DEPARTMENT OF PHYSICAL OCEANOGRAPHY

CMIP5 COMMUNITY STORAGE SERVER

User Manual and Misc. Documentation

Alexander K. Ekholm
Engineer Assistant I
Physical Oceanography
Office Phone: +1 (508) 289 - 4930
aekholm@whoi.edu

Version 0.04
Last Updated: July 23, 2013

1. Contents

2. Figures

3. Tables

4. Introduction

5. CMIP5 Data Reference Syntax (DRS) and Controlled Vocabularies

The CMIP5 *Data Reference Syntax (DRS)* provides a common naming system to be used in files, directories, metadata and URLS to identify datasets wherever they may be located in the CMIP5 archive. It provides a clear and structured set of conventions to facilitate naming of data entities within the data archive of files delivered to end-users.

5.1. Atomic Dataset

An *atomic dataset* is a subset of the output saved from a single model run which is uniquely characterized by a single activity, product, institute, model, experiment, data sampling frequency, modeling realm, variable name, MIP table, ensemble member, and version number.

5.2. Publication-level Dataset

A *publication-level dataset*[1] is the collection of atomic datasets which share a single combination of all DRS component values except variable name but which might include only selected time intervals (i.e., not necessarily the entire temporal domain) of the contributing atomic datasets. The publication-level dataset therefore represents, in general, an intersection of several atomic datasets.

5.3. Component Definitions and Controlled Vocabularies

The defining components of the DRS are described in this section. The DRS consists of eleven individual facets that may be used to uniquely index individual datum.

5.3.1. Activity

Activity identifies the model intercomparison activity or other data collection activity. For CMIP5 all the archived data will be discoverable under the "CMIP5" activity. For "Transpose AMIP", the data will be archived under the "TAMIP" activity. In some cases there may be other activities (e.g., CFMIP and PMIP), which have been coordinated with CMIP5, so these activities may be cross-referenced or aliased with CMIP5 for certain portions of the CMIP5 archive.

5.3.2. Product

Product currently has four options: "output", "output1", "output2", and "unsolicited". For CMIP5, files will initially be designated as "output" or "unsolicited". Subsequently, data from the requested variable list will be assigned a version (see below) and placed in either "output1" or "output2. Variables not specifically requested by CMIP5 will remain designated "unsolicited". In some cases a continuous sequence of model data will be split between "output1" and "output2" in order to facilitate archive management. Note that although output of some variables is requested only for limited time-periods, if output of those variables is made available for other time periods, it will also be treated as "output", not as "unsolicited".

It is likely that various data products derived from this output will be produced subsequently which could be identified by a different term (e.g., "derived" or "processed"), but this is not part of the current DRS.

5.3.3. Institute

Institute identifies the institute responsible for the model results (e.g. UKMO), and it should be as short as possible. For CMIP5 the institute name will be suggested by the research group at the institute, subject to final authorization by PCMDI. This name may differ somewhat from the official CMIP5 institute_id (recorded as a global attribute in CMIP5 output files), which should be used to identify models in journal articles. (The official institute_id might, for example, include characters such as a blank, a period, or a parenthesis, which are not allowed in the DRS "institute" component.)

[1] Publication-level datasets have previously been referred to as "Realm-level datasets" in internet communications related to CMIP5 such as email lists and wiki pages.

5.3.4. Model

Model identifies the model used (e.g. HADCM3, HADCM3-233). Subject to certain constraints imposed by PCMDI, the modeling group will assign this name, which might include a version number (usually truncated to the nearest integer). This name may differ somewhat from the official CMIP5 model_id (recorded as a global attribute in CMIP5 output files),, which should be used to identify models in journal articles. [The official model_id might, for example, include characters such as a blank, a period, or a parenthesis, which are not allowed in the DRS "model" component.] The model identifier will normally change if any aspect of the model is modified (e.g., if the resolution is changed). An exception may be made if the modifications to the model are clearly implied by the experiment design. If, for example, a coupled atmosphere-ocean model performs an AMIP simulation (which clearly implies prescribed SSTs and sea ice, rather than a fully interactive ocean), then the name may not necessarily be modified. Another exception is when closely-related "perturbed physics" versions of a model are run, in which case the different model versions can be uniquely identified by assigning each a different "p" value in defining the "ensemble member" (described below).

5.3.5. Experiment

Experiment identifies either the experiment or both the experiment family and a specific type within that experiment family. In CMIP5, for example, "rcp45" refers to a particular experiment in which a "representative concentration pathway" (RCP) has been specified which leads to an approximate radiative forcing of 4.5 W m-2. As another example, "historicalGHG" is a simulation of the historical" period, but with forcing other than anthropogenic "greenhouse gas" forcing suppressed. In this latter case, "historical" is the experiment family and "GHG" is used to designate the specific type of historical run. These experiment names are not freely chosen, but come from controlled vocabularies defined in the Appendix 1.1 of this document under the column labeled "Short Name of Experiment". Note that in some cases there will be slight variations of the same experiment (e.g., different simulations performed within the historicalMisc family might be forced with different individual forcings or suites of forcings, as discussed further under "Ensemble member" below).

5.3.6. Frequency

Frequency indicates the interval between individual time-samples in the atomic dataset. For CMIP5, 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). These are specified for each variable in the "standard_output" spreadsheet found at http://cmip-pcmdi.llnl.gov/cmip5/output_req.html. Note that for CMIP5, quantities derived from an atomic dataset of a given frequency will be assigned the same frequency, even in the case when a time-average has been performed. (See example under section 2.4 involving time averages.)

