Control System for the IRAM 30m Online Data Processing under Unix

Title: Online Data Processing
Identifier—MasterURL:
http://www.iram.es/IRAMES/documents/projectDataProcOnline/
File:
Revision: UNDER CONSTRUCTION, (needs review of section implementation and use cases)1.1.4
Revision Date: 2002-06-14
Expiration Date:
Supersedes: not applicable
Is superseded by: not applicable
Authors: W. Brunswig
Contributors: H. Ungerechts, A. Sievers
Affiliations:
Addresses:
Audience: technical staff, astronomers
Publisher: IRAM, Granada, Spain
Subject: NCS 30m
Keywords: data processing, data format, raw data, calibration, quick look

Description - about this document:
This projects will implement online data processing under Unix. It will be one of the main features of the new control system but we foresee to install part of it already also in the current control system. This version of the document concentrates on the installation in the current control system.

Note the list of pending items.

References:
See the list in Appendix B.
References and Related Documents—Short List:

1. use-case calibratePSwitch
2. use-case calibrateFSwitch
3. use-case calibrateWSwitch
4. use-case calibrateOTFMap
5. pipeline concept
6. project rawDataToUnix documentation
7. Project Abba Control

History of this Document:

1. v 1.0 2001-11-15 wb: draft of 1st version
2. v 1.1 2002-06-03 wb: document release 1

Pending Items:

1. check sections on implementation, installation, and user guide
2. check abbaToFits: the names of the pipe directories have changed!
3. add description on what shall be done for each observing mode

We produced this document using PDF-LaTeX with the standard package hyperref and our own prototype package ncs (HU).

Contents

1 Project Plan (*2002-05-29*) 5

2 Introduction (*2001-11-15*) 5

3 Requirements 5

3.1 * dataOnlineProcessing *state: done 2002-05-29 in release 1* . . . 5
3.2 dataTransferIramEA *state: done 2001-11-07 in release 1* . . . 5
3.3 dataTransferOthers *state: done 2001-11-07 in release 1* . . . 5
3.4 * calibrateSpectra *state: done 2002-05-29 in release 1* . . . 6
3.5 * quickLook *state: done 2002-05-29 in release 2* . . . 6
3.6 * processCancelledScans *state: fixed 2002-05-29 in release 2* . . . 6
3.7 feedbackObs *state: fixed 2001-11-07* . . . 6
3.8 * logging *state: done 2002-05-29 in release 1* . . . 6
3.9 plots *state: fixed 2001-11-07 in release 2* . . . 6
3.10 processPointing *state: open 2001-11-08 in release 2* . . . 7
3.11 processFocus *state: open 2001-11-08 in release 2* . . . 7
3.12 processSkidip *state: open 2001-11-08 in release 2* . . . 7
3.13 * calibratePSwitch *state: fixed 2002-04-30 in release 1* . . . 7
3.14 * calibrateFSwitch *state: fixed 2002-04-30 in release 1* . . . 7
3.15 * calibrateWSwitch *state: fixed 2002-04-30 in release 1* . . . 7
3.16 * calibrateRasterMaps *state: open 2002-04-30 in release 1* . . . 7
3.17 * calibrateOTFMaps *state: fixed 2002-04-30 in release 1* . . . 8
3.18 abbaToFits *state: done 2001-11-15 in release 1* . . . 9
3.19 processWhatElse *state: open 2001-11-08* . . . 9

4 Specifications 9

4.1 obsProcedure *state: done 2001-11-07* . . . 9
4.2 * endOfScan *state: done 2002-05-29* . . . 9
4.3 workingDirectory *state: fixed 2001-11-09* . . . 10
4.4 scriptCoding *state: fixed 2001-11-09* . . . 10
4.5 taskControl *state: fixed 2001-11-09* . . . 10
4.6 taskCrashes *state: fixed 2001-11-09* . . . 10

5 Design 10

5.1 * architectureOverview *state: done 2002-05-29* . . . 10
5.2 obsProcedure *state: done 2001-11-07* . . . 11
5.3 dataTransferIramEA *state: done 2001-11-07* . . . 11
5.4 dataTransferOthers *state: done 2001-11-07* . . . 11
5.5 * endOfScan *state: done 2002-05-29* . . . 11
5.6 processCancelledScans *state: open 2001-11-08* . . . 12
5.7 * processingPipeline *state: done 2002-05-29* . . . 12
5.8 * calibration *state: open 2002-05-29* . . . 13

5.9 feedbackObs *state: open 2001-11-07* . . . 13
5.10 * monitoring *state: done 2002-05-29* . . . 13
5.11 * processCalibration *state: done 2002-05-29* . . . 13
5.12 * processOTF *state: done 2002-05-29* . . . 14
5.13 abbaToFits *state: done 2001-11-15* . . . 14

