Data Handling and Planning Tools for IRAM Plateau de Bure Interferometer operation

R.Neri

May 7, 1999

GRENOBLE: DATA REDUCTION

The following set of commands is available in Grenoble for the handling of Plateau de Bure data:

```
addproj      - creates a new project account
creprojdat   - executes getproj then
                 - archives the entire project's arborescence to a single DAT
                 - on request, removes the project from disk
disaproj     - disables further logins to a project account (no data removal)
getproj      - retrieves projects related data from a series of DATs and CDROMs
init_don     - initializes an optical disk (DON)
movproj      - moves a project account from a disk to another disk
reaproj      - re-enables logins to a project account (new password)
remproj      - deletes a project account and removes its data from disk
savproj      - archives the project's arborescence to a single DAT or CDROMs
```

With the exception of `getproj`, `creprojdat` and `savproj` which are scripts written in `ksh` (the source code is in `/users/system/tools/bin`), all other commands are written in `C` (the source code is in `/users/system/tools/sources`). The executables are all in `/usr/local/bin`. Additional information on all these commands is available on the following pages. An entry in the manual is available for each of these commands.

A typical sequence of commands to execute before the arrival of a project's investigator is the following (authorized users are in brackets):

```
$ addproj  g056  - (bure)
$ getproj  g056  - (bure, berjaud) if arrival is notified well in advance
                   (g056)          if arrival is notified shortly in advance
$ init_don g056  - (bure) from case to case
```

after departure:

```
$ savproj  g056  - (bure, berjaud) only on request / (g056) before departure
$ remproj  g056  - (bure)
```

Further commands (`ksh` scripts) are available for the handling of Plateau de Bure data. The commands, which do not require any priviledge for execution, are in directory `/users/bure/EXEC`:

```
find_project - makes a list of the DATs and the CDROMs containing project data
note         - retrieves the latest revision of a project's note file
release      - retrieves pre-calibrated header files, summaries,...
```

More information on these commands can be found on the next pages.

GRENOBLE: PROGRAMME COMMITTEE MEETINGS (AND STATISTICS)

A TeX report is created for every submitted project with technical comments about the scheduling, Programme Committee comments, indications on the Local Contact and sun avoidance constraints. This is done using `tech.f`. The program requires four input files (the program and the following files are in directory `/users/bure/PROPOSAL/MAR98`):

```
i.comments       - contains different text segments:
                     - project numbers are identified by a leading *
                     - technical comments are the lines of text up to the leading [
                     - PC comments are the lines of text in between [ and [[
                     - indications on the Local Contact are after the [[
sun_i.sou        - contains the sun avoidance periods for every proposed target
submitted.txt    - lists every submitted proposal (project number, requested
                      configurations, project title, authors, lines, target name
                      and target type)
pre-ratings.tex  - contains the PC pre-ratings for every project
```

An official reply to the principal investigators which accompanyies the recommendations of the Programme Committe is created using `reply.f`. The program requires two input files (the following files are in directory `/users/bure/PROPOSAL/MAR98`):

```
submitted.txt    - lists every submitted proposal (project number, requested
                      configurations, project title, authors, lines, target name
                      and target type)
reply.tex        - is the body of the message
```

A TeX report (`submitted.tex`) is created using `submitted.f`. The program creates the summary of all submitted proposals for the Plateau de Bure interferometer. The program requires two input files (the following files are in directory `/users/bure/PROPOSAL/MAR98`):

```
submitted.txt    - lists every submitted proposal (project number, requested
                      configurations, project title, authors, lines, target name
                      and target type)
submitted.title  - a header to be included in the TeX report
```

A TeX table (`pre-ratings.tex`) containing the pre-ratings of the Programme Committee is created using `ratings.f`. The program needs, as input, the pre-rating summaries as sent by E-mail by the members of the Programme Committee in order to retrieve the pre-ratings to every project (a small manipulation on the source code is generally required to adapt the program to the habits of individual PC members). The TeX table is usually verified by a third party.

