HERSCHEL

SPIRE On-Board Software Verification and Validation Plan/Acceptance Test Plan

Document Ref.: SPIRE-IFS-DOC-001392

Issue: 1.0

Prepared by: Sergio Molinari

Distribution List:

RAL	**K. King**
	B. Swinyard
	S. "Duke" Sidher
IFSI	**R. Cerulli** **R. Orfei**
	J. Liù

1 Introduction

1.1 Purpose of the document

This document presents the test plan and procedures for the verification and validation of the On-Board Software of the SPIRE instrument at the unit, integration and system level. This test plan deals with all SPIRE OBS components as specified in AD2, except for the Handler of the interface to the Spacecraft CDMS, which is tested under a separate plan (RD1). A subset of this plan will constitute the SPIRE OBS acceptance test plan.

1.2 Acronyms and Glossary

AVM	Avionic Model
BC	Bus Controller
BP	BreakPoint
CDMS	Command and Data Management System
DM	Data Memory (DSP)
DPU	Digital Processing Unit
DSP	Digital Signal Processor
DTST	Dedicated Test Software Tools
EGSE	Electrical Ground Support Equipment
ESA	European Space Agency
HERSCHEL	Herschel Space Observatory
HK	Housekeeping
HW	Hardware
ICE	DSP In-Circuit Emulator
I/F	Interface
IFSI	Istituto di Fisica dello Spazio Interplanetario
NA	Not Applicable
OBS	On-Board Software
PM	Program Memory (DSP)
RAM	Random Access Memory
S/C	Spacecraft
S/S	Subsystem
SUT	Software Under Test
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TC	Telecommand
TM	Telemetry
VME	Virtual Machine Executable Code

1.3 Document List

1.3.1 Applicable Documents

Document Reference	Name	Number/version/date
AD1	SPIRE OBS User Requirements Document	
AD2	SPIRE OBS Software Specifications Document	
AD3	Packet Structure Interface Control Document (PSICD)	SCI-PT-ICD-7527 Issue 2.0
AD4	Herschel/Planck Instrument Data Rates	H-P-1-ASPI-TN-0204 Issue: 1

1.3.2 Reference Documents

Document Reference	Name	Number/version
RD1	DPU/ICU Spacecraft Interface Test Plan	
RD2	SPIRE Data ICD	
RD3	DRCU/DPU ICD	Sap-SPIRE-CCa-076-02 v0.7
RD4	Virtual Machine Compiler and Simulator	
RD5	MCU/SCU Command List	

2 Test Plan

2.1 Test Items

We identify Test Items at the unit, integration and system level. All test items will be tested in the software verification and validation phase, while only a set of items at the system level will be tested in the acceptance phase.

2.1.1 Unit level

At the unit level we identify a test item as a routine, or a group of routines, that perform a specific and self-contained function. A list is given below:

TIUL1. TC reception from the CDMS, verification and generation of the relative acceptance report.
TIUL2. Dispatch of TM packets to the CDMS
TIUL3. Identification and execution of DPU commands.
TIUL4. Transmission of commands to the S/Ss via the Low-Speed link.
TIUL5. Reception of S/S HK parameters via the Low-Speed link.
TIUL6. Reception of Science Frames from S/Ss.
TIUL7. Autonomy Functions.
TIUL8. Temperature Controls.
TIUL9. Event generation (including execution reports).
TIUL10. Virtual Machine (execution of command lists).

2.1.2 Integration level

At the integration level we identify a test item as a specific task; a task handles different functions. A list is given below:

TIIL1. Command sequencing.
TIIL2. Request, reception and packing of HK parameters.
TIIL3. Reception and packing of Science data.
TIIL4. HK monitoring.

2.1.3 System level

At the system level we have a set of test items that deal with the correct inter-task communication (Data & Controls flow). Specifically:

TISL1. TMTC ←→ CMD_SEQ
TISL2. CMD_SEQ ←→ LS
TISL3. HK_ASK ←→ LS
TISL4. HK_ASK ←→ TMTC
TISL5. HS ←→ TMTC
TISL6. HK_MONITOR ←→ Autonomy_Fnct
TISL7. Autonomy_Fnct ←→ LS
TISL8. PID_Controls ←→ LS

At the system level we also identify as a test item the ability to perform the following services specified in AD3 and required from the OBS according to AD1. The following set of system level test covers the full list of services required by AD3:

TISL9. Telecommand Verification
TISL10. Housekeeping & Diagnostic Data Reporting
TISL11. Memory Management
TISL12. Function Management
TISL13. Event Reporting
TISL14. Packet Transmission Control
TISL15. Time management
TISL16. Science Data Transfer
TISL17. Test service

2.2 Test Deliverables

The items that will be delivered at the end of tests are:

1. Test procedures
2. Test report

2.3 Testing Tasks

These are the tasks needed to prepare and carry out the tests:

1. Preparation of a SPIRE specific MIB for SCOS2000 to be able to a) generate all TC packets needed for the OBS tests, and b) open and interpret HK and Event TM packets
2. Upload the compiled OBS to the DPU
3. Prepare TBD dedicated test SW tools (DTST) to open science TM packets
4. Execute the tests and compile the test report

2.4 Environmental Needs

The following equipment must be installed at IFSI in order for the tests to be carried out

1. DRCU SW simulator
2. EGSE (SCOS2000+Router+CDMS simulator)

It is planned that a small subset of the tests, identified in boldface in the test procedures (see 4), and which involve testing of the TFL protocol, be also executed with the ESA CDMS Testbed. Pass criteria in this case will simply be the correct reception of TC packets by the DPU (verifying that the DPU reacts as expected), and the correct dispatch of TM packets (verifying that SCOS2000 accepts and, in case of events and HK packets, correctly interprets the packets). The functionality required for this Testbed is the ability to run pre-defined 1553 bus profiles according to AD3 and AD4.

