HCSS 0.1.1 Acceptance Test Report for SPIRE

Sunil Sidher and Steve Guest

29th October 2002

1. INTRODUCTION

This acceptance test report is based on the tests carried out by SPIRE at RAL on the Herschel Common Science System (HCSS) version 0.1.1 on 18th October 2002. All the tests listed in AD01 were attempted. The documentation listed in section 2 was used for the preparation and execution of the tests. Section 3 describes the RAL system setup and configuration for these tests. In section 4 the results from the execution of each test are described.

In cases where some tests could not be conducted because of the failure of earlier an earlier test in the chain, we chose to improvise and run additional tests to exercise the system as far as possible. Furthermore, during the course of these tests, we identified some new tests to test a different functionality of the HCSS. These tests are described in section 5.

All other auxiliary information relevant to the tests is included in the appendices.

2. DOCUMENTATION

2.1 Applicable Documents

AD01	HCSS Acceptance Test Plan, Draft 1.1, 19th June 2002

2.2 Reference Documents

RD01	MIB User Guide, web document ftp://astro.esa.int/pub/HERSCHEL/csdt/releases/doc/mib/guide/userguide.html, downloaded on 17th October 2002
RD02	HCSS telemetry ingestion software user manual, Issue 0.2, 17th May 2002, HERSCHEL-HSC-DOC-0231
RD03	Common Uplink System User's Guide, Issue 0.5, 19th September 2002
RD04	Database MIB Import ICD, Version 5.1, 26th October 2001, S2K-MCS-ICD-0001-TOS-GCI

3. TEST SETUP

3.1 System Setup

The SPIRE setup for the acceptance test was as follows:

Lincoln: S2K w/s running Linux SuSE 7.3, SCOS2000 v2.3e + TOPE and HCSS Build 82.

Truro: Data Server running Linux SuSE 7.3, HCSS Build 82.

Both Lincoln and Truro had Java 1.3 1-04 and Versant 6.0.1 installed

SSTDSDSVAIO: Laptop running Windows XP for QLA 0.2

SSTDKJKTOSHIBA: Laptop running Windows 2000, Java SDK 1.4.0 and EGSE tools for generating test science and HK telemetry for SPIRE.

NOTE: We used the locally modified properties file, hcss.props, originally installed by Rob Zondag & Craig Porrett. See Appendix A for the actual contents of this file as used at the start of these tests.

3.2 Personnel

The test setup and execution was performed by the following group:

Steve Guest, Ken King, Jeff Payne and Sunil Sidher

4. HCSS TEST CASES

4.1 Test Case TC-ICC-01: MIB Ingestion and TC Definition File Generation

4.1.1 Test Procedure

For this test AD01 only gives an outline test procedure to be followed and refers the user to the MIB User Guide (RD01). Test Case TC-ICC-02. We therefore set out below the actual procedure followed:

1. Copied the latest AVM MIB ASCII tables from sops23e account into /home/sds78/MIB_AT0 using scp, rather than FTP as suggested in AD01.

2. Created subdirectory aux in /home/sds78/MIB_AT0 lincoln/home/sds78/MIB_AT0>

3. Changed directory to /home/sds78/MIB_AT0 directory and ran preparemib which generated several warning and error messages (see Appendix B).

4. Tried to create registry using the createregistry tool. Successful execution generated following output:

   lincoln/home/sds78/MIB_AT0> createregistry
   Loading properties: CreateRegistry.defaults
   Loading properties: /home/hcssbld/hcss_builds/latest_build/config/devel.props
   Loading properties: /home/hcssbld/hcss.props
   [HCSS-INFO]:MibRegistry: Adding registry to database under root hcssbld_dbroot...
   [HCSS-INFO]:MibRegistry: Done.

4.1.2 Test Result

The test initially failed because the preparemib tool did not recognise dst table in the MIB. The test still failed despite extensive troubleshooting activities (see section 4.1.3). SPR issued.

4.1.3 Troubleshooting Activities

1. Introduced the dst table into the mib table definitions in directory /home/hcssbld/hcss_builds/latest_userrelease/data/mib/defns/table-defns.
   In the script /home/hcssbld/hcss_builds/latest_build/bin/preparemib added the dst table in the for loop

2. Tried to run mibchecker tool which failed because it required modification of the properties file (hcss.props). See Appendix C1 for the error log. We need to be able to change the directory for picking up the MIB tables from the command line. SPR issued.

