esa DOCUMENT

document title/ titre du document

HPMCS Science Ground Segment Real-Time Telemetry ICD

prepared by/*préparé par*	HPMCS Development Team
reference/*réference*	ESA ref: PT-CMOC-MDS-ICD-3101-OPS-GDS Planck SGS ref: PGS-ICD-016 Herschel SGS ref: HGS-ICD-004 TERMA ref: TERMA/SPD/63/HPMCS/TS/ICD/RTSI
issue/*édition*	1
revision/*révision*	5
date of issue/*date d'édition*	2008-06-24
status/*état*	Final
Document type/*type de document*	Interface Control Document
Distribution/*distribution*	

European Space Agency
Agence spatiale européenne

ESA	HP MOC Team:	
	G.D. Girolamo	HPMCS TO
	M. Schmidt	H/P SOM
	J. Dodsworth	H/P GSM
	Herschel SGS Team:	
	J. Brumfitt	
	K. Galloway	HSC
	T. Prusti	
	J.Riedinger	
	L. O'Rourke[RID-PDR2-62]	Science Operations Engineer
	Planck SGS Team:	
	J. Tauber	Planck Project Scientist
	D.Texier	
	J.P.Puget	HFI
	L.Vibert	HFI
	A. Zacchei	LFI
	R.C. Butler	LFI
	L. Aloy	H/P Project
Terma	M. Enne	
VCS	P. Demeo	
Siemens	G. Krames	

APPROVAL

Title *titre*	HPMCS Science Ground Segment Real-Time Telemetry ICD	issue *issue* 1	revision *revision* 4
author *auteur*	HPMCS Development Team	date *date*	2007-11-30
approved by *approuvé by*	G.Di Girolamo M. Schmidt C.J.Watson J. Dodsworth D.Texier L.Vibert A.Zacchei J.Riedlinger	date *date*	

Digitally signed by Damien Texier
Date: 2008.12.11 15:53:35 +01'00'

Laurent Vibert
2008.12.23 10:26:49 +01'00'

Digitally signed by Andrea Zacchei
Date: 2008.12.23 11:04:44 +01'00'

CHANGE LOG

reason for change */raison du changement*	issue/*issue*	revision/*revision*	date/*date*
Henrik Pedersen (Terma)	1	0 Draft A	2005-04-15
Gerfried Krames (Siemens)	1	0 Draft B	2005-09-02
Gerfried Krames (Siemens)	1	0	2005-10-27
Gerfried Krames (Siemens)	1	1	2006-06-08
Gerfried Krames (Siemens)	1	2	2007-04-03

reason for change */raison du changement*	issue/*issue*	revision/*revision*	date/*date*
Yi Tan (Terma)	1	3	2007-11-30
G.Di Girolamo (ESA/ESOC)	1	4	2008-01-24
G.Di Girolamo (ESA/ESOC)	1	5	2008-06-24

CHANGE RECORD

Issue: 1 Revision: 0 DRAFT B

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
Update of initial draft for HPMCS D2.	see change bars	

Issue: 1 Revision: 0

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
Update of 1.0 Draft B after HPMCS D2 PDR:	see change bars	
RID-PDR2-79	18	6.2.4
RID-PDR2-80	18	6.2.4
RID-PDR2-82	8	1.6 and 1.7
RID-PDR2-209	2, 3	Distribution and approval lists
RID-PDR2-64	7,8	1.6 and 1.7
RID-PDR2-62	2, 3	Distribution and approval lists

Issue: 1 Revision: 1

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
Request to provide a "template" client for the RTSI.	see change bars	New section 8.

