dr2xml User Guide

Stéphane Sénési, CNRM, Météo France - Marie-Pierre Moine, Cerfacs

Document version: 0.6 (20/12/2016)
Data Request Version: 01.beta.45
Dr2xml version: 0.8

Introduction

One of the big challenges in CMIP6 is to produce the ensemble of diagnostics requested by the CMIP6 Data Request (DR). Configuring the model outputs so that they comply with the Data Request is nearly unachievable by hand, given thousands of variables, hundreds of experiments to handle for groups participating to numbers of MIPs. Those models that use XIOS to handle their I/O can benefit from dr2xml to help writing XIOS configuration XML files so that output data files meet the CMIP6/CMOR format and metadata requirements. Dr2xml exploits the CMIP6 Data Request API to automate the generation of XIOS XML configuration files.

To meet the CMIP6/CMOR data requirements, dr2xml relies on essential XIOS functionalities:

- flexibility in the file naming and structure (multi or mono-variable files, mono-variable format being the CMIP standard)
- flexibility in the time splitting length (useful to generate manageable file size)
- possibility to glue any global attributes to the file (useful to include the set of mandatory CMOR global attributes)
- ability to perform "on the fly" temporal and spatial operations, used to conform to CMIP6 *"cell_methods"* and *"spatial_shapes"*

Dr2xml fully exploits the Data Request content and scoping tools offered by the Data Request Python API (https://earthsystemcog.org/projects/wip/CMIP6DataRequest) . Given some settings provided by the user, e.g. defining the MIP(s) the lab is involved in, the current experiment/simulation to be run, maximum priority for the outputs (see section 2 for the complete list of settings) dr2xml automatically:

- identifies the list of requested CMOR variables that applies;
- collects CMIP6 metadata associated to experiment(s) and CMOR variables (e.g.*: experiment_id, variable_id, standard_name, long_name*);
- gets temporal shape (e.g.: *'time mean over sea ice'*) and spatial shape (e.g.: *'global field 7 pressure levels'; 'ocean basin meridional section'*) of each required CMOR variable.

1. dr2xml key features

Dr2xml is a Python package for translating the CMIP6 Data Request to XIOS ***file_def*** XML files. It also produces ***complementary field_def*** and ***axis_def*** XML files [-not yet available-]; these will complement hand-written XML files: the ***native field_def*** of the model and what we call, in dr2xml jargon, the "***ping file"***.

Dr2xml generates one ***file_def*** XML file per XIOS context (i.e. per realm or group of realms). That *file_def* sets the file structure and name in conformance with the Data Reference Syntax (DRS, see the WIP position paper: https://drive.google.com/file/d/0B-X2uY_FGt7XNWI2TzBsUXdCVkE/view), includes the global attributes (metadata about the experiment, the model, the producing laboratory, etc.) and local attributes (metadata about the output CMOR variables) imposed by CMOR. Temporal and spatial operations consistent with the specified DR *cell_methods,* to be performed on the instantaneous and global variables of the ***native field_def***, are automatically coded by dr2xml with appropriate XIOS filters in a ***complementary field_def*** [-not yet available-], possibly referring to ***complementary axis_def*** [-not yet available-].

Complementary field_def files aggregate XIOS temporal and spatial operations required to produce CMOR Variables from model outputs (for e.g.: axis reduction to compute zonal means, temporal averages for global indices computation)

The ***ping file*** establishes the correspondence between model variable names listed in the *native model field_def* and the variables names referenced in the *file_def* produced by dr2xml (dr2xml being agnostic of the native model variable names, this *ping file* represents the interface between model and dr2xml). Given the long list of variables requested by the DR, a tool, provided along with dr2xml sources, enables to generate a template for an exhaustive *ping file*. The "only" modeller work is to scan each line of this skeleton and associate (when it exists) a native model variable name to each of the expected output variable. This work is to be done once for all, once the Data Request is stabilized (see section 4 for details).

Dr2xml also offers two functionalities allowing the user to deviate/modulate the variable list that comes from the official Data Request, giving the possibility to the user to define:

- a ***list of excluded variables***: among the DR variables list, these ones will be excluded from the resulting *file_def* (because they do not apply to the model or for any good reason the modeller decide not to output them); an alternate way to exclude variables is through ping file syntax (see section 4 - not yet implemented)
- a ***list of home variables***: a list of variables that the lab intends to produce in some case. This covers both purely *"home"* variables (i.e. which have no CMOR equivalent), and CMOR

Variables that are not requested by the DR for this MIP/experiment but the lab decides to output anyway; these ones will be added in the *file_def.*

Excluded-variables usage is described in section 2*; home-variables list* functionality is described in section 3.

