



**Procedure**

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 1 of 19

**SPIRE HCSS Acceptance Test Plan**

**Sunil Sidher, Steve Guest and Asier Aramburu**

**6<sup>th</sup> January 2006**



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 2 of 19

SPIRE HCSS Acceptance Test Plan .....	1
1. INTRODUCTION.....	5
1.1 Change Record .....	5
1.2 Conventions.....	5
2. DOCUMENTATION.....	5
2.1 Applicable Documents .....	5
2.2 Reference Documents.....	6
3. TEST SETUP.....	6
3.1 System Setup .....	6
4. HCSS TEST CASES.....	7
4.1 Generation of TC Definition File and MIB Ingestion thereof.....	7
4.1.1 Test Case Identifier .....	7
4.1.2 Goal.....	7
4.1.3 Input Specifications.....	7
4.1.4 Output Specifications .....	8
4.1.5 Assumptions.....	8
4.1.6 Test Procedure.....	8
4.2 Test Case TC-ICC-02: CUS: TC Definition File Import.....	8
4.2.1 Test Case Identifier .....	8
4.2.2 Goal.....	8
4.2.3 Input Specifications.....	8
4.2.4 Output Specifications .....	8
4.2.5 Assumptions.....	9
4.2.6 Test Procedure.....	9
4.3 Test Case TC-ICC-03: CUS: Observing Mode Import .....	9
4.3.1 Test Case Identifier .....	9
4.3.2 Goal.....	9
4.3.3 Input Specifications.....	9
4.3.4 Output Specifications .....	9
4.3.5 Assumptions.....	9
4.3.6 Test Procedure.....	9
4.4 Test Case TC-ICC-04: TOPE: Issuing Telecommands .....	10
4.4.1 Test Case Identifier .....	10
4.4.2 Goal.....	10
4.4.3 Input Specifications.....	10
4.4.4 Output Specifications .....	10
4.4.5 Assumptions.....	10
4.4.6 Test Procedure.....	10
4.5 Test Case TC-ICC-05: TestControl: Running Test Observation using the HCSS .....	10
4.5.1 Test Case Identifier .....	10
4.5.2 Goal.....	10
4.5.3 Input Specifications.....	10
4.5.4 Output Specifications .....	11
4.5.5 Assumptions.....	11
4.5.6 Test Procedure.....	11
4.6 Test Case TC-ICC-06: End to End: Run a Test Observation, Ingest TM packets, run QLA ....	12
4.6.1 Test Case Identifier .....	12
4.6.2 Goal.....	12
4.6.3 Input Specifications.....	12
4.6.4 Output Specifications .....	12
4.6.5 Assumptions.....	12



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 3 of 19

4.6.6	Test Procedure.....	12
4.7	Test Case TC-ICC-07: End to End: Run RTA/QLA on observation data stored in the HCSS database .....	12
4.7.1	Test Case Identifier .....	12
4.7.2	Goal .....	13
4.7.3	Input Specifications.....	13
4.7.4	Output Specifications .....	13
4.7.5	Assumptions .....	13
4.7.6	Test Procedure.....	13
4.8	Test Case TC-ICC-08: SPIRE Command List functionality in the CUS .....	13
4.8.1	Test Case Identifier .....	13
4.8.2	Goal .....	13
4.8.3	Input Specifications.....	13
4.8.4	Output Specifications .....	13
4.8.5	Assumptions .....	13
4.8.6	Test Procedure.....	13
4.9	Test Case TC-ICC-09: Ingest a Calibration Table and check size functionality in the CUS ....	14
4.9.1	Test Case Identifier .....	14
4.9.2	Goal .....	14
4.9.3	Input Specifications.....	14
4.9.4	Output Specifications .....	14
4.9.5	Assumptions .....	14
4.9.6	Test Procedure.....	14
4.10	Test Case TC-ICC-10: Correct setting of the OBSID values according to Site Id in the CUS14	
4.10.1	Test Case Identifier .....	14
4.10.2	Goal .....	15
4.10.3	Input Specifications.....	15
4.10.4	Output Specifications .....	15
4.10.5	Assumptions .....	15
4.10.6	Test Procedure.....	15
4.11	Test Case TC-ICC-11: Correct setting of command parameter values in the CUS .....	15
4.11.1	Test Case Identifier .....	15
4.11.2	Goal .....	15
4.11.3	Input Specifications.....	15
4.11.4	Output Specifications .....	15
4.11.5	Assumptions .....	15
4.11.6	Test Procedure.....	15
4.12	Test Case TC-ICC-12: TC History and OOL data retrieval and ingestion .....	16
4.12.1	Test Case Identifier .....	16
4.12.2	Goal .....	16
4.12.3	Input Specifications.....	16
4.12.4	Output Specifications .....	16
4.12.5	Assumptions .....	16
4.12.6	Test Procedure.....	16
4.13	Test Case TC-ICC-13: Overnight test at nominal SPIRE data rate .....	17
4.13.1	Test Case Identifier .....	17
4.13.2	Goal .....	17
4.13.3	Input Specifications.....	17
4.13.4	Output Specifications .....	17
4.13.5	Assumptions .....	17
4.13.6	Test Procedure.....	17
4.14	Test Case TC-ICC-14: Database Replication Test .....	18
4.14.1	Test Case Identifier .....	18



