Requirements for CMIP5 Output

Karl E. Taylor[1] and Charles Doutriaux

Program for Climate Model Diagnosis and Intercomparison (PCMDI)

19 March 2010

Click *here* to obtain a pdf version of this document.

Recent changes to this document are summarized in the last section below.

Overview

Past experience with model intercomparison projects, highlighted by the third phase of the Coupled Model Intercomparison Project (CMIP3), has demonstrated the exceptional value gained by archiving multi-model output in a structured and uniform way. The user community now expects to be able to extract data both efficiently and in a uniform way across all models. The effort to write the data in a uniform structure and format falls on the modeling centers contributing the data. Building on CMIP3, we are now preparing for the fifth phase of CMIP (CMIP5). In CMIP5 a more comprehensive suite of experiments is planned (see Taylor et al., 2009) and a more extensive list of model output is requested (see CMIP5 Requested Output). Here we provide the specifications for writing CMIP5 model output. It should be noted that these requirements also extend to the output from intercomparison experiments closely aligned with (or incorporated as part of) CMIP5, including AMIP, CFMIP, C4MIP, PMIP, and TAMIP.

A software library, CMOR2 (pronounced "see more two") has been written to facilitate writing model output that conforms to these requirements. This library is written in the C programming language, but can be accessed through interfaces from Fortran or Python programming lanaguages. Documentation for this library explains how it can substantially reduce the burden placed on the modeling centers preparing CMIP5 model output. The library accesses the information contained in the excel spreadsheets that define the characteristics of the requested model output (after they have been reformatted into CMOR2-readable tables), so a group choosing to write its data through CMOR2 needs to supply only information specific to its own model; most of the metadata required by CMIP5 will be automatically provided by CMOR2. For those groups choosing not to use CMOR2, the following requirements must nevertheless be strictly adhered to. Although we have attempted to make the following specifications complete, the safest

[1] Send questions/comments/suggestions to taylor13@llnl.gov.

way to ensure that model output conforms to CMIIP5 requirements is to process it through CMOR2. In this document some text is red, which is an indication that the user must be especially careful to adhere to the specifications, since it will be very difficult for anyone else to determine whether or not the information is correct. Compliance with the rest of the requirements is pretty much guaranteed if the data is written with CMOR2. CMOR2 is distributed with a Python-based checker (check_CMOR_compliant.py). Output that has not been written through CMOR2, but is thought to adhere to the requirements, may be passed through the CMOR2 checker to catch some errors.

The requirements for CMIP5 are similar to those required in CMIP3, but there are a few major changes:

- Model output may now be contributed on any native grid, even one that is not a Cartesian latitude-longitude grid.
- Filenames and directory structures are now mandated according to a defined template.
- A number of additional global attributes are now required.
- A few new variable attributes are now required (when appropriate).

The requirements for data contributed to the archive are listed in five sections below, the first specifying the general structure and format of the data, the second the directory structure and names of files and directories, the third the required and recommended "global attributes", the fourth the metadata required for describing the coordinates, and last the constraints imposed on the variables themselves.

Data format, data structure, and file composition requirements:

- Data must be written in the netCDF-3 format and conform to the CF metadata standards. The output must be readable through the netCDF-3 API (application program interface) and conform to the netCDF "classic" data model. This means that if the data is written using the netCDF-4 API, the mode must be set to NC_CLASSIC_MODEL.
- Each file must contain only a single output field from a single simulation (i.e., a single run). Each file will also include coordinate variables, attributes and other metadata as specified below. If the field is a function of time, more than one time sample (but not necessarily all time samples) may be included in a single file. Data representing a long time-series, typical of many coupled model simulations, will usually be split into several files, which should neither be too large (to be unwieldy) nor too small (as to create vexing I/O performance issues). Monthly data, for example, might be divided into multi-decade chunks. It is recommended that the same size chunks should be used for all variables found in the same table of the CMIP5 Requested Output.
- Some atmospheric fields that are functions of the vertical coordinate must be interpolated to standard pressure levels (as specified in the CMIP5 Requested Output list of variables). Other fields (e.g., the 3-d cloud fraction) will reside on

the original model levels. There are different metadata and attribute requirements specified below for these two types of "3-d" fields.

- Oceanic fields that are a function of the vertical coordinate should usually be reported on the native grid.

Structure and names of directories and names of files

The IPCC database will comprise output from many different models, dozens of experiments, and perhaps several ensemble members, which have been sampled (or averaged) in a number of different ways (e.g., monthly, daily, 3-hourly). The directory structure for all of the output is specified in the "CMIP5 and AR5 Data Reference Syntax (DRS)" document. The names of the directories must be drawn from the "controlled vocabularly" specified in the same document. Finally, the filenames themselves must strictly follow the template given in the DRS document.

The directory structure will be as follows (see the DRS document for further explanation):

<activity>/<product>/<institute>/<model>/<experiment>/<frequency>/<model ing realm>/<variable name>/<ensemble member>/

Here are two examples:

/CMIP5/output/UKMO/HADCM3/decadal1990/day/atmos/tas/r3i2p1/
/CMIP5/output/UKMO/HADCM3/rcp45/mon/ocean/uo/r1i1p1/

Note that <model> should be identical to model_id, one of the global attributes described in a subsequent section, except that the following characters, if they appear in model_id should be replaced by a hyphen (i.e., by '-'): _ ( ) . ; , [ ] : / * ? < > " ' { } & and/or a "space".

The filenames will follow the template that is described more fully in the DRS document:

filename = *<variable name>*_*<variable table>*_*<model>*_*<experiment>*_*<ensemble member>*[_*<temporal subset>*].nc

Note that the < *temporal subset*> is omitted for variables that are time-independent. For these "fixed" fields the ensemble member should invariably be set to r0i0p0, denoting that this field is valid for all "r", "i", and "p".

Also note that together, *<variable name>* and *<variable table>* constitute the "variable identifier", which uniquely defines the variable.

