



SUBJECT: **Operational Day Logging Tool Test Plan**

PREPARED BY: Edward Polehampton, Markos Trichas

DOCUMENT No: SPIRE-RAL-DOC-003093

ISSUE: 1.0 **Date:** 23 July 2008

APPROVED BY: **Date:**



TABLE OF CONTENTS

TABLE OF CONTENTS3

1. INTRODUCTION4

 1.1 SCOPE4

 1.2 STRUCTURE OF DOCUMENT4

 1.3 DOCUMENTS4

 1.3.1 *Applicable Documents*.....4

 1.3.2 *Reference Documents*.....4

2. OPERATIONAL DAY LOGGING TEST PROCEDURE.....5

 2.1 OBSERVATION LOGGING SCRIPT TEST PLAN5

 2.2 PROCEDURE FOR TESTING HEALTH MONITORING SCRIPTS.....6

3. APPENDIX 1: INPUT FILE SPECIFYING OD START AND END TIMES7



1. INTRODUCTION

1.1 Scope

This document describes the test plan for operational day logging tools to be run during SPIRE operations.

Operational day (OD) processing consists of 3 main tasks that must be carried out every day:

- Run Health Monitoring Scripts for one OD
- Run Observation Logging Script for one OD
- Run Trend Analysis Processing Script(s)

These will create a series of logs, product pools and web page entries. This document describes the test plan for the first two tasks: health monitoring and the observation log.

1.2 Structure of Document

The test plan is described for the Health Monitor Logs and the Observation Log. Input files needed for the tests are given in the Appendix. Procedures for running the individual tasks are referenced to the ICC Operations Manual.

1.3 Documents

1.3.1 Applicable Documents

AD1 Operational Day Logging Tools Requirements, SPIRE-RAL-DOC-003092

1.3.2 Reference Documents

RD1 ICC Operations Manual



2. OPERATIONAL DAY LOGGING TEST PROCEDURE

The starting configuration for the tests is assumed to be the following:

- Scripts for logging are stored in CVS in the `tools/scripts/` directory. These are tagged so that the latest version of the scripts is inside the build in the `tools/scripts/` directory.
- The scripts should be run from Leicester in own user directory (using the script from inside the build)
- The local store directory should be defined by a property in the `user.props` file
- Data files should be written to Wakefield into a directory that should be defined by a property in the `user.props` file

2.1 Observation Logging Script Test Plan

The procedure for running this test is:

- Locate the `observationLogScript.py` script in the build in the `herschel/spire/tools/scripts` directory
- Check the property that defines where the pool is written in local store (currently reads this property from `user.props`). For the ICC DP acceptance test this should be set to:
`hcss.ia.pal.pool.lstore.dir=/misc/winchester1/dptest063/data/products`
- Add the logging properties described above to `user.props`:
`loggingDir=/misc/winchester1/dptest063/operations/logs`
`loggingDatabase=pfm4_test@chesterfield`
`loggingTimesFile=/home/spire/etp87/icc_testing/operational_days.txt`
- Check there is a file specifying start and end times at the location pointed to by the property (see Appendix for contents of this file)
- Run the script (`observationLogScript.py`) following the procedure in RD1
- It does one query to the database for every OBSID it finds
- It should write out the following files:
 - `PFM4_test_DailyLog_<dayNum>.csv`
 - `combinedLog.csv`
 - `observationLog.html`
- It should write the following pools:



- PFM4_test_daily_log
- PFM4_test_combined_log
- Inside the CVS files there should be one row per OBSID.
- Inside the log products there should be a TableDataset containing one row per OBSID
- The script should not crash on an operational day which does not have any observations in it. This should be tested by using a time range which only has housekeeping but no observations (an over weekend test is used in the example given in the Appendix)
- Should verify that output created OK by:
 - Checking the CSV logs against the Excel log for that database on the Test Team Web site - are all of the expected tests present in the CSV log? All OBSID's should be present
 - Checking the pool created by browsing it using the Product Access Layer Browser
 - Checking that the web page is visible

2.2 Procedure for testing Health Monitoring Scripts

- Currently only two Health Monitoring scripts exist. These produce:
 - Event logs
 - Transparent packet logs
- The script writes two .txt files called:
 - DRCUEvents_<startDate>_<endDate>.txt
 - TransparentPackets_<startDate>_<endDate>.txt
- The script is called `getEvents.py` in `spire/tools/scripts/`

It requires the same 3 properties set as the observation logging script (see previous section), ie:

```
loggingDir=/misc/winchester1/dptest063/operations/logs
```

```
loggingDatabase=pfm4_test@chesterfield
```

```
loggingTimesFile=/home/spire/etp87/icc_testing/operational_days.txt
```

- Once these properties are set, run the script in JIDE (or HIPE)
- It reads the start and end times from the same file as observation logging script
- Check that two output files are created for each OD



3. APPENDIX 1: INPUT FILE SPECIFYING OD START AND END TIMES

The following file can be used for the tests described in this document with the pfm4_test database:

`operational_days.txt`

```
1, 2006-12-08T09:00:00.000000 TAI (1543647633000000), 2006-12-09T09:00:00.000000 TAI  
(1543658433000000)  
2, 2006-12-09T09:00:00.000000 TAI (1543647633000000), 2006-12-10T09:00:00.000000 TAI  
(1543658433000000)  
3, 2006-12-10T09:00:00.000000 TAI (1543647633000000), 2006-12-11T09:00:00.000000 TAI  
(1543658433000000)  
4, 2006-12-11T09:00:00.000000 TAI (1543647633000000), 2006-12-12T09:00:00.000000 TAI  
(1543658433000000)  
5, 2006-12-12T09:00:00.000000 TAI (1543647633000000), 2006-12-13T09:00:00.000000 TAI  
(1543658433000000)  
6, 2006-12-13T09:00:00.000000 TAI (1543647633000000), 2006-12-14T09:00:00.000000 TAI  
(1543658433000000)
```