

**IFSI  
INAF**

**Herschel PACS  
DPU On-Board Software  
Validation and Verification Plan  
Acceptance Test Plan**

Ref.: PACS-CR-PL-012  
Issue: 4.9  
Date: 4 June 2009  
Page: 1 of 48

## **Herschel PACS**

# **DPU On-Board Software Validation and Verification Plan/ Acceptance Test Plan**

**Document Ref.: PACS-CR-PL-012**

**Issue: 4.9**

---

Prepared by: Stefano Pezzuto

Date: 4 June 2009

Approved by: Renato Orfei



## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Purpose of the document . . . . .	5
1.2	Acronyms and Glossary . . . . .	5
1.3	Document List . . . . .	5
1.3.1	Applicable Documents . . . . .	5
1.3.2	Reference Documents . . . . .	6
1.4	Environmental Needs . . . . .	6
1.4.1	Switch on procedure . . . . .	7
1.5	Test case pass/fail criteria . . . . .	8
<b>2</b>	<b>Test procedures for OBS acceptance</b>	<b>8</b>
2.1	Test procedures for switch-on and loading a new image; test of the Housekeeping report (first part) . . . . .	10
2.2	Test of the Telecommand Verification Service . . . . .	12
2.3	Test of the Memory Management Service (first part) . . . . .	13
2.4	Test of the 1355 initialization . . . . .	16
2.5	Test of the Function Management Service (first part) . . . . .	18
2.5.1	DPU commands in standalone configuration . . . . .	19
	Set HK list . . . . .	19
	Enable/Disable and Start autonomy function . . . . .	19
	UPGRADE_SEQ, ADD_SEQ, DEL_SEQ . . . . .	20
2.6	Test of the Time Management Service . . . . .	24
2.7	Test of the Test Service . . . . .	24
2.8	1355 link activation . . . . .	25
2.9	Test of the Function Management Service (second part) . . . . .	27
2.10	Test of the Memory Management Service (second part) . . . . .	31
2.11	Test of the On Board Control Procedure Service . . . . .	32
2.12	Test of the Housekeeping report (second part) . . . . .	39
2.13	Test of the Packet Transmission Control Service and Telemetry Rate . . . . .	40
2.14	Last testing activities . . . . .	43
<b>A</b>	<b>Matrix of compliance</b>	<b>44</b>
<b>B</b>	<b>TC—TM reference table</b>	<b>48</b>



**IFSI**  
**INAF**

**Herschel PACS**  
**DPU On-Board Software**  
**Validation and Verification Plan**  
**Acceptance Test Plan**

Ref.: PACS-CR-PL-012  
Issue: 4.9  
Date: 4 June 2009  
Page: 3 of 48

## Document Status Sheet

<b>Document Title: DPU On-Board Software Validation and Verification Plan</b>			
<b>Issue</b>	<b>Revision</b>	<b>Date</b>	<b>Reason for change</b>
Draft 0.5		8th January 2002	First version
1	0	18th April 2002	Inserted comments from H. Feuchtgruber
1	1	3rd September 2002	Inserted detailed test procedures for acceptance tests
2	0	28th July 2003	Inserted 1553 test procedures and burst mode
2	1	3rd November 2003	Comments from ST (ref. SCI-PT/19793)
2	2	1st October 2004	Version to be delivered to CGS
2	3	18th October 2004	Changes due to SCOS2000 capabilities
3	0	25th November 2004	RID from CGS
3	1	30th November 2004	Comments from CGS
3	2	1st December 2004	Modification to take into account ECSS-E-40 standard
3	3	16th December 2004	Corrected on-line during acceptance test at IFSI
4	0	24th February 2006	Comments by Ed Bach (CAPTEC) Version for code review
4	1	5th May 2006	Comments during code review in CAPTEC OBS version 8.31
4	2	9th June 2006	Corrections based on the results of the test procedures run following issue 4.1 of this document. The present issue will be used during integration of OBS version 8.31 in PFM at CGS
4	3	20th July 2006	OBSW version 8.32
4	4	6th April 2007	Prepared for DRB (OBSW version 8.46 - FM)
4	9	4th June 2009	OBSW version 9.04 (jump in version number to follow the Test Report version number)



**IFSI**  
**INAF**

**Herschel PACS**  
**DPU On-Board Software**  
**Validation and Verification Plan**  
**Acceptance Test Plan**

Ref.: PACS-CR-PL-012  
Issue: 4.9  
Date: 4 June 2009  
Page: 4 of 48

## Document Change Records

<b>Document Title:</b> DPU On-Board Software Validation and Verification Plan	
<b>Document Reference Number:</b> PACS-CR-PL-012	
<b>Document Issue/Revision Number:</b> 4/9	
<b>Section</b>	<b>Reason For Change</b>
1.3.2	New version of RD-3 and RD-4
1.4	New versions of MIB and OBSW
2.1	Changed end address (2nd parameter) at step #10
2.2	Added a note to step #3
2.3	Changed checksum at step #4.1; end address and checksum at step #4.2; new step #4.3
2.5.1	Changed description of the procedure before step #4; toggled steps #4 and #7; deleted step #8 (all subsequent step have been renumbered); changed description before (new) step #8; adapted all the steps from #10 onward to the new OBSW
2.9	Added a note in the procedure description
2.11	Added a note to the procedure description; upgraded steps #1, #9.1 and #9.3
2.13	Changed procedure description; changed step#10



You'd better watch out  
there may be dogs about  
I have looked over Jordan and I have seen  
things are not what they seem.  
Roger Waters. Sheep

## 1 Introduction

### 1.1 Purpose of the document

This document presents the test plan and procedures for the verification and validation of the DPU OBS for the PACS instrument. This test plan deals with all the DPU PACS OBS components as specified in AD3 and constitutes the PACS DPU OBS acceptance test plan.

For the SW development PACS adopts the standard given in AD1 tailored to be fully equivalent to RD1.

### 1.2 Acronyms and Glossary

<b>AVM</b>	Avionic Model
<b>BC</b>	Bus Controller
<b>CDMS</b>	Command and Data Management System
<b>DEC/MEC</b>	Detector & Mechanism Controller
<b>DM</b>	Data Memory (DSP)
<b>DPU</b>	Digital Processing Unit
<b>DSP</b>	Digital Signal Processor
<b>EGSE</b>	Electrical Ground Support Equipment
<b>ESA</b>	European Space Agency
<b>HK</b>	Housekeeping
<b>HW</b>	Hardware
<b>IFSI</b>	Istituto di Fisica dello Spazio Interplanetario
<b>OBCP</b>	On-Board Control Procedure
<b>OBS</b>	On-Board Software
<b>PACS</b>	Photoconductor Array Camera and Spectrometer
<b>PM</b>	Program Memory (DSP)
<b>SPU</b>	Signal Processing Unit
<b>TC</b>	Telecommand
<b>TM</b>	Telemetry

### 1.3 Document List

#### 1.3.1 Applicable Documents

<b>Ref.</b>	<b>Name</b>	<b>Number/version/date</b>
AD1	Space engineering - Software - Part 1: Principles and requirements	ECSS-E-40 28 November 2003
AD2	PACS DPU OBS User Requirements Document	PACS-CR-RD-001 Issue 3.0. 5th May 2006
AD3	PACS DPU OBS Software Specifications Document	PACS-CR-SR-013 Issue 3.1. 5th May 2006
AD4	DPU/ICU On Board Software Product Assurance Plan	IFSI/OBS/PL/2000-001 Issue 1.1. 2 April 2001



### 1.3.2 Reference Documents

Ref.	Name	Number/version/date
RD1	Guide to applying the ESA software engineering standards to small software projects	BSSC(96)2 Issue 1. May 1996
RD2	Packet Structure Interface Control Document	SCI-PT-ICD-07527 Issue 5.0. 20 July 2004
RD3	PACS DPU OBS User Manual	PACS-CR-UM-024 Issue 3.4. 8 June 2009
RD4	PACS OBCPs and DMC Sequences	PACS-ME-LI-005 Issue 2.1. 8 June 2009
RD5	SMCS332 User Manual	DIPSAPII-DAS-31-06 Issue 2. 21 April 1999

Note: RD-3 and RD-4 have dates later than this document, see PACS-CR-TR-022, issue 4.9

### 1.4 Environmental Needs

The following equipment must be available in order for the complete tests to be carried out:

1. EGSE, complete of:
  - (a) SCOS2000 system, consisting of:
    - i. Personal computer running Linux SUSE 7.3
    - ii. SCOS2000 Version 2.3e + patch 5.1
    - iii. MIB version 9.4 (Flight Model) or 1.1 (Flight Spare)
  - (b) Router (java application SW provided by SRON)
  - (c) CDMS simulator, provided by RAL, consisting of:
    - i. Personal Computer running Windows
    - ii. DDC 1553 interface board version BU-65549
    - iii. Application SW CDMS.SIM version 2.4, including the buslists and the telecommands that will be used in the tests
    - iv. A set of buslists
  - (d) Subsystems simulators consisting of:
    - i. Personal Computer running Windows NT or 2000
    - ii. Spacewire 1355 interface board, drivers and test program version 1.4
    - iii. Application SW to simulate DEC/MEC<sup>1</sup>, Version 6.11b, provided by CSL
    - iv. Application SW to simulate SPU, Version 8 revision 7, provided by University of Wien. The same SW simulates both SPU: blue or short wavelength SPU connected to link 1 of the 1355 interface, and red or long wavelength SPU connected to link 2
2. Support SW Tools:
  - (a) ObswLoader script resident on the SCOS2000 computer, used to uplink the series of TC (6,2) commands with the image of the OBS executable. Loading procedure is described in RD3
  - (b) PacketRecorder script resident on the SCOS2000 computer, used to save on disk all the telemetry packets received from the router
  - (c) TCGEN application SW version 2 to generate the list of Telecommmands to uplink the OBS

<sup>1</sup>Strictly speaking DPU interfaces with MEC only. For simplicity, in the following, MEC, DEC, DEC/MEC or DMC will be used as synonyms.



- (d) 1355 Communication library provided by CSL, consisting of an application program running in a DOS window

The DPU OBS under test is version 9.04 and will be available on the SCOS2000 system as a set of TC (6,2) telecommands produced with the TCGEN procedure; instructions to upload the TCs are in RD3.

On the same PC hosting the CDMS Simulator the following buslists will be available: Pacs\_Nominal.txt and PACS\_Burst.txt.