5.3.7. Modeling Realm

Modeling realm indicates which high level modeling component is of particular relevance for the dataset. For CMIP5, permitted values are: "atmos", "ocean", "land", "landIce", "seaIce", "aerosol" "atmosChem", ocnBgchem (ocean biogeochemical). These are specified for each variable in the "standard_output" spreadsheet which can be accessed at http://cmippcmdi.llnl.gov/cmip5/output_req.html. Note that sometimes a variable will be equally (or almost equally relevant) to two or more "realms", in which case the atomic dataset might be assigned to a primary "realm", but cross-referenced or aliased to the other relevant "realms".

5.3.8. Variable Name

Variable name and the MIP table component of the DRS (defined next) identify the physical quantity and often imply something about the sampling frequency and modeling realm. For CMIP5 the variable name and MIP table for requested output appear in the "standard_output" spreadsheet available at http://cmip-pcmdi.llnl.gov/cmip5/output_req.html. Monthly mean surface air temperature, for example, has a "variable name" of "tas" and is found in the "Amon" MIP table., Note that hyphens (-) are forbidden in CMIP5 variable names.

5.3.9. MIP Table

MIP table: See description under the "variable name" component directly above. For CMIP5 each MIP table contains fields sampled only at a single frequency (although in the case of monthly mean data the DRS will place some of the monthly means in the "mon" DRS frequency category and others in the monClim DRS frequency category, as appropriate).

5.3.10. Ensemble Member (r<N>i<M>p<L>)

Ensemble member (r<N>i<M>p<L>): This triad of integers (N, M, L), formatted as shown above (e.g., "r3i1p21") distinguishes among closely related simulations by a single model. All three are required even if only a single simulation is performed.

The so-called "realization" number (a positive integer value of "N") is used to distinguish among members of an ensemble typically generated by initializing a set of runs with different, but equally realistic, initial conditions. CMIP5 historical runs initialized from different times of a control run, for example, would be identified by "r1", "r2", "r3", etc.). The data supplier must assign a realization number to each atomic dataset. It is generally recommended that the numbers be assigned sequentially starting with 1 (but other recommendations, specified below, may override this recommendation). In CMIP5, time-independent variables (i.e., those with frequency="fx") are not expected to differ across ensemble members, so for these N should be invariably assigned the value zero ("r0"). For TAMIP ("the Transpose AMIP activity), the "realization" number is used to distinguish among the 16 members of each of 4 ensembles (one for each of 4 "seasons") generated from different observed conditions, spaced 30 hours apart. So, for example, the 16-member ensemble of runs initialized at 00Z on 15 Oct 2008, 06Z 16 Oct 2008, 12Z 17 Oct 2008, and so-on, would be assigned "r1", "r2", "r3", etc.

Models used for forecasts that depend on the initial conditions might be initialized from observations using different methods or different observational datasets. These should be distinguished by assigning different positive integer values of "M" in the "initialization method indicator" (i<M>). For CMIP5 this indicator might in some cases be needed to distinguish among runs in the decadal-prediction suite of experiments (1.1-1.6). The data supplier must assign an initialization method number to each atomic dataset. It is recommended that the numbers be assigned sequentially starting with 1. In CMIP5, time-independent variables (i.e., those with frequency="fx") are not expected to differ across ensemble members, so for these M should invariably be assigned the value zero ("i0"). A key (i.e., a table) should be made available that associates each value of M with a particular initialization method and/or observational dataset.

If there are many closely related model versions, which, as a group, are generally referred to as a perturbed physics ensemble (e.g., QUMP or climateprediction.net ensembles), then these should be distinguishable by a "perturbed physics" number, p<L>, where the positive integer value of L is uniquely associated with a particular set of model parameters (e.g., r3i1p78 is a third realization of the seventy-eighth version of the perturbed physics model). If there are different "forcing" combinations prescribed in experiment 7.3 in CMIP5 (the "historicalMisc" runs), then each of these different runs are also assigned different values of L (in "p<L>"). Note that the data supplier must assign a physics version number to each atomic dataset. It is recommended that the numbers be assigned sequentially starting with 1. In CMIP5, time-independent variables (i.e., those with frequency="fx") are not expected to differ across ensemble members, so for these L should always be assigned the value zero ("p0"). A key (i.e., a table) should be made available that associates each value of L with a particular set of model parameter values and/or, in the case of the "historicalMisc" experiment, a particular suite of "forcing" agents.

Note that for a single model and experiment N, M, and L should be interpretable independently; for all members of the ensemble, the correspondence between the values of N, M, and L and the simulation characteristics they represent should be consistent. For example the two different ensemble members, r3i1p7 and r3i1p8, should both be initialized from exactly the same initial conditions using the same method (because the "r" and "i" values are identical) although the subsequent evolution of the simulations will presumably differ since they were produced by two different "perturbed physics" versions of the same model. Note that there may be cases where "gaps" could occur in the list of ensemble members. If, for example, two different initialization procedures were used, but the second procedure was tested with only a subset of the initial condition cases of the first procedure (say, every other case). Then the list of ensemble members would look like: r1i1p1, r2i1p1 r3i1p1, r4i1p1, r5i1p1, r6i1p1, …, r1i2p1, r3i2p1, r5i2p1, …

A recommendation for CMIP5 is that each so-called RCP (future scenario) simulation should when possible 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. Thus, for example, suppose a 3-member ensemble of historical runs of a model exists, and a single rcp45 simulation was produced, initialized from the third member of the historical ensemble. The recp45 simulation would be designated "r3" (rather than "r1"), even though it is the only existing ensemble member, in order to indicate that it was spawned from member 3 of the historical ensemble. A similar convention should be followed, when appropriate, with other simulations (e.g., the decadal simulations).

