E u r o p e a n S p a c e A g e n c y

Research and Science Support Department

Planetary Missions Division

ROSETTA

Rosetta Science Operations Centre to Experimenter
Interface Control Document
(RSOC to Experimenter ICD)

RO-EST-IF-5010

Draft 15

05 Nov 2001

Prepared by: D. Koschny

this page intentionally left blank

APPROVED BY

ALICE Principal Investigator		A. Stern
CONSERT Principal Investigator		W. Kofman
COSIMA Principal Investigator		J. Kissel
GIADA Principal Investigator		L. Colangeli
MIDAS Principal Investigator		W. Riedler
MIRO Principal Investigator		S. Gulkis
OSIRIS Principal Investigator		H.U. Keller
ROSINA Principal Investigator		H. Balsiger
RPC Spokesperson		
RSI Principal Investigator		M. Paetzold
VIRTIS Principal Investigator		A. Coradini
LANDER Project Manager		H. Scheuerle
ROSETTA Project Scientist		G. Schwehm
ROSETTA Project Manager		J. Ellwood

CHANGE RECORD SHEET

Date	Iss.	Rev.	P.	Description/Authority	CR No.
22 Aug - 22 Nov 2000	D	0 - 6		First draft	
15 Jan/22 Jan 2001	D	7/8		incorporate updates from GSDR	
29 Jan 2001	D	9	10 12	pointing request file has two time fields (as the POR/ITL format) format for "include_file" keyword changed	
06 Feb 2001	D	10	all	"last comments" by ESOC (MS and EM) and GS	
10 Sep 2001	D	11	all	update editoral comments by RS	
			2	Added picture "physical interfaces" and corresponding text. Add " (*e.g.* personal communication)" to password updating	
			3	delete "medium- and" in first sentence of 2.2.1.2 Update the way to put files on the DDS	
			4	delete "of the same type" Modify paragraphs on response time.	
			11	delete pointing mode table, refer to PTR document instead	
				In all ITL/EDF formatted files, exchange the comment character from "C" to "#"	
			25	Add details on Telecommand Sequence update requests	
			33	Add reference to Ground Segment System Test Plan	
			all	updated the time formats to YY-DDD...	
22 Sep 2001	D	12	ii	Delete B. Feuerbacher as Lander PM; p. 1 editorials	
				Updated wrt comments from HUK	
			4	add " The confirmation will be done either via an email with attachment or by a ftp put to the computer of the experimenter, depending on the capabilities of the DDS (*tbd*)." to section 1.1.2.2.	
			19	Update Lander-specific interface according RO-ESC-MN-0531.	
24 Sep		13		Editorials: Updated name of Lander PM; added reference to OSIRIS UDP document	
11/12 Oct		14		Updated example of Command Sequence request, add naming conventions for sequence names and sequence parameter names	
			19	Insert Section 3.2.1.9, RSI-specific interface to the RSOC (still empty)	
			21	Insert Section 3.2.3.1, Acknowledge file (ACKN)	
			78	add Appendix D, Format definitions of the keywords and their contents for the acknowledge files	
				Added explanation of "start time" and "end time" in all relevant tables	
			4	add requirement to wait for receipt acknowledge before sending new file	
			32	Add naming conventions for sequences and their parameters	
05 Nov 2001			4	Add " In case three erroneous accesses occur, the Spacecraft Operations Manger (SOM) at ESOC shall be contacted by telephone to reset the account"	

Revisions are indicated by a vertical bar at the outside border.

DISTRIBUTION LIST

Recipient	**Organisation**	**Recipient**	**Organisation**
M. Warhaut	TOS-OGR	G. Schwehm	SCI-SO
P. Ferri	TOS-OGR	R. Schulz	SCI-SO
M. Sweeney	TOS-OGR	D. Koschny	SCI-SO
E. Montagnon	TOS-OGR	R. Hoofs	SCI-SO
E. Sørensen	TOS-OGR	N. Biver	SCI-SO
J. Fertig	TOS-GFI	J. Zender	SCI-SO
M. Lauer	TOS-GFI		
V. Companys	TOS-GFI	J. Ellwood	SCI-PR
J. Rodriguez-Canabal	TOS-OFA	C. Berner	SCI-PRP
		G. Jung	SCI-PRP
all Rosetta PIs and TMs		B. Gramkow	SCI-PRP
		M. Schwetterle	SCI-PRP
		P. Kletzkine	SCI-P
		M. Nesbit	SCI-PRP
		J. van Casteren	SCI-PRS
		A. Marini	TOS-MMO
		C. Alexander	JPL
		J. P. Biebring	
		H. Rosenbauer	MPAE
		Lander PM	DLR
		P. Weiss	DLR
		M. Jeannot	CNES
		P. Gaudon	CNES

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Rosetta is the third cornerstone mission of the ESA long term scientific program Horizon 2000 and will be launched in January 2003 to comet 46P/Wirtanen. A Rosetta Science Operations Team (SOT) has been established to perform the scientific operational planning for this mission. The SOT together with their infrastructure forms the Rosetta Science Operations Centre (RSOC). The Rosetta Project Scientist is part of the SOT and has the overall responsibility of the RSOC. The general concepts of how the planning will be done are described in AD8, the top level document "RSOC Design Specification" (note that this document was formerly called "RSOC Design Concept"). This document will concentrate only on the formal definition of the interfaces between the Experimenter teams and the RSOC with all their technical formats.

1.1 Purpose and Scope

The purpose of this document is to specify the operational interfaces and procedures between RSOC and the ROSETTA orbiter PI and Lander teams (henceforth called "Experimenter teams") that are required to support ROSETTA science operations.

The areas addressed in this document include:

- the generic interfaces required to support the above functionality;
- any PI-specific interfaces required to support the above functionality;
- standards for exchange of software and data;
- testing of the interfaces;
- training in the use of the interfaces.

1.2 Applicable Documents

AD1: Experiment Interface Document Part A (EID-A), RO-EST-RS-3001, Iss. 2, Rev. 1, 01 Oct 2000
AD2: Science Operations Implementation Agreement (SOIA), RO-ESC-IF-5005, Iss. 1, 25 May 2000
AD3: Command Request Interface Document (CRID), RO-ESC-IF-5004, Draft B1, 20 Jul 2001
AD4: Data Delivery Interface Document (DDID), RO-ESC-IF-5003, Draft B1, 20 Jul 2001
AD5: Experiment Interface Document Part C (EID-C), RO-EST-RS-3001, Iss. 1, Rev. 4, 15 Dec 2000
AD6: ESA Software Engineering Standards, PSS-05-0, Issue 2, Feb 1991, ESA Board of Software Standardization and Control (BSSC).
AD7: RSOC Implementation Plan (RSOC IP), RO-EST-PL-3032, Draft h, 11 Oct 2001.
AD8: RSOC Design Specification, RO-EST-PL-2010, Draft 17, 23 Sep 2001.
AD9: PTR Software Specification Document, SOP-SSD-SP-002, Issue 1.1, 30 Aug 2001
AD10: Format definition of the Lander Flight Operations Plan, tbd
AD11: OSIRIS Command Language description, IDA-OCL-0001, Issue 0.7, Jul 2001

1.3 Reference Documents

RD2: Mission scenarios - Close encounter , RO-EST-TN-3027/D7, 21 Mar 2000
RD3: EPS Software User Manual, SOP-SSD-UM-001/1-, 15 Jan 2001
RD4: ECSS-40, Software Standard
RD5: deleted
RD6: RMOC Mission Planning Concept, RO-ESC-TN-5601/D1, March 1998
RD7: Rosetta System Database Naming Convention, RO-ALS-TN-4002/4-, May 2000
RD8: CCSDS Time Code Formats, CCSDS 301.0-B-2 Blue Book, Issue 2, Apr 1990
RD9: Ground Segment System Test Plan (GSSTP), RO-ESC-PL-5102, Issue 1, Nov 2000
RD10: Rosetta FOP Production Plan (FOPPP), RO-ESC-PL-5107, Issue 1, 31 Oct 2000
RD11: RLGS/RGS Interface Requirements Document, RLGS-EIF-RLGS/RGS-TECH-6-CNES, Issue 1.10, 25 Jul 2001

1.4 Naming conventions

The following naming conventions are used throughout this document::

- Field names are written in typeface Courier and in brackets, *e.g.* "`<start time>`".
- Keywords that must appear as they are written are given in typeface Courier, *e.g.* "`INERT`".
- A "`|`" is used to indicate an exclusive or, *i.e.* `<start time | event>` indicates that the field name is either "`start time`" or "`event`".
- In the file examples, the column numbers are given on top of the example. These are *not* part of the actual file.
- Whenever the term "Experimenter" is mentioned, the orbiter PI teams plus the Lander Lead Scientists are meant. Synonymously, the term "Experimenter" is used.

2. MANAGEMENT

2.1 Overall Management

The Rosetta Project Scientist is responsible for Rosetta Science Operations. He/she is the formal interface for all science-related matters, which includes pre/post-launch science planning and the provision of all science operations inputs. A team of people as described in RD1 aids him/her. This team forms the Science Operations Team (SOT), also called RSOC.

It is the responsibility of the Experimenter teams to provide the necessary information to allow the SOT to perform the science operations planning.

2.2 Communications and security

2.2.1 Network interface

As soon as the Data Distribution System (DDS) is operational, all file exchange between the RSOC and the Experimenter teams will be performed via it (expected spring 2002). The DDS is physically located at ESOC. The File Transfer System (FTS) of the DDS will redirect all files for the RSOC to a dedicated RSOC computer, called RSOC FTS. The Science Operations Team will have access via secure ftp and rlogin to this machine. Figure 1 shows the setup of the physical interfaces.

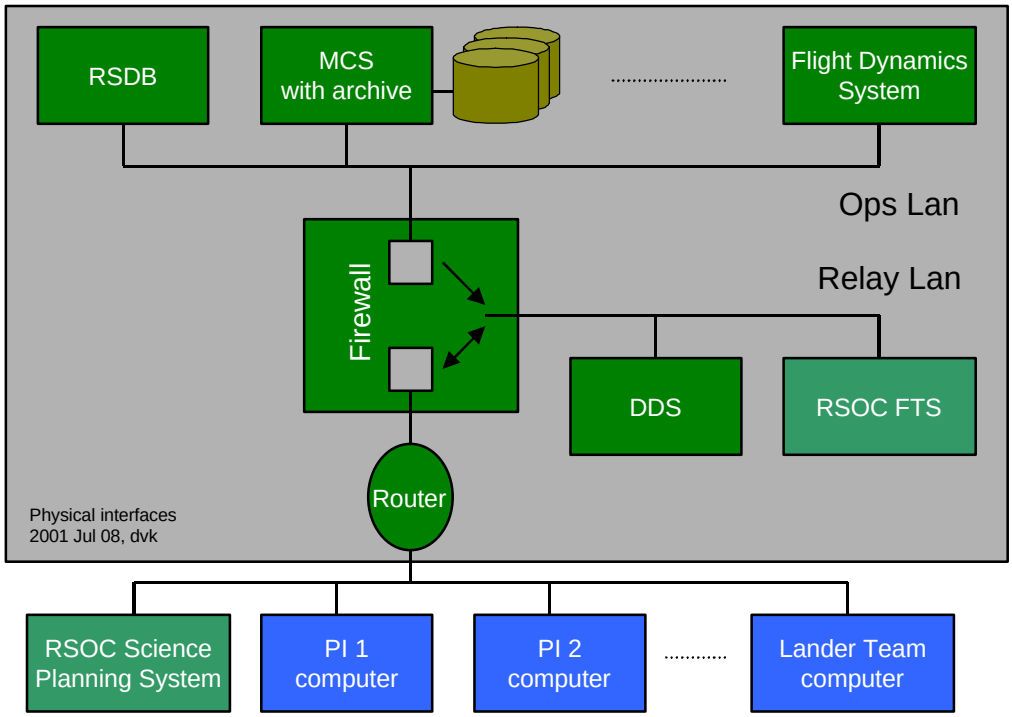

Figure 1: Physical interfaces between Experimenter team computers and the RSOC.

During the RSOC Development Phase and possibly during part of the RSOC Test Phase (*i.e.* before launch), a server located at the Space Science Department (SSD) of ESTEC will be used.

The network protocol supported by RSOC shall be TCP/IP. All network interactions between RSOC and Experimenter teams shall therefore use TCP/IP applications. For file transfer, FTP shall be used.

The Experimenter teams shall access the DDS or the RSOC computers at ESTEC via the public Internet. Thus secure applications shall be used wherever practicable.

To ensure the security of the RSOC computers, network access by Experimenter teams shall be permitted only from internet addresses notified to RSOC by the Experimenter teams. They shall be any addresses regularly used for access to RSOC, e.g. computers at the Experimenter's home institute, EGSEs at ESOC or computers at other institutes from where the Experimenters may wish to access RSOC. If an Experimenter wishes to add, change or delete items in the list of internet addresses allowed to access RSOC, he or she shall notify RSOC formally by Email or fax as shown in Section 2.2.1.5. RSOC shall verify the requested changes and then implement them as soon as possible.

During co-location at ESOC in Darmstadt, the PI/Lander team computers will be located at the Relay LAN.

2.2.1.1 Inputs from Experimenter teams

Electronic inputs from Experimenter teams, *e.g.* experiment software patches, shall be sent as computer files using the FTP application to the RSOC FTS computer.

Use the following procedure:

```
ftp  RDDS_ftp_address (TBD)
<enter  Name  + Password   (TBD)>
cd   particular_incoming_directory (TBD)
put  filename.ROS
<with filename following the FTS convention:
ffff_sordes_txxxxxxxxxxxxx_vvvvv.ROS>
```

The name of the file shall identify its contents; the naming scheme for each class of input shall be defined in the appropriate Section of this Interface Control Document (ICD).

All input files shall be ASCII or pdf files. Pdf inputs shall conform to the recommendations laid down within CSDS.

ASCII input files shall not contain tab characters. The implementation of tabs varies between different computers, so tabs cannot be used to simulate the fixed format. Files containing tabs shall be rejected as invalid.

The security of this input process will rely on the proper and careful use of the passwords which give write access to the ftp server. Thus RSOC shall enforce the following standards:

- password size
- password content
- regular updating
- account locking on error counts > 3

The Experimenter teams shall be in charge of updating their passwords. However, RSOC operations staff may update them, if requested by Experimenter teams. The updated password shall be sent to the team by secure means (*e.g.* personal communication).

In case three erroneous accesses occur, the Spacecraft Operations Manger (SOM) at ESOC shall be contacted by telephone to reset the account.

2.2.1.2 Checking, performance, response time

In the short-term planning cycle, there will be deadlines for the operational request files, *e.g.* every last Friday of the month at 12h00m UT (*tbc*). The operational request files are called Orbiter Instrument Operational Request (OIOR) and Lander Operational Request (LOR) file. The format of these files can be either the POR format (POR = Payload Operational Request), see Section 3.2.1.3.3, or the ITL format (Input Timeline), see Section 3.2.1.3.5.

Upon receipt of a file, the RSOC will perform a syntax check on the file. After having performed this check, a *receipt acknowledge* will be returned. If the syntax check was not successful, an error report will be provided. The syntax check and the receipt acknowledge will be performed by a software tool, the Operational Request File Acknowledger (ORF-A), which will run on the RSOC FTS computer.

Even though the RSOC can handle several arriving files simultaneously, it is required that the Experimenter teams only send another file after they have received a receipt acknowledge of the file that was sent previously. Otherwise, correct versioning cannot be ensured. Note that in the regular short-term planning process, it is expected that the Experimenter sends one and only one OIOR/LOR file or PTR file per planning cycle (covering typically one week). It has to send this file until a given deadline (see AD8, p. 7, for the deadlines). The latest available file of one type will be used for the planning.

The validity of the content of the file can in some cases only be confirmed after more processing has been done, *e.g.* accepting an OIOR file is only possible after it has been cross-checked that no conflicts with other OIORs occur. This, necessarily, needs more time, and will be done after the inputs from all Experimenter teams have been received, *i.e.* after the passing of the deadline for the respective planning cycle.

When it is done, a *file acceptance report* will be returned. Especially in the case that the file was not accepted, an error or conflict report will be provided (*e.g.* the CONF file for an OIOR or LOR).

The confirmation will be done by a ftp put to the computer of the experimenter. The server addresses need to be specified in this document, see Table 10.

The response times of the RSOC (only) for different mission phases is given in Table 1. "Emergency cases" are defined as cases where the s/c and/or payload safety might be endangered. In these cases, it is possible to circumvent the normal planning process to increase the response time.

Table 1: Maximum response times for file acknowledge and acceptance from RSOC (only).

	Commissioning	Cruise	Nominal science	Emergency cases
Receipt acknowledge	5 min	5 min	5 min	not applicable
File acceptance report	1 day after file receipt	1 week after file receipt	2 days after planning deadline	5 min after receipt

2.2.1.3 Outputs from RSOC

Currently, the main entry point for data distribution by the RSOC is the RSOC web site at

http://solarsystem.estec.esa.nl/RSOC/

Documents are available from the RSOC documentation server, at the web site

http://astro.estec.esa.nl/livelink/

This page is password protected. User names and passwords shall be distributed by secure means, *i.e.* personal communication. An automatic notification system for updates is available, see the online documentation.

After the DDS is fully operational (expected mid-2002), all outgoing information from the RMOC as described in AD4 (Data Delivery Interface Document, DDID) and from the RSOC as described in this document will be available via the DDS.

All output files shall be ASCII files with variable record length. For documents, the file format will be Microsoft Word files or (preferred) Portable Document Files.

2.2.1.4 Access to input and output directories

The access modes of the incoming/outgoing directories will be restricted by passwords, see previous Section.

2.2.1.5 Electronic mail

Any messages to RSOC which are not part of the nominal planning process shall be sent to:

rsoc@esa.int (*tbc*)

(Note: This email address is as of the time of writing of this version of the document not yet available. Use the Project Scientist's email instead, with copies to the SOT members:

gerhard.schwehm@esa.int, rita.schulz@esa.int, detlef.koschny@esa.int, nicolas.biver@esa.int, raymond.hoofs@esa.int, joe.zender@esa.int

2.2.2 Mailing address

All mail for the RSOC shall be sent to the Project Scientist with copy to his team. The current address is

Gerhard Schwehm
cc: Rita Schulz, Detlef Koschny, Nicolas Biver, Raymond Hoofs, Joe Zender
Keplerlaan 1
NL-2201 AZ Noordwijk ZH

2.2.3 Fax & telephone

Fax: +31-71-565-4692
Phone: +31-71-565-3539 (Gerhard Schwehm)

2.2.4 Handling of computer downtimes

All data exchange will be performed via the DDS, which features a cold redundant computer setup maintained by the RMOC at ESOC. The computer system will be monitored 24 hours a day, 7 days a week. If the RSOC FTS computer crashes, it will be rebooted. If a reboot is not possible, a backup machine will be installed.

It is expected that any unavoidable downtime of the DDS will be announced by ESOC.

RSOC shall report unscheduled downtime on the RSOC computers to the Experimenter teams if this is expected to exceed 4 hours on a working day. These reports will be sent by electronic mail using alternative electronic mail facilities at RSOC location. In the unlikely event that no alternative is available reports will be sent by fax.

2.3 Meetings

The following table gives an overview over the planned meetings concerning science operations. Note that to reduce travel costs, it is foreseen to use modern technology as video conferencing and the web as support media. In particular, for the monthly planning meetings the necessary information will be available on the web so that people can participate via telephone or video and have the documentation (*e.g.* the Science Activity Plan under discussion) available at their location. The short-term planning meeting will be done at a time where it allows the US Co-Is to participate remotely, *e.g.* at 16h ESOC time.

