Kronos data processing Pipeline and Database
Preliminary version

Henri Chain

April 2, 2015

1 Kronos Data Products

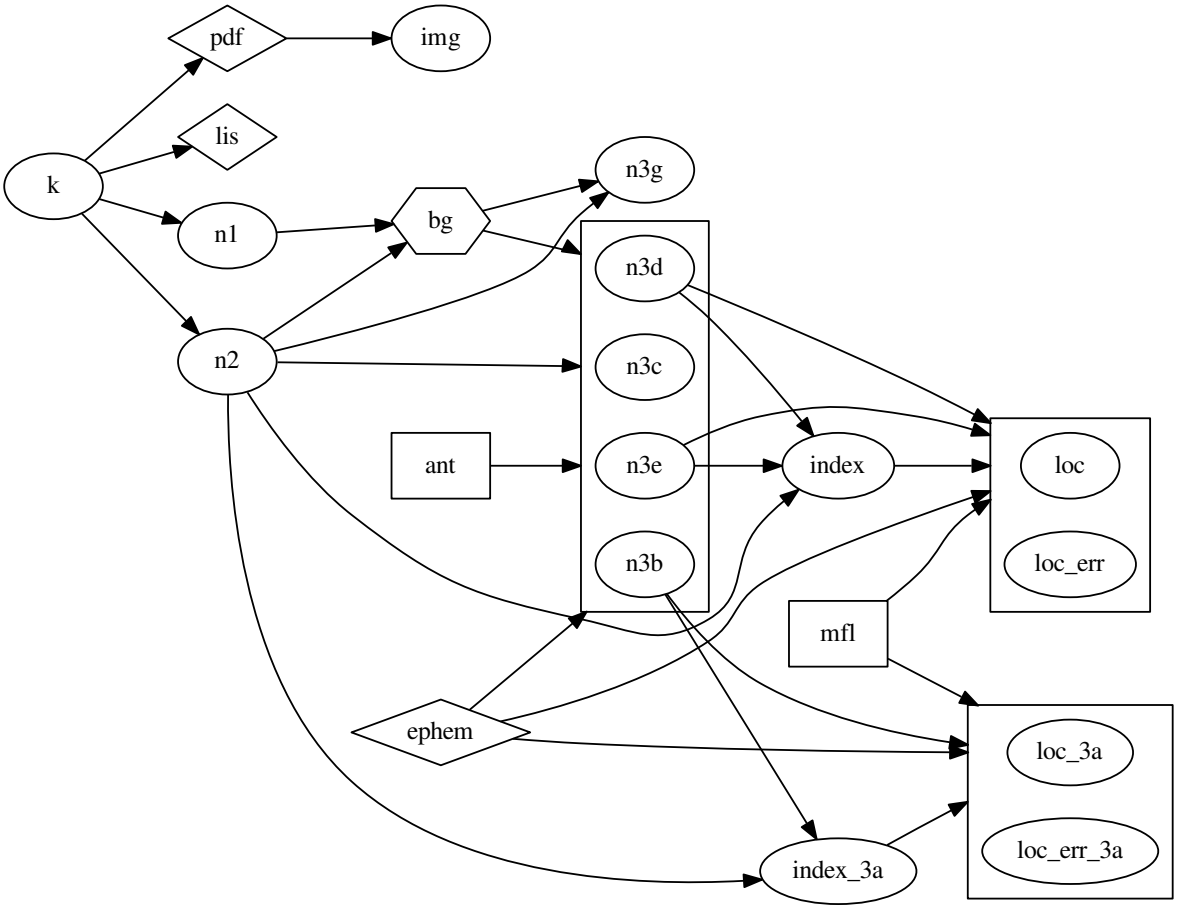

Figure 1: Kronos Data Flow

2 Ingesting Kronos files into a database

2.1 `kronos_sources` table

Name	Type	Comment
`source`	`char(2)`	Two-letter code of the ephemeris source (`sc`, `sq`...). Unique.
`name`	`varchar(128)`	Full Name of source.
`need`	`char(2)`	Optional source dependency. (can be `NULL`)
`att`	`char(4)`	Four-letter attitude extension (`qsse`, `qssq`)
`eph`	`char(4)`	Four-letter position extension (`vsat`, `vear`)
`body`	`varchar(128)`	Body name (`Saturn`, `Earth`...)
`radius`	`real`	Body radius (km)