2. How to install, configure and run dr2xml

Prerequisites

You need Python 2.7. For some examples provided as iPython notebooks, you need either iPython or the pure-python version of the notebooks (they are provided).

You need to have the CMIP6 Data Request package (available at http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest) installed.

```
svn co http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest <My-CMIP6Dreq-Rep>
cd <My-CMIP6Dreq-Rep>
python setup.py install --user
```

And you must include `<My-CMIP6Dreq-Rep>/dreqPy` in your `PYTHONPATH`.

You will also need a local copy of CMIP6 Controlled Vocabulary. You will need to provide the path *<My-CMIP6CVs-Rep>* where you install this CMIP6_CVs later on, at the dr2xml configuration stage.

```
git clone https://github.com/WCRP-CMIP/CMIP6_CVs <My-CMIP6CVs-Rep>
```

Installing dr2xml

Dr2xml is available on GitHub: https://github.com/senesis/dr2pub

```
git clone https://github.com/senesis/dr2pub <My-dr2xml-Rep>
```

Configuring dr2xml

To configure dr2xml, the user has to fill 2 Python dictionaries, one related to the model and the laboratory choices and another related to the simulation to run.

When keys of these two setting dictionaries are CMOR attributes they are flagged with stars:

- (***) *always required*
- (**) *required when relevant*
- (*) *never required*
- *(*r) never required but recommended*

For a detailed descriptions and possible values for these attributes, see the WIP position paper *"CMIP6 Global Attributes, DRS, Filenames, Directory Structure, and CV's"*: https://drive.google.com/file/d/0B-X2uY_FGt7XNWI2TzBsUXdCVkE/view)

A) Define the settings specific to the laboratory and the model

lab_and_model_settings is a dictionary containing:

(1) settings for global CMOR attributes referencing the laboratory and the model:

- `institution_id(***) [string]`: institution identifier
- `model_id [string]`: model identifier (corresponds to `source_id(***)`)
- `source_type(***) [string]`: model configuration
- `references(*) [string]`: paper or web-based document that describes the data or the way to produce them
- `info_url [string]`: location of documentation (corresponds to `further_info_url(***)`)
- `contact(*) [string]`: email address of the data producer

(2) settings providing information to let dr2xml know what to extract from the Data Request and how to complete it, the case being:

- `mips [python dictionary]`: a set which keys are MIP names (no values)
- `max_priority [integer]`: number indicating the maximum priority to consider for output variables
- `tierMax [integer]`: number indicating the maximum tier to consider for experiments
- `ping_variables_prefix [string]`: the tag used to prefix the variables in the '*field id*' namespaces of the ping file; may be an empty string
- `excluded_vars [python list]`: list of CMOR variables to exclude from the result based on previous Data Request extraction
- `listof_home_vars [string]`: full path to the file that contains the list of home variables to be taken into account, in addition to the Data Request (see section 3)

(3) settings to inform how the different realms are grouped into XIOS contexts:

- `realms_per_context [python dictionary]`: a python dictionary which keys are context names and values the lists of realms associated to each context.
- `orphan_variables [python dictionary]`: a python dictionary with (context name, list of variables) as (key,value) pairs, where the list indicates the variables to be re-affected to the key-context (initially affected to a realm falling in another context).
- `comments [python dictionary]`: a python dictionary which keys are CMOR variable names and values a free comment the user wants to associate to the key variable.

