes internal MCS generated packets, e.g. user defined constant packets, time correlation packets etc.

6.2.1 Batch Service

See chapter 4.1.2 of [A-1].
The batch service is intended to allow clients to make requests for mission data and receive a data response that contains a complete set of data that matches the request. In general, a request lists the data types and allows the user to apply a set of filters for each data type. The response data is sent to the client via the delivery method indicated in the request.
A batch request is submitted as an XML document. The data response can take form of a binary, XML or ASCII file (optionally compressed or encrypted).

6.2.2 Stream Service

See chapter 4.1.3 of [A-1].
An EDDS stream service allows a client application to receive a continuous stream of mission data rather than a finite stored data set. This service is applicable when the end user needs to achieve near real-time delivery of data. A stream request is also submitted as an XML document. The data responses are described in the sub-chapters of 4.1.3 in [A-1].

6.3 Size and Frequency of Transfers.

It is expected that during the cruise phase most users shall only retrieve their own payload data plus a limited set of spacecraft housekeeping data. At Sun, the SGS shall retrieve the full data set and shall be responsible for distributing to the PIs. Therefore at this stage it is expected that requests from PIs shall be limited to unplanned requests for maintenance engineering or contingency.

The size and frequency of transfers of data using the DDS is dependent on the requests made and constrained by quota assignment. Each user account shall be configured to have access to a restricted set of data, which in most cases will be the associated payload data and a sub-set of spacecraft housekeeping data, although, in some cases (i.e. for the SGS) full access to the housekeeping archive may be given. In addition, in order to avoid misuse or accidental requests being submitted for large amount of data which may be have a non-negligible performance effect on the operational systems, each user will be assigned a relative time-range over which he/she may request data. For example, a user may be time-restricted to only be able to request the last 12 months of data.

During the first months of the mission, the DDS usage shall be monitored by the SolarOrbiter DDS administrator and appropriate action may be taken if there is misuse of the system resulting in a degradation of the service for other users. This may be enforced by applying greater restrictions on access in terms of data type, volume and/or time.

7 DATA DEFINITION

Depending on the type of DDS requests, the requester will have the choice between different delivery formats. Not all request types support all response delivery formats. The response formats that are supported by each request type are summarised in chapter 6, table 3 of [A-1].

As described in more details in [], the DDS supports the following delivery format options:

- **Binary**:
  - o either physical files
  - o or data encoded using the Google Protocol Buffers (ProtoBuf) binary format, for which the .proto definitions of all possible response types are described in chapter 6.7.3 of []. It should be noted, that in the case of 'Packet Raw', the packet string that will be extracted from the ProtoBuf binary format, will correspond to an ASCII string representing the HEX packet data including the SCOS-2000 headers described here-after.
- **EDDS Raw Binary**: corresponds to an ASCII string representing the HEX packet data including the EDDS Raw header, described in chapter 6.1 of []. Regarding the header format of 'Packet' EDDS requests, using this EDDS Raw binary format, there will be no extra EDDS raw header fields configured (see chapter 6.1.2 of [A-1]). This means that no SolarOrbiter specific packet header information is envisaged to be added and the value N in chapter 6.1.1 Table 5 of [] will be zero. The packet would therefore start with '00' followed by the EDDS Sequence Count, the Data Length and then the actual source packet (excluding any SCOS-2000 header).
- **GDDS Binary**: corresponds to actual binary encoded data with a header, described in chapter 6.1.3 of [], and followed by the actual source packet (excluding any SCOS-2000 header). *Note that the 'SLE service Id' field of the GDDS binary header, identifies how the data was retrieved over the SLE service channel (i.e. via which SLE service and what channel type was used). This specific field is not believed relevant to the end user.*
- **XML** (and XML Transform): described in chapter 6.2 and 6.3 of []. The content of the *Packet* element of the *PktRawResponseElement* will be an ASCII string representing the HEX packet data, including the SCOS-2000 headers described here-after, within this chapter 7 and sub-chapters.
- **ASCII**: described in chapter 6.6 of [].

The SolarOrbiter EDDS will be configured such, that the byte order of all data available via the DDS is in big (network) endian format, i.e. the most significant byte (MSB) value, is stored at the memory location with the lowest address.

All fields below of type time specify an absolute date which conforms to the following format:

The first 4 bytes data is an integer containing the number of seconds since the 00:00:00 1st January 1970 epoch. However, this does not take into account any leap seconds, because for every leap second added since 1970, this number should be incremented by one, and for every leap second removed it should be decremented (though no leap seconds have been yet removed since 1970), in order to give the real number of seconds. The second 4 bytes data is an integer containing the number of microseconds to be added in case the date is not an integer number of seconds since the epoch.

7.1 SCOS-2000 Packet Format Definition

For a detailed description of the SCOS-2ooo packet, also known as a TDEV packet see chapter 5.2 and 5.3 of [R-9]. Each SCOS-2000 packet consists of two headers. A '*common header*' described below, followed by a '*packet type specific header*', depending on whether the packet is a telemetry (TM), telecommand (TC) or event (EV) packet.