3. Changed the hcss.props file definition from
   var.mib.data = ${var.hcss.dir}/data/mib/examplemibs/example-1
   to
   var.mib.data = /home/sds78/MIB_AT0/MIB-AUX-DATA

4. The mibchecker tool generated a Java error messages (see Appendix C2).

5. Changed the following line in the hcss.props file from
   var.hcss.dir = /home/hcssbld/hcss_builds/latest_user_release
   to
   var.hcss.dir = /home/hcssbld/hcss_builds/latest_userRelease
   SPR issued.

6. The mibchecker tool still generated warning and error messages (see Appendix D). Some of the errors appeared to be related to the tables supplied by Serge Valera during the installation of SCOS 2000 2.3e.

7. The MIB was further checked against RD04. Some redundant trailing tab characters were removed but the mibchecker still generated warning and error messages. These problems were reported to Hassan Siddiqui. An SPR has been raised.

4.2 Test Case TC-ICC-02: TC Definition File Import

This test could not be carried out because of the failure of TC-ICC-01.

4.3 Test Case TC-ICC-03: Observing Mode Import

From the CUS editor defined an observing mode called Mode_POF1_0. This mode consists of three Building Blocks (BBs):

- Block_Init_0
- Block_PCal_0
- Block_Chop_0

and three Commands:

- Cmd_Set_ObsID
- Cmd_Set_BBID
- Cmd_Set_ObsMode

These definitions are shown in Appendix E together with problems encountered in using the CUS editor. SPRs have been issued.

4.3.1 Test Procedure

1. Started the CUS GUI by typing cusgui from the user account on truro.
2. Using the "Import" button from the "Registry" menu loaded the aforementioned definition files.

3. Selected the "Commit" button from the "Registry" menu and clicked on the "OK" option. The commit was successful and no error messages or warnings were generated.
4. Selected the "Quit" button from the "File" menu to exit the CUS GUI.

4.3.2 Test Result

The test was completed successfully.

4.4 Test Case TC-ICC-04: Issuing Telecommands

This test is designed to check the basic functionality of SCOS 2000 independently of the HCSS.

4.4.1 Test Procedure

1. Started up the SCOS 2000 EGSEsrv tasks from the sops23e SCOS account.
2. Started up the EXIF task from SCOS
3. From EXIF GUI started TOPE1, EXIF_TMM, EXIF_CHM, EXIF_TM1 and EXIF_CH1 tasks.
4. Selected the SPIRE OBS alphanumeric display SAT00480 from the SCOS2000 Telemetry Desktop (MON2).
5. Started up the SCOS 2000 TC History application (TCHIST2)
6. Started up the EGSE tools on SSTDKJKTOSHIBA to respond to issued commands and generate appropriate HK and science telemetry.
7. Executed a simple test procedure called TestClient.tcl from the TOPE (see Appendix F for the script and the relevant problem reports).
8. Watched the issued telecommands in both the TestClient console and the TC History application. No anomalies observed.
   - The EGSE test tools responded with appropriate HK and science TM
9. Watched the Observing Mode, OBSID and BBID parameters change as expected on the SAT00480 alphanumeric display. No anomalies observed.

4.4.2 Test Result

The test was completed successfully.

4.5 Test Case TC-ICC-05: Running Test Observation on TestClient

This test could not be carried out because of the failure of TC-ICC-01.

4.6 End to End Test Case TC-ICC-06: Run a Test Observation, Ingest TM packets and run QLA

This test could not be carried out because of the failure of TC-ICC-01.

5. ADDITIONAL TEST CASES

As stated earlier the tests in this section were not formally a part of the HCSS acceptance test plan.

5.1 Test Case SPIRE-TC-01: Run Test Procedure from TOPE, Ingest TM packets, run QLA

This was an additional test case introduced by SPIRE to run a test procedure from TOPE in the absence of an HCSS generated test observation. A test procedure generated on the SCOS 2000 and TOPE system is executed as in TC-ICC-04. It demonstrates how TM ingestion and QLA perform with the HK and science TM. At the start of this test HK was already being generated.

5.1.1 Test Procedure

1. Prepared QLA (on laptop SSTDSDSVAIO)
2. Ran TM ingestion: *User Guide (RD02) says type "tmingest"; had to type "tmingestion". SPR issued*
   - Ingestor started correctly and immediately started ingesting HK packets.
   - QLA received HK packets
3. Ran TestClient.tcl as for test TC-ICC-04. Science TM generation started on SSTDKJKTOSHIBA.
   - Science packets from EGSE tools appeared to be recognised (by TM ingestion console).
   - QLA received science packets
4. Stopped the Packet stream.
   - 252 packets had been ingested in the database
5. Prepared QLA for playback (on lincoln). Successful playback from database on truro
   - Some QLA playback errors were generated. These are shown in Appendix G. An SPR has been issued.