6 Implementation **15**
6.1 obsProcedure *state: done 2001-11-07* . . . 15
6.2 dataTransferOthers *state: done 2001-11-07* . . . 15
6.3 processCalibration *state: open 2001-11-07* . . . 15
6.4 dataProcessingAbbaToFits *state: open 2001-11-07* . . . 15
6.5 processOTF *state: open 2001-11-15* . . . 15
6.6 abbaToFits *state: done 2001-11-15* . . . 17

7 Installation **17**
7.1 obsProcedure *state: done 2001-11-07* . . . 17
7.2 processCalibration *state: open 2001-11-07* . . . 18

8 Logbook **18**
8.1 meet2001-11-12 *state: done 2001-11-13* . . . 18

9 User Guide (*2002-06-06*) **18**
9.1 Operation () . . . 18
9.1.1 Start of odpCalibration process (*2002-05-29*) . . . 18
9.1.2 Restart of odpCalibration process (*2002-05-29*) . . . 18
9.2 Observer () . . . 19
9.2.1 File with calibrated data (*2002-06-06*) . . . 19
9.2.2 Files with quick look data (*2002-05-29*) . . . 19
9.2.3 Monitor online data processing (*2002-05-29*) . . . 19
9.2.4 Plotting calibrated data (*2002-05-29*) . . . 19

A List of Requirements/Specifications and Descendants **20**

B References **22**

C Use Cases **22**
C.1 Use Case Group: Calibrate . . . 22
C.1.1 Use Case: calibratePSwitch . . . 23
C.1.2 Use Case: calibrateFSwitch . . . 25
C.1.3 Use Case: calibrateWSwitch . . . 27
C.1.4 Use Case: calibrateOTFMap . . . 29

1 Project Plan (*2002-05-29*)

1. 2001-11 version 1.0: Project documentation, basis of installation in the current control system, fix requirements of release 1, first ideas about design
2. 2001-12: design architecture: DONE
3. 2001-12: design of the script to calibrate OTF Maps: DONE
4. 2002-05-29: release 1.0 installed and running: DONE
5. 2002-07: fix requirements for release 2.
6. 2002-10: install release 2.

2 Introduction (*2001-11-15*)

In the current MRT control system, antenna control is done on a slow Vax/VMS system (iramea). For some backends, the data is recorded on iramea and automatically processed at the end of subscans and scans (pointing fits, focus, skydips, calibrations, ...) by **process:RED** . However, due to the limited processing power of the antenna control computer it is desirable to do all the automatic data processing on a fast Unix workstation. In the future, we foresee that all backends will send their data directly to the MRT file server.

The new control system of the 30m RT foresees the implementation of a subsystem data processing. Any development of the current control system shall also take the future system architecture into consideration and can serve as a prototype for the design of the new control system.

3 Requirements

3.1 * dataOnlineProcessing *state: done 2002-05-29 in release 1*

The data recorded by the 30mRT control system shall be processed under Unix directly after the end of subscans and scans. The processing done depends on the observing procedure of the current scan and the receivers and backends used.

Note: *In release 1 the following observing modes will be calibrated: OTFMap, PositionSwitch, FrequencySwitch, WobbleSwitch, RasterMaps. Calibration scans will be analysed and the results (RxTemps,...) will be send to the logger.*

3.2 dataTransferIramEA *state: done 2001-11-07 in release 1*

All header and raw-data files produced by the antenna control software shall be transferred automatically to the MRT file server (curr. mrt-ux1) at the end of each subscan.

3.3 dataTransferOthers *state: done 2001-11-07 in release 1*

Some backends send data directly to the MRT file server: 4MHz, ABBA. This data has to be merged with the data files from the antenna control system.

3.4 * calibrateSpectra *state: done 2002-05-29 in release 1*

Data related to the heterodyne receivers shall be calibrated online under Unix. The data processing software shall write the calibrated data (CLASS format) in a file in a directory of the project.

3.5 * quickLook *state: done 2002-05-29 in release 2*

For most observing modes it is required that results (normally in form of plots) are displayed at the end of subscans. (This is also called "quick look".)

Note: *In release 1, we have impemented a simple quick look software. All calibrated spectra are displayed one after the other, but the display software does not allow (yet) to scroll through the plots. An implroved version of the plot software is foreseen for release 2.*

Note: *At the end of a scan, the data processing is different from the "quick look" processing.*

3.6 * processCancelledScans *state: fixed 2002-05-29 in release 2*

Data from cancelled scans shall be processed as well as far as reasonable, e.g. a pointing cancelled after subscan 2 could provide already the azimuth pointing error.

Note: *In release 1, we do also some processing of cancelled scans, but we have to check exactly what is done.*

3.7 feedbackObs *state: fixed 2001-11-07*

The data processing software has to send results (pointing offsets) back to the OBS program on iramea.