An example of *lab_and_model_settings* with useful comments is provided below (extracted from the Python Notebook `DR2XML.ipynb` delivered with dr2xml sources in `dr2pub/` repository).

```
lab_and_model_settings={
    'institution_id': "CNRM-CERFACS", # institution should be read in CMIP6_CV, if up-to-date
    #'institution'   : "Centre National de Recherches Meteorologiques",
    'source_types' : { "CNRM-CM6-1" : "AOGCM", "CNRM-CM6-1-HR" : "AOGCM",
                       "CNRM-ESM2-1": "ESM"  , "CNRM-ESM2-1-HR": "ESM" },
    'source_id'      : "CNRM-CM6-1",
    'source'         : "CNRM-CM6-1", # Useful only if CMIP6_CV is not uptodate
    #'source_type' : "AER"  # You may override here the source-type value deduced from source_id and sources_type
    'references'     :  "A character string containing a list of published or web-based "+\
        "references that describe the data or the methods used to produce it."+\
        "Typically, the user should provide references describing the model"+\
        "formulation here",
    'info_url'       : "http://www.umr-cnrm.fr/cmip6/",
    'contact'        : 'contact.cmip@cnrm.fr',

    # We account for the list of MIPS in which the lab takes part.
    # Note : a MIPs set limited to {'C4MIP'} leads to a number of tables and
    # variables which is manageable for eye inspection
    'mips': {'C4MIP', 'SIMIP', 'OMIP', 'CFMIP', 'RFMIP'} ,
    'mips_all' : {'AerChemMIP','C4MIP','CFMIP','DAMIP', 'FAFMIP' , 'GeoMIP','GMMIP','ISMIP6',\
                  'LS3MIP','LUMIP','OMIP','PMIP','RFMIP','ScenarioMIP','CORDEX','SIMIP'},
    #'mips' : {'HighResMIP'},

    # Max variable priority level to be output
    'max_priority' : 1,
    'tierMax'      : 3,

    # The ping file defines variable names, which are constructed using CMIP6 "MIPvarnames"
    # and a prefix which must be set here, and can be the empty string :
    "ping_variables_prefix" : "CMIP6_",

    # We account for a list of variables which the lab does not want to produce ,
    # oragnized by realms
    # excluded_vars_file="./inputs/non_published_variables.txt"
    "excluded_vars":[],

    # We account for a list of variables which the lab wants to produce in some cases
    #"listof_home_vars":"./inputs/my_listof_home_vars.txt",
    "listof_home_vars":"./config_utest/utest020_listof_home_vars.txt",

    # Each XIOS  context does adress a number of realms
    'realms_per_context' : { 'nemo': ['seaIce', 'ocean', 'ocean seaIce', 'ocnBgchem', 'seaIce ocean'] ,
                         'arpsfx' : ['atmos', 'atmos atmosChem', 'aerosol', 'atmos land', 'land',
                                     'landIce land',  'aerosol land','land landIce',  'landIce', ],
                        },
    # Some variables, while belonging to a realm, may fall in another XIOS context than the
    # context which handles that realm
    'orphan_variables' : { 'nemo' : ['dummy_variable_for_illustration_purpose'],
                           'arpsfx' : [],
                           },
    'vars_OK' : dict(),
    # A per-variable dict of comments valid for all simulations
    'comments'     : {
        'tas' : 'nothing special about tas'
        }
    }
```

B) Define the settings specific to the simulation

simulation_settings is a dictionary where the user chooses the simulation of interest and fixes the corresponding global attributes, following the DRS/CMOR standards:

(1) Attributes related to the current simulation:

- `experiment_id(***) [string]` : root experiment identifier
- `contact(*) [string]` : only if contact for this simulation differs from the default contact fixed in *lab_and_model_settings*
- `project [string]` : CMIP6 is the default
- `source_type(***) [string]` : model configuration ; needed if specific to the current simulation; overwrites `source_type` from *lab_and_model_settings*
- `activity_id(***) [string]` : MIP or MIPs name(s)
- `variant_info(*r) [string]` : description of run variant
- `realization_index(***) [integer]` : realization number
- `initialization_index(***) [integer]` : Index for variant of initialization method
- `physics_index(***) [integer]` : index for model physics variant
- `forcing_index(***) [integer]` : index for variant of forcing

(2) Attributes related to the parent simulation and branching scheme:

- `parent_experiment_id(**) [string]`: parent experiment identifier
- `branch_method(**) [string]`: branching procedure
- `branch_time_in_parent(**) [string]`: branch time with respect to parent time axis
- `branch_time_in_child(**) [string]`: branch time with respect to child's time axis
- `parent_time_units(**) [string]`: time units used in parent
- `parent_variant_label(**) [string]`: parent variant label
- `parent_mip_era(**) [string]`: parent's associated MIP cycle
- `parent_activity [string]`: parent activity identifier (corresponds to `parent_activity_id(**)`)
- `parent_source_id(**) [string]`: parent model identifier
- `sub_experiment_id(***) [string]`: sub-experiment identifier
- `sub_experiment(***) [string]`: description of sub-experiment
- `history(*) [string]`: In case of replacement of previously produced data, description of any changes in the production chain.
- `comments(*) [dictionary]`: A per variable dictionary of comments which are specific to this simulation. It will replace the all-simulation comment present in *lab_and_model_settings*