5.1.2 Test Result

Test was largely successful despite the QLA playback errors.

5.2 Test Case SPIRE-TC-02: RTA Mode Playback (i.e. database -> router):

This test was designed to test the functionality of QLA in RTA mode to playback HK data from the database via the router.

5.2.1 Test Procedure

1. Prepare QLA
2. Start the RTA-mode playback tool
   - the precise command is not particularly well-defined, which was the reason for the error.
   - java -Dproperty.path=${HCSS_PROPS} nl.esa.herschel.access.util.RtaPlayback should have worked.
3. Press the start button on the RTA-mode playback tool
   - as there is no selection step specified, we assume here that we started the acceptance test with an empty database and will playback everything
   - Observe data reception by both RTA and QLA
4. Press the stop button on the RTA-mode playback tool. Check (with QLA) that the same number of packets have been received as were sent.

5.2.2 Test Result

This test failed with the following error message:

```
> java nl.esa.herschel.access.util.RtaPlayback
java.lang.NullPointerException
```

at nl.esa.herschel.access.util.DataSelector$OpenHandler.run(DataSelector.java:348)
at java.lang.Thread.run(Thread.java:484)

Entering an incorrect command caused this error. In particular the properties file had not been picked up. This test will be tried again when the acceptance test is repeated.

6. CONCLUSIONS

Table 1 below lists all the Non Conformance Reports (NCRs) resulting from these tests and our response in each case. It is clear from the results of the above tests that the acceptance test for HCSS 0.1.1 has failed. Nevertheless we have managed to test significant elements of the system during these tests and are therefore confident that we will be in a good position when we conduct these tests again in the near future.

Table 1 List of NCRs raised during the acceptance test

NCR Details	Action taken
NCR-01: The dst table in the MIB is not recognised by the MIB ingestor	SPR-0162 raised
NCR-02: The MIB ingestor reports problems with the MIB tables as it is using version 4.2 of the MIB Database Import ICD.	SPR-0162 raised
NCR-03: An error in the hcss.props file caused the mibchecker to fail	
NCR-04: The CUS editor only allows opening of existing definitions bottom up, i.e. the command definitions have to be opened first, followed by the BBs and the observing mode. SPR has been issued.	SPR-0173 raised
NCR-05: The CUS does not allow command definitions to appear directly in the observing mode definition.	SPR-0174
NCR-06: The TM Ingestor User Guide (RD02) says type "tmingest"; had to type "tmingestion"	
NCR-07: TCL editor: Ctrl+V does not work	
NCR-08: TCL editor: While copying and pasting the screen did not refresh properly. Only upon scrolling up and down did the screen refresh	All these problems have been reported to Serge Valera.
NCR-09: TCL editor: The undo on a comment string appeared in black and not in green	
NCR-10: TOPE: On stopping the EXIF tasks, EXIF_TM1 does not get stopped. It has to be stopped again separately afterwards	
NCR-11: The mibchecker reported problems with MIB table entries supplied by ESTEC	

APPENDIX A: HCSS PROPERTIES FILE

This is the HCSS Properties File as used by SPIRE for conducting the HCSS 0.1.1 acceptance test.

```
# $Id$
# HCSS Properties File - location RAL
#
# Author: Craig Porrett
# RAL version: SG, 17th October 2002
#

# General
var.server = truro.bnsc.rl.ac.uk
var.database.server = @${var.server}
var.database.devel = ${user.name}${var.database.server}
var.router.host = ${var.server}
var.router.port = 9877
var.instrument = SPIRE
var.model = AVM

var.hcss.workdir = ${user.home}/
#var.hcss.dir = ${user.home}/hcss_builds/latest_build
#var.hcss.dir = /home/hcssbld/hcss_builds/latest_build
#var.hcss.dir = /home/hcssbld/hcss_builds/latest_user_release
var.hcss.dir = /home/hcssbld/hcss_builds/latest_userRelease

# Access
hcss.access.database = ${var.database.devel}
hcss.access.test.database = ${var.database.devel}
hcss.access.connection = nl.esa.herschel.access.db.LocalConnection
hcss.access.network = socket
hcss.access.socket.host = localhost
hcss.access.socket.port = 8050
hcss.access.url = http://scott1.bnsc.rl.ac.uk:8080/servlets/
hcss.access.packetprocessor = ${var.instrument}
hcss.access.instrumentmodel = ${var.model}
hcss.access.factory.query = nl.esa.herschel.access.db.VersantQueryFactory
hcss.access.router.host = ${var.router.host}
hcss.access.router.port = ${var.router.port}
hcss.access.query.allpks = select selfoid from nl.esa.herschel.versant.ccm.TmSourcePacketImpl
hcss.access.query.alldfs = select selfoid from nl.esa.herschel.versant.ccm.DataFrameImpl

# CCM
hcss.ccm.test.database = ${var.database.devel}

# Formatter
# formatter package needs to be changed to use the var.hcss.dir system
hcss.formatter.directory.root = ${var.source.dir}

# MIB
```

