ARCCSSive Documentation

Release 0.3.1.dev107+gfab958

ARCCSS CMS

Aug 07, 2017

Contents

ARCCSSive is a Python library developed by the CMS team at the ARC Centre of Excellence for Climate Systems Science for working with data at NCI. Contents:

CHAPTER 1

New Postgres Database

Install

Currently the v2 api is available as the 'postgres' branch of the repository:

```
git checkout https://github.com/coecms/ARCCSSive
cd ARCCSSive
git checkout -b postgres
conda env create -f conda/dev-environment
source activate arccssive-dev
pip install .
```

You could also use virtualenv if preferred

Use

Connect to the database:

```
from ARCCSSive.db import connect, Session
connect()
```

This will prompt you for your NCI password (it gets your username from $USER). Probably best not to use this in ipython notebook for the moment.

Create a session:

```
session = Session()
```

The session used to perform actual queries to the database (see http://docs.sqlalchemy.org/en/latest/orm/tutorial.html#querying)

Models

Models represent tables in the database, you can feed them to *session.query()*

Each model has a number of available relationships that can be used to *join()* to other models

Models are in the *ARCCSSive.model* package, divided into sub-packages by category (eg. the cmip5 package holds cmip5 related models)

ARCCSSive.model.cmip5.File

A CMIP5 file's metadata

Joins:

- dataset: *cmip5.Dataset* The dataset this file is part of
- version: *cmip5.Version*: This file's dataset version
- warnings: list[*cmip5.Warning*]: Warnings associated with this file
- timeseries: `cmip5.Timeseries` Holds all files in the dataset with the same variable at different time periods
- variables: list[*cfnetcdf.Variable*]: CF variables in the file (excluding axes)

ARCCSSive.model.cmip5.Dataset

A CMIP5 dataset, pretty much what you'd find listed on ESGF, with the exception that there's no version field (this is held separately in the *cmip5.Version* class)

Joins:

- versions: list[*cmip5.Version*]: Available versions of this dataset
- variables: list[`cmip5.Timeseries`]: The latest version of each of the variables in the dataset

ARCCSSive.model.cmip5.Version

A single version of a specific *cmip5.Dataset*. Different versions exist due to bugfixes after publication.

Joins:

- dataset: *cmip5.Dataset*: The dataset this version is associated with
- files: list[*cmip5.File*]: Files belonging to this version
- warnings: list[*cmip5.Warning*]: Warnings for this version
- variables: list[`cmip5.Timeseries`]: Collects the files by variable into timeseries

`ARCCSSive.model.cmip5.Timeseries`

All files for a single dataset, version and variable. The full output is often split into multiple files covering different time periods, the timeseries model joins these files back together again.

Joins:

- dataset: *cmip5.Dataset*: the dataset this timeseries is part of
- version: *cmip5.Version*: the dataset version
- files: list[*cmip5.File*]: Files belonging to the timeseries

ARCCSSive.model.cfnetcdf.File

Base class for CF compliant netcdf files

Joins:

- variables: list[*cfnetcdf.Variable*]: CF variables in the file (excluding axes)

ARCCSSive.model.cfnetcdf.Variable

A CF compliant variable

Joins:

- files: list[*cfnetcdf.File*]: Files containing the variable

Old-style models

These models are from the previous ARCCSSive iteration. They are mostly the same as the new version, however they also contain the variable name rather than this being a separate 'timeseries' table.

It's intended that these be moved to the new top-level *model* namespace

ARCCSSive.CMIP5.Model.Instance

Equivalent to the newer *cmip5.Dataset* class but also has the variable name

Joins:

- versions: list[*ARCCSSive.CMIP5.Model.Version*]: Versions of the instance
- latest_version: *ARCCSSive.CMIP5.Model.Version*: Latest version of the instance
- files: list[*cmip5.File*] Files belonging to the instance

ARCCSSive.CMIP5.Model.Version

Equivalent to the newer *cmip5.Version* class but also has the variable name

Joins:

- variable: *ARCCSSive.CMIP5.Model.Instance*: Instance this is a version of
- files: list[*cmip5.File*] Files belonging to the instance version
- new_version: *cmip5.Version*: Equivalent new-style version

CHAPTER 2

Installing

Raijin

The stable version of ARCCSSive is available as a module on NCI's Raijin supercomputer:

raijin $ module use ~access/modules raijin $ module load pythonlib/ARCCSSive

NCI Virtual Desktops

NCI's virtual desktops allow you to use ARCCSSive from a Jupyter notebook. For details on how to use virtual desktops see http://vdi.nci.org.au/help