Table 2: Planned meetings with approximate dates.

Name	Purpose	Time
SWT	Normal SWT agenda plus mission scenario definition	Twice per year, during the complete mission
SWT	Final planning meeting for the flybys (Earth 1, Earth 2, Mars, Asteroid 1, Asteroid 2)	6 months prior to respective flyby
SWT	Final long-term planning meeting, decide final long-term plan	Jun 2011
STPM	Short-term planning meeting - agree to any updates	During science phases: weekly, possibly daily

2.4 Change Control

This document shall be maintained by RSOC under ESA configuration control. Change requests should be sent to RSOC. RSOC will assess the impact of the change together with the proposer. When this assessment is complete, RSOC will circulate a formal change request. If the request is agreed, RSOC will update this ICD and make the new version available on the RSOC documentation server.

3. INTERFACE DESCRIPTION

A general description of the planning process for Rosetta is given in AD8. Here, we summarize the interfaces of the RSOC, since a part of these are the topic of this document. The most complex scenario is during the comet observation phase, which is described here. During the cruise phases and asteroid flybys, pre-validated procedures will be used. Note that the technical definition to the interfaces is identical to those described here, even if the planning process itself is easier.

One of the tasks of the Rosetta Science Operations Centre (RSOC) is to consolidate the operational requests by the Experimenter teams. These will send their operational requests by electronic files called Orbiter Instrument Operational Requests (OIORs) and Lander Operations Requests (LORs) to the RSOC. After conflict resolution between the payload experiments (which might require iterations with the Experimenter teams and the RMOC), the RSOC will generate a Payload Operation Request (POR) file in the format described in AD3.

In addition, the Experimenter teams will request attitudes and trajectories, which again will be consolidated by the RSOC and only then be forwarded to the RMOC.

In return, the RSOC will provide the Conflict File (CONF) generated by the planning software used at the RSOC for information of the Experimenter teams. After the consolidation and resolution of all conflicts, RSOC will provide

(a) a Science Activity Plan (SAP) after each long-term planning process, which is a listing of mission scenarios;

(b) a Master Science Plan (MSP) for each mission scenario, which goes into more detail. *E.g.* it will list the experiment operations at least down to mode level and include critical mission events as well as the planned pointing;

(c) the final, consolidated POR file whenever it is forwarded to the RMOC.

To ease the planning process within the Experimenter teams especially in the early phases of the project, the RSOC will also provide so-called "reference" information:

(d) a reference comet, giving some design values for the comet parameters;

(e) a reference attitude file, typically one per mission scenario;

(f) a reference trajectory file, typically one per mission scenario.

Figure 2 shows the Experimenter teams, the RSOC, and the RMOC with their interfaces. The purpose of this document is to define, down to the detailed level of file formats, the interfaces between the RSOC and the Experimenter teams. The interface between RMOC and Experimenter teams is described in AD4 (DDID).

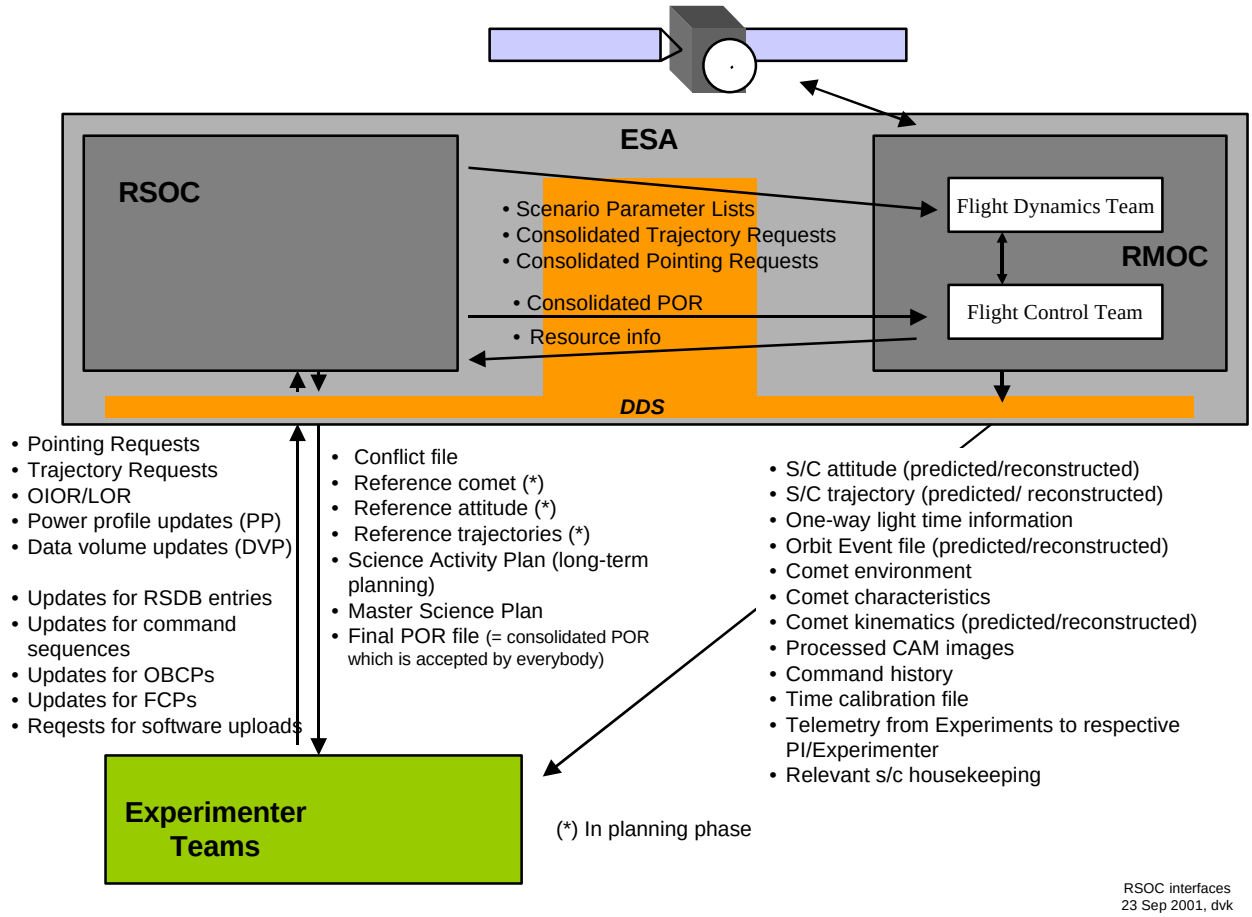

Figure 2: Interfaces between Experimenter teams, RSOC, and RMOC.

3.1 General file naming conventions

All filenames are based on a convention given in AD3 (CRID). The general format for a file name is

`ffff_sordes_txxxxxxxxxxxxxx_vvvvv.ROS`

where

`ffff` is the file type identifier, permitted values are:

ACKN	Acknowledge file
CONF	Conflict file
CS__	Command Sequence Update
DVP_	Data Volume Profile Update
ECF_	Expedite Command File
ITL_	Input Timeline
LFOP	Lander Flight Operations Plan
LOR_	Lander Operations Request
MCR_	Memory Checksum Request
MDR_	Memory Dump Request
MPR_	Memory Patch Request
MSP_	Master Science Plan
MCR_	Memory Checksum Request
OBCP	On-Board Control Procedure Update
OIOR	Orbiter Instrument Operations Request
POR_	Payload Operations Request
PP__	Power Profile Update
PTR_	Pointing Request
RPI_	Reference Pointing Information
RTI_	Reference Trajectory Information
RFDE	Reference Flight Dynamics Events
UDP_	Reference Flight Dynamics Events

`sor` is the source mnemonic, *i.e.* the mnemonic identifying the node from which the file originates. The permitted source mnemonics are:

PIn	Rosetta Principle Investigators (PIs) and Lander, with the value n assigned as in Table 3.
RSO	Rosetta Science Operations Centre (RSOC)
OTH	other source

Table 3: Experiments and code for file names.

Experiment	n	sor	Experiment	n	sor
ALICE	1	PI1	RPC_IES	B	PIB
CONSERT	3	PI3	RPC_MAG	C	PIC
COSIMA	4	PI4	RPC_ICA	D	PID
MIDAS	5	PI5	RPC_MIP	E	PIE
MIRO	6	PI6	RSI	F	PIF
OSIRIS	7	PI7	GIADA	H	PIH
ROSINA	8	PI8	VIRTIS	I	PII
RPC_PIU	9	PI9	LANDER	L	PIL
RPC_LAP	A	PIA	All	X	PIX
Note: PI2 was deleted.					

`des` is the destination mnemonic, *i.e.* the mnemonic identifying the node to which the file is being sent. In the context of this ICD, the following destinations are foreseen:

`PIn`	The computer of a specific Experimenter team, with n as defined in Table 3
`PIX`	The computers of all Experimenter teams
`RMA`	The RMOC primary computer
`RSO`	The RSOC primary computer
`ALL`	All of the above

`t` is the data type identifier. Currently, only one data type is foreseen, namely

`D` indicates that the file contains data.

`xxxxxxxxxxxxxx` is a file specific field. It allows to give a descriptive name to the file (must be 14 characters long). Several options are possible:

(a) If the file is an OIOR or LOR file and is intended to be included in another OIOR or LOR file as described in Section 3.2.1.3.3, keyword `Include_file`, this field shall be used to specify a descriptive name to the file, *e.g.* `_image_seq____`. Note that the name shall always start with an underscore character.

(b) If the field is not used to describe an include file, it shall contain the name of the experiment (*tbc*) as given in the first column of Table 3, preceded trailing underscore signs to yield a total number of 14 characters. If the source is the RMOC, this field will consist of 14 underscore characters or a text description of the file contents.

(c) If the file is an OIOR, LOR, or PTR file used in the nominal short-term planning process, it shall contain a "P" as the first character, followed by a three-digit number (preceding zeros of required) denoting the planning cycle for which the file is intended. It is the responsibility of the RSOC to precisely define the time range for each planning cycle.

`vvvvv` is the version number which starts at `00001` and increments for each `ffff_sordes` (with wraparound at `99999`).

As an example: `MCR__PI1RSO_D__________ALICE00010.ROS` is the 10th memory checksum request submitted by the ALICE team.

3.2 Interfaces concerning the timeline planning

3.2.1 Information from the Experimenter teams to RSOC

3.2.1.1 Trajectory requests (TRR)

The trajectory requests will be given to the RSOC in form of email communication to the Project Scientist or as minuted during operational planning meetings.

3.2.1.2 Pointing requests (PTR)

3.2.1.2.1 Description

This will be a request file giving time and pointing mode, *e.g.* NADIR means pointing to nadir. Some of the modes need parameters, these are currently being discussed with the RMOC and will be updated in a future update of this document.

3.2.1.2.2 File naming convention

The file naming convention shall follow the description given in Section 3.1, with the file type identifier `ffff` being "PTR_", the source `sor` is "PIn" as defined in Table 3, the destination `des` is "RS0".

3.2.1.2.3 File format

The file shall be an ASCII file. Each line shall have a maximum number of 128 characters, including a line feed (<LF>). Each file shall follow the definition given in AD10, Section 4. The date and time of creation and an author shall be given as a comment in the beginning of the file.

3.2.1.2.4 Example

```
0         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
# This is an example of a Pointing Request file
# Created 2001 Sep 10, 10:11:12 UTC, dvk

AT_PERICENTER    -00:01:30     OSIRIS     INERT_START (RA = 15.75 DEC = -37.60)
000_01:00:00                   OSIRIS     INERT (OBJECT = SUN)
000_01:01:00                   OSIRIS     INERT (OBJECT = SIRIUS, \
                                                 POINTING_AXIS = X, \
                                                 SLEW_POLICY = SMOOTH)
000_02:20:30                   OSIRIS     NADIR
000_03:21:10                   OSIRIS     NADIR (ROLL_MODE + ROLL_FIXED)
#-------------------------------------------------------------------------------
```

3.2.1.3 Orbiter Instrument Operational Requests (OIOR)

3.2.1.3.1 Description

The OIOR file is the main means of the orbiter PI teams to communicate their operational requests to the RSOC. Two formats will be supported:

(a) A format which contains the format of the POR file, but with (minor) additional capabilities to make the science planning process easier.

(b) A format called ITL (Input Timeline), which is a more readable version of the POR format in columnar structure.

A thoroughly tested and verified converting routine is available as part of the EPS software. It is available via the lifelink server as described in Section 2.2.1.3.

Note: Both the OIOR and the LOR file are called Operational Request Files (ORF).

3.2.1.3.2 File naming convention

The file naming convention shall follow the description given in Section 3.1, with the file type identifier `ffff` being "OIOR", the source `sor` is "PIn" as defined in Table 3, the destination `des` is "RSO".

If the file is for the regular short-term planning (once Rosetta will be at the comet), the first four characters of the 14 empty characters in the file name shall denote the planning cycle number for which the file is applicable. The first letter shall be a "P", followed by a three-digit number denoting the planning cycle (*e.g.* P013). It is the task of the RSOC to precisely define the time period relevant for each planning cycle.

3.2.1.3.3 File format for OIOR file similar to POR format

The file shall be an ASCII file. The format shall follow the description in AD3 (CRID), Section 7.1. In addition, the keyword `"Include"` as defined in Table 4 is allowed at any line.

The "delete" functionality of the POR format as defined in the CRID will *not* be supported in the nominal short-term planning process. This is to facilitate the correct versioning of the OIOR/LOR files.

Table 4: Including files in the OIOR file in addition to the definition in AD3.

Field name	Column	Format	Comments
<Start_time | Event_label>	1	20X	CCSDS format time (YY-DDDThh:mm:ssZ) or ITL format time (±DDD_hh:mm:ss) If both start and end time are given, the start time is the first allowed time for the execution of the command. Or event label as defined in AD4, annex H.
<End_time | Delta_time | Empty_label>	2	20X	End_time is the end time in CCSDS format time (YY-DDDThh:mm:ssZ) or in ITL time format (±DDD_hh:mm:ss) If both start and end time are given, the end time is the last allowed time for the execution of the command. Delta_time is in the format ±[DDD_]hh:mm:ss where: DDD is the number of days in the range 0 to 999; hh is the number of hours in the range 0 to +23; mm is the number of minutes in the range 00 to 59; ss is the number of seconds in the range 00 to 59; Leading zeros must always be inserted to ensure that the field is 4/2 digits long; or in ITL time format (±DDD_hh:mm:ss). Empty_label means that the field is empty, a dash ('-') has to be given.
<Exp_label> *optional field!*	3		Name of the experiment from column 1 of Table 4 (optional)
Include	4	7X	keyword to indicate an include file
<file_name>	5	37X	name of file to be included according to definition in Section 3.1.

3.2.1.3.4 Example

For an example of the regular POR format, see AD3 (CRID). An example for a line with the keyword Include_file is given here:

```
0        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
C--------------------------------------------------------------------------------
C This is an example of a line defining to include a file with
C the name tag "IMAGE_SEQ".
```

```
13-113T20:24:00Z      OSIRIS   Include "POR__PI1RSO_D_IMAGE_SEQ____00010.ROS"
C--------------------------------------------------------------------------------
```

3.2.1.3.5 File format for OIOR file similar to ITL format

The file shall be an ASCII file. Each line shall have a maximum number of 128 characters, including a line feed (<LF>). The date and time of creation and an author shall be given as a comment in the beginning of the file.

Each file may contain header lines that start with a keyword as defined in Table 5. The header is followed by a columnar description of the experiment activities, the columns are defined in Table 6.

Keywords that may appear anywhere in the file are defined in Table 7.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Each column shall be separated by one or more white spaces. Continuation lines can be marked by a "/" at the end of a line.

Table 5: Allowed keywords in the header of the ITL file.

Keyword	Parameters	Comments
`Start_time:`	`<date/time>`	`date/time:` is the start time in CCSDS time format Example: `Start_time: 12-313T20:15:13Z`
`Init_mode:`	`<exp_label>` `<mode_name>`	Mainly used in files to be included. Sets initial conditions for the experiments. `exp_label` is the name of the experiment according to Table 3, column 1 `mode_name` is the instrument mode as defined in the Experiment User Manuals. Example: `Init_mode: OSIRIS STDBY`

`Init_MS:`	`<exp_label>`
	`<module_label>`
	`<MS_label>`
		Mainly used in files to be included. Sets initial conditions for the experiments.
		`exp_label` is the name of the experiment according to Table 3, column 1
		`module_label` is the name of an experiment module as defined in *tbd* and used in the Experiment Description Files for the RSOC Experiment Planning System.
		`MS_label` is the name of a module state as defined in *tbd* and used in the Experiment Description Files for the RSOC Experiment Planning System.

Table 6: File format of the ITL file following the header.

Field name	Column	Format	Comments

<Start_time | <Event_label>	1	All columns shall be space-separated	CCSDS format time (YY-DDDThh:mm:ssZ) or ITL time format (±DDD_hh:mm:ss) If both start and end time are given, the start time is the first allowed time for the execution of the command. Or event name
<End_time | Delta_time | Empty_label>	2		End_time is the end time in CCSDS format time (YY-DDDThh:mm:ssZ) or in ITL time format (±DDD_hh:mm:ss) If both start and end time are given, the end time is the last allowed time for the execution of the command. Delta_time is in the format ±[DDD_]hh:mm:ss where: DDD is the number of days in the range 0 to 999; hh is the number of hours in the range 0 to +23; mm is the number of minutes in the range 00 to 59; ss is the number of seconds in the range 00 to 59; Leading zeros must always be inserted to ensure that the field is 3/2 digits long; or in ITL time format (±DDD_hh:mm:ss). Empty_label means that the field is empty, a dash ('-') has to be given.
<Exp_label>	3		Name of the experiment from column 1 of Table 3
<Mode_label>	4		Name of mode, from the Experiment User Manuals
<Action>	5		Name of the action, must be defined in the *Experiment Description File* for the relevant experiment; or Sequence name as defined in RSDB.
<Parameter(s)>			For the parameter format definition, see Appendix B, Section 7. "Parameters".

Table 7: Including a file in the ITL file.

Field name	Column	Format	Comments
<Start_time | Event_label>	1	20X	CCSDS format time (YY-DDDThh:mm:ssZ) or ITL format time (±DDD_hh:mm:ss)

			If both start and end time are given, the start time is the first allowed time for the execution of the command. Or event label as defined in AD4, annex H.
<End_time | Delta_time | Empty_label>	2	20X	End_time is the end time in CCSDS format time (YY-DDDThh:mm:ssZ) or in ITL time format (±DDD_hh:mm:ss) If both start and end time are given, the end time is the last allowed time for the execution of the command. Delta_time is in the format ±[DDD_]hh:mm:ss where: hhhh is the number of hours in the range -9999 to +9999; mm is the number of minutes in the range 00 to 59; ss is the number of seconds in the range 00 to 59; Leading zeros must always be inserted to ensure that the field is 4/2 digits long; or in ITL time format (±DDD_hh:mm:ss). Empty_label means that the field is empty, a dash ('-') has to be given.
<Exp_label> *optional field!*	3		Name of the experiment from column 1 of Table 4 (optional)
Include	4	7X	keyword to indicate an include file
<file_name>	5	37X	name of file to be included according to definition in Section 3.1.