Note: *This feature will not be availabe in the 1st version.*

3.8 * logging *state: done 2002-05-29 in release 1*

The data processing software shall write results into general log files (or into data bases in the future): calibration.log, pointing.log, (?? what else ?? please let me know). The observer shall have a window where this information is displayed.

Note: *In the NCS, this logging information will be send to the general monitoring process. It shall be checked if this concept can also be used in the current system. We also have to know, which software is accessing the current log files.*

3.9 plots *state: fixed 2001-11-07 in release 2*

Result from the online processing can be displayed by plots, e.g. pointing, focus, skydip, spectra (on-off). These plots shall be displayed to the observer (also remote). The plots shall be written into files to be displayed later. Information about the generated plots shall be kept also to allow for fast access (browsing) of the plots of a project. The plots shall be kept in a project directory.

3.10 processPointing *state: open 2001-11-08 in release 2*

Note: *For each observing procedure we should write down the requirements. This requirement is just an incomplete example.*

The processing of pointing scans shall:

1. at the end of each subscan display the recorded data, including a gauss fit
2. at the end of a scan determine the pointing corrections for azimuth and elevation
3. if the backends have been calibrated before, the data shall be plotted as temperatures
4. process all receivers

Note: *RM also suggested to check if the data of several pointing scans could be added together. This feature could be implemented in a later version.*

3.11 processFocus *state: open 2001-11-08 in release 2*

.. to be done ..

3.12 processSkidip *state: open 2001-11-08 in release 2*

.. to be done ..

3.13 * calibratePSwitch *state: fixed 2002-04-30 in release 1*

Calibrate symetric and asymetric position switch scans using the last calibration scan. See details in

use-case calibratePSwitch

3.14 * calibrateFSwitch *state: fixed 2002-04-30 in release 1*

Calibrate frequency switch scans using the last calibration scan. See details in

use-case calibrateFSwitch

3.15 * calibrateWSwitch *state: fixed 2002-04-30 in release 1*

Calibrate wobble switch scans using the last calibration scan. See details in

use-case calibrateWSwitch

3.16 * calibrateRasterMaps *state: open 2002-04-30 in release 1*

Calibrate raster map scans using the last calibration scan. Details are still to be worked out.

3.17 * calibrateOTFMaps *state: fixed 2002-04-30 in release 1*

Calibrate OTFMap scans using the last calibration scan. In release 1 we will request that OTF maps start with a reference subscan! See details in

use-case calibrateOTFMap

Note: *from e-mail HU, 2001-11-15:*

```
Before a spectral-line on-the-fly (OTF) scan a valid CAL COLD
must have been taken.

This CAL COLD scan will be used to calibrate the OTF scan.

During a standard OTF scan:

1    the first subscan is taken on an off-source reference position (REF).
2    one or several OTF subscans are taken.
3    normally another subscan is taken at a reference position.

     Optionally: continue with step 2.

     Normally the last subscan is a REF;
     but in general the sequence can also end with an OTF subscan.

The processing of spectral-line on-the-fly observations shall:

     after the first subscan (REF): do nothing

     after an OTF subscan that is not the last subscan: do nothing

     after the n-th REF subscan (n > 1):
     process all OTF subscans taken between the (n-1)-th and n-th
     REF subscan, using both the (n-1)-th and the n-th REF subscan.

     at the end of the OTF scan, if the last subscan was not a REF:
     process all OTF subscans taken since the last REF subscan,
     using (only) the last REF subscan.

     after any group of OTF subscans has been procesed:
     generate plots (TBD)


Notes:
------
```

```
Only "standard" OTF scans in the sense explained above shall be
processed correctly by ODP.

ODP requires that REF and OTF data are taken in the same scan.


Desiderata for future iterations:
---------------------------------

Support frequency-switched OTF.

Could consider using CAL COLD scans taken after the OTF scan and
processing of non-standard OTF scans.
```

Note: *In release 1, we ALSO produce intermediate results of OTF maps: After each subscan on the source, we calibrate just this subscan and its result is shown in the "quick look". The spectra are not included in the standard class file however. See details in design.*

3.18 abbaToFits *state: done 2001-11-15 in release 1*

Data of bolometer observations with the Abba backend shall be transformed automatically to the FITS format.

3.19 processWhatElse *state: open 2001-11-08*

.. to be done ..

4 Specifications

4.1 obsProcedure *state: done 2001-11-07*

The remote data processing software has to know the observing procedure of a scan. This information has to be recorded in such a way that data processing can also be done later (offline).

4.2 * endOfScan *state: done 2002-05-29*

The data processing software has to know if a scan has finished or not in order to do "quick look" processing or endOfScan processing. E.g. for pointing "quick look" just is means plotting the recorded data whereas at the end of a pointing scan the pointing corrections are calculated.

