

schedulecreator User Manual

M. Bartolini, P. Libardi, S. Righini

IRA Technical Report
IRA 466-13

Release 0.3
July 10th 2013

1. Overview	2
2. Getting and installing <i>schedulecreator</i>	2
3. Running <i>schedulecreator</i>	3
3.1. Clarifications on the scan-subscan definition	3
3.2. General rules	4
4. Input files: parameters lists	5
4.1. Accepted formats for coordinates	5
4.2. Accepted scan/target frames	5
4.3. Configuration file	5
4.3.1. Backend setup	6
4.3.2. Scan definitions	7
4.4. Target file	8
5. Output files	10
6. Examples	11
6.1. Configuration file	11
6.2. Target file	12
6.3. Notes on T_{sys}	13

1. Overview

schedulecreator is an application created to help the observers using the Italian radiotelescopes (Medicina, Noto and SRT) in single-dish mode. Its aim is to create the schedule files required by the ESCS/Nuraghe system.

What it does:

- this tool reads and interprets a set of parameters provided by the user; it then **produces the schedule files** required by the ESCS/Nuraghe *ScheduleExecutor* in the proper format;
- it produces schedules for **continuum** [*and soon for spectroscopy*] observations for the execution of OTF (On-The-Fly) **cross-scans** or **maps in OTF and raster mode**;

What it does not do:

- **it is not a source visibility checker**: if a target is listed, *schedulecreator* will insert a scan for it in the schedule, not considering whether the target is going to be visible at execution time;
- **it is not an observation planner**: the scans in the schedule files will be executed in the same order as the sorted list of the targets;
- **it is not a format checker**: if users manually create or modify the schedule files for the ESCS system, they can **not** use this tool to check for format correctness or completeness (they could, however, use the *scheduleChecker* tool provided in ESCS/Nuraghe: ask to the support astronomer).

2. Getting and installing *schedulecreator*

The package is temporarily available for download here:

<http://www.ira.inaf.it/~bartolini/schedulecreator/>

Installation instructions are given inside the README file contained in the archive.

3. Running *schedulecreator*

Templates for the input files are provided. To get them,

```
> schedulecreator [-f] -t <destination_directory>
```

creates <destination_directory> and copies *configuration.txt* and *targets.txt* user templates into the folder (see next chapter). If run with **-f** it will override eventually existing files.

To generate a schedule:

```
> schedulecreator [-f] -c <input_configfile_path> <destination_directory>
```

Generates the output schedule files into the destination directory, creating the folder if necessary. <input_configfile_path> is the path to a valid *Configuration file*. Path is relative to the working folder, unless an absolute path is specified. If run with **-f** it overrides eventually existing files.

Example: if the user edited the input files and renamed the *Configuration* one into *MyConfig.txt*, written inside the *Test/* subfolder, and wants to produce the output in the *Output/* folder, then the correct command is

```
> schedulecreator -c Test/MyConfig.txt Output
```

Help instructions can be retrieved by using

```
> schedulecreator --help
```

3.1. Clarifications on the scan-subscan definition

Following the MBFITS definition, we call:

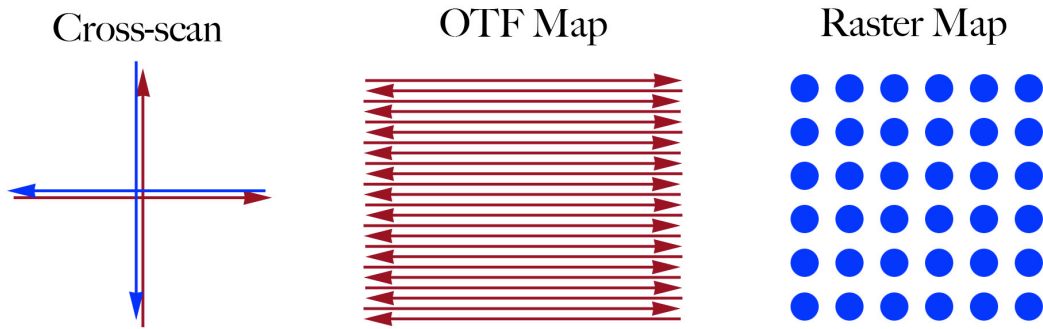
Scan

it is the lowest level object normally used by an observer. *It is a sequence of one or more subscans that share a single goal:* for instance cross-scans and maps involve a pattern of subscans. Whether OTF maps mosaicing observations are considered a single scan or a series of scans is rather a matter of how the user would like to define it. In our implementation each map is considered a scan.