Here is an example:

tas_Amon_HADCM3_ historical_r1i1p1_185001-200512.nc

Requirements for global attributes:

- Required global attributes:
  - → branch_time = time in parent experiment when this simulation started (in the units of the parent experiment). [See parent_experiment_id for more information about the "parent".] For example, if this run were spun off from a control run at "2000 days since day 500", then the user would store branch_time=2000 (not 1500). The branch_time should be set to 0.0 if not applicable.
  - → contact = name and contact information (e.g., email, address, phone number) of person who should be contacted for more information about the data.
  - → Conventions = 'CF-1.4'
  - → creation_date = a string representation of the date when the file was created in the format: "*YYYY-MM-DD*-T*HH*:*MM*:*SSZ*" with replacement of all but "T" and "Z" by the obvious date or time indicator (e.g., "2010-03-T05:56:23Z".
  - → experiment = a string providing a title for the experiment, as specified in the controlled vocabulary found in the table column labeled "Experiment Name" in the Appendix of the "CMIP5 and AR5 Data Reference Syntax (DRS)" document.
  - → experiment_id = a short string identifying the experiment, as specified in the controlled vocabulary found in the table column labeled "Short Name of Experiment" in the Appendix of the "CMIP5 and AR5 Data Reference Syntax (DRS)" document.
  - → forcing = a string containing a list of the "forcing" agents that could cause the climate to change in the experiment. For a control run with no variation in radiative forcing or for any other experiment in which there are no externally imposed changes in radiative forcing agents, set this to "N/A". Otherwise, the forcing should be expressed as a comma separated list of identifying strings that are part of the so-called DRS controlled vocabulary described in Appendix 1.2 of the "CMIP5 and AR5 Data Reference Syntax (DRS)" document. Within or following this machine-interpretable list may be text enclosed in parentheses providing further information.
  - → frequency = a string indicating the interval between individual time-samples in the atomic dataset. The following are the only options: "yr", "mon", "day", "6hr", "3hr", "subhr" (sampling frequency less than an hour), "monClim" (climatological monthly mean) or "fx" (fixed, i.e., time-independent). The sampling frequency is specified at the top of each spreadsheet (cell G1) in CMIP5 Requested Output. For a few tables some variables within the table are sampled differently, as indicated by an entry in column T of the spreadsheets.
  - → initialization_method = an integer (≥1) referring to the initialization method used. If only a single method was used to initialize the model, then this argument should be given the value 1 (which is also the default value). For fields appearing in table "fx" in the CMIP5 Requested Output, set

initialization_method=0 (violating the general rule that it should be a positive definite integer). See the CMIP5 and AR5 Data Reference Syntax (DRS)" document for guidance on assigning initialization_method.

→ institute_id = a short acronym describing "institution" (e.g., 'GFDL')

→ institution = character string identifying the institution that generated the data [e.g., 'GFDL (Geophysical Fluid Dynamics Laboratory, Princeton, NJ, USA']

→ model_id = a string containing an acronym that identifies the model used to generate the output. For CMIP5, the model_id should be officially approved by the CMIP Panel (through PCMDI). It should be as short as possible, so that it can be used, for example, in labeling curves on multi-model plots (e.g., as might appear in the Fifth Assessment Report of the IPCC). The acronym may include the acronym of the modeling center and the model name/version separated by a hyphen (e.g., "IPSL-CM4"), but it may be o.k. to omit the modeling center. Please note that you might in the future want to submit results from a successor to the present model, so if appropriate, you may want to indicate a model version, but please keep it simple e.g., CCSM4, not CCSM4.1.2. Full version information will appear in the "source" global attribute described below. The model_id, possibly modified as necessary to eliminate characters not permitted by the CMIP5 and AR5 Data Reference Syntax (DRS), will be used to construct directory and filenames. For further information, see the earlier section describing the directory and filenames.

→ modeling_realm = a string that indicates the high level modeling component which is particularly relevant. For CMIP5, permitted values are: "atmos", "ocean", "land", "landIce", "seaIce", "aerosol" "atmosChem", or "ocnBgchem" (ocean biogeochemical). Note that sometimes a variable will be equally (or almost equally relevant) to two or more "realms", in which case a primary "realm" is assigned, but cross-referenced or aliased to the other relevant "realms". The modeling_realm(s) is (are) specified in column S of the spreadsheets found in CMIP5 Requested Output.

→ parent_experiment_id = experiment_id indicating which experiment this simulation branched from. This should match the experiment_id of the parent unless the "parent" is irrelevant, in which case this should be set to "N/A". The experiment_id's can be found in the table column labeled "Short Name of Experiment" in the Appendix of the "CMIP5 and AR5 Data Reference Syntax (DRS)" document.

→ physics_version = an integer (≥1) referring to the physics version used by the model If there is only one physics version of the model, then this argument should be given the value 1 (which is also the default value). For fields appearing in table "fx" in the CMIP5 Requested Output, set physics_version=0 (violating the general rule that it should be a positive definite integer).

→ product = "output", which indicates that the data you are writing is model output.

→ project_id = "CMIP5"

→ realization = an integer (≥1) distinguishing among members of an ensemble of simulations (e.g., 1, 2, 3, etc.). If only a single simulation was performed, then it is recommended that realization=1. For fields appearing in table "fx" in the CMIP5 Requested Output, set realization=0 (violating the general rule that it should be a positive definite integer). Note that if two different simulations were started from the same initial conditions, the same realization number should be used for both simulations. For example if a historical run with "natural forcing" only and another historical run that includes anthropogenic forcing were initiated from the same point in a control run, both should be assigned the same realization. Also, each so-called RCP (future scenario) simulation should be assigned the same realization integer as the historical run from which it was initiated. This will allow users to easily splice together the appropriate historical and future runs. A similar convention should be followed, when appropriate, with other simulations (e.g., the decadal simulations).

→ source = character string fully identifying the model and version used to generate the output. The first portion of the string should be a copy of the global attribute "model_id". Additionally, this attribute must include the year (i.e., model vintage) when this model version was first used in a scientific application. Finally, it should include information concerning the component models. The following template should be followed in constructing this string: '*<model_id> <year>* atmosphere: *<model_name>* (*<technical_name>, <resolution_and_levels>*); ocean: *<model_name>* (*<technical_name>, <resolution_and_levels>*); sea ice: *<model_name>* (*<technical_name>)*; land: *<model_name>* (*<technical_name>)*" For some models, it may not make much sense to include all these components, and nothing following "<year>" is absolutely mandatory. As an example, "source" might contain the string: 'CCSM2 2002 atmosphere: CAM2 (cam2_0_brnchT_itea_2, T42L26); ocean: POP (pop2_0_ver_1.4.3, 2x3L15); sea ice: CSIM4; land: CLM2.0'. For some models it might be appropriate to list only a single component, in which case the descriptor (e.g., 'atmosphere') may be omitted along with the other model components (e.g., for an aquaplanet experiment: 'CAM2 2002 (cam2_0_brnchT_itea_2, T42L26)'). Additional explanatory information may follow the required information.

→ table_id = should be assigned a character string identifying the CMIP5 Requested Output table where this variable appears. This string should be of the form "Table <table name as it appears in cell F1 of the spreadsheets in CMIP5 Requested Output>" (e.g., "Table Amon").

→ tracking_id = a string that is almost surely unique to this file and must be generated using the OSSP utility which supports a number of different DCE 1.1 variant UUID options. For CMIP5 version 4 (random number based) is required. Download the software from http://www.ossp.org/pkg/lib/uuid/. The tracking_id might look something like: `02d9e6d5-9467-382e-8f9b-9300a64ac3cd`.