**Procedure**

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 4 of 19

---

4.14.2	Goal .....	18
4.14.3	Input Specifications.....	18
4.14.4	Output Specifications .....	18
4.14.5	Assumptions .....	18
4.14.6	Test Procedure.....	18
4.15	Test Case TC-ICC-15: Playback data from database into the EGSE router .....	19
4.15.1	Test Case Identifier: .....	19
4.15.2	Goals .....	19
4.15.3	Input Specifications.....	19
4.15.4	Output Specifications .....	19
4.15.5	Test Procedure.....	19



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 5 of 19

## 1. INTRODUCTION

This acceptance test plan is based on the original HCSS Acceptance Test Plan (RD01). It was successfully used by SPIRE at RAL on the Herschel Common Science System (HCSS) version 0.1.4 release candidate #1 (Build #263) and also on earlier versions of the HCSS. The documentation listed in section 2 is expected to be used for the preparation and execution of the tests. Section 3 describes the RAL system setup and configuration for these tests.

In addition to the standard test cases the philosophy behind the latest acceptance test plan is to

- incorporate specific tests for SPIRE requested functionality in the HCSS
- include further tests for SPRs detected by or directly affecting SPIRE

Following each acceptance test a test report will be issued and the acceptance test plan will be updated as appropriate to include new tests for problems encountered or changes requested.

### 1.1 Change Record

Date	Issue	Changes
14 <sup>th</sup> June 2005	1.0	First version  Test case TC-ICC-02 is now obsolete

### 1.2 Conventions

Commands entered are in indented courier new font:

```
dbuser -add -P egse@lichfield
```

Output from the system is in the same font but starting without indentation:

```
Exception in thread "main" java.lang.NullPointerException  
    at herschel.cus.gui.CusEditor.<init>(CusEditor.java:98)  
    at herschel.cus.gui.CusEditor.main(CusEditor.java:1332)
```

## 2. DOCUMENTATION

### 2.1 Applicable Documents

AD01	Database MIB Import ICD, Version 5.2, 2 <sup>nd</sup> July 2003, S2K-MCS-ICD-0001-TOS-GIC
AD02	CUS Functionality to Support Command Lists for SPIRE, HCSS SCR-0110 (see <a href="http://www.rssd.esa.int/herschel_webapps/servletsuite/ProblemReportServlet?area=hcss&amp;mode=displaypr&amp;id=110">http://www.rssd.esa.int/herschel_webapps/servletsuite/ProblemReportServlet?area=hcss&amp;mode=displaypr&amp;id=110</a> ), Ken King and Sunil Sidher, 13 <sup>th</sup> Aug 2002.
AD03	Contents of SPIRE VM Table File, Issue 1.0 Draft 2, 5 <sup>th</sup> Feb 2004, SPIRE-RAL-NOT-001907
AD04	SPIRE EGSE-ILT Startup Procedures, Issue 0.7, 24 <sup>th</sup> June 2003, SPIRE-RAL-DOC-001630



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 6 of 19

## 2.2 Reference Documents

RD01	HCSS Acceptance Test Plan, Issue 1.0, 14 <sup>th</sup> June 2005
RD02	<a href="#">MIB User Guide</a>
RD03	<a href="#">HCSS telemetry ingestion software user manual</a>
RD04	<a href="#">Common Uplink System User's Guide</a>
RD05	<a href="#">HCSS Installation Instructions</a>
RD06	EGSE TestControl User Guide, Draft 5, 16 <sup>th</sup> Dec 2003, PICC-ME-MN-002