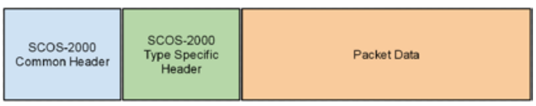

Figure 1: SCOS-2000 Packet structure

The '*packet data*' field then contains the actual packet.

In the case of spacecraft telemetry packets, this '*packet data*' will be the full un-stripped source packet complete with CCSDS/PUS primary header, secondary header (if applicable) and the application data. I.e. the first field after the SCOS-2000 header will be the Version Number of the Primary Header. In some cases the packet data may actually contain a full telemetry frame (see 7.3) or contain MCS internally generated data (see for example a subset in chapter **Error! Reference source not found.**). These internal SCOS-2000 'telemetry' packets are used for the MCS' own purposes, e.g. for system monitoring etc. These packets are stored on a separate datastream (see 'stream ID' below in chapter 7.1.1). These packets should be ignored by external users.

In the case of event packets, the '*packet data*' will contain MCS specific event message information. These are mostly out of scope of this document, except for the TM file associated meta data (see chapter 8.4.2.2).

In case of telecommand packets, the '*packet data*' will contain further MCS TC information used for the TC history application, including static information, the verification status, and finally the full packet which was sent to the spacecraft. It should be noted that in addition to TC packets for released telecommands, the MCS also stores a separate TC packet for each command in the archive containing static (found in the MIB) information concerning the PTV/CEV details of the command, e.g. parameters and expected values for parameter based verification. This packet, which is stored with a SPID of **200** (see 'Filing Key' below in chapter 7.1.1) should be ignored by external users.

7.1.1 SCOS-2000 Common Packet Header Structure

The following details are taken from chapter 5.3.1 of [R-9], but the descriptions simplified for the Solar Orbiter context. All values below are in decimal radix.
Default values for fields that are not used, or not applicable, will be zero.

Field	Type	Length (Bits)	Description
C-Tree	Fixed	16	Used in earlier version of SCOS-2000 by c-tree to support packets storing in files and be kept for backward compatibility.
Access Flag	Unsigned Int	8	Storing mode, values are **0** for an inserted packet and **1** for an update
Spare	N/A	6	Not used
Sim Flag	Unsigned Int	2	Simulated packet, values are **0** for a non-simulated packet generated by the spacecraft and **1** for a simulated packet generated by a Simulator or test tool
FT secs	Unsigned Int	32	Filing Time (in seconds since 1st Jan 1970 00:00:00) – this is part of the primary key for retrievals and corresponds to the 'Generation Time' in EDDS. For spacecraft TM, this is the timestamp and is dependent on the Timestamp and Time Quality fields here-after. For example, it will be the correlated on-board generation time (SCET) if the timestamp field is 2 and the time quality field is 0. For TCs, this is the release time.
FT usecs	Unsigned Int	32	Sub-seconds (in microseconds) of the Filing Time
CT secs	Unsigned Int	32	Creation Time (in seconds since 1st Jan 1970 00:00:00) – this is a secondary retrieval key and corresponds to the 'Storage Time' or 'Reception Time' in EDDS. Fand for spacecraft TM, it is the time at which the SCOS-2000 packet was created, which is also approximately the time that the packet was received by the MCS and stored. For TCs, this is its (estimated) execution time
CT usecs	Unsigned Int	32	Sub-seconds (in microseconds) of the Creation Time
Create ID	Unsigned Int	32	Used by the old SCOS-2000 archive for generating a retrieval key and set by the application which generated the packet
S/C ID	Unsigned Int	16	In the case of telemetry, Spacecraft ID – for Solar Orbiter this will be: **650**
G/S ID	Unsigned Int	16	In the case of telemetry, the numerical ID of the ground station which receives the frame containing this packet: **21** – Kourou **22** – Perth **23** – New Norcia **24** – Cebreros **25** – Malargüe **30** – Maspalomas as well as: **201** – Reference Test Station (ESOC) **202** – NDIU Lite **203** – NDIU IFMS
Size	Unsigned Int	32	Size of the entire SCOS-2000 packet, including all headers and the packet data
Type	Unsigned Int	4	Determines the type of SCOS-2000 packet: **1** – for TM **2** – for TC **3** – for Event
Version	Unsigned Int	4	The version of this packet structure – for Solar Orbiter this will be: **5**
Spare	N/A	2	Not used
Filing Flag	Boolean	1	Set to **1** if this packet is to be filed in the MCS archive, otherwise **0**
Dist Flag	Boolean	1	Set to **1** if this packet is to be distributed to the MCS applications, otherwise **0**

