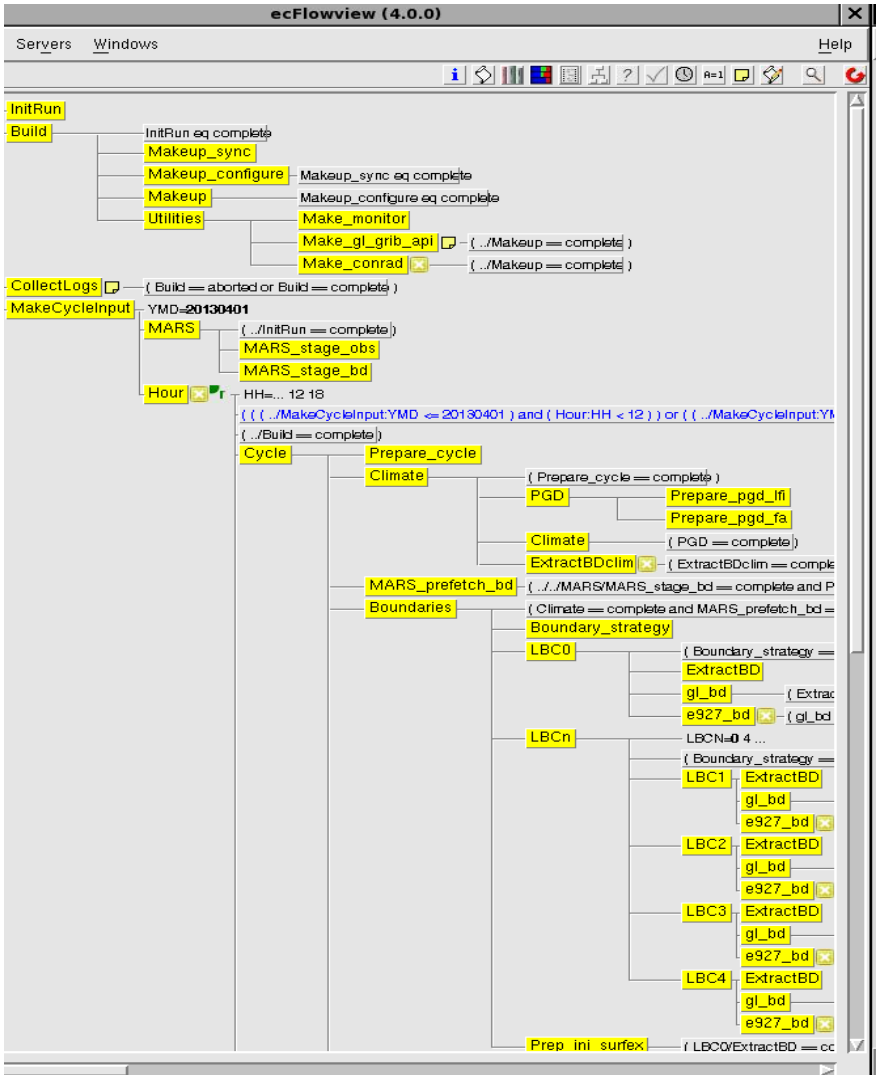
HARMONIE-AROME SCRIPTING SYSTEM AND DA

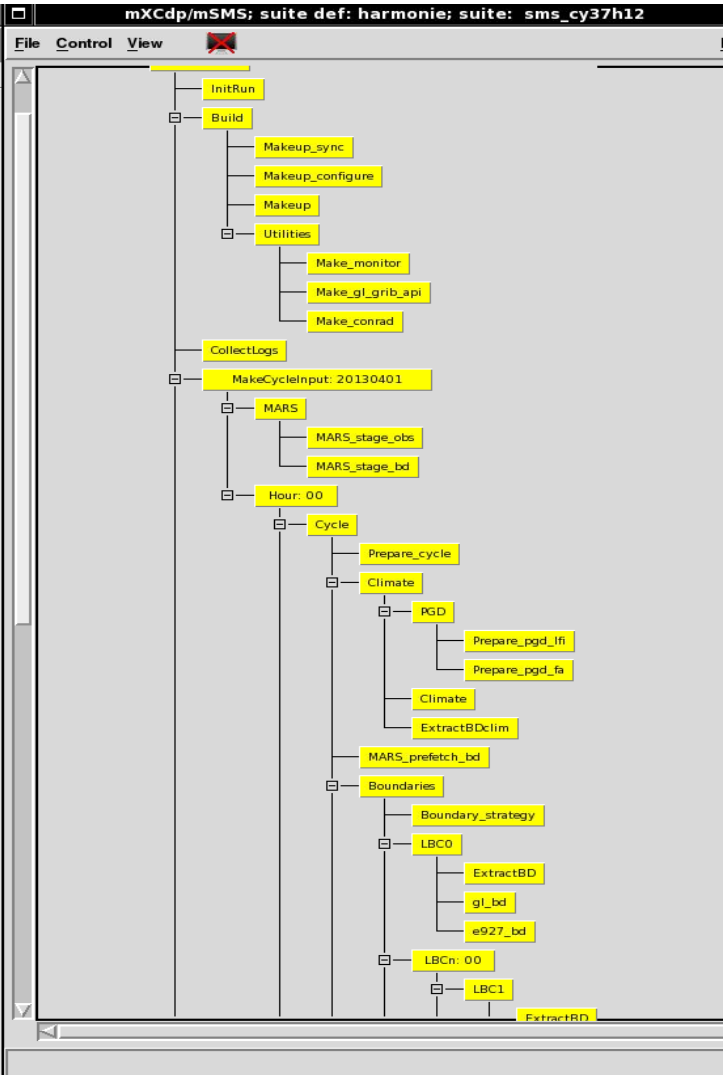
ALADIN Data Assimilation basic kit Working Days

22-23/03/2017 Lisbon, Portugal

Daniel Santos (AEMET) Eoin Whelan (MetEireann)

mini-SMS and ecFLOW





Harmonie scripting sequence mSMS

Harmonie
(top level script, perl)

Main
(old top level script, s

Action
s/Acti

Start
(reads config_exp.h
Harmonie_domains.pm
harmone_namelists.pm
surfex_namelists.pm
surfex_selected_output.pm)

1. Prepare **harmonie.def**
(and harmonie.html)

mSMS.pl
(input: harmonie.tdf;
template definition file)

2. Play **harmonie.def**

mXCdp.pl
(if $mSMS_WEBPORT == 1)

mini-SMS

- Mini-SMS is a simple **job scheduler** (perl script **mSMS.pl**). I
- Written by Gerard Cats (2000), former Hirlam system manager, in order to make the Hirlam runs at ECMWF (and locally) run more efficiently.
- It was inspired by, and is a subset of **SMS**, the **Supervisor and Monitoring System**, developed at ECMWF.
- The main advantages of mini-SMS compared to full SMS is that it is easy to port to a new system, e.g., your laptop and also makes it easier for the system managers to implement new features when this is required.

- Mini-SMS working:

- A description of the suite of programs to run, possibly **distributed** **(template) suite definition file.**
- The order of the programs must be executed are controlled using **control structures**

- In Harmonie, things related to mini-SMS are located in subdirectory **msms**. The main template definition file is **harmonie.tdf**, but there are also others.

mini-SMS