Technical Report

HCSS 0.1.1 Acceptance Test Report
Sunil Sidher & Steve Guest

Ref:	SPIRE-RAL-REP-001427
Issue:	1.0
Date:	29th October 2002
Page:	10 of 18

```
var.mib.defns = ${var.hcss.dir}/data/mib/defns
#var.mib.data  = ${var.hcss.dir}/data/mib/example-mibs/example-1
var.mib.data  = /home/sds78/MIB_AT1/MIB-AUX-DATA
var.mib.aux   = ${var.mib.data}/auxil
var.mib.raw   = ${var.mib.data}/ascii-tables
hcss.mib.database          = ${var.database.devel}
hcss.mib.datadir           = ${var.mib.raw}
hcss.mib.tablelist         = ${var.mib.aux}/tablelist
hcss.mib.tc_command_durns     = ${var.mib.aux}/tc-durns
hcss.mib.tm_param_list        = ${var.mib.aux}/tmparams
hcss.mib.test_tc_command_list = ${var.mib.aux}/tcmds
hcss.mib.test_tm_param_list   = ${hcss.mib.tm_param_list}
hcss.mib.tabledefs         = ${var.mib.defns}/table-defns/
hcss.mib.dbroot            = hcssbld_dbroot
hcss.mib.uplink_id         = 1
hcss.mib.test_uplink_id    = 1
hcss.mib.downlink_id       = 1
hcss.mib.test_downlink_id    = 1
hcss.mib.errorsonly        = false
hcss.mib.logfile           = mibchecker.log


# TM Ingest
hcss.tmingest.database = ${var.database.devel}
hcss.tmingest.instrument = ${var.instrument}
hcss.tmingest.model = ${var.model}
hcss.tmingest.tmpacketprocessor = NONE
hcss.tmingest.server = ${var.router.host}
hcss.tmingest.port = ${var.router.port}

# Store
hcss.store.test.database = ${var.database.devel}
```

APPENDIX B: WARNING AND ERROR MESSAGES FROM PREPAREMIB

lincoln/home/sds78/MIB_AT0> preparemib
Processing table caf..
Processing table cap..
Processing table cca..
Processing table ccf..
Processing table ccs..
.
.
.
Processing table dst..
File name dst.dat hasn't a valid name. Ignoring this file.
.
.
.
Processing table pic3..
File name pic3.dat hasn't a valid name. Ignoring this file.
Processing table pid..
Processing table pid3..
File name pid3.dat hasn't a valid name. Ignoring this file.
.
.
.
Processing table vdf..
Processing table vpd..
Using ./ccf.dat to create a TEMPLATE command duration file MIB-AUX-DATA/auxil/tc-durns ..
Done.
lincoln/home/sds78/MIB_AT0>

APPENDIX C: INITIAL WARNING AND ERROR MESSAGES FROM MIBCHECKER

C1: First Error Message

lincoln/home/sds78/MIB_AT0> mibchecker
Loading properties: MibChecker.defaults
Loading properties: /home/hcssbld/hcss_builds/latest_build/config/devel.props
Loading properties: /home/hcssbld/hcss.props
Exception in thread "main" java.io.FileNotFoundException:

/home/hcssbld/hcss_builds/latest_user_release/data/mib/example-mibs/example-1/auxil/tablelist (No such file or directory)
    at java.io.FileInputStream.open(Native Method)
    at java.io.FileInputStream.<init>(FileInputStream.java:64)
    at nl.esa.herschel.mib.tools.MibChecker.exec(MibChecker.java:52)
    at nl.esa.herschel.mib.tools.MibChecker.main(MibChecker.java:126)
lincoln/home/sds78/MIB_AT0>