Table 1: `kronos_sources` schema

source	name	need	att	eph	body	radius
su	Sun		qsun	vsun	Sun	695500
ea	Earth		qgse	vear	Earth	6378.14
ju	Jupiter		qjse	vjup	Jupiter	71492
sc	Saturn (Center) - ecliptic frame		qsse	vsat	Saturn	60268
sq	Saturn (Center) - equatorial frame		qssq	vsat	Saturn	60268
sj	Saturn (Barycenter) - J2000 frame		qj20	vstb	Saturn	60268
ti	Titan - ecliptic frame	sc	qsse	vtit	Titan	2575
tq	Titan - equatorial frame	sq	qssq	vtit	Titan	2575

Table 2: `kronos_sources` typical contents

2.2 `kronos_antennae` table

Name	Type	Comment
`ant`	`char(1)`	One-letter code of the antenna parameter set (`d`, `j`, `r`...). Unique.
`name`	`varchar(128)`	Full Name of antenna set.
`filename`	`varchar(128)`	Name of antenna file in `/ant` directory.

Table 3: `kronos_antennae` schema

ant	name	filename
j	Jupiter-Cal(calJGR2004)	calJGR2004
d	December-04-cal	calDec04
r	Rheometry	rheometry.ant
x	Default Calibration	

Table 4: `kronos_antennae` typical contents

2.3 kronos_mfl table

Name	Type	Comment
model	varchar(32)	Magnetic field model code (SPV, VIP4, SPVR...). Unique.
name	varchar(128)	Description of model.
body	varchar(128)	Body Name.
filename	varchar(128)	Name of stored iso_fc_grid file in /rpws/bin/ directory.

Table 5: kronos_mfl schema

model	name	body	filename
D	7 Gauss (Z96) model	Jupiter	
O6	O_6 Octupole model (Connerney et al. 1993)	Jupiter	
VIP4	VIP4 model (Connerney et al. 1998)	Jupiter	
VIT4	VIT4 model (Connerney et al. 2007)	Jupiter	
Z3	Z_3 model (Connerney et al. 1982)	Saturn	
SPV	Saturn Pioneer Voyager model (David and Smith, 1990)	Saturn	iso_fc_grid_SPV.sav
SPVR	SPV-R model, 5th order truncated	Saturn	

Table 6: kronos_mfl typical contents

2.4 kronos_periods table

Name	Type	Comment
period	varchar(128)	Name of the data directory (2013_091_180, Venus1...). Unique.
start_year	integer	Year of first day in period. 1997 to 2017.
start_day	integer	First day in period. 001 to 366.
end_year	integer	Year of last day in period. 1997 to 2017.
end_day	integer	Last day in period. 001 to 366.
bg	timestamp	Date the background file was last computed[1] (/bg/bg_yyyyddd_yyyyddd)
bg_tmp	timestamp	Timestamp of temporary background file[1] (/bg/bg_yyyyddd_yyyyddd_tmp)

Table 7: kronos_periods schema

2.5 kronos_days table

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366, (year, day) must be unique.
period	varchar(128)	Must be present in kronos_periods
source	char(2)	Source used for plots. Must be present in kronos_sources.
ant	char(1)	Antenna set used for plots. Must be present in kronos_antennae.
raw	timestamp	Date the daily Raw plot was last computed[1] (/pdf/yyyyddd.pdf)
sve	timestamp	Date the daily 2-Antenna S/V plot was last computed[1] (/pdf/yyyyddd-SVe.pdf)
svb	timestamp	Date the daily 3-Antenna S/V plot was last computed[1] (/pdf/yyyyddd-SVb.pdf)
ltb	timestamp	Date the daily 3-Antenna L/T plot was last computed[1] (/pdf/yyyyddd-LTb.pdf)
lis	timestamp	Date the daily .lis file was last computed[1] (/lis/yyyyddd.lis)
skr	timestamp	Date the daily .skr file was last computed[1] (/skr/yyyyddd.skr)

Table 8: kronos_days schema

[1] NULL if unavailable or of size 0

2.6 kronos_ephem table

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366, (year, day) must be present in kronos_days.
source	char(2)	Must be present in kronos_sources.
att	timestamp	Date the hourly file was last downloaded[1] (/ephem/yyyyddd.qxxx)
eph	timestamp	Date the hourly file was last downloaded[1] (/ephem/yyyyddd.vyyy)

Table 9: kronos_ephem schema

(year, day, source) must be unique.