Plateau de Bure statistics are done using a hotchpotch of different software tools. Some correspondance with Pico Veleta statistical output in view of the Annual Report, the SAC statistics, ... is desirable, but not yet realized.

GRENOBLE: DATA ARCHIVE

A batch exists which allows an automatic handling of the Plateau de Bure data archive in Grenoble. The name of the batch is `rcp_archive`. It resides in directory `/users/bure/ARCHIVE`. Currently, it is executed on `iraux4` three times a day according to file `/users/bure/cronjob`. An execution error is immediately reported by E-mail to the owner of the bure account. To defer the execution of the batch:

- modify the execution times in file `cronjob`
- execute: `crontab -r` — cancels the batch
- execute: `crontab -l /users/bure/cronjob` — reactivates the batch

For manual execution start `ARCHIVE/rcp_archive` from the home directory of the bure account. The batch code is only partially commented (it's self-explanatory, at least a bit). The following Internet pages are updated at each start of `rcp_archive`:

```
file:/users/bure/ARCHIVE/operator.report
http://iram.fr/PDBI/project.html
http://iram.fr/PDBI/ongoing.html
http://iram.fr/local/operator_reports/config.gif
http://iram.fr/local/operator_reports/oper
http://iram.fr/local/astronomer_reports/aodr
```

Currently, consultation of the Plateau de Bure interferometer data archive is also possible via Internet at `http://iram.fr/local/archive/archive`

PLATEAU DE BURE: DATA ARCHIVE

- The batch `sum_day` which is executed three times a day at Plateau de Bure (half an hour before `rcp_archive` in Grenoble) prepares the data transfer to Grenoble. The batch is found in directory `/users/oper/operator/ARCHIVE` on `bure2`. To defer batch execution apply the the same rules as for `rcp_archive`. In case of execution failure, an error message is immediately reported by E-mail to the owner of the bure account on `iraux4`.

- The batch `autoconfig` on directory `/users/oper/operator/CONFIG/` is executed three times a day, 10 minutes before `sum_day`. It provides an update of `config.gif`, a file which contains in graphical form the latest status of the observations scheduled at Plateau de Bure (see `http://iram.fr/local/operator_reports/config.gif`. The plot gives the projects in priority order (the priority is read from the `STATUS` file – first priority projects are in bold type), projects marked as `DONE` are not taken into account.

ADDPROJ

The `addproj` command needs the priviledges of the `bure` account for execution. Can be executed from any machine. With the creation of a project account the following files are created on the account's main directory:

```
.profile            {if [ ! "$DT" ]; then
                        ~bure/EXEC/note_project `whoami`
                        ~bure/EXEC/release `whoami`
                        if [ ! -a "`whoami`.list" ]; then
                            ~bure/EXEC/find_project `whoami`
                        fi
                     fi}
init.graphic        {input}
init.clic           {sic\@ pr:define}
```

Also created is a link `tmp -> /users/project17/tmp/g056`) to the scratch disk. reHere a fac simile of the associated entry in the manual (`man addproj`):

```
NAME
     addproj - to add a project account

SYNOPSIS
     addproj projname user_ident project_disk

DESCRIPTION
     addproj adds a new project account by creating a new entry in the
     /etc/passwd file, updating the /etc/group file and creating the home
     directory which contain the necessary files like HOME/tmp. Usage is
     restricted to privileged users.  It also maintains a listing of all
     projects created, in the file: "/etc/project_list".

     The home directory is located in /users/projectxx where xx is a
     project disk number.

     projname is the project name.

     user_ident is a user identification string.  project_disk is a project
     disk. Only "project1x", where: 1 < x < 9 and
                "project6x", where: 1 < x < 7 are available.

EXAMPLES
     # addproj a044 "test project" project13

     Add the project a044 with the user identification test project and
     create the HOME directory /users/project13/a044.
```