3.2.1.3.6 Example

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------------------
# Example for an ITL file
# Created 2001 Sep 22, 10:00:00 UTC, dvk

13-113T20:15:17Z -                    OSIRIS   NOMINAL    TAKE_IMAGE (01.000)
# Go to MOVIE mode
13-113T20:16:00Z 2013-313T20:17:00Z OSIRIS   NOMINAL    NOMINAL_TO_MOVIE
13-113T20:16:00Z 2013-313T20:17:00Z OSIRIS   MOVIE      TAKE_MOVIE (T_EXP = 10.0, N = 10)
13-113T20:18:00Z                    OSIRIS   Include "POR__PI1RSO_D_IMAGE_SEQ____00010.ROS"
13-113T20:22:00Z 2013-313T20:23:00Z OSIRIS   MOVIE      TAKE_MOVIE (T_EXP = 20.0, 10)
13-113T20:24:00Z                    OSIRIS   Include "POR__PI1RSO_D_IMAGE_SEQ____00010.ROS"
#-------------------------------------------------------------------------------------------
```

3.2.1.4 Expedite Command Files (ECF)

The use of Expedite Command Files is discouraged for Rosetta. If used, see AD3., Section 7.2, for a definition.

3.2.1.5 Power Profile updates (PP)

3.2.1.5.1 Description

A telecommand or a telecommand sequence may be a combination of many different actions within the actual instrument. Thus, the power may not be a constant value but might change versus time during the execution of the telecommand. In power-critical phases, the scientific operational planning may have to account for such a detailed power profile. This Section specifies the format to describe the power profile linked to a telecommand or telecommand sequence. Note that as a baseline, an average power will be used in the planning. The RSOC together with the Experimenter team will decide on a case by case basis for which commands a more detailed profile is required.

In the case of a new telecommand sequence, the resource requirements may also be specified in the Command Sequence Update Request, see the relevant section.

3.2.1.5.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`PP__`", the source `sor` is "`PIn`" as defined in Table 3, the destination `des` is "`RS0`".

3.2.1.5.3 File format

For actions that are critical with respect to resources, the Experimenter teams shall provide one of the following:

(a) A list of "actions", in which the telecommands can be broken down, with associated power values;

(b) a file describing the necessary power of a telecommand as a function of time.;

(c) a file describing the necessary power of a sequence as a function of time.

The file shall be an ASCII file, the first line of which starts with the keyword "`Action: xxx`" where `xxx` is the mnemonics of the telecommand as defined in the RSDB. The next lines shall list the power versus time as defined in Table 8. The date and time of creation and an author shall be given as a comment in the beginning of the file.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Each column shall be separated by one or more white spaces. Continuation lines can be marked by a "/" at the end of a line.

Table 8: File format for Power Profile changes.

Field name	Column	Format	Comments
<Relative_time>	1	20X	Time after which a power change occurs, relative to the beginning of the telecommand. Format ±[DDD_]hh:mm:sswhere hhhh is the number of hours in the range -9999 to +9999; mm is the number of minutes in the range 00 to 59; ss is the number of seconds in the range 00 to 59. Leading zeros must always be inserted to ensure that the field is 4/2 digits long. Or ITL time format (±DDD_hh:mm:ss).
<Power>	2	20X	In Watt, accuracy ±0.1 W.

3.2.1.5.4 Example

```
0         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
# Example for a power profile change
# Created 2001 Sep 22, 10:00:00 UTC, dvk

Action: ACQUIRE_HISTOGRAM
+000:00:00           2.0
+000:00:02           4.2
+000:00:06           2.0
+000:00:10           0.0
#-------------------------------------------------------------------------------
```

3.2.1.6 Data Volume Profile updates (DVP)

3.2.1.6.1 Description

What was said above for the power is similarly applicable to data volume. Note that the alternative use of "data rate" is still *tbd*.

3.2.1.6.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier ffff being "DVP_", the source sor is "PIn" as defined in Table 3, the destination des is "RSO".

3.2.1.6.3 File format

For actions that are critical with respect to resources, the Experimenter teams shall provide one of the following:

(a) a list of "actions", in which the telecommands can be broken down, with associated power values;

(b) a file describing the necessary power of a telecommand as a function of time.;

(c) a file describing the necessary power of a sequence as a function of time.

The file shall be an ASCII file, the first line of which starts with the keyword "`Action: xxx`" where `xxx` is the mnemonics of the telecommand as defined in the RSDB. The next lines shall list the power versus time as defined in Table 9. The date and time of creation and an author shall be given as a comment in the beginning of the file.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Each column shall be separated by one or more white spaces. Continuation lines can be marked by a "/" at the end of a line.

Table 9: File format for Data Volume Profile changes.

Field name	Column	Format	Comments
`<Relative_time>`	1	20X	Time after which a data volume change occurs, relative to the beginning of the telecommand. Format ±`[DDD_]hh:mm:ss`where hhhh is the number of hours in the range -9999 to +9999; `mm` is the number of minutes in the range 00 to 59; `ss` is the number of seconds in the range 00 to 59. Leading zeros must always be inserted to ensure that the field is 4/2 digits long. Or ITL time format (±DDD_hh:mm:ss).
`<Data volume>`	2	20X	Data volume in kbit (*tbc*) Note: The alternative use of data rate is still *tbd*.

3.2.1.6.4 Example

```
0        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
# Example for a data volume profile change
# Created 2001 Sep 22, 10:00:00 UTC, dvk

Action: ACQUIRE_HISTOGRAM
+000:00:00          0
+000:00:05          1024
+000:01:05          1024
+000:02:05          1024
#-------------------------------------------------------------------------------
```

3.2.1.7 User Defined Programs (UDPs) of OSIRIS

3.2.1.7.1 Description

The Experiment OSIRIS is commanded via so-called User Defined Programs (UDPs). Updating UDPs is similar to memory maintenance and will be handled as follows:

A new UDP shall be defined and documented via a document change request (DCR) to the OSIRIS User Manual. The DCR shall contain, as an appendix, the source code listing of the UDP. In the header of the UDP, between comment characters, shall be a description of the UDP in the format of the Experiment Description File (EDF), 'action', as defined in Appendix B, Section 7. The source of the UDP file shall also be passed to the RSOC as an electronic file (via ftp to the DDS, as described in Section 2.2.1.1). The RSOC shall ingest the UDP into the Experiment Description File for OSIRIS.

Once the change request is accepted, the compiled version of the UDP shall be passed to the s/c via the Service 6 command (see Section 3.4.3.1).

Each UDP shall have a unique ID number which is at any time known to the OSIRIS team, to the RSOC, and the RMOC.

3.2.1.7.2 File naming convention

The file naming convention shall follow the description given in Section 3.1 with the file type identifier `ffff` being "`UDP_`", the source `sor` is "`PIn`" as defined in Table 3, the destination `des` is "`RSO`".

3.2.1.7.3 File format

Two files shall be provided:

(a) An ASCII file with the source code of the UDP, in a format as defined in AD11. In the header of the source code, a description of the code in the EDF format shall be given, this format is described in Appendix B, Section 7.
(b) The actual byte code of the UDP shall be provided as a Service 6 command, see Section 3.4.3.1.

The date and time of creation and an author shall be given as a comment in the beginning of the file.

3.2.1.7.4 Example of the UDP source code

```
0        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
//-----------------------------------------------------------------------------
// This is an example of an OSIRIS UDP file
//
// Author: nt
// Date: 2001 Sep 22, 10:00:00 UTC

/* File description in EDF format

Action: nt_nac_shutter_power_on
Action_parameters: CAM
Run_type: ABSOLUTE
Run_actions:  00:00:00 PowerShutterOn (CAM)
              00:01:00 sleep (sleeptime = 2000)

# NOTE: Assumes that 'PowerShutterOn' and 'sleep' are defined in the
# OSIRIS EDF

*/
// Here starts the OSIRIS UDP

#include "OsirisLib.h"

void nt_nac_shutter_power_on()
{
UNS32 cam;

  cam=NAC;
```

```
  PowerShutterOn(cam);
  sleep(2000);
}
//END
#-------------------------------------------------------------------------
```

3.2.1.7.5 Example of the Service 6 command

tbd

3.2.1.8 Calling UDPs via Telecommand Sequences

3.2.1.8.1 Description

The execution of the loaded UDP actually has to be done via another UDP. To ensure the transparency of the commanding, OSIRIS shall provide two timeline files, one in the ITL format where a model of the calling UDP is given (this will allow RSOC to perform planning), a second one in the OIOR format which will contain the compiled versions of the calling UDPs.

OSIRIS shall define two private telecommands in the RSDB:

- `START_UDP (id)` - this starts a UDP with the ID number `id;`
- `SET_PARAMS (a, b, c, d)` - allows to set parameters.

OSIRIS shall define a telecommand sequence called `CALL_UDP (i, a, b, c, d),` where `id` is the ID number of the UDP as defined in the OSIRIS Symbol Table; a,b c, and d are additional parameters (maximum 4). The definition is given as an example in Section 3.2.1.8.4. The `CALL_UDP` will be used in the timeline file in ITL format.

3.2.1.8.2 File naming convention

See the following section.

3.2.1.8.3 File format

For the actual timeline, OSIRIS shall provide two files:

(a) A file in ITL format (see Section 3.2.1.3.5) where the telecommand sequences are given in a human-readable way. The filename shall follow the naming conventions given in Section 3.1 and shall begin with '`ITL_`';

(b) a second file in POR format, containing the compiled version of the UDPs. The filename shall follow the naming conventions given in Section 3.1 and shall begin with '`OIOR`'. It shall have the same version number as the corresponding ITL file. It is the responsibility of the OSIRIS PI that the OIOR file indeed is identical to the ITL file.

3.2.1.8.4 Example for the definition of CALL_UDP

```
0        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
#-------------------------------------------------------------------------
# Fragment of the EDF file where OSIRIS is modelled
# Assumes that SET_PARAMS and START_UDP are defined elsewhere in the
# OSIRIS EDF. Both of these shall also be defined in the RSDB.

Action: CALL_UDP   # Name shall follow the naming convention in RD10, p. 37
Action_parameters: id, a, b, c, d
Run_actions:
             SET_PARAMS (PAR1 = a  PAR2 = b  PAR3 = c  PAR4 = d)
```

```
            START_UDP (PAR5 = id)

#-------------------------------------------------------------------------
```

3.2.1.8.5 Example for a timeline in ITL format

```
0        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
#-------------------------------------------------------------------------
# Example timeline file in ITL format, calling a UDP with the ID number
#  123, which is assumed to take in image. As parameters, we pass the
# name of the camera (NAC or WAC), the exposure time in ms, and the filter
# wheel positions of the two wheels
# This assumes that a UDP with the ID 123 has been defined in the
# OSIRIS EDF.
#
# Created 2001 Sep 22, 10:00:01 UTC, dvk

13-313T20:13:13Z  OSIRIS  IMAGING_MODE  CALL_UDP (id = 123 \
                                                  a  = WAC \
                                                  b  = 10.0 \
                                                  c  = 2 \
                                                  d  = 1)
13-313T20:14:13Z  OSIRIS  IMAGING_MODE  CALL_UDP (id = 123 \
                                                  a  = WAC \
                                                  b  = 20.0 \
                                                  c  = 1 \
                                                  d  = 2)
#-------------------------------------------------------------------------
```

3.2.1.9 RSI-specific interface to the RSOC

3.2.1.9.1 Description

RSI will have requirements concerning the configuration of the on-board Telemetry and Command subsystem and the ground station RSI IFMS. These shall be transferred to the RSOC in a format compatible with the Experiment Planning System. Details are still *tbd*.

3.2.1.9.2 File naming convention

tbd

3.2.1.9.3 File format

The file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>).
Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

3.2.1.9.4 Example

tbd

3.2.2 Lander-specific interfaces to the RSOC

3.2.2.1 General remarks

Formally, the operations of the Lander will be communicated to ESA via the LOR file. However, the LOR file is to complex to allow science planning directly. Therefore, the Lander team will provide the "Lander Operations Plan" to the RSOC (possibly converted to ITL format, tbc). An overview over the Lander planning process and the interface to the RSOC is given in AD8. The Lander team shall take into account the RSOC contraints in the elaboration of their Experiment Flight Operations Plan (FOP). They shall provide the Lander FOP to the RSOC. The Lander FOP will be used by the RSOC for the science planning of the complete mission.

.

3.2.2.2 Lander Operational Requests (LOR)

3.2.2.2.1 Description

The LOR file is the main means of the Lander team (more specific, the LCC) to communicate their operational requests to the RSOC. The format is identical to that of the OIOR file and we refer to the definition in the previous section. Note that to ensure quick forwarding of this file to the RMOC, there will be a guaranteed acceptance time as specified in Section 2.2.1.2. No format conversion is necessary if the files from the Lander are submitted in the POR format without the additional keywords as previously defined, again to reduce the time needed to forward it to the RMOC and to reduce conversion errors.

Note: Both the OIOR and the LOR file are called Operational Request Files (ORF).

3.2.2.2.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`LOR_`", the source `sor` is "`PIL`", the destination `des` is "`RSO`".

3.2.2.2.3 File format

The format of the LOR is identical to the OIOR file, see Section 3.2.1.3.

3.2.2.3 Lander Parameter information

3.2.2.3.1 Description

The Lander team will, in co-operation with the MOC, propose the spacecraft position, attitude, and time for the separation. This will be provided in a text document very similar to the Scenario Parameter Lists (SPL) used as an interface between the RSOC and the RMOC Flight Dynamics team. The detailed content is still under definition by the Lander team.

3.2.2.3.2 File naming convention

This will be a text document with a number according to the Lander documentation system.

3.2.2.3.3 File format

The file format may be Word, Pdf, or ASCII. The date and time of creation and an author shall be given in the document.

3.2.2.4 Lander Flight Operations Plan (FOP)

3.2.2.4.1 Description

The Lander team will produce a Flight Operations Plan (FOP), which is a human-readable form of the Lander operations. It is generated with a software tool called MOST. This FOP will be converted to the LOR file.

3.2.2.4.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "LFOP", the source `sor` is "`PIL`" as defined in Table 3, the destination `des` is "RSO".

3.2.2.4.3 File format

The file format shall be an ASCII file. The format is defined by the Lander team. It is documented in AD10. The date and time of creation and an author shall be given as a comment in the beginning of the file.

3.2.3 Information from the RSOC to the Experimenter teams

3.2.3.1 Acknowledge file (ACKN)

3.2.3.1.1 Description

For each file sent from the Experimenter teams to the RSOC via the DDS, a software tool called "Operational Request File Acknoledger (ORF-A)" will acknowledge the receipt of this file. It will do this by ftp-ing an acknowledge file as described in this section to a server to be defined by the Experimenter. This ftp server shall be defined in Table 10 by the Experimenters. User name and password shall be transferred via secure means (personal communication).

Table 10: File servers of Experimenter teams for acknowledge files.

Experiment	`server`	Experiment	`server`
ALICE	`tbd`	RPC_IES	`tbd`
CONSERT	`tbd`	RPC_MAG	`tbd`
COSIMA	`tbd`	RPC_ICA	`tbd`
MIDAS	`tbd`	RPC_MIP	`tbd`
MIRO	`tbd`	RSI	`tbd`
OSIRIS	`tbd`	GIADA	`tbd`
ROSINA	`tbd`	VIRTIS	`tbd`
RPC_PIU	`tbd`	LANDER	`tbd`
RPC_LAP	`tbd`	All	`tbd`

3.2.3.1.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "ACKN", the source `sor` is "RSO" as defined in Table 3, the destination `des` is "PIx" with x the identifier for the Experimenter team.

3.2.3.1.3 File format

The file shall be an ASCII file. Each line shall have a maximum number of 128 characters, including a line feed (<LF>).

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

The file shall give a description of the acknowledge in XML format. For the defined keywords, see the example.

3.2.3.1.4 Example

```
0        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
<?xml version="1.0"?>
<!-- ORFA -->
<acknowledgement>
  <receive_date_utc>2001-10-08</receive_date_utc>
  <receive_time_utc>14:45:08</receive_time_utc>
  <receive_date_system>2001-10-08</receive_date_system>
  <receive_time_system>16:45:08</receive_time_system>
<orfa>
    <version>1.0</version>
    <host>redstar.so.estec.esa.nl</host>
    <pid>18473</pid>
    <ftp>
      <destination_server>bear.so.estec.esa.nl</destination_server>
      <user>arjan</user>
      <upload_directory>/home/arjan/temp</upload_directory>
      <transfer_attempts>1</transfer_attempts>
    </ftp>

<receive_filename>/home/arjan/pihome/OIOR_PI7RSO_D_____________0005.ROS</receive_filename>

    <result>failure</result>
    <error>
      <error_number>256</error_number>
      <error_string>incorrect submission number, number too
big</error_string>
      <error_severity>fatal</error_severity>
    </error>
    <epstest>
      <version>0.6</version>
      <output>
      </output>
    </epstest>
    <log_message>
      Submission number sequence incorrect (submitted: 00005)
      File was succesfully ingested into the ESOC-database.
    </log_message>
  </orfa>
</acknowledgement>
```

3.2.3.2 Science Activity Plan (SAP)

3.2.3.2.1 Description

The Science Activity Plan is a time-ordered listing of Mission Scenarios. The current baseline is that this will be a descriptive text document.

3.2.3.2.2 File naming convention

The file will be a text document following the Rosetta numbering scheme: RO-EST-PL-xxxx. The file name shall be of the form `RO-EST-PL-xxxx_I_R_<Short title>.ext` where xxxx is the number of the document, I the issue, R the revision number, `<Short title>` is an optional short title, and `ext` is the extension (`doc` for a Word document, `pdf` for a Portable Document Format document).

3.2.3.2.3 File format

The file will be a text document which will be produced in Word and made available in pdf format. The date and time of creation and an author shall be given in the document.

3.2.3.3 Master Science Plan (MSP)

3.2.3.3.1 Description

The Master Science Plan will be more detailed than the SAP. It will include the experiment operations at a minimum down to mode level.

3.2.3.3.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`MSP_`", the source `sor` is "`RSO`", the destination `des` is "`ALL`".

3.2.3.3.3 File format

This file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>). The date and time of creation and an author shall be given as a comment in the beginning of the file.

Each file lists the time, experiment name and mode or s/c name and pointing request in a columnar way as defined in Table 11.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Each column shall be separated by one or more white spaces. Continuation lines can be marked by a "/" at the end of a line.

Table 11: File format of the MSP file.

Field name	Column	Format	Comments
`<Start_time | Event_label>`	1	20X	CCSDS format time (YY-DDDThh:mm:ssZ) Or ITL time format (±DDD_hh:mm:ss). If both start and end time are given, the start time is the first allowed time for the execution of the command. Or event name as defined in AD4
`<End_time | Delta_time | Empty_label>`	2	20X	`End_time` is the end time in CCSDS format time (YY-DDDThh:mm:ssZ) or in ITL time format (±DDD_hh:mm:ss) If both start and end time are given, the end time is the last allowed time for the execution of the command. `Delta_time` is in the format

			±[DDD_]hh:mm:ss where: DDD is the number of days in the range 0 to 999; hh is the number of hours in the range 0 to 23; mm is the number of minutes in the range 00 to 59; ss is the number of seconds in the range 00 to 59; Leading zeros must always be inserted to ensure that the field is 3/2 digits long; or in ITL time format (±DDD_hh:mm:ss). Empty_label means that the field is empty, a dash ('-') has to be given.
<Exp_label | s/c name>	3	10X	Exp_label is the name of the experiment from column 1 of Table 3 s/c name is ROSETTA
<Exp_Mode | Pointing_ Request>	4	*tbd*	Exp_Mode is the name of the mode as defined in the Experiment User Manuals. Pointing_Request is the pointing mode as defined in AD9.

3.2.3.3.4 Example

Note: The mode names in the following example are fictive.

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
# Example MSP file
# Created 2001 Sep 22, 10:00:02 UTC, dvk
#-------------------------------------------------------------------------------
12-313T22:00:00Z  ROSETTA  NADIR
12-313T22:00:00Z  ROSETTA  ROLL_FIXED
12-313T22:00:00Z  OSIRIS   ANNEAL
12-313T22:00:00Z  MIRO     SUB_MM
12-313T22:00:00Z  MIDAS    EXPOSE
12-313T22:10:00Z  ROSETTA  POWER_OPTIMIZED
#-------------------------------------------------------------------------------
```

3.2.3.4 Final POR file

3.2.3.4.1 Description

The final POR file is available after all the individual OIOR/LORs have been consolidated and iterated with the Experimenter teams to be free of conflicts. They will be sent to the RMOC and at the same time made available to the Experimenter teams via the DDS.