#### **1.4.1 Switch on procedure**

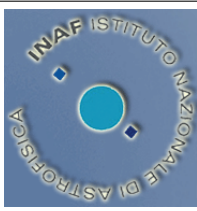
The following steps should be performed to setup the system ready for use:

##### **1. Start SCOS2000**

- (a) Login as user sops23e passwd .TestControl
- (b) Type startx and press return
- (c) Open a terminal window
- (d) Go to /home/sops23e (type cd)
- (e) Execute script set\_links\_PACS\_MIB (type ./set\_links\_PACS\_MIB)
- (f) Execute script s2.start (type s2.start)
- (g) Change directory to local-bin (type cd local-bin)
- (h) Execute script StartRouter (type ./StartRouter)
- (i) Select EGSEServ, MON1, MSTK2, VPD2 and TCHIST2 buttons
- (j) Press Start and confirm
- (k) When two bars appear in the top and bottom of the desktop, login as username Matt (passwd Matt) with role SUPE\_001
- (l) Disable warning bell. Press button Alarm tone disable, click alarm checkbox and confirm
- (m) On the MON telemetry desktop window, click the AND button at bottom left of the window and select, one after the other, the following items
  - PA000380- -->DPU HouseKeeping (show all the DPU HK)
  - PA026380- -->TC VERIFICATION (report all the TM (1,x))
  - PA003390- -->SPUL HouseKeeping (show the counter SPUL\_CIB used in the procedure 2.1.2)
  - PA004400- -->SPUS HouseKeeping (show the counter SPUS\_CIB used in the procedure 2.1.2)
  - PA019420- -->DMC CONTROLLER AND STATUS (show the counter DMC\_DPU\_CTRL\_PAC used in the procedure 2.1.2)
  - PA025380- -->TM PACKET HEADER (used in the procedure 2.7 for TM (17,2))
  - PA027380- -->EVENT (report all the TM (5,x))
  - PA028380- -->DPU TIME (used in the procedure 2.6 for TM (9,9))
  - PA029380- -->MEMORY CHECK REPORT (show the content of TM (6,10))
- (n) Wait until the Manual Command Stack (MSTK), the Variable Packet Displayer (VPD) and the Telecommand History (TCHIST) windows appear

At this point the SCOS2000 system is up and running

##### **2. Start the CDMS Simulator**



- (a) Click on icon CDMS\_SIM
- (b) Select the PACS\_Nominal buslist
- (c) Click on Launch Router Command Interface
- (d) Click on Connect
- (e) Select NAME\_CLIENT; write pacs and click Send Command
- (f) Select ADD\_CLIENT, write 480, click Send Command
- (g) Click on Close Without Sending

The CDMS is ready to be started.

3. Start the subsystem simulators. This step is intended to be performed just before test procedure 2.8 and it is reported here for the sake of readability. If it is not executed at that moment it is likely that links will not be started. However, the PC running the simulators must be switched on before DPU.
  - (a) In the directory containing the DEC simulator edit the file Spacewire.cfg; comment all the lines with a trailing %; add the following line 0 1355 SLAVE 0; save the file
  - (b) In the directory containing the blue SPU simulator edit the file Spacewire.cfg; comment all the lines with a trailing %; add the following line 1 1355 SLAVE 1; save the file
  - (c) In the directory containing the red SPU simulator edit the file Spacewire.cfg; comment all the lines with a trailing %; add the following line 1 1355 SLAVE 2; save the file
  - (d) Double-Click on the Comexe.exe icon on the desktop
  - (e) Double-Click on the DEC simulator icon on the desktop
  - (f) Double-Click on the blue SPU simulator icon on the desktop
  - (g) Double-Click on the red SPU simulator icon on the desktop

## 1.5 Test case pass/fail criteria

Test criteria are based on the direct inspection of TC confirmation, Event and HK TM Packets received by the EGSE. No facility is foreseen to inspect the content of science packets so that the simple reception of these packets by the EGSE is the pass criterium.

## 2 Test procedures for OBS acceptance

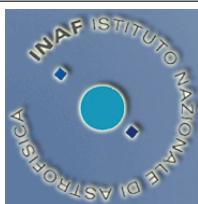
The “Start CDMS Simulator” step implies that the initialization procedure is carried out as specified in Section 1.4.1-2.

Unless otherwise noted, all commands are to be sent from SCOS2000: go on the Manual Stack window, click on the CMD button and select the command from the list. Press ok and choose the telecommand parameters if specified in the test procedure. Press ok, then ARM and GO.

In the tables, column Time gives the Release time as reported in the TCHIST2 window; when a telemetry packet is indicated as TM (x,y), x stands for packet type and y stands for packet subtype. Unless otherwise specified, to verify reception of a packet means to use the Telemetry Desktop window PA022380 where the fields HD\_PACKET\_TYPE, HD\_PCKT\_SUBTYPE and HD\_APID report the type x, subtype y and APID, respectively. “Verify reception of TM (1,x)” means instead to use the Telemetry Desktop window PA023380.

The VPD2 window is used for the following TM packets: (6,6), (14,4), (18,9), (18,11) and (18,13). The first time the window appears type **65535** in the Data Stream cell and **1152** in the APID cell, these values will be never changed. “Verify reception of TM (x,y)”, for instance (14,4), means to push STOP button, to type 14 in the cell named Type and 4 in the cell Sub-Type, to push Apply button and then to push button LIVE in the right top bar.





**IFSI**  
**INAF**

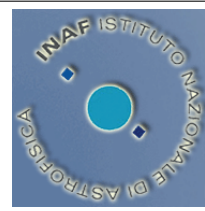
**Herschel PACS**  
**DPU On-Board Software**  
**Validation and Verification Plan**  
**Acceptance Test Plan**

Ref.: PACS-CR-PL-012  
Issue: 4.9  
Date: 4 June 2009  
Page: 9 of 48

Unless otherwise noted, each test procedure starts from the final conditions of the previous one.

**Important!** If the DPU under test is the redundant unit the APID reported in the procedures must be changed according to this table

If a test procedure reports APID use this APID for the redundant unit	
0x480	0x481
482	483
484	485
48A	48B
488	489
486	487



### 2.1 Test procedures for switch-on and loading a new image; test of the Housekeeping report (first part)

The purpose is to demonstrate the ability of the OBS to correctly initialize the system, to start-up an image stored in EEPROM or to upload a new image. The new image is then written in EEPROM with an OBCP. With this test procedure the following requirements are verified: OBS-OPR-SR1, OBS-FSR-HK1 to OBS-FSR-HK3; OBS-PRF-SR4; OBS-OPR-SR16 and OBS-OPR-SR18.

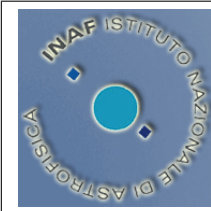
#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Start CDMS Simulator by pressing the Start/Stop BC button	None			
2	Switch-on the DPU. At this point the Boot Software check the OBS image in the EEPROM. After completion, the Boot SW waits for a telecommand	An event TM (5,1) should be received by SCOS2000			
3	On a terminal window of the computer hosting SCOS2000 type the following commands > cd > ../local-bin/PacsDpuObswLoader -interval.500 DPU_PACS_TC/*	Verify that the telecommands appear in the TC log window of CDMS; the boot SW sends an event (5,1) for each TC but no TM (5,4) should be received			
4	Wait until all the TC are uploaded (prompt > appears again in the window). About 13 minutes are needed to complete the operation				
5	Send DPULLSW_LOAD_TC_AND_BOOT command to copy the OBS image from DM to PM and start the OBS	Both essential (APID 0x480) and nominal (APID 0x482) HK TM packets TM (3,25) should be received			
6		Verify that an HK packet with APID 0x482 is received every 2 seconds and an HK packet with APID 0x480 is received every 10 seconds			
7	Stop the Bus Controller by pressing the Start/Stop BC button				

Test Conductor:	PA:	Date:
-----------------	-----	-------



8	Open the file TelemetryA.txt in the sub-directory "TC&TM Log Files" of the CDMS simulator	Verify on CDMS that the first two HK packets are equal, with the exception of: the APID — 0x0C82 for the first and 0x0C80 for the second packet; time stamp; SID (is the 9th word) — 3 for the first and 4 for the second packet; the checksum			
9		Verify that in the window Telemetry Desktop PA000380 all the values appear in green with the exception of 9 values in red just after DP_T			
10	Send DPU_START_OBCP command with Repeater Value: 5 and PROCEDURE_ID= 20 PAR_ID= 1 PAR_VALUE= 0x4000 PAR_ID= 2 PAR_VALUE= 0x10CB4 PAR_ID= 3 PAR_VALUE= 1 PAR_ID= 4 PAR_VALUE= 0x19660502 PAR_ID= 5 PAR_VALUE= 0	Verify reception of TM (1,1) and TM (1,3). Wait about 10/12 seconds until a TM (1,7) is received			
11	Switch-off DPU. Repeat steps #1 and 2	An event TM (5,1) should be received by SCOS2000			
12	Send DPULLSW_FORCE_BOOT command with Repeater Value 1 and DPU_PARTITION = 1 to start the OBS	Both essential (APID 0x480) and nominal (APID 0x482) HK TM packets TM (3,25) should be received			

Test Conductor:	PA:	Date:
-----------------	-----	-------

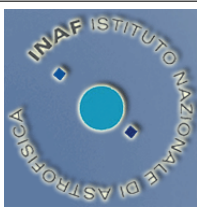


## 2.2 Test of the Telecommand Verification Service

With this procedure the correct reception of telecommands from the CDMS is tested. A correct command (8,1) is sent to receive an acceptance report — Success (1,1), while a set of incorrect commands is provided to test the acceptance report — Failure (1,2). This set of requirement is verified: OBS-FSR-TC1, OBS-FSR-TC7 to OBS-FSR-TC10, OBS-FSR-TC12 to OBS-FSR-TC14. The command (8,1) has been chosen because it does not require any action after the generation of the TM report (1,1).