Issue: 1 Revision: 2

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
DCR 979	see change bars	Inserted 7.1.1
DCR 980	see change bars	Inserted 7.1.2

Issue: 1 Revision: 3

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
DCR 1161	see change bars	Inserted 7.1.1

Issue: 1 Revision: 4

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
DCR 900	Cover pages	

Issue: 1 Revision: 5

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/ *paragraph(s)*
DCR 1354 (MANTIS CR # 93)	1.6	9

Table Of Contents

1 INTRODUCTION

1.1 Purpose

This document is intended to describe the science ground segment interface for real-time telemetry data passed from the HP MOC to the HP PI client workstations located at the MOC, i.e. IW@MOC for Planck instruments and ICC@MOC for Herschel instruments.

1.2 Scope

The scope of the document covers all real-time telemetry data at the level of telemetry source packets. These telemetry source packets are passed from the HP MCS TM Packetiser to HPSDS. With the term real-time telemetry it is meant all real-time telemetry routed to the HERSCHEL-PLANCK MOC as downlinked from the Spacecraft and acquired during the normal passes (VC-0, VC-4, VC-1).

In particular the document describes the interface between the PI client workstations (IW@MOC or ICC@MOC) and the HP MCS Real-Time Science Interface (RTSI) System, which will be located on the HPSDS outside the ESOC firewall, receiving telemetry source packets from the HP MCS TM Packetiser and forwarding these to the PI client workstations.
The possibility to play-back telemetry data is not foreseen within the HP MCS RTSI system.

1.3 Summary

This document describes the interface specification applicable to the transfer of real-time telemetry source packet data between the HP MCS RTSI and the PI client workstations (IW@MOC or ICC@MOC).

1.4 Amendment History

See Document Change record at start of document.

1.5 Change Forecast

This document is the output from the HPMCS delivery 2 AD phase after PDR updates and no further updates are planned for the time being.

1.6 Applicable Documents[RID-PDR2-64,82]

A-1. Deleted

A-2. HPMCS Glossary, Definitions and Acronyms, Ref: TERMA/SPD/63/HPMCS/TS/GLOS, Issue 2.0, 31 October, TERMA

A-3. SCOS-2000 TM Packet Data Provision Services Interface Control Document, Ref: EGOS-MCS-S2K-ICD-0022-TOS-GIC, Issue 1.6, 2007-11-26, ESA DCR-1354

A-4. CORBA based External I/F Services for SCOS-2000 Data Provision Interface Control Document, Ref: S2K-MCS-ICD-0020-TOS-GIC, Issue 1.6, 02 March 2004, ESA

A-5. HERSCHEL-PLANCK Packet Structure ICD, Ref: SCI-PT-ICD-07527, Issue 5.0, 20 July 2004, ESA

A-6. HERSCHEL-PLANCK OGS-SGS ICD Chapter 4.1: Real-Time Telemetry Data (MOC-HPSDS) - May 26th 1998 - INT–MOC-SYS-ICD-0001-OGI

A-7. HERSCHEL-PLANCK MCS SRS, Ref: PT-CMOC-MCS-SRS-3201-OPS-GDS, Issue 2.3, April 2005, ESA

A-8. HPMCS SAD System Level Architecture, Ref: TERMA/SPD/63/HPMCS/DDF/SAD/SYS, Issue 2.0, 31 October 2005, TERMA

1.7 Reference Documents ***RID-PDR2-64,82***

R-1. Deleted

1.8 Abbreviations

All abbreviations and acronyms used in this document are explained in Applicable Document A-2.

2 OPERATIONAL ASSUMPTIONS AND CONSTRAINTS

2.1 Assumptions

The following assumptions apply to this interface:

1. Real-time data transfer between the HP MCS RTSI and the PI client workstations will be achieved by point-to-point communications using CORBA as middleware.

2. HERSCHEL-PLANCK MOC is separated by a Firewall from the external world.

3. HP MCS RTSI is made up of a single node (workstation) and located on the HPSDS outside the MOC firewall in ESOC.

4. No data time and quality checks will be performed by HP MCS RTSI neither at telemetry transfer frame level nor at telemetry source packet.