Subscan

it is the minimal amount of data acquisition that can be commanded at the script language level. It is highly desirable that it is a simple enough element. For example, it is the single OTF “line” of a map or of a cross-scan.

The following plot visually represents what cross-scans, OTF maps and raster maps are.



In the case of cross-scan, a subscan is a single arrow (a line across the target), four arrows – i.e. two full crosses – constitute the schema which might be repeated as many times as needed within the scan.

For OTF maps, the subscan is again the single arrow, and the scan coincides with the whole map obtained with lines along one axis only (e.g. along RA or Dec). For raster maps, which are based on discrete acquisitions, each point is a subscan, and the final map constitutes the scan.

3.2. General rules

The creation of the schedule files is based on parameters to be set in the two input files. Parameters in these files are written in the form “**keyword = value**”.

Some general rules apply to the reading procedures:

- keyword and value must be separated by a '=' character;
- the case of keywords is ignored;
- spaces, tabular and newline character at the beginning and at the end of lines are ignored;
- comment lines start with a '#' character (ignoring spaces and/or tabular);
- empty lines are ignored.

4. Input files: parameters lists

The user is required to create and/or edit two files before running *schedulecreator* (see dedicated chapters):

- 1) *Configuration file* – listing the basic information on the project and on the types of data acquisition required; it also lists the names of the other input file;
- 2) *Target file* – providing the list of targets to be observed;

Filenames can be freely chosen - any GNU/Linux valid filename is accepted.

4.1. Accepted formats for coordinates.

Whenever celestial coordinates (Equatorial, Horizontal or Galactic) are specified, the allowed formats are:

- **decimal degrees**, using a ‘d’ suffix, for any coordinate → e.g. 30.00d
- **sexagesimal degrees**, with no suffix, for any coordinate → 30:00:00
- **hh:mm:ss**, with a ‘h’ suffix, for longitudes only → 02:00:00h (*not accepted for offsets*)

4.2. Accepted scan/target frames

Subscans can be executed moving the antenna along the axes of the most common coordinate frames: **Equatorial (J2000.0)**, **Horizontal** and **Galactic**. The same frames are accepted to specify the target coordinates.

The scan frame must coincide with the target frame. There is only an exception: Horizontal scans can be performed also on a celestial source whose J2000.0 Equatorial coordinates are provided.

4.3. Configuration file

This file contains the main parameters for the schedule creation, the reference to the other input file (the *Target file*), and all the scan definitions that are going to be used to produce the output schedule.

Here follows the list of parameters to be included in this main file.

projectID

Label for the project. It will contribute to the output schedule filename and header, and will consequently be written, during the observations, in the headers and names of the data files.

observer

Label for the observer name.

scheduleLabel

Label for the output schedule filenames.

radiotelescope

Code for the telescope: SRT or MED

receiver

Code for the receiver.

For MED the available choices are: C, X, K.

For SRT the available choices are: L, P, C, K, KM. *K* corresponds to the use of the central feed only (data streams from the other feeds will not be recorded), while *KM* means the whole multi-feed is employed.

outputFormat

Format for data recording: FITS or MBFITS

repetitions

Number of repetitions generally valid for all scans. In case the scan is a cross, setting this keyword to 1 corresponds to a double cross (each line is run back and forth), while for maps it means the whole sequence of subscans will be repeated only once.

tsys

Rule for the execution of T_{sys} measurements generally valid for all scans.

Accepted values:

- negative value: do not measure any T_{sys}
- zero: measure the T_{sys} just once for each scan, at the beginning of the scan
- positive integer: measure T_{sys} every “tsys” subscans

This choice applies to all the targets (i.e. all the scans), unless a different value is specified, for the individual targets, in the *Target file*.

The T_{sys} acquisition is performed in a “far” position from the current target. Depending on the subscan parameters, this off-source position is determined in order to minimise the antenna slewing time.

targetsFile

Name of the *Target file*, which must be in the same folder of the *Configuration file*.

After this general information, the file contains two sections, specifically devoted to the backend setup and to the scan definitions.

4.3.1. Backend setup

This section is introduced by the mandatory line

[backend]

after which there must be the following keywords:

integration

Integration time for each sample which gets recorded in the output file.