- Mini-XCdp is a graphical user interface **(GUI**) to the mini-SMS scheduler (perl script **mXCdp.pl**).
- It communicates with mini-SMS by sending HTTP requests (small **WebServer.pl** is included by `mSMS.pl` on demand
- The name mini-XCdp is not as closely ECMWF's **XCdp** (X Control and display program) as mini-SMS follows full SMS. Possibilities to interact with the scheduler:
  - If a task aborts, it can be restarted from the GUI, without rerunning the whole suite.
  - Log files (and job container scripts) can be viewed.
  - Job status can be overridden, e.g. forced to “complete".
  - Tasks or families can be suspended (and resumed).
  - (Active) jobs can be killed (if implemented for the local system)

- The GUI can be started also if the scheduler is currently not running:
  - State the scheduler was when it terminated
  - Restart the scheduler from exactly where it left off and continue the run (manual intervention).

mini-SMS

- Mini-XCdp is a graphical user interface **(GUI)** to the mini-SMS scheduler (perl script **mXCdp.pl**).
- It communicates with mini-SMS by sending HTTP requests (small **WebServer.pl** is included by `mSMS.pl` on demand
- The name mini-XCdp is not as closely ECMWF's **XCdp** (X Control and display program) as mini-SMS follows full SMS. Possibilities to interact with the scheduler:
  - If a task aborts, it can be restarted from the GUI, without rerunning the whole suite.
  - Log files (and job container scripts) can be viewed.
  - Job status can be overridden, e.g. forced to “complete".
  - Tasks or families can be suspended (and resumed).
  - (Active) jobs can be killed (if implemented for the local system)