5.3.11. Version Number

Version number (vN): The version number will be 'v' followed by an integer, which uniquely identifies a particular version of a publication-level dataset (e.g., perhaps distinguishing between an original version of the output that might have been found to be flawed in some respect--perhaps due to some improper post-processing procedure-- and a subsequent version in which the data were corrected). For CMIP5 the version number is supposed to reflect the date of publication: for example, "v20100105" for a version provided on 5th January 2010. Software interpreting version numbers should not, however, assume the integer has invariably been correctly encoded (e.g., sometimes a single digit number might appear as in "v3").

Version numbers are assigned to publication-level datasets (and therefore the version generally applies to multiple atomic datasets). The version number of a publication-level dataset (and all the atomic datasets in it) is updated when:

a) any file included in the publication-level dataset is modified, replaced, or removed, or
b) an additional file is added to the publication-level dataset.

5.4. Extended Path

5.5. Using the DRS Syntax

The section describes best-practice use of file and directory encoding using the CMIP5 DRS.

5.5.1. CMOR Directory Structure

The standard CMIP5 output tool CMOR2[2] optionally writes output files to a directory structure mapping DRS components to directory names as:

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

Example:

/CMIP5/output/MOHC/HadCM3/decadal1990/day/atmos/tas/r3i2p1/

This structure, based on a previous version of the DRS, is incompatible with the recommended current DRS directory structure (see below). However it remains relevant as a possible structure for model output prior to transforming into the DRS directory structure.

5.5.2. ESGF Data Node Directory Structure

It is recommended that ESGF data nodes should layout datasets on disk mapping DRS components to directories as:

<activity>/<product>/<institute>/<model>/<experiment>/<frequency>/<modeling realm>/<MIP table>/<ensemble member>/<version number>/<variable name>/<CMOR filename>.nc

Example:

/CMIP5/output1/UKMO/HadCM3/decadal1990/mon/atmos/Amon/r3i2p1/v20100105/tas/tas_Amon_HADCM3_decadal1990_r3i2p1_199001-199012.nc

5.5.3. CMIP5 Filename Encoding

Because users will download data into a file system that will usually differ from the archival directory structure (and because in some cases it aids in archive management), the filename structure should include some DRS content. For CMIP5 the filename will be constructed as follows:

filename = <variable name>_<MIP table>_<model>_<experiment>_<ensemble member>[_<temporal-subset>][_<geographical info>].nc

[2] See the Climate Model Output Rewriter: http://www2-pcmdi.llnl.gov/cmor/documentation/

where:

- <variable name>, <MIP table>, <model>, <experiment>, and <ensemble member> are DRS components,
- The < temporal subset> (along with the preceding underscore) is omitted for variables that are time-independent, and the geographical information (preceded by an underscore) is included only when needed.

Example:

tas_Amon_HADCM3_ historical_r1i1p1_185001-200512.nc

In CMIP5 there is a single exception to use of the above template. For so-called gridspec files,which describe the grids used in a model, the filename should be constructed as follows:

gridspec filename = gridspec_<modeling realm>_fx_<model>_ <experiment>_r0i0p0.nc

where <modeling realm> is now included and the variable name is replaced by "grid_spec". Note also that this is a time-independent field, so the CMIP5 table is "fx" and the ensemble member is set to "r0i0p0".

Example:

gridspec_atmos_fx_IPSL-CM5_historical_r0i0p0.nc

5.5.4. Publication-level Dataset ID Encoding

Publication-level datasets are assigned an identifier dataset_id within THREDDS catalogs on ESGF data nodes. The CMIP5 best practices document4defines a publication-level dataset_idas:

<activity>.<product>.<institute>.<model>.<experiment>.<frequency>.<modeling realm>.<MIP table>.<ensemble member>

Each publication-level dataset version will have the THREDDS id:

<activity>.<product>.<institute>.<model>.<experiment>.<frequency>.<modeling realm>.<MIP table>.<ensemble member>.<version>

Note that the version number assigned to the dataset by ESG is supposed to reflect the date of ESG publication, but the version will usually be assigned by the user so this cannot generally be guaranteed. The user will be instructed to provide ESG with the date that appears in the ESGF data node directory structure for the dataset being published (assuming that the directory version number is a correctly encoded date). In many cases the directory structure will be generated some days prior to publication, so the date will not in fact reflect the date of publication, but the date that the directory structure was created.

5.5.5. URL Syntax

URLs referencing the data files will have a site dependent prefix (that may change due to site-specific data management tasks) followed by the directory structure. This directory structure should (but may not) follow the recommendations of section 5.5.3 above.

6. Data Access and Availability

6.1. The ESGF Search RESTful API

The ESGF search service exposes a RESTful URL that can be used by clients (browsers and desktop clients) to query the contents of the underlying search index, and return results matching the given constraints. Because of the distributed capabilities of the ESGF search, the URL at any Index Node can be used to query that Node only, or all Nodes in the ESGF system.