It is expressed in [ms].

It must be a positive integer number.

samplingInterval

Sampling interval for the backend data acquisition.

It is expressed in [ms]. The fastest sampling (or “dump” time) is 1 ms. The above “integration” can coincide with the sampling interval, or be a longer time interval.

bandwidth

Bandwidth filter selected for the section.

It is expressed in [MHz].

Accepted values for the continuum backend: 300, 730, 1250, 2000.

Notice that these nominal values are usually different from the resulting actual bandwidth, as this depends on the frontend-backend coupling. See the ESCS/Nuraghe User Manuals for details.

4.3.2. Scan definitions

This section is introduced by the mandatory line

[scan]

after which the various scan definitions are listed.

Depending on the scan type, the definitions have a different syntax, as described in the following sections.

Cross scans are defined as

NAME = *CROSS* FRAME LENGTH SPEED

where

NAME	label/ID for this definition, it can be any string
CROSS	mandatory string: ‘CROSS’
FRAME	subscan frame, i.e. the coordinate frame along which the subscans will be performed. It can be ‘EQ’, ‘HOR’ or ‘GAL’
LENGTH	actual subscan span, on sky [degrees]
SPEED	actual subscan speed, on sky [degrees/min, equivalent to arcmin/s]

OTF maps are defined as

NAME = *OTFMAP* FRAME AXIS START LENGTH_X LENGTH_Y SPEED SEPARATION

where

NAME	label/ID for this definition, it can be any string
OTFMAP	mandatory string: ‘OTFMAP’
FRAME	subscan frame, i.e. the coordinate frame along which the subscans will be performed. It can be ‘EQ’, ‘HOR’ or ‘GAL’
AXIS	frame axis along which the subscans will be performed. It can be ‘LON’

or 'LAT' for any frame, or more specifically 'RA', 'DEC', 'AZ', 'EL' according to the scan frame specified above. Assigning the value 'BOTH' to this keyword, two separate scans will be generated: one along longitude and one along latitude

START code for the map starting point, visualizing the map as projected "on sky". Accepted values: 'TL' = Top Left corner, 'TR' = Top Right corner, 'BL' = Bottom Left corner, 'BR' = Bottom Right corner

LENGTH_X actual map size in longitude, on sky [degrees]

LENGTH_Y actual map size in latitude, on sky [degrees]

SPEED actual subscan speed, on sky [degrees/min, equivalent to arcmin/s]

SEPARATION interspacing between adjacent subscans [degrees]

Raster maps are defined as

NAME = RASTERMAP FRAME AXIS START LENGTH_X LENGTH_Y DURATION SEPARATION

where

NAME label/ID for this definition, it can be any string

RASTERMAP mandatory string: 'RASTERMAP'

FRAME subscan frame, i.e. the coordinate frame along which the sequence of acquisitions will be performed. It can be 'EQ', 'HOR' or 'GAL'

AXIS frame axis along which the subscans will be performed. It can be 'LON' or 'LAT' for any frame, or more specifically 'RA', 'DEC', 'AZ', 'EL' according to the scan frame specified above

START code for the map starting point, visualizing the map as projected "on sky". Accepted values: 'TL' = Top Left, 'TR' = Top Right, 'BL' = Bottom Left, 'BR' = Bottom Right

LENGTH_X actual map size in longitude, on sky [degrees]

LENGTH_Y actual map size in latitude, on sky [degrees]

DURATION duration of each subscan [s]

SEPARATION interspacing between adjacent subscans [degrees]

4.4. Target file

This file provides the **sorted list of targets** used to define the sequence of scans within the schedule. Each line contains one target and corresponds to the creation of one scan in the schedule.

The first five values are mandatory, then some optional data can follow.

We stress again that the order of the targets in this file determines the order of the scans in the schedule. If one target is to be observed N times, producing N separate scans, then it must appear N times in the list.

Here is the syntax for each target:

LABEL SCANTYPE TARGET_FRAME LONGITUDE LATITUDE
[tsys, repetitions, offset_lon, offset_lat, offset_frame]

where

LABEL label for target. It will end up in the data files
SCANTYPE name of the scan definition, it must be among the ones listed in the *Configuration file*
TARGET_FRAME frame of the target coordinates specified in the next two positions. It can be 'EQ', 'HOR' or 'GAL'
LONGITUDE target longitude [degrees or - for RA only - hours]
LATITUDE target latitude [degrees]

tsys [optional]

Choice for the T_{sys} measurements, to be applied only to this target.