An example of *simulation_settings* with useful comments is provided below (extracted from the Python Notebook `DR2XML.ipynb` delivered with dr2xml sources in `dr2pub/` repository).

```
simulation_settings={
    # Dictionnary describing the necessary attributes for a given simulation

    # Warning : some lines are commented out in this example but should be
    # un-commented in some cases. See comments

    "experiment_id"  : "historical",
    #"experiment_id"  : "Forced-Atmos-Land",
    #"experiment_id"  : "Coupled",
    #"experiment_id"  : "DCPP-C13",

    #"contact"        : "", set it only if it is specific to the simualtion
    #"project"        : "CMIP6",  #CMIP6 is the default

    #'source_type'    : "ESM" # If source_type is special only for this experiment (e.g. : AMIP)
                      #(i.e. not the same as in lab_and_model settings), you may tell that here

    # MIP specifying the experiment. For historical, it is CMIP6 itself
    # In a few cases it may be appropriate to include multiple activities in the activity_id
    # (with multiple activities allowed, separated by single spaces).
    # An example of this is "LUMIP AerChemMIP" for one of the land-use change experiments.
    "activity_id"       : "CMIP", # examples : "PMIP", "LS3MIP LUMIP"; defaults to "CMIP6"

    # It is recommended that some description be included to help identify major differences among variants,
    # but care should be taken to record correct information.  Prudence dictates that this attribute includes
    # a warning along the following lines:  "Information provided by this attribute may in some cases be flawed.#
    # Users can find more comprehensive and up-to-date documentation via the further_info_url global attribute."
    "variant_info"      : "Start date chosen so that variant r1i1p1f1 has the better fit with Krakatoa impact on tos",
    #
    "realization_index"    : 1, # Value may be omitted if = 1
    "initialization_index" : 1, # Value may be omitted if = 1
    "physics_index"        : 1, # Value may be omitted if = 1
    "forcing_index"        : 1, # Value may be omitted if = 1
    #
    # All about the parent experiment and branching scheme
    "parent_experiment_id" : "piControl", # omit this setting (or set to 'no parent') if not applicable
                                          # (remaining parent attributes will be disregarded)
    "branch_method"        : "standard", # default value='standard', meaning ~ "select a start date"
    "branch_time_in_parent": "365.0D0", # a double precision value, in parent time units,
    "branch_time_in_child" : "0.0D0", # a double precision value, in child time units,
    #'parent_time_units'    : "" #in case it is not the same as child time units
    #'parent_variant_label' :""  #Default to 'same as child'. Other cases should be expceptional
    #"parent_mip_era"       : 'CMIP5'   # only in special cases (e.g. PMIP warm start from CMIP5/PMIP3 experiment)
    #'parent_activity   '   : 'CMIP'    # only in special cases, defaults to CMIP
    #'parent_source_id'     : 'CNRM-CM5.1' #only in special cases, where parent model is not the same model
    #
    "sub_experiment_id"    : "none", # Optional, default is 'none'; example : s1960.
    "sub_experiment"       : "none", # Optional, default in 'none'
    "history"              : "none", #Used when a simulation is re-run, an output file is modified ....
    # A per-variable dict of comments which are specific to this simulation. It will replace
    # the all-simulation comment present in lab_and_model_settings
    'comments'     : {
        'tas' : 'tas diagnostic uses a special scheme in this simulation',
        }
    }
```

Running dr2xml

Once filled-in the previous 2 dictionaries of settings, dr2xml is simply executed by:

a) importing the appropriate module from dr2xml:

```
from dr2xml import generate_file_defs
```

b) setting the paths to your local copy of CMIP6 CVs:

```
my_cvspath=<My-CMIP6CVs-Rep>
```

c) generating *file_def* for one of the context defined in the settings, here *"nemo"*:

```
generate_file_defs(lab_and_model_settings, simulation_settings,
year=2000,context='nemo', printout=True, cvs_path=my_cvspath)
```

where `year` is an optional argument (default is `None)` that allows to take into account the current simulated year in the Data Request filtering.