For these tests, it is required that the following equipment is available at the test site:

1) ESA CDMS Testbed
2) DPU
3) OBS development system (hosting a licensed copy of the VIRTUOSO operating system)

The ESA CDMS Testbed should provide these functionalities:

- Run pre-defined buslists, handling the exchange of TM and TC packets
- Be able to switch automatically between normal/burst modes buslists
- Be able to handle the "retry-at-packet-level"

Based on these needs, the instruments required that the ESA CDMS Testbed is made available in a suitcase for shipping to the test site. In case, following the tests with the ESA CDMS Testbed, it is necessary to debug and update the OBS, the instruments anticipate that they will not be able to support such activities on any site other than IFSI.

2.5 Test case pass/fail criteria

Test criteria are based on the direct inspection Science, Event and HK TM Packets received by the EGSE. DTST will be used to inspect Science TM packets which SCOS2000 does not open. In case a test item has to be verified before the transmission of a TM packet, the evaluation criteria will be based on the direct inspection of the DSP DM.

3 Test case specifications

3.1 DPU_COMMAND_EXEC

The purpose is to demonstrate the link S/C-DPU by verifying the:

a) Reception, validation and interpretation of TCs
b) Command identification and execution
c) Verification reporting

3.1.1 Test Items

TIUL1, TIUL2, TIUL3, TIUL9, TISL1, TISL9, TISL15p, TISL17

3.1.2 Input specifications

The input to this test case is a set of TCs built according to AD3, and requiring specific functions to be performed by the DPU. For this first test case this set shall be limited to self-contained commands that do not affect units not tested in this test case. Some of the TCs will contain invalid fields (e.g, APID etc.); if SCOS2000 is unable to send invalid packets, those packets will have to available as HEX text files in the CDMS simulator. The set of TCs is specified below:

TC Code	Description
TC17.1.1	*Perform Connection Test* standard TC
TC17.1.2	Same as TC17.1.1, but with an incorrect APID of 0x400
TC17.1.3	Same as TC17.1.1, but with an incorrect packet length of 0xA
TC17.1.4	Same as TC17.1.1, but with an incorrect checksum of 0x1111
TC17.1.5	Same as TC17.1.1, but with an incorrect packet type of 0x1
TC17.1.6	Same as TC17.1.1, but with an incorrect packet subtype of 0xA
TC9.7.1	*Enable Time Verification* standard TC
TC14.3.1	*Report Enabled Telemetry Packets* standard TC

3.1.3 Output specifications

The output for this test case will consist in TM packets normally expected for the input TCs.

3.1.4 Environmental needs

The required set of input TCs will reside in SCOS2000 or on the CDMS Simulator. At the OBS start-up, packets generation will be enabled for all APIDs.

3.2 DPU_MEM

The purpose is to demonstrate the ability to load, check and dump memory areas resident on the DPU. This will be done by absolute (via Service 6) and relative (via dedicated functions with Service 8) addresses in memory.

3.2.1 Test Items

TIUL2, TISL11, TISL12p. Test item TIUL2 is verified again here because SCIENCE packets (like the memory dump TM packets) are managed via a different memory pool with respect to EVENT packets (like the TC verification reports, see 3.1.1).

3.2.2 Input specifications

A set of TCs will be available:

TC Code	Description
TC6.2.1	*Memory Load* standard TC. Application data is structured according to RD2 with the following parameter values: • Memory_ID = 1 (DM) • Start_Address = 0x2FF00 • Length = 0x64 • 100 data words all = 0xA5A5
TC6.2.2	Same as TC6.2.1, but an incorrect Memory_ID = 4
TC6.2.3	Same as TC6.2.1, but an incorrect Start_Address = 0x80000
TC6.2.4	Same as TC6.2.1, but with Start_Address = 0x7FFF0, length = 0x10 and 32 data words
TC6.2.5	Same as TC6.2.1, but with an incorrect number of 20 data words
TC6.2.6	Same as TC6.2.1, but an incorrect Application Data CRC of 0x1111
TC6.5.1	*Memory Dump* standard TC. Application data is structured according to RD2 with the following parameter values: • Memory_ID = 1 (DM) • Start_Address = 0x2FF00 • Length = 0x64
TC6.9.1	*Memory Check* standard TC. Application data will be as in RD2
TC8.4.1-1.1	*Set Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x01 • Table_ID = 0x40 • Length = 0x32 (in units of 32-bit words)
TC8.4.1-1.2	Same as TC8.4.1-1.1, but with Function_ID of 0xE0
TC8.4.1-1.3	Same as TC8.4.1-1.1, but with Activity_ID of 0xA
TC8.4.1-1.4	Same as TC8.4.1-1.1, but with SID of 0x902
TC8.4.1-1.5	Same as TC8.4.1-1.1, but with Table_ID of 0x200
TC8.4.1-1.6	Same as TC8.4.1-1.1, but with Length = 0x4E20
TC8.4.1-2.1	*Report Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x02 • Table_ID = 0x40 • Index = 0 • Length = 0x32
TC8.4.1-3.1	*Update Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters:

	• Function_ID = 0x01 • Activity_ID = 0x03 • Table_ID = 0x40 • Index = 0 • Length = 0x32 • 50 16-bit words with pattern 0x5A5A
TC8.4.1-3.2	Same as TC8.4.1-3.1, but with Table_ID = 0x30
TC8.4.1-3.3	Same as TC8.4.1-3.1, but with Index = 0x64
TC8.4.1-3.4	Same as TC8.4.1-3.1, but with Length = 0x40
TC8.4.1-3.5	Same as TC8.4.1-3.1, but with only 40 16-bit data words

3.2.3 Output specifications

The output will consist of the set of TM packets expected in response to input TCs.

3.2.4 Environmental needs

The required set of input TCs will reside in SCOS2000 or on the CDMS Simulator. At the OBS start-up, packets generation will be enabled for all APIDs.

3.3 HK_COLLECT

The purpose is to test the DPU-S/S chain by demonstrating the collection and transmission of HK packets. The ability to support the TM transmission retry at packet level will also be tested here.

3.3.1 Test Items

TIUL2, TIUL4, TIUL5, TIIL2, TISL3, TISL4, TISL10. Test item TIUL2 is verified again here because HK packets are managed via a different memory pool with respect to EVENT packets (like the TC verification reports, see 3.1.1) and SCIENCE packets (like memory dump packets, see 3.2.1).