## 3. TEST SETUP

### 3.1 System Setup

The SPIRE setup for an acceptance test would typically be as follows:

**Lincoln:** SCOS 2000 w/s running Linux SuSE 7.3, SCOS2000 v2.3e P5 + TOPE, EGSE part of the TestControl and the latest HCSS User Release

**Lichfield:** Data Server running Linux SuSE 9.3 and latest HCSS User Release.

**Salisbury:** Linux machine running a schema compatible version of QLA.

Lincoln, Lichfield and Salisbury should have agreed versions of Java and Versant installed.

The tests are to be run from user account sg55@lincoln and egse@lichfield with setting as "tester".

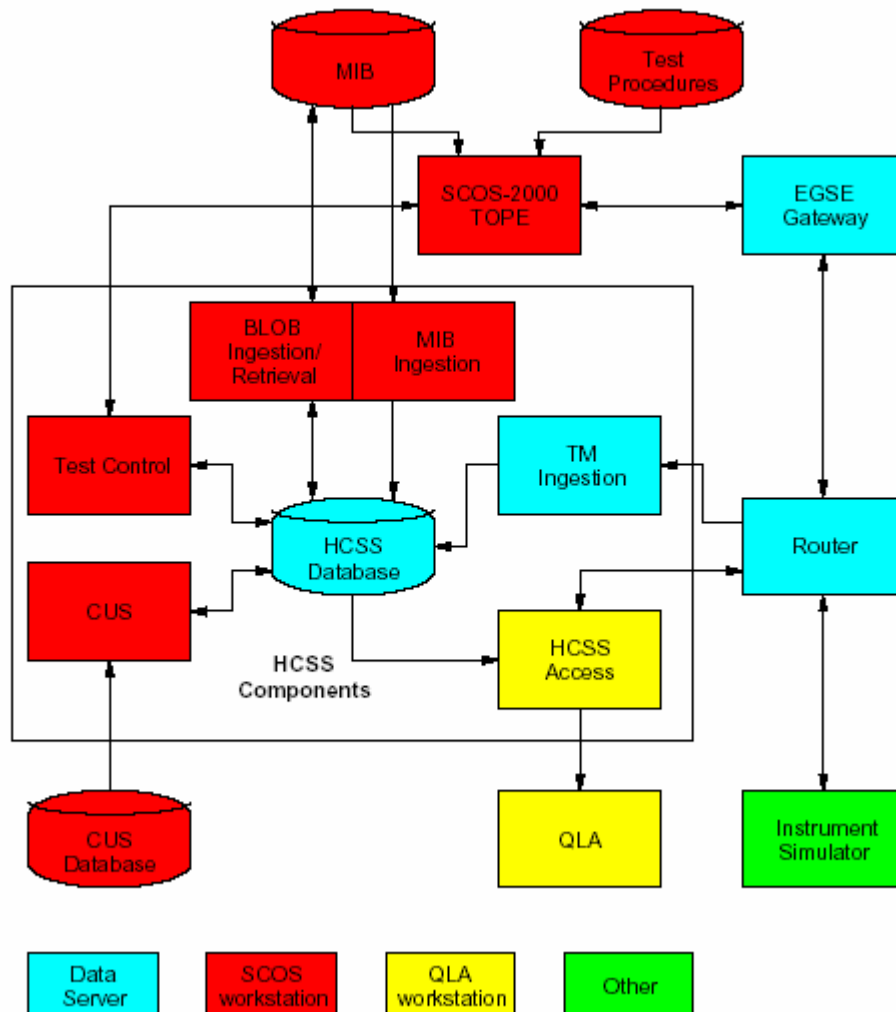


Figure 1: Configuration for the HCSS Acceptance Test

## 4. HCSS TEST CASES

### 4.1 Generation of TC Definition File and MIB Ingestion thereof

#### 4.1.1 Test Case Identifier

TC-ICC-01

#### 4.1.2 Goal

To ingest a set of MIB data files describing the behaviour of the SPIRE instrument into the HCSS system.

#### 4.1.3 Input Specifications

ILT instrument MIB.



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 8 of 19

### 4.1.4 Output Specifications

MIB ingested into the HCSS.

