Uniqname

Bernie Simon

10 October 96

Usage

The task in run by entering the name of the task followed by a list of files you wish to rename. For example,

```
uniqname dark.r3h flat.r4h
```

Wildcard patterns can be used to expand the list of filenames, but care should be taken that the pattern expands to the image header file name and not the data file name. If a file being renamed contains the name of another reference file and that file is not already in the list, its name will be added to the list.

The task changes the filename root to a unique name based on one of three naming conventions. The instrument, as read from the cdbs catalog, determines the naming convention. Reference files whose instrument is synphot add a counted suffix to a fixed root. For example, if the file `hst_ota_004.tab` exists, the next file of that type renamed will be named `hst_ota_005.tab.` Only the synphot throughput tables and some of the non-opus reference files are in the synphot instrument class and use this naming convention. Reference files for the new instruments (stis and nicmos) create a unique name from the current time, plus a suffix indicating the reference file type. For example, a Nicmos background file may be renamed to `gae11178n_bac.fits`. Reference files for the outher instruments create a unique name from the time without any suffix. For example, an fos table might be renamed to `gae1119oy.cy2`.

The algorithm for creating a new filename from the time creates a new filename root with the format `YMDHHMMSSI` where

- Y denotes the year (where 1981 = 1)
- M denotes the month of the year (where January = 1)
- D denotes the day
- HH denotes the hour (00 - 23)
- MM denotes the minutes
- S denotes the seconds divided by 2
- I denotes the instrument code

The algorithm used to generate the name has a time granularity of two aeconds. If asked to rename two separate files, the task will wait until two seconds have elapsed to rename the second, to prevent name clashes. Files which originally have the same root name receive the same new name when renamed.

Reference files that contain their own names or the names of other reference files inside the file have those names changed within the file. The files mentioned in the reference file must be accessible to the program or the reference file will not be renamed.

The task checks the online cdbs directory that the file will eventually be stored in to make sure that it does not duplicate an existing reference file name. Hence the task must be run on a machine from which the cdbs directories are accessible.

Data Files

Uniqname uses the catalog file to determine the reference file type of each reference file it processes, as well as the instrument it is associated with. The catalog file contains the selection rules used to determine the instrument name and reference file type for a reference file. The first match that succeeds determines the reference file type. If no records in the catalog match the reference file, the task prints a warning message. The fields in the catalog file are described in the documentation for the certify task. Uniqname uses three fields in the catalog file: it uses the filetype field to determine what kind of file the reference file is (image, table, or load file). It uses the instrument field to determine how to reanme the file. If the instrument is synphot, it renames the file by adding a counted suffix to the file name. If the instrument is anything else, it sets the last letter in the root to a letter representing the instrument. If the instrument is synphot, it uses the reftype field to determine which keyword contains the root name prefix.

Algorithm

The main function in uniqname, `task`, loops over the files in the argument list, calling uniqname, to rename each file. It also sets up three data structures: a lookup table containing the information from the CDBS catalog file, a list of files to process, and an associative array that lists the old and new root file names. The lookup table is used to identify the reference file type of each file processed. The list of files is created so that new files can be added to the files passed on the command line if any of the renamed files contains the name of another file. The associative array is created so that the task can make sure that two files with the same root receive the same new file name.

The `uniqname` function checks to see if the file can be renamed, generates the new file name, changes keywords in the file to agree with the new name, and then renames the file. The checking is done by function renamable. It looks to see if all files mentioned in the file to be renamed are present on the system and can be written to. If not, the function prints a warning message. The function does this checking so that the task does not leave files in a half changed state, with some files renamed and others not. The name generation is done by function `getuniqname`. It determines which renaming convention to use from the instrument name and either calls `thruname`, `oldinstname`, or `newinstname` to rename them. `Thruname` calls `getsynroot` to generate the new synphot root name, while the other two functions call `makeid` to create the root name from the time. `Makeid` generates the file name from the current time according to the pattern explained previously. `Getsynroot` looks in the header of the table to be renamed for the keyword used to generate the new name. The keyword use depends on the reference file type. `Setnames` changes the keyword within the file containing the file name to the new name. If the file contains a mention of another file, this mention is also renamed and the file is added to the list of files to be renamed.

Test Data

The files in the test directory an be used to test the task. Since we do not wish to change the originals, first copy them to new names, such as test1 and test2. Then generate a load file for the reference files with mkload. Finally, reanme all of the files with the command

```
uniqname *.lod
```

Renaming the load files should also cause all the reference files to be renamed.