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send TEST_START_FUNCNT command	Verify reception of TM (1,1) in window PA026380			
2.1	<u>WRONG APID</u> Stop the CDMS simulator by pressing the Start/Stop BC button and then kill the application. In the subdirectory "Other Files" of the CDMS simulator open the file APID2RT.txt. Modify the line 0x480 25 PACS with 0x4FD 25 PACS Save and close the file. Start the CDMS simulator following Section 1.4 but writing 4FD at Step (f)				
2.2	Send TEST_APID command (this command has the APID set to 1277. In the telemetry packet seen on SCOS, the number 7421 is read since the CDMS adds 6144)	Verify reception of TM (1,2) with ACP_FAIL_CODE Illegal APID ACP_INVALID_VAL 7421			

Test Conductor:	PA:	Date:
-----------------	-----	-------



2.3	<p>Stop the CDMS simulator by pressing the Start/Stop BC button and then kill the application. Open the file APID2RT.txt. Modify the line 0x4FD 25 PACS back to 0x480 25 PACS Save and close the file. Start the CDMS simulator according to Section 1.4</p> <p><u>WRONG PACKET LENGTH</u></p>					
3	<p>Send TEST_LENGTH command. The length of the application data field is set to 40 bits, while it must be N x 16 bits</p>	<p>Verify reception of TM (1,2) with ACP_FAIL_CODE Invalid LENGTH ACP_INVALID_VAL 17 AC.EXPECT_VALUE 16</p>				<p>Because the CDMS is started again the first command sent to DPU has the same TC confirmation as the previous one, so it is not recognized as a new command inside the DPU. Send this command twice</p>
4	<p><u>WRONG TYPE</u> Send TEST_TYPE command</p>	<p>Verify reception of TM (1,2) with ACP_FAIL_CODE Illegal TYPE ACP_INVALID_VAL 4</p>				
5	<p><u>WRONG SUBTYPE</u> Send TEST_SUBTYPE command</p>	<p>Verify reception of TM (1,2) with ACP_FAIL_CODE Illegal SUBTYPE ACP_INVALID_VAL 4</p>				

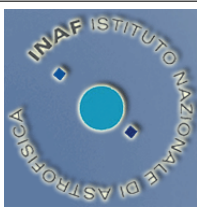
Test Conductor:	PA:	Date:
-----------------	-----	-------

### 2.3 Test of the Memory Management Service (first part)

By using the standard manual commanding application of SCOS2000 the memory load and dump services can be tested only for DRAM. To test the behaviour of OBS in case of wrong checksum after data have been written in memory, we use the address of SMC332 ISR which can only be read. When the DPU reads the register to verify that memory writing was successful it finds a different value (see step 1.3). Note that DPU reports a TM (1,1) after every telecommand. For simplicity this TM packet is not reported in the last column of the table.

To test memory load we use the absolute address which corresponds to the counter of lost telecommands (HK not used in these procedures): at startup its value is 0. After step #1.4 the counter jumps to value 2000 (0x7D0).

The following set of requirements is covered: OBS-FSR-TC16, OBS-FSR-TC17; OBS-FSR-EV3; OBS-FSR-MM1 to OBS-FSR-MM9.



#	Action	Expected Reaction	Observed Reaction	Time	Notes
	<b>MEMORY LOAD</b> The command to send is always DPU_MEMORY_LOAD_DRAM with Repeater Value: 1. The other parameters are specified for each step				
1.1	Parameters: DPU_MEMORY_BLOCK_ID = 0xA100 DPU_MEMORY_ADDRESS = 0x1D11 DPU_4_BYTES_WORDS_DATA=0x7D0 DPU_CHECKSUM = 0xD62A	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid MEMID INVALID_VALUE = 0xA1 (161)			
1.2	Parameters: DPU_MEMORY_BLOCK_ID = 0x1100 DPU_MEMORY_ADDRESS = 0x1D11 DPU_4_BYTES_WORDS_DATA=0x7D0 DPU_CHECKSUM = 0x9000	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Inv CRC 1 CHK INVALID_VALUE = 0xD62A (54826)			
1.3	Parameters: DPU_MEMORY_BLOCK_ID = 0x1200 DPU_MEMORY_ADDRESS = 4 DPU_4_BYTES_WORDS_DATA=0x7D0 DPU_CHECKSUM = 0xD62A	Verify reception of TM (1,8) with EXE_FAIL_CODE = Resource FAIL FAILURE_REASON = Inv CRC 2 CHK INVALID_VALUE = 0x84C0 (33984)			
1.4	Parameters: DPU_MEMORY_BLOCK_ID = 0x1100 DPU_MEMORY_ADDRESS = 0x1D11 DPU_4_BYTES_WORDS_DATA=0x7D0 DPU_CHECKSUM = 0xD62A	Verify reception of TM (1,7). In window PA000380 the HK DP_TC_LOST is now 2000			
1.5	Parameters: DPU_MEMORY_BLOCK_ID = 0x1300 DPU_MEMORY_ADDRESS = 0x0000 DPU_4_BYTES_WORDS_DATA=0x7D0 DPU_CHECKSUM = 0xD62A	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid MEMID INVALID_VALUE = 0x13			
	<b>MEMORY DUMP</b>				
Test Conductor:		PA:			Date:



<p>2.1</p> <p>Send DPU_MEMORY_DUMP command with parameters: DPU_MEMORY_BLOCK_ID = 0x1300 (EEPROM) DPU_MEMORY_ADDRESS = 2 DPU_DATA_LENGTH = 1</p>	<p>On VPD Display Window select Type and Subtype (6,6) and verify reception of TM (6,6) with MEMORY_ID = 0x1300 START_ADDRESS = 2 LENGTH = 1 DATA_WORD = 0x80000400 CHECKSUM=0x953C</p>			
<p>2.2</p> <p>Send DPU_MEMORY_DUMP command with parameters: DPU_MEMORY_BLOCK_ID = 0x1600 (DM in PM) DPU_MEMORY_ADDRESS = 0x15 DPU_DATA_LENGTH = 5</p>	<p>On VPD Display Window select Type and Subtype (6,6) and verify reception of TM (6,6) with MEMORY_ID = 0x1600 START_ADDRESS = 0x15 LENGTH = 5 DATA_WORD = 0x31 DATA_WORD = 0x32 DATA_WORD = 0x20 DATA_WORD = 0x67 DATA_WORD = 0x69 CHECKSUM=0x5138</p>			
<p>3.1</p> <p>MEMORY CHECK Send DPU_MEMORY_CHECK command with parameters: DPU_MEMORY_BLOCK_ID = 0x1300 (EEPROM) DPU_MEMORY_ADDRESS = 2 DPU_DATA_LENGTH = 1</p>	<p>Verify reception of TM (6,10) on PA029380 window with MEMORY_ID = 0x1300 START_ADDRESS = 2 LENGTH = 1 CHECKSUM = 0x953C</p>			
<p>3.2</p> <p>Send DPU_MEMORY_CHECK command with parameters: DPU_MEMORY_BLOCK_ID = 0x0100 (PM) DPU_MEMORY_ADDRESS = 0x30 DPU_DATA_LENGTH = 1</p>	<p>Verify reception of TM (6,10) on PA029380 window with MEMORY_ID = 0x0100 START_ADDRESS = 0x30 LENGTH = 1 CHECKSUM = 0xB874</p>			

Test Conductor:	PA:	Date:
-----------------	-----	-------



4.1	Send DPU_CHECK_PM command with parameters: DPU_START_ADDRESS = 0x5551 DPU_END_ADDRESS = 0x10CB4 DPU_CHECKSUM = 0x69E1	Verify reception of TM (1,7)			
4.2	Send DPU_CHECK_PM command with parameters: DPU_START_ADDRESS = 0x5551 DPU_END_ADDRESS = 0x10CB4 DPU_CHECKSUM = 0xA990	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Invalid CRC INVALID_VALUE = 0xA99069E1; verify reception of an event (5,2) in CDMS window with EVENT_ID = PM FAILURE EVENT_SID = 1 FIRST_VALUE_1 = 0xA990 SECOND_VALUE_1 = 0x69E1			
4.3	Send DPU_CHECK_PM command with parameters: DPU_START_ADDRESS = 0x5551 DPU_END_ADDRESS = 0x4000 DPU_CHECKSUM = 0xA990	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid PARAM INVALID_VALUE = 0			

#### 2.4 Test of the 1355 initialization

The requirements OBS-IRF-SR77 to OBS-IRF-SR82 are here covered by reading the SMCS332 registers (see RD5).

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 2 DPU_MEMORY_ADDRESS = 0	On VPD Display verify reception of TM (6,6) with 1st DATA_WORD = 2 2nd DATA_WORD = 0x10			

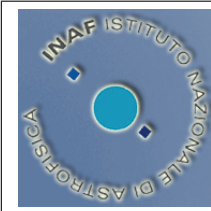
Test Conductor:	PA:	Date:
-----------------	-----	-------





2.1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0x10	Verify reception of TM (6,6) with 1st DATA_WORD = 0xC			
2.2	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0x30	Verify reception of TM (6,6) with 1st DATA_WORD = 0xC			
2.3	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0x50	Verify reception of TM (6,6) with 1st DATA_WORD = 0xC			
3.1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0x1F	Verify reception of TM (6,6) with 1st DATA_WORD = 0x33			
3.2	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0x3F	Verify reception of TM (6,6) with 1st DATA_WORD = 0x33			
3.3	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1200 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0x5F	Verify reception of TM (6,6) with 1st DATA_WORD = 0x33			

Test Conductor:	PA:	Date:
-----------------	-----	-------

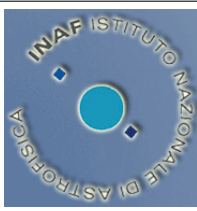


## 2.5 Test of the Function Management Service (first part)

The following set of requirements is here covered: OBS-FSR-FM1 TO OBS-FSR-FM3, OBS-FSR-FM5, OBS-FSR-FM8.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send TEST_START_FUNCTION command with DPU_FUNC_ID = 100 DPU_ACTIVITY_ID = 0	Verify reception of TM (1,1)			
2	Send TEST_STOP_FUNCTION command with DPU_FUNC_ID = 100 DPU_ACTIVITY_ID = 0	Verify reception of TM (1,1)			
3	Send TEST_REPORT_FUNCTION command with DPU_FUNC_ID = 100 DPU_ACTIVITY_ID = 0	Verify reception of TM (1,1)			
4	Send TEST_FUNC_ID command with DPU_FUNC_ID= 0 DPU_ACTIVITY_ID= 1 DPU_STRUCTURE_ID=0	Verify reception of TM (1,1) and TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid FUN- ID INVALID_VALUE = 0			
5	Send TEST_FUNC_ID command with DPU_FUNC_ID= 104 DPU_ACTIVITY_ID= 1 DPU_STRUCTURE_ID= 0	Verify reception of TM (1,1) and TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid FUN- ID INVALID_VALUE = 104 (0x68)			