4.3 workingDirectory *state: fixed 2001-11-09*

A scheme shall be setup to define a working directory for each post processing task (e.g., otfcal, abbaToFits).

4.4 scriptCoding *state: fixed 2001-11-09*

Coding rules for the post processing scripts shall be defined. I assume that these scripts will be written in SIC.

4.5 taskControl *state: fixed 2001-11-09*

The operator shall be able to restart the postprocessing tasks.

4.6 taskCrashes *state: fixed 2001-11-09*

If a postprocessing task (e.g. OTFCAL) crashes, the operator and observer shall be informed and the task shall be restarted automatically.

5 Design

5.1 * architectureOverview *state: done 2002-05-29*

We have the following tasks:

1. data-producer: the current telescope control software produces so-called raw data and header files: header files with specification of the observation and one data file for each backend (see also subsections dataTransfer...). All files are finally stored on **host:mrt-lx1** .
2. post-processing tasks (data-consumer): After raw header and data files are produced, other task will process these files. Possible tasks are:

   (a) calibration of heterodyne data (odpCal)
   (b) fits converters (odpAbbaToFits)
   (c) data analysis of pointing, focus, skidip (odpPointing, ...)
   (d) data plots (odpPlot)

The "communication" between producer and consumer is based on the pipeline concept . Consumer task inform the producer that they want to be informed about new data. It is up to the consumer to decide what to do with the data. Detail are given below in subsection "dataProcessingPipeline".

5.2 obsProcedure *state: done 2001-11-07*

The OBS program shall write the observing procedure of a scan into the scan header using a new OBSINP command OBSP. The procedure can be up to 8 characters long.

Note: *I has to be checked if more information about the observing procedure has to be provided by OBS.*

5.3 dataTransferIramEA *state: done 2001-11-07*

In the current system, for most backends complete raw data files are generated on **host:iramea** . For these backends, all raw data generated is transferred automatically to **host:mrt-ux1** . See also

project rawDataToUnix documentation .

5.4 dataTransferOthers *state: done 2001-11-07*

1. 4MHz backend:
   (a) the backend data is transferred directly from the backend processor **host:vbe4m** to the file server **host:mrt-ux1** .
   (b) The 4MHz backend processor also send UTC for all data dumps to the backend process **process:beorga** on **host:iramea** .
   (c) The antenna control software reserves space in the raw data file and adds DAPs for the specific UTC time given.
   (d) When this raw data file is transferred at the end of a subscan to the MRT file server, the actual data is written into the raw data file
2. ABBA:
   (a) Raw data files are only written for ONE channel in wobbler and skydip mode.
   (b) In the fast-scanning mode, data from the continuum backend is recorded every 250ms in order to have DAPs.
   (c) At the end of a subscan:
      i. the raw data files are transferred
      ii. data for all channels is retrieved from ABBA
      iii. FITS files are generated from rowdata and ABBA files

5.5 * endOfScan *state: done 2002-05-29*

At the end of a scan, the telescope control software (current control system) increases (mod 10000, no scan 0000 ?) the scan number. A utility checks the scan number and generates a message (id=tcs:scanDone). The messages indicates the finished scan and the last subscan number. The message is send to a server program **process:tcsLogServer** . The server program submits a "job" to the pipeline processes and then forwards the message to the message logger.

Note: *The next section is not used for design, it is just kept for reference:*

The antenna control software does not record the total number of subscans in the header. The total number is defined by using 5 parameters (SRPs) plus an option for each of these parameters. We are trying to calculate the total number of subscan from these parameters. An alternative could be to record the total number of subscans by as (as part of the OBSP parameter).

5.6 processCancelledScans *state: open 2001-11-08*

(According to JB Schraml) the preheader of the raw data files have an indication if the subscan was cancelled. The header files are normaly written before the end of a subscan (when the first backend is ready to be written) and therefore cannot contain this type of information.

Note: *As a general rule for the NCS it should be foreseen that all subsystems can write information before, during, and after a subscan (or whatever unit is choosen).*

5.7 * processingPipeline *state: done 2002-05-29*

1. The data-producer (see also subsection on "architectureOverview") will put raw data of a new subscan into a directory **file:"/mrt-lx1/mrt/data/"project"/r"date"** .
2. After this, it will put a link to the header file of the new subscan in all subdirectories that of **file:/mrt-lx1/mrt/data/Pipe/raw/pODP** .
3. Each data processing task has its own subdirectory (e.g. /mrt-lx1/mrt/data/Pipe/raw/cal) and monitors this directory for new links.
4. If a task detects a new link, it does what it has to do and then removes the link.