3.2.3.4.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier ffff being "POR_", the source sor is "RSO", the destination des is "RMA" (even if it is made available to the PIs).

3.2.3.4.3 File format

The file format will be as defined in AD3 (CRID), Section 7.1.

To maintain traceability, the following comment lines shall be added at the beginning of the file:

```
C SOURCE_FILE(S): <name_of_source_file(s)>
C GENERATION_DATE: <date_of_generation_in_CCSDS_format>
```

where `name_of_source_file(s)` is the name of the file(s) which were used to generate the final POR, and `date_of_generation_in_CCSDS_format` is the date of the generation of this (not the source) file, in CCSDS format.

3.2.3.4.4 Example

For an example, see AD3 (CRID).

3.2.3.5 Conflict file (CONF)

3.2.3.5.1 Description

As an output of the planning, conflicts might result between different experiments. The Experiment Planning System creates a file listing these conflicts. While this file is actually used only by RSOC internally, it will be distributed for information to the Experimenter teams. This should allow clarifying conflicts more easily.

3.2.3.5.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`CONF`", the source `sor` is "`RSO`", the destination `des` is "`ALL`".

3.2.3.5.3 File format

The file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>). The date and time of creation and an author shall be given as a comment in the beginning of the file.

Each file lists the time and a message for the conflict in a columnar way as defined in Table 12.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Each column shall be separated by one or more white spaces. Continuation lines are marked by a "/" at the end of the line.

Table 12: File format for the reference pointing file.

Field name	Column	Format	Comments
`<Time | Event>`	1	20X	CCSDS format time (YY-DDDThh:mm:ssZ) Or ITL time format (±DDD_hh:mm:ss). Or event label as given in AD4, Appendix H.
`<Conflict>`	2	20X	The conflict message, as defined in the Experiment Description Files.

3.2.3.5.4 Example

Note: The given conflict examples are fictive.

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
# Example CONF file
# Created 2001 Sep 22, 10:00:04 UTC, dvk

12-313T22:00:00Z VIRTIS_COOLER_NOT_OFF_START
12-313T22:00:00Z NAC_HAS_SUN_IN_FOV_START
12-313T23:00:00Z VIRTIS_COOLER_NOT_OFF_END
#-------------------------------------------------------------------------------
```

3.2.3.6 Reference pointing information (RPI)

3.2.3.6.1 Description

For planning purposes, RSOC will provide a *reference pointing information* file, typically one for each Mission Scenario.

Note that during the mission, an attitude information file will be distributed via the DDS from ESOC Flight Dynamics Team (FDT). The format is described in the DDID Annex H (AD4). The files provided by the FDT overrides the RPI files.

3.2.3.6.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`RPI_`", the source `sor` is "`RSO`", the destination `des` is "`PIX`".

3.2.3.6.3 File format

The file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>). The date and time of creation and an author shall be given as a comment in the beginning of the file.

Each file lists the time and a message for the conflict in a columnar way as defined in Table 13. Each column shall be separated by one or more white spaces. Continuation lines are marked by a "/" at the end of the line.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Table 13: File format of the RPI file.

Field name	Column	Format	Comments
`<Start_time | Event_label>`	1	20X	CCSDS format time (YY-DDDThh:mm:ssZ) Or ITL time format (±DDD_hh:mm:ss). If both start and end time are given, the start time is the first allowed time for the execution of the command. Or event label as defined in AD4, Annex H

<End_time | Delta_time | Empty_label>	2	20X	End_time is the end time in CCSDS format time (YY-DDDThh:mm:ssZ) or in ITL time format (±DDD_hh:mm:ss) If both start and end time are given, the end time is the last allowed time for the execution of the command. Delta_time is in the format ±[DDD_]hh:mm:sswhere: DDD is the number of days in the range 0 to 999; hh is the number of hours in the range 0 to 23; mm is the number of minutes in the range 00 to 59; ss is the number of seconds in the range 00 to 59; Leading zeros must always be inserted to ensure that the field is 4/2 digits long; or in ITL time format (±DDD_hh:mm:ss). Empty_label means that the field is empty, a dash ('-') has to be given.
<Pointing_mode>	2	20X	See separate description in AD9.
<Roll_request>	3	10X	field empty: No constraint on roll. 'ROLL_FIXED': It is requested that there is no roll about the +z axis. Note that this request can be fulfilled only in exceptional cases. 'POWER_OPTIMIZED': Solar arrays are oriented perpendicular to sun (typically identically to 'no constraint on roll').

3.2.3.6.4 Example

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
# Example RPI file
# Created 2001 Sep 22, 10:02:00 UTC, dvk

000_00:00:00    NADIR
000_00:01:00    ROLL_FIXED
000_00:01:00    INERT (LAT = 315.15 LONG = +31.12)
#-------------------------------------------------------------------------------
```

3.2.3.7 Reference trajectory information (RTI)

3.2.3.7.1 Description

For planning purposes, RSOC will provide a *reference trajectory information* file, typically one for each Mission Scenario.

Note that later in the mission a trajectory information file will be distributed via the DDS from ESOC Flight Dynamics. The format is described in the DDID Annex H (AD4).

3.2.3.7.2 File naming convention

The current files, provided by the Mission Analysis Section of ESOC, do not conform to the convention used in the previous sections. Currently, the following names are used:

`Trajec.SUN`	Trajectory around the Sun in Sun-centered coordinates
`EarthEscape.EAR`	Escape from the Earth after separation from the upper stage
`Earth1ec.EAR`	1st Earth flyby in Earth-centered coordinates
`Earth1ec.SUN`	1st Earth flyby in Sun-centered coordinates
`Otawaraec.AST`	Otawara flyby in asteroid-centered coordinates
`Earth2ec.EAR`	2nd Earth flyby in Earth-centered coordinates
`Earth2ec.SUN`	2nd Earth flyby in Sun-centered coordinates
`Siwaec.AST`	Siwa flyby in asteroid-centered coordinates

3.2.3.7.3 File format

The file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>).

Each file lists the time and a message for the conflict in a columnar way as defined in Table 14.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

Table 14: File format of the RTI file.

Field name	Column	Format	Comments
`<Time>`	3-14		Time in decimal days since 1 Jan 2000, 0h UT
`<x pos>`	16-31		X position of the s/c in km
`<y pos>`	33-47		Y position of the s/c in km
`<z pos>`	49-63		Z position of the s/c in km
`<x vel>`	65-74		X velocity of the s/c in km/s
`<y vel>`	76-85		Y velocity of the s/c in km/s
`<z vel>`	87-96		Z velocity of the s/c in km/s

3.2.3.7.4 Example

This example gives the trajectory of the Rosetta spacecraft during the first days of the mission in sun-centered coordinates.

```
        1         2         3         4         5         6         7         8         9
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
  1116.1178704    -74937115.159   126714252.275       -3210.514 -36.795701 -14.120886   1.212388
  1117.1178704    -77482287.310   125193115.614       94136.578 -28.793775 -17.669116   1.002390
  1118.1178704    -79951979.521   123650780.449      178791.307 -28.397580 -18.045964   0.964674
  1118.9535289    -81992156.220   122335428.439      247930.319 -28.119352 -18.391786   0.951781
  1119.9535289    -84407663.035   120728235.123      329733.746 -27.796071 -18.812184   0.942529
  1120.9535289    -86795350.004   119084691.180      410880.796 -27.474175 -19.232446   0.936187
  1121.9535289    -89155104.447   117404965.754      491544.653 -27.149025 -19.649377   0.931183
  1122.9535289    -91486552.680   115689418.781      571807.741 -26.818737 -20.061467   0.926833
  1123.9535289    -93789209.355   113938508.966      651710.234 -26.482462 -20.467849   0.922794
  1124.9535289    -96062538.584   112152755.399      731269.958 -26.139822 -20.867956   0.918875
```

3.2.3.8 Reference comet model (RCM)

3.2.3.8.1 Description

For planning purposes, RSOC will provide a *reference comet model*.

At the beginning of the mission, no direct information is available on the comet. Therefore, some standardized models should be available to coordinate the planning process. RSOC will provide a comet reference model for design studies (size, mean density, rotation period, design cases for rotation axis orientation).

Later in the mission, a *comet kinematics file* will be generated by the RMOC (Flight Dynamics) and distributed via the DDS. The format of this file is described in the DDID Annex H (AD4).

3.2.3.8.2 File naming convention

The file will be a text document that follows the Rosetta numbering scheme: RO-EST-SP-xxxx. The file name shall be of the `form RO-EST-PL-xxxx_I_R_<Short title>.ext` where xxxx is the number of the document, `I` the issue, `R` the revision number, `<Short title>` is a short title, and `ext` is the extension (`doc` for a Word document, `pdf` for a Portable Document Format document).

3.2.3.8.3 File format

The file will be a text document which will be produced in Word and made available in pdf format.

3.2.3.9 Reference Flight Dynamics Events (RFDE) file

3.2.3.9.1 Description

For planning purposes, RSOC will provide a *Reference Flight Dynamics Event* (RFDE) file, typically one per mission scenario.

Note that later in the mission a Flight Dynamics Event file will be generated by the RMOC (Flight Dynamics) and distributed via the DDS. The format of this file is described in the DDID Annex H (AD4).

3.2.3.9.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`RFDE`", the source `sor` is "`RSO`", the destination `des` is "`ALL`".

3.2.3.9.3 File format

The file format shall be as described in the RD4 (DDID), Annex H. To ease the planning process, the keywords defined in Table 15 are allowed in addition. The date and time of creation and an author shall be given as a comment in the beginning of the file.

Table 15: Allowed keywords in the header of the RFDE file.

Keyword	Parameters	Comments
`Start_time:`	`<date/time>`	`date/time:` is the start time in CCSDS time format or ITL format. Example: `Start_time: 12-313T20:15:13Z`
`Init_value:`	`<event_label>`	Allows to set an initial value for an event state.

3.2.3.9.4 Example

```
0         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
#
# EPS Event File example (from Smart-I)
# Created 2001 Jan 10, 10:00:05 UTC, smart-I ptb

Start_date: 1-July-2004

Init_time: 000_00:00:00
Init_value: ESTEC_AOS_0
Init_value: TENERIFE_AOS_0
Init_value: KOUROU_AOS_0
Init_value: TENERIFE_AOS_10
Init_value: KOUROU_AOS_10

000_00:00:01   SUBSAT_ILLUM_GE_0
000_00:00:01   SUBSAT_ILLUM_GE_10
000_00:29:23   SUBSAT_ILLUM_GE_20
000_01:08:41   AT_APOLUNE
000_01:41:22   ESTEC_LOS_0
000_02:17:14   SUBSAT_ILLUM_GE_30
000_03:43:31   TENERIFE_LOS_10
000_04:38:52   TENERIFE_LOS_0
000_04:47:14   SUBSAT_ILLUM_GE_45
000_06:37:54   AT_LUNAR_EQUATOR
000_07:19:18   KOUROU_LOS_10
# etc.
#-------------------------------------------------------------------------------
```

3.2.3.10 Derived information

These are graphs and events derived from the attitude and trajectory files, *e.g.* curves with illumination conditions, distances to landmarks, *etc.* Details on what will be provided will depend on the requests of the Experimenter teams.

The RSOC will provide reference data for preliminary planning purposes within the "Mission Scenario Documents, see RD2 for an example.

Later in the mission, the RMOC will generate this information.

3.3 Maintenance Of Rosetta System Database (RSDB)

3.3.1 General Definition

The Rosetta System Database links the telecommands and telecommand sequences of the experiments with the bit patterns which are actually uplinked. The database is in the process of being filled up in a direct interface with ESOC. Updates, however, will go via the RSOC keeping the one-line interface concept to keep the RSOC aware of changes. The following sections describe these updates.

3.3.2 Operational Database In

3.3.3 terface Procedures

3.3.3.1 Command Sequences

3.3.3.1.1 Description

If updates need to be made to the telecommand sequence entries of the RSDB, these can be done via the Command Sequence Update Request. The format of the Command Sequence Update Request follows the format used in the Experiment Description Files (EDFs) used in the science operations planning. To formally document the new or updated telecommand sequences, the updated sequence shall be provided as an electronic file, which is an attachment to a formal change to the respective experiment User Manual.

3.3.3.1.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`CS__`", the source `sor` is "`PIn`" as defined in Table 3, the destination `des` is "`RS0`".

NOTE: The name of the telecommand sequence shall follow the naming convention given in RD10, Section B7.2, p. 41. It is repeated here for convenience:

Each command sequence shall adopt the following naming convention:

- a descriptive name (or command sequence title)
- a unique command sequence ID (8 characters max length)

The command sequence ID shall follow the convention

AXXYnnnZ

where

- A is a fixed character, imposed by the RSDB naming convention, to identify that this RSDB item is a command sequence;
- XX is the two-letter abbreviation for an instrument, according to Table 16,
- Y is always the letter 'S' to denote a pure command sequence,
- nnn is a procedure number,
- Z is 'A', 'B', 'C', .. To be used to distinguish command sequence generated from the same procedure. If a single sequnce is generated from the prcedure, it shall be set to A.

The sequence parameters shall follow the naming convention

VXXYnnnn

where

- V is a fixed character;
- XX and Y are identical to the letters used in the telecommand sequence
- nnnn is a parameter number.

Table 16: Two-letter abbreviations for the experiments

Experiment	Abbreviation	Experiment	Abbreviation
ALICE	AL	ROSINA	RN
CONSERT	CN	RPC	RP
COSIMA	CS	RSI	n/a
MIDAS	MI	GIADA	GD
MIRO	MR	VIRTIS	VR
OSIRIS	SR	LANDER	LZ

3.3.3.1.3 File format

The file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>). The date and time of creation and an author shall be given as a comment in the beginning of the file.

Each file defines a name and the contents for a telecommand sequence following the format definition of "actions" in the Experiment Description Files. The format definition is given in Appendix B, page 43 of this document.

Comment lines are allowed anywhere in the file. They start with the character "#" at the beginning of a line and may be followed by any ASCII character. A "#" in the middle of a line indicates that all following characters will be a comment. Empty lines are allowed anywhere in the file.

3.3.3.1.4 Example

```
0         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#-------------------------------------------------------------------------------
#
# Example of a minimal telecommand sequence update request
# Note that according to the definition of the EDF format, many things can
# be added, e.g. resource requirements, detailed durations, etc.
#
# Created 2001 Oct 11, 10:10:10 UTC, dvk + rh

Action: AALS020A "This sequence is an example from the AFT input"
Action_parameters: VALS0201 VALSO202 VALS0203
Run_type: ABSOLUTE
Run_actions:
000_00:00:00 ZAL19222 (PALD2301 = VALS0201) # discriminator setpoint
000_00:01:00 ZAL19211 (PALD1701 = 30 \
                       PALD1702 = VALS0202)
000_00:02:00 ZAL00301 (PALG0SID = VALS0203)  # default value

#-------------------------------------------------------------------------------
```

3.3.3.2 On-board Control Procedures (OBCPs)

3.3.3.2.1 Description

OBCPs are procedures that are being executed continuously on board the Rosetta spacecraft. OBCPs are built up of regular sequences for an experiment. Information on times for start and stop has to be passed as well.

3.3.3.2.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "OBCP", the source `sor` is "PIn" as defined in Table 3, the destination `des` is "RSO".

3.3.3.2.3 File format

The file shall be an ASCII file. Each line shall have a maximum of 128 characters, including a line feed (<LF>).

The file format follows the OIOR/LOR definition as given in Section 3.2.1.3. In addition, the keywords defined in Table 17 must be provided.

Comment lines are allowed anywhere in the file. They start with the character "C" at the beginning of a line and may be followed by any ASCII character. Empty lines are allowed anywhere in the file.

Table 17: Required keywords in the header of the OBCP update file.

Keyword	Parameters	Comments
`Start_time:`	`<date/time>`	`date/time:` is the start time in CCSDS time format Example: `Start_time: 12-313T20:15:13Z`
`Stop_time:`	`<date/time>`	`date/time:` is the start time in CCSDS time format Example: `Start_time: 12-313T20:15:13Z`

3.3.3.2.4 Example

To come

3.3.3.3 Onboard Monitoring

3.3.3.3.1 Description

The Rosetta spacecraft allows the onboard monitoring of certain items, *e.g.* temperatures of non-operational heaters. Certain activities may be performed if the sensors reach defined limits, *e.g.* start a certain OBCP. An on-board monitoring request is normally done via the Experiment User Manual (UM). To update or add a monitoring request, a Document Change Request to the UM is required. This would be a text document describing the requested item to monitor and the actions if certain limit values are reached. If the resulting action is to perform an OBCP which has not yet been defined in the RSDB, the OBCP shall be provided in the format described in Section 3.3.3.2.

3.3.3.3.2 File naming convention

Text document, following the Experimenters documentation numbering system.

3.3.3.3.3 File format

The file format may be Word, Pdf, or ASCII. The following items shall be clearly identified: Item(s) to be monitored, start and stop time of request, threshold value(s), required action(s). The date and time of creation and an author shall be given in the document.

3.3.3.3.4 Example

```
#-------------------------------------------------------------------------
# This is an example for an on-board monitoring request, it is an ASCII file
#-------------------------------------------------------------------------
# Created 2001 Sep 22, 10:09:01 UTC, dvk
On-board monitoring request for OSIRIS, 11 Mar 2012, dvk

Item to be monitored: Non-op heater no. OSI-001
Start_time: 12-313T00:00:00Z
```

```
Stop_time:  14-313T00:00:00Z
Threshold temperature: -20 deg C
Required action if above threshold: none
Required action if below threshold: notify PI by email, email address is
pi@institute.eu
#--------------------------------------------------------------------------
```

3.3.3.4 Ground Monitoring

3.3.3.4.1 Description

Under certain circumstances, it is possible to request the SOT to monitor the state of different experiments. Currently, it is assumed that each Experimenter team will take care of this activity by itself.

3.3.3.4.2 File naming convention

As in the case of the on-board monitoring request, this shall be documented in the Experiment User Manual. Any updates or additions shall be introduced via a Document Change Request.

3.3.3.4.3 File format

Text document, following the Experimenters documentation numbering system. The date and time of creation and an author shall be given in the document.

3.3.3.4.4 Example

```
#--------------------------------------------------------------------------
# This is an example for a ground monitoring request, it is an ASCII file
#--------------------------------------------------------------------------
# Created 2001 Sep 08, 04:00:00 UTC, jjk

Ground monitoring request for GIADA, 11 Mar 2012, JJK

Item to be monitored: Non-op heater no. GIA-002
Start_time: 12-313T00:00:00Z
Stop_time:  14-313T00:00:00Z
Threshold temperature: -20 deg C
Required action if above threshold: none
Required action if below threshold: notify PI by email, email address is
giada_pi@institute.it
#--------------------------------------------------------------------------
```

3.4 On-board Software Maintenance

3.4.1 Description

The Rosetta payload experiments provide the capability to the user to reprogram their internal software. This is expected to be used during the mission, although as an exceptional measure and not as a routine operation.