To install the stable version of ARCCSSive:

vdi $ pip install –user ARCCSSive vdi $ export CMIP5_DB=sqlite:////g/data1/ua6/unofficial-ESG-replica/tmp/tree/cmip5_raijin_latest.db

or to install the current development version (note this uses a different database):

vdi $ pip install –user git+https://github.com/coecms/ARCCSSive.git vdi $ export CMIP5_DB=sqlite:////g/data1/ua6/unofficial-ESG-replica/tmp/tree/cmip5_raijin_latest.db

Once the library is installed run *ipython notebook* to start a new notebook

CHAPTER 3

CMIP5

The CMIP5 module provides tools for searching through the CMIP5 data stored on NCI's */g/data* filesystem

Getting Started:

The ARCCSSive library is available as a module on Raijin. Load it using:

```
module use ~access/modules
module load pythonlib/ARCCSSive
```

To use the CMIP5 catalog you first need to connect to it:

```
>>> from ARCCSSive import CMIP5
>>> cmip5 = CMIP5.connect()
```

The session object allows you to run queries on the catalog. There are a number of helper functions for common operations, for instance searching through the model outputs:

```
>>> outputs = cmip5.outputs(
...     experiment = 'rcp45',
...     variable   = 'tas',
...     mip        = 'day',
...     ensemble   = 'r1i1p1')
```

You can then loop over the search results in normal Python fashion:

```
>>> for o in outputs.filter_by(model='ACCESS1.3'):
...     (o.model, o.filenames())
('ACCESS1.3', ['tas_day_ACCESS1-3_rcp45_r1i1p1_20310101-20551231.nc'])
```

Examples

Get files from a single model variable

```
>>> outputs = cmip5.outputs(
...     experiment = 'rcp45',
...     variable   = 'tas',
...     mip        = 'day',
...     model      = 'ACCESS1.3',
...     ensemble   = 'r1i1p1')

>>> for f in outputs.first().filenames():
...     f
'tas_day_ACCESS1-3_rcp45_r1i1p1_20310101-20551231.nc'
```

Get files from all models for a specific variable

```
>>> outputs = cmip5.outputs(
...     experiment = 'rcp45',
...     variable   = 'tas',
...     mip        = 'day',
...     ensemble   = 'r1i1p1')

>>> for m in outputs:
...     model = m.model
...     files = m.filenames()
```

Choose more than one variable at a time

More complex queries on the *Session.outputs()* results can be performed using SQLalchemy's filter():

```
>>> from ARCCSSive.CMIP5.Model import *
>>> from sqlalchemy import *

>>> outputs = cmip5.outputs(
...     experiment = 'rcp45',
...     model      = 'ACCESS1-3',
...     mip        = 'Amon',) \
...     .filter(Instance.variable.in_(['tas','pr']))
```

Get results from a specific output version

Querying specific versions currently needs to go through the *Session.query()* function, this will be simplified in a future version of ARCCSSive:

```
>>> from ARCCSSive.CMIP5.Model import *

>>> res = cmip5.query(Version) \
...         .join(Instance) \
...         .filter(
...     Version.version     == 'v20120413',
```

```
...     Instance.model      == 'ACCESS1-3',
...     Instance.experiment == 'rcp45',
...     Instance.mip        == 'Amon',
...     Instance.ensemble   == 'r1i1p1')

>>> # This returns a sequence of Version, get the variable information from
>>> # the .variable property
>>> for o in res:
...     o.variable.model, o.variable.variable, o.filenames()
```

Compare model results between two experiments

Link two sets of outputs together using joins:

```
>>> from ARCCSSive.CMIP5.Model import *
>>> from sqlalchemy.orm import aliased
>>> from sqlalchemy import *

>>> # Create aliases for the historical and rcp variables, so we can
>>> # distinguish them in the query
>>> histInstance = aliased(Instance)
>>> rcpInstance  = aliased(Instance)
>>> rcp_hist  = cmip5.query(rcpInstance, histInstance).join(
...         histInstance, and_(
...             histInstance.variable == rcpInstance.variable,
...             histInstance.model    == rcpInstance.model,
...             histInstance.mip      == rcpInstance.mip,
...             histInstance.ensemble == rcpInstance.ensemble,
...         )).filter(
...             rcpInstance.experiment  == 'rcp45',
...             histInstance.experiment == 'historicalNat',
...         )

>>> for r, h in rcp_hist:
...     r.versions[-1].path, h.versions[-1].path
```