3.3.2 Input specifications

A set of TCs will be available:

TC Code	Description
TC8.4.1-1.10	*Set Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x01 • Table_ID = 2 • Length = 0x14
TC8.4.1-1.11	Same as TC8.4.1-1.10 but with Table_ID = 3
TC8.4.1-3.10	*Update Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters:

	• Function_ID = 0x01 • Activity_ID = 0x03 • Table_ID = 2 • Index = 0 • Ndata = 0x14 (in units of 32-bit words) • 40 16-bit data words which will represent 20 HK collection commands (TBD)
TC8.4.1-3.11	Same as TC8.4.1-3.10 but with Table_ID = 3 and a different set of HK collection commands (TBD)
TC3.2.1	*Define New Diagnostic Parameter Report* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • HKPCKTID = 2 • Least significant octet of HKSID = 0x02 • HKINTERVAL = 100
TC3.2.2	Same as TC3.2.1, but with HKPCKTID = 3 and LSO of HKSID = 0x03
TC3.2.3	Same as TC3.2.1, but with HKPCKTID = 4
TC3.2.4	Same as TC3.2.1, but with HKINTERVAL = 5
TC3.2.5	Same as TC3.2.1, but with LSO of HKSID = 0x28
TC3.4.1	*Clear Diagnostic Parameter Report Definition* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • HKPCKTID = 2
TC3.4.2	Same as TC3.4.1, with HKPCKTID = 3
TC3.9.1	*Report Housekeeping Parameter Report Definition* standard TC
TC3.11.1	*Report Diagnostic Parameter Report Definition* standard TC
TCTest.1	*Perform Activity of Function* standard TC with function ID = 0xCB and activity ID = 0x01. This TC is used to force a wrong CRC to be attached to a TM packet being dispatched.

3.3.3 Output specifications

The output for this test case will consist in TM packets containing the HK data.

3.3.4 Environmental needs

The DRCU Simulator will be connected to the DPU. The structure of the HK packets will be defined in SCOS2000 so that the packets can be opened and checked. Alternatively, DTSTs will have to be used. At the OBS start-up, packets generation will be enabled for all APIDs, and the default HK and Diagnostic packet structure will be defined on-board. It is assumed that the DRCU simulator will conform to RD3 in its ability to identify and execute commands. The DRCU simulator will allow on-the-fly modification of any HK parameter, without having to stop and restart its software.

3.4 VM

The purpose is to demonstrate that all Virtual Machines available in the OBS as specified in AD2 can run in parallel without interfering with one another; this is a potential risk since all VMs use the same interface to send commands and receive parameters from the DRCU.

3.4.1 Test Items

TIUL10

3.4.2 Input Specifications

The following set of TCs will be available:

TC Code	Description
TC8.4.1-1.20	*Set Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x01 • Table_ID = 0x20 • Length = *length of GET_HK_PAR1*
TC8.4.1-1.21	Same as TC8.4.1-1.20, but with: • Table_ID = 0x21 • Length = *length of GET_HK_PAR2*
TC8.4.1-1.22	Same as TC8.4.1-1.20, but with: • Table_ID = 0x22 • Length = *length of GET_HK_PAR3*
TC8.4.1-1.23	Same as TC8.4.1-1.20, but with: • Table_ID = 0x28 • Length = *length of GET_HK_PAR4*
TC8.4.1-3.20	*Update Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x03 • Table_ID = 0x20 • INDEX = 0 • NDATA = *length of VME GET_HK_PAR1* • DATA = VME GET_HK_PAR1
TC8.4.1-3.21	Same as TC8.4.1-3.20, but with: • Table_ID = 0x21 • NDATA = *length of VME GET_HK_PAR2* • DATA = VME GET_HK_PAR2
TC8.4.1-3.22	Same as TC8.4.1-3.20, but with: • Table_ID = 0x22 • NDATA = *length of VME GET_HK_PAR3* • DATA = VME GET_HK_PAR3
TC8.4.1-3.23	Same as TC8.4.1-3.20, but with: • Table_ID = 0x28 • NDATA = *length of VME GET_HK_PAR4*

	• DATA = VME GET_HK_PAR4
TC8.1.2.1	*Start Cooler Control* standard TC as specified in RD2, with all parameters set to 0
TC8.1.3.1	*Start SCAL Control* standard TC as specified in RD2, with all parameters set to 0
TC8.1.4.1	*Start 300mK Control* standard TC as specified in RD2, with all parameters set to 0
TC8.2.2.1	*Stop Cooler Control* standard TC as specified in RD2
TC8.2.3.1	*Stop SCAL Control* standard TC as specified in RD2
TC8.2.4.1 TC8.4.C0-2.1	*Stop 300mK Control* standard TC as specified in RD2 *Start Command List* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0xC0 • Activity_ID = 0x02 • Table_ID = 0x28 • Index = 0 • N = 0

The following set of VMEs will be available:

VME Code	Pseudo-Code
GET_HK_PAR1	• *While (1)* o *For I=0,79* ▪ *GetTestPar1* (DCU Get command, CID 0x7FA) ▪ *If (parameter != 0x7FA) Generate_Event (5,1)* o *Wait (1 second)*
GET_HK_PAR2	• *While (1)* o *For I=0,79* ▪ *GetTestPar2* (MCU Get command, CID 0x7FB) ▪ *If (parameter != 0x7FB) Generate_Event(5,1)* *Wait (1 second)*
GET_HK_PAR3	• *While (1)* o *For I=0,79* ▪ *GetTestPar3* (SCU Get command, CID 0x7FC) ▪ *If (parameter != 0x7FC) Generate_Event(5,1)* • *Wait (1 second)*
GET_HK_PAR4	• *While (1)* o *For I=0,79* ▪ *GetTestPar4* (DCU Get command, CID 0x7FD) ▪ *If (parameter != 0x7FD) Generate_Event(5,1)* • *Wait (1 second)*

The DRCU commands *GetTestPar1, GetTestPar2 and GetTestPar3* will be custom generated on the DRCU simulator. The CIDs listed in the table above are not used for any of the commands specified in RD3 and RD5. The output buffers of the DRCU simulator will be configured so that the parameters sent in response to the above commands will be identical to the CID; no HK parameter returned in response to standard HK requests will contain any of those values.