Instrument software patches can range in complexity from simple updates of parameters to significant modifications of the running code.

The procedure to be followed (which is detailed below) for patching is dependent on whether or not the instrument supports Service 6 (memory management). For definition of Service 6 refer to EID-C (AD5).

3.4.2 Onboard Software Maintenance Responsibilities

The RSOC is responsible for the consistency between the transferred patch files and the related scheduling requests. Correctness of the patch files is the responsibility of the Experimenter teams, who generated the

files. The RMOC is responsible for the correct execution of the patch activity in strict accordance with the procedures and to make sure that the received patch files are implemented on-board the spacecraft.

3.4.3 Onboard Software Maintenance Procedures

3.4.3.1 Memory Patch Requests for experiments supporting Service 6

3.4.3.1.1 Description

On-board software running in the experiments can be patched by using the Memory Patch Request. More information can be found in AD5 (EID-C), Section 8, p. 9.

3.4.3.1.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "MPR_", the source `sor` is "`PIn`" as defined in Table 3, the destination `des` is "`RMA`".

3.4.3.1.3 File format

The file format shall follow the definition in AD3 (CRID), Section 7.3.

3.4.3.1.4 Example

An example is given in AD3 (CRID), Section 7.3.

3.4.3.2 Memory Dump Requests for experiments supporting Service 6

3.4.3.2.1 Description

A memory dump of the on-board experiment software can be requested with the Memory Dump Request. More information can be found in AD5 (EID-C), Section 8, p. 9.

3.4.3.2.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`MDR_`", the source `sor` is "`PIn`" as defined in Table 3, the destination `des` is "`RMA`" (*tbc*).

3.4.3.2.3 File format

The file format shall follow the definition in AD3 (CRID), Section 7.4.

3.4.3.2.4 Example

An example is given in AD3 (CRID), Section 7.4.

3.4.3.3 Memory Checksum Requests for experiments supporting Service 6

3.4.3.3.1 Description

Verifying the memory checksum can be requested using the Memory Checksum Request. More information can be found in AD5 (EID-C), Section 8, p. 9.

3.4.3.3.2 File naming convention

The file naming convention follows the description given in Section 3.1, with the file type identifier `ffff` being "`MCR_`", the source `sor` is "`PIn`" as defined in Table 3, the destination `des` is "`RMA`".

3.4.3.3.3 File format

The file format shall follow the definition in AD3 (CRID), Section 7.5.

3.4.3.3.4 Example

An example is given in AD3 (CRID), Section 7.5.

3.4.3.4 Instruments Not Supporting Service 6

For this case there are no services available at ESOC or on-board the spacecraft. The following will apply:

- The experiment team shall define in the Rosetta System Database (RSDB) all necessary patch/dump TCs (with parameterized data).
- The experiment team shall define in the Rosetta System Database (RSDB) all necessary TC sequences for OBSM activities.
- The experiment team shall schedule OBSM activities via the OIOR/LOR files.

3.5 Software and Documentation

3.5.1 General Definition

All software modules and all documents that affect the interface specification and/or procedures can be updated during the mission, following the rules listed in this Section.

RSOC documents affecting this interface are listed in Section 1.3.

Relevant Experimenter documents are *e.g.* the User Manuals, descriptions of any software to be run by the RSOC, *etc*.

3.5.2 Software and Documentation Update Responsibilities

Each center is responsible for maintaining the software and documentation produced by the center itself. Changes originated by one center which have impact on the other centers have to be channeled via the Rosetta Project office, which will solve conflicts and provide final approval.

Configuration control is applied in RSOC on all operational software and documentation from a number of months before launch (depending on the item) onwards.

3.5.3 Software and Documentation Update Interface Procedures

3.5.3.1 Software Updates

Whenever software is updated, it shall undergo formal testing and approval. A small change (patch) in software shall increase the revision number. Any major change shall increase the issue number.

3.5.3.2 Documentation Updates

Once a document has been released as "Issue 1", it shall be under formal change control. This means that any change has to be requested via a formal change request approved by the Project Scientist. For minor changes (*e.g.* only requiring single pages to be exchanged in the document), the revision number of the document will increase. Major changes (*i.e.* requiring an update of the complete document) shall increase the issue number of the document.

3.5.4 Standards

3.5.4.1 Documentation standards

Documents written by RSOC will be produced using either of the following two methods:

- Microsoft Word
- Plain ASCII files:
  - maximum line length of 80 characters (128 characters for certain files)
  - <LF> code can be used to aid formatting
  - <TAB> shall ***not*** be used.

RSOC documents are predominantly written using Microsoft Word from which Portable Document Format (pdf) files are generated, currently using Acrobat Writer 3.01.

RSOC will accept hardcopy documents or PDF-files from the Experimenters produced by whatever word processor they have, however, electronic (PDF-) files are preferred; RSOC has no requirement to import document source files.

Documents will be interchanged between RSOC and the Experimenter teams by one of the following methods:

- Electronic mail:
  - Plain ASCII files can be sent "as is"
  - Small Word files should be sent as email attachments
  - Small PDF files should be sent as email attachments
- FTP:
  - It is preferable to transport large PDF files by FTP rather than by Electronic mail
  - If the document is compressed using ZIP, binary mode FTP must be used.
  - The ZIP utilities are described in Technical Note *tbd* which is on the ESTEC document server, this technical note also describes how to obtain these utilities. Details on how to access the ESTEC document server can be obtained from the Project at ESTEC.
- Use of the RSOC documentation server, currently based on the tool *livelink*.
- Paper hardcopy by postal services (supplemented by fax if the circumstances warrant and if the document is not too long) (discouraged!).

3.5.4.2 RSOC-supplied software

3.5.4.2.1 Standards for design & coding

RSOC is utilizing several platforms for developing and running software:

- the IRIX operating system running on SGI Workstations, together with Java and ANSI C;
- Microsoft Windows (Win98, Win2000, and NT) running on Intel-based Personal Computers, together with Java, ANSI C, and IDL (Interactive Data Language).

All software developed by the RSOC will follow the guidelines defined in AD6. A deviation is made in some cases by combining multiple documents into a single physical document. However, all mandatory practices as defined in AD6 will be followed.

Software developed by RSOC for delivery to the Experimenter teams will be designed to be portable where practicable but this will not ensure that it will run without change in other environments. Thus, support for operations in a different environment may be limited. RSOC will, within its design process, assess the portability issues and use best efforts to demonstrate portability.

3.5.4.2.2 Transfer procedure

RSOC will make available the source code of all software written for the science operations planning, *e.g.* the Experiment Planning System (EPS). It will be made available to the Experimenter teams via the RSOC web site or via using the RSOC documentation server.

Supportive documentation (installation notes, test and verification procedures, user notes, (*e.g.* see RD3) will be made available as specified in Section 3.5.4.1.

3.5.4.2.3 Software Support

Software support for software written by the RSOC will be given on a best-effort basis only. Note, however, that each software package developed at the RSOC shall be accompanied with the documentation following ECSS guidelines.

3.5.4.3 Experimenter-supplied software

3.5.4.3.1 Guidelines for design & coding

Software written by the Experimenter teams for execution within RSOC, if any, is expected to conform to the standards specified in AD6, unless overriding circumstances make this impractical. In this case the Experimenter group and RSOC must make a careful assessment of the impact on the Experimenter software expected to run in the RSOC.

The Experimenter groups are implementing software on a variety of platforms but RSOC is utilizing the IRIX system on SGI Workstations, and Microsoft Windows on Personal Computers, together with Java and ANSI C, for developing and running software. Experimenter groups using other platforms are expected to pay particular care to the problems associated with porting their software into the RSOC environment.

3.5.4.3.2 Transfer procedure

RSOC will prefer executable files for any software. Source code and associated build files/procedures must be provided in addition. The transfer media for software from the Experimenter teams to RSOC will be one of the following:

- FTP of source code and associated procedures;
- electronic mail of "small" amounts of the above;
- floppy disk if necessary;
- CD-Rom;
- use of the RSOC documentation server.

Supportive documentation (installation notes, test and verification procedures, user notes) is expected, though the type and extent provided might be negotiated between RSOC and the individual Experimenter teams. Such documentation should be made available as specified in Section 3.5.3.1.

3.5.4.3.3 Software Support

Because of the multiple platforms being used by the Experimenter teams the question of the Experimenter groups providing support to RSOC for this software is a complex issue. It is clear, though regrettable, that Experimenter software developed on source platforms which are highly compatible with the RSOC platform will be easier for the Experimenter team to support in terms of building the PI software at RSOC and trouble shooting it.

Part of the transfer procedure will be for the Experimenter teams to supply installation notes, test and verification procedures and user notes as appropriate to the software.

3.6 Performance Monitoring

3.6.1 Payload health monitoring

Certain housekeeping parameters can be monitored by the RSOC for the Experimenter teams when they are not co-located at ESOC. If a housekeeping value goes out of limits, the Experimenter team will be notified. The formal request for monitoring shall be done in the Experiment User Manuals. To quickly implement monitoring, a format as specified in Section 3.3.3.3 shall be used.

3.6.2 Inter-experiment calibration

Inter-experiment calibration is requested by some experiments in the EID B/LID B documents. The RSOC will consolidate the OIOR/LOR files in such a way to ensure that each request can be fulfilled. In cases where this is not possible, the Project Scientist has the final authority to resolve any conflicts.

3.7 Experiment Specific Issues

tbd

4. INTERFACE VERIFICATION

4.1 Testing

As defined in the RSOC Implementation Plan (RD7), the RSOC will be in the Development Phase until end of 2001. After that, there will be a Pre-launch and Test Phase. As the name implies, this is the time of detailed testing. Different levels of testing are foreseen:

- Individual interface tests between individual Experimenter teams and the RSOC
- Individual interface tests between RMOC and RSOC
- End-to-end tests where the Experimenter supplies OIOR/LOR files, the RSOC consolidates and checks them and forwards them to the RMOC, and the RMOC runs them on the Rosetta simulator
- End-to-end tests with the real spacecraft during the SVT1 (Apr 2002).
- Exercising of the planning concept using the software tools (EPS, PTB) during the commissioning phase

The SVT1 will be the formal verification of the RSOC interfaces. The test shall be documented in a dedicated RSOC SVT Test Report.

The testing between RSOC and RMOC is described in Rosetta Ground Segment System Test Plan (GSSTP).

4.2 Training

RSOC shall perform training sessions for the Experimenter teams for any software they will supply, *e.g.* the Experiment Planning System. Dates and lengths of these training sessions are *tbd*.

5. Schedule

The schedule for the development and operation of RSOC/Experimenter interfaces is given in the RSOC Implementation Plan (AD7).

6. APPENDIX A - ACRONYMS AND ABBREVIATIONS

ALICE	Rosetta experiment: UV spectrometer
APPR	Mission phase: Approach
AST1/2	Mission phase: Asteroid flyby
AST1A	Mission phase: Asteroid 1 approach
AST1P	Mission phase: Asteroid 1 post flyby
AST2A	Mission phase: Asteroid 2 approach
AST2P	Mission phase: Asteroid 2 post flyby
ATP	Approach transition point
BERENICE	Rosetta experiment: Isotopic measurements of cometary volatiles
CAM	Rosetta Navigation Camera
CAP	Comet acquisition point
CAT	Mission phase: Close approach
CLOSE	Mission scenario: Close encounter
CONSERT	Rosetta experiment: Comet Nucleus Sounding Experiment by Radio Transmission
COSIMA	Rosetta experiment: Cometary Secondary Ion Mass Analyzer
CR1..6	Mission phase: Cruise phase 1 to 6
CRID	Command Request Interface Document
CVP	Mission phase: Commissioning phase
DCR	Document Change Request
DDID	Data Delivery Interface Document
DRIF	Mission phase: Comet drift
EAR1/2	Mission phase: Earth flyby
ECSS	European Committee on Space Standardization
EID	Experiment Interface Document
EOID	Experiment Operations Interface Document
EPS	Experiment Planning System
ESOC	European Space Operations Centre
EXT	Mission phase: Extended mission
FAT	Mission phase: Far approach
FOP	Flight Operations Plan
GIADA	Rosetta experiment: Grain Impact Analyzer and Dust Accumulator
GMP	Global mapping insertion point
GSSTP	Ground Segment System Test Plan
HIGH	Mission phase: Comet activity - high activity
ICA	Ion Composition Analyzer (part of RPC)
ICD	Interface Control Document
IES	Ion Electron Sensor (part of RPC)
INERT	Pointing mode: Inertially fixed pointing
INSCI	Mission scenario: Initial science
INSLW	Pointing mode: Slewing between two inertial positions
ITL	Input Timeline
LAN	Local Area Network
LAP	Langmuir Probe (part of RPC)
LEOP	Mission phase: Launch phase
LIMB	Pointing mode: Limb pointing
LIMBOF	Pointing mode: Limb pointing with offset
LOW	Mission phase: Comet low activity phase
MAG	Fluxgate Magnetometer (part of RPC)
MARS	Mission phase: Mars flyby

MIDAS	Rosetta experiment: Micro-Imaging Dust Analysis System
MINC	Mission phase: Comet activity: moderate increase
MIP	Mutual Impedance Probe (part of RPC)
MIRO	Rosetta experiment: Microwave Instrument for the Rosetta Orbiter
MOST	tbd (a software tool used by the Lander team to generate their FOP)
NADIR	Pointing mode: Nadir pointing
NADOF	Pointing mode: Pointing with a given offset to the nadir
NMCO	Mission phase: Nucleus mapping/close encounter
OCL	OSIRIS command language
OIP	Orbit insertion point
ORF-A	Operation Request File Acknowledger
OSIRIS	Rosetta experiment: Science camera
PERI	Mission phase: Near perihelion
PI	Principal Investigator
PODT	Payload Operations Dynamical Timeline
POR	Payload Operations Request
RDVM	Rendezvous maneuver
RMOC	Rosetta Mission Control Centre
ROSINA	Rosetta experiment: Rosetta Orbiter Spectrometer for Ion and Neutral Analysis
RPC	Rosetta experiment: Rosetta Plasma Consortium
RPI	Reference Pointing Information
RSOC	Rosetta Science Operations Centre
SINC	Mission phase: Comet activity: sharp increase
SOT	Science Operations Team
SSP	Surface Science Package (also referred to as "Lander" or "Rosetta Lander")
START	Start of mission
TGM	Mission phase: Transition to global mapping
TRACK	Pointing mode: Tracking an object on comet
UDP	User Defined Program (used by OSIRIS)
VIRTIS	Rosetta experiment: Infrared spectrometer