If present, this value overrides the general setup specified in the *Configuration file*, so that the user can change this preference for each target.

The value must be in the format “tsys=<number>”, <number> must be an integer number.

Accepted values:

- negative value: do not measure any T_{sys}
- zero: measure the T_{sys} just once for each scan, at the beginning of the scan
- positive integer: measure T_{sys} every “tsys” subscans

repetitions [optional]

Choice for the number of repetitions of the whole set of subscans. If present, this value overrides the general setup specified in the *Configuration file*, so that the user can change this preference for each target.

offset_lon [optional]

Offset longitude to be applied to all the subscans for the current target [degrees].

It is considered as “sky offset” (actual offset on sky).

Default value, if not specified, is 0.0d.

offset_lat [optional]

Offset latitude to be applied to all the subscans for the current target [degrees].

It is considered as “sky offset” (actual offset on sky).

Default value, if not specified, is 0.0d.

offset_frame [optional]

Coordinate frame associated to the specified offsets. For OTF acquisitions it must coincide with the scan frame, the one specified in the scan definition appearing in the *Configuration file*, and this is the default if the keyword is absent.

For raster acquisitions, any frame can be specified: 'EQ', 'HOR', 'GAL'.

5. Output files

The tool creates **five files**, fully described in the ESCS/Nuraghe User Manual, to be given in input to the ESCS/Nuraghe system when the observation is carried out.

It is a duty of the users to copy/move these output files to the directory as required by ESCS/Nuraghe.

The output files share a common basename: it is the **scheduleLabel** specified in the *Configuration file*.

Here follows the list of the five output files that constitute the schedule:

- **backend configuration** setup: scheduleLabel.bck
- set of **procedures** to be applied during the schedule execution: scheduleLabel.cfg
- sequence of **scan-level definitions** and **subscan references** for each target: scheduleLabel.lis
- set of **subscan-level definitions** for each target: scheduleLabel.scd
- **additional parameters**, not required by ESCS/Nuraghe, needed to fill in the MBFITS files (produced only if MBFITS is chosen as the output format): scheduleLabel.dat *[NOT PRODUCED IN THIS DRAFT RELEASE!]*

A more accurate description of these files can be found in the MBFITS Technical Report (IRA 461/12) and in the ESCS/Nuraghe User Manuals. However, users ideally should never edit the schedule files, so their structures might remain totally transparent to them. We advise only expert users to attempt direct modifications to the schedules.

6. Examples

6.1. Configuration file

```
#####
```

```
projectID = ProjectName
observer = YourName
scheduleLabel = TestSched
# MED or SRT
radiotelescope = SRT
receiver = C
# fits or mbfits
outputFormat = fits
repetitions = 1
tsys = 0
targetsFile = targets.txt
```

```
[backend]
  integration = 10
  samplingInterval = 10.0
  bandwidth = 300
```

```
[scans]
```

```
#CROSSSCANNAME = CROSS FRAME LENGTH SPEED
```

```
EqCross1_3 = CROSS EQ 1.0d 3.0
```

```
EqCross1_5 = CROSS EQ 1.0d 5.0
```

```
HorCross1_3 = CROSS HOR 1.0d 3.0
```

```
#OTFSCANNAME = OTFMAP FRAME AXIS START LENGTH_X LENGTH_Y SPEED SEPARAT.
```

```
ELMAP1x1TL = OTFMAP HOR EL TL 1.0d 1.0d 5.0 0.04d
```

```
AZMAP1x1TR = OTFMAP HOR AZ TR 1.0d 1.0d 5.0 0.04d
```

```
RAMAP1x1BL = OTFMAP EQ RA BL 1.0d 1.0d 5.0 0.04d
```

```
DECMAP1x1BR = OTFMAP EQ DEC BR 1.0d 1.0d 5.0 0.04d
```

```
GLONMAP2x2TL = OTFMAP GAL LON TL 2.0d 2.0d 10.0 0.1d
```

```
GLATMAP2x2TL = OTFMAP GAL LAT TL 2.0d 2.0d 10.0 0.1d
```

```
EQMap1x1 = OTFMAP EQ BOTH TL 1.0d 1.0d 5.0 0.04d
```

```
#RASTERMAPNAME = RASTERMAP FRAME AXIS START LENGTH_X LENGTH_Y DURATION SEPARAT.
```

```
HorRasterTL = RASTERMAP HOR LON TL 1.0d 1.0d 10.0 0.1d
```

```
HorRasterTR = RASTERMAP HOR LAT TR 1.0d 1.0d 5.0 0.1d
```

```
#####
```

Note: you can list as many scan definitions as you want. Only the ones invoked in the *Target file* will be actually employed.