Test Conductor:	PA:	Date:
-----------------	-----	-------



### 2.5.1 DPU commands in standalone configuration

#### Set HK list

This procedure covers the following requirements: OBS-FSR-HK1, OBS-FSR-HK4, OBS-FSR-HK5.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 2, SPU_ARRAY = 1	Verify on Telemetry Desktop window PA000380 that DP_TM_RATE = PHOT			
2	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 1, SPU_ARRAY = 1	Verify that DP_TM_RATE = SPEC			
3	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 4, SPU_ARRAY = 1	Verify that DP_TM_RATE = NO PRIME			

#### Enable/Disable and Start autonomy function

This command makes DPU execute an autonomy function even if the conditions that should trigger the function are not met. For this purpose the autonomy function with ID=23 is used: it is called after an error in one of the three 1355 links and generates the event PACS Nominal Off. After switch on this autonomy function is disabled. This procedure covers the following requirements: OBS-FSR-TC17, OBS-FSR-EV2, OBS-FSR-SR2, OBS-FSR-SR5.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
4	Send DPU_SET_FUNC command with DPU_INT_FUNC_ID = 23, ENABLE_DISABLE = 1	Verify reception of TM (1,1)			
5	Send DPU_FORCE_EXECUTION_AF command with DPU_INT_FUNC_ID = 23	Verify on Telemetry Desktop window PA027380 that the event with EVENT_ID = HK_DPU_SOFT and then the event PACS_NOM_OFF are generated, and that the field EVENT_COUNTER of the second event is incremented by 1 with respect to the first event			

Test Conductor:

PA:

Date:



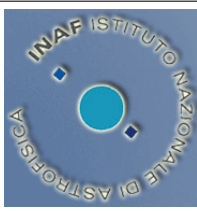
6	Send DPU_SET_FUNCT command with DPU_INT_FUNCT_ID = 23 ENABLE_DISABLE = 0	Verify reception of TM (1,1)			
7	Send DPU_FORCE_EXECUTION_AF command with DPU_INT_FUNCT_ID = 23	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Funct_STOPPED INVALID_VALUE = 2			

**UPGRADE\_SEQ, ADD\_SEQ, DEL\_SEQ**

For test purpose a dummy sequence is used with SEQUENCE\_ID=20, and code: 0x1, 0x10000, 0x30000, with checksum=0x8EEB. This will be upgraded with the code: 0x1, 0x10000, 0x20000, 0x30000, with checksum=0xCE93. This procedure covers the requirements OBS-FSR-TC2 and OBS-OPR-SR6.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
8	Send DPU_ADD_SEQ command with Repeater Value: 3 and SEQUENCE_ID = 33 DPU_4_BYTES_WORDS_DATA = 0x1 DPU_4_BYTES_WORDS_DATA = 0x10000 DPU_4_BYTES_WORDS_DATA = 0x30000 DPU_CHECKSUM = 0x8EEB	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid SEQ_ID INVALID_VALUE = 33			
9	Send DPU_ADD_SEQ command with Repeater Value: 3 and SEQUENCE_ID = 10 DPU_4_BYTES_WORDS_DATA = 0x1 DPU_4_BYTES_WORDS_DATA = 0x10000 DPU_4_BYTES_WORDS_DATA = 0x30000 DPU_CHECKSUM = 0x8EEB	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Invalid SEQ_ID INVALID_VALUE = 10			

Test Conductor:	PA:	Date:
-----------------	-----	-------



10	Send DPU_DEL_SEQ command with SEQUENCE_ID = 20	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Invalid SEQ-ID INVALID_VALUE = 20			
11	Send DPU_UPGRADE_SEQ command with Repeater Value: 4 and SEQUENCE_ID = 20 DPU_4_BYTES_WORDS_DATA = 0x1 DPU_4_BYTES_WORDS_DATA = 0x10000 DPU_4_BYTES_WORDS_DATA = 0x20000 DPU_4_BYTES_WORDS_DATA = 0x30000 DPU_CHECKSUM = 0xCE93	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Invalid SEQ_ID INVALID_VALUE = 20			
12	Send DPU_ADD_SEQ command with Repeater Value: 3 and SEQUENCE_ID = 20 DPU_4_BYTES_WORDS_DATA = 0x1 DPU_4_BYTES_WORDS_DATA = 0x10000 DPU_4_BYTES_WORDS_DATA = 0x30000 DPU_CHECKSUM = 0x8EEC	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid CRC INVALID_VALUE = 0x8EEB (36587)			
12.1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1100 DPU_MEMORY_ADDR = 0xB91 DPU_DATA_LENGTH = 3	Verify on VPD displayer the reception of TM (6,6) with all the data words equal to 0, CRC = 0x84F9			

Test Conductor:	PA:	Date:
-----------------	-----	-------



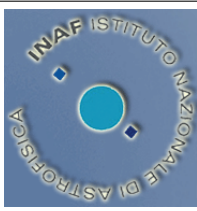
13	Send DPU_ADD_SEQ command with Repeater Value: 3 and SEQUENCE_ID = 20 DPU_4_BYTES_WORDS_DATA = 0x1 DPU_4_BYTES_WORDS_DATA = 0x10000 DPU_4_BYTES_WORDS_DATA = 0x30000 DPU_CHECKSUM = 0x8EEB	Verify reception of TM (1,1)			
13.1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1100 DPU_MEMORY_ADDR = 0xB91 DPU_DATA_LENGTH = 3	Verify on VPD displayer the reception of TM (6,6) with 1st DATA_WORD = 0x1 2nd DATA_WORD = 0x10000 3rd DATA_WORD = 0x30000 DPU_CHECKSUM = 0x8EEB			
14	Send DPU_DEL_SEQ command with SEQUENCE_ID = 20	Verify reception of TM (1,1)			
14.1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x1100 DPU_MEMORY_ADDR = 0xB91 DPU_DATA_LENGTH = 3	Verify on VPD displayer the reception of TM (6,6), with all the data words equal to 0, and CRC = 0x84F9			
15	Send DPU_ADD_SEQ command with Repeater Value: 3 and SEQUENCE_ID = 20 DPU_4_BYTES_WORDS_DATA = 0x1 DPU_4_BYTES_WORDS_DATA = 0x10000 DPU_4_BYTES_WORDS_DATA = 0x30000 DPU_CHECKSUM = 0x8EEB	Verify reception of TM (1,1)			

Test Conductor:	PA:	Date:
-----------------	-----	-------



16	<p>Send DPU_UPGRADE_SEQ command with Repeater Value: 4 and SEQUENCE_ID = 20  DPU_4_BYTES_WORDS_DATA = 0x1  DPU_4_BYTES_WORDS_DATA = 0x10000  DPU_4_BYTES_WORDS_DATA = 0x20000  DPU_4_BYTES_WORDS_DATA = 0x30000  DPU_CHECKSUM = 0xCE93</p>	<p>Verify reception of TM (1,1)</p>		
16.1	<p>Send DPU_MEMORY_DUMP command with  DPU_MEMORY_BLOCK_ID = 0x1100  DPU_MEMORY_ADDR = 0xB91  DPU_DATA_LENGTH = 4</p>	<p>Verify on VPD displayer the reception of TM (6,6) with  1st DATA_WORD = 0x1  2nd DATA_WORD = 0x10000  3rd DATA_WORD = 0x20000  4th DATA_WORD = 0x30000  DPU_CHECKSUM = 0xCE93</p>		
17	<p>Send DPU_UPGRADE_SEQ command with Repeater Value: 55 and SEQUENCE_ID = 20  all DPU_4_BYTES_WORDS_DATA = 0  DPU_CHECKSUM = 0xEEAF  This telecommand has a total length of 242 bytes, the largest allowed for a TC</p>	<p>Verify reception of TM (1,1)</p>		

Test Conductor:	PA:	Date:
-----------------	-----	-------



## 2.6 Test of the Time Management Service

The following requirements are here covered: OBS-FSR-TM3 and OBS-FSR-TM4.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_TIME_VERIFICATION command	Verify reception of TM (9,9) in Telemetry Desktop window PA028380			
2		Verify that the time in the application data field is 1 second greater than the time in the packet header. Assuming that the CDMS sends the SYNC signal at the beginning of each second, fractions of seconds in the application data field must be zero			

## 2.7 Test of the Test Service

The following requirements are here covered: OBS-FSR-TS1 and OBS-FSR-TS2.

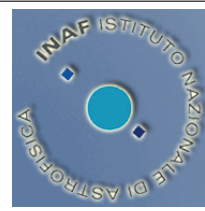
#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_TEST_CONN command	Verify reception of TM (1,1) and TM (17,2) in Telemetry Desktop window PA025380			

Test Conductor:

PA:

Date:





## 2.8 1355 link activation

To further proceed with the test we need to activate the 1355 links in order to communicate with the subsystems simulators. A specific OBCP has been prepared, with ID = 19, which accepts two parameters: the first is the link (0 for DEC, 1 for blue SPU, 2 for red SPU), and the second to select the starting mode (1 means DPU is master, 2 means DPU is slave). It has been decided that DPU is master for all the three links, so only this mode will be tested, even if with this OBCP DPU can be started also as slave. This OBCP covers the requirements OBS-FSR-TC15, OBS-FSR-TC16; OBS-FSR-CP5; OBS-IRF-SR83.