2.7 kronos_2hours table

This table contains timestamps for half-hourly data products (every two hours) as well as average ephemeris data.

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366, (year, day) must be present in kronos_days.
hour	integer	00 to 22, even, (year, day, hour) must be unique.
img_raw1	timestamp	Date the raw plot image was last computed[1] (/img/yyyyddd-1-hh.png).
img_raw2	timestamp	Date the raw bg-subtracted plot image was last computed[1] (/img/yyyyddd-2-hh.png).
img_sve1	timestamp	Date the 2-Antenna S plot image was last computed[1] (/img/yyyyddd-SVe-1-hh.png).
img_sve2	timestamp	Date the 2-Antenna V plot image was last computed[1] (/img/yyyyddd-SVe-2-hh.png).
img_svb1	timestamp	Date the 3-Antenna S plot image was last computed[1] (/img/yyyyddd-SVb-1-hh.png).
img_svb2	timestamp	Date the 3-Antenna V plot image was last computed[1] (/img/yyyyddd-SVb-2-hh.png).
img_ltb1	timestamp	Date the 3-Antenna L plot image was last computed[1] (/img/yyyyddd-LTb-1-hh.png).
img_ltb2	timestamp	Date the 3-Antenna T plot image was last computed[1] (/img/yyyyddd-LTb-2-hh.png).
dist	real	Average distance from ephemeris data (can be NULL).
ra	real	Average local time from ephemeris data (can be NULL).
lat	real	Average latitude from ephemeris data (can be NULL).

Table 10: kronos_2hours schema

2.8 kronos_hours table

This table contains timestamps for hourly data products.

[2]Relative to http://cassini.physics.uiowa.edu/~wsk/cas/daspage/cas_data/

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366, (year, day) must be present in kronos_days.
hour	integer	00 to 23, (year, day, hour) must be unique
k	timestamp	Date the hourly file was last downloaded[1] (/k/Kyyyyddd.hh)
bad	boolean	Whether the K file is unusable (therefore located in /k_bad/Kyyyyddd.hh)
src_dir	varchar(128)	Location of the K file on Iowa server [2]
n1	timestamp	Date the hourly file was last computed[1] in level n1 (/n1/Ryyyyddd.hh)
n2	timestamp	Date the hourly file was last computed[1] in level n2 (/n2/Pyyyyddd.hh)
n3g	timestamp	Date the hourly file was last computed[1] in level n3g (/n3g/Fyyyyddd.hh)
index	timestamp	Date the 2-Antenna index file was last computed[1] (/index/INDEX_yyyyddd.hh)
index_3a	timestamp	Date the 3-Antenna index file was last computed[1] (/index/INDEX_3A_yyyyddd.hh)

Table 11: kronos_hours schema

2.9 kronos_loc table

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366
hour	integer	00 to 23, (year, day, hour) must be present in kronos_hours.
mfl	varchar(32)	Must be present in kronos_mfl.
loc	timestamp	Date the localization file was last computed[1] (/loc/loc_MFL_yyyyddd.hh)
loc_3a	timestamp	Modification date of 3-antenna .loc file[1] (/loc/loc_3A_MFL_yyyyddd.hh)

Table 12: kronos_loc schema

(year, day, hour, mfl) must be unique.

2.10 kronos_loc_err table

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366
hour	integer	00 to 23.
mfl	varchar(32)	(year, day, hour, mfl) must be present in kronos_loc.
n_error	integer	Number of error vectors.
loc_err	timestamp	Modification date[1] of /loc/loc_err_MFL_nNN_yyyyddd.hh
loc_err_3a	timestamp	Modification date[1] of /loc/loc_err_3A_MFL_nNN_yyyyddd.hh

Table 13: kronos_loc schema

(year, day, hour, mfl, n_error) must be unique.