Please note:

1. data processing tasks can also "forward" links to other tasks, e.g. after a calibration the link can be forwarded to a plot task.
2. the raw data producer can code (e.g. in the file name) what class of observation we have: heterodyne, hera, abba, ... This can help the consumer to decide if the link has to be processed or can be ignored.
3. The current pipeline concept only foresees consumer that monitor themself their directory. In the future, we also foresee to have consumers that can be woken up or that are started each time a jobs is available.

A tool shall display the state of the pipeline: which jobs are pending for how many seconds. (This shall be a requirement of the general pipeline software.) The 1st version of the pipeline will have this structure:

```
raw
   pODP
   abbaToFits
```

This means: raw is the generator of tasks in directories calibration, abbaToFits. Files and links in these subdirectories are processed by processes odpCal, odpAbbaToFits. The odp.. processes can produce new tasks in their subdirectories (or other directories).

5.8 * calibration *state: open 2002-05-29*

All data of heterodyne receivers will be processed by the calibration task. The calibration tasks will prepare script:

1. define SIC variables
   (a) the observing procedure (odpProcedure)
   (b) the project(odpProject)
   (c) scan, subscan number(odpScan, odpSubscan)
   (d) end of scan,0: no, 1:yes (odpEndOfScan)
   (e) sky (0) or reference (¿0) (odpSkyReference)
   (f) calibration scan (1) or no (0) (odpCalibration)
   (g) scan number of last calibration or -1 (odpCalibrationLast)
   (h) backend number (1,2,3,4,or 7) (odpBackend)
   (i) the directory where the raw data is (odpDirectory)
   (j) the filename of the backend file (odpFileName)
2. call a script with the name of the observing procedure
3. wait for the end of the script WITH a TIMEOUT

Note: *In release 1, the scripts produced are not as described above. For release 2, we shall either implement the design as described here or document here the actual design.*

Note: *Please check if more parameters are needed.*

5.9 feedbackObs *state: open 2001-11-07*

Note: *We do not plan to install this feature in the first version.*

OBS will have a global section and a second process (obsServer) shall map to this memory area and run a server software that allows to set/get values of this memory area. OBS uses values from this array to forward the results of the data processing to the antenna control program (internally via OBSINP commands).

5.10 * monitoring *state: done 2002-05-29*

The data processing software writes a log record that is monitored on the observer and operator screen (as part as the general monitoring of the MRT). It displays the scan/subscan number, the procedure and possible results.

Note: *In release 1, the log messages are displayed directly. Release 2 will produce a more readable output.*

5.11 * processCalibration *state: done 2002-05-29*

AS will design a script for calibration.

Note: *Release 1 does not support TESTCAL observations. This will be done in release 2.*

5.12 * processOTF *state: done 2002-05-29*

Note: *from e-mail HU, 2001-11-15:*

```
after the n-th REF subscan (n>1):
---------------------------------

   otfcal is called with:
   1 a pre-script that assigns the approprate values to the
     following required SIC GLOBAL variables (all integers)
          odp#backendCode         ! backend 2 3 4 (5) or 7
          odp#calScan             ! CAL scan number
          odp#flyScan             ! OTF scan number
          odp#refSubscan1         ! REF subscan before OTF subscans
          odp#flySubscanList[1]   ! OTF subscan
              ...                 ! more OTF subscans (otional)
          odp#refSubscan2         ! REF subscan after OTF
   2 a script that processes the OTF data as explained in the
     requirements/specifications and according to the values
     of the variables listed above.
   3 a "post"-script. unused.

at the end of the OTF scan, if the last subscan was not a REF:
--------------------------------------------------------------

   otfcal is called as above, except that:
   1      odp#refSubscan2 = 0     ! must be set to 0 (zero)


generate plots (TBD)
--------------------
```

5.13 abbaToFits *state: done 2001-11-15*

A script abbaToFits will monitor directory **file:/mrt-lx1/mrt/data/Pipe/raw/abbaToFits** for new links to raw data headers. If the observation used the bolometer with Abba it:

1. will transfer the subscan data from the Abba backend
2. and execute the abbaToFits program to generate a Fits file in the directory **file:/mrt-lx1/mrt/data/"project"/fits** .

Note: *A future version of the script will also allow to do the transformation offline.*
Note: *The name of the parameters will be modified in release 2 to use the same name for all calibration scripts if possible (e.g.,use onScan instead of flyScan).*

6 Implementation

6.1 obsProcedure *state: done 2001-11-07*

The OBS software code has been modified and now writes the observing procedure into the header using Obsinp comand OBSP.