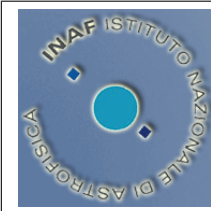
#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Select Telemetry Desktop window PA000380. In the top bar of the window push the little button 2. Now the PA000380 and PA026380 windows are visible at the same time				
2	Start subsystem simulators following Step 3 of Section 1.4.1				
3.1	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID= 19 PAR_ID= 1 PAR_VALUE= 0 PAR_ID= 2 PAR_VALUE= 1	Verify reception of TM (1,1), TM (1,3) and TM (1,7). The HK DP.DMC_LINK goes from OFF to ON			
3.2	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID= 19 PAR_ID= 1 PAR_VALUE= 1 PAR_ID= 2 PAR_VALUE= 1	Verify reception of TM (1,1), TM (1,3) and TM (1,7). The HK DP_SPS_LINK goes from OFF to ON			

Test Conductor:	PA:	Date:
-----------------	-----	-------



3.3	Send DPU_START_OBCP command with Repeater Value: 2 PROCEDURE_ID= 19 PAR_ID= 1 PAR_VALUE= 2 PAR_ID= 2 PAR_VALUE= 1	Verify reception of TM (1,1), TM (1,3) and TM (1,7). The HK DP_SPL_LINK goes from OFF to ON			
4	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID= 19 PAR_ID= 1 PAR_VALUE= 3 PAR_ID= 2 PAR_VALUE= 1	Verify reception of TM (1,1), TM (1,3) and TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid DATUM INVALID_VALUE= 1			
5	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID= 19 PAR_ID= 1 PAR_VALUE= 0 PAR_ID= 2 PAR_VALUE= 0	Verify reception of TM (1,1), TM (1,3) and TM (1,8) with EXE_FAIL_CODE = Invalid Data FAILURE_REASON = Invalid DATUM INVALID_VALUE= 2			
6	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID= 19 PAR_ID= 1 PAR_VALUE= 0 PAR_ID= 2 PAR_VALUE= 3	Verify reception of TM (1,1), TM (1,3) and TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Invalid DATUM INVALID_VALUE= 2			

Test Conductor:	PA:	Date:
-----------------	-----	-------



## 2.9 Test of the Function Management Service (second part)

The subsystems commanding can now be tested. The following requirements are here covered: OBS-FSR-TC17; OBS-FSR-EV1, OBS-FSR-EV2; OBS-FSR-FM1, OBS-FSR-FM4 to OBS-FSR-FM7; OBS-OPR-SR5, OBS-OPR-SR7, OBS-OPR-SR10.

Note that the steps #4.9 and 4.10 can end with a TM(1,8) because when generating science data the SPU simulators are not able to send the acknowledgment in time. This behaviour, not systematic, can be considered nominal. If a TM(1,8) is received send the command DPU\_SET\_FUNCNT with DPU\_INT\_FUNCNT\_ID = 101 (blue SPU) or 102 (red SPU), and ENABLE\_DISABLE = 1, before the next step.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1.1	Send DMC_SET_OBSID command with OBSERVATION_ID = 1	Verify reception of TM(1,1); verify that DP_COM_DMC is 1 in window PA000380; verify in window PA019420 that DM_OBSID = 1			
1.2	Send DMC_SET_BBID command with BUILDING_BLOCK_ID = 1	Verify reception of TM(1,1) and that DP_COM_DMC is 2 in window PA000380; verify in window PA019420 that DM_BBID = 1			
1.3	Send DMC_START_DET_SIMULATOR command with DET_ID_PERIOD = 0x10	Verify reception of TM(1,8) and that in window PA000380 DP_DMC_CMD = SS STOPPED; verify reception of event in window PA027380 with EVENT_ID = NACK EVENT_SID = 6 EVENT_OBSID = 1 EVENT_BBID = 1 1355_LINK_6 = DEC and that EVENT_COUNTER has been incremented			
1.4	Send DMC_SET_BBID command with BUILDING_BLOCK_ID = 5	Verify reception of TM(1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = UNIT STOPPED INVALID_VALUE = 0			

Test Conductor:

PA:

Date:



1.5	Send DPU_SET_FUNCT command with DPU_INT_FUNCT_ID = 103 ENABLE_DISABLE = 1	Verify reception of TM(1,1) and that in window PA000380 DP-DMC-CMD = SS ENABLED			
1.6	Send DMC_SET_BBID command with BUILDING_BLOCK_ID = 5	Verify reception of TM(1,1); verify in window PA000380 that DP_DMC_CMD is 103; verify in window PA019420 that DM_BBID = 5			
2.1	Send SPUS_STOP_REDUCT_COMPR command	Verify reception of TM(1,1); verify that DP_COM_SPUS in window PA000380 is 1			
2.2	Send TEST_SPUS_INVALID_ACT_ID command	Verify reception of TM(1,8) and that in window PA000380 DP-SPUS-CMD = SS STOPPED; verify reception of event in window PA027380 with EVENT_ID = NACK EVENT_SID = 6 EVENT_OBSID = 1 EVENT_BBID = 5 1355_LINK_6 = SPU_S and that EVENT_COUNTER has been incremented			
2.3	Send SPUS_STOP_REDUCT_COMPR command	Verify reception of TM(1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = UNIT STOPPED INVALID_VALUE = 1			
2.4	Send DPU_SET_FUNCT command with DPU_INT_FUNCT_ID = 101 ENABLE_DISABLE = 1	Verify reception of TM(1,1); verify that in window PA000380 DP-SPUS-CMD = SS ENABLED			
2.5	Send SPUS_STOP_REDUCT_COMPR command	Verify reception of TM(1,1); verify that DP_COM_SPUS is 102 in window PA000380			
3.1	Send SPUL_STOP_REDUCT_COMPR command	Verify reception of TM(1,1); verify that DP_COM_SPUL in window PA000380 is 1			

Test Conductor:	PA:	Date:
-----------------	-----	-------



3.2	Send TEST_SPUL_INVALID_ID_ACT_ID command	Verify reception of TM(1,8) and that in window PA000380 DP-SPUL-CMD = SS STOPPED; verify reception of event in window PA027380 with EVENT_ID = NACK EVENT_SID = 6 EVENT_OBSID = 1 EVENT_BBID = 5 1355_LINK_6 = SPU_L and that EVENT_COUNTER has been incremented			
3.3	Send SPUL_STOP_REDUCT.COMPR command	Verify reception of TM(1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = UNIT STOPPED INVALID_VALUE = 2			
3.4	Send DPU_SET_FUNC command with DPU_INT_FUNC_ID = 102 ENABLE_DISABLE = 1	Verify reception of TM(1,1); verify that DP-SPUL-CMD = SS ENABLED in window PA000380			
3.5	Send SPUL_STOP_REDUCT.COMPR command	Verify reception of TM(1,1); verify that DP_COM.SPUL is 102 in window PA000380			
4	Science data reception				
4.1	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 1, SPU_ARRAY = 1	Verify reception of TM(1,1); verify in window PA000380 that DP_TM_RATE is SPEC			
4.2	Send SPUS_WRT_SIM_DATA command with SPUS_4_BYTES_WORDS_DATA set to 33, 23, 64, 0 (repeated 21 times), SPUS_CHECKSUM = 0x34AC	Verify reception of TM (1,1)			

Test Conductor:	PA:	Date:
-----------------	-----	-------



4.3	Send command SPUL_WRT_SIM_DATA with SPUL_4_BYTES_WORDS_DATA set to 33, 23, 64, 0 (repeated 21 times), SPUL_CHECKSUM = 0x34AC	Verify reception of TM (1,1)			
4.4	Send SPUS_WRT_DXS6 command with SPUS_4_BYTES_WORDS_DATA set to 3, 134, 450, 0xFFFF FFFF (repeated 14 times), 0xC000 0000, 0 (repeated 6 times), SPUS_CHECKSUM = 0x3049	Verify reception of TM (1,1)			
4.5	Send SPUL_WRT_DXS6 command with SPUL_4_BYTES_WORDS_DATA set to 3, 134, 450, 0xFFFF FFFF (repeated 14 times), 0xC000 0000, 0 (repeated 6 times), SPUL_CHECKSUM = 0x3049	Verify reception of TM (1,1)			
4.6	Send SPUS_ACT_TEST_SPEC command	Verify reception of TM (1,1)			
4.7	Send SPUL_ACT_TEST_SPEC command	Verify reception of TM (1,1)			
4.8		Verify that CDMS is receiving science packets from both SPU by looking at the TM log window: packets with APID 0x488 and 0x48A should appear after few seconds			
4.9	Send SPUS_STOP_REDUCT.COMPR command	Verify reception of TM (1,1)			
4.10	Send SPUL_STOP_REDUCT.COMPR command	Verify reception of TM (1,1)			
4.11		Verify that CDMS is no longer receiving science packets from any SPU			
Test Conductor:		PA:	Date:		



### 2.10 Test of the Memory Management Service (second part)

It is now possible to test this service for the subsystems. Memory commands for subsystems are received by DPU which then sends them to the emulators. The requirements OBS-FSR-MM10, OBS-FSR-MM11, OBS-FSR-MM12 and OBS-OPR-SR5 are covered.

Note that subsystem simulators are not intended to implement this service: all the commands make the DMC simulator crash and for this reason commands are sent to SPU simulator only. It accepts all the memory commands but the output is meaningless so that the only real test for this procedure is that DPU sends the command in a correct way (no NACK from the subsystems).

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1.1	Send DPU_MEMORY_LOAD_DRAM command with Repeater Value: 1 and DPU_MEMORY_BLOCK_ID = 0x5100 DPU_MEMORY_ADDRESS = 0 DPU_4_BYTES_WORDS_DATA = 0 DPU_CHECKSUM = 0x84C0	With the Telemetry Desktop showing both window PA000380 and PA023380 verify the reception of TM(1,1) and TM(1,7), and that DP_COM_SPUS is incremented			
1.2	Send DPU_MEMORY_LOAD_DRAM command with Repeater Value: 1 and DPU_MEMORY_BLOCK_ID = 0x7100 DPU_MEMORY_ADDRESS = 0 DPU_4_BYTES_WORDS_DATA = 0 DPU_CHECKSUM = 0x84C0	Verify the reception of TM(1,1) and TM(1,7), and that DP_COM_SPUL is incremented			
2.1	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x4100 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0	Verify the reception of TM (1,1) and that DP_COM_SPUS is incremented			
2.2	Send DPU_MEMORY_DUMP command with DPU_MEMORY_BLOCK_ID = 0x6100 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0	Verify the reception of TM (1,1) and that DP_COM_SPUL is incremented			

Test Conductor:	PA:	Date:
-----------------	-----	-------



3.1	Send DPU_MEMORY_CHECK command with DPU_MEMORY_BLOCK_ID = 0x4100 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0	Verify the reception of TM (1,1) and that DP_COM_SPUS is incremented			
3.2	Send DPU_MEMORY_CHECK command with DPU_MEMORY_BLOCK_ID = 0x6100 DPU_DATA_LENGTH = 1 DPU_MEMORY_ADDRESS = 0	Verify the reception of TM (1,1) and that DP_COM_SPUL is incremented			

### 2.11 Test of the On Board Control Procedure Service

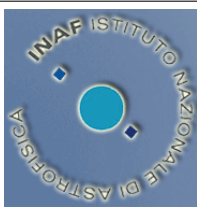
All the functionalities related to On Board Control Procedure Service are tested. The following requirements are covered: OBS-FSR-TC15; OBS-FSR-CP2 to OBS-FSR-CP4, OBS-FSR-CP6 to OBS-FSR-CP12.