- The GUI can be started also if the scheduler is currently not running:
  - State the scheduler was when it terminated
  - Restart the scheduler from exactly where it left off and continue the run (manual intervention).

mini-SMS

- The user should normally never invoke the scripts `mSMS.pl` or `mXCdp.pl`
- SMS accepts several things in the .def file that mini-SMS does not, e.g., if- and loop-statements. These are only understood in the preparation step of mini-SMS, making these constructs less dynamic than in full SMS.
- **`Harmonie start DTG=...`** will usually automatically invoke also **`mXCdp.pl`** in addition to **`mSMS.pl`**.
- A new GUI can be opened later by the command `Harmonie mon`
- It is possible to have more than one GUI open for the same experiment, e.g., one at work and another one at home.
- At ecgb, mSMS is now submitted as a batch job in a special queue

mini-SMS

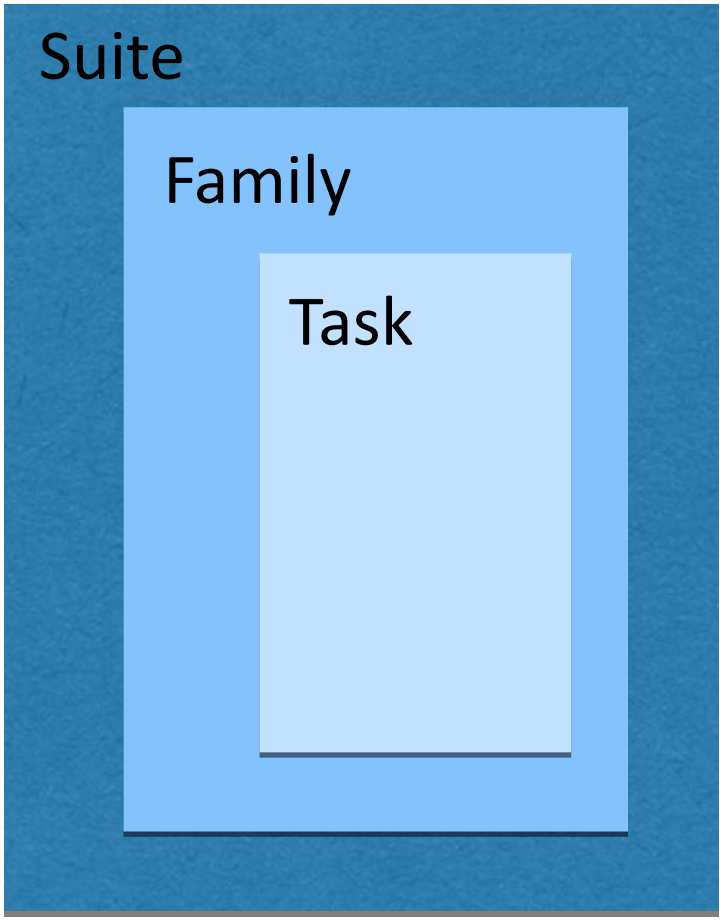
- The definition file(s) (e.g., harmonie.tdf) describes the system to be run in terms of **suites, families, tasks** and **control structures**

- **Family** is just a group of tasks and/or other families.
- **Suite** is a top-level family.
- **Tasks** need a **"task".sms container** script)

  - In Harmonie, many containers are simply symbolic links to **default.sms**, which invokes a script named **“task"**
  - All containers should include the file **sms.h** and at the top, and **end.h** at the bottom.

- **Triggers** allow to specify dependencies between tasks, e.g. that one task must wait for another task to complete before it can start execution.

mini-SMS


Suite
Family
Task


Definition file
sms
container
Script
executable
…

mini-SMS task execution

- **%SMSTRYNO%** is the attempt number of the task. %SMSTRYNO% runs from 1 to **%SMSTRIES%** (default 1) for automatically submitted tasks, but %SMSTRIES% is ignored for tasks that are rerun through the GUI.
- **"task".job%SMSTRYNO%-q**. Headers (for the queueing system) and footers might have been added.

”task”.sms (container scri | **”task”.job%SMSTRYNO%** (sh script)

Submit.pl (Universal Job Submission Filter) | **submission.db** (**Env_submit** reader and header and foorter adder)

”task”.job%SMSTRYNO%-q (sh script)

ecFLOW

- **ecFLOW** is the new ECMWF's workflow manager
- Development of SMS has now stopped. (Not supported on new platforms)
- ecFlow is a complete rewrite using **object oriented methodology**. The rewrite will help improve maintainability, allow easier modification and introduce object orientated features.
- Proprietary script languages, such as CDP, have been replaced by **Python**.
- **Multiple suites can monitored at the same time**
- **Server/Client structure with backup server and log server capabilities**.
- **Dynamical suite definition in python**. Not template parsing needed.
- **Text based suite definition** allow backwards compatibility. More control options of each family, tasks, variables and more powerful GUI than mSMS.