C2: Second Error Message

lincoln/home/sds78/MIB_AT0> mibchecker
Loading properties: MibChecker.defaults
Loading properties: /home/hcssbld/hcss_builds/latest_build/config/devel.props
Loading properties: /home/hcssbld/hcss.props
[HCSS-INFO]:TabRead: Reading table caf
Exception in thread "main" java.io.FileNotFoundException:

/home/hcssbld/hcss_builds/latest_user_release/data/mib/defns/table-defns//table_caf.def (No such file or directory)
    at java.io.FileInputStream.open(Native Method)
    at java.io.FileInputStream.<init>(FileInputStream.java:64)
    at
nl.esa.herschel.mib.tables.MibRecordDefnReader.readFromFile(MibRecordDefnReader.java:44)
    at nl.esa.herschel.mib.tools.MibChecker.exec(MibChecker.java:86)
    at nl.esa.herschel.mib.tools.MibChecker.main(MibChecker.java:126)

APPENDIX D: FINAL WARNING AND ERROR MESSAGES FROM MIBCHECKER

lincoln/home/sds78/MIB_AT0> mibchecker
Loading properties: MibChecker.defaults
Loading properties: /home/hcssbld/hcss_builds/latest_build/config/devel.props
Loading properties: /home/hcssbld/hcss.props
[HCSS-INFO]:TabRead: Reading table caf
[HCSS-INFO]:TabRead: Reading table cap
[HCSS-INFO]:TabRead: Reading table cca
.
.
.
[HCSS-INFO]:TabRead: Reading table vpd
[HCSS-ERROR]:MibCheckErrors: There was one or more serious problems in the imported MIB files. Check the

file mibchecker.log for details.
Exception in thread "main" nl.esa.herschel.mib.utils.MibException: One or more serious validation problems encountered.
      at nl.esa.herschel.mib.tools.MibChecker.exec(MibChecker.java:103)
      at nl.esa.herschel.mib.tools.MibChecker.main(MibChecker.java:126)

- Typical error message in the mibchecker.log file in the MIB_AT0 dir:

=== Table ccf START
A problem was found. Events prior to problem:
Table ccf record 1: Field 1: got [LOADTT]
Table ccf record 1: Field 2: got [LOAD TIME TAG]
Table ccf record 1: Field 3: got [Command inserting time-tagged commands into the on-board queue]
Table ccf record 1: Field 4: got []
Warning: UseofDefaultWarning Table ccf record 1: In field 4 the field value is null. Using default value of [null]
A problem was found. Events prior to problem:
Table ccf record 1: Field 5: got [N]
Table ccf record 1: Field 6: got [STANDARD]
Table ccf record 1: Field 7: got [11]
Table ccf record 1: Field 8: got [4]
Table ccf record 1: Field 9: got []
Warning: NumericConversionWarning Table ccf record 1: In field 9: Cannot convert [null] to an Numeric

type. Setting value to null

A problem was found. Events prior to problem:
Table ccf record 1: Field 10: got [2]
Table ccf record 1: Field 11: got [A]
Table ccf record 1: Field 12: got [N]
Table ccf record 1: Field 13: got [N]
Table ccf record 1: Field 14: got [R]
Table ccf record 1: Field 15: got []

Table ccf record 1: Field 16: got [N]
ERROR: IncorrectFieldNumber Table ccf record 1: Number of fields in input data [16] is less than the

number of mandatory fields expected [20] and the total number of fields for this table [20].

++++++++++Summary of Validation Problems++++++++++
IncorrectFieldNumber 33
UseofDefaultWarning 54
NumericConversionWarning 13
=== Table ccf END.

APPENDIX E: CUS DEFINITIONS OF COMMANDS, BUILDING BLOCKS AND AN OBSERVING MODE

Observing Mode Mode_POF1_0

```
mode Mode_POF1_0 {
   int a = 10;
   int b = 2;
   int c = 3;
   int obsmode = 0;
}{
   // Photometer Observatory Function - Chop without jiggling

   Block_Init_0(obsmode);
   Block_PCal_0(b);
   Block_Chop_0(c);
   Block_PCal_0(2 * c - b);
}
```

Building Block Block_Init_0

```
block Block_Init_0 SPIRE 2 {
   int obsmode = 0;
}{
   Cmd_Set_ObsID($OBSID);
   Cmd_Set_ObsMode(obsmode);
}
```