Field	Type	Length (Bits)	Description
Timestamp	Unsigned Int	2	Determines the timestamp policy for spacecraft TM, i.e. what will be set in the Filing Time fields above: **0** – packet is time-stamped with the creation time – i.e. SCOS-2000 packet reception and creation – PR **1** – that the packet is time-stamped with the time of transmission of the frame it was received in, i.e. Earth Reception Time minus the propagation delay (applicable to spacecraft TM only) – FT **2** – the correlated SCET (for spacecraft TM only) – PG *Note: it is useful to refer to chapter 4.2.5.4 of [R-14], in order to understand the timestamp and time quality flag settings depending on the packet's plausibility checks.* *Further Note: time-stamping by packet reception (0) occurs when the user selects this time-stamping mode explicitly, in order to by-pass time-correlation for example.*
Time Quality	Unsigned Int	2	Time quality of the timestamp, used only if the timestamp type above is set to **2:** **0** – Good **1** – Inaccurate **2** – Bad
Stream ID	Unsigned Int	16	The datastream identifier in the MCS, which for TM maps to VCs: **1000** – VC0 Real-Time Non-Science (online) **1001** – VC1 Playback Non-Science (online) **1002** – VC2 Science (online) **1003** – VC3 File-Transfer (online) **2000** – VC0 Real-Time Non-Science (offline) **2001** – VC1 Playback Non-Science (offline) **2002** – VC1 Playback Non-Science (offline) **2003** – VC2 Science (offline) For TCs stream **1** is used. For EVs stream **1000** and **1001** is used. **65535** – is used for internal non spacecraft TM packets, which should be ignored by external users(e.g. superseded TCO coefficients)
Seq Counter	Unsigned Int	32	Internal MCS counter used for detecting gaps – not related to the spacecraft source sequence count value in the PUS header
Filing Key	Unsigned Int	32	This is the SCOS-2000 SPID – the numerical packet ID which identifies the packet structure, and maps to packet definitions in the TM/TC MIB database. Note: For TC packets, SPIDs **100** and **200** are used (the latter for storing the command PTC/CEV details as described in [R-11]).
Retr Key1	N/A	16	Not used
Retr Key2	N/A	16	Not used
Mission ID	Unsigned Int	16	For telemetry, the MCS mission identifier – for Solar Orbiter this will be: **650**. It should not be used by external users.
Context ID	Unsigned Int	16	For telemetry, the MCS context identifier – for Solar Orbiter this will be: **0**. It should not be used by external users.
Domain ID	Unsigned Int	16	For telemetry, the MCS domain identifier – for Solar Orbiter this will be: **0**. It should not be used by external users.
DB Rel	Unsigned Int	16	For telemetry, the release number of the TM/TC MIB database used when the packet was archived (see VDF_RELEASE in chapter 3.3.1.1 [R-10]) . It should not be used by external users.

Field	Type	Length (Bits)	Description
DB Iss	Unsigned Int	16	For telemetry, the issue number of the TM/TC MIB database used when the packet was archived (see VDF_ISSUE in chapter 3.3.1.1 [R-10]) . It should not be used by external users.
Spare	N/A	16	Not used

7.1.2 *SCOS-2000 TM Packet Header Structure*

The following details are taken from chapter 5.3.4 of [R-9], but the descriptions simplified for the Solar Orbiter context.

Field	Type	Length (Bits)	Description
Validity	N/A	32	Not used
TPSD	Unsigned Int	32	Structure identifier for variable length packets (see PID_TPSD in chapter 3.3.2.4.1 [R-10])
Route ID	N/A	16	See below for usage of LSB of RouteID (MSB is not used).
PUS APID	Unsigned Int	16	The value of the 11-bit Application Process ID, copied from the spacecraft source packet's primary header (see [R-6])
PUS SSC	Unsigned Int	16	The value of the 14-bit Source Sequence Count, copied from the spacecraft source packet's primary header (see [R-6])
PUS Service Type	Unsigned Int	8	The value of the 8-bit Service Type, copied from the spacecraft source packet's secondary data field header (see [R-6])
PUS Service Sub-Type	Unsigned Int	8	The value of the 8-bit Service Sub-type, copied from the spacecraft source packet's secondary data field header (see [R-6])

The LSB of the RouteID field is used by SCOS-2000 to identify the Data Unit Type of Telemetry packet and its qualifier (see chapter 5.3.4.1 of [R-9]). The Most significant nibble of the LSB is used to identify the type, and the least significant nibble to identify a qualifier.

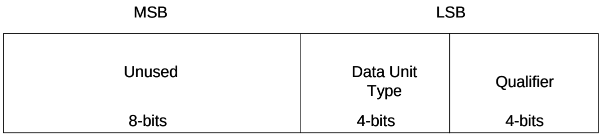

Figure 2: RouteID field structure

The following values are used for Solar Orbiter:

Data Unit Type	Qualifier
TM Transfer Frame (value = 0)	Good (value = 0)
TM Transfer Frame (value = 0)	Bad (value = 1)
TM Transfer Frame (value = 0)	Idle (value = 2)
TM Source Packet (value = 1)	Good (value = 0)
TM Source Packet (value = 1)	Bad (value = 1)
TM Source Packet (value = 1)	Idle (value = 2)
TM Source Packet (value = 1)	Unknown (value = 3)
TM Source Packet (value = 1)	A service 9 time packet (value = 4)
Internal MCS TM Packet (value =2)	CLCW (value = 0)
Internal MCS TM Packet (value =2)	user defined constants (value = 1)
Internal MCS TM Packet (value =2)	status consistency check (value =2)
Internal MCS TM Packet (value =2)	dynamic MISC (value =3)
Internal MCS TM Packet (value =2)	on-line MIB changes (value = 4)
Internal MCS TM Packet (value =2)	SPPG (value = 5)
Internal MCS TM Packet (value =2)	SPID Validity (value = 6)
Internal MCS TM Packet (value =2)	TPKT Configuration (value = 7)
Internal MCS TM Packet (value =2)	External Source (value = 8)

SCOS-2000 TC Packet Header Structure

The following details are taken from chapter 5.3.6 of [R-9], but the descriptions simplified for the Solar Orbiter context.