3.4.3 Output specifications

Output for this test case will consist of standard HK packets.

3.4.4 Environmental needs

The required set of input TCs will reside in SCOS2000 or on the CDMS Simulator. The DRCU Simulator will be connected to the DPU. A Logic State Analyser will also be used to monitor the GATE lines of the three cables going from the DPU to the DRCU simulator; this will provide evidence of the HK parameter requests traffic on the LS port.

3.5 COMMAND_LIST_EXEC

The purpose is to demonstrate the ability to execute in a timely fashion command lists, both resident on-board and uplinked as part of a TC, via a Virtual Machine as specified in RD4. The command lists will implement simulated data acquisition so that the reception (from S/Ss), control, packing and transmission (to S/C) of science frames will also be tested here. The execution of particular commands or command lists will allow testing of other OBS features like the Time Management, and the Information Distribution Service.

3.5.1 Test Items

TIUL6, TIUL9, TIIL1, TIIL3, TISL2, TISL5, TISL13, TISL14, TISL15, TISL16.

3.5.2 Input specifications

The following set of TCs will be available:

TC Code	Description
TC8.4.C0-1.1	*Execute Command List* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0xC0 • Activity_ID = 0x01 • Length = *length of VME ACQ_PHT* • Data field contains VME ACQ_PHT
TC8.4.C0-3.1	*Stop Command List* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0xC0 • Activity_ID = 0x03
TC8.4.1-1.30	*Set Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x01 • Table_ID = 0x30

	• Length = *length of VME ACQ_PHT*
TC8.4.1-1.31	Same as TC8.4.1-1.30, but with: • Table_ID = 0x31 • Length = *length of VME ACQ_SPT*
TC8.4.1-1.32	Same as TC8.4.1-1.30, but with: • Table_ID = 0x32 • Length = *length of VME P-SW_ACQ*
TC8.4.1-1.33	Same as TC8.4.1-1.30, but with: • Table_ID = 0x33 • Length = *length of VME P-MW_ACQ*
TC8.4.1-1.34	Same as TC8.4.1-1.30, but with: • Table_ID = 0x34 • Length = *length of VME P-LW_ACQ*
TC8.4.1-1.35	Same as TC8.4.1-1.30, but with: • Table_ID = 0x35 • Length = *length of VME S-SW_ACQ*
TC8.4.1-1.36	Same as TC8.4.1-1.30, but with: • Table_ID = 0x36 • Length = *length of VME S-LW_ACQ*
TC8.4.1-1.37	Same as TC8.4.1-1.30, but with: • Table_ID = 0x37 • Length = *length of VME PH-TEST_ACQ*
TC8.4.1-1.38	Same as TC8.4.1-1.30, but with: • Table_ID = 0x38 • Length = *length of VME SP-TEST_ACQ*
TC8.4.1-1.39	Same as TC8.4.1-1.30, but with: • Table_ID = 0x39 • Length = *length of VME PH-OFF_ACQ*
TC8.4.1-1.40	Same as TC8.4.1-1.30, but with: • Table_ID = 0x3A • Length = *length of VME SP-OFF_ACQ*
TC8.4.1-1.41	Same as TC8.4.1-1.30, but with: • Table_ID = 0x3B • Length = *length of VME CONF_PHT*
TC8.4.1-1.42	Same as TC8.4.1-1.30, but with: • Table_ID = 0x3C • Length = *length of VME CONF_SPT*
TC8.4.1-1.43	Same as TC8.4.1-1.30, but with: • Table_ID = 0x3D • Length = *length of VME ACQ_C_PHT*
TC8.4.1-3.30	*Update Table* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0x01 • Activity_ID = 0x03 • Table_ID = 0x30 • INDEX = 0 • NDATA = *length of VME ACQ_PHT*

	• DATA = VME ACQ_PHT
TC8.4.1-3.31	Same as TC8.4.1-3.30, but with: • Table_ID = 0x31 • NDATA = *length of VME ACQ_SPT* • DATA = VME ACQ_SPT
TC8.4.1-3.32	Same as TC8.4.1-3.30, but with: • Table_ID = 0x32 • NDATA = *length of VME P-SW_ACQ* • DATA = VME P-SW_ACQ
TC8.4.1-3.33	Same as TC8.4.1-3.30, but with: • Table_ID = 0x33 • NDATA = *length of VME P-MW_ACQ* • DATA = VME P-MW_ACQ
TC8.4.1-3.34	Same as TC8.4.1-3.30, but with: • Table_ID = 0x34 • NDATA = *length of VME P-LW_ACQ* • DATA = VME P-LW_ACQ
TC8.4.1-3.35	Same as TC8.4.1-3.30, but with: • Table_ID = 0x35 • NDATA = *length of VME S-SW_ACQ* • DATA = VME S-SW_ACQ
TC8.4.1-3.36	Same as TC8.4.1-3.30, but with: • Table_ID = 0x36 • NDATA = *length of VME S-LW_ACQ* • DATA = VME S-LW_ACQ
TC8.4.1-3.37	Same as TC8.4.1-3.30, but with: • Table_ID = 0x37 • NDATA = *length of VME PH-TEST_ACQ* • DATA = VME PH_TEST_ACQ
TC8.4.1-3.38	Same as TC8.4.1-3.30, but with: • Table_ID = 0x38 • NDATA = *length of VME SP-TEST_ACQ* • DATA = VME SP_TEST_ACQ
TC8.4.1-3.39	Same as TC8.4.1-3.30, but with: • Table_ID = 0x39 • NDATA = *length of VME PH-OFF_ACQ* • DATA = VME PH_OFF_ACQ
TC8.4.1-3.40	Same as TC8.4.1-3.30, but with: • Table_ID = 0x3A • NDATA = *length of VME SP-OFF_ACQ* • DATA = VME SP_OFF_ACQ
TC8.4.1-3.41	Same as TC8.4.1-3.30, but with: • Table_ID = 0x3B • NDATA = *length of VME CONF_PHT* • DATA = VME CONF_PHT
TC8.4.1-3.42	Same as TC8.4.1-3.30, but with:

	• Table_ID = 0x3C • NDATA = *length of VME CONF_SPT* • DATA = VME CONF_ SPT
TC8.4.1-3.43	Same as TC8.4.1-3.30, but with: • Table_ID = 0x3D • NDATA = *length of VME ACQ_C_PHT* • DATA = VME ACQ_C_PHT
TC8.4.C0-2.10	*Start Command List* standard TC. Application data will be structured as specified in RD2 and will contain the following parameters: • Function_ID = 0xC0 • Activity_ID = 0x02 • Table_ID = 0x30 • Index = 0 • N = 0
TC8.4.C0-2.11	Same as TC8.4.C0-2.10, but with Table_ID = 0x31
TC8.4.C0-2.12	Same as TC8.4.C0-2.10, but with Table_ID = 0x32
TC8.4.C0-2.13	Same as TC8.4.C0-2.10, but with Table_ID = 0x33
TC8.4.C0-2.14	Same as TC8.4.C0-2.10, but with Table_ID = 0x34
TC8.4.C0-2.15	Same as TC8.4.C0-2.10, but with Table_ID = 0x35
TC8.4.C0-2.16	Same as TC8.4.C0-2.10, but with Table_ID = 0x36
TC8.4.C0-2.17	Same as TC8.4.C0-2.10, but with Table_ID = 0x37
TC8.4.C0-2.18	Same as TC8.4.C0-2.10, but with Table_ID = 0x38
TC8.4.C0-2.19	Same as TC8.4.C0-2.10, but with Table_ID = 0x39
TC8.4.C0-2.20	Same as TC8.4.C0-2.10, but with Table_ID = 0x3A
TC8.4.C0-2.21	Same as TC8.4.C0-2.10, but with Table_ID = 0x3B
TC8.4.C0-2.22	Same as TC8.4.C0-2.10, but with Table_ID = 0x3C
TC8.4.C0-2.23	Same as TC8.4.C0-2.10, but with Table_ID = 0x3D
TC8.4.C1-1.1 TC8.4.C1-2.1	*Set Observation ID* standard TC. Application data will be structured as specified in RD2 and will contain the following 2 data words: 0xA 5A5, 0x5A5A *Set Building Block ID* standard TC. Application data will be structured as specified in RD2 and will contain the following 2 data words: 0x1212, 0x2121
TC8.4.C1-3.1	*Set Observing Mode* standard TC. Application data will be structured as specified in RD2 and will contain the data words: 0xC1C1
TC8.4.C1-4.1	*Set Observation Step* standard TC. Application data will be structured as specified in RD2 and will contain the data word: 0x1
TC8.4.C1-4.2	*Set Observation Step* standard TC. Application data will be structured as specified in RD2 and will contain the data word: 0x2
TC8.4.C1-4.3	*Set Observation Step* standard TC. Application data will be structured as specified in RD2 and will contain the data word: 0x3
TC8.4.C1-4.4	*Synchronize DRCU Counters* standard TC as in RD2
TC14.1.1	*Enable Generation of Telemetry Packets* standard TC. Application data will as specified in AD3 with the following parameters: • N=2 • 1st block o Type = 21 o Subtype = 1

	o SID = 0x200 • 2nd block o Type = 3 o Subtype = 25 o SID = 0x300
TC14.2.1	*Disable Generation of Telemetry Packets* standard TC. Application data will as specified in AD3 with the following parameters: • N = 1 • Type = 21 • Subtype = 1 • SID = 0x200
TC14.2.2	Same as TC14.2.1, but with: • N=1 • Type = 3 • Subtype = 25 • SID = 0x300
TC14.3.1	*Report Enabled Telemetry Packets* standard TC.

A list of VMEs will also be available:

VME Code	**Pseudo-Code**
FRAME_ACQ	• *SetFrameNber (0xFF)* • *SetStartFrame (1)* • *Wait (5 seconds)* • *SetStartFrame (0)* • *Flush FIFOs*
CONT_ACQ	• *SetFrameNber (0)* • *SetStartFrame (1)*
ACQ_PHT	• *SetDataMode (00000)* • *CALL FRAME_ACQ*
ACQ_C_PHT	• *SetDataMode (00000)* • *CALL CONT_ACQ*
ACQ_SPT	• *SetDataMode (00100)* • *CALL FRAME_ACQ*
CONF_PHT	• *SetPhotoSampFreq (0x96)* • *SetPhotoBiasFreq (0x190)* • *SetPhotoBiasMode (0xFF)* • *SetPhotoDemodPh (0x40, 0)* • *SetPhotoDemodPh (0x50, 1)* • *SetPhotoDemodPh (0x60, 2)* • *SetPhotoDemodPh (0x70, 3)* • *SetPhotoBiasAmpl (0x45, 0)* • *SetPhotoBiasAmpl (0x55, 1)* • *SetPhotoBiasAmpl (0x65, 2)* • *SetPhotoBiasAmpl (0x75, 3)*
CONF_SPT	• *SetSpectroSampFreq (0x26)*

	• *SetSpectroBiasFreq (0xC8)* • *SetSpectroBiasMode (0xFF)* • *SetSpectroDemodPh (0xA, 0)* • *SetS pectroDemodPh (0x1A, 1)* • *SetSpectroBiasAmpl (0xAA, 0)* • *SetSpectroBiasAmpl (0xBB, 1)*
P-SW_ACQ P-MW_ACQ	• *SetDataMode (00001)* • *CALL FRAME_ACQ* • *SetDataMode (00002)* • *CALL FRAME_ACQ*
P-LW_ACQ S-SW_ACQ	• *SetDataMode (00003)* • *CALL FRAME_ACQ* • *SetDataMode (00005)* • *CALL FRAME_ACQ*
S-LW_ACQ	• *SetDataMode (00006)* • *CALL FRAME_ACQ*
PH-TEST-ACQ	• *SetDataMode (01000)* • *CALL FRAME_ACQ*
SP-TEST_ACQ	• *SetDataMode (01100)* • *CALL FRAME_ACQ*
PH-OFF_ACQ	• *SetDataMode (11000)* • *CALL FRAME_ACQ*
SP-OFF_ACQ	• *SetDataMode (11100)* • *CALL FRAME_ACQ*
	• *TO BE COMPLETED WHEN I UNDERSTAND HOW TO GENERATE ALL OTHER FRAME TYPES*

3.5.3 Output specifications

The output for this test case will consist of science and event TM packets that will be opened using a DTST. HK Packets will be accessed via SCOS2000 to check, via inspection of relevant HK parameters that the expected sequence of actions has been performed as expected.