The occurrence of a TM(1,8) (SEQ NOT Compl) after step #3.1 is considered nominal, it is due to a real-time problem of the DMC simulator.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_REPORT_LIST_OBCP command	On VPD Display Window select Type 18 and Subtype 9; verify NUMBER_OF_TIMES = 35 OBCP_ID = (names of the 35 procedures)			
2	Send DPU_REPORT_ACTIVE_OBCP command	On VPD Display Window select Type 18 and Subtype 11; verify NUMBER_OF_TIMES = 0			

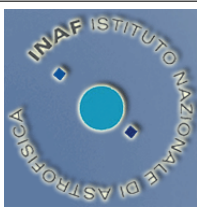
Test Conductor:	PA:	Date:
-----------------	-----	-------





3.1	Send DPU_START_OBCP command with Repeater Value: 5 and PROCEDURE_ID = 7 PAR_ID = 1 PAR_VALUE = 3 PAR_ID = 2 PAR_VALUE = 8000 PAR_ID = 3 PAR_VALUE = 10 PAR_ID = 4 PAR_VALUE = 16 PAR_ID = 5 PAR_VALUE = 16 Wait until OBCP is completed	Verify reception of TM (1,1) and TM (1,3); on window PA000380 verify that DP_STATUS is xx1xxxxxb and DP_WHICH_OBCP is STAR-ING_PHOT			
3.2	Wait until OBCP is completed	On SCOS2000 monitor verify that DP_STATUS is xx0xxxxxb and DP_WHICH_OBCP is NO_OBCP			
3.3	Increase the time during which the procedure is active. Send DPU_START_OBCP command with Repeater Value: 1 and PROCEDURE_ID = 7 PAR_ID = 2 PAR_VALUE = 60000	Verify reception of TM (1,1) and TM (1,3); on window PA000380 verify that DP_STATUS is xx1xxxxxb and DP_WHICH_OBCP is STAR-ING_PHOT			
3.4	Send DPU_REPORT_ACTIVE_OBCP command	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 11, verify NUMBER_OF_TIMES = 1 PROCEDURE_ID = STARING_PHOT			
3.5	Wait until OBCP is completed	On SCOS2000 monitor verify that DP_STATUS is xx0xxxxxb and DP_WHICH_OBCP is NO_OBCP			

Test Conductor:	PA:	Date:
-----------------	-----	-------



3.6	Send DPU_START_OBCP command with Repeater Value: 0 and PROCEDURE_ID = 50	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Start DEL OBCP INVALID_VALUE = 50			
3.7	Send DPU_START_OBCP command with Repeater Value: 0 and PROCEDURE_ID = 7	Verify reception of TM (1,1) and TM (1,3); on SCOS2000 monitor verify that DP_STATUS is xx1xxxxxb and DP_WHICH_OBCP is STAR-ING_PHOT			
3.8	Send DPU_START_OBCP command with Repeater Value: 0 and PROCEDURE_ID = 7	Verify reception of TM (1,1). Nothing else happens because this command is ignored if the requested OBCP is already running			
3.9	Send DPU_START_OBCP command with Repeater Value: 0 and PROCEDURE_ID = 19	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Running OBCP INVALID_VALUE = 7			
3.10	Wait until OBCP is completed	On SCOS2000 monitor verify that DP_STATUS is xx0xxxxxb and DP_WHICH_OBCP is NO_OBCP			

Test Conductor:	PA:	Date:
-----------------	-----	-------



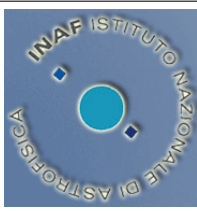
<p>3.1.1</p> <p>Send DPU_START_OBCP command with Repeater Value: 6 and PROCEDURE_ID = 7 PAR_ID = 1 PAR_VALUE = 3 PAR_ID = 2 PAR_VALUE = 8000 PAR_ID = 3 PAR_VALUE = 10 PAR_ID = 4 PAR_VALUE = 16 PAR_ID = 5 PAR_VALUE = 16 PAR_ID = 6 PAR_VALUE = 1</p>	<p>Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Too much PARAM INVALID_VALUE = 6</p>	
<p>4</p> <p>Send DPU_REPORT_STATUS_OBCP command with PROCEDURE_ID = 7</p>	<p>Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 13, verify PROCEDURE_ID = STARING_PHOT STEP_ID = 0 OBCP_STATUS = STOPPED NUMBER_OF_TIMES = 5 PAR_ID = 1, 2, 3, 4, 5 PAR_VALUE = 3, 60000, 10, 16, 16</p>	
<p>5.1</p> <p>Send DPU_COM_PARAM_TO_OBCP command with Repeater Value: 1 and PROCEDURE_ID = 7 PAR_ID = 2 PAR_VALUE = 80000</p>	<p>Verify reception of TM (1,1)</p>	

<p>Test Conductor:</p>	<p>PA:</p>	<p>Date:</p>
------------------------	------------	--------------



5.2	Send DPU_REPORT_STATUS_OBCP command with PROCEDURE_ID = 7	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 13, verify PROCEDURE_ID = STARING_PHOT STEP_ID = 0 OBCP_STATUS = STOPPED NUMBER_OF_TIMES = 5 PAR_ID = 1, 2, 3, 4, 5 PAR_VALUE = 3, 80000, 10, 16, 16			
5.3	Send DPU_COM_PARAM_TO_OBCP command with Repeater Value: 1 and PROCEDURE_ID = 7 PAR_ID = 6 PAR_VALUE = 1	Verify reception of TM (1,8) with EXE_FAIL_CODE = Invalid DATA FAILURE_REASON = Illegal PAR_ID INVALID_VALUE = 6			
6.1	Send DPU_START_OBCP command with Repeater Value: 1 and PROCEDURE_ID = 7 PAR_ID = 2 PAR_VALUE = 600000	Verify reception of TM (1,1) and TM (1,3)			
6.2	Send DPU_REPORT_ACTIVE_OBCP command	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 11, verify NUMBER_OF_TIMES = 1 OBCP_ID = STARING_PHOT			
6.3	Send DPU_SUSPEND_OBCP command with PROCEDURE_ID = 7 STEP_ID = 0	Verify reception of TM (1,1)			

Test Conductor:	PA:	Date:
-----------------	-----	-------



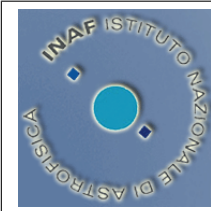
6.4	Send DPU_REPORT_STATUS_OBCP command with PROCEDURE_ID = 7	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 13, verify PROCEDURE_ID = STARING_PHOT STEP_ID = 0 OBCP_STATUS = SUSPENDED NUMBER_OF_TIMES = 5 PAR_ID = 1, 2, 3, 4, 5 PAR_VALUE = 3, 600000, 10, 16, 16			
7.1	Send DPU_RESUME_OBCP with PROCEDURE_ID = 7	Verify reception of TM (1,1)			
7.2	Send DPU_REPORT_STATUS_OBCP command with PROCEDURE_ID = 7	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 13, verify PROCEDURE_ID = STARING_PHOT STEP_ID = 0 OBCP_STATUS = RUNNING NUMBER_OF_TIMES = 5 PAR_ID = 1, 2, 3, 4, 5 PAR_VALUE = 3, 600000, 10, 16, 16			
8.1	Send DPU_STOP_OBCP command with PROCEDURE_ID = 7	Verify reception of TM (1,1)			
8.2	Send DPU_REPORT_STATUS_OBCP command with PROCEDURE_ID = 7	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 13, verify PROCEDURE_ID = STARING_PHOT STEP_ID = 0 OBCP_STATUS = STOPPED NUMBER_OF_TIMES = 5 PAR_ID = 1, 2, 3, 4, 5 PAR_VALUE = 3, 600000, 10, 16, 16			

Test Conductor:	PA:	Date:
-----------------	-----	-------



9.1	Send DPU_REPORT_LIST_OBCP command	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 9, verify NUMBER_OF_TIMES = 35 OBCP_ID = (names of the 35 procedures)			
9.2	Send DPU_DELETE_OBCP command with PROCEDURE_ID = 34	Verify reception of TM (1,1)			
9.3	Send DPU_REPORT_LIST_OBCP command	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 9, verify NUMBER_OF_TIMES = 34 OBCP_ID = (names of the 34 procedures)			
9.4	Send DPU_REPORT_STATUS_OBCP command with PROCEDURE_ID = 34	Verify reception of TM (1,1) and on VPD Display Window select Type 18 and Subtype 13, verify PROCEDURE_ID = ACWE STEP_ID = 0 OBCP_STATUS = NOT EXISTING NUMBER_OF_TIMES = 0			
9.5	Send DPU_START_OBCP command with Repeater Value: 0 and PROCEDURE_ID = 34	Verify reception of TM (1,8) with EXE_FAIL_CODE = Illegal STATUS FAILURE_REASON = Start DEL OBCP INVALID_VALUE = 34			

Test Conductor:	PA:	Date:
-----------------	-----	-------



### 2.12 Test of the Housekeeping report (second part)

The following requirements are now verified: OBS-PRF-SR4; OBS-OPR-SR8, OBS-OPR-SR10 and OBS-OPR-SR11.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 4 SPU_ARRAY = 1	With the Telemetry Desktop showing both windows PA000380 and PA026380 verify the reception of TM (1,1) and that DP_TM_RATE is NO PRIME; new packets arrive every 2 sec. Verify that DM_DPU_SEND_PAC (in window PA019420), SPL_CI (in window PA003390) and SPS_CI (in window PA004400) are incremented every new packet			
2	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 1 SPU_ARRAY = 1	Verify the reception of TM (1,1) and that DP_TM_RATE is SPEC; new packets arrive every 2 sec. Verify that DM_DPU_SEND_PAC, SPL_CI and SPS_CI are incremented every new packet			
3	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 2 SPU_ARRAY = 1	Verify the reception of TM (1,1) and that DP_TM_RATE is PHOT; new packets arrive every 2 sec. Verify that DM_DPU_SEND_PAC, SPL_CI and SPS_CI are incremented every new packet			

Test Conductor:	PA:	Date:
-----------------	-----	-------



### 2.13 Test of the Packet Transmission Control Service and Telemetry Rate

In this test procedure we make use of OBCP 29 that generates dummy science packets. Two parameters are required: the first is the duration of the OBCP in seconds, the other parameter sets the rate in terms of number of packets per second. This test requires that DPU is in SPEC mode and that the SPU simulators, if active, are not sending science packets. The first two steps of the procedure perform these operations.