- Optional global attributes
  - → comment = A character string containing additional information about the data or methods used to produce it. The user might, for example, want to provide a description of how the initial conditions for a simulation were specified and how the model was spun-up (including the length of the spin-up period).
  - → history = A character string containing an audit trail for modifications to the original data. Each modification is typically preceded by a "timestamp". The "history" attribute provided here will be a global one that should not depend on which variable is contained in the file. A variable-specific "history" can also be included as an attribute attached to the output variable.
  - → 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.
  - → title = "<*institute_name*> model output prepared for CMIP5 <*experiment*>" where <*institute_name*> should be replaced by the contributing institution's acronym or name (e.g., "GFDL", "CCCma", "IPSL", etc.) and <*experiment*> should be replaced by one of the experiment id's found in the table column labeled "Experiment Name" in the Appendix of the "CMIP5 and AR5 Data Reference Syntax (DRS)" document. A sample title is: 'IPSL-CM5 model output prepared for CMIP5 historical"

Requirements for coordinate variables:

- All coordinate variables must be written as double precision floating point numbers (netCDF type NC_DOUBLE).
- The names for all coordinate variables are specified in the "output dimension name" column (column C) of the "dims" table in CMIP5 Requested Output.
- The units for all coordinate variables are specified in the "units" column (column H) of the "dims" table in CMIP5 Requested Output.
- For all time coordinates the units must be "days since <basetime>", where <basetime> must be specified by the user, typically in the form year-month-day (e.g., "days since 1800-1-1"). Follow these rules:
  - → The same 'base time' should apply to all time samples in a single simulation (i.e., when creating a series of files containing model output for a single run, don't change the units or 'base time' from one file to the next; in the above example 1800-1-1 is the 'base time').
  - → For simulations meant to represent a particular historical period, set the 'base time' to the time at the beginning of the simulation. A historical run initialized with forcing for year 1850 would, for example, have units of "days since 1850-1-1". The 'base time' used in the control run is irrelevant as is the time in the control run when the historical run was initialized.

→ For the future scenario runs, retain the same 'base time' as used in the historical run from which it was initiated.

→ For simulations not tied to a particular date, the 'base time' is arbitrary, but it should correspond to the beginning of the run. Thus, in these simulations, 'base time' is whatever time one decides to assign to the beginning of the run.

- For time-mean data, a time coordinate value must be defined as the mid-point of the interval over which the average is computed. (More generally, this same rule applies whenever time-bounds are included: the time coordinate value should be the mean of the two time bounds.)
- The "bounds" are required for certain coordinate variables as indicated by the "bounds?" column (column K) of the dims table in CMIP5 Requested Output. The bounds variable should be double precision.
- For grids other than Cartesian latitude-longitude grids, the output field can be a function of 1, 2, 3, or more horizontal "dimensions", depending on the natural logical structure of the model's grid. For these grids the dimensions will usually be simple index dimensions with the dimension names drawn from the list: 'i', 'j', 'k', 'l', 'm'. In some cases when there are two horizontal dimensions, these dimensions might represent the earth according to a well-know map projection involving actual spatial dimensions, in which case the names for these dimensions should be 'x' and 'y'. In both of these cases a "coordinates" attribute is required and should be attached to the output variable. This attribute will include the string "lat lon", indicating that variables lat and lon contain the latitude and longitude coordinates. Those variables must have the standard names "latitude" and "longitude" respectively and must have the units "degrees_north" and "degrees_east", respectively. In addition, these coordinate variables must include "bounds" attributes pointing to 'lat_vertices" and "lon_vertices", respectively, which should be stored arrays named lat_vertices(lat,vertices) and lon_vertices(lon,vertices), respectively. The length of the "vertices" dimension should be just long enough to accommodate the maximum number of vertices needed to describe any cell in the grid. If a cell does not have the maximum number of vertices, the remaining values should be set to 1.e20.
- In addition for grids other than Cartesian latitude-longitude grid, a file recording all the grid configuration information called for by "gridspec" must be created.
- For a coordinate representing model atmospheric level, the user must include in the file all the information needed to positively and uniquely indicate the vertical location of the data. Usually in this case the "formula_terms" attribute must be defined, and additional variables or parameters will need to be stored in the file (e.g., surface pressure and pressure at the top of the model for an atmospheric-sigma coordinate system). The names of the variables and parameters needed to describe each of the known vertical coordinate systems are provided in Appendix 1 below. See example 5 below, and section 4.3.2 of the CF-conventions and Appendix D of the CF-conventions.
- Variables that are stored as a function of "basin", "xline", or "type" (which are simple index dimensions) will require auxiliary coordinate variables named "region", "passage", and "typeDescription", respectively, which are pointed to by

a "coordinates" attribute attached to the variable. These will be netCDF type NC_CHAR and will be dimensioned region(basin, strlen), passage(xline, strlen), and typeDescription(type, strlen), respectively, where strlen is the maximum string length. For each of these auxiliary coordinate variables, a long_name attribute and in the case of "region", the standard_name attribute should be stored as specified in the dims table of CMIP5 Requested Output. The values and order of the labels stored in these auxiliary variables should be consistent with the lists found in column S of the dims table in CMIP5 Requested Output. The values and order of the labels stored in "typeDescription" are not standardized.

- Required attributes for coordinate variables:
  - → axis = "X", "Y", "Z", or "T" as and if appropriate (see section 4 of the CF-conventions). For a dimension that represents an axis other than a spatial or temporal dimension, this attribute should be omitted. Also omit this attribute for grids that are not Cartesian latitude-longitude grids unless the grid is logically represented as two-dimensional with these two dimensions being actual coordinates (not simple index dimensions), in which case the "X" and "Y" axis attributes should be assigned as one would like to see a plotted figure of the field in that two-dimensional space.
  - → bounds = a character string containing the name of the variable where the cell bounds are stored. Whether or not bounds are required for a given coordinate is specified in the "bounds?" column (column K) of the dims table in CMIP5 Requested Output. Usually, the bounds name should be the same as the coordinate name, but with the suffix "_bnds" appended. For grids other than those that are Cartesian in latitude and longitude, however, the bounds name should be the same as the coordinate name, but with the suffix "_vertices" appended. For Cartesian latitude-longitude grids, the "bounds" variable should be a function of the coordinate and have a second dimension (varying most quickly) of length 2 in the normal case. For example for a lon coordinate the bounds would be lon_bnds(lon,2).[2] For other grids an example is lon_vertices(lon,vertices).
  - → calendar = (but for time coordinates only), one of the options described in section 4.4.1 of the CF-conventions. Note that if a model has a gregorian calendar and the base time is defined to be prior to 1582, then the user should specify 'proleptic_gregorian', rather than 'gregorian' for the calendar attribute. If none of the calendars defined by the CF-conventions applies, then the user is free to describe the calendar in the calendar attribute, or it is permissible to set "calendar=non_standard". For such unusual calendars include, as needed, the month_lengths, leap_year, and leap_month attributes.
  - → formula_terms = (but for dimensionless vertical coordinates only) a character string, as described in section 4.3.2 of the CF-conventions and as illustrated in example 5 below. Further information can be found in Appendix D of the CF-conventions.