6.1.1. Syntax

The general syntax of the ESGF search service URL is:

http://<base_search_URL>/search?[keyword parameters as (name, value) pairs][facet parameters as (name,value) pairs]

where <base_search_URL> is the base URL of the search service at a given Index Node.

All parameters (keyword and facet) are optional. Also, the value of all parameters must be URL-encoded, so that the complete search URL is well formed.

6.1.2. Keywords

Keyword parameters are query parameters that have reserved names, and are interpreted by the search service to control the fundamental nature of a search request: where to issue the request to, how many results to return, etc.

The following keywords are currently used by the system - see below for usage examples:

- **facets=** to return facet values and counts
- **shards=** to specify an explicit list of shards to be queried
- **offset=, limit=** to paginate through the available results (default: offset=0, limit=10)
- **fields=** to return only specific metadata fields for each matching result (default: fields=*)
- **format=** to specify the response document output format

6.1.3. Core Facets

Facet parameters are "search categories" that can be used to apply constraints to the search, and thus reduce the number of results returned. Internally, facets are metadata fields (single valued or multi-valued) that are stored for each search record. The search service will select records for which the metadata field values match the corresponding facet constraints.

The following facets are core system facets, and their names are reserved in the system. These facets can be used as valid query parameters at all sites in the federation.

- **query=** for free text searches (default: query=*)
- **distrib=true** to execute a distributed query, distrib=false to execute a local query (default: distrib=true)
- **id, master_id, instance_id**: core record identifiers carrying different semantics - see later for detailed explanation.
- **title**: record (short) title
- **description**: record (longer) description
- **type**: denotes the intrinsic type of the record. Currently supported values: Dataset, File, Aggregation (default: Dataset)
- **replica**: indicates wether the record is the "master" copy, or a replica. Use replica=false to return only originals, replica=true to return only replicas (default: no replica flag specified, i.e. return both replicas and originals)
- **latest**: indicates wether the record is the latest available version, or a previous version. Use latest=true to return only the latest version of all records, latest=false to return previous versions (default: no latest flag specified, i.e. return all versions)
- **data_node**: indicates the Data Node where the data is stored
- **index_node**: the Index Node where the data is published
- **version**: the record version (a string)
- **timestamp**: the date and time when the record was last modified
- **url**: specific URL(s) to access the record
- **access**: high level access capability available for a record
- **xlink**: reference to external record documentation, such as technical notes
- **size**: record size (for Datasets or Files)
- **checksum**, checksum_type: file checksum value and type
- **number_of_files**: number of files contained in a dataset
- **number_of_aggregations**: number of aggregations in a dataset
- **dataset_id**: the "id" value of the enclosing dataset (Files and Aggregations only)

- **tracking_id**: the UUID assigned to a File by some special publication software, if available
- **drs_id**: a templated string assigned to a Dataset by some special publication software, if available. Note: this field is deprecated.
- **start=, end=** to execute a temporal range query
- **bbox=[west,south,east,north]** to execute a spatial coverage query
- **from=, to=** to execute a query based on the record last update date and time

6.1.4. Custom Facets

Additionally, each ESGF Index Node can harvest and make available additional custom facets that are relevant to its projects and users. For example, most Index Nodes support the set of CMIP5 facets, plus others. These custom facets are configured by the Node administrator in the file /esgf/config/facets.properties and can be discovered by the user through the following query:

Error! Hyperlink reference not valid.

Example - Determine all the allowed facet names and values at a specific site:

http://esg-datanode.jpl.nasa.gov/esg-search/search?facets=&limit=0&distrib=false*

6.1.5. CMIP5 Facets

The following set of facets is supported by most ESGF Index Nodes in the federation, and can be used to discover/query/retrieve CMIP5 data.

- **CF Standard Name**: cf_standard_name
- **Ensemble**: ensemble
- **Experiment**: experiment
- **Experiment Family**: experiment_family
- **Institute**: institute
- **MIP Table**: cmor_table
- **Model**: model
- **Project**: project
- **Product**: product
- **Realm**: realm
- **Time** Frequency: time_frequency
- **Variable**: variable
- **Variable** Long Name: variable_long_name
- **Instrument**: source_id

Example - all the possible values of the "model", "experiment" and "project" facets throughout the federation:

http://esg-datanode.jpl.nasa.gov/esg-search/search?facets=model,experiment,project&limit=0

6.1.6. Default Queries

If no parameters at all are specified, the search service will execute a query using all the default values, specifically:

- query=* (query all records)
- distrib=true (execute a distributed search)
- type=Dataset (return results of type "Dataset")

Example:

http://esg-datanode.jpl.nasa.gov/esg-search/search

6.1.7. Free Text Queries

The keyword parameter query= can be specified to execute a query that matches the given text anywhere in the records metadata fields. The parameter value can be any expression following the Apache Lucene query syntax (because it is passed "as-is" to the back-end Solr query), and must be URL-encoded.