Field	Type	Length (Bits)	Description
UT secs	Unsigned Int	32	Uplink Time – initially set to the release time (FT), but updated with the timestamp copied from the radiation ground station response message providing the actual time of uplink from the antenna
UT usecs	Unsigned Int	32	Sub-seconds (in microseconds) of the Uplink Time
ET secs	Unsigned Int	32	Execution Time – For time-tagged commands this is the UTC user provided execution time. For immediate commands this is the release time + propagation delay and later updated to be Uplink time + propagation delay
ET usecs	Unsigned Int	32	Sub-seconds (in microseconds) of the Execution Time
LUT secs	Unsigned Int	32	Last Update Time, this is the MCS system time at which the TC packet was last updated with a change of status
LUT usecs	Unsigned Int	32	Sub-seconds (in microseconds) of the Last Update Time
Request ID	Unsigned Int	32	Internal MCS identifier
Request element index	Unsigned Int	16	Internal MCS sub-identifier
Variable Address Size	Unsigned Int	16	Combined size of the fixed and variable parts of the command details data (see chapter 7.1.4)
PUS APID	Unsigned Int	16	The value of the 11-bit Application Process ID, copied from the spacecraft source packet's primary header (see [R-6])
PUS SSC	Unsigned Int	16	The value of the 14-bit Sequence Count, copied from the spacecraft source packet's primary header (see [R-6])
PUS Service Type	Unsigned Int	8	The value of the 8-bit Service Type, copied from the spacecraft source packet's secondary data field header (see [R-6])
PUS Service Sub-Type	Unsigned Int	8	The value of the 8-bit Service Sub-type, copied from the spacecraft source packet's secondary data field header (see [R-6])
PUS Ack Flags	Unsigned Int	8	The value of the 4-bit Ack, copied from the spacecraft source packet's secondary data field header (see [R-6])
Uplink Flag	Boolean	8	`1` if the command is to be released (passed PTV) `0` if the command is not to be released (failed PTV)
Source Host	Unsigned Int	8	Numerical ID of the MCS machine running the command source
Source Type	Unsigned Int	8	Enumeration providing the type of the command source
Request Details Fixed Size	Unsigned Int	16	Size of the fixed part of the command details data (see 7.1.4)

7.1.4 Command Details Data

Unlike for the telemetry, in which the packet data portion of the SCOS-2000 is dedicated to the source packet (or telemetry frame), in the case of telecommands, this portion is used for storing other command related data. This is because all static information which is defined in the TM/TC MIB, such as the command mnemonic, parameter names and descriptions, verification information etc. must be stored in this packet. This is in order for it to be accessible from the TC History application, because it is not possible to ensure that the correct version of the runtime MIB database installed at any given time on the MCS will be the same as the one when the command was generated. If it is not and the command structure has changed, this will lead to problems in interpreting the command, therefore this data is stored in the SCOS-2000 TC packet. Along with dynamic information which is useful to the user, including release conditions, status of verification checks etc.

The full definition of the packet data portion of the SCOS-2000 TC packet is defined in [**A-7**]. Unlike for telemetry, this content is subject to change during the development of the MCS Telecommanding system and is driven by its design. Experience shows that this only stabilises once the flight control team have had a chance to see the TC History in use during operations validation. Therefore, the definition is not duplicated here in order to avoid potential conflicts in the documentation.
It is important to note that the command data details portion of the SCOS-2000 TC packet (described in [**A-7**]) is generated in the MCS by overlaying the member data of a C++ class over an area of memory in a 32-bit platform. This means that the fields inside of this portion of the data are 32-bit aligned. This has the effect that there will be some padding and consecutive fields are not necessarily next to each other in terms of memory addressing. For example, if a 8-bit field (which starts on a 32-bit memory address boundary, e.g. 100) is followed by a 32-bit field, there will actually be 3 bytes of padding in between as the 32-bit field is necessarily aligned to a 4 byte address. Therefore, in the above example, the relative address of the second field would be 104 and not 101.

7.2 SCOS-2000 Event Header Packet Structure

There is currently no event specific header, i.e. the MCS internally generated event information is added directly into the packet data portion of the SCOS-2000 packet.

7.3 Missing Frames and Bad Packets

When the MCS is building science packets out of the multiple telemetry frames, the entire 64KB packet will be embedded into the packet data area of a SCOS-2000 packet. However, it may be that after the MCS has started building a packet, it finds that some frames are missing (checked by the contiguity check on the virtual channel frame count field). In this case, the MCS is required to file frames, in order to be able to re-construct the inner data as much as possible, if required.