The nominal rate of 100kbps corresponds to 12.5 full packets (1024 bytes each). This rate has to take into account also HK packets, TC verification packets (in principle 2 per second) and possible event packets. For simplicity in the test procedure we will generate 13 science packets per second. This rate is slightly larger than required but of course does not invalidate the result. The duration of the test is 1 minute.

For the burst mode the rate is 300 kbps with a maximum of 40 TM packets allocated, with the same limitations of the nominal mode. The test is run again for 1 minute with a rate of 38 packets per second.

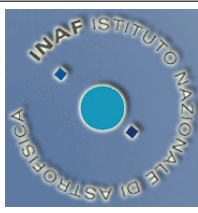
Note that the CDMS simulator is considered a development tool for which a full performance test of the supported telemetry rate is not foreseen. The 1 minute test is then long enough since a longer test would not increase the significance of the result.

The following requirements are covered: OBS-FSR-TC5, OBS-FSR-TC15, OBS-FSR-TC16; OBS-FSR-PT1 to OBS-FSR-PT4; OBS-PRF-SR5, OBS-PRF-SR6; OBS-OPR-SR13 (the packets that can not be disabled, see OBS-FSR-PT3, are (1,1), (1,2)).

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Send DPU_SET_HK_LIST command with HK_PACKET_ID = 1 SPU_ARRAY = 1	Verify reception of TM (1,1)			
2	Send SPUS_STOP_REDUCT_COMPR command	Verify reception of TM (1,1)			
3	Send SPUL_STOP_REDUCT_COMPR command	Verify reception of TM (1,1)			
4	Send DPU_REPORT_ENABLED_PACKET command	Verify reception of TM (1,1) and on VPD Displayer of TM (14,4) with NUMBER_OF_TIMES = 45			
5	Send DPU_DISABLE_PACKET_TRAN command with Repeater Value: 1 and PACKET_TYPE = 3 PACKET_SUBTYPE = 25 DPU_STRUCTURE_ID = 1	Verify reception of TM (1,1); verify that the OBS sends only the additional HK packets at a rate of 0.1 Hz			

Test Conductor:	PA:	Date:
-----------------	-----	-------



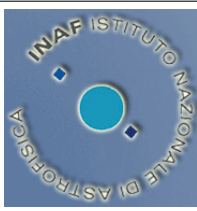


6	Send DPU_REPORT_ENABLED_PACKET command	Verify reception of TM (1,1) and on VPD Displayer of TM (14,4) with NUMBER_OF_TIMES = 44			
7	Send DPU_ENABLE_PACKET_TRAN command with Repeater Value: 1 and PACKET_TYPE = 3 PACKT_STRUCTURE_ID = 1	Verify reception of TM (1,1); verify that the OBS is sending again nominal and additional HK packets			
8	Send DPU_REPORT_ENABLED_PACKET command	Verify reception of TM (1,1) and on VPD Displayer of TM (14,4) with NUMBER_OF_TIMES = 45			
9	Send DPU_DISABLE_PACKET_TRAN command with Repeater Value: 1 and PACKET_TYPE = 1 PACKT_STRUCTURE_ID = 0	Verify reception of TM (1,1)			
10	Send DPU_DISABLE_PACKET_TRAN command with Repeater Value: 1 and PACKET_TYPE = 3 PACKT_STRUCTURE_ID = 4	Verify reception of TM (1,1) which means that the command at previous step (disable transmission of TM (1,1)) did not have any effect; wait some seconds (10 at most) to verify that the additional HK packet (SID 4) is no longer sent			
11	Send DPU_DISABLE_PACKET_TRAN command with Repeater Value: 1 and PACKET_TYPE = 1 PACKT_STRUCTURE_ID = 0	Verify reception of TM (1,1)			
12	Send TEST_LENGTH command (a wrong command, see step #3 of procedure 2.2)	Verify reception of TM (1,2) with ACP_FAIL_CODE Invalid LENGTH ACP_INVALID_VAL 17 AC_EXPECT_VALUE 16 which means that the command at previous step (disable transmission of TM (1,2)) did not have any effect			

Test Conductor:	PA:	Date:
-----------------	-----	-------



13.1	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID = 29 PAR_ID = 1 PAR_VALUE = 60 PAR_ID = 2 PAR_VALUE = 13	Verify reception of TM (1,1) and TM (1,3). Science packets appear in the log window of the CDMS simulator. After 1 minute TM (1,7) is reported and the science packets are no longer delivered			
13.2	Open the TM log file of the CDMS, its name is reported in a little window in the right bottom corner and can be either TelemetryA.txt or TelemetryB.txt	Verify that $13 \times 60 = 780$ science packets are in. Check also in any of the science packet that the length is 0x3F9 or 1017 (which corresponds to a maximum packet length of 1024 bytes)			
14.1	Select in the CDMS simulator the buslist PACS_Burst				
14.2	Send DPU_SET_BUSLIST command with ENABLE_DISABLE set to 1	Verify reception of TM (1,1); verify that DPU_STATUS is xxx1xxxxxb. Now DPU and CDMS are in burst mode			
14.3	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID = 29 PAR_ID = 1 PAR_VALUE = 60 PAR_ID = 2 PAR_VALUE = 38	Verify reception of TM (1,1) and TM (1,3). Science packets appear in the log window of the CDMS simulator. After 1 minute TM (1,7) is reported and the science packets are no longer delivered			
14.4	Open the TM log file	Verify that $38 \times 60 = 2280$ science packets are in			
14.5	Send DPU_SET_BUSLIST command with ENABLE_DISABLE set to 0	Verify reception of TM (1,1); verify that DPU_STATUS is xxx0xxxxxb			
14.6	Select in the CDMS simulator the buslist PACS_Nominal. Now DPU and CDMS are back in nominal mode				
Test Conductor:		PA:	Date:		



15.1	Send DPU_START_OBCP command with Repeater Value: 2 and PROCEDURE_ID = 29 PAR_ID = 1 PAR_VALUE = 60 PAR_ID = 2 PAR_VALUE = 38	Verify reception of TM (1,1) and TM (1,3). Science packets appear in the log window of the CDMS simulator. After 1 minute TM (1,7) is reported but the science packets are still delivered for a while			
15.2	Open the TM log file	Verify that the number of received packets is less than $38 \times 60 = 2280$ because the nominal bus list does not support the burst telemetry rate. Verify that an event has been sent with EVENT_ID= BUFFER FULL and EVENT_SID= 1. Verify that the number of packets received by CDMS plus the new value of the HK DP_GEN_TM_LOST is 2280			

## 2.14 Last testing activities

The remaining requirements are now tested: OBS-OPR-SR9 and OBS-OPR-SR20.

#	Action	Expected Reaction	Observed Reaction	Time	Notes
1	Kill the application window running DEC simulator	Verify in SCOS2000 display that DP_DMC_HK is NO NEW HK			
2	Send DPU_RESET_1355 command	Verify reception of TM (1,1); verify that all the three HK values DP_DMC_LINK, DP_SPS_LINK and DP_SPL_LINK are set to OFF; verify that an error is signaled in the PC running the simulators			
3	Send DPU_RESET command	Verify that the SW reset has been performed controlling that the counter of the HK packet has been reset			

Test Conductor:	PA:	Date:
-----------------	-----	-------



## A Matrix of compliance

In the following table all the requirements written in AD3 are reported, along with the test procedure and the step(s) inside the procedure that cover it. If the step is not reported this means that the whole procedure covers the requirement.

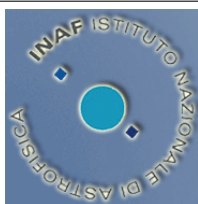
Requirement	Test procedure	Requirement	Test procedure
OBS-FSR-TC1	2.2 but in principle verified each time a telecommand is sent; see also the definition of struct TC_packet in LT_TMdef.h	OBS-FSR-TC2	2.5.1 step 18
OBS-FSR-TC3	Verified with each TC received	OBS-FSR-TC4	It can be verified with code inspection: see the definition of struct TM_packet in LT_TMdef.h
OBS-FSR-TC5	2.13 step 13.2	OBS-FSR-TC6	Verified with each TM packet sent
OBS-FSR-TC7	2.2, but also verified for each telecommand sent to DPU	OBS-FSR-TC8	2.2, but also verified with each TM(1,x) packet
OBS-FSR-TC9	2.2 step 2.2	OBS-FSR-TC10	2.2 step 3
OBS-FSR-TC11	This requirement can not be tested with SCOS2000	OBS-FSR-TC12	2.2 step 4
OBS-FSR-TC13	2.2 step 5	OBS-FSR-TC14	2.2 step 1, but also verified for each telecommand sent to DPU since the TM (1,1) or (1,2) are sent well within 2 seconds
OBS-FSR-TC15	2.8 all steps but the first two; 2.11 steps 3.1, 3.3, 3.7, 6.1; 2.13 steps 13.1, 14.3, 15.1	OBS-FSR-TC16	2.3 steps 1.4 and 4.1; 2.8 steps 3.1, 3.2, 3.3; 2.13 steps 13.1, 14.3, 15.1
OBS-FSR-TC17	2.3 steps 1.1, 1.2, 1.3, 1.5, 4.2; 2.5.1 steps 4, 8; 2.9 steps 1.4, 2.3, 3.3		
OBS-FSR-HK1	2.1 steps 3 and 11; 2.5.1 steps 1, 2, 3	OBS-FSR-HK2	2.1 step 3
OBS-FSR-HK3	2.1 steps 4 and 6	OBS-FSR-HK4	2.5.1 steps 1, 2, 3
OBS-FSR-HK5	2.5.1 steps 1, 2, 3	OBS-FSR-HK6	It can be verified with code inspection of the routine that generates HK packets: see T5_HKMON.c
OBS-FSR-EV1	2.9 steps 1.3, 2.2, 3.2	OBS-FSR-EV2	2.9 steps 1.3, 2.2, 3.2
OBS-FSR-EV3	2.3 step 4.2	OBS-FSR-EV4	There is one only event (5,4) in the present code; it has been verified with a specific setup (JTAG)
OBS-FSR-MM1	2.3 steps 1.4, 2.1, 3.1, 4.1	OBS-FSR-MM2	2.3 steps 1.2, 1.3, 1.4
OBS-FSR-MM3	2.3 steps 1.1, 1.2, 1.5	OBS-FSR-MM4	2.3 steps 2.1 and 2.2
OBS-FSR-MM5	2.3 steps 2.1 and 2.2	OBS-FSR-MM6	2.3 step 2.2
OBS-FSR-MM7	2.3 steps 3.1 and 3.2	OBS-FSR-MM8	2.3 steps 3.1 and 3.2
OBS-FSR-MM9	2.3 steps 3.2	OBS-FSR-MM10	2.10