CREPROJDAT

The `creprojdat` command needs the priviledges of the `bure` account for execution. The command can be launched from any machine, but is de facto executed on `iraux6`. Project's data are written to `/users/projecttmp/dummy` with the arborescence used for the `getproj` command. Here a fac simile of the associated entry in the manual (`man creprojdat`):

```
NAME
     cretprojdat - to get project IPB datas from DAT or CDROM and build a
     single DDS DAT.

SYNOPSIS
     creprojdat <project> [<subproj>...]

DESCRIPTION
     creprojdat gets a listing of each RAW data file found on DAT archives
     and CDROM. Then it checks if enought space disk is available to
     retrieve all datas on the scratch disk. If there is enought free disk
     space, it tries to retrieve each files from the appropriate drive,
     asking each time to load a DAT or a CDROM with the appropriate label.
     When all datas have beeen retrieved, it tries to build a complete DDS
     DAT.

     creprojdat can also retrieve subproject datas if <subproj> parameter
     if specified.

EXAMPLES
         $ creprojdat g056

         Project requires diskspace for 1172 Mbytes of raw data
         Available local diskspace is 2493 MBytes

         File g056.list created

         Put the DAT: 192 into the DAT drive of: iraux1
         and press <return> when ready

         Getting: 30-dec-1996-g056.ipb from DAT: 192,9...
         x 30-dec-1996-g056.ipb, 262321664 bytes, 512347 tape blocks
         x 30-dec-1996.log-obs, 280831 bytes, 549 tape blocks
         x 30-dec-1996.log-rdi, 724584 bytes, 1416 tape blocks

         Put the DAT: 196 into the DAT drive of: iraux1
         and press <return> when ready

         Getting: 01-feb-1997-g056.ipb from DAT: 197,2...
         x 01-feb-1997-g056.ipb, 163542528 bytes, 319419 tape blocks
         x 01-feb-1997.log-obs, 585108 bytes, 1143 tape blocks
```

```
x 01-feb-1997.log-rdi, 1037912 bytes, 2028 tape blocks
Getting: 02-feb-1997-g056.ipb from DAT: 197,3...
x 02-feb-1997-g056.ipb, 225078784 bytes, 439607 tape blocks
x 02-feb-1997.log-obs, 357221 bytes, 698 tape blocks
x 02-feb-1997.log-rdi, 814377 bytes, 1591 tape blocks

You must used the CDROM labeled: PDB01005
Load the CDROM disque in the drive, and press return when ready

CDROM mounted on: iraux1
CDROM mounted on: iraux2
CDROM mounted on: iraux4
CDROM mounted on: iraux6

Getting: H302G056.IPB H302G056.OBS H302G056.RDI from CDROM: 01005...

CDROM Dismounted from iraux1
CDROM Dismounted from iraux2
CDROM Dismounted from iraux4
CDROM Dismounted from iraux6
CDROM Dismounted from iraux5

Put an empty TAPE into the DAT drive of: iraux1
and press <return> when ready

Project requires 1172 Mbytes"
Size of the tape you have inserted in the drive: 1630 MBytes"

Do you want to remove files on disk ? y

$
```

DISAPROJ

The `disaproj` command needs the priviledges of the `bure` account for execution. Here a fac simile of the associated manual entry `(man disaproj)`:

```
NAME
     disaproj - to disable a project account

SYNOPSIS
     disaproj projname

DESCRIPTION
     disaproj disables a project account by canceling project access on the
     machine. Usage is restricted to privileged users on iraux2. It didn't
     remove project files and directories.

     projname is the project name to disable.
```