[2] Note that the ordering of array indices specified here assumes the C language convention (opposite the FORTRAN convention) in which the last coordinate varies most rapidly.

→ positive = (but for vertical coordinates only) “down” or “up”, as appropriate. See section 4.3 of the CF-conventions.

→ standard_name = a character string containing the standard name specified in the dims table of CMIP5 Requested Output.

→ units = a character string containing the units specified in the dims table of CMIP5 Requested Output. Note that in the case of a dimensionless vertical coordinate, this attribute may be omitted or set to “1”.

Requirements for output variables:

- All output fields must be written consistent with the data type (e.g., float or integer) specified in the “type” column (column P) of the CMIP5 Requested Output tables. Almost all variables will be written as single precision floating point numbers (netCDF type NC_FLOAT).
- The variable names (in the netCDF file) must be the names specified in the “output variable name” column (column F) of the CMIP5 Requested Output tables (e.g., surface air temperature data will be stored in a variable named "tas").
- The units required for the output fields are given in the CMIP5 Requested Output tables.
- The positive direction of vertical fluxes must be consistent with that specified in the “unformatted units” column (column I) of the CMIP5 Requested Output table and is also indicated by the standard_name (e.g., "surface_upward_latent_heat_flux” and "surface_downward_eastward_stress").
- For each variable, the ordering of the required dimensions is given in the “CMOR dimensions” column (column Q) of the CMIP5 Requested Output tables, but note that the *names* of these dimension, as they will appear in the netCDF files, are given in the “output dimension name” column (column C) of the “dims” table of CMIP5 Requested Output. Note that the ordering given in the tables assumes the C language convention (opposite the FORTRAN convention) in which the last coordinate varies most rapidly. Note that for grids that cannot be expressed as the intersections of points along constant longitude lines and constant latitude lines (i.e., not Cartesian latitude-longitude), the longitude and latitude dimensions should be replaced with a 1-, 2-, … n-dimensional set of axes, as appropriate, depending on the logical structure of the non-Cartesian grid. Finally note that some of the dimensions appearing in the “CMOR dimensions” column are so-called scalar dimensions and the variable should *not* include this dimension explicitly. Rather, as described below, a “coordinates” attribute should be included and should list this dimension as specified in the CF conventions.
- For Cartesian latitude-longitude grids, the data must be ordered with longitude increasing from west to east, starting with the first grid point greater than or equal to 0 degrees east. All coordinate locations must be unique (e.g., don't include both 0 and 360 degrees east).
- For Cartesian latitude-longitude grids, the data must be ordered with latitude increasing from south to north.

- If there is a vertical coordinate, data must be stored starting with the grid point nearest the surface. Thus, for the atmosphere the lowest layer is stored first, whereas for the ocean and land surface the top layer is stored first.
- If there is a time dimension, data must be stored with time increasing.
- If there is a "basin" coordinate or an "xline" coordinate used to distinguish ocean basins, the data should be stored in the order consistent with the specifications for these coordinates given in the previous section.
- All "missing data"[3] must be assigned the single precision floating point value of 1.e20
- Required attributes for variables:
  - → associated_files = a string listing the base URL for CMIP5 and the location of the model's gridspec file, followed, as appropriate, by the name of the file containing the grid cell areas and/or grid cell volumes. For CMIP5 this string is: "baseURL:http://www-pcmdi.llnl.gov/dataLocation/ gridspecFile:<gridspec file name> [cellAreaFile:<cell area file name>] [cellVolumeFile:<cell volume file name>]", where cell area and cell volume are only sometimes required. Here is an example: associated_files= "baseURL:http://www-pcmdi.llnl.gov/ gridspecFile: gridspec_fx_IPSL-CM5_historical_r0i0p0.nc cellAreaFile:areacello_fx_IPSL-CM5_historical_r0i0p0.nc cellVolumeFile:volcello_fx_IPSL-CM5_historical_r0i0p0.nc" Note that the cell_measures column (column U) of the CMIP5 Requested Output tables indicates whether or not cellAreaFile and/or cellVolumeFile should be included as associated_files.
  - → cell_measures = the string that appears in the "cell measures" column (column U) of the CMIP5 Requested Output tables, or it is omitted if an entry in the table is blank.
  - → cell_methods = the string that appears in the "cell methods" column (column J) of the CMIP5 Requested Output tables, or it is omitted if an entry in the table is blank.
  - → coordinates = a character string that must be included for the following types of variables:
    - – When an output field has a coordinate that is single-valued (e.g., surface wind may have a vertical coordinate value of 10 meters), the "coordinates" attribute must be defined and must include the name of the scalar coordinate (as specified in column C of the "dims" table of CMIP5 Requested Output). The scalar value for the coordinate must be stored in the variable pointed to by that attribute. The dimensions appearing in the "CMOR dimensions" column of the CMIP5 Requested Output tables that are actually scalar dimensions that need special treatment are identified by the presence of an entry in the "value" column of the "dims" table (i.e., if a value is found in that column, then the dimension is a scalar

[3] As an example of "missing data", consider a globally gridded dataset of ocean surface salinity. The data representing land points on this dataset would be missing. Likewise, atmospheric data on portions of pressure surfaces that are underground are usually considered missing.

dimension). See section 5.7 of the CF-conventions for more information. Note that the attributes required to be attached to scalar dimensions are the same as the attributes required to be attached to a normal dimension.

- For fields that are a function of "xline", "basin", or "type" the "coordinates" attribute should be defined as "region", "passage", and "typeDescription", respectively. The coordinate variable found in the coordinates attribute must be defined in the file and be assigned the values listed in the "requested" column (column J) of the "dims" table found in the CMIP5 Requested Output. See example 4 below and section 6.1.1 of the CF-conventions for more information.
- For fields stored on grids other than a Cartesian latitude-longitude grid, the "coordinates" attribute should include the string "lon lat", and the longitude and latitude values for each grid cell should be stored in the variables "lon" and "lat", respectively, as called for by section 5.2 of the CF-conventions.

→ _FillValue and missing_value = 1.e20 (but required only for variable that might have "missing values"). This is the value to be assigned to portions of the variable field that are "missing". Note that defining both of these attributes and giving them the same value seems to be redundant, but some software that reads netCDF files expects the missing_value attribute, while other software expects the _FillValue attribute, so including both is required here.

→ flag_values and flag_meanings should be defined (at least for the "Region Selection Index), with the values taken from the "flag values" and "flag meanings" columns of the CMIP5 Requested Output tables (columns V and W).

→ long_name = a character string that might serve as a title for the variable on plots (e.g., "Surface Air Temperature"). It is strongly recommended that this string be copied from the "long name" column of the CMIP5 Requested Output tables.

→ standard_name = a character string containing a CF standard name that can be found in the "standard name" column (column G) of the CMIP5 Requested Output tables. Although usually required, for a few fields the standard_name is omitted from the table, and therefore it should be omitted as an attribute for these variables.