### 4.1.5 Assumptions

- The MIB to be ingested has been successfully imported into SCOS.
- A versant database called HCSSv<j.k.l>\_AT for the HCSS acceptance test has been created, where <j.k.l> represents the HCSS version to be tested.
- There is enough disk space available for the data to be generated during the acceptance test.

### 4.1.6 Test Procedure

Follow the HCSS MIB user guide to reach the goal.

- Check that the Site ID is set correctly (for SPIRE ILT it should be 3)
- Create a directory for the MIB on sg55@lincoln
- In this directory create subdirectories ascii-tables and auxil
- Copy the MIB from SCOS into the ascii-tables directory
- Set the current working directory one level above the ascii-tables directory
- Run the command:  

```
gencmddurns ascii-tables auxil/tc-durns
```
- Check for any error messages
- Run the program to check the MIB  

```
mibchecker -properties hcss.mib.datadir=$PWD
```
- Run the command to ingest the MIB into the HCSS  

```
ingestmib -properties hcss.mib.datadir=$PWD
```
- Check warning messages – usually warnings relating to default values of certain MIB parameter fields are not serious.
- Start the CUS GUI  

```
cusgui &
```
- Select the “Load a new MIB” option from the Mib tab. A popup window will appear with “Version1” of the ingested MIB. Click on “Version1” and press the Select button. When prompted make this the default version of the MIB and commit it to the database.

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

### 4.2.1 Test Case Identifier

TC-ICC-02

### 4.2.2 Goal

Use the TC Definition File generated by the MIB service to import the TC definitions from the MIB into CUS. **This test case is now obsolete as TC-ICC-01 performs this task.**

### 4.2.3 Input Specifications

Instrument MIB and TC-ICC-01.

### 4.2.4 Output Specifications

TC definitions into CUS





## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 9 of 19

### 4.2.5 Assumptions

- The MIB has been successfully ingested into the HCSS.

### 4.2.6 Test Procedure

- Start the CUS GUI:  
`cusgui &`
- Select the "Import ..." button from the "Registry" menu in the menu bar.
- In the dialog step to the "ascii-tables" directory specified in TC-ICC-01 and select the `cus.script` file. Press the "Open" button.
- Select the "Commit" button from the "Registry" menu in the menu bar. Press "OK" in the dialog.
- Select the "Quit" button from the "File" menu in the menu bar. Press "Quit" in the dialog.
- Check that the TC definitions are correctly displayed in the text box entitled "MIB Command Definitions"
- Verify that the TC definitions appear in the database by running the `vdbinspector` program.

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

### 4.3.1 Test Case Identifier

TC-ICC-03

### 4.3.2 Goal

Enter the necessary observing mode and building block definitions into the DB registry.

### 4.3.3 Input Specifications

ASCII file (previously exported from the CUS registry) containing the observing mode and building block definitions necessary for the acceptance test.

### 4.3.4 Output Specifications

Observing mode and building block definitions are in the CUS registry.

### 4.3.5 Assumptions

### 4.3.6 Test Procedure

- Start the CUS GUI:  
`cusgui &`
- Select the "Import ..." button from the "Registry" menu in the menu bar.
- In the dialog step to the directory containing the ASCII file of observing mode and building block definitions. Select the file and press the "Open" button.
- Select the "Commit" button from the "Registry" menu in the menu bar. Press "OK" in the dialog.
- Select the "Quit" button from the "File" menu in the menu bar. Press "Quit" in the dialog.
- Check that the observing mode and building block definitions are correctly displayed in the text box entitled "Registry Definitions".



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 10 of 19

- Verify that the observing mode and building block definitions appear in the database by running the `vdbinspector` program.

### 4.4 Test Case TC-ICC-04: TOPE: Issuing Telecommands

#### 4.4.1 Test Case Identifier

TC-ICC-04

#### 4.4.2 Goal

Execute a simple test procedure via TOPE which *does not* involve the HCSS. This test is intended to prove the functionality of SCOS 2000 and its External Interfaces.

#### 4.4.3 Input Specifications

A TOPE script written in TCL in compliance with the document EGSE based on SCOS 2000, EGSE and Mission Control System Software User Manual, TOS-EMG/01-1029/bm/sv EMCS SUM, (Version for SCOS 2000, 2.3eP5 is Issue 4, Revision 0, 9<sup>th</sup> March 2004).