FIND_PROJECT

The `find_project` command needs no priviledges for execution. The command makes a list of the DATs and the CDROMs on which project data reside and writes the list to a file. There is no associated entry in the manual.

```
SYNOPSIS

     find_proj projname

EXAMPLES

     $ find_project g056
     [30-dec-1996]          GC30G056.IPB      261738496 bytes -> CDROM: 0216
     [28-jan-1997]          H128G056.IPB       48858624 bytes -> CDROM: 0229
     [01-feb-1997]          H201G056.IPB       44787712 bytes -> CDROM: 0233
     [02-feb-1997]          H202G056.IPB      224592896 bytes -> CDROM: 0233
     [23-feb-1997]          H223G056.IPB      167541248 bytes -> CDROM: 1003
     [02-mar-1997]          H302G056.IPB      177159680 bytes -> CDROM: 1005
     [03-mar-1997]          H303G056.IPB      189634560 bytes -> CDROM: 1006

     Project requires diskspace for 1058 Mbytes of raw data
     Available local diskspace is 2071 MBytes

     File g056.list created
```

GETPROJ

The `getproj` command needs the priviledges of the `bure` account or of the project's owner account for execution. It is only executable locally (`iraux1` or `iraux6`). The command checks the existence of the project's account (→ `addproj`), then, depending on project type, it creates the following arborescence:

- `DATA/` if the project is not divided into subprojects. For example: `getproj g056` creates the `DATA/` subdirectory.
- `subproj/` where `subproj` is the capital name of a subproject, with as many subdirectories as subprojects. This is the case of projects divided into subprojects at the time of scheduling at Plateau de Bure. For example: `getproj f054 fa54 fb54` will not create the `DATA/` subdirectory, but two directories named `FA54` and `FB54`.

Finally, the command makes on request a global or a partial retrieval of the project's data from DATs and CDROMs (related log-files are also retrieved). The data is stored according to the predefined subdirectory structure (subproject data in the associated subdirectory). Here a fac simile of the associated entry in the manual (`man getproj`):

```
NAME
     getproj - to get project IPB datas from DAT or CDROM.

SYNOPSIS
     getproj <project> [<subproj>...]

DESCRIPTION
     getproj gets a listing of each RAW data file found on DAT archives and
     CDROM. Then it checks if enought space disk is available to retrieve
     all datas. If there is enought free disk space, it tries to retrieve
     each files from the appropriate drive, asking each time to load a DAT
     or a CDROM with the appropriate label.

     getproj can also retrieve subproject datas if <subproj> parameter if
     specified.

EXAMPLES
         $ getproj g056

         Project requires diskspace for 1172 Mbytes of raw data
         Available local diskspace is 2493 MBytes

         File g056.list created

         Put the DAT: 192 into the DAT drive of: iraux1
         and press <return> when ready
```

```
Getting: 30-dec-1996-g056.ipb from DAT: 192,9...
x 30-dec-1996-g056.ipb, 262321664 bytes, 512347 tape blocks
x 30-dec-1996.log-obs, 280831 bytes, 549 tape blocks
x 30-dec-1996.log-rdi, 724584 bytes, 1416 tape blocks

Put the DAT: 196 into the DAT drive of: iraux1
and press <return> when ready

Getting: 01-feb-1997-g056.ipb from DAT: 197,2...
x 01-feb-1997-g056.ipb, 163542528 bytes, 319419 tape blocks
x 01-feb-1997.log-obs, 585108 bytes, 1143 tape blocks
x 01-feb-1997.log-rdi, 1037912 bytes, 2028 tape blocks
Getting: 02-feb-1997-g056.ipb from DAT: 197,3...
x 02-feb-1997-g056.ipb, 225078784 bytes, 439607 tape blocks
x 02-feb-1997.log-obs, 357221 bytes, 698 tape blocks
x 02-feb-1997.log-rdi, 814377 bytes, 1591 tape blocks

You must used the CDROM labeled: PDB01005
Load the CDROM disque in the drive, and press return when ready
CDROM mounted on: iraux1
CDROM mounted on: iraux2
CDROM mounted on: iraux4
CDROM mounted on: iraux5
CDROM mounted on: iraux6

Getting: H302G056.IPB H302G056.OBS H302G056.RDI from CDROM: 01005...

CDROM Dismounted from iraux1
CDROM Dismounted from iraux2
CDROM Dismounted from iraux4
CDROM Dismounted from iraux6
CDROM Dismounted from iraux5
$
```

