LINEDB

Sébastien BARDEAU
(IRAM/Grenoble)
on behalf of the LINEDB developers
J.Pety, S.Maret, M.Lonjaret

9th IRAM Millimiter Interferometry School
Oct. 10th-14th 2016, St Martin d'Hères

LINEDB: motivation

Provide line catalogs access to all GILDAS programs

CLASS / WEEDS:

- Online catalog requests
- Local cache
- Plot
- Selection
- Modeling
- ...

SIC / LINEDB:

- Line catalog access
- Online and Offline
- Requirements: Python (Gildas-Python binding) and network for online databases
- CLASS-like browsing (commands and variables)

LINEDB: need help?

```
SIC> help linedb\
              LINEDB\ Command Language Summary
    This language is an interface to on-line and off-line line catalogs.

    USE       Define the atomic and molecular line database(s) used as input
              and/or output
    SELECT    Select lines in the input database(s) according to user defined
              filters
    LIST      List lines from the current line index
    INSERT    Insert lines from the current line index into the output database
    REMOVE    Remove lines from the current line index from the output database

SIC> help use
LINEDB\USE = "python linedb/use.py"
        LINEDB\USE  IN|OUT|BOTH  dbname1  [[dbname2]   [dbname3...]] [/OVER-
    WRITE]

    Define the atomic and molecular line database(s) used  as  input  and/or
    output.
    ...
```

LINEDB: selecting the database

```
SIC> use in cdms
I-USE,  cdms (online) selected

or

SIC> use in jpl
I-USE,  jpl (online) selected

or

SIC> use in mycatalog.db  ! Local sqlite catalog
I-USE,  mycatalog.db (offline) selected

or

SIC> use in cdms jpl mycatalog.db  ! Concurrent access to these 3 inputs
                                   ! => Be careful to duplicate lines
I-USE,  cdms (online) selected
I-USE,  jpl (online) selected
I-USE,  mycatalog.db (offline) selected
```

LINEDB: selecting the lines

Select all lines (beware this downloads ALL the online catalogs!):

```
SIC> select
```

Select by frequency range:

```
SIC> select /frequency 83000 117000
```

Select by species name (beware the names are catalog-dependent!):

```
SIC> select /species C-13-O+
```

More selectors:

```
/origin: when mixing several origins in offline catalogs
/energy: maximum energy of the upper level
/Aij: minimum Einstein coefficient
```

LINEDB: listing the lines

List selected lines:

```
SIC> list
  # Species    Freq[MHz] Err[MHz] Eup[K] Gup  Aij[s-1]  Up level -- Low level  Origin
  1 C-13-O+  111226.538   0.730     5.4   3  5.84e-07     1 2 1 -- 0 1 1        cdms
  2 C-13-O+  112465.938   0.120     5.5   1  3.76e-05     1 1 0 -- 0 1 1        cdms
  3 C-13-O+  112695.175   0.080     5.5   3  3.72e-05     1 1 1 -- 0 1 1        cdms
  4 C-13-O+  112753.480   0.040     5.4   3  3.73e-05     1 2 1 -- 0 1 0        cdms
  5 C-13-O+  112902.557   0.040     5.5   5  3.80e-05     1 2 2 -- 0 1 1        cdms
  6 C-13-O+  114222.051   0.733     5.5   3  6.32e-07     1 1 1 -- 0 1 0        cdms
```

List all individual species:

```
SIC> list /toc
 Species              LinesCount
 C-13-O+              6
```

Lines available in SIC variables:

```
SIC> exa lines%
LINES%          ! Structure GLOBAL
LINES%N          =            6               ! Integer GLOBAL
LINES%SPECIES   is a character*   14 Array     of dimensions  6
LINES%UNCERTAINTY is a real Array     of dimensions  6
LINES%FREQUENCY is a double precision Array     of dimensions  6
LINES%AIJ       is a double precision Array     of dimensions  6
LINES%EUP       is a real Array     of dimensions  6
LINES%GUP       is a real Array     of dimensions  6
LINES%QNUP      is a character*   12 Array     of dimensions  6
LINES%ELOW      is a real Array     of dimensions  6
LINES%GLOW      is a real Array     of dimensions  6
LINES%QNLOW     is a character*   12 Array     of dimensions  6
```

LINEDB: saving lines in a local database (1)

Create a (sqlite) database:

```
SIC> use out mycatalog.db /overwrite
```

Copy the current selection in the output database:

```
SIC> insert
```

Reuse the local database:

```
SIC> use in mycatalog.db
SIC> select
```

LINEDB: saving lines in a local database (2)

Example: make a local copy of the CDMS catalog

```
SIC> use in cdms
SIC> use out mycdms.db /overwrite

SIC> select  ! HUGE REQUEST => rejected by the server

SIC> define real fmin fmax
SIC> let fmin 0                ! From 0
SIC> for /while fmax.lt.1e6    ! Up to 1 THz
SIC:   let fmax fmin+2e4
SIC:   say 'fmin' 'fmax'
SIC:   select /freq 'fmin' 'fmax'
SIC:   insert
SIC:   let fmin fmin+2e4
SIC: next
```

Execution time: ~1.4 hrs
Database size: ~460 MB
+ Offline, no latency, frozen quantities
- Missing updates, e.g. new lines or improved measurements

LINEDB: deleting lines from a local database

Select the lines to be deleted:

```
SIC> select /freq 114222.051
I-SELECT,  1 lines found in the frequency range 114222.041 to 114222.061 MHz
```

Open the output database for writing and remove the selected lines:

```
SIC> use out mycatalog.db
SIC> remove
I-REMOVE, 1 lines deleted
```

Check:

```
SIC> select
I-SELECT,  5 lines found in the frequency range 0 to infinity MHz
```

LINEDB: private lines?

Build an ASCII file following the JPL* data format:

```
SIC> type demo-53001.cat
               53001  C2H3CN
  110184.8053  3.3931 -6.7588 3  651.8432 83  53001140540 635 241  41 536 242
  110184.8057  3.3931 -6.7800 3  651.8432 79  53001140540 635 239  41 536 240
  110184.8319  3.3931 -6.7694 3  651.8431 81  53001140540 635 240  41 536 241
  110187.0300 30.7489 -8.7939 3  844.9262113  53001140555 352 256  56 155 257
  110187.0633 30.7489 -8.8094 3  844.9262109  53001140555 352 254  56 155 255
  110187.1829 30.7489 -8.8017 3  844.9261111  53001140555 352 255  56 155 256
```

You have to provide also the partition function of the species:

```
SIC> type partfunc.cat
species C2H3CN  53001
temperatures 300. 225. 150. 75. 37.5 18.75 9.375
qpart 2. 5.0769 4.8334 4.5016 3.9982 3.5418 4.9
```

Then use it as input database:

```
SIC> use in demo-53001.cat
I-USE,  demo-53001.cat (offline) selected with /(partfunc.json|partfunc.cat) for
partition functions
SIC> select
I-SELECT,  6 lines found in the frequency range 0 to infinity MHz
```

Convert to Sqlite database:

```
SIC> use out demo-53001.db
SIC> insert
```

*See `http://spec.jpl.nasa.gov/ftp/pub/catalog/README`

LINEDB: next improvements

Support VAMDC standard:

CDMS is switching to VAMDC standard.
Current CDMS server will close at some point.
Work in progress, interacting with CDMS group.