→ units = a character string containing the units given in the "unformatted units" column (column I) of the CMIP5 Requested Output tables, which must be in a form recognizable by UNIDATA's Udunits package version 2 (e.g., "W m-2".)

- Optional attributes for variables:

→ comment = a character string providing further information concerning the variable (e.g., if the variable is mrso (soil_moisture_content), the comment

might read "includes subsurface water, both frozen and liquid, but not surface water, snow or ice").

→ history = a character string containing an audit trail for modifications to the original data (e.g., indicate what calculations were performed in post-processing, such as interpolating to standard pressure levels or changing the units). Each modification is typically preceded by a "timestamp". Note that this history attribute is variable-specific, whereas the global history attribute defined above (see "Optional attributes" under "Requirements for global attributes") provides information concerning the model simulation itself or refers to processing procedures common to all variables.

→ original_name = the name of the variable as it is commonly known at the user's modeling institute. If the variable being written was computed in some simple way from two or more original fields (e.g., subtracting the upwelling and downwelling fluxes to get a net flux, or multiplying by a scalar or changing the sign), then it is recommended that this be indicated in the "original_name" (e.g., "irup – irdown", where "irup" and "irdown" are the names of the original fields that were subtracted). If more complicated processing is required, this attribute would likely be omitted and the information would more naturally be included in a "history" attribute for this variable, described next.

→ original_units = a character string indicating the original units of the data before it has been processed to be consistent with the units appearing in the CMIP5 Requested Output tables.

→ grid_mapping = the name of the variable containing the grid_mapping information describing the grid for fields stored on certain grids other than a Cartesian latitude-longitude grid, as called for in section 5.6 of the CF-conventions.

THINGS TO DO: Should probably add appendices describing more fully the grid_mapping info. and dimensionless vertical coordinate information. Also revise appendices to be consistent with the new CMIP5 tables.

Recent Changes to this Document:

8 March 2010: Initial document written, drawing from its predecessor: Requirements for IPCC Standard Output Contributed to the PCMDI Archive.

19 March 2010: Revised version created, including input from Charles Doutriaux.

This example should be updated:

Example 1: Surface latent heat flux
(a function of longitude, latitude, month)

```
netcdf hfls_A1 {
dimensions:
      lon = 4 ;
      lat = 3 ;
      time = UNLIMITED ; // (2 currently)
      bnds = 2 ;
variables:
      double lon(lon) ;
            lon:standard_name = "longitude" ;
            lon:long_name = "longitude" ;
            lon:units = "degrees_east" ;
            lon:axis = "X" ;
            lon:bounds = "lon_bnds" ;
      double lon_bnds(lon, bnds) ;
      double lat(lat) ;
            lat:standard_name = "latitude" ;
            lat:long_name = "latitude" ;
            lat:units = "degrees_north" ;
            lat:axis = "Y" ;
            lat:bounds = "lat_bnds" ;
      double lat_bnds(lat, bnds) ;
      double time(time) ;
            time:standard_name = "time" ;
            time:long_name = "time" ;
            time:units = "days since 2030-1-1" ;
            time:axis = "T" ;
            time:calendar = "360_day" ;
            time:bounds = "time_bnds" ;
      double time_bnds(time, bnds) ;
      float hfls(time, lat, lon) ;
            hfls:standard_name = "surface_upward_latent_heat_flux" ;
            hfls:long_name = "Surface Latent Heat Flux" ;
            hfls:units = "W m-2" ;
            hfls:cell_methods = "time: mean (interval: 20 minutes)" ;
            hfls:_FillValue = 1.e+20f ;
            hfls:missing_value = 1.e+20f ;
            hfls:original_name = "LATENT" ;
            hfls:history = " At   17:54:19 on 06/07/2004: CMOR altered
                the data in the following ways: replaced missing value
                flag (1.00000E+28) with standard missing value
                (1.00000E+20); multiplied by -1.00000E+00 to yield
                output units;" ;

// global attributes:
            :title = "GICC  model output prepared for IPCC Fourth
                Assessment 2xCO2 equilibrium experiment" ;
            :institution = "GICC (Generic International Climate Center,
                Geneva, Switzerland)" ;
            :source = "GICCM1 (2002): atmosphere:  GICAM3
                (gicam_0_brnchT_itea_2, T63L32); ocean: MOM
```

```
                (mom3_ver_3.5.2, 2x3L15); sea ice: GISIM4; land:
                GILSM2.5" ;
            :contact = "Rusty Koder (koder@middle_earth.net)" ;
            :project_id = "IPCC Fourth Assessment" ;
            :table_id = "Table A1 (7 April 2004)" ;
            :experiment_id = "2xCO2 equilibrium experiment" ;
            :realization = 1 ;
            :Conventions = "CF-1.0" ;
            :history = "Output from archive/giccm_03_std_2xCO2_2256.  At
                17:54:19 on 06/07/2004, CMOR rewrote data to comply with
                CF standards and IPCC Fourth Assessment requirements" ;
            :references = "Model described by Koder and Tolkien (J.
                Geophys. Res., 2001, 576-591).  Also see
                http://www.GICC.su/giccm/doc/index.html  2XCO2
                simulation described in Dorkey et al. (Clim. Dyn., 2003,
                323-357.)" ;
            :comment = "Equilibrium reached after 30-year spin-up after
                which data were output starting with nominal date of
                January 2030" ;
data:

 lon = 0, 90, 180, 270 ;

 lon_bnds =
  -45, 45,
  45, 135,
  135, 225,
  225, 315 ;

 lat = 10, 20, 30 ;

 lat_bnds =
  5, 15,
  15, 25,
  25, 35 ;

 time = 15, 45 ;

 time_bnds =
  0, 30,
  30, 60 ;

 hfls =
  19, 15, 11, 7,
  3, -1, -5, -9,
  -13, -17, -21, -25,
  18, 14, 10, 6,
  2, -2, -6, -10,
  -14, -18, -22, -26 ;
}
```

This example should be updated:

Example 2: Air temperature
(a function of longitude, latitude, pressure, month)

```
netcdf ta_A1 {
dimensions:
      lon = 4 ;
      lat = 3 ;
      plev = 5 ;
      time = UNLIMITED ; // (2 currently)
      bnds = 2 ;
variables:
      double lon(lon) ;
            lon:standard_name = "longitude" ;
            lon:long_name = "longitude" ;
            lon:units = "degrees_east" ;
            lon:axis = "X" ;
            lon:bounds = "lon_bnds" ;
      double lon_bnds(lon, bnds) ;
      double lat(lat) ;
            lat:standard_name = "latitude" ;
            lat:long_name = "latitude" ;
            lat:units = "degrees_north" ;
            lat:axis = "Y" ;
            lat:bounds = "lat_bnds" ;
      double lat_bnds(lat, bnds) ;
      double plev(plev) ;
            plev:standard_name = "air_pressure" ;
            plev:long_name = "pressure" ;
            plev:units = "Pa" ;
            plev:axis = "Z" ;
            plev:positive = "down" ;
      double time(time) ;
            time:standard_name = "time" ;
            time:long_name = "time" ;
            time:units = "days since 2030-1-1" ;
            time:axis = "T" ;
            time:calendar = "360_day" ;
            time:bounds = "time_bnds" ;
      double time_bnds(time, bnds) ;
      float ta(time, plev, lat, lon) ;
            ta:standard_name = "air_temperature" ;
            ta:long_name = "Temperature" ;
            ta:units = "K" ;
            ta:cell_methods = "time: mean (interval: 20 minutes)" ;
            ta:_FillValue = 1.e+20f ;
            ta:missing_value = 1.e+20f ;
            ta:original_name = "T" ;
            ta:history = " At   17:54:19 on 06/07/2004: CMOR altered the
                data in the following ways: replaced missing value flag
                (1.00000E+28) with standard missing value
                (1.00000E+20);" ;

// global attributes:
```

```
            :title = "GICC  model output prepared for IPCC Fourth
                Assessment 2xCO2 equilibrium experiment" ;
            :institution = "GICC (Generic International Climate Center,
                Geneva, Switzerland)" ;
            :source = "GICCM1 (2002): atmosphere:  GICAM3
                (gicam_0_brnchT_itea_2, T63L32); ocean: MOM
                (mom3_ver_3.5.2, 2x3L15); sea ice: GISIM4; land:
                GILSM2.5" ;
            :contact = "Rusty Koder (koder@middle_earth.net)" ;
            :project_id = "IPCC Fourth Assessment" ;
            :table_id = "Table A1 (7 April 2004)" ;
            :experiment_id = "2xCO2 equilibrium experiment" ;
            :realization = 1 ;
            :Conventions = "CF-1.0" ;
            :history = "Output from archive/giccm_03_std_2xCO2_2256.  At
                17:54:19 on 06/07/2004, CMOR rewrote data to comply with
                CF standards and IPCC Fourth Assessment requirements" ;
            :references = "Model described by Koder and Tolkien (J.
                Geophys. Res., 2001, 576-591).  Also see
                http://www.GICC.su/giccm/doc/index.html  2XCO2
                simulation described in Dorkey et al. (Clim. Dyn., 2003,
                323-357.)" ;
            :comment = "Equilibrium reached after 30-year spin-up after
                which data were output starting with nominal date of
                January 2030" ;
data:

 lon = 0, 90, 180, 270 ;

 lon_bnds =
  -45, 45,
  45, 135,
  135, 225,
  225, 315 ;

 lat = 10, 20, 30 ;

 lat_bnds =
  5, 15,
  15, 25,
  25, 35 ;

 plev = 10000, 20000, 30000, 40000, 50000 ;

 time = 15, 45 ;

 time_bnds =
  0, 30,
  30, 60 ;

 ta =
  150.5, 152.5, 154.5, 156.5,
  158.5, 160.5, 162.5, 164.5,
  166.5, 168.5, 170.5, 172.5,
  182.5, 184.5, 186.5, 188.5,
  190.5, 192.5, 194.5, 196.5,
  198.5, 200.5, 202.5, 204.5,
```

```
  214.5, 216.5, 218.5, 220.5,
  222.5, 224.5, 226.5, 228.5,
  230.5, 232.5, 234.5, 236.5,
  246.5, 248.5, 250.5, 252.5,
  254.5, 256.5, 258.5, 260.5,
  262.5, 264.5, 266.5, 268.5,
  278.5, 280.5, 282.5, 284.5,
  286.5, 288.5, 290.5, 292.5,
  294.5, 296.5, 298.5, 300.5,
  151, 153, 155, 157,
  159, 161, 163, 165,
  167, 169, 171, 173,
  183, 185, 187, 189,
  191, 193, 195, 197,
  199, 201, 203, 205,
  215, 217, 219, 221,
  223, 225, 227, 229,
  231, 233, 235, 237,
  247, 249, 251, 253,
  255, 257, 259, 261,
  263, 265, 267, 269,
  279, 281, 283, 285,
  287, 289, 291, 293,
  295, 297, 299, 301 ;
}
```

This example should be updated:

Example 3: Treatment of a scalar (i.e., singleton) dimension
Moisture in upper 0.1 m of soil column
(a function of longitude, latitude, month)

```
netcdf mrsos_A1 {
dimensions:
      lon = 4 ;
      lat = 3 ;
      time = UNLIMITED ; // (2 currently)
      bnds = 2 ;
variables:
      double lon(lon) ;
            lon:standard_name = "longitude" ;
            lon:long_name = "longitude" ;
            lon:units = "degrees_east" ;
            lon:axis = "X" ;
            lon:bounds = "lon_bnds" ;
      double lon_bnds(lon, bnds) ;
      double lat(lat) ;
            lat:standard_name = "latitude" ;
            lat:long_name = "latitude" ;
            lat:units = "degrees_north" ;
            lat:axis = "Y" ;
            lat:bounds = "lat_bnds" ;
      double lat_bnds(lat, bnds) ;
      double time(time) ;
            time:standard_name = "time" ;
            time:long_name = "time" ;
            time:units = "days since 2030-1-1" ;
            time:axis = "T" ;
            time:calendar = "360_day" ;
            time:bounds = "time_bnds" ;
      double time_bnds(time, bnds) ;
      double depth ;
            depth:standard_name = "depth" ;
            depth:long_name = "depth" ;
            depth:units = "m" ;
            depth:axis = "Z" ;
            depth:positive = "down" ;
            depth:bounds = "depth_bnds" ;
      double depth_bnds(bnds) ;
      float mrsos(time, lat, lon) ;
            mrsos:standard_name = "moisture_content_of_soil_layer" ;
            mrsos:long_name = "Moisture in Upper 0.1 m of Soil Column" ;
            mrsos:units = "kg m-2" ;
            mrsos:cell_methods = "time: mean (interval: 20 minutes)" ;
            mrsos:coordinates = "depth" ;
            mrsos:_FillValue = 1.e+20f ;
            mrsos:missing_value = 1.e+20f ;
            mrsos:original_name = "SOIL_WET" ;
            mrsos:comment = "includes subsurface frozen water but not
                surface snow and ice" ;
```

```
            mrsos:history = " At   17:54:19 on 06/07/2004: CMOR altered
                the data in the following ways: replaced missing value
                flag (1.00000E+28) with standard missing value
                (1.00000E+20);" ;

// global attributes:
            :title = "GICC  model output prepared for IPCC Fourth
                Assessment 2xCO2 equilibrium experiment" ;
            :institution = "GICC (Generic International Climate Center,
                Geneva, Switzerland)" ;
            :source = "GICCM1 (2002): atmosphere:  GICAM3
                (gicam_0_brnchT_itea_2, T63L32); ocean: MOM
                (mom3_ver_3.5.2, 2x3L15); sea ice: GISIM4; land:
                GILSM2.5" ;
            :contact = "Rusty Koder (koder@middle_earth.net)" ;
            :project_id = "IPCC Fourth Assessment" ;
            :table_id = "Table A1 (7 April 2004)" ;
            :experiment_id = "2xCO2 equilibrium experiment" ;
            :realization = 1 ;
            :Conventions = "CF-1.0" ;
            :history = "Output from archive/giccm_03_std_2xCO2_2256.  At
                17:54:19 on 06/07/2004, CMOR rewrote data to comply with
                CF standards and IPCC Fourth Assessment requirements" ;
            :references = "Model described by Koder and Tolkien (J.
                Geophys. Res., 2001, 576-591).  Also see
                http://www.GICC.su/giccm/doc/index.html  2XCO2
                simulation described in Dorkey et al. (Clim. Dyn., 2003,
                323-357.)" ;
            :comment = "Equilibrium reached after 30-year spin-up after
                which data were output starting with nominal date of
                January 2030" ;
data:

 lon = 0, 90, 180, 270 ;

 lon_bnds =
  -45, 45,
  45, 135,
  135, 225,
  225, 315 ;

 lat = 10, 20, 30 ;

 lat_bnds =
  5, 15,
  15, 25,
  25, 35 ;

 time = 15, 45 ;

 time_bnds =
  0, 30,
  30, 60 ;

 depth = 0.05 ;

 depth_bnds = 0, 0.1 ;
```

```
 mrsos =
  10, 50, 90, 130,
  170, 210, 250, 290,
  330, 370, 410, 450,
  20, 60, 100, 140,
  180, 220, 260, 300,
  340, 380, 420, 460 ;
}
```