GETPROJINFO

The `getprojinfo` command needs the priviledges of the `bure` account for execution. For fastest execution, it should be executed from `iraux6`. Here a fac simile of the associated entry in the manual (`man getprojinfo`):

```
NAME
     getprojinfo - to get project disk usage.

SYNOPSIS
     getprojinfo

DESCRIPTION
     getprojinfo scans for each project disk, all files encountered.
     Normally, the file must be a directory associated with an active
     project account (If it is not the case, an error message is display to
     indicate that the file found is not a directory or that the directory
     is not associated with a project account in the /etc/passwd file).

     For each project directory, getprojinfo gets the total disk usage, the
     project identification string found in the /usr/local/lib/project_list
     database, the number of IPB files and the date of the last log in.

EXAMPLES
     $ getprojinfo
     Error: /disk12/TT_DB is not an active project
     Error: /disk62/quota.lis is not a project directory
     Error: /disk62/bibi is not an active project

      disk  proj size(Mb) ipb  last login Owner
     ~~~~~~ ~~~~ ~~~~~~~~ ~~~~ ~~~~~~~~~~ ~~~~~
     disk12 f061      0      0  97/04/09  T.Forveille
     disk12 f062      0      0  97/03/23  T.Forveille
        .     .       .      .       .         .
        .     .       .      .       .         .

     size(Mb)  disk  proj ipb  last login Owner
     ~~~~~~~~ ~~~~~~ ~~~~ ~~~~ ~~~~~~~~~~ ~~~~~
       1276   disk66 g070    8  97/06/16  J.Wink
       1264   disk18 g048   19  97/05/18  R.Neri
       1148   disk13 h034    1  97/06/31  S.Guilloteau
        .     .       .      .       .         .
        .     .       .      .       .         .

     Owner                  proj  disk  size(Mb) ipb  last login
     ~~~~~~~~~~~~~~~~~~~~~~ ~~~~ ~~~~~~ ~~~~~~~~ ~~~~ ~~~~~~~~~~
     A.Baudry               c018 disk64      2      0  96/08/18
     A.Dutrey               c026 disk63     84      0  96/10/25
```

```
A.Dutrey               c028 disk63    160      0  97/02/28
    .      .        .       .        .         .
    .      .        .       .        .         .

last login proj  disk  size(Mb) ipb  Owner
~~~~~~~~~~ ~~~~ ~~~~~~ ~~~~~~~~ ~~~~ ~~~~~
 92/02/04  b028 disk63      0      0
 93/00/25  b027 disk63      0      0
 93/00/28  b046 disk63      0      0
    .      .        .       .        .         .
    .      .        .       .        .         .
```

INIT_DON

The `init_don` command needs the priviledges of the `bure` account for execution. Here a fac simile of the associated entry in the manual (`man init_don`):

```
NAME
     init_don - to initialize a user or a project DON

SYNOPSIS
     init_don n project mode

DESCRIPTION
     init_don initializes a project or a user DON. The initialization for
     project DON consists on creating two files:

          /project
                  to identify the DON type.

          /xyyy   to identify the DON project owner ( x is a letter and yyy
                  a number).

     project is the project name or an identification string in case of
     user DON

     mode determine which is the DON type. It must be ipb, map or user

EXAMPLES
     $ init_don 1 a044 map

     Initialize the DON 1 for project a044.

     $ init_don 4 dumontro user

     Initialize the user DON dumontro
```

MOVPROJ

The `movproj` command needs the priviledges of the `bure` account for execution. Here a fac simile of the associated entry in the manual (`man movproj`):

```
NAME
     movproj - to move project account from a disk location to another.

SYNOPSIS
     movproj projname directory

DESCRIPTION
     movproj Moves a project account to the new location specified by
     directory parameter. Usage is restricted to privileged users on
     iraux2.

     projname is the project name to reactivate.

     directory is the new disk location for the project.
```

NOTE (NOTE_PROJECT)

The `note` command needs no priviledges for execution and can be executed from any machine. The command writes locally a copy of the latest revision of a project's note file. There is no entry in the manual.