5. No time correlation information will be provided by HP MCS RTSI.

6. There will be no data replay capability; i.e. if for any reason delivery of on-line telemetry will fail, no capability to recall the missed telemetry will be supported by the HP MCS RTSI system.

7. No data archiving functionality will be implemented by HP MCS RTSI at any level.

2.2 Constraints

The HP MCS TM Packetiser link set-up to HPMCS RTSI system located on the HPSDS outside the firewall is instigated by the HP MCS TM Packetiser, which acts as a client and HP MCS RTSI as a server.

The link set-up between HP MCS RTSI and the PI workstation applications are instigated by the PI workstation applications, which act as clients and HP MCS RTSI as a server.

The real-time transfer between the HP MCS RTSI and PI workstations is constrained by the bandwidth of the link available and the LAN, currently the link is foreseen to be TBD kbps, the LAN handles 100 Mbps.
With a CORBA-based connection, the number of packet transfers per time unit is an important variable that determines the maximum throughput. According to the performance requirements

outlined in [A-7], the consolidated maximum is 87 packets / second, which results in the same number of CORBA requests per second for each client.
Obviously the transfer rate to the PI workstations depends on the number of connected clients and the configuration, i.e. the exact packet ID's requested for real-time provision.

3 REQUIREMENTS

3.1 Functional Requirements

As specified in reference document [A-7].

3.2 On-line Data Delivery Requirements

As specified in reference document [A-7].

3.3 Off-line Data Delivery Requirements

Not applicable to this document.

4 INTERFACE CHARACTERISTICS

4.1 Interface Location and Medium

This interface deals with real-time telemetry source packet data transfer between the HP MCS RTSI and PI workstation applications (IW@MOC for Planck and ICC@MOC for Herschel).
The HP MCS RTSI and the PI workstations are based at ESOC in Darmstadt, Germany outside the OPSLAN (operational network). The HPMCS RTSI system will be running on the HPSDS (HP Security Data Server).
Data transfer will be electronic, using point-to-point communications based on the TCP/IP protocol via CORBA.
The following figure provides a representation of the HP MCS science ground segment real-time telemetry distribution system architecture.
Grayed-out area includes systems not part of the HP MCS RTSI - PI workstation interface but is reported for completeness.