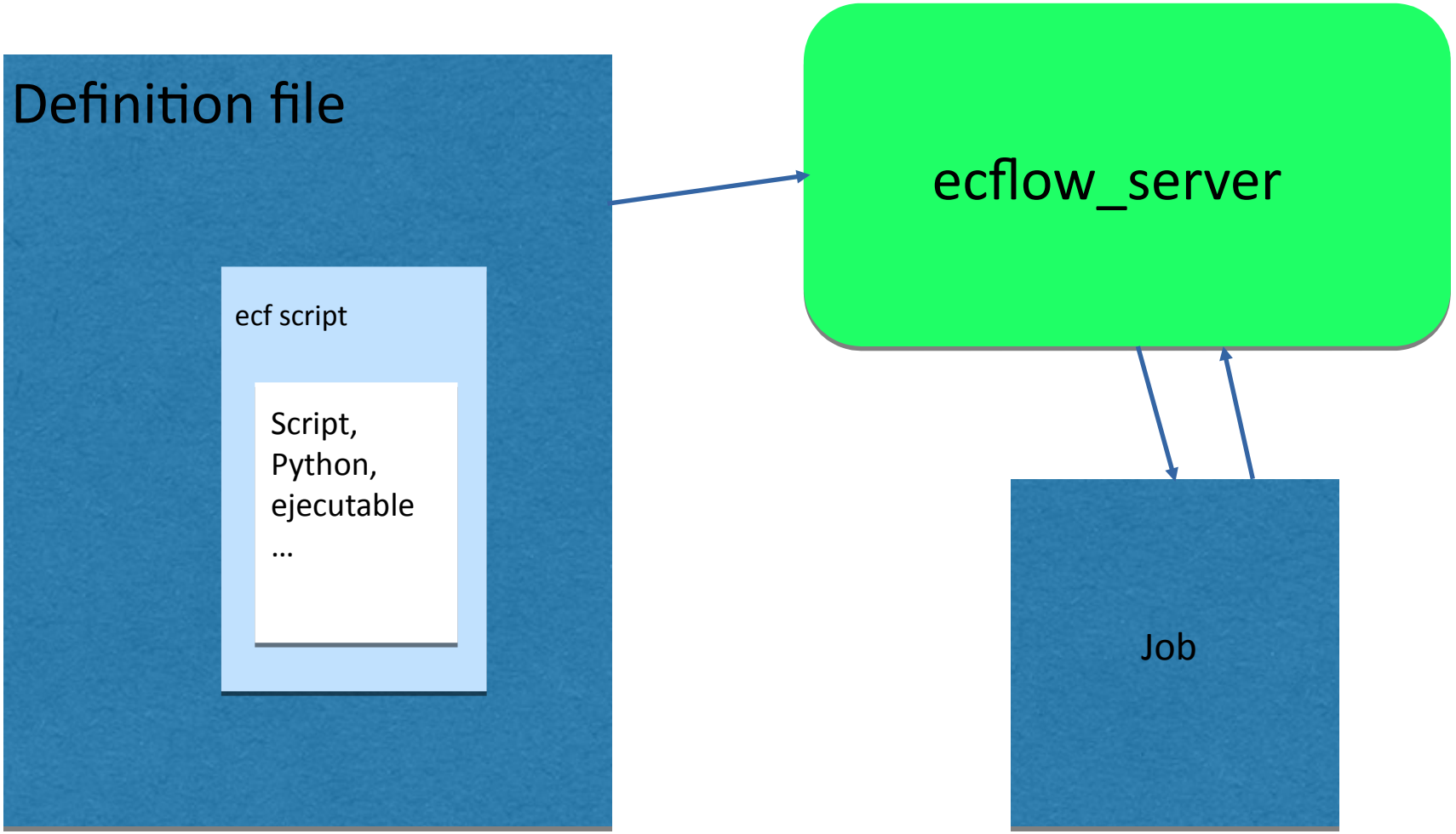
ecFLOW

ecFLOW	*mSMS*
Server/Client	*batch Script*
Dynamical suite def	*Static suite def*
Python/CDP	*CDP*
*Scripts *.ecf*	*Scripts *.sms*
More control and options	*Few options*

ecFLOW

- **ecflow_server** The schedule a daemon runs continuously(nohup &)
- **ecflow_client** Command line interface with ecFlow Child commands updtate the task state in ecflow_server
- **Python API**
- **ecflowview** or **ecflow_UI** ecFlow GUI
- **Several experiments and servers can be monitored with the same viewer**
- More than one experiment is not allowed with the same name monitored in the same server.