7. Appendix B: Format definition of the Experiment Description File - template, notes, and example

```
#
# Filename: experiment.template
#
# Authors:  Peter van der Plas & Raymond Hoofs
# Date:     5 October 1999
#
# (c) ESA/Estec
#
# $Id: experiment.template,v 1.12 2001/07/04 13:49:54 rosetta Exp $
#
# $Log: experiment.template,v $
# Revision 1.12  2001/07/04 13:49:54  rosetta
# Renamed some parameter keywords.
# Added keyword Action_dataflow.
# Split keywords Run_actions and Condition.
# Applied some small textuals corrections.
#
# Revision 1.11  2001/05/23 11:41:43  rosetta
# Implementation of parameters added, arguments removed.
# Units may now be specified on all values.
# MTL_command, compression and data_volume keywords added.
#
# Revision 1.10  2000/12/08 11:33:41  rosetta
# Added the dataflow keyword.
#
# Revision 1.9  2000/10/31 16:36:41  rosetta
# Counter keywords now optional.
# Update_when_ready has multiple arguments.
# Floating point values not starting with E or e.
#
# Revision 1.8  2000/03/13 19:33:25  rosetta
# Changed implementation of modules
# Added module power an data rate
# Added module level and sub-modules
# Changed action type to action level
# Various minor changes and document updates
#
# Revision 1.7  2000/02/21 16:51:17  rosetta
# Added more detailed description on labels and values.
#
# Revision 1.6  1999/12/20 16:54:20  rosetta
# Changed terminology: changed instrument into experiment
# Added the area section, added FOV blinding duration
# Removed action detail level, added mode actions
# Implemented some cosmetic changes
#
# Revision 1.5  1999/11/29 08:52:18  rosetta
# Added mode classes INITIALISE and MAINTENANCE,
# Removed constraint severity OK.
#
# Revision 1.4  1999/11/22 17:14:34  rosetta
# Changed average power and bitrate to equivalent
# power and bitrate for the mode, now with average
# value, peak and low value.
```

```
#
# Revision 1.3  1999/11/11 13:34:18  peter
# Updated after some thourough internal review.
#
# Revision 1.2  1999/10/07 14:56:21  peter
# Updated after internal review.
#
# Revision 1.1  1999/10/06 10:08:17  peter
# Initial revision
#
#
--- template definition...
#
# <Experiments start here...>
#
Nr_of_experiments: <i>
#
Include_file: <filename1>
Include_file: <filename2>
...
--- each experiment in one file...
#
# <Experiment definition starts here...>
#
Experiment: <label> [<description>]
#
# <Global properties start here...>
#
Local_memory: <value> [<unit> (Mbytes)]
Dataflow: <TO|FROM> <experiment>
Global_actions: <action #1> <action #2> ...
Global_constraints: <constraint #1> <constraint #2> ...
#
# <Field of views start here...>
#
Nr_of_FOVs: <j>
#
   FOV: <label> [<description>]
   FOV_lookat: <X> <Y> <Z>
   FOV_upvector: <X> <Y> <Z>
   FOV_geometric_angles: <width> <height> [<unit> (degrees)]
   FOV_straylight_angles: <width> <height> [<unit> (degrees)]
   FOV_straylight_duration: <duration> [<unit> (seconds)]
   #
   ... (j FOVs)
#
# <Areas start here...>
#
Nr_of_areas: <k>
#
   Area: <label> [<description>]
   Area_orientation: <X> <Y> <Z>
   Area_lighting_angle: <halfcone angle> [<unit> (degrees)]
   Area_lighting_duration: <duration> [<unit> (seconds)]
   #
   ... (k areas)
#
# <Modules start here...>
#
```

```
Nr_of_modules: <m>
#
   Module: <label> [<description>]
   Module_level: <LEVEL1|LEVEL2|LEVEL3>
   Sub_modules: <module #1> <module #2> ...
   Nr_of_module_states: <n>
      Module_state: <label> [<description>]
      MS_power: <average> [<peak> [<low>]] [<unit> (Watts)]
      MS_data_rate: <average> [<peak> [<low>]] [<unit> (bits/sec)]
      MS_constraints: <constraint #1> <constraint #2> ...
      Repeat_action: <action [(parameter list)]>
      #
      ... (n module states)
   #
   ... (m modules)
#
# <Modes start here...>
#
Nr_of_modes: <p>
#
   Mode: <label> [<description>]
   Mode_class: <OFF|INITIALISE|STANDBY|MEASUREMENT|CALIBRATION|MAINTENANCE>
   Module_states: <module #1> <module state #1> \
                  <module #2> <module state #2> ...
   Nominal_power: <value> [<unit> (Watts)]
   Nominal_data_rate: <value> [<unit> (bits/sec)]
   Equivalent_power: <average> [<peak> [<low>]] [<unit> (Watts)]
   Equivalent_data_rate: <average> [<peak> [<low>]] [<unit> (bits/sec)]
   Mode_transitions: <action #1> <action #2> ...
   Mode_actions: <action #1> <action #2> ...
   Mode_constraints: <constraint #1> <constraint #2> ...
   #
   ... (p modes)
#
# <Parameters start here...>
#
Nr_of_parameters: <q>
#
   Parameter: <label> [<description>]
   Raw_type: <BOOL|ENUM|UINT|INT|REAL|STRING|DATE|TIME>
   Eng_type: <UINT|INT|REAL|TEXT>
   Default_value: <value> [[<unit>|RAW|ENG] [FIXED]]
   Unit: <unit>
   Raw_limits: <min> <max>
   Eng_limits: <min> <max> [<unit>]
   Resource: <DURATION|COMPRESSION| \
               (AVG_|PEAK_|LOW_|ALL_)POWER_INCREASE| \
               (AVG_|PEAK_|LOW_|ALL_)DATA_RATE_INCREASE| \
               (AVG_|PEAK_|LOW_|ALL_)DATA_VOLUME>
   Nr_of_parameter_values: <r>
      Parameter_value: [<value>] [<label>]
      Parameter_update: <MODE|MS|MSP> <label> [<label>]
      Parameter_run: <action [(parameter list)]>
      #
      ... (r parameter values)
   #
   ... (q parameters)
#
# <Actions start here...>
```

```
#
Nr_of_actions: <s>
#
   Action: <label> [<description>]
   Action_level: <LEVEL1|LEVEL2|LEVEL3>
   MTL_command: <YES|NO>
   Action_parameters: <parameter #1> \
                         [= <value #1> [[<unit>|RAW|ENG] [FIXED]]] \
                      <parameter #2> \
                         [= <value #2> [[<unit>|RAW|ENG] [FIXED]]] ...
   Duration: <action duration> [<unit> (seconds)]
   Compression: <factor>
   Action_dataflow: <TO|FROM> <experiment>
   Power_increase: <average> [<peak> [<low>]] [<unit> (Watts)]
   Data_rate_increase: <average> [<peak> [<low>]] [<unit> (bits/sec)]
   Data_volume: <average> [<peak> [<low>]] [<unit> (bytes)]
   Update_when_ready: <MODE|MS|MSP> <label> [<label>] ...
   Action_constraints: <constraint #1> <constraint #2> ...
   Run_type: <SEQUENTIAL|PARALLEL|RELATIVE|ABSOLUTE>
   Run_actions: [<delta time #1>] <action #1 [( \
                   <parameter #1> [= <value #1> [<unit>|RAW|ENG]] \
                   <parameter #2> [= <value #2> [<unit>|RAW|ENG]] ... )]> \
                [<delta time #2>] <action #2 [( \
                   <parameter #3> [= <value #3> [<unit>|RAW|ENG]] \
                   <parameter #4> [= <value #4> [<unit>|RAW|ENG]] ... )]> \
                ...
   #
   ... (s actions)
#
# <Constraints start here...>
#
Nr_of_constraints: <t>
#
   Constraint: <label> [<description>]
   Constraint_type: <TIME|GEOMETRIC|PARAMETRIC|CAPACITY>
   Severity: <FATAL|ERROR|WARNING|INFO>
   Condition: <EVENT|MODE|MS|ACTION> <IS|NOT> <label #1> ...
   Condition_experiment: <experiment>
   #
   ... (t constraints)
#
# <File ends here...>
#
--- include the next file...
... (i experiments)
```

```
#
# Filename: experiment.notes
#
# Author: Peter van der Plas
# Date:   6 October 1999
#
# (c) ESA/Estec
#
# $Id: experiment.notes,v 1.12 2001/07/04 13:49:54 rosetta Exp $
#
# $Log: experiment.notes,v $
# Revision 1.12  2001/07/04 13:49:54  rosetta
# Renamed some parameter keywords.
# Added keyword Action_dataflow.
# Split keywords Run_actions and Condition.
# Applied some small textuals corrections.
#
# Revision 1.11  2001/05/23 11:41:43  rosetta
# Implementation of parameters added, arguments removed.
# Units may now be specified on all values.
# MTL_command, compression and data_volume keywords added.
#
# Revision 1.10  2000/12/08 11:33:41  rosetta
# Added the dataflow keyword.
#
# Revision 1.9  2000/10/31 16:36:41  rosetta
# Counter keywords now optional.
# Update_when_ready has multiple arguments.
# Floating point values not starting with E or e.
#
# Revision 1.8  2000/03/13 19:33:25  rosetta
# Changed implementation of modules
# Added module power an data rate
# Added module level and sub-modules
# Changed action type to action level
# Various minor changes and document updates
#
# Revision 1.7  2000/02/21 16:51:17  rosetta
# Added more detailed description on labels and values.
#
# Revision 1.6  1999/12/20 16:54:20  rosetta
# Changed terminology: changed instrument into experiment
# Added the area section, added FOV blinding duration
# Removed action detail level, added mode actions
# Implemented some cosmetic changes
#
# Revision 1.5  1999/11/29 08:52:18  rosetta
# Added mode classes INITIALISE and MAINTENANCE,
# Removed constraint severity OK.
#
# Revision 1.4  1999/11/22 17:14:34  rosetta
# Changed average power and bitrate to equivalent
# power and bitrate for the mode, now with average
# value, peak and low value.
#
# Revision 1.3  1999/11/11 13:34:18  peter
# Updated after some thourough internal review.
#
# Revision 1.2  1999/10/07 14:56:21  peter
```

```
# Updated after internal review.
#
# Revision 1.1  1999/10/06 10:07:48  peter
# Initial revision
#
#

A few guidelines on the experiment description files:

Basic syntax
============
- comment lines start with '#'
- all characters after '#' are ignored
- a '\' indicates a continuation line: it allows keyword values to
  continue on the next line
- a '\' should be the last character of a line, otherwise it is ignored
- white spaces/tabs are used to separate words
- a keyword is followed by a ':' and then by its keyword values
- no white spaces within keywords/labels, use '_' instead
- keyword/label recognition is not case sensitive
- '|' stands for an exclusive or
- values between [ ] brackets are optional
- an optional number of values is indicated by ...
- all keywords (lines in the file) are optional, except when resulting
  in an inconsistent definition
- a <description> starts and ends with a '"', upper/lowercase is
  maintained here, white spaces and other special characters are allowed

Use of labels
=============
- a <label> is the base type for identifiers, units and any values
- no white spaces are allowed
- a label shall consist of 'a..z', 'A..Z', '0..9' or any of '_+-.:/'
- the '_' and '-' characters are considered different
- the '_' character is considered as a letter, e.g. the identifier
  ABCD is not the same as the identifier AB_CD
- labels are not case sensitive, e.g. the identifier ABCD is the same
  as identifier aBcD
- labels used as identifier, e.g. as experiment label, mode label, etc,
  shall not contain any of '+-.:/'
- labels used as units shall not contain any of '+-.:', the '/'
  character is allowed here indeed
- all identifiers need to be unique within their specific category,
  e.g. for all actions each label needs to be different, but e.g.
  two different modules may both have a module state that is <OFF>
- a <keyword> is considered as an identifier followed by a ':'

Use of units
============
- each (engineering) value may have an optional unit specifier
- the use of a unit for a parameter value automatically implies
  that the value is given as an engineering value
- the unit label needs to be specified between [ ] brackets
- if no unit is specified then the default unit will be applied
- unit sets will define the various unit options for a single
  physical unit, e.g. Volts may be V, mV, volts, mvolts, etc
- the unit specified for a value needs to be compatible with the
  value's unit set, if a unit set is applicable for the value
- unit conversions to the default unit will be done automatically
```

```
- for keywords with multiple values, a unit may either be given
  for each value, or a unit may be given after the last value,
  setting the same unit also on all previous values

Use of values
=============
- <value> means a single value
- an unsigned integer value shall consist of '0..9' or '+'
- an unsigned integer may also be given in hexadecimal or octal format
- a hexadecimal value is in the format '0xhhhhhhhh', where h is one of
  '0-9', 'a-f' or 'A-F' and where the number of h ranges from 1 to 8
- an octal value is an unsigned integer value that starts with a '0'
  and only consists of any of the '0-7' characters
- a (signed) integer value shall consist of '0..9', '+' or '-'
- a floating point value shall consist of '0..9', '+', '-', '.', 'e' or 'E'
- a floating point value shall be in the IEEE format

Date values
===========
- a date value may be given in the POR or EPS absolute time format
- an EPS <date> value is in the format [dd-month-yyyy[_hh:mm:ss]]
- [dd] is the day number
- [dd] may consist of one or two digits, and may start with zero
- [month] is the full (spelled out) month or any abbreviation
  with a minimum of 3 characters
- [yyyy] is the full year number
- [_hh:mm:ss] is optional and is defined similarly as in the time format
- the '_' character is mandatory here if the time of the day is given
- the time of the day defaults to _00:00:00
- a POR absolute time value is in the format [yy-dddThh:mm:ss[.mmm]Z]
- [yy] is the year in the 21st century and must have two characters
- [ddd] is the day number within the year, counting from 1
- [hh:mm:ss] is defined similarly as in the time format
- [.mmm] is optional and specifies the number of milliseconds
- the EPS software will always ignore the [.mmm] value

Time values
===========
- a time value may be given in the POR or EPS relative time format
- an EPS <time> field is in the format [[sign][ddd_]hh:mm:ss]
- [sign] is an optional sign ('-' or '+')
- [ddd] is an optional day number
- [ddd] may consist of one, two or three digits, and may be zero
- [hh] is the number of hours (00..23)
- [mm] is the number of minutes (00..59)
- [ss] is the number of seconds (00..59)
- [hh], [mm], [ss] must be specified and must have two characters each
- a POR relative time value is in the format [[-][ddd.]hh:mm:ss[.mmm]]
- [ddd] is the optional number of days
- [hh], [mm], [ss] is defined similarly as above
- [.mmm] is optional and specifies the number of milliseconds
- the EPS software will always ignore the [.mmm] value

Parameter lists
===============
- an action may have an optional parameter list defined between ( )
- a parameter list consists of any number of parameters
- each parameter may have an optional value
- a '=' is used to separate a parameter from its value
```

- a value may have an optional unit, or an indication if the value
  is of RAW or ENG type
- the use of a unit automatically implies the ENG type
- a value may have the FIXED qualifier indicating that the parameter
  value may not be changed from a higher level calling action
- value options such as units need to be given between [ ] brackets

Counter keywords
================
- the counter keywords (i.e. starting with Nr_of_) are optional
- if a counter keyword is specified for a section, then its number
  needs to be correct
- in this case all the section instances are expected right after the
  counter keyword
- if the counter keyword is not specified for a section, then the
  number of instances for a section is determined automatically
- in this case the instances may be even spread, but must be within
  the same section level

Global data
===========
- All actions defined as either global actions or mode actions are
  considered to be command sequences callable from the POR file
- All command sequences and parameters used in the command sequences
  should have unique names on a spacecraft wide basis (so not only
  within a single experiment)
- All command sequence names and their parameter names shall not be
  longer than 8 characters
- All unit labels used in command sequence parameters shall not be
  longer than 4 characters

Overall
=======
- a TO dataflow will point to a mass memory device
- if an experiment has no local memory, then data is directly send to the
  mass memory device, no negative datarates are allowed here, otherwise
  negative datarates are used for data downlink to the mass memory
- a FROM dataflow will point to a mass memory device
- in this case the experiment is not allowed to have any local memory
- the experiment is not allowed to have any negative datarates,
  a positive datarate is used for data downlink from mass memory to ground
- a mass memory device must not have its own dataflow defined
- a mass memory device must have local memory defined
- a mass memory device must not not have any negative datarates
- global actions are the actions allowed for an experiment regardless
  of the current experiment mode, these actions may appear in the
  input timeline
- global constraints are constraints which are always checked regardless
  of the current experiment mode

FOVs
====
- any number of FOVs are allowed per experiment
- the geometric angles specify the scientific FOV
- the blinding angles can be used to determine if Sun blinding occurs
- constraints can be put on the FOVs by the SUN_IN_FOV event (reported
  immediately when the Sun shines in the FOV) and the SUN_IN_FOV_DURATION
  event (when the Sun shines in the FOV for a duration longer than
  specified in the experiment description file)

- either global contraints or mode dependent constraints can be put on FOVs

Areas
=====
- any number of areas are allowed per experiment
- the lighting angle specifies when the Sun is shining on the area, this is the halfcone angle between the Sun vector and the area orientation
- constraints can be put on the areas by the SUN_ON_AREA event (reported immediately when the Sun shines on the area) and the SUN_ON_AREA_DURATION event (when the Sun shines on the area for a duration longer than specified in the experiment description file)
- either global contraints or mode dependent constraints can be put on areas

Modules
=======
- any number of modules are allowed per experiment
- the default module level if LEVEL1
- a module with a certain level can only have sub-modules of equal or higher detail level, e.g. a module at LEVEL2 may have a sub-module at LEVEL2 or LEVEL3, but may not have a sub-module at LEVEL1
- modules are computed starting from toplevel modules, toplevel modules are modules which are not sub-modules from other modules
- a sub-module may only be referenced by one single module
- recursive modules (causing endless loops) are not allowed
- any number of module states are allowed per module
- the first module state is used as the default state
- the values for a module state specify the average, peak and low power and bitrate values, peak and low values are optional, the low value may only be specified if the peak value is given
- a positive bitrate indicates data production inside the experiment (so local memory will be occupied), a negative bitrate indicates data download on the spacecraft bus (so local memory is released)
- a list of constraints can be given for a certain module state
- the repeat action is an optional action which is executed repeatedly when the module is in the given state
- the repeat action is a single action with optional parameters

Modes
=====
- any number of modes are allowed per experiment
- the first mode is assumed to be the default
- the default mode class is OFF
- the module states shall be defined as pairs of module and module state, no more than one module state may be given for a single module here
- when a module state is explicitly defined here, the module state is not allowed to be update from an action during the mode
- the nominal values for a mode specify the power and bitrate values for the mode, without any specific actions running
- the equivalent values for a mode specify the average, peak and low power and bitrate values, peak and low values are optional, the low value may only be specified if the peak value is given
- a positive bitrate indicates data production inside the experiment (so local memory will be occupied), a negative bitrate indicates data download on the spacecraft bus (so local memory is released)
- the equivalent values can be used without having to compute the detailed values resulting from various actions
- per mode the allowed mode transitions can be specified, these mode transitions may appear in the input timeline
- per mode the allowed mode actions can be specified, these actions

```
  may appear in the input timeline
- per mode any number of constraints can be specified

Parameters
==========
- any number of parameters are allowed per experiment
- a parameter shall have either its raw value type or its engineering
  value type defined
- the value types matching the possible parameter types are described
  above, the raw STRING type should be specified as a string using '"'
  and may contain any characters, the engineering TEXT type is defined
  as an identifier, so as a label not containing any of '+-.:/'
- it is possible to define both the raw and engineering types
- if both types are defined then for each value reference an explicit
  RAW or ENG qualifier has to be given
- if only one of the raw or engineering value types is defined, the RAW
  or ENG qualifier may be omitted when specifying the parameter value
- the use of a unit label will automatically imply the ENG qualifier
- no units can be given on raw parameter values and limits
- it is possible to have a FIXED default value for a parameter, to
  prevent changing the value from a higher level calling action
- a parameter unit may only be given if the engineering type is REAL
- the raw or engineering limits may only be given if the matching
  value type is defined
- if a limit set is specified, both lower and upper limits must be given
- the raw STRING type and engineering TEXT type may not have any limits
- the parameter resource specifies what action resource the parameter
  will update, but it is applied only if the action does not have any
  sub-actions to which the parameter is passed
- the parameter resource of POWER_INCREASE, DATA_RATE_INCREASE and
  DATA_VOLUME will have an effect on the average value
- it is possible here to specify peak, low or all three values of the
  specified resource to be effected
- Parameters changing a resource must be of engineering type REAL and
  have to be called with the ENG qualifier if also the raw type is defined
- for parameters with raw type BOOL, ENUM, UINT or INT, and for parameters
  with engineering type TEXT, any number of parameter values may be defined
- depending on the raw or engineering type either the parameter values
  or text labels may be given (or both)
- if values are defined for a parameter, then the validity of either the
  raw value or the engineering text will be verified when the parameter
  value is set
- it is possible to specify a mode or module state change for a specific
  parameter value or text label
- the following parameter update changes can be specified:
     MODE refers to mode by <mode>
     MS   refers to module state by <module> <module state>
     MSP  refers to module state by <module state>
- the MSP defines a parametrised module state update where the module
  will be defined in the action using the parameter
- it is possible to specify an action to be run for a specific parameter
  value or text label, this action is run from the start of the current
  action and in parallel with any sub-actions of the current action
- this action must be a single action with optional parameters

Actions
=======
- any number of actions are allowed per experiment
- the default action level if LEVEL1
```

- an action with a certain level can only call actions of equal or higher detail level, e.g. an action at LEVEL2 may call an action at LEVEL2 or LEVEL3, but may not call an action at LEVEL1
- any action level can be called from the input timeline, presuming that the action is either allowed in the mode or if it is a global action
- the default MTL_command value if NO
- any number of action parameters can be specified
- it is possible to define a default value for each parameter
- an optional default value unit may be specified, which has to be compatible with the parameter unit (if any)
- if a value unit is specified, the value is assumed to be an engineering value
- the value type specification RAW or ENG is optional and defaults to the value type of the parameter
- if the parameter value type is defined for both raw and engineering values, then the RAW or ENG qualifier is mandatory (or a unit may be defined, in which case the ENG type is defined implicitly)
- the action duration is optional
- for actions with sub-actions, the action duration is computed from the duration of the sub-actions, in this case the specified action duration is ignored
- the EPS execution level may cause the sub-actions not to be called, in which case the current action duration is used instead
- the compression factor default value is 1.0
- the action dataflow may overwrite the experiment dataflow for the current action, the definition has to follow the same rules as defined for the experiment dataflow
- for the power and bitrate increment values, and for the data volume value, the average, peak and low values can be given, peak and low values are optional, the low value may only be specified if the peak value is given
- a positive bitrate or data volume indicates data production inside the experiment (so local memory will be occupied), a negative bitrate or data volume indicates data download on the spacecraft bus (so local memory is released)
- a mode change and/or a number of module state changes can be specified, these are executed when the action is completed
- only a single MODE change is allowed
- any number of MS or MSP changes are allowed
- a MODE change may be combined with MS or MSP changes
- the following update-when-ready changes can be specified:
  MODE refers to mode by <mode>
  MS refers to module state by <module> <module state>
  MSP refers to module state by <module> <parameter>
- a MS or MSP in an update-when-ready change is only allowed when the module state is not explicitly defined in the current experiment mode
- a MSP must use a parameter defining a parametrised module update
- recursive actions (causing endless loops) are not allowed
- the default run actions type is SEQUENTIAL, this field may be omitted
- the run actions list may contain any number of actions
- for each action an optional delta time may be defined
- if the delta time is not defined it defaults to zero
- the SEQUENTIAL type executes actions in sequence, the specified action delta times are extra time gaps to be inserted after the previous action completion time and before the current action start time
- the PARALLEL type executes all actions at once, the specified action delta times are applied from the start of the current action, so resulting in an execution timeline with absolute values with respect to the action start time

- the RELATIVE run actions type behaves similarly as the SEQUENTIAL type, with the difference that the delta time is measured from the start of the previous action to the start of the current action
- the ABSOLUTE run actions type behaves similarly as the PARALLEL type, and is only available to make the behaviour of the delta time value more obvious
- delta times may never be negative
- in PARALLEL or ABSOLUTE mode the delta times need to be consecutive
- for each action an optional parameter list may be defined, this list is described above, however the FIXED qualifier is not allowed here
- an action is not allowed to be instanciated more than once, i.e. if an action is running then the same action can not be started either from the input timeline or from another action