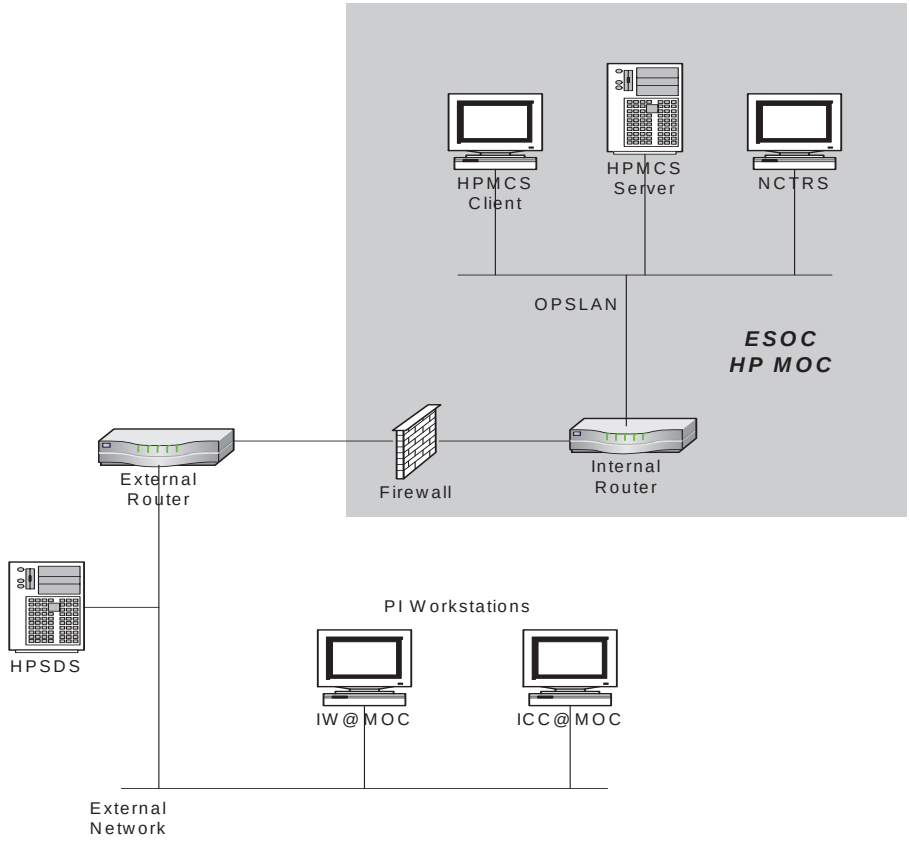

Figure 1 – HP MCS RTSI - PI workstation Interfaces

The figure reports a representation of the physical location of the systems involved in the science ground segment real-time telemetry distribution infrastructure. In particular:

- Network Control TM/TC and Routing System (NCTRS) and HP MCS systems are connected on the ESOC OPSLAN.
- A firewall separates the HP MOC (OPSLAN network) from the ESOC external network.
- HPSDS is connected on the ESOC ExtServerLAN.
- The SGS – PI workstation applications receiving telemetry source packets from HP MCS RTSI is here represented as a system connected on a local LAN separated by the external network (WAN) via a firewall.

4.2 Hardware Characteristics and Limitations

The hardware/operating system for the HPSDS workstation is expected to be Linux Pentium IV PC (CPU >= 2GHz, RAM => 1GB) or an equivalently powered SUN workstation, but the provided interface will be independent from the hardware/operating system of the PI client workstations. The only significant hardware limitation imposed on the interface is the available bandwidth (see section 2.2).

4.3 Data Source, Destination and Transfer Mechanism

Data source is the HP MCS RTSI system running on the HPSDS and data destinations are the PI client workstations. Transfer mechanism is CORBA via TCP/IP. Up to 10 PI client workstation connections are supported.

4.4 Node and Device Addressing

The initial task of any CORBA-based client software is to find the CORBA Naming Service by host name and port number. For RTSI, the CORBA Naming Service will be running on the RTSI workstation (HPSDS). The port number of the CORBA Naming Service will be configurable. Since the CORBA Naming Service is the entry point, locating the SCOS-2000 External Interface TMP server object is straight-forward using standard CORBA operations. Further information can be found in [A-3] and [A-4].

4.5 Relationships with other Interfaces

The real-time telemetry source packet interface provided by the HP MCS RTSI is the CORBA level interface provided by the SCOS-2000 TM Packet Data Provision Service, see [A-3] and [A-4].

5 ACCESS

5.1 Interface Utility Software

The IW@MOC and ICC@MOC PI applications shall use the CORBA interface definitions as defined in [A-3] and [A-4].

5.2 Failure Protection, Detection and Recovery Procedures

The HP MCS RTSI system will perform no check for completeness of the delivered real-time data, but forward this to connected PI applications as received from HP MCS TM Packetiser.
Failure detection or incomplete data detection may be achieved from the PI clients by checking the Source Sequence Counter. The SSC is incremented by 1 for each telemetry source packet sent from a specific application on-board the satellite. SCC = function{TM source packet, APID}.
Recovery of data that has failed real-time transfer is outside the scope of the RTSI.

5.3 File Naming Convention

Not applicable.

5.4 Storage and File Detection Requirements

Not applicable.

5.5 Security Requirements

Not applicable.

5.6 Data Integrity Checks

No further data integrity checks as mentioned in section 5.2 will be carried out by the tasks responsible for the transmission of real-time data between the HP MCS RTSI and the PI workstations.
Note: This does not preclude data integrity checks being carried out by PI workstation applications, e.g. for telemetry data, the PI applications can upon TM packet reception perform validation checks using TM packet Source Sequence Counter etc.