6.2 dataTransferOthers *state: done 2001-11-07*

1. 4 MHz backend: the design is implemented. Currently, the 4MHz processor writes data to files on the file-server via NFS. This data is merged with the pseudo raw data files after the transfer to the file server as part of the data transfer software.(source: /mrt-ux1/usr/local/bin/b4merge.icn)
2. Abba: the process abbaToFits.py transfers ABBA data files via FTP and then executes a merger program abbaToFits. (sources in /mrt-lx1/mrt/data/src): TO BE CHECKED
3. Hera: TO BE CHECKED

6.3 processCalibration *state: open 2001-11-07*

the process odpCalibration (source /mrt-lx1/mrt/data/src/CalProc/pipeCal.icn) does:

1. it checks if new links are in /mrt-lx1/mrt/data/Pipe/raw/pODP
2. a script with the name of the observing procedure is executed
3. the link it removed

The OTFCAL scripts are under development (by AS, HU).
Note: *We plan to replace the existing program by a python script.*

6.4 dataProcessingAbbaToFits *state: open 2001-11-07*

To be done. See also Project Abba Control .

6.5 processOTF *state: open 2001-11-15*

Note: *from e-mail HU, 2001-11-15:*

```
"odp#prescriptOTF.otf"
----------------------

Template script showing how to assign values to the required
variables.  In this template the values are taken from the SIC
parameters &amp;1 to &amp;9; in a real case they should be set directly.

Up to 99 OTF subscans can be specified:
```

```
odp#flySubscanList[1] = i
...
odp#flySubscanList[99] = j


"odp#processOTF.otf"
--------------------

Process the OTF subscans as explained above:

For each OTF subscan in odp#flySubscanList (for each element of
odp#flySubscanList that is > 0) the scripts odp#prescript1OTF.otf and
odp#process1OTF.otf are called.


Scripts, Used Scripts, and Helper Scripts:
------------------------------------------

odp#defineOTF.otf

    ensures the correct definition of the SIC GLOBAL variables

odp#examineOTF.otf

    makes a neat list of the SIC GLOBAL variables and their values

odp#plot1OTF.class
    (pre-prototype, unused)

    makes a plot

odp#prescript1OTF.otf

    sets values for processing of 1 OTF subscan

odp#prescriptOTF.otf

    sets values for processing of OTF subscans from one CRO cycle

odp#process1OTF.otf

    processes 1 OTF subscan
    the maximum number of OTF "dumps" is set to 1999 in this script;
```

```
    maybe this should be determined by another variable ...
    NOTE: THIS IS THE (ONLY) SCRIPT THAT DOES "REAL WORK".
          SHOULD BE CHECKED AND TESTED ON MORE DATA.

odp#processOTF.otf

    organizes processing of OTF subscans fromone CRO cycle


Implementation Notes:
---------------------

These scripts assume that a SIC logical "ODP:" is defined pointing to
the directory containing the scripts.

All names of these scripts, as well as the GLOBAL SIC variables they
use, start with the string "odp#".

Some overhead could be avoided by processing the CAL in
odp#processOTF.otf.  In the current version the CAL is processed in
odp#process1OTF.otf for each OTF subscan (and each backend). For
development and test purposes this has the advantage that all the
"real work" is encapsulated in one script.


Note:
-----

The prototypes of these scripts are on mrt-ux1 in:
/users/astro/ungerech/ncs30m/onlineDataProcessing
```

6.6 abbaToFits *state: done 2001-11-15*

TBD

7 Installation

7.1 obsProcedure *state: done 2001-11-07*

A new OBS version has been installed as the default version on **host:iramea**
that records the observng procedure in the raw data header.

7.2 processCalibration *state: open 2001-11-07*

The process pipeCal has to be started during boot of the file server

8 Logbook

8.1 meet2001-11-12 *state: done 2001-11-13*

Meeting by AS, HU, WB: discussed concepts as written in "predraft" (2001-11-09). HU suggested to:

1. separate calibration and further processing (data plot, pointing, ...)
2. start and stop OTFCAL for each subscan
3. check if calibration of data could also be done with calibrations done after an observation

AS mentions that data of focus, skidip, and calibration are not recorded in CLASS files. We agreed on which steps to do next (see also section "plan").

9 User Guide (*2002-06-06*)

9.1 Operation ()

9.1.1 Start of odpCalibration process (*2002-05-29*)

The odpCalibration process is started during boot of
host:mrt-lx1 by script
script:/etc/rc.d/tcs .

9.1.2 Restart of odpCalibration process (*2002-05-29*)

Note: *This note on how to restart odpCalibration will be replaced soon by a script the operator can execute.*

In case of a failure of the odpCalibration process, the operator can restart the process on **host:mrt-lx1** under account root:

1. Check if the process ist still running:

```
ps uxa | grep odpCal
Result (similar to this, numbers can be different)
mrt       3529  0.2  0.4  3384 2108 pts/1    S    09:39   0:06
  /usr/bin/python ./odpCal.py --pipeDir /mrt-lx1/mrt/data/Pipe/raw/pODP/ --name od
```

2. If there is still such a process, stop it:

```
kill -9 3529   # replace 3529 with the current process number
```

3. restart odpCal with command:

```
/etc/rc.d/tcs.d/odpCal
```

9.2 Observer ()

9.2.1 File with calibrated data (*2002-06-06*)

Calibrated spectra are written into
file:/mrt-lx1/mrt/data/(project-directory)/spectraOdp.30m .
During creation of projects we set up a **link:data.30m**
to this file in the default directory of the project. We also set up a
link:data to the directory where project data (rawdata, fits files, ...) are stored. Access to the data directory and file data.30m is readonly !