In this case the source packet, containing the data built from all the frames up to the first missing frame will be generated and marked as 'bad'. This means that the Qualifier (see RouteID in 7.1.2) field of the SCOS-2000 containing the partial packet, will be set to **`1`** (and the Data Unit Type will be **`1`**). Subsequent frames belonging to that packet if any are found (identified by searching for the next packet header, indicating the start of the next science packet) will be filed as good frames in SCOS-2000 packets, i.e. one SCOS-2000 packet will contain one frame, which will have a Data Unit Type of **`0`** and a Qualifier of **`0`**).

NOTE: If there should also be missing frames at the start of the next science packet, while the MCS is filing frames, then this cannot be detected by the MCS, because it will not be able to realise that the frames it is currently receiving actually belong to a new packet and not the original one where it initially detected gaps. This means that all subsequent frames will still be filed, until the next packet header is detected, which will look like they all belong to the same packet. External users can use the timestamp of the SCOS-2000 packets to determine which frames belong to which packet.

8 DATA PRODUCTS

The following section details the data provided by the Solar Orbiter DDS.

8.1 Packetised Data

Depending on the type of request, the DDS may return:
- either the full raw SCOS-2000 packet, containing both the SCOS-2000 header information and the spacecraft source data as a series of bytes (e.g. 'Packet Raw PARC data' EDDS request),
- or the request can be 'decoded', in which case only the part containing the spacecraft source packet shall be returned as a raw series of bytes, and the SCOS-2000 header will be interpreted and returned in a human readable format.

The format and header details of the various possible packet retrieval EDDS requests are described in chapter 7.

Time packets sent by the SolarOrbiter S/C (see 10.1) have their APID set to 0 (zero).

In addition to TM source packets from the SolarOrbiter S/C, a number of MCS internally generated packets will be contained in the 'telemetry' archive and used for the MCS' own purposes. These packets should be ignored by external users, either by filtering out APID **1966** or by filtering out their individual SPID numbers.
The following table lists, MCS internally generated packets:

SPID	APID	Description
66810	1966	File Transfer Model meta-data
66981	1966	Bad mini transfer frame
66980	1966	Good mini transfer frame
66982	1966	Idle mini transfer frame
66983	1966	Bad Telemetry Packets
66985	1966	Missing Packets Detection
66999	1966	TM packet with no assigned SPID
66830	1966	Dynamic UDCs
66831	1966	TC Spacon Configuration packet
66832	1966	MISC variables Configuration packet
66840	1966	Time Correlation Report packet
66841	1966	Time Couple packet
66842	1966	Time Correlation Coefficient packet
66850	1966	Releaser TM packet incl. G/S Response

An exception are time correlation related packets, which are further described here-after and will be available from the DDS.

8.1.1 Time Correlation Coefficient Packet

Time correlation coefficient packets contain the information necessary to enable the UTC time of a source packet to be obtained from its spacecraft time. See section 0 for a description of the end-to-end processing involved in generating the time correlation packets.

Here are the details of the SCOS-2000 Time Correlation Coefficient TM packet which can be requested from the DDS:

SPID	**66842**
PUS APID	**1966**
PUS Service Type	**164**
PUS Service Sub-Type	**3**

The content of the packet (i.e. the packet data portion of the SCOS-2000 packet) is defined in chapter 3.6.2 of [A-3].

8.1.2 Time Couple Packet

Time couple packets contain the OBT from the S/C time packet and the associated ground times. A set of these packets is then used by the time correlation software to derive a coefficient packet.

Here are the details of the SCOS-2000 Time Couple TM packet which can be requested from the DDS:

SPID	**66841**
PUS APID	**1966**
PUS Service Type	**164**
PUS Service Sub-Type	**2**

The content of the packet (i.e. the packet data portion of the SCOS-2000 packet) is defined in chapter 3.6.2 of [A-3].

8.1.3 Time Correlation Report Packet

Time correlation report packets contain information related to time conversion, as described in chapter 6.2.2.5.9.6 of [A-6].

Here are the details of the Time Correlation Report TM packet which can be requested from the DDS:

SPID	**66840**
PUS APID	**1966**
PUS Service Type	**164**
PUS Service Sub-Type	**1**

The content of the packet (i.e. the packet data portion of the SCOS-2000 packet) is defined in chapter 3.6.2 of [A-3].

8.2 Parameters

The parameter data type is described in chapter 5.2 of [A-1].

8.3 Reports

8.3.1 Command History Report

The EDDS supports the retrieval of Packet TC Reports. A packet TC report request returns the command packets with their data and parameters.
For use with other applications, the report can be returned in binary or XML format. Alternatively, an ASCII format, which matches the information shown in the TC History application of the MCS, can be retrieved. The ASCII version of the command history report is described in chapter 6.6 of [A-1]. The binary format of the report is described in chapter 6.7.3.3.6 of [A-1]. The XML schema description of the packet TC report is available in the Annex of [A-1].