2.11 kronos_n3 table

Name	Type	Comment
year	integer	1997 to 2017
day	integer	001 to 366
hour	integer	00 to 23, (year, day, hour) must be present in kronos_hours.
source	char(2)	(year, day, source) must be present in kronos_ephem.
ant	char(1)	Must be present in kronos_antennae.
n3a	timestamp	Date the hourly file was last computed[1] in level n3a (/n3a/N3a_xyyyyyyddd.hh)
n3b	timestamp	Date the hourly file was last computed[1] in level n3b (/n3b/N3b_xyyyyyyddd.hh)
n3bfull	timestamp	Same with all solutions (/n3b/N3b-full_xyyyyyyddd.hh)
n3c	timestamp	Date the hourly file was last computed[1] in level n3c (/n3c/N3c_xyyyyyyddd.hh)
n3cfull	timestamp	Same with all solutions (/n3b/N3c-full_xyyyyyyddd.hh)
n3d	timestamp	Date the hourly file was last computed[1] in level n3d (/n3d/N3d_xyyyyyyddd.hh)
n3dfull	timestamp	Same with all solutions (/n3b/N3d-full_xyyyyyyddd.hh)
n3e	timestamp	Date the hourly file was last computed[1] in level n3e (/n3e/N3e_xyyyyyyddd.hh)
n3efull	timestamp	Same with all solutions (/n3b/N3e-full_xyyyyyyddd.hh)
n3f	timestamp	Date the hourly file was last computed[1] in level n3f (/n3f/N3f_xyyyyyyddd.hh)
n3	timestamp	Date the hourly file was last computed[1] in level n3 (not fully specified yet)

Table 14: kronos_n3 schema

(year, day, hour, source, ant) must be unique.

3 Exploiting the database

We are going to use a library for convenient access to the database directly from IDL: https://github.com/segasai/pg_idl
The project doesn't seem to be very active.
2 compilation issues:

1. PostgreSQL and pg_idl have to be built for 32 bits because IDL expects a 32 bit binary for modules as there is only a 32 bit IDL on Kronos.
2. pg_idl relies on IDL to build the module itself, and the parameters used by IDL are wrong, which results in a "duplicate symbol _kw" error, because the flag -fno-common was set.

First compile your pgsql (or modify the pg_idl to include the right configure flags) for 32 bit arch:

```
cd postgresql-9.4.1
CFLAGS=-m32 CPPFLAGS=-m32 CXXFLAGS=-m32 ./configure --build=i386-apple-darwin --host=i386-apple-darwin --t
make
```

Then you can compile pg_idl and link it with libpq.a, still using 32 bit arch:

```
cc -m32 -Ipostgresql-9.4.1/src/interfaces/libpq/ -Ipostgresql-9.4.1/src/include/ -fPIC -dynamic -I"/Applic
cc -m32 -Ipostgresql-9.4.1/src/interfaces/libpq/ -Ipostgresql-9.4.1/src/include/ -fPIC -dynamic -I"/Applic
cc -m32 -bundle -flat_namespace -undefined suppress -o "DLM/pgsql_query.so" "pgsql_query.o" "pgsql_query_u
```

Finally I added the files generated in DLM/ to $ROOT_RPWS/pro/pg_idl and added a line in the bash and csh conf files to modify IDL_DLM_PATH to load the module:

```
export IDL_DLM_PATH="<IDL_DEFAULT_DLM>:+/$ROOT_RPWS/pro/pg_idl/"
```

4 Examining available IDL code

Code in the /Volumes/HomeDirs/rpws/pro directory spans several decades, encodings, commenting style, authors and IDL versions.
There also exists an out-of-date copy on /Volumes/KronosRaid/rpws_data/rpws/pro that will be discarded.

4.1 Creation time

```
for file in *.pro; do
    date -j -f '%s' `stat -f%B $file` '+%Y'
done|sort -un|tr '\n' ' '

2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013
```

4.2 Encoding

```
file -b *.pro|sort|uniq -c|sort -rn
    197 ASCII text
      9 Non-ISO extended-ASCII text
      7 ASCII text, with CR line terminators
      3 UTF-8 Unicode text
      3 Non-ISO extended-ASCII text, with CR line terminators
      2 FORTRAN program,
      2 ASCII text, with CR, LF line terminators
      1 Pascal source, ASCII text
      1 ISO-8859 text
      1 ASCII text, with CRLF, LF line terminators
```

4.2.1 Line Ending

These files were probablly written on a VERY old Macintosh (all have a birth time around 2008-2009).

```
file -b $(grep -l $'\r' *.pro)|sort|uniq -c|sort -rn
      7 ASCII text, with CR line terminators
      3 Non-ISO extended-ASCII text, with CR line terminators
      2 ASCII text, with CR, LF line terminators
      1 ASCII text, with CRLF, LF line terminators
```