Examples:

- Search for any text, anywhere: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=* (the default value of the query parameter)
- Search for humidity in all metadata fields: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=humidity
- Search for the exact sentence specific humidity in all metadata fields: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=%22specific%20humidity%22
- Search for the words specific AND humidity, but not necessarily in an exact sequence: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=specific%20humidity
- Search for the word observations ONLY in the metadata field product: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=product:observations
- Using logical AND: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=airs%20AND%20humidity (must use upper case "AND")
- Using logical OR: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=airs%20OR%20humidity
- (must use upper case "OR"). This is the same as using simply a blank space: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=airs%20humidity
- Search for all datasets that match an id pattern: http://esg-datanode.jpl.nasa.gov/esg-search/search?query=id:obs4MIPs.NASA-JPL.AIRS.*

6.1.8. Facet Queries

A request to the search service can be constrained to return only those records that match specific values for one or more facets. Specifically, a facet constraint is expressed through the general form: <facet_name>=<facet_value>, where <facet_name> is chosen from the controlled vocabulary of facet names configured at each site, and <facet_value> must match exactly one of the possible values for that particular facet.

When specifying more than one facet constraint in the request, multiple values for the same facet are combined with a logical OR, while multiple values for different facets are combined with a logical AND. For example, experiment=decadal2000&variable=hus will return all records that match experiment=decadal2000 AND variable=hus, while variable=hus&variable=ta will return all records that match variable=hus OR variable=ta.

A facet constraint can be negated by using the != operator. For example, model!=CCSM searches for all items that do NOT match the CCSM model. Note that all negative facets are combined in logical AND, for example model!=CCSM&model!=HadCAM searches for all items that do not match CCSM, and do not match HadCAM.

By default, no facet counts are returned in the output document. Facet counts must be explicitly requested by specifying the facet names individually (for example: facets=experiment,model) or via the special notation facets=*. The facets list must be comma-separated, and white spaces are ignored. Note also that at this time, the special notation facets=* will only count those facets that are explicitly configured in the file application-context.xml.

If facet counts is requested, facet values are sorted alphabetically (facet.sort=lex) , and all facet values are returned (facet.limit=-1), provided they match one or more records (facet.mincount=1)

The facet type must be always specified as part of any request to the ESGF search services, so that the appropriate records can be examined and returned. If not specified explicitly, the default value is type=Dataset.

Examples:

- http://esg-datanode.jpl.nasa.gov/esg-search/search?cf_standard_name=air_temperature
- http://esg-datanode.jpl.nasa.gov/esg-search/search?cf_standard_name=air_temperature&project=obs4MIPs
- Combining two values of the same facet with a logical OR: http://esg-datanode.jpl.nasa.gov/esg-search/search?project=obs4MIPs&variable=hus&variable=ta (search for all observational files that have variable ta or hus)
- Using a negative facet: http://esg-datanode.jpl.nasa.gov/esg-search/search?project=obs4MIPs&variable=hus&variable=ta&model!=Obs-AIRS (search for all observational datasets that have variable ta or hus, excluding those produced by AIRS)
- http://esg-datanode.jpl.nasa.gov/esg-search/search?project=obs4MIPs&variable!=ta&variable!=huss (search for all observational datasets that do not contain neither variable ta nor variable huss)
- Search by tracking id: http://esg-datanode.jpl.nasa.gov/esg-search/search?type=File&tracking_id=2209a0d0-9b77-4ecb-b2ab-b7ae412e7a3f
- Search by checksum: http://esg-datanode.jpl.nasa.gov/esg-search/search?type=File&checksum=cbff465c9cd8c9833fd7b85235be2d47
- Issue a query for all supported facets and their values at one site, while returning no results (note that only facets with one or more values are returned): http://esg-datanode.jpl.nasa.gov/esg-search/search?facets=*&limit=0&distrib=false

6.1.9. Temporal Coverage Queries

6.1.10. Spatial Coverage Queries

6.1.11. Timestamp (last update) Queries

6.1.12. Distributed Queries

6.1.13. Shard Queries

6.1.14. Replica Queries

6.1.15. Latest and Version Queries

6.1.16. Results Pagination

6.1.17. Sorting

6.1.18. Output Format

6.1.19. Returned Metadata Fields

6.1.20. Identifiers

6.1.21. Access URLs

6.1.22. Wget Scripting

7. Variables of Interest

This section describes the variables of interest for the CMIP5 experiment. Initial estimates of minimum storage capacity are given in Table 1.

Table 1: Initial estimates of minimum storage capacity.

Experiment	Years/ Model	No. of Models	Ocean (GB)	Amos (GB)	Land (GB)	Sea Ice (GB)	Ocean BCG (GB)	Total (GB)
Historical	150	46	2000	750	115	69		2934
RCP4.5	95	40	1100	400	60	38		1598
RCP8.5	95	36	1000	400	54	34		1488
Pre-industrial Control	500	41	6000	2200	410	205		8815
LGM	100	6	200	70	9	6		285
Past 1000 Years	1000	5	1400	530	25	50		2005
Middle Holocene	300	12	1000	400	18	36		1454
Total (GB)	2240	186	12700	4750	691	438	0	18579

7.1. Oceanographic Components

Table 2: 3D Oceanographic Variables.

Variable	Long Name	Units	Output Variable Name
T	Sea Water Potential Temperature	K	thetao
S	Sea Water Salinity	psu	so
U	Sea Water X Velocity	$\frac{m}{s}$	uo
V	Sea Water Y Velocity	$\frac{m}{s}$	vo

Table 3: 2D Oceanographic Variables.