#### 4.4.4 Output Specifications

Commands issued from the TOPE script appear in the SCOS 2000 TC History Display.

#### 4.4.5 Assumptions

The version of SCOS 2000 for tests has already been installed, correctly configured and acceptance tested.

#### 4.4.6 Test Procedure

- Start the EGSE following the start up procedure defined in DOC??. This involves starting the CDMS simulator, the EGSE router and gateway, SCOS 2000 and TOPE.
- From the TOPE Command Window execute the TOPE script `TOPE_Test.tcl`
- Monitor the TC History Display screen on SCOS 2000 and observe the commands as they are transmitted.
- If all the commands in the `TOPE_Test.tcl` script appear in the TC History Display then the test is successful.

### 4.5 Test Case TC-ICC-05: TestControl: Running Test Observation using the HCSS

#### 4.5.1 Test Case Identifier

TC-ICC-05

#### 4.5.2 Goal

Execute a simple test procedure via TOPE to demonstrate the interaction between TOPE and the HCSS (TestControl).

#### 4.5.3 Input Specifications

A TOPE script written in TCL in compliance with the document EGSE based on SCOS 2000, EGSE and Mission Control System Software User Manual, TOS-EMG/01-1029/bm/sv



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 11 of 19

EMCS SUM, (Version for SCOS 2000, 2.3eP5 is Issue 4, Revision 0, 9<sup>th</sup> March 2004).

### 4.5.4 Output Specifications

Commands issued in response to execution of the HCSS TOPE script appear in the SCOS 2000 TC History Display.

### 4.5.5 Assumptions

- TC-ICC-04 has been successfully run.
- **Latest version of the EGSE part of the TestControl is installed on sops23e@lincoln – this will need to be checked with Erich Wiezorrek (MPE) before the start of the acceptance test.**

### 4.5.6 Test Procedure

- Start the EGSE following the start up procedure defined in AD04. This involves starting the CDMS simulator, the EGSE router and gateway, SCOS 2000 and TOPE. This procedure has instructions on starting the TestControl Server but they are also included here since this test is designed to test this functionality:
  1. From the SCOS 2000 sops23e account on lincoln, ensure that the environment variable TCH\_TCL\_DISPLAY is set to lincoln:

```
setenv TCH_TCL_DISPLAY 130.246.32.24:0.0
```
  2. From the SCOS 2000 sops23e account on lincoln, issue the command

```
xauth list
```

Several lines of output will be displayed. Copy into a buffer the entire contents of the line beginning with the text `lincoln.bnsc.rl.ac.uk:0`
  3. From the SCOS 2000 sops23e account on lincoln, log in as user sg55 using ssh -X.
  4. From sg55@lincoln execute the xauth add command to set up the magic cookie:

```
xauth add <Magic Cookie >
```

where <Magic Cookie> represents the contents of the copy buffer mentioned in 1.

For example:

```
xauth add lincoln.bnsc.rl.ac.uk:0 MIT-MAGIC-COOKIE  
1f26cc4296fe01149c14dc05ba38f7f1a
```

5. Start the TestControl server by typing:

```
testcontrol-server
```
- From the TOPE Command Window import the local TOPE script TOPE\_HCSSTest.tcl into the HCSS.
  - Execute the imported TOPE\_HCSSTest.tcl script from the HCSS.
  - Monitor the TC History Display screen on SCOS 2000 and observe the commands as they are transmitted.
  - If all the commands in the TOPE\_HCSSTest.tcl script appear in the TC History Display then the test is successful.



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 12 of 19

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

### 4.6.1 Test Case Identifier

TC-ICC-06

### 4.6.2 Goal

Perform the end-to-end test. A test observation defined in the CUS system is executed via TOPE/TestControl. TM packets generated by the TestClient triggered by telecommands from this test observation are ingested into the HCSS database (tmingestion). SCOS and QLA are used to display the resulting TM packets.

### 4.6.3 Input Specifications

SCOS 2000, Instrument or instrument simulator, HCSS and QLA.

### 4.6.4 Output Specifications

Working QLA.