5.7 Backup Requirements

Not applicable.

5.8 Error Handling

5.8.1 TRANSPORT/NETWORK LAYER

The CORBA standard error handling capabilities are used for the RTSI. The basic pattern is that whenever a CORBA operation fails due to a transport problem, this will trigger an error reporting mechanism on the level of the client's implementation language. For C++ and Java, this means that an exception will be thrown.
Further details about error handling (and the language mapping in general) are to be found in the user manual of the CORBA library used by the PI software.

5.8.2 APPLICATION LAYER

HP MCS RTSI will monitor the status of the any connections, whether TCP/IP or CORBA. Whenever an anomalous disconnection is detected, RTSI will send an alarm message to the RTSI operator as an event to the event logger that is part of the SCOS instance deployed on the HP RTSI workstation (HPSDS).
In case of link failure it is up to PI clients to re-connect to HP MCS RTSI in order to re-establish the data transfer session.

6 DETAILED INTERFACE SPECIFICATIONS

The following sections provide detailed specifications for the network transport and application layers of the interface.

6.1 Transport Level

The transport level is entirely encapsulated by CORBA.

6.2 Application Level

6.2.1 DATA MESSAGES

Basically, data messages are defined in terms of CORBA IDL (Interface Definition Language). The SCOS-2000 External Interface / TM Packet provision service defines a data structure (TMpacketNotifyData) that contains the octet string of the source packet (m_pktBodyRawData) and some additional data which may be ignored by client software.

```
    struct TMpacketNotifyData
    {
        TMpacketAttributes m_pktAttributes;    // packet attributes
        IBASE::ByteString m_pktHeaderRawData;  // S2K packet header byte stream
        IBASE::ByteString m_pktBodyRawData;    // S2K packet body byte stream
        TMpacketParams m_pktParamNames;    // A list of the names of a VPDs
params DCR-1161
        TMpacketParams m_pktParams;            // VPD parameter
    };
```

6.2.2 GENERATION METHOD

Not applicable.

6.2.3 DATA PASSED ACROSS THE INTERFACE AND THEIR DIRECTION

This ICD deals solely with data passed between the HP MCS RTSI system on the HPSDS and the PI applications located on the PI workstations (IW@MOC and ICC@MOC):

- Connection establishment and filter request criteria are passed from the PI applications to the RTSI. Technically, this is done by invocation of the CORBA method registerTMpackets(). Details can be found [A-3].
- Real-time telemetry data are passed from the HP MCS RTSI to the PI workstations. These are CORBA callbacks to the PI client as defined in [A-3]. The actual TM packet data (see 6.2.1) are parameters of the callback.

6.2.4 SIZE AND FREQUENCY OF TRANSFERS

The size and frequency of transfer of PI application connection establishment and filter request data messages is considered negligible.

The size and frequency of transfer of TM data messages is shown in the table below. Details are extracted from [R-1].

Data Item	Size in Bytes	Frequency (Data Items/Sec)
VC0 TM data message	22 (TM source packet) + overhead	< 28
VC4 TM data message	300 (TM source packet) + overhead	< 10
VC1 TM data message	800 (TM source packet) + overhead	< 49

In above table, "overhead" is placeholder for [size of packet metadata] + [protocol overhead]. In order to be on the safe side, we make the pessimistic assumption that this can be up to 1000 bytes for each transfer.

The crucial factor is the maximum frequency of transfers (which is 87 TM data messages per second). Due to the protocol overhead, the difference between a transfer of a 22 bytes data unit and 800 bytes data unit is minor. Therefore, the maximum size (800 bytes + 1000 bytes = 1800 bytes) is assumed for the subsequent estimation.

For a single PI client registered for all TM packets (i.e. no filter), the above assumption implies a bandwidth consumption of 1800 * 87 * 8 = 1252800 bits per second, i.e. ~ 1.2 Mbit/sec.