Constraints
==========
- the default constraint type is TIME
- the default constraint severity is WARNING
- constraint condition either "is"/"is not" relation
    EVENT refers to a global event by <event> label
    MODE refers to mode by <mode>
    MS refers to module state by <module> <module state>
    ACTION refers to action by <action>
- note that the constraint happens if the condition is valid
- events may have an optional argument if required (e.g. for a FOV event to define which FOV should be used)
- the constraint normally applies to the current experiment, unless the constraint experiment is defined

```
#
# Filename: experiment.example
#
# Author: Peter van der Plas
# Date:   5 October 1999
#
# (c) ESA/Estec
#
# Description: This is an example experiment description file for MIDAS.
#
# $Id: experiment.example,v 1.12 2001/07/04 13:49:54 rosetta Exp $
#
# $Log: experiment.example,v $
# Revision 1.12  2001/07/04 13:49:54  rosetta
# Renamed some parameter keywords.
# Added keyword Action_dataflow.
# Split keywords Run_actions and Condition.
# Applied some small textuals corrections.
#
# Revision 1.11  2001/05/23 11:41:43  rosetta
# Implementation of parameters added, arguments removed.
# Units may now be specified on all values.
# MTL_command, compression and data_volume keywords added.
#
# Revision 1.10  2000/12/08 11:33:41  rosetta
# Added the dataflow keyword.
#
# Revision 1.9  2000/10/31 16:36:41  rosetta
# Counter keywords now optional.
# Update_when_ready has multiple arguments.
# Floating point values not starting with E or e.
#
# Revision 1.8  2000/03/13 19:33:25  rosetta
# Changed implementation of modules
# Added module power an data rate
# Added module level and sub-modules
# Changed action type to action level
# Various minor changes and document updates
#
# Revision 1.7  2000/02/21 16:51:17  rosetta
# Added more detailed description on labels and values.
#
# Revision 1.6  1999/12/20 16:54:20  rosetta
# Changed terminology: changed instrument into experiment
# Added the area section, added FOV blinding duration
# Removed action detail level, added mode actions
# Implemented some cosmetic changes
#
# Revision 1.5  1999/11/29 08:52:18  rosetta
# Added mode classes INITIALISE and MAINTENANCE,
# Removed constraint severity OK.
#
# Revision 1.4  1999/11/22 17:14:34  rosetta
# Changed average power and bitrate to equivalent
# power and bitrate for the mode, now with average
# value, peak and low value.
#
# Revision 1.3  1999/11/11 13:34:18  peter
# Updated after some thourough internal review.
```

```
#
# Revision 1.2  1999/10/07 15:50:43  peter
# Added RCS tags.
#
#
Experiment: MIDAS "The MIDAS experiment"
#
Local_memory: 32 [Mb]
Dataflow: TO SSMM
Global_actions: ARM_FOLD ARM_UNF
Global_constraints: SUN_BLINDING
#
Nr_of_FOVs: 1
#
   FOV: CAMERA_FOV "The hi-resolution camera pointing to +Z"
   FOV_lookat: 0.0 0.0 1.0
   FOV_upvector: 1.0 0.0 0.0
   FOV_geometric_angles: 30.0 15.0 [degrees]
   FOV_straylight_angles: 45.0 25.0 [degrees]
   FOV_straylight_duration: 30.0 [seconds]
#
Nr_of_areas: 1
#
   Area: MIDAS_BOX "MIDAS located on -X panel"
   Area_orientation: -1.0 0.0 0.0
   Area_lighting_angle: 60.0 [degs]
   Area_lighting_duration: 300.0 [seconds]
   #
#
Nr_of_modules: 6
#
   Module: RS "Robotic System"
   Module_level: LEVEL1
   Nr_of_module_states: 2
      Module_state: FOLD "Robot arm folded"
      MS_constraints: NOT_SCANNING
      Module_state: UNFOLD "Robot arm unfolded"
      MS_power: 0.3 0.7 0.2 [Watts]
#
   Module: AFM "AFM System"
   Module_level: LEVEL1
   Nr_of_module_states: 2
      Module_state: OFF "Off"
      Module_state: SCAN "Scan mode"
#
   Module: DPU "Data Processing Unit"
   Module_level: LEVEL1
   Sub_modules: MEMORY_UNIT CPU_UNIT IO_UNIT
   Nr_of_module_states: 2
      Module_state: OFF "Off"
      Module_state: P "Processing"
#
   Module: MEMORY_UNIT "Memory board"
   Module_level: LEVEL2
#
   Module: CPU_UNIT "CPU board"
   Module_level: LEVEL2
#
   Module: IO_UNIT "Interface board"
```

```
   Module_level: LEVEL2
#
Nr_of_modes: 2
#
   Mode: STANDBY "Standby mode"
   Mode_class: STANDBY
   Module_states: AFM OFF DPU OFF
   Nominal_power: 2.3
   Mode_transitions: TO_SCAN
#
   Mode: SCAN "Scanning mode"
   Mode_class: MEASUREMENT
   Module_states: AFM SCAN DPU OFF
   Nominal_power: 2.5 [Watts]
   Nominal_data_rate: 2500.0 [bits/sec]
   Equivalent_power: 3.2 7.4 2.8 [Watts]
   Equivalent_data_rate: 4000.0 [bits/sec] 5000.0 [bits/sec]
   Mode_actions: SCAN IMAGE MLT_IMG3
   Mode_constraints: OSIRIS_CLASH
#
Nr_of_parameters: 5
#
   Parameter: SET_AFM "Parametrised module state change"
   Raw_type: BOOL
   Eng_type: TEXT
   Default_value: OFF [ENG]
   Raw_limits: 0 1
   Nr_of_parameter_values: 2
      Parameter_value: 0 OFF
      Parameter_update: MSP OFF
      Parameter_value: 1 SCAN
      Parameter_update: MSP SCAN
#
   Parameter: WAITTIME "Wait duration resource parameter"
   Eng_type: REAL
   Default_value: 10.0 [secs]
   Unit: secs
   Eng_limits: 5.0 20.0 [secs]
   Resource: DURATION
#
   Parameter: IMG_TIME "Image duration resource parameter"
   Eng_type: REAL
   Default_value: 5.0 [seconds FIXED]
   Unit: secs
   Eng_limits: 1.0 10.0 [seconds]
   Resource: DURATION
#
   Parameter: SCAN_PWR "Power increase during scanning"
   Eng_type: REAL
   Default_value: 0.5 [Watts]
   Unit: W
   Resource: AVG_POWER_INCREASE
#
   Parameter: RUN_IMAGE "Run a specified image sequence"
   Raw_type: UINT
   Nr_of_parameter_values: 3
      Parameter_value: 1
      Parameter_run: MLT_IMG3 (IMG_TIME)
      Parameter_value: 2
```

```
      Parameter_run: MULTI_IMAGE_ATL (IMG_TIME)
      Parameter_value: 3
      Parameter_run: MULTI_IMAGE_RTL (IMG_TIME)
#
Nr_of_actions: 9
#
   Action: WAIT "Wait a while"
   Action_level: LEVEL3
   Action_parameters: WAITTIME
#
   Action: ARM_FOLD "Fold the robot arm"
   Action_level: LEVEL3
   Duration: 15.0 [seconds]
   Power_increase: 0.3 [Watts] 0.5 [Watts] 0.2 [Watts]
   Update_when_ready: MS RS FOLD
   Action_constraints: CHECK_ARM_UNFOLD
#
   Action: ARM_UNF "Unfold the robot arm"
   Action_level: LEVEL3
   Action_parameters: SET_AFM = SCAN [ENG]
   Duration: 15.0 [seconds]
   Power_increase: 0.3 0.5 0.2 [Watts]
   Update_when_ready: MS RS UNFOLD MSP AFM SET_AFM
   Action_constraints: CHECK_ARM_FOLD
#
   Action: SCAN "Scan an image"
   Action_level: LEVEL3
   Action_parameters: IMG_TIME SCAN_PWR
   Data_rate_increase: 1000.0 [bits/sec]
#
   Action: IMAGE "Create an image"
   Action_level: LEVEL2
   MTL_command: YES
   Action_parameters: IMG_TIME
   Action_constraints: SCAN_MODE
   Run_actions: SCAN (IMG_TIME)
#
   Action: MLT_IMG3 "Create 3 images"
   Action_level: LEVEL1
   Action_parameters: IMG_TIME WAITTIME
   Run_type: SEQUENTIAL
   Run_actions: IMAGE (IMG_TIME) \
                WAIT (WAITTIME = 15.0 [secs]) \
                IMAGE (IMG_TIME) \
                WAIT (WAITTIME = 15.0 [secs]) \
                IMAGE (IMG_TIME)
#
   Action: MULTI_IMAGE_ATL "Image sequence using absolute timeline"
   Action_parameters: IMG_TIME
   Run_type: ABSOLUTE
   Run_actions: 00:00:00 IMAGE (IMG_TIME) \
                00:00:20 IMAGE (IMG_TIME) \
                00:00:40 IMAGE (IMG_TIME)
#
   Action: MULTI_IMAGE_RTL "Image sequence using relative timeline"
   Action_parameters: IMG_TIME
   Run_type: RELATIVE
   Run_actions:          IMAGE (IMG_TIME) \
                00:00:15 IMAGE (IMG_TIME) \
```

```
                00:00:15 IMAGE (IMG_TIME)
#
   Action: TO_SCAN "Switch to scan mode"
   Action_level: LEVEL2
   Update_when_ready: MODE SCAN
#
Nr_of_constraints: 6
#
   Constraint: SUN_BLINDING "Report an error if Sun in FOV"
   Constraint_type: GEOMETRIC
   Severity: FATAL
   Condition: EVENT IS SUN_IN_FOV CAMERA_FOV
#
   Constraint: SCAN_MODE "Check if we are in scan mode"
   Constraint_type: TIME
   Severity: ERROR
   Condition: MODE NOT SCAN
#
   Constraint: NOT_SCANNING "Ensure that we are not scanning"
   Constraint_type: TIME
   Severity: WARNING
   Condition: MODE IS SCAN
#
   Constraint: CHECK_ARM_FOLD "Check if the robot arm is folded"
   Constraint_type: TIME
   Severity: ERROR
   Condition: MS NOT RS FOLD
#
   Constraint: CHECK_ARM_UNFOLD "Check if the robot arm is unfolded"
   Constraint_type: TIME
   Severity: ERROR
   Condition: MS NOT RS UNFOLD
#
   Constraint: OSIRIS_CLASH "Report an error if OSIRIS in imaging mode"
   Constraint_type: TIME
   Severity: ERROR
   Condition: MODE IS IMAGING
   Condition_experiment: OSIRIS
#
Experiment: OSIRIS "The OSIRIS experiment"
Mode: IMAGING "Imaging mode"
#
Experiment: SSMM "Solid state mass memory"
Local_memory: 16 [MB]
#
```

8. Appendix C: Format definition of the Input Timeline (ITL) file - template, notes, and example

```
#
# Filename: input.template
#
# Authors:  Peter van der Plas
# Date:     10 January 2000
#
# (c) ESA/Estec
#
# $Id: input.template,v 1.5 2001/07/11 13:00:21 rosetta Exp $
#
# $Log: input.template,v $
# Revision 1.5  2001/07/11 13:00:21  rosetta
# Updated to be compatible with the CRID issue B1.
# New implementation of parameters, also on events.
# New timing functionality and include file layout.
# PTR information can now be embedded in the ITL.
#
# Revision 1.4  2000/10/31 17:20:35  rosetta
# Removed all simulator state references.
# Added the Start_timeline and Stop_timeline keywords.
# Added the Event_time keyword.
# Updated the use of arguments.
# Direct mode updates are allowed now.
# Include files can now be scheduled on events.
#
# Revision 1.3  2000/03/13 19:33:25  rosetta
# Changed implementation of modules
# Added module power an data rate
# Added module level and sub-modules
# Changed action type to action level
# Various minor changes and document updates
#
# Revision 1.2  2000/02/21 16:51:54  rosetta
# Added use of include files and delta time values.
# More detailed description of arguments, labels and values.
#
# Revision 1.1  2000/01/10 12:30:17  rosetta
# Initial revision
#
#
--- template definition...
#
# <Header starts here...>
#
Version: <file version>
#
Ref_date: <date>
#
Start_time: <time>
End_time: <time>
#
# <Experiment modes can be initialised here...>
#
Init_mode: <experiment label> <mode label>
```

```
...   (i experiment mode initialisations)
#
# <Module states can be initialised here...>
#
Init_MS: <experiment label> <module label> <MS label>
...   (j module state initialisations)
#
# <Global include files in header only...>
#
Include_file: [<experiment>] <filename> [<delta time>]
...   (k global include files)
#
# <Timeline starts here...>
#
# <Available request timing formats...>
#
<execution time>                                    <request specification>
<execution time> \
   <earliest start>   <latest start>                <request specification>
<event label [(parameter list)]>                    <request specification>
<event label [(parameter list)]>   <delta time>     <request specification>
<event label [(parameter list)]>   <delta time> \
   <earliest delta> <latest delta>                  <request specification>
#
# <Event parameters...>
#
<event label [( \
   TIME_REF = <UTC|UTC+PD|UTC-PD> \
   EXP = <experiment label> \
   ITEM = <item label> \
   COUNT = <event count> \
   EARLIEST = <earliest event start time> \
   LATEST = <latest event start time> \
   MIN_NR = <minimum number of events> \
   MAX_NR = <maximum number of events>)]>
#
# <Available request specification formats...>
#
<request timing>   <experiment>   <mode>
<request timing>   <experiment>   <mode>   <action [(parameter list)]>
<request timing>   <experiment>   *        <action [(parameter list)]>
<request timing>   <experiment>   PTR      <pointing [(parameter list)]>
<request timing>   RSOC           PTR      <pointing [(parameter list)]>
<request timing>   <experiment>   <pointing [(parameter list)]>
<request timing>   RSOC           <pointing [(parameter list)]>
#
# <Action parameters...>
#
<action label [( \
   REPEAT = <number of repeats> \
   SEPARATION = <separation time> [<unit> (seconds)] \
   <parameter #1> [= <value #1> [<unit>|RAW|ENG]] \
   <parameter #2> [= <value #2> [<unit>|RAW|ENG]] ... )]
#
# <Time specific include files...>
#
<request timing>   [<experiment>]   INCLUDE   <include file>
#
# <Informational messages allowed between timeline entries...>
```

```
#
Comment: <message string>
Operator_request: <message string>
Description: <message string>
#
... (m timeline entries)
--- end of template...
```

```
#
# Filename: input.notes
#
# Author: Peter van der Plas
# Date:   10 January 2000
#
# (c) ESA/Estec
#
# $Id: input.notes,v 1.5 2001/07/11 13:00:21 rosetta Exp $
#
# $Log: input.notes,v $
# Revision 1.5  2001/07/11 13:00:21  rosetta
# Updated to be compatible with the CRID issue B1.
# New implementation of parameters, also on events.
# New timing functionality and include file layout.
# PTR information can now be embedded in the ITL.
#
# Revision 1.4  2000/10/31 17:20:35  rosetta
# Removed all simulator state references.
# Added the Start_timeline and Stop_timeline keywords.
# Added the Event_time keyword.
# Updated the use of arguments.
# Direct mode updates are allowed now.
# Include files can now be scheduled on events.
#
# Revision 1.3  2000/03/13 19:33:25  rosetta
# Changed implementation of modules
# Added module power an data rate
# Added module level and sub-modules
# Changed action type to action level
# Various minor changes and document updates
#
# Revision 1.2  2000/02/21 16:51:54  rosetta
# Added use of include files and delta time values.
# More detailed description of arguments, labels and values.
#
# Revision 1.1  2000/01/10 12:30:17  rosetta
# Initial revision
#
#

A few guidelines on the input timeline files:

Basic syntax
============
- comment lines start with '#'
- all characters after '#' are ignored
- a '\' indicates a continuation line: it allows keyword values to
  continue on the next line
- a '\' should be the last character of a line, otherwise it is ignored
- white spaces/tabs are used to separate words
- a keyword is followed by a ':' and then by its keyword values
- no white spaces within keywords/labels, use '_' instead
- keyword/label recognition is not case sensitive
- '|' stands for an exclusive or
- values between [ ] brackets are optional
- an optional number of values is indicated by ...
- all keywords (lines in the file) are optional, except when resulting
  in an inconsistent definition
```

```
- a string starts and ends with a '"', upper/lowercase is maintained
  here, white spaces and other special characters are allowed

Use of labels
=============
- a <label> is the base type for identifiers, units and any values
- no white spaces are allowed
- a label shall consist of 'a..z', 'A..Z', '0..9' or any of '_+-.:/'
- the '_' and '-' characters are considered different
- the '_' character is considered as a letter, e.g. the identifier
  ABCD is not the same as the identifier AB_CD
- labels are not case sensitive, e.g. the identifier ABCD is the same
  as identifier aBcD
- labels used as identifier, e.g. as experiment label, mode label, etc,
  shall not contain any of '+-.:/'
- labels used as units shall not contain any of '+-.:', the '/'
  character is allowed here indeed
- a <keyword> is considered as an identifier followed by a ':'

Use of units
============
- each (engineering) value may have an optional unit specifier
- the use of a unit for a parameter value automatically implies
  that the value is given as an engineering value
- the unit label needs to be specified between [ ] brackets
- if no unit is specified then the default unit will be applied
- unit sets will define the various unit options for a single
  physical unit, e.g. Volts may be V, mV, volts, mvolts, etc
- the unit specified for a value needs to be compatible with the
  value's unit set, if a unit set is applicable for the value
- unit conversions to the default unit will be done automatically
- for keywords with multiple values, a unit may either be given
  for each value, or a unit may be given after the last value,
  setting the same unit also on all previous values

Use of values
=============
- <value> means a single value
- an unsigned integer value shall consist of '0..9' or '+'
- an unsigned integer may also be given in hexadecimal or octal format
- a hexadecimal value is in the format '0xhhhhhhhh', where h is one of
  '0-9', 'a-f' or 'A-F' and where the number of h ranges from 1 to 8
- an octal value is an unsigned integer value that starts with a '0'
  and only consists of any of the '0-7' characters
- a (signed) integer value shall consist of '0..9', '+' or '-'
- a floating point value shall consist of '0..9', '+', '-', '.', 'e' or 'E'
- a floating point value shall be in the IEEE format

Date values
===========
- a date value may be given in the POR or EPS absolute time format
- an EPS <date> value is in the format [dd-month-yyyy[_hh:mm:ss]]
- [dd] is the day number
- [dd] may consist of one or two digits, and may start with zero
- [month] is the full (spelled out) month or any abbreviation
  with a minimum of 3 characters
- [yyyy] is the full year number
- [_hh:mm:ss] is optional and is defined similarly as in the time format
- the '_' character is mandatory here if the time of the day is given
```

```
- the time of the day defaults to _00:00:00
- a POR absolute time value is in the format [yy-dddThh:mm:ss[.mmm]Z]
- [yy] is the year in the 21st century and must have two characters
- [ddd] is the day number within the year, counting from 1
- [hh:mm:ss] is defined similarly as in the time format
- [.mmm] is optional and specifies the number of milliseconds
- the EPS software will always ignore the [.mmm] value

Time values
===========
- a time value may be given in the POR or EPS relative time format
- an EPS <time> field is in the format [[sign][ddd_]hh:mm:ss]
- [sign] is an optional sign ('-' or '+')
- [ddd] is an optional day number
- [ddd] may consist of one, two or three digits, and may be zero
- [hh] is the number of hours (00..23)
- [mm] is the number of minutes (00..59)
- [ss] is the number of seconds (00..59)
- [hh], [mm], [ss] must be specified and must have two characters each
- a POR relative time value is in the format [[-][ddd.]hh:mm:ss[.mmm]]
- [ddd] is the optional number of days
- [hh], [mm], [ss] is defined similarly as above
- [.mmm] is optional and specifies the number of milliseconds
- the EPS software will always ignore the [.mmm] value

Parameter lists
===============
- an action may have an optional parameter list defined between ( )
- a parameter list consists of any number of parameters
- each parameter may have an optional value
- a '=' is used to separate a parameter from its value
- a value may have an optional unit, or an indication if the value
  is of RAW or ENG type
- the use of a unit automatically implies the ENG type
- value options such as units need to be given between [ ] brackets

Initialisation keywords
=======================
- the version is mandatory must be an integer from 0 to 99999
- the version may be specified only once
- the reference date has to be given if relative times are used
- the reference date may be specified only once
- the start time and end time may be specified for each file
- the end time shall not be before the start time
- the start time and end time are mandatory for the toplevel file
- the start time is mandatory for files with absolute time specifications
  that are included at specific times (see below)
- lower level start and end times need to be inside the toplevel
  time window, if not a error message will be reported
- the initial experiment mode is optional
- any number of initial experiment modes may be specified
- the experiment mode is referenced by [experiment] [mode]
- the initial module state is optional
- any number of initial module states may be specified
- the MS is referenced by [experiment] [module] [MS]
- initialisation keywords may not appear after the first timeline entry

Request timing
==============
```