Some additional Science Kernel specific fields are contained within the binary and the XML responses. To decode these fields within the binary format, the GPB PacketTCReport message definition can be used (chapter 6.7.3.3.6 of [A-1]). Within the XML format, the Science Kernel specific fields are embedded within CustomField elements and characterised by a name. type and value. These are embedded in the 3 child elements FieldName, FieldType and Value.Amongst the CustomFields, the following Science Kernel specific fields are covered:

isForcedSubScheduleId	specifies whether the Sub-Schedule ID was forced for that command
CevCheckStatePB	CEV check state for play-back data-stream (either CHECK_DISABLED or CHECK_ENABLED)
uniqueID	Umique ID command attribute, originally set for commands loaded from a Command Request File. Enables traceability of a command to its originating source. see [A-4].
s13ObFileId	Flags whether the command was sent via service 13 by specifying the File ID the encapsulated command belonged to.
s13ObFileAttribute	Service 13 File Attribute. The possible values are: PLAIN, OBCP, TC_DELAYED, TC_IMMEDIATE or PAYLOAD_CONTEXT.

8.3.2 Packet TM Gap Report

The packet TM gap report is described in chapter 6.7.3.3.8 of [A-1].

8.3.3 Other EDDS Reports

Other available EDDS Reports that can be retrieved are:

Report type	Described in
Request Summary Report	6.2.3 of [A-1]
EDDS Usage Report	6.2.3 of [A-1]
Packet TM, TC & EV Reports	5.2.3 of [A-2]
Out of Limit Report	6.7.3.3.9 of [A-1] and [A-2]

8.4 Archived Files

8.4.1 Auxiliary File Types

The following auxiliary file types are configured in the mission file archive and are available from the DDS, to be retrieved on demand or subscribed to (as explained already in chapter 2).
The archive file type to be used in the EDDS request filters is specified in the last column.

Type	Description	Described in ICD	Archive File Type Name
MIB	Mission Database file	[R-10]	MIB
Attitude	Attitude file	[R-7]	ATT
Attitude	Pointing timeline events	[R-7]	AttitudePTEL
Events	Long term events	[R-7]	EVT
EventsLT	Standard events file based on orbit determination and prediction and on nominal attitude prediction	[R-7]	EVT
OrbitST	S/C orbit for the until the end of the nominal mission – STP Orbit File	[R-7]	Orbit
OrbitLT	S/C orbit for the until the end of the nominal mission – LTP Orbit file	[R-7]	Orbit
OWLT	One way light time	[R-7]	OWLT
VISTA	World map with stations Predictionand and Linkbudget file	[R-7]	VISTA
HGA	HGA Blockage file	[R-7]	HGAB
SolarAspectST	Short-term solar aspect angle prediction (SolarAspectST)	[R-7]	SolarAspectST
CRFG	Zip file containing CRF files	[R-7]	CRFG
UEVT	User Events File	[R-8]	UEVT
EVFM	ESA Ground station scheduling events	[R-8]	EVFM

8.4.2 TM Files

TM files are downlinked via the SolarOrbiter specific service 13 protocol, and stored in the mission file system, available to the EDDS users via the EDDS MMI. These files contain raw spacecraft packets.

Though it is technically possible to extract the telemetry packets from the file for storage in the packet archive as SCOS 2000 TM Packets, this is not planned to be done as part of the baseline.

8.4.2.1 TM Files format

The inner structure of the TM files is a simple concatenation of TM packets. TM Packet formats are described in [R-4], [R-5] and [R-6].

8.4.2.2 Associated Meta Data Event packets

The following meta data associated to these TM files is contained in event (EV) packets with SPID **4000**. These packets are packetised data (see chapter 8.1) and allow to maintain the transaction history log on the mission control system.
As described in chapter 6.2.2.6.3 of [A-6], these EV packets contain the following information:

Field Name	Field Size
Transaction ID	4 Bytes
Packet Store ID	4 Bytes
Status (*)	4 Bytes
Number of FDUs received	4 Bytes
Number of FDUs expected	4 Bytes
Number of Retransmissions	4 Bytes
Start Time	4 Bytes
Last Packet Time	4 Bytes
End Time	4 Bytes
Size of Filename field	4 Bytes
Filename	*previous field's value (in Bytes)*
Injection kind (**)	4 Bytes
Injection time	4 Bytes

(*) The 'Status' field values correspond to the following:

Status	Decimal value
STATUS_UNKNOWN	0
ACTIVE	1
COMPLETED	2
INCOMPLETE	3
INJECTED	4
PENDING_ABORT	5
CANCELLED	6
ABORTED	7
PENDING_INJECTION	8
INJECTION_ERROR	9
CLEANED	10

(**) The 'Injection kind' field values correspond to the following:

Injection kind	Decimal value
INJECTION_AUTOMATIC	0

Injection kind	Decimal value
INJECTION_MANUAL	1

8.4.2.3 TM Files retrieval via DDS

The TM files are available from the DDS, to be retrieved on demand or subscribed to. The process for archive file retrieval is described in chapter 6.1.11 of [A-2]. It is highly recommended to use *zip* or *tar.gz* data compression as part of the delivery options, when performing TM file retrieval.

Complete TM Files are stored in the mission's file archive (FARC) under the '**data**' directory and as '**DLPP Files**' file type. These shall be used as filters when performing the EDDS requests for TM files.

