

# Notes on Urs Graf's task command and synchronization protocol for the KOSMA telescope

A. Harris  
7 August 2002

## 1 General description

Different tasks on the KOSMA telescope communicate with each other by recognizing updated files that contain parameters in ASCII format. Communication files generally have representative names: rx2aos would be the receiver task communicating with the AOS backend task, for instance. Each communication channel thus has a corresponding file that resides in a directory established for all of the communication channels, and the control task for a given piece of hardware must know about and be able to read all of the relevant files. These files provide a snapshot of actions within the system rather than a snapshot of the overall system state.

Values are passed in a *keyword value* formalism. Keywords may appear in the file in any order and are recognized as strings with whitespace to either side that have been defined in the configuration file. A configuration file establishes the keyword names and input and output files for each communication channel.

Fine-scale time synchronization is through the first line in each file, which contains system time information in the operating system's elapsed time format to microsecond precision. The programs are written in C.

The program is built for flexibility and power rather than safety: multiple definitions of a given variable can exist in different communications files, and there is no automatic check that they are all consistent. The configuration files must be carefully written and maintained to insure that the parameters are properly updated by the correct tasks. Files are protected during write by operating system file locking, but there is no other protection. The advantage to this scheme over a scheme with only a single file of parameters is that it is possible to bypass some file writes to control systems that are not otherwise in use for testing or operation in unconventional modes.

## 2 Time synchronization

Absolute time is the basis for synchronization of different tasks on a given or on different computers. (Time synchronization between computers is by the NTP network utility.) Synchronization is a two-step process. First, each specific task has a process which runs continuously, checking the timestamp on the communication channel file. When the process detects a change in file time, it opens the file and reads the first line. Action is based on the elapsed time value, allowing synchronization on a finer timescale than from the file timestamp alone. If the timestamp is later than the timestamp of the last action, then the process initiates a new action; otherwise, there is no action.

### 3 Configuration file

The parameter names (keyword) are established by configuration files that are specific to each communication channel (e.g. rx2aos). This configuration file contains the names of the input file, if any; the output file or files; and a default value for initialization. Some example lines from a typical configuration file, with separation in two sections for two principal output files, are below (spaces between individual entries added for clarity):

```
! Catalog

""      c_sourcename   %s
KOSMA_catalog.in   KOSMA_catalog.in   !Name of the observed source

1      c_coordsys     %12d
KOSMA_catalog.in   KOSMA_catalog.in   !Coordinate System eg. 1=J2000, 9=B1950

0      c_scanduration %12d
KOSMA_catalog.in   KOSMA_catalog.in   !Scan duration [s]

YES    c_trueanglam   %s
KOSMA_catalog.in   KOSMA_catalog.in   !True angle for lambda

! Astra

0      a_inpos        %12d
NULL   KOSMA_astra.status   ! -1 not reachable, 0 not in position, 1 position reached

0      a_elvs         %12.8g
NULL   KOSMA_astra.status   ! Command Elevation Position [degree]
```

The entry structure is

```
default_value  keyword          format_specification
input_file     output_file1,output_file2,...  !Descriptive_comment
```

Each keyword can be read from a single file or generated by the task, in which case the input file name is NULL. A keyword can be written to multiple output files. At the global level it is possible to modify a value that appears in multiple files in some files but not in others.