This example should be updated:

Example 4: Treatment of a ocean basin labels
Northward ocean heat transport
(a function of latitude, ocean basin, month)

```
netcdf hfogo_O1 {
dimensions:
      lat = 3 ;
      region = 4 ;
      time = UNLIMITED ; // (2 currently)
      bnds = 2 ;
      strlen = 14 ;
variables:
      double lat(lat) ;
            lat:standard_name = "latitude" ;
            lat:long_name = "latitude" ;
            lat:units = "degrees_north" ;
            lat:axis = "Y" ;
            lat:bounds = "lat_bnds" ;
      double lat_bnds(lat, bnds) ;
      char geo_region(region, strlen) ;
            geo_region:standard_name = "region" ;
            geo_region:long_name = "Ocean Basin" ;
      double time(time) ;
            time:standard_name = "time" ;
            time:long_name = "time" ;
            time:units = "days since 2030-1-1" ;
            time:axis = "T" ;
            time:calendar = "360_day" ;
            time:bounds = "time_bnds" ;
      double time_bnds(time, bnds) ;
      float hfogo(time, region, lat) ;
            hfogo:standard_name = "northward_ocean_heat_transport" ;
            hfogo:long_name = "Northward Ocean Heat Transport" ;
            hfogo:units = "W" ;
            hfogo:cell_methods = "time: mean (interval: 20 minutes)
                longitude: sum" ;
            hfogo:coordinates = "geo_region" ;
            hfogo:_FillValue = 1.e+20f ;
            hfogo:missing_value = 1.e+20f ;
            hfogo:original_name = "OFLUX" ;
            hfogo:history = " At   18:06:56 on 06/07/2004: CMOR altered
                the data in the following ways: replaced missing value
                flag (1.00000E+28) with standard missing value
                (1.00000E+20);" ;

// global attributes:
            :title = "GICC  model output prepared for IPCC Fourth
                Assessment 2xCO2 equilibrium experiment" ;
            :institution = "GICC (Generic International Climate Center,
                Geneva, Switzerland)" ;
            :source = "GICCM1 (2002): atmosphere:  GICAM3
                (gicam_0_brnchT_itea_2, T63L32); ocean: MOM
```

```
                (mom3_ver_3.5.2, 2x3L15); sea ice: GISIM4; land:
                GILSM2.5" ;
            :contact = "Rusty Koder (koder@middle_earth.net)" ;
            :project_id = "IPCC Fourth Assessment" ;
            :table_id = "Table O1 (4 June 2004)" ;
            :experiment_id = "2xCO2 equilibrium experiment" ;
            :realization = 1 ;
            :Conventions = "CF-1.0" ;
            :history = "Output from archive/giccm_03_std_2xCO2_2256.  At
                18:06:56 on 06/07/2004, CMOR rewrote data to comply with
                CF standards and IPCC Fourth Assessment requirements" ;
            :references = "Model described by Koder and Tolkien (J.
                Geophys. Res., 2001, 576-591).  Also see
                http://www.GICC.su/giccm/doc/index.html  2XCO2
                simulation described in Dorkey et al. (Clim. Dyn., 2003,
                323-357.)" ;
            :comment = "Equilibrium reached after 30-year spin-up after
                which data were output starting with nominal date of
                January 2030" ;
data:

 lat = 10, 20, 30 ;

 lat_bnds =
  5, 15,
  15, 25,
  25, 35 ;

 geo_region =
  "atlantic_ocean",
  "indian_ocean  ",
  "pacific_ocean ",
  "global_ocean  " ;

 time = 15, 45 ;

 time_bnds =
  0, 30,
  30, 60 ;

 hfogo =
  -1.9e+15, -1.5e+15, -1.1e+15,
  -3e+14, 1e+14, 5e+14,
  1.3e+15, 1.7e+15, 2.1e+15,
  2.9e+15, 3.3e+15, 3.7e+15,
  -1.8e+15, -1.4e+15, -1e+15,
  -2e+14, 2e+14, 6e+14,
  1.4e+15, 1.8e+15, 2.2e+15,
  3e+15, 3.4e+15, 3.8e+15 ;
}
```

This example should be updated:

Example 5: Treatment of model level data
Cloud fraction
(a function of longitude, latitude, model level, month)

```
netcdf cl_A1 {
dimensions:
      lon = 4 ;
      lat = 3 ;
      lev = 5 ;
      time = UNLIMITED ; // (2 currently)
      bnds = 2 ;
variables:
      double lon(lon) ;
            lon:standard_name = "longitude" ;
            lon:long_name = "longitude" ;
            lon:units = "degrees_east" ;
            lon:axis = "X" ;
            lon:bounds = "lon_bnds" ;
      double lon_bnds(lon, bnds) ;
      double lat(lat) ;
            lat:standard_name = "latitude" ;
            lat:long_name = "latitude" ;
            lat:units = "degrees_north" ;
            lat:axis = "Y" ;
            lat:bounds = "lat_bnds" ;
      double lat_bnds(lat, bnds) ;
      double lev(lev) ;
            lev:standard_name =
                "atmosphere_hybrid_sigma_pressure_coordinate" ;
            lev:long_name = "hybrid sigma pressure coordinate" ;
            lev:units = "1" ;
            lev:axis = "Z" ;
            lev:positive = "down" ;
            lev:bounds = "lev_bnds" ;
            lev:formula = "p(n,k,j,i) = a(k)*p0 + b(k)*ps(n,j,i)" ;
            lev:formula_terms = "p0: p0 a: a b: b ps: ps" ;
      double lev_bnds(lev, bnds) ;
            lev_bnds:standard_name =
                "atmosphere_hybrid_sigma_pressure_coordinate" ;
            lev_bnds:formula = "p(n,k,j,i) = a(k)*p0 + b(k)*ps(n,j,i)" ;
            lev_bnds:formula_terms = "p0: p0 a: a_bnds b: b_bnds ps: ps"
                ;
      double time(time) ;
            time:standard_name = "time" ;
            time:long_name = "time" ;
            time:units = "days since 2030-1-1" ;
            time:axis = "T" ;
            time:calendar = "360_day" ;
            time:bounds = "time_bnds" ;
      double time_bnds(time, bnds) ;
      double a_bnds(lev, bnds) ;
            a_bnds:long_name = "hybrid sigma coordinate A coefficient
                for layer bounds" ;
```

```
      double b_bnds(lev, bnds) ;
            b_bnds:long_name = "hybrid sigma coordinate B coefficient
               for layer bounds" ;
      float p0 ;
            p0:long_name = "reference pressure for hybrid sigma
               coordinate" ;
            p0:units = "Pa" ;
      double a(lev) ;
            a:long_name = "hybrid sigma coordinate A coefficient for
               layer" ;
      double b(lev) ;
            b:long_name = "hybrid sigma coordinate B coefficient for
               layer" ;
      float ps(time, lat, lon) ;
            ps:long_name = "Surface Pressure" ;
            ps:units = "Pa" ;
      float cl(time, lev, lat, lon) ;
            cl:standard_name = "cloud_area_fraction" ;
            cl:long_name = "Total Cloud Fraction" ;
            cl:units = "%" ;
            cl:cell_methods = "time: mean (interval: 20 minutes)" ;
            cl:_FillValue = 1.e+20f ;
            cl:missing_value = 1.e+20f ;
            cl:original_name = "CLOUD" ;
            cl:history = " At   17:54:19 on 06/07/2004: CMOR altered the
               data in the following ways: replaced missing value flag
               (1.00000E+28) with standard missing value
               (1.00000E+20);" ;

// global attributes:
            :title = "GICC  model output prepared for IPCC Fourth
               Assessment 2xCO2 equilibrium experiment" ;
            :institution = "GICC (Generic International Climate Center,
               Geneva, Switzerland)" ;
            :source = "GICCM1 (2002): atmosphere:  GICAM3
               (gicam_0_brnchT_itea_2, T63L32); ocean: MOM
               (mom3_ver_3.5.2, 2x3L15); sea ice: GISIM4; land:
               GILSM2.5" ;
            :contact = "Rusty Koder (koder@middle_earth.net)" ;
            :project_id = "IPCC Fourth Assessment" ;
            :table_id = "Table A1 (7 April 2004)" ;
            :experiment_id = "2xCO2 equilibrium experiment" ;
            :realization = 1 ;
            :Conventions = "CF-1.0" ;
            :history = "Output from archive/giccm_03_std_2xCO2_2256.  At
               17:54:19 on 06/07/2004, CMOR rewrote data to comply with
               CF standards and IPCC Fourth Assessment requirements" ;
            :references = "Model described by Koder and Tolkien (J.
               Geophys. Res., 2001, 576-591).  Also see
               http://www.GICC.su/giccm/doc/index.html  2XCO2
               simulation described in Dorkey et al. (Clim. Dyn., 2003,
               323-357.)" ;
            :comment = "Equilibrium reached after 30-year spin-up after
               which data were output starting with nominal date of
               January 2030" ;
data:
```

```
lon = 0, 90, 180, 270 ;

lon_bnds =
 -45, 45,
 45, 135,
 135, 225,
 225, 315 ;

lat = 10, 20, 30 ;

lat_bnds =
 5, 15,
 15, 25,
 25, 35 ;

lev = 0.100000001490116, 0.300000004470348, 0.500000014901161,
   0.700000002980232, 0.900000013411045 ;

lev_bnds =
 0, 0.200000006705523,
 0.200000006705523, 0.400000005960464,
 0.400000005960464, 0.599999994039536,
 0.599999994039536, 0.799999982118607,
 0.799999982118607, 1 ;

time = 15, 45 ;

time_bnds =
 0, 30,
 30, 60 ;

a_bnds =
 0, 0.150000005960464,
 0.150000005960464, 0.25,
 0.25, 0.25,
 0.25, 0.150000005960464,
 0.150000005960464, 0 ;

b_bnds =
 0, 0.0500000007450581,
 0.0500000007450581, 0.150000005960464,
 0.150000005960464, 0.349999994039536,
 0.349999994039536, 0.649999976158142,
 0.649999976158142, 1 ;

p0 = 100000 ;

a = 0.100000001490116, 0.200000002980232, 0.300000011920929,
   0.200000002980232, 0.100000001490116 ;

b = 0, 0.100000001490116, 0.200000002980232, 0.5, 0.800000011920929 ;

ps =
 97100, 97500, 97900, 98300,
 98700, 99100, 99500, 99900,
 100300, 100700, 101100, 101500,
 97200, 97600, 98000, 98400,
```

```
  98800, 99200, 99600, 100000,
  100400, 100800, 101200, 101600 ;

 cl =
  50.1, 50.5, 50.9, 51.3,
  51.7, 52.1, 52.5, 52.9,
  53.3, 53.7, 54.1, 54.5,
  56.5, 56.9, 57.3, 57.7,
  58.1, 58.5, 58.9, 59.3,
  59.7, 60.1, 60.5, 60.9,
  62.9, 63.3, 63.7, 64.1,
  64.5, 64.9, 65.3, 65.7,
  66.1, 66.5, 66.9, 67.3,
  69.3, 69.7, 70.1, 70.5,
  70.9, 71.3, 71.7, 72.1,
  72.5, 72.9, 73.3, 73.7,
  75.7, 76.1, 76.5, 76.9,
  77.3, 77.7, 78.1, 78.5,
  78.9, 79.3, 79.7, 80.1,
  50.2, 50.6, 51, 51.4,
  51.8, 52.2, 52.6, 53,
  53.4, 53.8, 54.2, 54.6,
  56.6, 57, 57.4, 57.8,
  58.2, 58.6, 59, 59.4,
  59.8, 60.2, 60.6, 61,
  63, 63.4, 63.8, 64.2,
  64.6, 65, 65.4, 65.8,
  66.2, 66.6, 67, 67.4,
  69.4, 69.8, 70.2, 70.6,
  71, 71.4, 71.8, 72.2,
  72.6, 73, 73.4, 73.8,
  75.8, 76.2, 76.6, 77,
  77.4, 77.8, 78.2, 78.6,
  79, 79.4, 79.8, 80.2 ;
}
```