API

connect()

ARCCSSive.CMIP5.**connect**()

Connect to the CMIP5 catalog

Returns A new *Session*

Example:

```
>>> from ARCCSSive import CMIP5
>>> cmip5   = CMIP5.DB.connect()
>>> outputs = cmip5.query()
```

Session

The session object has a number of helper functions for getting information out of the catalog, e.g. *Session.models()* gets a list of all available models.

class ARCCSSive.CMIP5.**Session**
Holds a connection to the catalog

Create using *ARCCSSive.CMIP5.connect()*

experiments()
Get the list of all experiments in the dataset

Returns A list of strings

files(***kwargs*)
Query the list of files

Returns a list of files that match the arguments

Parameters **kwargs** – Match any attribute in *Model.Instance*, e.g. *model = 'ACCESS1-3'*

Returns An iterable returning Model.File matching the search query

mips()
Get the list of all MIP tables in the dataset

Returns A list of strings

models()
Get the list of all models in the dataset

Returns A list of strings

outputs(***kwargs*)
Get the most recent instances matching a query

Arguments are optional, using them will select only matching outputs

Parameters

- **variable** – CMIP variable name
- **experiment** – CMIP experiment
- **mip** – MIP table
- **model** – Model used to generate the dataset
- **ensemble** – Ensemble member

Returns An iterable sequence of *ARCCSSive.CMIP5.Model.Instance*

query(**args*, ***kwargs*)
Query the CMIP5 catalog

Allows you to filter the full list of CMIP5 outputs using SQLAlchemy commands

Returns A SQLalchemy query object

variables()
Get the list of all variables in the dataset

Returns A list of strings

Model

The model classes hold catalog information for a single entry. Each model run variable can have a number of different data versions, as errors get corrected by the publisher, and each version can consist of a number of files split into a time sequence.

Each model class has a number of relationships, which can be used in a query to efficiently return linked data e.g.:

```
>>> q = (cmip5.query(Instance, VersionFile)
...          .join(Instance.latest_version)
...          .join(Version.files))
```

This query returns an iterator of (*Instance*, *ARCCSSive.model.cmip5.File*) pairs and only needs to query the database once, whereas using a loop requires a database query for each iteration.

class ARCCSSive.CMIP5.Model.**Instance**(***kwargs*)

A combination of a CMIP5 Dataset and a single variable

Relationships:

versions
list[*Version*]: List of all available versions of this dataset

latest_version
Version: The most recent version of this dataset

files
list[*ARCCSSive.model.cmip5.File*]: All files belonging to this dataset and variable, regardless of version

Attributes:

variable
Variable name

experiment
CMIP experiment

mip
MIP table specifying output frequency and realm

model
Model that generated the dataset

ensemble
Ensemble member

realm
Realm: ie atmos, ocean

filenames()
Returns the file names from the latest version of this variable

Returns List of file names

drstree_path()
Returns the drstree path for this instance latest version

class ARCCSSive.CMIP5.Model.**Version**(***kwargs*)

A version of a model run's variable

Relationships:

variable
Instance: Dataset and variable this version is attached to

warnings
[*ARCCSSive.model.cmip5.Warning*]: Warnings attached to this dataset version

files
[*ARCCSSive.model.cmip5.File*]: Files belonging to this dataset version

Attributes:

version
Version identifier

path
Path to the output directory

```
>>> instance = cmip5.query(Instance).filter_by(dataset_id = 'c6d75f4c-793b-5bcc-
↪28ab-1af81e4b679d', variable='tas').one()
>>> version = instance.latest()
>>> version = instance.versions[-1]
```

glob()
Get the glob string matching the CMIP5 filename

```
>>> six.print_(version.glob())
tas_day_ACCESS1.3_rcp45_r1i1p1*.nc
```