To configure and run dr2xml, you can either use the iPython notebook `DR2xml.ipynb` available in `dr2pub/` repository or the classical Python script `DR2xml.py` available in `dr2pub/doc/`.

3. How to use home-variables list functionality

Although the Data Request is plethoric, modellers may want to output more or different variables than the ones planned by the DR, either existing or not as a CMOR variable. They also may simply want to ensure that a crucial variable will be output, without worrying whether this latter is included in the DR or not.

In such cases, the user is invited to fill-in a list of home variables; an example is shown below.

```
------------------------------------------------------------------------------------------------------------------------------------------
TYPE;   VARNAME;        REALM;    FREQUENCY;    TABLE;      TEMPORAL_SHP;       SPATIAL_SHP;      EXPNAME;                MIP
------------------------------------------------------------------------------------------------------------------------------------------
perso;  hmv1;           seaIce;   mon;          SImon;      time-mean;          XY-na;            ANY;                    ANY
perso;  hmv2;           atmos;    day;          day;        time-mean;          XY-na;            Coupled;                HighResMIP
perso;  hmv3;           ocean;    mon;          Omon;       time-point;         XY-na;            ANY;                    DCPP
perso;  hmv4;           atmos;    6hr;          6hrPlevpt;  time-point;         XY-na;            Forced-Atmos-Land;      HighResMIP
perso;  hmv5;           landIce;  mon;          LImon;      time-mean;          XY-na;            ANY;                    ANY
perso;  hmv6;           ocean;    day;          day;        time-mean;          XY-na;            DCPP-C13;               DCPP
cmor;   tos;            ocean;    day;          day;        time-mean;          XY-na;            Coupled;                HighResMIP
cmor;   zos;            ocean;    mon;          Omon;       time-point;         XY-na;            ANY;                    DCPP
cmor;   tas;            atmos;    6hr;          6hrPlevpt;  time-point;         XY-na;            ANY;                    HighResMIP
cmor;   mlotst;         ocean;    mon;          Omon;       time-mean;          XY-na;            DCPP-C13;               DCPP
cmor;   hfls;           atmos;    mon;          Amon;       time-mean;          XY-na;            ANY;                    ANY
perso;  hmv7;           ocean;    mon;          Omon;       time-mean;          XY-na;            ANY;                    HighResMIP
cmor;   sic;            seaIce;   day;          day;        time-mean;          XY-na;            ANY;                    HighResMIP
cmor;   sithick;        seaIce;   day;          SIday;      time-mean;          XY-na;            Coupled;                HighResMIP
cmor;   siconc;         seaIce;   day;          SIday;      time-mean;          XY-na;            Coupled;                HighResMIP
cmor;   omldamax;       ocean;    day;          day;        time-mean;          XY-na;            Coupled;                HighResMIP
perso;  sst;            ocean;    3hr;          3hr;        time-mean;          XY-na;             ANY;                   HighResMIP
```

The expected format and content of this file is detailed in *dr2xml/doc/listof_home_vars.help*.

The list of home variables is an 8-column file: `TYPE`, `VARNAME`, `REALM`, `FREQUENCY`, `TABLE`, `TEMPORAL_SHP`, `SPATIAL_SHP`, `EXPNAME`, `MIP`.

`'TYPE=cmor'` means that it is a CMOR variable. In this case, ***VARNAME must be a CMOR Variable name and all other parameters (REALM, FREQUENCY, TABLE, TEMPORAL_SHP, SPATIAL_SHP) much match the ones of the targeted CMOR variable.*** Requesting for TEMPORAL_SHAPE and SPATIAL_SHAPE can be seen as excessive, but this duly justify by the presence of ambiguities in the DR definition of variables (discussed at the end of section 4).

`'TYPE=perso'` means that the variable does not exist in CMOR/CF world. ***In this case, VARNAME must differ from existing CMOR Variable names.*** TABLE, TEMPORAL_SHP and SPATIAL_SHP can be anything, they are not taken into account (but must be provided, even if `'dummy'` – keep in mind that TABLE is part of the DRS). A next version of dr2xml will offer the possibility to read a *"home table"* [-not yet implemented-]; it has been considered as non-urgent functionality, since `'perso'` variables are -a priori- not candidate to ESG publication.