In this worst-case assumption scenario, 10 client PI workstations (as required by [A-7]) would consume a 12 Mbit/sec bandwidth on the network interface of a single instance of HPSDS. (There will be 2 independent instances, one for Herschel and one for Planck. Therefore, only one mission needs to be considered in this calculation.)

Note: The frequency for TM data messages applies only to those times during which the HP MCS is receiving telemetry on VC-0, VC-4 and/or VC-1. The actual transfer frequency to a particular PI application depends on selected filter of APID’s and SPID's, i.e. the above figures can be considered as maximum figures in case all APID’s and SPID's are selected.
The TM source packet sizes are mean sizes.

6.2.5 TIMING AND SYNCHRONISATION REQUIREMENTS

Timing and synchronisation requirements are met by the use of both the TCP/IP and CORBA protocols.

7 DATA DEFINITION

7.1 Connection and Filter Data Messages

For details on Connection and Filter data messages used in the CORBA level interface see [A-3] and [A-4].

7.1.1 NOTE ON CONNECTION MANAGEMENT

(DCR-979)

The CORBA `CosNaming` name of the EXIF TM Packet Provision service is calculated as follows:

```
-----------------------------------------------
|Index | id | kind |
-----------------------------------------------
| 0    | <DOMAIN_ID> | |
-----------------------------------------------
| 1    | <S2K_LOCAL_SRV_FAMILY> | |
-----------------------------------------------
| 2    | "TMP_PRO_002" | |
-----------------------------------------------
```

Where <DOMAIN_ID> and <S2K_LOCAL_SRV_FAMILY> are strings with a value dependent on the HPMCS configuration.
The number in TMP_PRO_002 means that HPMCS provides version 002 of the SCOS EXIF IDL definition. However, the client should not use the service identification string hard-coded. Instead, it is recommended to use the symbolic constant **`ITMP::ServiceName`** defined in `ITMP_PRO.idl`.

7.1.2 NOTE ON FILTERING

(DCR-980)

The structure **`ITMP::TMpacketFilter`** supports filtering for a list (CORBA sequence) of SPIDs as well as APIDs (alternatively).

Compatibility:
SCOS-2000 versions earlier than 3.1 provide only filtering for a list APIDs. Therefore, if RTSI client software shall be tested connecting it to the TM Packet Provision Service of an earlier version of SCOS-2000, the client is restricted to using APID-based filtering.

Therefore, if you are using an old version of SCOS-2000, you have to replace the following code in the example RTSI client (RTSItmpAccess.C, method registerClient()):

```
packetFilter.m_apIds.length(0);
```

```
// Filter on SPIDs (= Filingkeys):
packetFilter.m_filingKeys.length(1);
packetFilter.m_filingKeys[0] = 100001000;
```

with

```
packetFilter.m_apIds.length(1);
// LFI scientific packets
packetFilter.m_apIds[0] = 0x604;
// Filter on SPIDs (= Filingkeys):
//packetFilter.m_filingKeys.length(1);
//packetFilter.m_filingKeys[0] = 100001000;
```

From S2k 4.2 the new fields in packet filter have to set the following value: DCR-1161

packetFilter.m_types.length(0);
packetFilter.m_subTypes.length(0);
packetFilter.m_variablePackets = false;

7.2 Telemetry Data Message

For details on Telemetry data messages used in the CORBA level interface see [A-3] and [A-4].

8 APPENDIX: RTSI TEMPLATE CLIENT

The RTSI template client is provided to demonstrate the interaction with the CORBA interface provided on the SDS.

8.1 Source Code

8.1.1 OVERVIEW

The RTSI template client (briefly "RTSI") is delivered as part of the HPMCS source code deliverables. The source code and "Makeinfo" file are located at:
`<cvs-root>/hpmcs/addon-exif-rep/src/RTSI/`

The RTSI client source comprises several files, of which major parts should be reusable for actual client development.