ecFLOW


Definition file
ecf script
Script,
Python,
ejecutable
…
ecflow_server
Job

HARMONIE-AROME SCRIPTING SYSTEM UPGRADE STRATEGY

- Reduce the number of languages: Shell Script, Perl, Python …
- Improve the use of env Variables
- Use ECMWF scheduler like glameps in stead of job1, job1-q strategy.
- Set up control variables at scheduler level not inside the script.
- Improve the control over the task making them more modular

HARMONIE-AROME DEFAULT SETUP

The highest level of configuration is done in [config_exp.h.]

Assimilation

Data assimilation settings. More assimilation related settings, in particular what observations to assimilate, can be found in include.ass

```
# **** Assimilation ****
ANAATMO=3DVAR                          # Atmospheric analysis (3DVAR|4DVAR|blending|none)
ANASURF=CANARI_OI_MAIN                 # Surface analysis (CANARI|CANARI_OI_MAIN|CANARI_EKF_SURFEX|none)
                                       # CANARI            : Old style CANARI
                                       # CANARI_OI_MAIN    : CANARI + SURFEX OI
                                       # CANARI_EKF_SURFEX : CANARI + SURFEX EKF ( experimental )
                                       # none              : No surface assimilation
ANASURF_MODE="before"                  # When ANASURF should be done
                                       # before            : Before ANAATMO
                                       # after             : After ANAATMO
                                       # both              : Before and after ANAATMO (Only for ANAATMO=4DVAR)
INCV="1,1,1,1"                         # Active EKF control variables. 1=WG2 2=WG1 3=TG2 4=TG1
INCO="1,1,0"                           # Active EKF observation types (Element 1=T2m, element 2=RH2m and element 3=Soil moisture)

MAKEODB2=no                            # Conversion of ODB-1 to ODB-2 using odb_migrator

SST=BOUNDARY                           # Which SST fields to be used in surface analysis
                                       # BOUNDARY          : SST interpolated from the boundary file. ECMWF boundaries utilize a special m
                                       #                     HIRLAM and HARMONIE boundaries applies T0M which should be SST over sea.
LSMIXBC=no                             # Spectral mixing of LBC0 file before assimilation
[ "$ANAATMO" = 3DVAR ] && LSMIXBC=yes
JB_INTERPOL=no                         # Interpolation of structure functions from a pre-defined domain to your domain
```

- *ANAATMO*: Atmospheric analysis (3DVAR|4DVAR|blending|none)
- *ANASURF*: Surface analysis (CANARI|CANARI_OI_MAIN|CANARI_EKF_SURFEX|none). See surfex_namelists.pm for more info.
- *ANASURF_MODE*:When the surface should be called (before|after|both)
- *INCV*: Active EKF control variables. 1=WG2 2=WG1 3=TG2 4=TG1 (0|1)
- *INCO*: Active EKF observation types (Element 1=T2m, element 2=RH2m and element 3=Soil moisture) (0|1)
- *MAKEODB2*: Option to convert ODB-1 databases to ODB-2 files for DA monitoring
- *SST*: which sea surface temperature field to use in the surface analysis
- *LSMIXBC* Spectral mixing of LBC0 file before assimilation (no|yes)
- *JB_INTERPOL* Interpolation of structure functions from a pre-defined domain to your domain (no|yes). Note that this has to be used with some caution.

https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/ConfigureYourExperiment

HARMONIE-AROME DEFAULT SETUP

The highest level of configuration is done in config_exp.h.