`'MIP'` can be set to one of CMIP6 MIPs or to `'ANY'` (which means: output this home variable whatever the MIP considered).

`'EXPNAME'` can be set to one of CMIP6 experiment name or to `'ANY'` (which means: current home variable will be output for all experiments in the given MIP(s)).

4. How to write a dr2xml ping file

A dr2xml ***ping file*** describes, for every CMIP6 requested variable name, which is the corresponding model field in XIOS namespace (i.e. as described by some ***'field definition'*** specific to the model). It is used for interfacing the field reference generated by dr2xml to model "native" field definitions. Its syntax is the one of an XIOS ***'field definition'*** item.

Example: CMIP6 request includes 'tos', the sea surface temperature, while Nemo sends field 'sst' to XIOS. Here is an example of a ping file making the relation between 'sst' and 'tos' and ensuring a transformation of units from Celsius to Kelvin:

```
<field_definition>
<field id="CMIP_tos"    field_ref="sst"> sst + 273.15 <field/>
</field_definition>
```

where 'CMIP_' is the prefix that you have defined in ***lab_and_model_settings***. In the case of an empty string prefix, you have to take care that this does not generate name collision with some XIOS field identifier already used by the model.

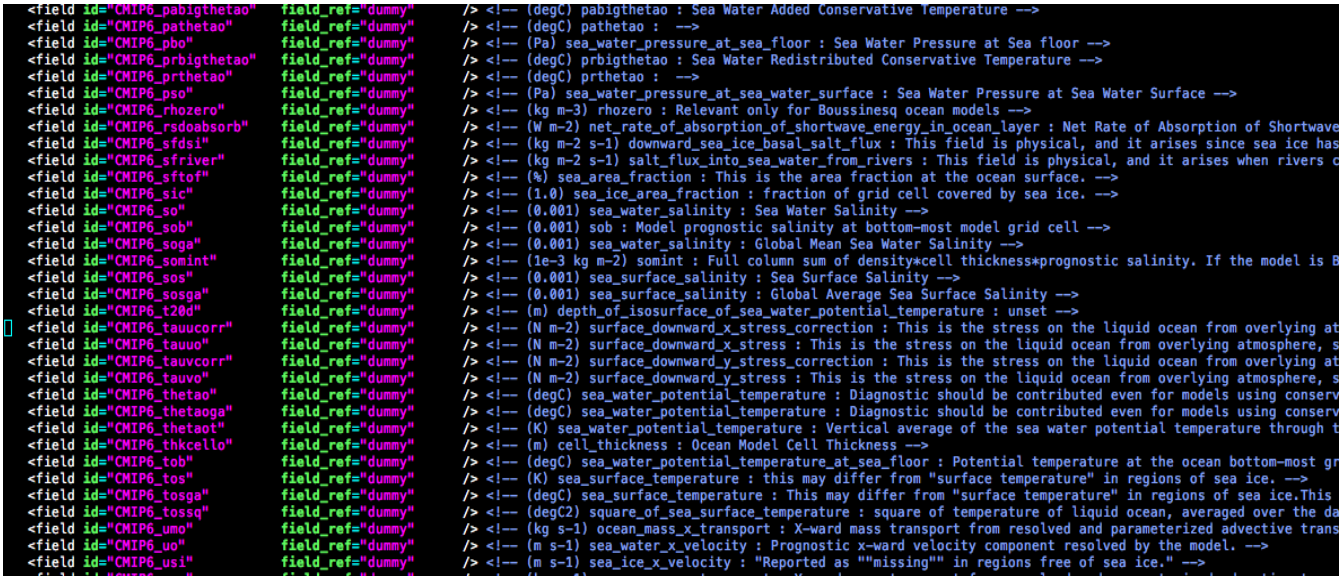
"Home" variables have not to be defined in ping files, but only in the model *native field_def* (without any prefix).

The ping file can also be used to tell dr2xml that some DR-requested variables are not to be produced, either because the model cannot produce it or because the lab does not want to. Any field definition entry that has a *'field ref'* beginning with dummy will not be included in output files by dr2xml (this feature will actually be implemented in next version of dr2xml)

To help you writing the ping file, a skeleton per realm is provided in directory output_samples/ where the exhaustive list of all variables for all MIPs at all priority level is provided as XIOS *'field ids'*.