File name	Description
Makeinfo	Makefile for use in SCOS development environment only.
RTSIcorbaHelper.C /.H	Tool functions to manage the CORBA layer.
RTSImain.C	Main program
RTSIstreaming.C / .H	Support for C++ ostreams using the "<<" operator.
RTSIthread.C	Tool functions to manage the POSIX Thread layer.
RTSItimeAccess.C / .H	Client-side implementation of EXIF timing services.
RTSItmpAccess.C / .H	Client implementation for the EXIF TMP_PRO service.
RTSIview.C / .H	Generic EXIF client.

"Interesting" sections in the code are marked with "Your code here". These mainly comprise registration criteria and processing of the incoming packets.

8.1.2 ALGORITHM

The algorithm is basically the following one:

- Bring up CORBA layer and establish connection to the Naming Service
- Connect to EXIF TMP_PRO service (shared real-time server)
- Register at the server (currently, registration takes place for SPID=100001000 which is used for user-defined constants)
- Wait forever. At this point, incoming real-time TM packets notification will trigger the user-supplied call-backs (in our example client, data are printed to standard output).

8.2 Compiling and Linking

8.2.1 SCOS DEVELOPMENT ENVIRONMENT

The executable
`hpmcs/addon-exif-rep/testbin/{Platform}/RTSI`
is automatically built by the standard build process (see HPMCS SRN for details).

8.2.2 STAND-ALONE DEVELOPMENT ENVIRONMENT

Preliminaries:

- Recent C++ compiler
- OmniORB4.0x library and tools.
- C++ STL and POSIX thread libraries.
- HPMCS EXIF IDL files copied from (`s2k/addon-exif-rep/src/AAIDL/*.idl`)

The RTSI client is already prepared for reuse in a non-SCOS environment by avoiding SCOS-specific libraries. In this spirit, the C++ STL classes are used instead of their SCOS counterparts (e.g. "std::string" instead of "String").
Since the Makeinfo file is SCOS-specific, a new Makefile has to be created. The following hints might be helpful:

- Obtain the CORBA IDL files from the HPMCS source code distribution
- Make sure that the IDL compiler (omniidl) produces so-called "TIE" interfaces.
- Make sure that the C++ compiler also processes the C++ files generated by the IDL compiler.
- Make sure that the linker can find the omniORB libraries.

Note: If you are using a different version of SCOS-2000 than HPMCS, see also section 7.1.

8.3 Running

1. Environment variables

```
setenv NAMES_SERVERS        sdshost:20000
setenv DOMAIN_ID            0       # this is the default
setenv S2K_LOCAL_SRV_FAMILY PRIME   # this is the default
```

Notes: `NAMES_SERVERS` is compatible with SCOS (in fact, it is already set in a SCOS runtime environment). The other two variables have default values.

2. Start program

For Herschel, ~HPMCS/STATIC/testbin/RTSI `-ORBendPoint giop:tcp::32000`
For Planck, ~HPMCS/STATIC/testbin/RTSI `-ORBendPoint giop:tcp::22000`

The port number is fixed for the mission.

3. Expected result

```
atpcypvc /home/sopshp> RTSI -ORBendPoint giop:tcp::32000
Starting RTSI client...
Background thread for event loop created...
RTSI: Ready
Warning: variable "DOMAIN_ID" is not set, using default "0".
Enter 'Q' to quit
```

3. Example Output

Whenever a packet is received, some properties and the contents of the packet are printed:

```
FilingTime: 2006.159.16.18.03.954
CreateTime: 2006.159.16.18.03.954
VCID:  0
APID:  1900
SSC:   0
TYPE:  0
STYPE: 135
PI1:   0
PI2:   0
D/S:   1
SPID:  100001000
GSID:  0

0000004100000000000ff0000003000
```

Note:

- All diagnostic and error messages are written to standard error.
- All packet output is written to standard output.