```
NAME
     note - to retrieve a note file

SYNOPSIS
     note projname

EXAMPLE

     $ note g056
     File g056.note created (latest revision Nov 7 1997)
```

REAPROJ

The `reaproj` command needs the priviledges of the `bure` account for execution. Here a fac simile of the associated entry in the manual (`man reaproj`):

```
NAME
     reaproj - to reactivate project account

SYNOPSIS
     reaproj projname

DESCRIPTION
     reaproj reactivates a project account previously disabled with the
     disaproj command. Usage is restricted to privileged users on iraux2.

     projname is the project name to reactivate.
```

RELEASE

The `release` command needs no priviledges and can be executed from every machine. The command looks on the local directory for existing files associated to a project and which are of the following type:

```
dd-mmm-yyyy-proj.ps.gz         - pre-calibration postscript protocol
dd-mmm-yyyy-proj.hpb.gz        - pre-calibrated header files
dd-mmm-yyyy-proj.summary.gz    - pre-calibration summary files
setup-proj.obs-dd-mmm-yyyy.gz  - update of the observing procedure
show-dd-mmm-yyyy-proj.ps.gz    - postscript trace of a run (tsys, cable,...)
```

where `dd` is the day, `mmm` the month, `yyyy` the year and `proj` the name of the project. Existing files will not be overwritten, not existing one will be made locally available. There is no associated entry to the `release` command in the on-line manual.

```
NAME
     release - to download pre-calibrated data and others

SYNOPSIS
     release projname

EXAMPLES

     $ release i004
     ... copying setup-i004.obs-13-apr-1998.gz
     ... copying 13-apr-1998-i004.hpb.gz
     ... copying 13-apr-1998-i004.ps.gz
     ... copying 13-apr-1998-i004.summary.gz
```

REMPROJ

The `remproj` command needs the priviledges of the `bure` account for execution. Here a fac simile of the associated entry in the manual (`man remproj`):

```
NAME
     remproj - to remove a project account

SYNOPSIS
     remproj projname

DESCRIPTION
     remproj removes a project account by deleting the entry in the
     /etc/passwd file, updating the /etc/group file and removing the home
     directory. Usage is restricted to privileged users.

     projname is the project name to remove.
```

SAVPROJ

The `savproj` command needs the priviledges of the `bure` account for execution. The command is not executed if the project data exceeds the capacity of the DAT. Here a fac simile of the associated manual entry (`man savproj`):

```
NAME:
      savproj - to save essential project directories on a DDS DAT or
      CDROM(s).

SYNOPSYS
savproj checks if you are logged on a project account. Then it gives
      you the choice to save your calibrated and/or raw data on a DDS DAT or
      on CDROM(s). If you choose to save your data on a DDS DAT, the DAT
      size is checked. If it is too small, savproj asks the user to put
      another bigger DAT tape. If you choose to save your data on CDROM,
      savproj saves them on as many CDROMs as needed. Each CDROM contains an
      identification file (of the form: PART<x>-<y> where <x> is the CDROM
      number and <y> the total number of CDROMs) which contains the full
      listing of all the files saved on the <y> CDROMs. Calibrated and raw
      data are put on different CDROMs.

      Raw data are generally on tmp/DATA, but if savproj finds raw data into
      your home directory ($HOME/DATA), it asks you if you want to save
      them. In this case, raw data can be retrieve (on DAT or CD) in the
      directories: DATAHOME and DATA.

WARNING
savproj moves: headers, reports and maps directories in a temporary
      place and replace them by simple empty files to avoid rebuilding if
      you create a new window or you make a new login during the operation.
      For this reason and because raw data must not be modified, avoid to
      work or to create new sessions when you are using savproj.

SEE ALSO
      addproj(1), disaproj(1), movproj(1), reaproj(1), remproj(1),
      getprojinfo(1) getproj(1), creprojdat(1)
```