Observations

```
# **** Observations ****
OBDIR=$HM_DATA/observations             # Observation file directory
RADARDIR=$HM_DATA/radardata             # Radar observation file directory
SINGLEOBS=no                            # Run single obs experiment with observation created by scr/Create_single_obs (no|yes)

USE_MSG=no                              # Use MSG data for adjustment of inital profiles, EXPERIMENTAL! (no|yes)
MSG_PATH=$SCRATCH/CLOUDS/               # Location of input MSG FA file, expected name is MSGcloudYYYYMMDDHH
```

- *OBDIR*: Defines the directory that your (BUFR) observation files (obYYYYMMDDHH) are to read from
- *RADARDIR*: Defines the directory that your (OPERA HDF5) radar observation files are to be read from. BALTRAD OPERA HDF5, MF BUFR and LOCAL files are treated in scr/Prepradar
- *SINGLEOBS* Run single obs experiment with synthetic observation created by [source:trunk/harmonie/scr/Create_single_obs scr/Create_single_obs) (no|yes)
- *USE_MSG*: Use MSG data for adjustment of inital profiles, EXPERIMENTAL! (no|yes)
- *MSG_PATH*: Location of input MSG FA file, expected name is MSGcloudYYYYMMDDHH. Note that the pre-processing software to generate input files is not yet included in HARMONIE

4DVAR settings

4DVAR settings (experimental)

```
# **** 4DVAR ****
NOUTERLOOP=1                            # 4DVAR outer loops, need to be 1 at present
ILRES=2,2                               # Resolution (in parts of full) of outer loops
TSTEP4D=360,360                         # Timestep length (seconds) of outer loops TL+AD
TL_TEST=yes                             # Only active for playfile tlad_tests
AD_TEST=yes                             # Only active for playfile tlad_tests
```

- *NOUTERLOOP*: Number of outer loops, need to be 1 at present
- *ILRES*: Resolution (in parts of full) of outer loops
- *TSTEP4D*: Timestep length (seconds) of outer loops TL+AD
- *TL_TEST*: Only active for playfile tlad_tests (yes|no)
- *AD_TEST*: Only active for playfile tlad_tests (yes|no)

Digital filter settings

Digital filter initialization settings if DFI is not equal to "none"

```
# **** DFI setting ****
TAUS=5400                               # cut-off frequency in second
TSPAN=5400                              # 7200s or 5400s
```

- *TAUS* cut-off frequency in seconds
- *TSPAN* length of DFI run in seconds

https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/ConfigureYourExperiment

HARMONIE-AROME DEFAULT SETUP

The highest level of configuration is done in config_exp.h.

Observation monitoring and general diagnostics

```
# *** Observation monitoring ***
OBSMONITOR=obstat:plotlog                # Create Observation statistics plots
                                         # Format: OBSMONITOR=Option1:Option2:...:OptionN
                                         # obstat: Daily usage maps and departures
                                         # plotlog: IFS log statistics
                                         #  - Grid point and spectral norms evolution
                                         #  - Cost function evolution, if applicable
                                         #  - Observation usage from the minimization, if applicable
                                         # no: Nothing at all
                                         #
                                         # The assimilation related monitoring is
                                         # Only active if ANAATMO != none
```

