SPIRE

Trend analysis system test plan

Prepared by: **Peter Hargrave**

Document number:	**SPIRE-UCF-NOT-003137**		
Issue:	**Issue 1**	**Date:**	**6th August 2008**

Contents

1 Reference documents

RD1 - SPIRE trend analysis system (SPIRE-UCF-DOC-003136)
RD2 - SPIRE trend analysis requirements (SPIRE-RAL-DOC-002815)

2 Introduction

This document details the plan for testing the SPIRE trend analysis system.
In its current issue, the document only covers the test plan for the automatic TA monitoring system (RD1), and not the manual database query system (RD2). The manual database system is intended for follow-up investigations once a potential problem has been flagged by the automatic system.
The automatic system is based around a Python script that will be run as a routine daily scheduled task. This script will extract data from the database, and create TA products, according to rules and formats specified in RD1.

3 SOVT trend analysis test plan

At present, only the SPIRE photometer will be fully tested during SOVT-1, due to the orientation of the spacecraft.

3.1 Scripts to be run in SOVT-1

The trend analysis script to be run is SPIRE_TA.py, and is located on the CVS system under develop/main/Herschel/spire/tools/scripts.

3.2 Test scheduling

The script will be run as a manually-called task at various times throughout SOVT testing.
It should be run according to the following criteria:-

- Following a cooler-recycle operation on day n, the script should be run once on day n+1, and once on day n+2.
- The day following any BSM operations.
- The day following any PCal operations.
- On one day, towards the end of SOVT testing, the script should be run twice. This is to check that no duplicate products are produced.

3.3 Functionality to be tested

The script to be tested will be a scaled-down version of the final TA system in terms of the data analysis and presentation. Also, some parameters required for the final system may not be available, such as spacecraft data (telescope temperatures) and data from the PCal pipeline module. However, all functional elements of the final system will be tested.
The script will be run to check for:-

- Successful extraction of data, and creation of products, according to the formatting and rules specified in RD1.
- Robust, bug-free operation.
- Successful creation of sample plots
- Successful creation of warning flags where necessary

3.3.1 Data handling

The script will extract data and populate products as specified in RD1, with the following caveats:-

- TASubKcooler product – all parameters will be recorded except TC123TEMP, PSWTEMP, PMWTEMP, PLWTEMP, SSWTEMP, and SLWTEMP. This is because the derivation of these temperatures has not yet been implemented in the relevant pipeline modules.
- TAPCalPDet and TAPCalSDet products – these products will only be populated if the relevant pipeline modules are available at the time of testing, specifically ScalPhotPcal and SCalSpecPcal.

The script is written in such a way that it may be run more than once a day without creating duplicate products. For instance, a new TASubKcooler product will only be produced if the script finds that a new

BBID/OBSID related to a new cooler recycle operation has been produced since the generation of the last product.

3.3.2 Sample plots

The script will, for the purposes of testing, just produce two sample plots. These are:-

- From the TASubKcooler product, plot PUMPHTRTEMP, PUMPHSTEMP, EVEPHSTEMP, SHUNTTEMP, SUBKTEMP vs. time for that particular cooler recycle (Start of recycle (BBID) to start+48hrs)
- From the TABSM product, plot CHOPMOTORCURR, CHOPSENSSIG, JIGGMOTORCURR, JIGGSENSSIG vs. time for that BSM operation.

3.3.3 Warning flags

Each time the script runs, a log file will be generated which will contain any warning flags for parameters outside their expected range. The exact list of parameters and ranges is still TBD, but for the purposes of testing the script, we will monitor only SubKTemp, and raise a flag whenever this value rises above 300mK.