Since it is very likely your lab is not involved in all MIPs, neither is concerned with all tiers and all variable priority levels, you may wish to regenerate these ping file templates to reduce their length and avoid to have to keep numerous dummy entries. For that purpose, create_ping_files.ipynb (available in dr2pub/ repository) python notebook drives you to create ping files templates with some user control (including accounting for a list of excluded variables). If not familiar with notebooks, you can alternatively use the equivalent classical Python script create_ping_files.py (available in dr2pub/doc/ repository).

Next, what you have to do is to identify, for each prefixed MIP variable (in the *'field id'* namespace) the associated model variable (in the *'field ref' namespace*) in replacement of 'dummy' when existing; see sample of the ping file template for ocean realm below:



On the basis of this example, you have to take care:

- that the ***field_ref*** (`'sst'`) is the name of a field already known to XIOS, and actually sent by the model (either un-conditionally, or when the model uses '`xios_field_is_active`' to know if it should send it)
- that you use in ***field id***, attached to the prefix, only the so-called MIP variable names (hereabove: `'tos'`) listed at http://clipc-services.ceda.ac.uk/dreq/index/var.html ; in some cases, you may add a suffix (see below)
- for the case of variables required on ocean transect, or as ocean zonal means or ocean sections, to provide as *'field ref'* the model variable with the relevant shape
- in the case of variables that are defined both at half and full model levels, that you use suffix '`_half`' behind the (prefix+CMIP variable name) for defining the values at half levels;

What you do **not** have to manage is:

- describe some spatial operations that are explicitly described by the so-called CMOR variables, such as e.g. zonal means from lat-lon grids, or extracting profiles or values at given locations (as e.g. '`rlu`' requested in table *cfSites* as profiles on sampled locations and on atmospheric levels, or as '`ta`' requested on 3 pressure levels in table *em1hr*) [-not yet implemented-];
- describe the vertical interpolations from atmospheric model levels to pressure of height levels; this applies only to dimensions sets: *'alt16'*, *'alt40'* and *'plev*'* [-not yet available-];
- describe time operation (averages, min, max, climatologies) nor variables such as `tasmin` or `tasmax`, these are automatically handled in generated *field_def*;

The ping file offers some facilities:

1. when the units of the native model variable does not match the expected standard, you can code, in the ping file, the units conversion as an XIOS arithmetic operation (cf. conversion from '`degC`' to '`K`' in the example above);

2. you may derive a requested variable by some arithmetics on XIOS-known fields, as e.g. in (where uppercase ids are native model *'field ids'*):

```
<field id="CMIP_mrsos"  field_ref="WG1_ISBA" > WG1_ISBA + WGI1_ISBA
</field>
```

3. you may decide that your model is more cost-effective than XIOS for computing some vertical interpolation; in that case, you may indicate that by a ping-file entry having as identifier e.g. `CV_ta_plev19`, i.e. the concatenation of your prefix, the CMIP varname and the *dimension label* (as stated in the Data Request at http://clipc-services.ceda.ac.uk/dreq/index/grids.html); this works for interpolation to pressure and height levels, [-not yet implemented-]

4. The same applies for zonal means, with suffix `'_zm'`, which should stand after any vertical interpolation suffix (e.g. `CV_tas_zm`, or `CV_ta_plev19_zm`) [-not yet implemented-]

In addition, dr2xml addresses a number of shortcomings of CMIP6 Data Request beta.45:

- in some cases, there is not enough information in the DR to derive a variable from another one, such as for `ta850`, the temperature at pressure level 850 hPa; the ping_file templates include some *field_defs* for that [-not yet implemented-];

- ambiguous MIP variables names: 64 MIP variables names are ambiguous in the sense that the set of corresponding CMOR Variables are not homogeneous regarding the area part of the '*cell_method*'. In that case, dr2xml will suffix the MIP variable name with a shortcut for area type, as derived by the code below; this occurs both in *file_def* files and *ping* files. So, you may have to fill in some consecutive ping file lines with the same content if you think the actual geophysical field is the same.