Type	Description	Archive File Type
TM Files	Downlink file transfer files reconstructed on ground	DLPP Files

8.4.2.4 TM Files Naming Convention

The file naming convention for re-constructed TM files is as follows:

<PS_ID>_<file_reconstruction_timestamp>

Where:

- <PS_ID> is the packet store identification,
- <file_reconstruction_timestamp> is the time of file transfer completion.

For example: 13_2014.336.13.54.07.057

8.5 Statistics and Data Volume

The EDDS supports Packet TM and TC Statistics, described in [A-2].
These requests allow to retrieve simple statistical information between two times. The response will return the dates of the first and last packets corresponding to the filtered, or unfiltered, data that was requested. Additionally, it will provide the number of packets that match to the request. In case the requester needs to know the data volume that the matching data corresponds to, then this needs to be computed based on the average packet size and the number of packets reported in the statistics.

9 DDS USER INFORMATION

User accounts will be configured by the SolarOrbiter DDS administrator at ESOC for the external users. Each of these user accounts will have privilege to create sub-users under their responsibility and configure their access and quota, in terms of specific data and optionally an allowed relative time range.

The DDS supports user data access restriction in terms of the type of data that is to be retrieved. Filtering will be defined for:

- APIDs,
- SPIDs,
- parameter names,
- file names
- and/or file types.

For more details see [A-2].

Note: Underlying details such as the contact details for each user, and the IP address of the machine that they will use to access the DDS are not listed in this document but kept separate in [A-5], to avoid re-issuing whenever there is a small change in these details, which have no operational impact.

Further Note: The specific user access details will be sent individually to each user and not published on the present document for security reason, but will be added to [A-5].

10 RAPID FILE EXCHANGE

The nominal distribution method, between the OGS and the SGS, will be using the GFTS infrastructure software to forward RAPID binary files, that contain all packet data which is received processed and archived by the MCS. This is a one-way exchange configured at the OGS. It can ensure that all required data is transferred to the SGS**,** as soon as it is ready at the OGS, without the need for explicit requests.

10.1 Packet Data forwarded from OGS to SGS

The MCS is not only configured to store SCOS-2000 packets in its packet archive, but also to produce special binary files called RAPID files, see []. These files consist of SCOS-2000 packets stored within the MCS archive system. The MCS is configured to ensure that all the SCOS-2000 packets it handles, are inserted into RAPID files. After every configurable number of seconds, the current RAPID file is closed and a new one is opened in order to avoid the files from being too large to handle.

The data source for RAPID files will be the MCS LTA server. Via a file transfer mechanism software called GFTS, the RAPID files will be transferred across the operational LAN (OPSLAN) firewall to the DDS server on the Relay-LAN. From there a second GFTS instance will transfer all RAPID files to the SGS across the ESOC corporate firewall and the SGS firewall. This guarantees the completeness of the file transfer between the MCS and the network on which the RAPID file ingestion software resides, at the SGS.

Telemetry packets are placed in RAPID files in the order they are received and processed by the MCS. This means that any data which is re-dumped from the spacecraft or re-called from a ground station, will be likewise transferred once it arrives at the MCS, and therefore out of generation order. This must be taken into account by the SGS ingestion software, if used to fill in gaps.

In addition it is important to note that telecommand packets can be 'updated'. This means that when the status of a command is changed by the MCS, e.g. when its verification status changes, the MCS updates the same SCOS-2000 packet and effectively overwrites the one stored in the archive. Consequently, the updated packet (which contains the complete contents and not just what has been updated) will be stored inside the next RAPID file. Therefore the packet extraction software reading packets out of the RAPID files should take into account this fact and deal with the updates accordingly, ensuring that it is known that the update is referring to a packet which was previously transferred and not a new one. The identification of the packet can be made via the command's release time which is a unique key.

Note: Event and telemetry packets cannot be updated, i.e. each TM and event packet found in a RAPID file is a unique instance. However it is possible that the same packet may be stored more than once by the MCS (e.g. as a result of overlaps when performing a re-call to fill gaps). In this case, the contents of the packet will be identical, with the exception of certain fields in the SCOS-2000 packet header, e.g. it will have a new reception (storage) time.

The forwarding of telemetry packets via RAPID files will not be in real-time. Once a RAPID file is opened, it remains open for the writing of newly received packets until it is closed after the configurable length of time. Only then is it transferred to the SGS via GFTS. The time taken will depend upon the size of the file, and the network connection.

10.2 RAPID Size and Frequency of Transfers.

The RAPID file generation on the MCS LTA shall be configured to close after 50 seconds, which at maximum reception of telemetry will result in files of no larger than approximately TBC MB in size.
The configuration (frequency and maximum size) will be defined by OGS and fixed for all the mission, because the RAPID mechanism is also used for PRIME / BACKUP Mission Control System synchronisation.