3.5.4 Environmental needs

The required set of input TCs will reside in SCOS2000 or on the CDMS Simulator. The DRCU Simulator will be connected to the DPU. At the OBS start-up, packets generation will be enabled for all APIDs, and the default HK packet structure will be defined on-board.

4 Test Procedures

The loading/start/stop/debug functionalities for the OBS on the DPU are managed from a PC using the DSP In-Circuit Emulator software. In case SCOS2000 can be used to send TC packets, it is assumed that full chain SCOS2000+Router+CDMS Simulator is operational.

4.1 TP1

This procedure executes the test cases DPU_COMMAND_EXEC and DPU_MEM. Procedure steps, which will be repeated as part of the acceptance tests, are lightly shaded. The TCs are identified by their codes as specified in 3.1. OBS loading is performed via ICE; OBS run/stop/restart functions are performed in ICE in CBUG mode.

Step #	Action	Pass/Fail	Test Item
1	Open VIRTUOSO project file in directory where the code resides.		
2	Assign the HK_ASK task to the EXE_NOBOOT group and compile the OBS		
3	Load the OBS in the DPU.		
4	Set a BP in OBS where the TC acceptance report is generated.		
5	Start the OBS.		
6	Start the CDMS Simulator.		
7	Send TC17.1.1	OBS stops at BP	TISL1
8	Inspect the location in the DM where the report TM packet has been written.	Verify format in conformity with AD3.	
		Verify content of packet to reflect TC type (valid/invalid)	TIUL1
9	Remove BP. Restart OBS.		
10	**Send TC17.1.1**	Verify reception of: TM (17,2), TM (1,1)	TIUL2, TISL1, TISL17
11	Send TC14.3.1	Verify reception of: TM (14,4), TM (1,1) Verify that the list of SIDs in TM (14,4) matches the list of enabled TM packets (all of them are enabled by defaults at start-up).	
12	Send TC9.7.1	Verify reception of: TM (9,9), TM (1,1)	TISL15p
13	Stop OBS. Stop CDMS. Open CDMS file APID2RT.txt and associate SPIRE with APID 0x400; this is needed to force the CDMS to send TCs with wrong APID to SPIRE.		
14	Start OBS. Start CDMS.		
15	Send TC17.1.2	Verify reception of TM (1,2) with failure code 0	
16	Stop OBS. Stop CDMS. Open CDMS file APID2RT.txt and change SPIRE's APID back to nominal.		

17	Start OBS. Start CDMS.		
18	Send TC17.1.3	Verify reception of TM (1,2) with failure code 1	
19	Send TC17.1.4	Verify reception of TM (1,2) with failure code 2	
20	Send TC17.1.5	Verify reception of TM (1,2) with failure code 3	
21	Send TC17.1.6	Verify reception of TM (1,2) with failure code 4	TISL9
22	Send TC6.5.1	Verify reception of TM (1,1) and TMs (6,6). Check that the received words are different from the pattern contained in TC6.2.1	TIUL3
23	Send TC6.2.1	Verify reception of TM (1,1)	
24	Send TC6.5.1	Verify reception of TM (1,1) and TMs (6,6). Compare received data words to the pattern uplinked in TC6.2.1. Store received words into a text file on the CDMS computer. Run program CRC on this file and record the computed CRC.	
25	Send TC6.9.1	Verify reception of TM (1,1) and TM (6,10). Verify that the received Checksum is identical to CRC computed in the previous step.	
26	Send TC6.2.2	Verify reception of TM (1,8) with error code 0x601	
27	Send TC6.2.3	Verify reception of TM (1,8) with error code 0x602	
28	Send TC6.2.4	Verify reception of TM (1,8) with error code 0x603	
29	Send TC6.2.5	Verify reception of TM (1,8) with error code 0x604	
30	Send TC6.2.6	Verify reception of TM (1,8) with error code 0x605	TISL11
31	Send TC8.4.1-2.1	Verify reception of TM (1,8) with error code TBD	
32	Send TC8.4.1-1.1	Verify reception of TM (1,1)	
33	Send TC8.4.1-2.1	Verify reception of TM (1,1) and TM (21,1). Verify that the received patter is different from the one contained in TC8.4.1-1.1	

34	Send TC8.4.1-3.1	Verify reception of TM (1,1)	
35	Send TC8.4.1-2.1	Verify reception of TM (1,1) and TM (21,1). Verify that the received pattern is identical to that uplinked in TC8.4.1-1.1	
36	Send TC8.4.1-1.2	Verify reception of TM (1,8) with error code 0x801	
37	Send TC8.4.1-1.3	Verify reception of TM (1,8) with error code 0x802	
38	Send TC8.4.1-1.4	Verify reception of TM (1,8) with error code 0x804	
39	Send TC8.4.1-1.5	Verify reception of TM (1,8) with error code 0x805	
40	Send TC8.4.1-1.6	Verify reception of TM (1,8) with error code 0x806	
41	Send TC8.4.1-3.2	Verify reception of TM (1,8) with error code TBD	
42	Send TC8.4.1-3.3	Verify reception of TM (1,8) with error code 0x806	
43	Send TC8.4.1-3.4	Verify reception of TM (1,8) with error code 0x807	
44	Send TC8.4.1-3.5	Verify reception of TM (1,8) with error code 0x808	

4.2 TP2

This procedure executes test case HK_COLLECT. It is assumed at this stage that procedure TP1 has been executed successfully. The DPU-S/C interface and the capability of the OBS to receive, interpret and execute commands should have been successfully tested. Procedure steps, which will be repeated as part of the acceptance tests, are lightly shaded.