If cell_method includes:	**Automatic suffix is:**
where floating_ice_shelf	`_fisf`
where grounded_ice_shelf	`_gisf`
where snow over sea_ice area	`_sosi`
where ice_free_sea over sea	`_ifs`
where land	`_land`
where sea_ice	`_si`
where sea	`_sea`
where snow	`_snow`
"where cloud	`_cloud`
where landuse	`_lu`
where ice_shelf	`_isf`

The list of such MIP varnames is: `nep tnpeo hfss lai albisccp mrlsl hfgeoubed treeFrac mrsos sisnthick cVeg topg fbddtdife parasolRefl rlds lwsnl snc snm snw hfgeou o2sat fddtdisi rlus cWood prra agesno ts cMisc grassFrac prsn fbddtdic fbddtdin fbddtdip cSoil sbl orog cLitter prveg tpf fLuc fbddtalk fddtdife fddtalk pctisccp mrros lithk sootsn mrro tas tsn tran rsds hfdsn pflw fddtdic fddtdin fddtdip fbddtdisi rsus cProduct sftgif hfls dms`

Example: `'hfss'`, the *Surface Upward Sensible Heat Flux*, is ambiguous (the variable related to the whole atmospheric mesh appears in table *Amon*, but the Ice Sheet part only in table *LImongre*), while `'hfssIs'` (also Ice Sheet part in table *LImon*) is not ambiguous:

```
<field id="hfss_landIce"      field_ref="H_ISBA_P3" />
<field id="hfss"  field_ref="H" />
<field id="hfssIs"      field_ref="H_ISBA_P3" />
```

- in table Omon, there are some references to 'zfull' and 'zhalf' instead of 'zfullo' and 'zhalfo'; [-no special processing is done by dr2xml for that yet-].

Annexe: Some Data Request vocabulary

In the Data Request world, there are different collections used to describe and refer to variables. It is useful to have their definition in mind.

MIP Variable:

A ***MIP Variable*** ('MIPvar') defines a variable name and refers to the corresponding CF-standard name when that standard exists. There is no notion of frequency nor spatial or temporal dimensions for a MIP Variable.

A MIP variable is defined by the following attributes (for the exhaustive list of attributes see for example, http://clipcservices.ceda.ac.uk/dreq/u/0e5d376315a376cd2b1e37f440fe43d3.html):

- A label (e.g.: *label='tos'*);
- A title (e.g.: *title='Sea Surface Temperature'*);
- (possibly) A CF standard name (*e.g: standarname='SeaSurfaceTemperature'*);
- Units (e.g.*: units='K'*).

Note that several MIP variables can refer the same CF standard name, e.g. the following MIP variables all refer to the CF standard name *'SurfaceAlbedo'*:

- al: Albedo [%]
- alb: Surface Albedo
- albsrfc: surface albedo
- fal: Forecast Albedo
- lialb: Land ice or snow albedo
- lialbIs: Ice Sheet Ice or Snow Albedo
- surf-albedo: surface_albedo

CMOR Variable

A ***CMOR Variable*** ('CMORvar') is a declination of a MIP variable, associated to a realm and to a MIP table with a particular output frequency and spatial/temporal structure.

A CMOR Variable is defined by the following attributes (for the exhaustive list of attributes see for example, http://clipc-services.ceda.ac.uk/dreq/u/baa52de0-e5dd-11e5-8482-ac72891c3257.html)

- a variable identifier (e.g: *vid='0e5d376315a376cd2b1e37f440fe43d3'*) that links to the base MIP Variable;
- a label (e.g.: `label='tos'`);
- a title (eg: `title='Sea Surface Temperature'`);
- a sampling frequency (e.g.: `frequency='mon'`);
- a modelling realm (e.g.: `realm='atmos'`);
- a mipTable (e.g.: `table='Omon'`);
- a structure identifier (e.g.: `stid='f7ddef0c-562c-11e6-a2a4-ac72891c3257'`) that gather both temporal and spatial shapes information

- a default priority (`DefaultPriority =1`)

Note that the label of the CMOR Variable may differ from the label associated MIP Variable (e.g.: `'hus10'` and `'hus27'` – specific humidity on different sets of pressure levels- are CMOR Variables referring to the same MIP Variable `'hus'`)

Acknowledgements

The authors gratefully acknowledge Martin Juckes for the development of the CMIP6 Data Request package without which model configuration for CMIP6 would be much more tricky.