### 4.6.5 Assumptions

- Instrument or instrument simulator is available (e.g. DRCU Simulator or the Warm Electronics).
- TC-ICC-05
- SCOS 2000, TOPE, TestControl Server are already running

### 4.6.6 Test Procedure

- Startup the Instrument or instrument simulator, i.e. the DRCU simulator or the Warm Electronics
- Start up QLA from salisbury:  
qla
- Startup the HCSS TM Ingestor:  
tmingestion
- From the TOPE Command Window import the local TOPE script HCSSTest.tcl into the HCSS.
- Using TOPE execute the test procedure HCSSTest.tcl
- Watch the issued Telecommands in the TC History Display.
- Watch the incrementing TM sequence counters using the Packet Display and SCOS 2000 TM Desktop
- Watch for errors in the tmingestion console.
- Use QLA displays to monitor test execution as appropriate.

## 4.7 Test Case TC-ICC-07: End to End: Run RTA/QLA on observation data stored in the HCSS database

### 4.7.1 Test Case Identifier

TC-ICC-07



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 13 of 19

### 4.7.2 Goal

Test the retrieval of data from the database and run QLA on it.

### 4.7.3 Input Specifications

HCSS and QLA.

### 4.7.4 Output Specifications

Working QLA.

### 4.7.5 Assumptions

- TC-ICC-06

### 4.7.6 Test Procedure

- Start up QLA from salisbury:
  - qla
    - Select local database playback mode
    - Select OBSID
    - Create QLA displays as appropriate
    - Start playback and monitor displays
- Repeat test by selecting time range instead of OBSID (clear the OBSID field first)

## 4.8 Test Case TC-ICC-08: SPIRE Command List functionality in the CUS

### 4.8.1 Test Case Identifier

TC-ICC-08

### 4.8.2 Goal

To check that the SPIRE requested command list functionality (see AD02) is available in the CUS to take a Virtual Machine (VM) table, associate a building block with it and store it in the CUS registry.

### 4.8.3 Input Specifications

Command list VM table.

### 4.8.4 Output Specifications

Command list definition in the CUS

### 4.8.5 Assumptions

- The command list VM table name and format are specified in accordance with AD03.
- HCSS CUS is available.

### 4.8.6 Test Procedure

- Start the CUS GUI
  - cusgui &
- From the Registry tab select Create Command List Definition
  - Assign a Building Block (BB) ID of 770 for the Update Table command selection.



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 14 of 19

- Select table Flash\_Table\_2.1.3\_1.1\_050127112906.tbl from directory /home/sops23e/SCOS2.3eP5/tcl/TC/VMTables/Table070-Flash/TC on lincoln.
- If no errors or warnings appear save the definition and commit to the registry from the Registry tab.

### **4.9 Test Case TC-ICC-09: Ingest a Calibration Table and check size functionality in the CUS**

#### **4.9.1 Test Case Identifier**

TC-ICC-09

#### **4.9.2 Goal**

- To check that a calibration can be ingested in the CUS
- To check the number of rows in a calibration table stored in the CUS can be retrieved and used in ILT, IST and operations. Without this functionality an external TCL application resident on the SCOS machine is needed together with a copy of the calibration table to read its size.

#### **4.9.3 Input Specifications**

HCSS CUS

#### **4.9.4 Output Specifications**

- Ingested calibration table in the HCSS CUS.
- Size of calibration table

#### **4.9.5 Assumptions**

- A calibration table is available.
- CUS GUI is available.

#### **4.9.6 Test Procedure**

- Checkout table PLWNominalSettings.txt from CVS for specific model as appropriate.
- Start the CUS GUI  
`cusgui &`
- From the Registry tab select Upload calibration table select the PLWNominalSettings.txt table and commit to registry.
- Run CUS procedure to read the calibration table and report its size by selecting the Duration tab.
- If no errors or warnings appear save the definition and commit to the registry from the Registry tab.

### **4.10 Test Case TC-ICC-10: Correct setting of the OBSID values according to Site Id in the CUS**

#### **4.10.1 Test Case Identifier**

TC-ICC-10



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 15 of 19

### 4.10.2 Goal

To check that the OBSID is set correctly according to the Site ID property.

### 4.10.3 Input Specifications

HCSS CUS.

### 4.10.4 Output Specifications

HCSS CUS commands.

### 4.10.5 Assumptions

CUS GUI is available

### 4.10.6 Test Procedure

- Set the hcsc.ccm.siteid to the value to be tested (e.g. SPIRE ILT is 3, SPIRE IST is 11)
- Start the HCSS CUS  
cusgui &
- Select ModeDcuFunc01 and open definition
- Click on the Commands tab and accept the default values of parameters.
  - Check that the commands are generated in a text window
  - Check the OBSID value for the test corresponds to the Site ID.