Step #	Action	Pass/Fail	Test Item
1	Open VIRTUOSO project file in directory where the code resides.		
2 3	Assign the HK_ASK task to the EXE group and compile the OBS Configure DRCU Simulator to assign pre-defined values to the set HK parameters that will be sent to the DPU.		
4	Load the OBS in the DPU.		
5	Set a BP in the OBS where task LS reads the commands stored in the low priority command queue, after the commands are actually sent to the S/Ss.		
6	Start OBS	OBS stops at BP about one	TIUL3

		second after start, at the first periodic request of HK parameters	
7	Remove previous BP. Set a new BP in the OBS where the LS task receives the HK parameters from the S/S.		
8	Start DRCU simulator.		
9	Start OBS.	OBS stops at BP as in step 6+ 2 msec. Verify that the value of the received parameter matches the input value pre-defined in the DRCU Simulator.	TIUL5 TISL3
10	Remove previous BP. Set a new BP in OBS when the notification of complete HK packet is sent to TMTC.		
11	Start OBS.	When the OBS stops, examine the DM area where the HK packet has been stored and inspect its integrity.	TIIL2
12	Remove previous BP.		
13	**Start OBS. Start CDMS Simulator.**	Verify periodic (1/sec) reception of TM (3,25) HK packets with SID 0x300. Verify periodic (0.5/sec) reception of TM (3,26) diagnostic packets with SID 0x301	TIUL2, TISL4
14	**Send TCTest.1 10 times, spaced by at least 3 seconds**	Verify that no TM (3,25) or TM (3,26) packets are lost by checking that the received packet counter in the CDMS log window shows no jumps	TIUL2
15	Send TC8.4.1-1.10		
16	Send TC8.4.1-1.11		
17	Send TC8.4.1-3.10		
18	Send TC8.4.1-3.11		
19	Send TC3.2.1	Verify periodic (10/sec) reception of additional TM (3,26) diagnostic packets with SID 0x302	
20	Send TC3.2.2	Verify periodic (10/sec) reception of additional TM (3,26) diagnostic packets with SID 0x303	

21	Send TC3.9.1	Verify reception of TM (3,10) with HKPCKTID = 0	
22	Send TC3.11.1	Verify reception of TM (3,12) with HKPCKTID = 1,2,3	
23	Send TC3.4.1	Verify that reception of TM (3,26) with SID 0x302 has stopped	
24	Send TC3.11.1	Verify reception of TM (3,10) with HKPCKTID = 1,3	
25	Send TC3.4.2	Verify that reception of TM (3,26) with SID 0x303 has stopped	
26	Send TC3.11.1	Verify reception of TM (3,10) with HKPCKTID = 1	TISL10

4.3 TP3

This procedure executes test case VM. It is assumed at this stage that procedures TP1 and TP2 have been successfully executed. We will progressively flood the LS port with HK parameter requests to the DRCU simulator up and beyond the maximum number of requests that can be handled in 1 second (about 500). The measurement with the oscilloscope will be used to verify that this is actually happening.
The requests will come from the HK_ASK task, from the H/W VM and from the 3 S/W VMs that should nominally run the PID controls. Each request source expects different parameter values; the test will be passed if each source receives exactly the expected parameter values without any response mixing.

Step #	Action	Pass/Fail	Test Item
1	Create custom commands in the DRCU simulator, one per subsystem, with CIDs 0x7FA, 0x7FB and 0x7FC. Set the returned parameter to be equal to CID and make sure these values are not returned by DRCU simulator for any of the standard DRCU commands.		
2	Start OBS. Start CDMS simulator. *At this stage, about 220 HK parameter requests are sent to the DRCU simulator. Each request requires 2msec to be served.*	Verify the periodic reception of HK (1/sec) and diagnostic (1/2 sec) packets.	
3	Send TC8.4.1-1.20		
4	Send TC8.4.1-1.21		
5	Send TC8.4.1-1.22		
6	Send TC8.4.1-1.23		

7	Send TC8.4.1-3.20		
8	Send TC8.4.1-3.21		
9	Send TC8.4.1-3.22		
10	Send TC8.4.1-3.23		
11	Send TC8.4.C0-2.1 *Now there are 80 additional HK parameter requests going to the LS port.*	Verify that no TM (5,1) events are received Verify increase of traffic with the DCU on the LSA display.	
12	Send TC8.1.2.1 *Now there are 80 additional HK parameter requests going to the Ls port*	Verify that no TM (5,1) events are received Verify increase of traffic with the DCU on the LSA display.	
13	Send TC8.1.3.1 *Now there are 80 additional HK parameter requests going to the Ls port*	Verify that no TM (5,1) events are received Verify increase of traffic with the MCU on the LSA display.	
14	Send TC8.1.4.1 *Now there are 80 additional HK parameter requests going to the Ls port. At this point we have passed the number of total requests (about 500) that can go through the LS port each second: we might be losing some HK packets, but this is no problem for the current tests.*	Verify that no TM (5,1) events are received Verify increase of traffic with the SCU on the LSA display.	
15	Wait 10 seconds, then stop the OBS		
16	Open the file TelemetryA.txt resident on the CDMS simulator		
17	Perform a search for the values 0x7FA, 0x7FB, 0x7FC and 0x7FD	Verify that search produced negative results.	TIUL10

4.4 TP4

This procedure executes test case COMMAND_LIST_EXEC. It is assumed at this stage that procedures TP1, TP2 and TP3 have been successfully executed. The DPU correctly interfaces with the CDMS simulator and the DRCU simulator. Procedure steps that will be repeated as part of the acceptance tests are lightly shaded.

Step #	Action	Pass/Fail	Test Item
18	Configure DRCU Simulator to assign pre-defined values to the set HK parameters and the science data that will be sent to the DPU.		
19	Set a BP in the OBS soon after reception of Half-FIFO-Full interrupt.		