Note: *Check if these links are defined for your project: if not your project account and execute:*

```
ln -s /mrt-lx1/mrt/data/(project-directory)/spectraOdp.30m data.30m
ln -s /mrt-lx1/mrt/data/(project-directory)/ data
```

9.2.2 Files with quick look data (*2002-05-29*)

Intermediate spectra used for "quick look" are written into
file:/mrt-lx1/data/mrt/(project-directory)/spectraPlot.30m .

9.2.3 Monitor online data processing (*2002-05-29*)

On **host:mrt-lx1** enter:

```
/mrt/tcs/tools/tcsMessageMonXterm odpCal
```

9.2.4 Plotting calibrated data (*2002-05-29*)

The plot utility will display quick look data and the final calibrated data. On **host:mrt-lx1** you can start the plot program by entering:

```
/mrt/tcs/tools/odpPlot
```

Note: *In order to kill the plot process, enter "jobs" to find out the number of the plot job and enter "kill -9 n" with n being the job number.*

A List of Requirements/Specifications and Descendants

1. • req dataOnlineProcessing done 2002-05-29 in release 1
2. • req dataTransferIramEA done 2001-11-07 in release 1
   • des dataTransferIramEA done 2001-11-07
3. • req dataTransferOthers done 2001-11-07 in release 1
   • des dataTransferOthers done 2001-11-07
   • impl dataTransferOthers done 2001-11-07
4. • req calibrateSpectra done 2002-05-29 in release 1
5. • req quickLook done 2002-05-29 in release 2
6. • req processCancelledScans fixed 2002-05-29 in release 2
   • des processCancelledScans open 2001-11-08
7. • req feedbackObs fixed 2001-11-07
   • des feedbackObs open 2001-11-07
8. • req logging done 2002-05-29 in release 1
9. • req plots fixed 2001-11-07 in release 2
10. • req processPointing open 2001-11-08 in release 2
11. • req processFocus open 2001-11-08 in release 2
12. • req processSkidip open 2001-11-08 in release 2
13. • req calibratePSwitch fixed 2002-04-30 in release 1
14. • req calibrateFSwitch fixed 2002-04-30 in release 1
15. • req calibrateWSwitch fixed 2002-04-30 in release 1
16. • req calibrateRasterMaps open 2002-04-30 in release 1
17. • req calibrateOTFMaps fixed 2002-04-30 in release 1
18. • req abbaToFits done 2001-11-15 in release 1
    • des abbaToFits done 2001-11-15
    • impl abbaToFits done 2001-11-15
19. • req processWhatElse open 2001-11-08
20. • spec obsProcedure done 2001-11-07
    • des obsProcedure done 2001-11-07
    • impl obsProcedure done 2001-11-07
    • inst obsProcedure done 2001-11-07
21. • spec endOfScan done 2002-05-29

- des endOfScan done 2002-05-29

22. • spec workingDirectory fixed 2001-11-09
23. • spec scriptCoding fixed 2001-11-09
24. • spec taskControl fixed 2001-11-09
25. • spec taskCrashes fixed 2001-11-09

B References

1. use-case calibratePSwitch
http://www.iram.es/IRAMES/documents/projectDataProcOnline//2002-06-14v1.1.4/ucCalibratePSwitch.html

2. use-case calibrateFSwitch
http://www.iram.es/IRAMES/documents/projectDataProcOnline//2002-06-14v1.1.4/ucCalibrateFSwitch.html

3. use-case calibrateWSwitch
http://www.iram.es/IRAMES/documents/projectDataProcOnline//2002-06-14v1.1.4/ucCalibrateWSwitch.html

4. use-case calibrateOTFMap
http://www.iram.es/IRAMES/documents/projectDataProcOnline//2002-06-14v1.1.4/ucCalibrateOTFMap.html

5. pipeline concept
http://www.iram.es/IRAMES/documents/projectPipeline/

6. project rawDataToUnix documentation
http://www.iram.es/IRAMES/documents/prRawDataToUnix/

7. Project Abba Control
http://www.iram.es/IRAMES/documents/projectAbbaControl

C Use Cases

C.1 Use Case Group: Calibrate

C.1.1 Use Case: calibratePSwitch

Use Case: calibratePSwitch
Subsystem: OnlineDataProcessing
Identifier—MasterURL: //gra-lx1/users/brunswig/ncs/wg_sw_eng/projects/projectDataProcOnline/odpPSwitch
File: ucCalibratePSwitch
Revision: under construction, 0.1
Revision Date: 2002-04-30
Expiration Date:
Authors: W.Brunswig
Priority: Essential
Status: open
Performance: online
Frequency: High.
Extends:

Description - about this use case:

How to process a position switch scan automatically at the end of the scan.