The GFTS will attempt to transfer as quickly as possible maximizing the bandwidth. This means that in the nominal case RAPID files will be transferred one at a time, as it is not expected to take longer to transfer a file than the length of time a typical file is open. However, in case of connection issues, a backlog may accumulate on the OGS side. In order to remove the backlog, the GFTS shall be configured to manage up to 5 connections in parallel, such that it can transfer the accumulated files as quickly as possible. This implies that some file transfers will start before a previous one has completed, and will result in the possibility that files may arrive on the SGS machine out of generation order.

10.3 RAPID Interface Utility Software

A library of C++ functions providing access to RAPID files can be delivered to the SGS. It can be used (either directly integrated or re-written in Java) for opening and reading the files and for extracting individual packets and meta-data.

10.4 RAPID File Naming Convention

The RAPID file follows the following naming convention:

PARC_<Type>_<FirstPacketTime>__<LastPacketTime>

Where:

- <Type> is either be TM, TC or EV, depending on the type of SCOS-2000 packets that the file contains, e,g, telemetry, (which includes time packet, coefficients packets etc.) telecommands or MCS internal event packets.
- <FirstPacketTime> is the UTC time of the first packet inserted in the RAPID file. In the case of telemetry this is the reception time (also known as Storage Time and Creation Time) – not the on-board generation time. For Telecommand packets it is the Last Update Time. For MCS internal event packets it is the system time on the MCS when the packet was created.
- <LastPacketTime> is the UTC time of the last packet inserted in the RAPID file, with the same times dependent on type as described for <FirstPacketTime>.

The times are in the format `YYYY-MM-DDTHH:MM:SS.mmmmmm`

The following is an example of a RAPID file name:
`PARC_TM_2019-04-04T12:30:00.000000__2019-04-04T12:32:50.000000`

10.5 RAPID File Format Definition

The packet archive process of the MCS is responsible for creating and opening a new RAPID file, and closing it after the configurable length of time. Every SCOS-2000 packet that the process receives for archiving is appended to the end of the currently open RAPID file, depending on the packet's type. I.e. there exist separate RAPID files for TM, TC and Event packets, which means that there are three open RAPID files being written to at any one time in the MCS.

Because the RAPID file is a mechanism developed for mirroring MCS packet archives deployed across a firewall, the structure of the RAPID file allows for the encoding of actions. This means that in between each packet stored in the file, there is meta-data which describes what action the target packet archive must take with the packet.

In most cases this will be an insert action, signifying that the packet is a new packet which must be archived. In the case of TC packets, which can change their contents (due to changes in its verification status), second and subsequent instances of a given TC packet will have an update action associated with them. This tells the target packet archive that the packet is an update of a previously received packet and not a new one, implying that it most probably should overwrite the previous once. Finally, an action could be a delete action, which could be used by the OGS software support team to request to delete packets from the packet archive. However, for the purposes of this data exchange mechanism, the generation of RAPID files is being triggered solely by the reception of packets by the MCS packet archive, and so *the RAPID files forwarded to the SGS shall NOT include delete actions.* If they do, due to RAPID files being inter-mixed with files used for mirroring internally in the MCS, those records should be ignored by the SGS.

The structure of a RAPID files consists of a contiguous set of RAPID records. Each record is preceded by a 32-bit integer field which contains the length of the following RAPID record.

Rec-1 Length	Rec-1	Rec-2 Length	Rec-2	Rec-3 Length	Rec-3

Figure 1: RAPID file structure, this example containing 3 RAPID records

Each RAPID record consists of two parts. The first is a fixed length part containing various meta data fields. The second part is of variable length containing the SCOS-2000 packet data itself (plus a filter field)

FIXED									VARIABLE	
ID	ST	ET	AT	DS	TYPE	SM	NS	FS	FIL	PACKET

Figure 2: RAPID record structure

Field	Type	Length (Bytes)	Description
ID	Unsigned Int	8	Version ID of the record structure – this will have the value: **1**
ST	Time	8	Time of the first packet to be deleted – NOT TO BE USED but field exists. 64 bit integer containing micro-seconds since 1970.001.00.00.00.
ET	Time	8	Time of the last packet to be deleted – NOT TO BE USED but field exists. 64 bit integer containing micro-seconds since 1970.001.00.00.00.
AT	Time	8	Action Time, i.e. the MCS system time at which the packet was inserted or updated into the original packet archive. 64 bit integer containing micro-seconds since 1970.001.00.00.00.
DS	Unsigned Int	8	Data size of the packet, i.e. how many bytes long the PACKET field will be
TYPE	Unsigned Int	4	The type of action. This will be : • **0** for Insert (i.e. a new packet - applicable for all packet types) • **1** for Update (i.e. a previous Telecommand packet has been updated and can be overwritten) • **2** for Delete (however, this action should not occur and should records with this action should be ignored)
SM	Fixed Value	4	A sync marker whose value in hexadecimal is: **1ACFFC1D**
NS	N/A	4	Not used
FS	Unsigned Int	4	The size of the filter information used for deleting, how many bytes long the FIL field will be. This should always be set to zero for this Mission.
FIL	N/A	0	Filter information – NOT USED, i.e. this field is zero length and effectively does not exist, assuming FS above is zero.
PACKET	Binary	variable	The SCOS-2000 packet (see chapter 7.1)