20	Start OBS. Start CDMS Simulator.		
21	Send TC8.4.C0-1.1	OBS should stop at BP.	
22 23	Using the ICE GUI proceed step-by-step in the code to read the science data present on the FIFOs Remove previous BP. Set new BP where a complete science TM packet is ready to be sent and the notification from HS is received by TMTC.	Verify correct reception and interpretation of science frames.	TIUL6
24	Restart OBS		
25	Send TC8.4.C0-1.1	OBS stops at BP.	
		Using the ICS GUI check the locations of DM where the built packet is held and inspect integrity of header (APID, counter, type and subtype) and content (compare to input data from DRCU Simulator).	TISL5
26	Remove BP.		
27	Restart OBS		
28	Send TC8.4.C0-1.1	Verify reception of TM (21,1) science packets containing 255 Frames. Check correctness of APID, count, type and subtype as packets appear on the CDMS Simulator GUI. Use a DTST to inspect and verify received packets aga inst pattern sent by DRCU	TIUL2, TIIL3, TISL16
29	Send TC8.4.1-1.43		
30	Send TC8.4.1-3.43		
31	Send TC8.4.C0-2.23. After 5 seconds execute next step.	Verify reception of incoming TM (21,1) science packets	
32	Send TC8.4.C0-3.1	Verify that TM (21,1) transmission stops	
33	Send TC8.4.1-1.30		
34	Send TC8.4.1-3.30		
35	Send TC8.4.C1-1.1	Verify that OBSID value has been updated in HK packet	
36	Send TC8.4.C1-2.1	Verify that BBID value has been updated in HK packet	
37	Send TC8.4.C1-3.1	Verify that MODE value has been updated in HK packet	
38	Send TC8.4.C1-4.1	Verify that STEP value has been updated in HK packet	

39	Send TC8.4.C1-4.4	Verify that time of last DRCU sync has be reset in the HK packet	TISL15
40	Send TC8.4.C0-2.10	Verify reception of TM (21,1) science packets containing 255 Frames.	
		TM packets will have APID = 0x504 and SID = 0x200	
		Content of Science Data should be checked against DRCU input data.	
41	Send TC8.4.1-1.31		
42	Send TC8.4.1-3.31		
26	Send TC8.4.C0-2.11	Verify reception of TM (21,1) science packets containing 255 Frames.	
		TM packets will have APID = 0x505 and SID = 0x201	
		Content of Science Data should be checked against DRCU input data.	
27	Send TC8.4.1-1.32		
28	Send TC8.4.1-3.32		
29	Send TC8.4.C0-2.12	Verify reception of TM (21,2) science packets containing 255 Frames.	
		TM packets will have APID = 0x504 and SID = 0x102	
		Content of Science Data should be checked against DRCU input data.	
30	Send TC8.4.1-1.33		
31	Send TC8.4.1-3.33		
32	Send TC8.4.C0-2.13	Verify reception of TM (21,2) science packets containing 255 Frames.	
		TM packets will have APID = 0x504 and SID = 0x103	
		Content of Science Data should be checked against DRCU input data.	
33	Send TC8.4.1-1.34		
34	Send TC8.4.1-3.34		
35	Send TC8.4.C0-2.14	Verify reception of TM (21,2) science packets containing 255 Frames.	
		TM packets will have APID = 0x504 and SID = 0x104	

		Content of Science Data should be checked against DRCU input data.	
36	Send TC8.4.1-1.35		
37 38	Send TC8.4.1-3.35 Send TC8.4.C0-2.15	Verify reception of TM (21,2) science packets containing 255 Frames.	
		TM packets will have APID = 0x505 and SID = 0x105	
		Content of Science Data should be checked against DRCU input data.	
39	Send TC8.4.1-1.36		
40	Send TC8.4.1-3.36		
41	Send TC8.4.C0-2.16	Verify reception of TM (21,2) science packets containing 255 Frames.	
		TM packets will have APID = 0x505 and SID = 0x106	
		Content of Science Data should be checked against DRCU input data.	
42	Send TC8.4.1-1.37		
43	Send TC8.4.1-3.37		
44	Send TC8.4.C0-2.17 (PH_TEST)	Verify reception of TM (21,3) science packets containing 255 Frames.	
		TM packets will have APID = TBD and SID = TBD	
		Content of Science Data should be checked against DRCU input data.	
45	Send TC8.4.1-1.38		
46	Send TC8.4.1-3.38		
47	Send TC8.4.C0-2.18 (SP-TEST)	Verify reception of TM (21,3) science packets containing 255 Frames.	
		TM packets will have APID = TBD and SID = TBD Content of Science Data should be checked against DRCU input data.	
48	Send TC8.4.1-1.39		
49	Send TC8.4.1-3.39		
50	Send TC8.4.C0-2.19	Verify reception of TM (21,4) science packets containing 255 Frames.	

		TM packets will have APID = 0x504 and SID = 0x208	
51	Send TC8.4.1-1.40	Content of Science Data should be checked against DRCU input data.	
52	Send TC8.4.1-3.40		
53	Send TC8.4.C0-2.20	Verify reception of TM (21,4) science packets containing 255 Frames.	
		TM packets will have APID = 0x505 and SID = 0x209	
		Content of Science Data should be checked against DRCU input data.	TISL16
54	Send TC8.4.1-1.41		
55	Send TC8. 4.1-3.41		
56	Send TC8.4.C0-2.21	Verify that configured parameters are reflected in HK packets	
57	Send TC8.4.1-1.42		
58	Send TC8.4.1-3.42		
59	Send TC8.4.C0-2.22	Verify that configured parameters are reflected in HK packets	
60	Send TC14.3.1	Verify reception of a TM (14,4) packet. Transmission of all packets should be enabled.	
61	Send TC8.4.C0-2.23	Verify continuous reception of TM (21,1) packets	
62	Send TC14.2.1	Verify that TM (21,1) are no longer received	
63	Send TC14.2.2	Verify that TM (3,25) HK Packets are no longer received	
64	Send TC14.3.1	Verify reception of TM (14,4) packet. Generation for TM (21,1,0x200) and TM (3,25,0x300) should not be present in the report.	
65	Send TC14.1.1	Verify that TM (21,1,0x200) and TM (3,25,0x300) are again received	
66	Send TC14.3.1	Verify reception of a TM (14,4) packet. Transmission of all packets should be enabled.	TISL14