- time entries may be defined either using a single execution time or an execution time and a time window with earliest and latest start time, this time window is given relative w.r.t. the execution time
- the earliest start time must be a negative time value, the latest start time must be a positive time value
- the execution time may be specified using an absolute date value or a relative time value w.r.t. to the given reference date
- within a single file the execution time entries must be consecutive
- the execution time must be within the timeline start and end times
- event entries may be defined either using a single event, an optional delta time and an optional delta time window
- the delta time window with earliest and latest delta time is only allowed when the delta time is given, and is defined relative w.r.t. this delta time
- the earliest delta time must be a negative time value, the latest delta time must be a positive time value
- event entries may be located anywhere in the timeline, although it is recommended to define the events before the time entries
- the location of the event entries does not have any impact on the event execution time
- an event may specify a time reference as described in the CRID, the event parameter TIME_REF defaults to UTC
- an event may be defined for a specific experiment and item, e.g. the event SUN_IN_FOV may be defined for EXP = OSIRIS and ITEM = WAC
- events with no specific time definition or event count are global events, these events are executed every time they are encountered
- events with the COUNT parameter defined are executed only once when found in the event file as defined by the ESOC FD team
- events with the EARLIEST and/or LATEST time parameters defined will be executed when encountered within the given time window
- the EARLIEST time defaults to the start time value, the LATEST time defaults to the end time value
- it is possible here to limit the number of event executions using the parameter MAX_NR, the parameter MIN_NR can be used to verify that the event was found at least the given number of times in the specified time window

Request specification
=====================
- for each action call the experiment must be defined
- for each action call the mode must be defined, if the action may be executed from any mode, a '*' may be specified for the mode
- if no action is given, then the request is a direct mode transition, in this case the mode must be defined explicitly
- an action may have an optional parameter list with any number of parameters, a parameter may have an optional value and for a value an optional unit and/or RAW|ENG qualifier may be specified
- two special parameters REPEAT and SEPARATION may be defined for an action to specify an action has to be executed repeatedly
- for a pointing request the experiment has to be defined, or RSOC has to be used for pointing request with no specific source
- the label PTR has to be placed in the mode column, if the pointing request is embedded in an ITL file
- in a PTR file the above label is optional and may be omitted
- a pointing request consists of a pointing label and a number of parameters, these are described in the PTR SSD document

Include files
=============

- two types of include files are allowed, global and time specific
- global include files are defined using the Include_file keyword in the ITL header, but shall be defined after the Ref_date, Start_time and End_time keywords
- an optional delta time may be given here
- both POR and ITL files may be included here
- both relative and absolute time values in ITL time entries and absolute time values in POR requests are incremented with the given delta time
- events are not modified except for time specific events, where the EARLIEST and LATEST time values will be incremented
- time specific include files are defined by the INCLUDE label in the experiment column after the request timing, followed by the filename
- relative time values in ITL time entries are incremented with the current time definition, if included from a time entry
- absolute time values in ITL or POR time entries are made relative first using the file start time before being incremented with the current time definition
- files with ITL absolute time values must have the start time defined in the included file
- events are not modified except for time specific events, where the EARLIEST and LATEST time values will be incremented
- modified EARLIEST and LATEST time values must be within the given timeline start and end times
- events may not be included from an event based request timing
- all time entries must be located within the given timeline start and end times, all referenced events must follow the same rule, finally all execution times of entries based on events must do the same
- an optional experiment may be specified on an include file, for all included actions it will be checked if the action is defined for the experiment as given by the include statement
- both POR and ITL files may include PTR files
- PTR files may not include POR or ITL files

File types
==========
- ITL files should have the *.itl extension or shall start with ITL_
- POR files should have the *.por extension or shall start with POR_
- PTR files should have the *.ptr extension or shall start with PTR_
- if no file type is recognised, the file type shall default to ITL

Information message
===================
- persistent info messages can be defined using the keywords Comment, Operator_request and Description
- the info messages will be attached to the timeline entry that comes next after the given messages, the order will be maintained during sorting of the timeline entries
- no info messages are allowed in the ITL file header
- no info messages are allowed after the last timeline entry

POR specifics
=============
- the POR format is defined in the CRID (issue B1)
- files may be included from a POR file using the CE INCLUDE [<experiment>] <filename> [<delta time>] statement, note that the CE INCLUDE label must be positioned at the beginning of a line and then the experiment, filename (embedded between double quotes) and delta time (given as a POR relative time value), note that experiment and delta time are optional and behave similar to the definitions made for the ITL format

- persistent info messages can be defined using the CE <information type:> <message string> statement, note that the CE label must be positioned at the beginning of a line, followed by the info type and then the comment text (embedded between double quotes), using the same definitions as made for the ITL messages

```
#
# Filename: input.example
#
# Author: Peter van der Plas
# Date:   10 January 2000
#
# (c) ESA/Estec
#
# Description: This is an example input timeline file.
#
# $Id: input.example,v 1.5 2001/07/11 13:00:21 rosetta Exp $
#
# $Log: input.example,v $
# Revision 1.5  2001/07/11 13:00:21  rosetta
# Updated to be compatible with the CRID issue B1.
# New implementation of parameters, also on events.
# New timing functionality and include file layout.
# PTR information can now be embedded in the ITL.
#
# Revision 1.4  2000/10/31 17:20:35  rosetta
# Removed all simulator state references.
# Added the Start_timeline and Stop_timeline keywords.
# Added the Event_time keyword.
# Updated the use of arguments.
# Direct mode updates are allowed now.
# Include files can now be scheduled on events.
#
# Revision 1.3  2000/03/13 19:33:25  rosetta
# Changed implementation of modules
# Added module power an data rate
# Added module level and sub-modules
# Changed action type to action level
# Various minor changes and document updates
#
# Revision 1.2  2000/02/21 16:51:54  rosetta
# Added use of include files and delta time values.
# More detailed description of arguments, labels and values.
#
# Revision 1.1  2000/01/10 12:30:17  rosetta
# Initial revision
#
#
Version: 00123
#
Ref_date: 21-February-2011
#
Start_time: 11:30:00
End_time: 14:00:00
#
Init_mode: MIDAS STANDBY
Init_mode: OSIRIS INIT
#
Init_MS: MIDAS FILTER MODE_C
Init_MS: OSIRIS AFM ON
#
Include_file: "generic.por"
Include_file: MIDAS "midas.por" +02:30:00
#
SUN_IN_FOV (EXP = OSIRIS ITEM = WAC) \
```

```
                 OSIRIS    *             TO_SAFE_MODE
000_12:00:00     MIDAS     STANDBY       STANBY_TO_MEASUREMENT
000_12:00:00     OSIRIS    INIT          INIT_TO_STANDBY
000_12:00:00     OSIRIS    PTR           NADIR_START
#
Comment: "Switch OSIRIS to measurement mode"
#
000_12:05:00     OSIRIS    STANDBY       STANBY_TO_MEASUREMENT
000_12:10:00     MIDAS     MEASUREMENT   MAKE_IMAGE (
                                            IMAGE_TIME = 30.0 [seconds])
000_12:15:00     INCLUDE   "generic_sequence.itl"
000_12:35:50 -01:00:00 +01:30:00 \
                 MIDAS     MEASUREMENT   MAKE_IMAGE ( \
                                            IMAGE_TIME = 60.0 [seconds])
000_12:45:20     MIDAS     MEASUREMENT   MAKE_IMAGE ( \
                                            IMAGE_TIME = 60.0 [seconds] \
                                            REPEAT = 3 \
                                            SEPARATION = 120.0 [seconds])
ALT_BELOW_500M (COUNT = 712) \
                 OSIRIS    MEASUREMENT   START_TESTS
000_13:02:45     MIDAS     MEASUREMENT   MEASUREMENT_TO_STANDBY
000_13:08:00     MIDAS     STANDBY       MOVE_ARM ( \
                                            ROLL = 15.2 [degrees] \
                                            YAW = 8.8 [degrees])
#
Description: "Include an image sequence at perigee"
Description: "A time window is specified for this operation"
#
AT_PERIGEE (EARLIEST = 000_12:45:00 LATEST = 000_13:14:00) \
   -00:01:00 -00:02:00 +00:02:00 \
                 OSIRIS    INCLUDE       "image_sequence.itl"
000_13:13:00     MIDAS     STANDBY       MOVE_ARM (
                                            ROLL = 12.0 [degrees] \
                                            YAW = 33.0 [degrees])
#
# This comment will get lost after processing
#
ALT_ABOVE_500M (COUNT = 713) \
                 OSIRIS    MEASUREMENT   STOP_TESTS
000_13:14:00     OSIRIS    MEASUREMENT   MEASUREMENT_TO_STANDBY
000_13:14:00     OSIRIS    PTR           NADIR_END
#
```

9. Appendix D - Format definitions of the keywords and their contents for the acknowledge files (ACKN) in XML format

```
<?xml version="1.0"?>
<experimentDefinitionResources>
  <experiment>
    <experimentName>   ALICE             </experimentName>
    <mnemonic>         ALICE             </mnemonic>
    <PIName>           Alan Stern        </PIName>
    <PITitle>          Dr.               </PITitle>
    <PIInstitute>                        </PIInstitute>
    <PIEmail>                            </PIEmail>
    <PIftp>
      <ftpServer>                        </ftpServer>
      <ftpUserid>                        </ftpUserid>
      <ftpPassword>                      </ftpPassword>
      <ftpUploadDir>                     </ftpUploadDir>
    </PIftp>
  </experiment>

 <experiment>
    <experimentName>   CONSERT           </experimentName>
    <mnemonic>         CONSERT           </mnemonic>
    <PIName>           Wlodak Kofman     </PIName>
    <PITitle>          Dr.               </PITitle>
    <PIInstitute>                        </PIInstitute>
    <PIEmail>                            </PIEmail>
    <PIftp>
      <ftpServer>                        </ftpServer>
      <ftpUserid>                        </ftpUserid>
      <ftpPassword>                      </ftpPassword>
      <ftpUploadDir>                     </ftpUploadDir>
    </PIftp>
  </experiment>

 <experiment>
    <experimentName>    COSIMA           </experimentName>
    <mnemonic>          COSIMA           </mnemonic>
    <PIName>            Joachim Kissel   </PIName>
    <PITitle>           Dr.              </PITitle>
    <PIInstitute>                        </PIInstitute>
    <PIEmail>                            </PIEmail>
    <PIftp>
      <ftpServer>                        </ftpServer>
      <ftpUserid>                        </ftpUserid>
      <ftpPassword>                      </ftpPassword>
      <ftpUploadDir>                     </ftpUploadDir>
    </PIftp>
  </experiment>

<experiment>
    <experimentName>    GIADA            </experimentName>
    <mnemonic>          GIADA            </mnemonic>
    <PIName>            Luigi Colangeli  </PIName>
    <PITitle>           Dr.              </PITitle>
    <PIInstitute>                        </PIInstitute>
    <PIEmail>                            </PIEmail>
    <PIftp>
      <ftpServer>                        </ftpServer>
      <ftpUserid>                        </ftpUserid>
      <ftpPassword>                      </ftpPassword>
      <ftpUploadDir>                     </ftpUploadDir>
    </PIftp>
  </experiment>

 <experiment>
    <experimentName>   MIDAS             </experimentName>
    <mnemonic>         MIDAS             </mnemonic>
    <PIName>           Willi Riedler     </PIName>
    <PITitle>          Dr.               </PITitle>
```

```
   <PIInstitute>                        </PIInstitute>
   <PIEmail>                            </PIEmail>
   <PIftp>
     <ftpServer>                        </ftpServer>
     <ftpUserid>                        </ftpUserid>
     <ftpPassword>                      </ftpPassword>
     <ftpUploadDir>                     </ftpUploadDir>
   </PIftp>
 </experiment>

<experiment>
   <experimentName>   MIRO             </experimentName>
   <mnemonic>         MIRO             </mnemonic>
   <PIName>           Samuel Gulkis    </PIName>
   <PITitle>          Dr.              </PITitle>
   <PIInstitute>                       </PIInstitute>
   <PIEmail>                           </PIEmail>
   <PIftp>
     <ftpServer>                       </ftpServer>
     <ftpUserid>                       </ftpUserid>
     <ftpPassword>                     </ftpPassword>
     <ftpUploadDir>                    </ftpUploadDir>
   </PIftp>
 </experiment>

<experiment>
   <experimentName>   OSIRIS           </experimentName>
   <mnemonic>         OSIRIS           </mnemonic>
   <PIName>           Uwe Keller       </PIName>
   <PITitle>          Dr.              </PITitle>
   <PIInstitute>                       </PIInstitute>
   <PIEmail>                           </PIEmail>
   <PIftp>
     <ftpServer>                       </ftpServer>
     <ftpUserid>                       </ftpUserid>
     <ftpPassword>                     </ftpPassword>
     <ftpUploadDir>                    </ftpUploadDir>
   </PIftp>
 </experiment>

<experiment>
   <experimentName>   ROSINA           </experimentName>
   <mnemonic>         ROSINA           </mnemonic>
   <PIName>           Hans Balsiger    </PIName>
   <PITitle>          Dr.              </PITitle>
   <PIInstitute>                       </PIInstitute>
   <PIEmail>                           </PIEmail>
   <PIftp>
     <ftpServer>                       </ftpServer>
     <ftpUserid>                       </ftpUserid>
     <ftpPassword>                     </ftpPassword>
     <ftpUploadDir>                    </ftpUploadDir>
   </PIftp>
 </experiment>

<experiment>
   <experimentName>   RPC            </experimentName>
   <mnemonic>         RPC            </mnemonic>
   <PIName>           Karl-Heinz Glassmeier</PIName>
   <PITitle>          Dr.            </PITitle>
   <PIInstitute>                     </PIInstitute>
   <PIEmail>                         </PIEmail>
   <PIftp>
     <ftpServer>                     </ftpServer>
     <ftpUserid>                     </ftpUserid>
     <ftpPassword>                   </ftpPassword>
     <ftpUploadDir>                  </ftpUploadDir>
   </PIftp>
 </experiment>

<experiment>
   <experimentName>    RSI          </experimentName>
   <mnemonic>          RSI          </mnemonic>
   <PIName>            Martin Paetzold</PIName>
```

```
    <PITitle>           Dr.           </PITitle>
    <PIInstitute>                     </PIInstitute>
    <PIEmail>                         </PIEmail>
    <PIftp>
      <ftpServer>                     </ftpServer>
      <ftpUserid>                     </ftpUserid>
      <ftpPassword>                   </ftpPassword>
      <ftpUploadDir>                  </ftpUploadDir>
    </PIftp>
  </experiment>



 <experiment>
    <experimentName>   VIRTIS         </experimentName>
    <mnemonic>         VIRTIS         </mnemonic>
    <PIName>           Angioletta Coradini      </PIName>
    <PITitle>          Dr.            </PITitle>
    <PIInstitute>                     </PIInstitute>
    <PIEmail>                         </PIEmail>
    <PIftp>
      <ftpServer>                     </ftpServer>
      <ftpUserid>                     </ftpUserid>
      <ftpPassword>                   </ftpPassword>
      <ftpUploadDir>                  </ftpUploadDir>
    </PIftp>
  </experiment>

 <experiment>
    <experimentName> ROSETTA_LANDER   </experimentName>
    <mnemonic>       SSP              </mnemonic>
    <PIName>         Helmut Rosenbauer</PIName>
    <PITitle>        Dr.              </PITitle>
    <PIInstitute>                     </PIInstitute>
    <PIEmail>                         </PIEmail>
    <PIftp>
      <ftpServer>                     </ftpServer>
      <ftpUserid>                     </ftpUserid>
      <ftpPassword>                   </ftpPassword>
      <ftpUploadDir>                  </ftpUploadDir>
    </PIftp>
  </experiment>

</experimentDefinitionResources>

<?xml version="1.0"?>

<errorCodes>
  <errorRaisedBy>
      <componentName>    Communication and Line Problems   </componentName>
      <errors>
        <errorNumber>    001                               </errorNumber>
        <errorString>    upload not completed              </errorString>
        <errorSeverity>  fatal                             </errorSeverity>
      </errors>
      <errors>
        <errorNumber>    002                               </errorNumber>
        <errorString>    not all include files exist       </errorString>
        <errorSeverity>  fatal                             </errorSeverity>
      </errors>

  </errorRaisedBy>

 <errorRaisedBy>
      <componentName>    Incoming Userid, password, file and directory checks
</componentName>
      <errors>
        <errorNumber>    064                               </errorNumber>
        <errorString>    userid does not match instrument team </errorString>
        <errorSeverity>  fatal                             </errorSeverity>
      </errors>
      <errors>
        <errorNumber>    065                               </errorNumber>
        <errorString>    file size is zero                 </errorString>
```

```
        <errorSeverity>  fatal                              </errorSeverity>
      </errors>
  </errorRaisedBy>

<errorRaisedBy>
      <componentName>    Outgoing userid, password, file and directory checks
</componentName>
      <errors>
        <errorNumber>    128                                </errorNumber>
        <errorString>    experiment team ftp server not reachable
</errorString>
        <errorSeverity>  warning                            </errorSeverity>
      </errors>
      <errors>
        <errorNumber>    129                                </errorNumber>
        <errorString>    could not login to experiment team ftp
server</errorString>
        <errorSeverity>  error                              </errorSeverity>
      </errors>
      <errors>
        <errorNumber>    130                                </errorNumber>
        <errorString>    upload failure during result transmitting to
experiment team ftp server</errorString>
        <errorSeverity>  error                              </errorSeverity>
      </errors>
  </errorRaisedBy>

 <errorRaisedBy>
      <componentName>   File Name Syntax Checking      </componentName>
      <errors>
        <errorNumber>   256                              </errorNumber>
        <errorString>   incorrect submission number, number too big
</errorString>
        <errorSeverity> fatal                            </errorSeverity>
      </errors>
      <errors>
        <errorNumber>   257                              </errorNumber>
        <errorString>   incorrect submission number, number too small
</errorString>
        <errorSeverity> fatal                            </errorSeverity>
      </errors>

  </errorRaisedBy>

<errorRaisedBy>
      <componentName>   EPSTEST Errors raised  </componentName>
      <errors>
        <errorNumber>    284                                </errorNumber>
        <errorString>                                       </errorString>
        <errorSeverity> fatal                               </errorSeverity>
      </errors>

  </errorRaisedBy>

</errorCodes>
```