## 4.11 Test Case TC-ICC-11: Correct setting of command parameter values in the CUS

### 4.11.1 Test Case Identifier

TC-ICC-11

### 4.11.2 Goal

To check that the parameter values of individual commands do not get corrupted.

### 4.11.3 Input Specifications

HCSS CUS.

### 4.11.4 Output Specifications

HCSS CUS commands.

### 4.11.5 Assumptions

- CUS GUI is available

### 4.11.6 Test Procedure

- Start the HCSS CUS  
cusgui &
- Select ModeDcuFunc01 and open definition
- Click on the Commands tab and accept the default values of parameters.



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 16 of 19

- Check that the commands are generated in a text window
- Check the command parameter values for each command have the expected values.

## 4.12 Test Case TC-ICC-12: TC History and OOL data retrieval and ingestion

### 4.12.1 Test Case Identifier

TC-ICC-12

### 4.12.2 Goal

To retrieve TC history and OOL data files from SCOS 2000 and ingest them into the HCSS.

### 4.12.3 Input Specifications

SCOS 2000 and HCSS.

### 4.12.4 Output Specifications

TC history and OOL objects in the HCSS database.

### 4.12.5 Assumptions

- TC history and OOL data are available.
- HCSS tools for TC history and OOL data retrieval/ingestion are available.
- SCOS 2000 is already running

### 4.12.6 Test Procedure (Manual)

- Log on as sg55@lincoln and retrieve TC history and OOL data  
tchoolretrive
- From the popup window select the datatype to be Telecommand History
  - Select time range as appropriate.
  - Click on retrieve file and make a note of output file created.
- Repeat above step with datatype Out-of-Limits
- Logon as egse@lichfield (if not already)
- Start sftp and go to the THF\_DAT directory on lichfield  
sftp sg55@lincoln  
lcd THF\_DAT
- On lincoln go to the directory where the TC history and OOL data files are stored  
cd /spired/HPR
- Copy the TC history files across  
get <THF\_YYMMDD\_nnnn.DAT>  
where nnnn is a counter
- Now go to the ORF\_DAT directory on lichfield  
lcd ../ORF\_DAT
- Copy the OOL files across  
get <ORF\_YYMMDD\_nnnn.DAT>  
where nnnn is a counter
- Ingest the TC history file  
tchingest <THF\_YYMMDD\_nnnn.DAT>
- Check for ingestion errors in response to tchingest command
- Ingest the OOL file  
oolingest <ORF\_YYMMDD\_nnnn.DAT>





## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 17 of 19

- Check for ingestion errors in response to oolingest command.
- Use `vdbinspector` to check for associations of commands with building blocks and observations.
- Use `vdbinspector` to check for associations of out of limit parameters with building blocks and observations.

### 4.12.7 Test Procedure (Automatic)

- Log on as `sg55@lincoln`
- From the same console start TCH SERVER application by typing `thfserver`
- On another console on `sg55@lincoln` start OOL SERVER application by typing `orfserver`
- On `egse@lichfield` type
  - `tchingest` and on one console
  - `oolingest` on another consoleIn this configuration TCH and OOL data will be ingested continuously into the DB on Lichfield.
- Send a few TCs like `SET_OBSERVATION_STEP`, etc.
- Using TOPE execute the test procedure `HCSSTest.tcl`
- Monitor the messages appearing on the `tchingest` console to confirm that the newly executed TCs have been ingested in the DB.
- If there are TM parameters going in and out of limits then `oolingest` console will also display messages confirming the ingestion of OOL data in the DB.

## 4.13 Test Case TC-ICC-13: Overnight test at nominal SPIRE data rate

### 4.13.1 Test Case Identifier

TC-ICC-13

### 4.13.2 Goal

Perform an overnight test at nominal SPIRE data rate to

- Check that TM ingestion queue sizes remain within reasonable limits.
- Check that the database remains operational.
- Check that QLA displays remain operational and do not lag behind.

### 4.13.3 Input Specifications

SCOS 2000, Instrument or Instrument Simulator, HCSS and QLA.