Building Block Block_PCal_0

```
block Block_PCal_0 SPIRE 0 {
   int pcal = 0;
}{
   int a =6;
   Cmd_Set_BBID($BBID);
   delay(a*2);
}
```

Building Block Block_Chop_0

```
block Block_Chop_0 SPIRE 1 {
   int a = 2;

}{
   Cmd_Set_BBID($BBID);
   delay(a*a*a);
```

}

Command Cmd_Set_ObsMode

```
command Cmd_Set_ObsMode SC002480 {
  int obsmode = 0;
} {
}
```


Command Cmd_Set_ObsID

```
command Cmd_Set_ObsID  SC000480{
  int obsid = 0;
} {
}
```


Command Cmd_Set_BBID

```
command Cmd_Set_BBID SC001480 {
  int bbid = 0;
} {
}
```

Problems in using the CUS Editor

- The CUS editor only allows opening of existing definitions bottom up, i.e. the command definitions have to be opened first, followed by the BBs and the observing mode. SPR has been issued.
- The BB Block_Init_0 was created only because the CUS does not allow command definitions to appear directly in the observing mode definition. This BB contains the commands Cmd_Set_ObsID and Cmd_Set_ObsMode which I would expect to appear in the observing mode. SPR has been issued.

APPENDIX F: TEST PROCEDURE FOR TEST CASE TC-ICC-04

```
#
#   TestClient.tcl
#

TCsend SC002480 Checks {DPTV_OFF} ReferBy cmd TCrequestID 0x0 \
   {SP02N480 0 UL}
TCsend SC000480 Checks {DPTV_OFF} ReferBy cmd1 TCrequestID 0x1 \
   {SP00N480 0 UL}
TCsend SC001480 Checks {DPTV_OFF} ReferBy cmd2 TCrequestID 0x2 \
   {SP01N480 0 UL}
WaitTime 5000

# Set Observing Mode
TCsend SC002480 Checks {DPTV_OFF} ReferBy cmd TCrequestID 0x0 \
   {SP02N480 2 UL}
WaitTime 2000
# Set OBSID

TCsend SC000480 Checks {DPTV_OFF} ReferBy cmd1 TCrequestID 0x1 \
   {SP00N480 1 UL}
WaitTime 2000
# Set BBID 1

TCsend SC001480 Checks {DPTV_OFF} ReferBy cmd2 TCrequestID 0x2 \
   {SP01N480 1 UL}
 WaitTime 2000
# Set BBID 2

TCsend SC001480 Checks {DPTV_OFF} ReferBy cmd3 TCrequestID 0x3 \
   {SP01N480 2 UL}
WaitTime 2000
# Set OBSID

TCsend SC000480 Checks {DPTV_OFF} ReferBy cmd1 TCrequestID 0x4 \
   {SP00N480 2 UL}
WaitTime 2000
```

Problems and Anomalies

1. TCL editor: Ctrl+V does not work
2. TCL editor: While copying and pasting the screen did not refresh properly. Only upon scrolling up and down did the screen refresh
3. TCL editor: The undo on a comment string appeared in black and not in green.
4. TOPE: On stopping the EXIF tasks, EXIF_TM1 does not get stopped. It has to be stopped again separately afterwards.

All these problems have been reported to Serge Valera.

APPENDIX G: QLA PLAYBACK ERRORS FROM SPIRE-TC-02

Accessing with params

APID1=1280&NAME=nl.esa.herschel.versant.ccm.TmSourcePacketImpl&APID5=1286&APID4=1285&APID3=1284&APID2=1282

Making a local connection to sds78@truro.bnsc.rl.ac.uk
Query completed, size is 252
Got instance of class nl.esa.herschel.access.db.LocalConnection
Packet 120 is of type Unknown packet type 0
Packet 123 is of type Unknown packet type 0
Packet 124 is of type Unknown packet type 0
Packet 129 is of type Unknown packet type 0
Error getting data from nl.esa.herschel.access.CachedProductStream:

herschel.spire.qla.NoSuchParameterException: Failed to match science frame time parameter to SID 1554

- RTA-mode playback failed (ie database -> router):
> java nl.esa.herschel.access.util.RtaPlayback
java.lang.NullPointerException
        at nl.esa.herschel.access.util.DataSelector$OpenHandler.run(DataSelector.java:348)
        at java.lang.Thread.run(Thread.java:484)