Variable	Long Name	Units	Output Variable Name
BSF			
Qsw	Insolation (sunlight)		
Qlw	Net Infrared Radiation		
Qsen	Surface Upward Sensible Heat Flux	$\frac{W}{m^2}$	hfss
Qlat	Surface Upward Latent Heat Flux	$\frac{W}{m^2}$	hfls
	Surface Longwave Radiation (Downwelling)	$\frac{W}{m^2}$	rlds
	Surface Longwave Radiation (Upwelling)	$\frac{W}{m^2}$	rlus
	Surface Shortwave Radiation (Downwelling)	$\frac{W}{m^2}$	rsds
	Surface Shortwave Radiation (Upwellng)	$\frac{W}{m^2}$	rsus
SSH	Sea Surface Height Above Geoid	M	zos
$\boldsymbol{\tau}$	Surface Downward Wind Stress (Eastward)	Pa	tauu
$\boldsymbol{\tau}$	Surface Downward Wind Stress (Northward)	Pa	tauv

7.2. Atmospheric Components

Table 4: 3D Atmospheric Variables.

Variable	Long Name	Units	Output Variable Name
T	Air Temperature	K	ta

	Geopotential Height	m	zg
Hu	Humidity (Relative)	%	hus
Hu	Humidity (Specific)	1	hur
Ω	Omega ($= \frac{dp}{dt}$)	$\frac{Pa}{s}$	wap
U	Wind (Eastward)	$\frac{m}{s}$	ua
V	Wind (Northward)	$\frac{m}{s}$	va

Table 5: 2D Atmospheric Variables.

Variable	Long Name	Units	Output Variable Name
T	Near-surface Air Temperature (2m)	K	tas
T_{min}	Daily Minimum Near-surface Air Temperature (2m)	K	tasmin
T_{max}	Daily Minimum Near-surface Air Temperature (2m)	K	tasmax
ST	Surface (skin) Temperature (i.e. SST for open ocean)	K	ts
Hu	Near-Surface Humidity (Relative, 2m)	%	hurs
Hu	Near-Surface Humidity (Specific, 2m)	1	huss
Pr	Precipitation	$\frac{kg}{m^2 s}$	pr
U	Near-surface Wind (Eastward, 10m)	$\frac{m}{s}$	uas
V	Near-surface Wind (Northward, 10m)	$\frac{m}{s}$	vas
UV_{mean}	Near-surface Wind Speed (10m)	$\frac{m}{s}$	sfcWind
τ	Surface Downward Wind Stress (Eastward)	Pa	tauu
τ	Surface Downward Wind Stress (Northward)	Pa	tauv

7.3. Land Components

Table 6: 3D Land Variables.

Variable	Long Name	Units	Output Variable Name
	Water Content of Soil Layer	$\frac{kg}{m^2 s}$	mrlsl

Table 7: 2D Land Variables.

Variable	Long Name	Units	Output Variable Name
	Moisture in Upper Portion of Soil Colum (10cm)	$\frac{kg}{m^2}$	mrsos
	Total Soil Moisture Content	$\frac{kg}{m^2}$	mrso
	Surface Runoff	$\frac{kg}{m^2 s}$	mrros
	Total Runoff	$\frac{kg}{m^2 s}$	mrro
	Water Evaporation from Soil	$\frac{kg}{m^2 s}$	evspblsoi

7.4. Sea Ice Components

Table 8: 3D Sea Ice Variables.

Variable	Long Name	Units	Output Variable Name

Table 9: 2D Sea Ice Variables

Variable	Long Name	Units	Output Variable Name
	Sea Ice Area Fraction	%	sic

8. Application Layout

Figure 1 shows my proposed application layout. Users will access pages served by the Tomcat Server via the web. Individual pages are mapped to unique servlets that serve various functions, including querying ESGF Index Nodes, processing XML response documents, and downloading to the file system and analyzing raw datasets.