### 4.13.4 Output Specifications

HCSS

### 4.13.5 Assumptions

- Instrument or instrument simulator is available (e.g. DRCU Simulator or the Warm Electronics).
- TC-ICC-05
- SCOS 2000, TOPE, TestControl Server are already running

### 4.13.6 Test Procedure

- Startup the Test Client, i.e. the DRCU simulator or the Warm Electronics



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 18 of 19

- Start up QLA from salisbury:  
qla
- Startup the HCSS TM Ingestor from egse@lichfield:  
tmingestion |&  
/home/egse/tmingest\_logs/HCSSv<j.k.l>\_AT.log
- From the TOPE Command Window import the local TOPE script HCSSNightTest.tcl into the HCSS.
- Using TOPE execute the test procedure HCSSNightTest.tcl
- Watch the initially issued Telecommands in the TC History Display.
- Briefly watch the incrementing TM sequence counters using the Packet Display and SCOS 2000 TM Desktop
- Briefly watch for errors in the tmingestion console.
- Use QLA displays to initially monitor test execution as appropriate.
- On the following day examine the TM ingestion log to look for error messages.
- Run vdbinspector to see if all TM objects are present in the database.

### 4.14 Test Case TC-ICC-14: Database Replication Test

#### 4.14.1 Test Case Identifier

TC-ICC-14

#### 4.14.2 Goal

The database is correctly replicated from the AIV machine to the ICC machine.

#### 4.14.3 Input Specifications

Databases on the AIV and ICC machines

#### 4.14.4 Output Specifications

Replicated database on the ICC machine.

#### 4.14.5 Assumptions

Replication has been pre-configured between the two databases.

#### 4.14.6 Test Procedure

- Run the tool to list the contents of the AIV database  
db2tty -d <Database Name>@<AIV machine> >  
AIV\_DB\_HCSSv<j.k.l>\_AT.txt
- Run the tool to list the contents of the ICC database  
db2tty -d <Database Name>@<AIV machine> >  
ICC\_DB\_HCSSv<j.k.l>\_AT.txt
- Compare the contents of these two output files using the Unix diff command  
diff AIV\_DB\_HCSSv<j.k.l>\_AT.txt ICC\_DB\_HCSSv<j.k.l>\_AT.txt  
> AIV\_ICC\_DBs\_Diff\_HCSSv<j.k.l>\_AT.txt
- Examine the contents of the differences file created above. The only differences should be minor ones, e.g. machine name.



## Procedure

SPIRE HCSS Acceptance Test Plan  
Sunil Sidher, Steve Guest & Asier Aramburu

**Ref:** SPIRE-RAL-PRC-002456  
**Issue:** 1.1  
**Date:** 6<sup>th</sup> January 2006  
**Page:** 19 of 19

### 4.15 Test Case TC-ICC-15: Playback data from database into the EGSE router

#### 4.15.1 Test Case Identifier:

TC-ICC-15

#### 4.15.2 Goals

- To test the ability to simulate near real time reception using playback.
- To test that the sequence of packets observed is the same as the original data in real time. i.e. that it is correctly time ordered.
- To stress test the performance of the TM ingestor and the router.

#### 4.15.3 Input Specifications

HCSS, QLA and database containing SPIRE TM.

#### 4.15.4 Output Specifications

SPIRE acceptance test database containing re-ingested TM, which is unlikely to be useful after this test.

#### 4.15.5 Test Procedure

- Stop TM ingestion
- Stop SCOS 2000
- Create a new database to store TM played back through the router
- Start QLA and select EGSE router as data source
- Select OBSID and BBFULLTYPE HK parameters from the QLA parameter selector
- Start Packet Display
- Start RTA playback tool (`rtaplayback`)
- Select time range or an OBSID to playback through the router
- Set the initial speed of playback to 10Hz (if science data present, otherwise 1-2Hz)
- Start playback
- Monitor QLA parameter values of OBSID and BBFULLTYPE.
- Ensure that the QLA times and RTA playback times are consistent and that they update at the specified playback speed.
- Start TM ingestor (`tmingestion`) – ensure that it starts correctly and connects to the router.
- Wait until the TM ingestor issues a commit - check the queue sizes and any error messages
- Check the router display for error messages.

Increase the playback frequency in steps of 10-20Hz (up to a maximum of at least 100Hz) – keep monitoring TM ingestor and router displays for errors and warnings.