Fix them:

```
sed -i 's:\r:\n:g' $(grep -l $'\r' *.pro)
```

Now:

```
file -b *.pro|sort|uniq -c|sort -rn
    207 ASCII text
     12 Non-ISO extended-ASCII text
      3 UTF-8 Unicode text
      2 FORTRAN program,
      1 Pascal source, ASCII text
      1 ISO-8859 text
```

4.2.2 Encoding

The files marked as `Non-ISO extended-ASCII text` and `ISO-8859 text` are actually in `Mac OS Roman` encoding. To fix them:

```
for file in *.pro; do
    if file "$file"|grep -Eq "(8859|Non-ISO)"; then
        iconv -f 'macintosh' -t 'UTF-8' "$file" > "$file.conv"
        mv "$file" "$file.original"
        mv "$file.conv" "$file"
    fi
done
```

Finally:

```
file -b *.pro|sort|uniq -c
    207 ASCII text
      2 FORTRAN program,
      1 Pascal source, ASCII text
     16 UTF-8 Unicode text
```

4.3 Authors

```
cat authors.txt|tr -s '\t'|cut -f 2|sort|uniq -c|sort -rn
    134 BC
     45 PZ
     25 GB
     20 LL
     12 ??
      5 BC & LL
      3 BC & PZ
      2 EX
      1 LL | PZ
      1 EH
      1 BC & LL & PZ
      1 BC & GB
```

4.4 Comments and Documentation

`IDLDOC` was chosen to provide documentation to the Kronos library. A script, `make_idldoc.sh`, was written to automatically generate comments in the new `rst` format.
Then existing comments were merged by hand with the new format.
To generate the `HTML` documentation, `IDLDOC` can be called with:

```
idldoc, root=getenv('ROOT_RPWS')+'/pro', output=/Volumes/KronosRaid/rpws_doc/', $
   format_style='rst', markup_style='rst', /statistics, /use_latex, /debug, $
   overview=getenv('ROOT_RPWS')+'/pro/OVERVIEW.rst', title='Kronos IDL library', $
   subtitle='Library to retrieve and process CASSINI-RPWS-HFR data'
```

4.5 Versioning

All code has been moved to an `SVN` repository at address `https://version-lesia.obspm.fr/repos/waves/cassini/kronos/`. A small guide to `SVN` is available **here**.

5 Retrieving `K` files

`getkfiles` cannot be called outside of the `kronosdp` graphical interface, and `check_corres_file_directory` is buggy. Here is a simple way to mirror the Iowa repository:

```
wget -NS -r -np -nH --cut-dirs=4 \
    --accept-regex "http://cassini\.physics\.uiowa\.edu/~wsk/cas/daspage/cas_data/\
    (c|s)[0-9]{2,3}flight/(K[0-9]{7}\.[0-9]{2})?$" \
    --http-user=casrpws --http-passwd=uiowa \
    http://cassini.physics.uiowa.edu/~wsk/cas/daspage/cas_data/
```

Initial Download:

```
Total wall clock time: 18h 18m 52s
Downloaded: 78912 files, 21G in 12h 46m 6s (484 KB/s)
```

Update only:

```
Total wall clock time: 5h 20m 57s
Downloaded: 114 files, 15M in 33s (446 KB/s)
```

5.1 Disk usage

```
for file in $NAS_RPWS/*/k/K*; do
   du $file;
done | awk '{i+=$1} END {print i}'

Total size: 59200568K (59G)

du -ch $NAS_RPWS
3,7T
```

6 Rebuilding the database from scratch

To delete the current database, run:

```
psql91 -f drop_kronosdb.sql
```

To recreate the tables, functions and views:

```
psql91 -f kronos_db.sql
```

Then insert all files currently in the local file tree:

```
$ROOT_RPWS/pro/kronosdb/upsert.sh -v
```

Delete the file list in the iowa file tree to reset the `newfiles.sh` script:

```
rm -f $NAS_RPWS/iowa/.filelist
```

Now you can run the script (no need to run wget). This will copy the K files again, and more importantly, it will keep a record of the correspondance between iowa and local path in the database. K files without a counterpart on the Iowa server will still be in the database thanks to the previous steps.

```
$ROOT_RPWS/pro/kronosdb/newfiles.sh -v
```

Finally, you shoud run the `fill_periods` SQL function to set the interval for periods with a special name:

```
psql91 -c "select fill_periods();"
```