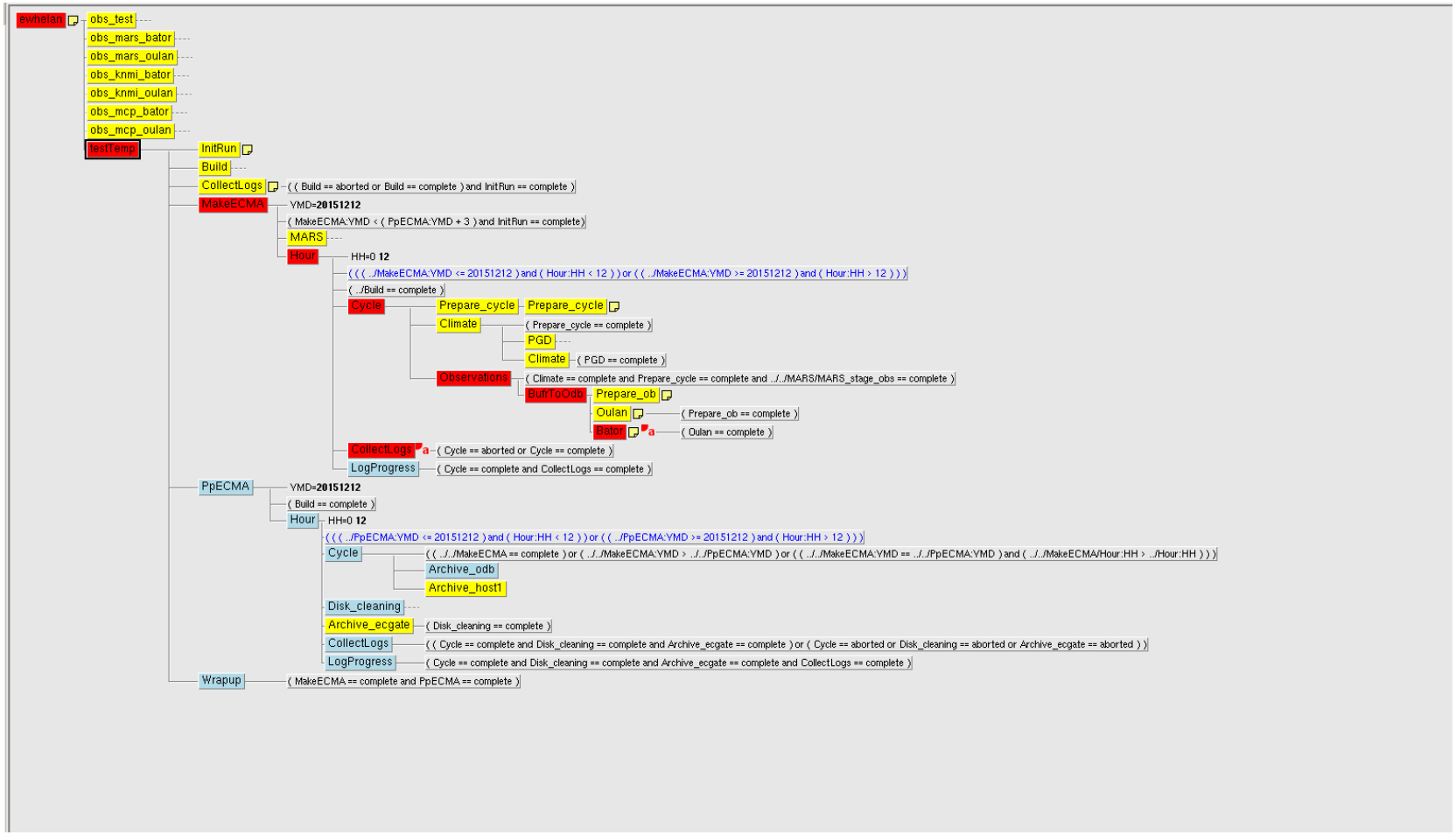
OBSMONITOR Selection for observation statistics plots

- obstat Observations usage. Read more here.
- plotlog IFS log statistics
  - Grid point and spectral norms evolution
  - CPU cost
  - Surface assimilation increments, if applicable
  - Cost function evolution, if applicable
  - Observation usage from the minimization, if applicable
- no No monitoring

Note that this is only active if ANAATMO != none

https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/ConfigureYourExperiment

DATA ASSIMILATION



Input files - observations (ODB)

- Oulan used by default to convert conventional BUFR to ASCII
- BATOR can process conventional BUFR
  - ECMWF, KNMI, SMHI, MET Norway, Met Éireann data tested
  - Set USEOBSOUL=0
- BATOR used to process all other data types
- COPE prototype available too

Processing of radar data

- OPERA (HDF5) radar data can be read directly by BATOR
  - https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/RadarData
- prepopera.py python script to further process data
  - Simple thinning
  - Superobbing
  - More documentation to come!

Structure function derivation

- Based on data generated with ensemble forecasts downscaled from ECMWF EPS runs
- 4-member ensemble used to generate forecast differences (FESTAT=yes) – single suite
- FEMARS used to produce structure function files
  - run outside of system
- https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/Structurefunctions_ensys