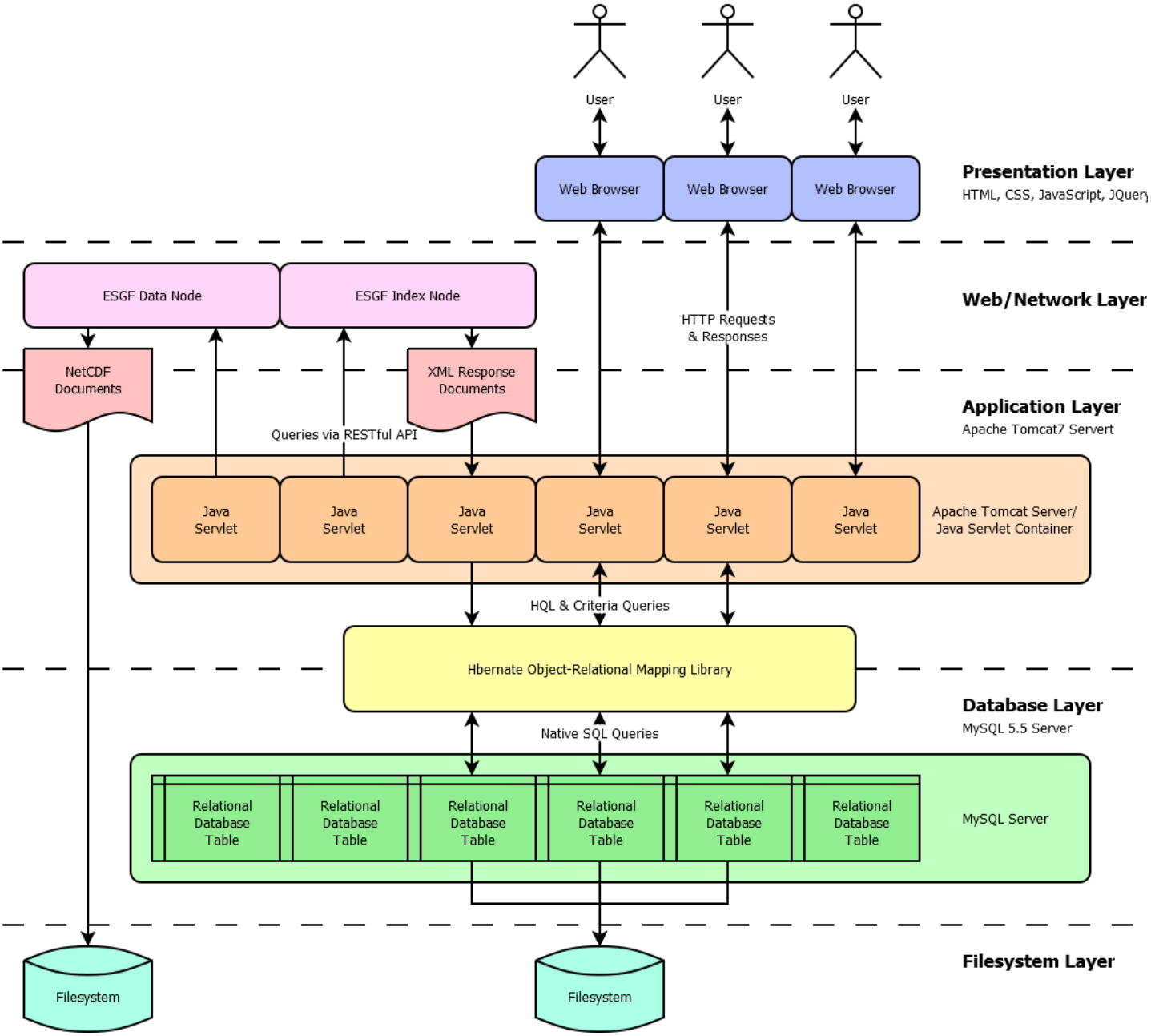

Figure 1: Proposed Application Layout

Java servlets communicate with the MySQL database via the Hibernate Object-Relation Mapping Library. Hibernate serves two important functions. First, by using javax.persistence annotations in the Java class definitions, the process of defining and generating relational database tables becomes completely automated. Secondly, Hibernate provides an abstracted interface to the MySQL server that supports programmatic generation of SQL queries.

9. Database Tables

Figure 2 is an EER diagram that shows the database tables that have been generated via Hibernate from existing Java classes. At the core of this schema is **Cmip5Dataset**, representing a publication-level dataset, which is uniquely indexed by 12 fields (as defined by the CMIP5 Data Reference Syntax). These 12 fields will also be used to generate the filesystem hierarchy on the server.