Requirement	Test procedure	Requirement	Test procedure
OBS-FSR-MM11	2.10 steps 2.1 and 2.2	OBS-FSR-MM12	2.10 steps 3.1 and 3.2
OBS-FSR-FM1	2.5; 2.9 from step 1.1 to 3.5	OBS-FSR-FM2	2.5 step 1
OBS-FSR-FM3	2.5 step 2	OBS-FSR-FM4	2.9
OBS-FSR-FM5	2.5; 2.9 from step 1.1 to 3.5	OBS-FSR-FM6	2.9 steps 1.1, 1.2, 1.6, 2.1, 2.5, 3.1, 3.5
OBS-FSR-FM7	2.9 step 1.4, 2.3, 3.3	OBS-FSR-FM8	2.5 step 3
OBS-FSR-TM1	It can be verified with code inspection: see case <code>RT_MODE_SYNCHRONIZE</code> in <code>isr1553.c</code>	OBS-FSR-TM2	It can be verified with code inspection: see function <code>update_TM_buffer</code> in <code>LT_upTMb.c</code>
OBS-FSR-TM3	2.6 step 1	OBS-FSR-TM4	2.6 step 2
OBS-FSR-PT1	2.13 step 7	OBS-FSR-PT2	2.13 step 5
OBS-FSR-PT3	2.13 steps 9, 10, 11, 12	OBS-FSR-PT4	2.13 steps 4, 6, 8
OBS-FSR-PT5	2.13 steps 4, 6, 8		
OBS-FSR-TS1	2.7 step 1	OBS-FSR-TS2	2.7 step 1
OBS-FSR-CP1	This requirement has been tested with an ad hoc procedure	OBS-FSR-CP2	2.11 steps 9.1, 9.2, 9.3
OBS-FSR-CP3	2.11 steps 3.1, 3.3, 3.6, 3.7, 3.8, 3.9, 3.11, 6.1, 9.5	OBS-FSR-CP4	2.11 steps 3.1, 3.3, 3.6, 3.7, 3.8, 3.9, 3.11, 6.1, 9.5
OBS-FSR-CP5	2.8 all the steps but the first two	OBS-FSR-CP6	2.11 steps 8.1 and 8.2
OBS-FSR-CP7	2.11 steps 6.3 and 6.4	OBS-FSR-CP8	2.11 steps 7.1 and 7.2
OBS-FSR-CP9	2.11 steps 5.1, 5.2, 5.3	OBS-FSR-CP10	2.11 steps 1, 9.1, 9.3
OBS-FSR-CP11	2.11 steps 2, 3.4, 6.2	OBS-FSR-CP12	2.11 steps 4, 5.2, 6.4, 7.2, 8.2, 9.4
OBS-FSR-MC1	It can be verified with code inspection: see the functions <code>from_1DM_to_2DM</code> and <code>from_2DM_to_1DM</code> in <code>MM_21020.s</code>	OBS-FSR-MC2	It can be verified with code inspection: see the functions <code>from_1DM_to_2DM</code> and <code>from_2DM_to_1DM</code> in <code>MM_21020.s</code>
OBS-FSR-MC3	In all TM packets DPU uses <code>unsigned int</code> which corresponds to PTC 3, PFC 12 (for 16 bits) or PTC 3, PFC 14 (for 32 bits)	OBS-FSR-MC4	It can be verified with code inspection: see <code>TC_acceptance</code> function in <code>L4_LIB.c</code>
OBS-FSR-MC5	It can be verified with code inspection. All the TM packets used in the OBS are listed in the <code>Tm_packet_enabled</code> array defined in <code>init.h</code>	OBS-FSR-MC6	It can be verified with code inspection. All the APIDs used in the OBS are statically stored in the function <code>get_APID</code> in <code>LT_upTMb.c</code>
OBS-FSR-MC7	In case one TM packet has a wrong checksum SCOS2000 immediately reports the problem		



Requirement	Test procedure	Requirement	Test procedure
OBS-PRF-SR1	Verified with an external tool	OBS-PRF-SR2	It can be verified with code inspection. The TC_acceptance function is called just after the reception of a TC, see T4CNTRLR.c
OBS-PRF-SR3	See previous requirement	OBS-PRF-SR4	2.1 step 4; 2.12
OBS-PRF-SR5	2.13 steps 13.1 and 13.2	OBS-PRF-SR6	2.13 steps 15.1 and 15.2
OBS-PRF-SR7	Two independent tasks are used to packetize science data: T7_SPSRX and T8_SPLRX	OBS-PRF-SR8	The current dimension is less than 0x1505A PM words
OBS-IRF-SR1 to OBS-IRF-SR76	The whole set of these requirements has been tested at ALENIA and ASTRUM premises. Moreover, many of them are verified for every received telecommand packet and for every sent telemetry packet		
OBS-IRF-SR77	2.4 step 1	OBS-IRF-SR78	2.4 step 1
OBS-IRF-SR79	2.4 steps 2.1, 2.2, 2.3	OBS-IRF-SR80	2.4 steps 3.1, 3.2, 3.3
OBS-IRF-SR81	2.4 steps 3.1, 3.2, 3.3	OBS-IRF-SR82	Transparent mode is the default in the SMCS chip
OBS-IRF-SR83	2.8 steps 3.1, 3.2, 3.3		
OBS-OPR-SR1	The transition from one mode to the other is performed with specific OBCP that can be tested at instrument level	OBS-OPR-SR2	See previous requirement
OBS-OPR-SR3	See OBS-OPR-CP1	OBS-OPR-SR4	No TC can be executed until the previous one has been completed. See T4CNTRLR.c
OBS-OPR-SR5	2.9; 2.10	OBS-OPR-SR6	For DEC sequences 2.5.1 steps 14, 14.1, 15, 15.1, 16, 17, 17.1
OBS-OPR-SR7	2.9 steps 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8	OBS-OPR-SR8	2.12
OBS-OPR-SR9	2.14 step 1	OBS-OPR-SR10	2.9 steps 1.1 and 1.2
OBS-OPR-SR11	It can be verified with code inspection, see T5_HKMON.c	OBS-OPR-SR12	This requirement can be verified with code inspection of the routine event_packet in T5_HKMON.c
OBS-OPR-SR13	The buffers dimension is defined in the file 1553_def.h: #define EV_NUM 32 #define HK_NUM 64 #define SC_NUM 400	OBS-OPR-SR14	The DPU internal time is initialized in the file T1_INIT.c. It is possible that the first two HK packets are generated before the DPU received the first synch from the CDMS, in this case the time stamp of the packets have the MSb set to 1



Requirement	Test procedure	Requirement	Test procedure
OBS-OPR-SR15	This requirement can be verified with inspection of the HK monitoring task code in T5_HKMON.c	OBS-OPR-SR16	2.1 step 7
OBS-OPR-SR17	This requirement is actually on the boot SW	OBS-OPR-SR18	2.1 step 10
OBS-OPR-SR19	This requirement can be verified with inspection of the architecture file pacs.ach	OBS-OPR-SR20	2.14 step 3
OBS-OPR-SR21	For the DPU this is just a command sent to DEC		
OBS-SEC-SR1	This requirement can be verified with inspection of the HK monitoring task code in T5_HKMON.c	OBS-SEC-SR2	2.5.1 step 6
OBS-SEC-SR3	2.5.1 step 6	OBS-SEC-SR4	Autonomy functions are started inside T5_HKMON task which runs every two seconds, so it is impossible that the same function can be started after less than 2 seconds
OBS-SEC-SR5	2.5.1 steps 4, 5, 6, 7, 8	OBS-SEC-SR6	The list of HK to be written in each HK packet is defined in init.h and reported in RD3
OBS-SEC-SR7	2.3 steps 4.1 and 4.2	OBS-SEC-SR8	This requirement can only be verified with a specific setup using the JTAG
OBS-QTY-SR1	This requirement can be verified with code inspection	OBS-QTY-SR2	This requirement can be verified with code inspection



## B TC—TM reference table

Here is a table that shows the combinations (type,subtype) of each TC accepted by the DPU and the telemetry (TM) packets associated. This list is a subset of all the services allowed in RD2. If a TM packet is not associated to a TC packet this means that it is generated autonomously by the DPU. TM (1,1) and TM (1,2) are not reported.

Name	TC	TM	Report on completion	
			Success	Failure
HK reports	—	(3,25)		
Event report	—	(5,1)		
Exception report	—	(5,2)		
Error report	—	(5,4)		
Memory load	(6,2)	—	(1,7)	(1,8)
Memory dump	(6,5)		(6,6)	(1,8)
Memory check	(6,9)		(6,10)	(1,8)
Start function	(8,1)			
Stop function	(8,2)			
Perform activity	(8,4)		(1,7) <sup>2</sup>	(1,8)
Status of function	(8,5)			
Time verification	(9,7)		(9,9)	
Enable TM packet	(14,1)			
Disable TM packet	(14,2)			
Report TM packet	(14,3)		(14,4)	
Test connection	(17,1)		(17,2)	
Load OBCP	(18,1)		(1,7)	(1,8)
Delete OBCP	(18,2)			(1,8)
Start OBCP	(18,3)	(1,3) <sup>3</sup>	(1,7)	(1,8)
Stop OBCP	(18,4)			(1,8)
Suspend OBCP	(18,5)			(1,8)
Resume OBCP	(18,6)			(1,8)
Communicate parameters	(18,7)			(1,8)
List of OBCP	(18,8)		(18,9)	
List of active OBCP	(18,10)		(18,11)	(1,8)
OBCP status	(18,12)		(18,13)	(1,8)
Science data A	—	(21,1)		
Science data B	—	(21,2)		
Science diagnostic data	—	(21,3)		

<sup>2</sup>Only for particular DPU commands.

<sup>3</sup>Sent just before start of execution.