6.2. Target file

```
#####
# Comment lines to exclude them from the schedule computation
#LABEL SCANTYPE TARGET_FRAME LONGITUDE LATITUDE [tsys, repetitions,
offset_lon, offset_lat, offset_frame]
Alpha EqCross1_3 EQ 12:00:00h +45:00:00
Alpha EqCross1_3 EQ 180:00:00 +45:00:00
Alpha EqCross1_3 EQ 180.0d +45.0d
Alpha EqCross1_3 EQ 180.0d +45.0d repetitions=3
Beta HorCross1_3 HOR 124.5d 38.2d
Gamma EqCross1_5 EQ 13:45:12.3h +34:56:43.0 tsys=2 repetitions=4
Gamma EqCross1_5 EQ 13:45:12.3h +34:56:43.0 offset_lon=-0.5d
Delta RAMAP1x1BL EQ 12:00:00h +45:00:00
Delta DECMAP1x1BR EQ 12:00:00h +45:00:00
Delta EQMap1x1 EQ 12:00:00h +45:00:00
Epsilon GLONMAP2x2TL GAL 275.0d -12.0d tsys=10
Epsilon GLATMAP2x2TL GAL 275.0d -12.0d tsys=10
Zeta HorRasterTL EQ 12:00:00h +45:00:00 offset_lon=0.1d offset_frame=EQ
#####
```

Comments on the above examples:

- The first three target lines are totally equivalent: the only difference is the format of the coordinates. If none of them is commented or deleted, three identical scans (in this case, cross-scans with 1 repetition, i.e. 4 subscans each) will appear in the schedule and will be sequentially executed;
- The fourth line on *Alpha* produces one scan only, containing $4 \times 3 = 12$ subscans (3 double crosses);
- The *Beta* target is a fixed Az,El position on which an Horizontal cross scan is centered;
- The first scan on target *Gamma* sees an intensification of the T_{sys} measurements: instead of taking only one T_{sys} at the beginning of the scan, a new measurement is performed every 2 subscans. Since **repetitions** is set to 4, there will be $4 \times 4 = 16$ subscans, thus 8 T_{sys} measurements;
- In the following scan, **tsys** and **repetitions** are omitted, so the general values given in the *Configuration file* hold. Notice the longitude offset applied to this scan: since the scan frame is EQ, this is a RA offset (expressed in degrees, as for all offsets);
- The first scan on *Delta* is a $1^\circ \times 1^\circ$ map performed scanning in RA, starting from the bottom left corner;
- Then follows a $1^\circ \times 1^\circ$ map performed scanning in DEC, starting from the bottom right corner;
- The last line devoted to target *Delta*, as the 'BOTH' value is given for the AXIS keyword, generates two scans - i.e. two maps - starting from the top left corner;
- Two maps achieved scanning in the Galactic frame then follow;
- The last target is observed in raster map mode. A longitude offset applies to all the points composing the map.

6.3. Notes on T_{sys}

The T_{sys} measurements are inserted before the execution of the selected subscan. If the scan is composed by 10 subscans and the `tsys` keyword is set to 3, the measurements will take place before the subscans number 1, 4, 7 and 10.

The measurement actually takes place as a post-scan operation within a virtual SIDEREAL subscan (i.e. performing a pointing in the same frame selected for the scan), whose duration is zero. This way no data is recorded, but the T_{sys} value is employed inside ESCS/Nuraghe in order to compute the counts-to-K conversion.

If the FITS output is selected, this conversion factor is used to calibrate data, and the resulting stream is saved in a dedicated table (ANTENNA TEMP), while raw counts data is written in the previous table. See the ESCS/Nuraghe User Manual for details.

WARNING: the conversion factor remains valid until a new T_{sys} is taken.

Avoiding to update the T_{sys} for too long, or after a frequency/receiver change, will lead to the use of an obsolete or totally wrong conversion factor.