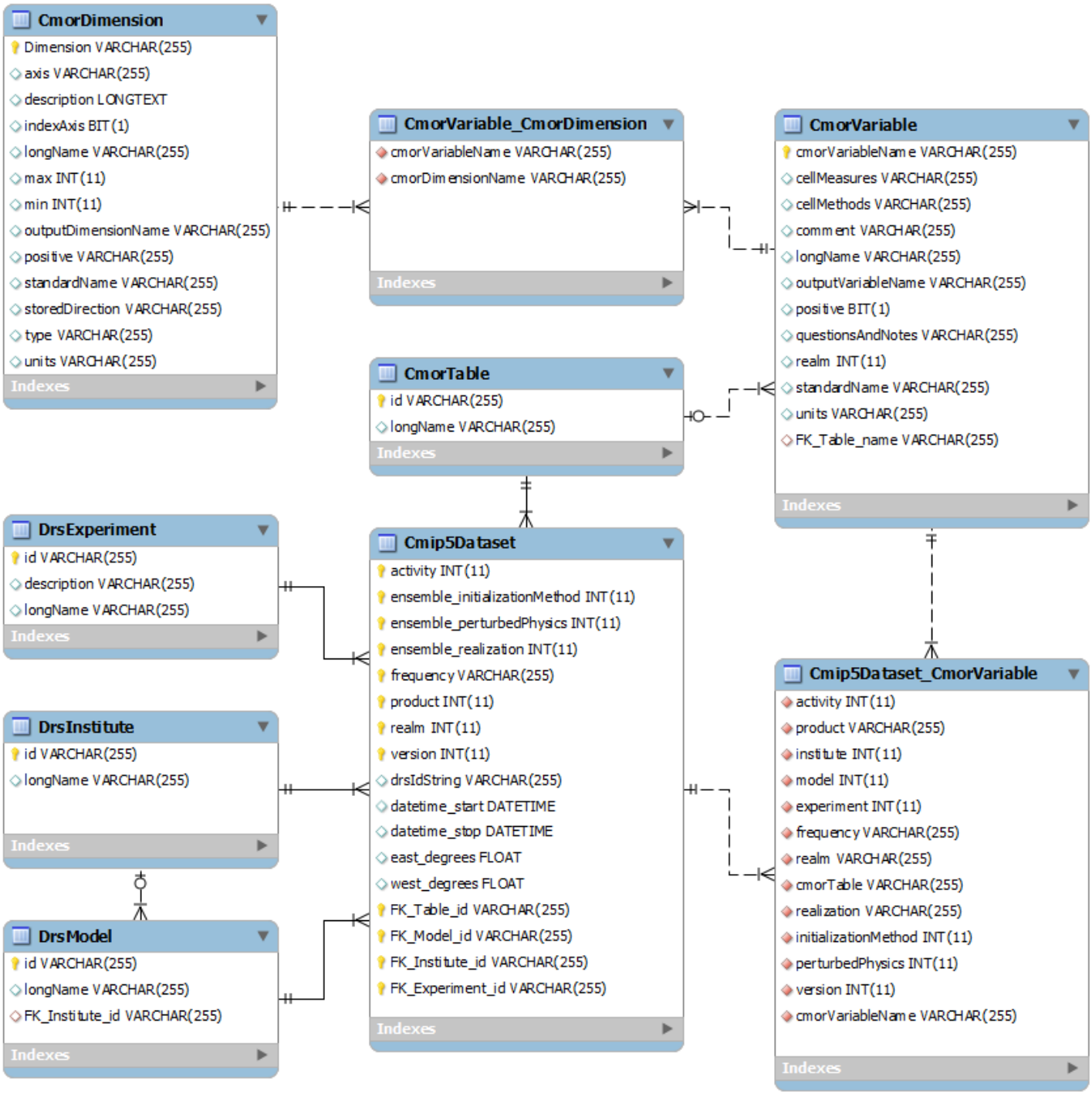

Figure 2: Database tables & relationships EER diagram, generated from Java classes via Hibernate object-relational mapping library

10. Java Classes

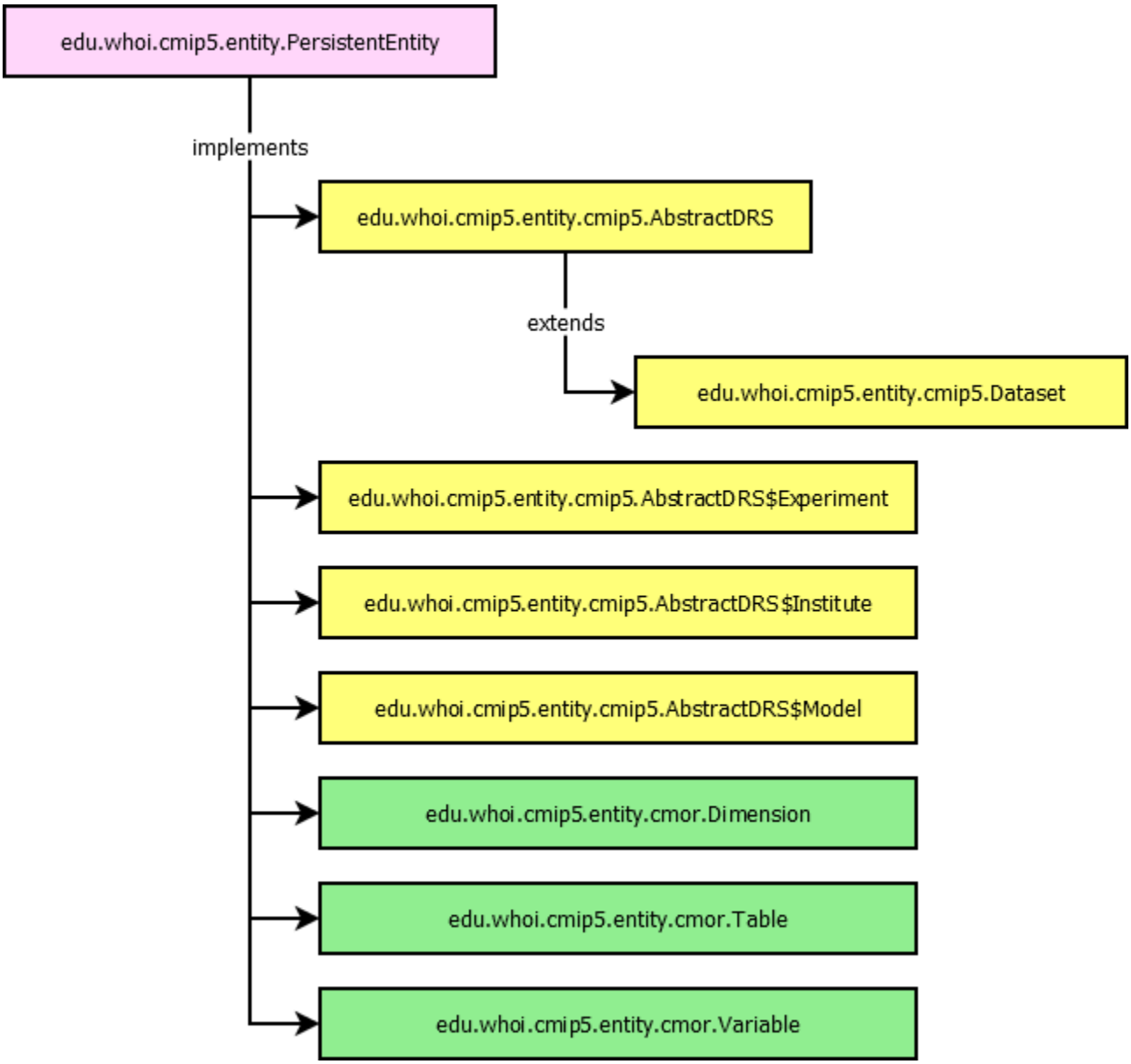

Figure 3: Java class type hierarchy

Figure 3 shows the Java classes from which the database tables are generated.

11. Index