Actors

Preconditions

1. The raw data files of the scan are available.

basic Scenario — Normal flow of events

normal Path —

1. Determine if we do symetric a asymetric position switching:
2. Determine which backends were used.
3. Check last calibration:
4. Determine the project of the data.

5. Generate a script for otfcal. The results of the calibration are written into file "spectra.30m" in the /mrt/data/project/ directory.
6. Call otfcal with the generated script.
7. Generate a message "odpCalDone" with new observation numbers in spectra.30m file.

Post Conditions

1. File spectra.30m in the project data directory has calibrated spectra (sum of all subscans).

Extension Points

Used Use Cases

Subordinate Use Cases

C.1.2 Use Case: calibrateFSwitch

Use Case: calibrateFSwitch
Subsystem: OnlineDataProcessing
Identifier—MasterURL: //gra-lx1/users/brunswig/ncs/wg_sw_eng/projects/projectDataProcOnline/odpPSwitch
File:
Revision: under construction, 0.1
Revision Date: 2002-04-30
Expiration Date:
Authors: W.Brunswig
Priority: Essential
Status: open
Performance: online
Frequency: High.
Extends:

Description - about this use case:

How to process a frequency switch scan automatically at the end of the scan.

Actors

Preconditions

1. The raw data files of the scan are available.

basic Scenario — Normal flow of events for each subscan

normal Path —

1. Determine which backends were used.
2. Check last calibration:
3. Determine the project of the data.
4. Generate a script for otfcal. The results of the calibration are written into file "spectra.30m" in the /mrt/data/project/ directory.

5. Call otfcal with the generated script.
6. Generate a message "odpCalDone" with new observation numbers in spectra.30m file.

Post Conditions

1. File spectra.30m in the project data directory has calibrated spectra (sum of all subscans).

Extension Points

Used Use Cases

Subordinate Use Cases

C.1.3 Use Case: calibrateWSwitch

Use Case: calibrateWSwitch
Subsystem: OnlineDataProcessing
Identifier—MasterURL: //gra-lx1/users/brunswig/ncs/wg_sw_eng/projects/projectDataProcOnline/odpPSwitch
File:
Revision: under construction, 0.1
Revision Date: 2002-04-30
Expiration Date:
Authors: W.Brunswig
Priority: Essential
Status: open
Performance: online
Frequency: High.
Extends:

Description - about this use case:

How to process a frequency switch scan automatically at the end of the scan.

Actors

Preconditions

1. The raw data files of the scan are available.

basic Scenario — Normal flow of events for every second subscan

normal Path —

1. Determine which backends were used.
2. Check last calibration:
3. Determine the project of the data.
4. Generate a script for otfcal. The results of the calibration are written into file "spectra.30m" in the /mrt/data/project/ directory.

5. Call otfcal with the generated script.
6. Generate a message "odpCalDone" with new observation numbers in spectra.30m file.

Post Conditions

1. File spectra.30m in the project data directory has calibrated spectra (sum of all subscans).

Extension Points

Used Use Cases

Subordinate Use Cases

C.1.4 Use Case: calibrateOTFMap

Use Case: calibrateOTFMap
Subsystem: OnlineDataProcessing
Identifier—MasterURL: //gra-lx1/users/brunswig/ncs/wg_sw_eng/projects/projectDataProcOnline/odpPSwitch
File:
Revision: under construction, 0.1
Revision Date: 2002-04-30
Expiration Date:
Authors: W.Brunswig
Priority: Essential
Status: open
Performance: online
Frequency: High.
Extends:

Description - about this use case:

How to process a OTF map scan automatically at the end of the scan.

Actors

Preconditions

1. The raw data files of the scan are available.

basic Scenario — Normal flow of events, first subscan is taken on off-source reference position. For each subscan do:

normal Path —

1. Alternatives:
2. Determine which backends were used.
3. Check last calibration:
4. Determine the project of the data.

5. Generate a script for otfcal. The results of the calibration are written into file "spectra.30m" in the /mrt/data/project/ directory.
6. Call otfcal with the generated script.
7. Generate a message "odpCalDone" with new observation numbers in spectra.30m file.

Post Conditions

1. File spectra.30m in the project data directory has calibrated spectra (sum of all subscans).

Extension Points

Used Use Cases

Subordinate Use Cases