Diagnostics: DFS

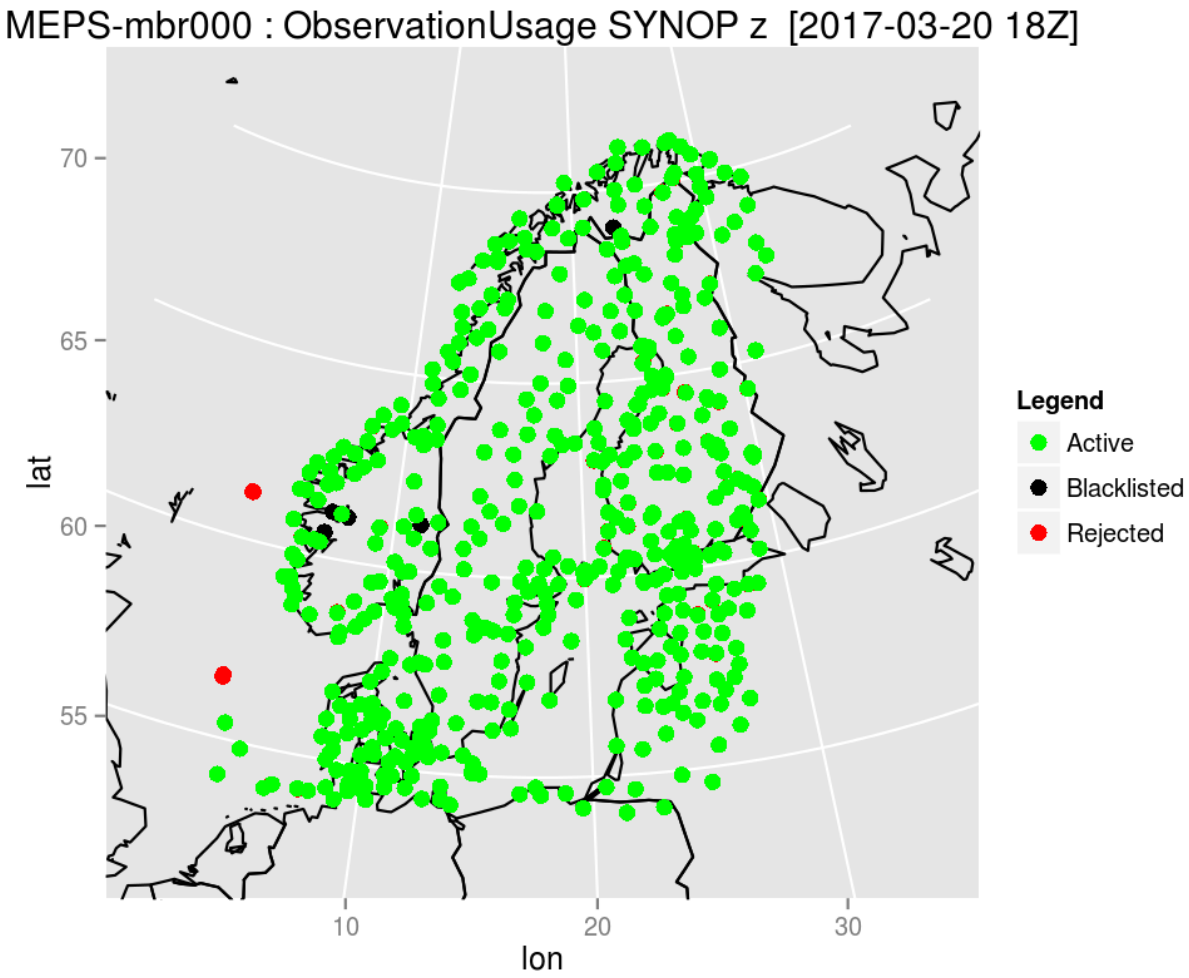
- Degrees of freedom for signal diagnostic
- Chapnik et al, 2006
- DFS=(y*-y)R^-1(Hx*-Hx)
  - * denotes perturbed values
- Still under development
- To be implemented using ensemble system
- Code and graphics software available on wiki
  - https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/DFS

Diagnostics: MTEN

- Moist total energy norm
- Storto & Randriamampianina, 2010
- Evaluate observation impact
- Still under development
- Computation has been tested
- To be implemented in ensemble setup

Obsmon

- Fortran + SQLite
  - Normally run as a part of your experiment
  - obsmon_stat: Reading ODB, do some statistics and write selected
  - information from ODB to SQLite tables.
  - obsmon_link_stat: Gather individual SQLite databases into large ones
- R + SQLite + shiny
  - Visualization. Reading the SQLite databases and visualize it in a browser.
  - Offline
  - From a server
  - needs shiny-server


MEPS-mbr000 : ObservationUsage SYNOP z [2017-03-20 18Z]
lat
70
65
60
55
lon
10
20
30
Legend
Active
Blacklisted
Rejected

ODB2 output

- Possible to generate ODB2 (feedback) output
- Output suitable for use by OBSTAT
- ODB2 format useful for monitor DA systems
- Uses ODB-API software (odb_migrator)
  - https://software.ecmwf.int/wiki/display/ODBAPI/ODB+API+Home
  - SQL engine
  - C/C++, FORTRAN, Python interfaces
- Used in COPE

COPE

- Still working as prototype in cy40h1 branch
- ODB splitting was an issue
- Splitting less of an issue with ODB-API and ECML
  - (I think!)
- SAPP/ODB Server implementation may not be attractive for local use
- ODB reading by IFS will change in future
  - More flexible
  - Read ODB2 directly
  - Read other formats?