build_filepaths()
Returns the list of files matching this version

Returns List of file names

```
>>> pprint.pprint(version.build_filepaths())
['/g/data1/ua6/unofficial-ESG-replica/tmp/tree/pcmdi9.llnl.gov/thredds/
↪fileServer/cmip5_css02_data/cmip5/output1/CSIRO-BOM/ACCESS1-3/rcp45/day/
↪atmos/day/r1i1p1/tas/1/tas_day_ACCESS1-3_rcp45_r1i1p1_20060101-20301231.nc',
 '/g/data1/ua6/unofficial-ESG-replica/tmp/tree/pcmdi9.llnl.gov/thredds/
↪fileServer/cmip5_css02_data/cmip5/output1/CSIRO-BOM/ACCESS1-3/rcp45/day/
↪atmos/day/r1i1p1/tas/1/tas_day_ACCESS1-3_rcp45_r1i1p1_20310101-20551231.nc',
 '/g/data1/ua6/unofficial-ESG-replica/tmp/tree/pcmdi9.llnl.gov/thredds/
↪fileServer/cmip5_css02_data/cmip5/output1/CSIRO-BOM/ACCESS1-3/rcp45/day/
↪atmos/day/r1i1p1/tas/1/tas_day_ACCESS1-3_rcp45_r1i1p1_20560101-20801231.nc',
 '/g/data1/ua6/unofficial-ESG-replica/tmp/tree/pcmdi9.llnl.gov/thredds/
↪fileServer/cmip5_css02_data/cmip5/output1/CSIRO-BOM/ACCESS1-3/rcp45/day/
↪atmos/day/r1i1p1/tas/1/tas_day_ACCESS1-3_rcp45_r1i1p1_20810101-21001231.nc']
```

filenames()
Returns the list of filenames for this version

Returns List of file names

```
>>> sorted(version.filenames())
['tas_day_ACCESS1-3_rcp45_r1i1p1_20060101-20301231.nc', 'tas_day_ACCESS1-3_
↪rcp45_r1i1p1_20310101-20551231.nc', 'tas_day_ACCESS1-3_rcp45_r1i1p1_
↪20560101-20801231.nc', 'tas_day_ACCESS1-3_rcp45_r1i1p1_20810101-21001231.nc
↪']
```

tracking_ids()
Returns the list of tracking_ids for files in this version

Returns List of tracking_ids

```
>>> sorted(version.tracking_ids())
['54779e2d-41fb-4671-bbdf-2170385afa3b', '800713b7-c303-4618-aef9-f72548d5ada6
↪', 'd2813685-9c7c-4527-8186-44a8f19d31dd', 'f810f58d-329e-4934-bb1c-
↪28c5c314e073']
```

drstree_path()
Returns the drstree path for this particular version

class ARCCSSive.model.cmip5.**File**(***kwargs*)
A CMIP5 output file's attributes

Relationships:

attribute:: dataset Dataset: The dataset this file is part of

attribute:: version *Version*: This file's dataset version

attribute:: warnings [Warning]: Warnings associated with this file

attribute:: timeseries Timeseries holding all files in the dataset with the same variables

Attributes:

attribute:: experiment_id attribute:: frequency attribute:: institute_id attribute:: model_id attribute:: modeling_realm attribute:: product attribute:: table_id attribute:: tracking_id attribute:: version_number attribute:: realization attribute:: initialization_method attribute:: physics_version

CHAPTER 4

CF-NetCDF Data

class ARCCSSive.model.cfnetcdf.**File**(***kwargs*)
A CF-compliant NetCDF file's attributes

attributes
dict: Full metadata

collection
str: Data collection this file belongs to

institution
str: Generating institution

open()
Open the file

path
str: Path to data file

source
str: Dataset source

title
str: File title

variables
list[*Variable*]: Component variables

class ARCCSSive.model.cfnetcdf.**Variable**(***kwargs*)
A CF-Compliant variable

aliases
list[str]: Aliases of this variable

amip
AMIP name

canonical_unit
Canonical unit

description
Description of the variable

files
list[*File*]: Files containing this variable

grib
Grib code

name
Variable standard name

class ARCCSSive.model.cfnetcdf.**VariableAlias**(**kwargs)

name
Alias name

variable
Variable: Variable this is an alias to

CHAPTER 5

CMIP5 Outputs

class ARCCSSive.model.cmip5.**Dataset**(***kwargs*)
A CMIP5 Dataset, as you'd find listed on ESGF

drstree_path()
Get the drs tree path to variables within this dataset

ensemble_member
str: Ensemble member

frequency
str: Data output frequency

institute_id
str: ID of the institute that ran the experiment

latest_version
The latest *Version* for this dataset

mip_table
str: MIP Table

model_id
str: ID of the model used

modeling_realm
str: Model component - atmos, land, ocean, etc.

variables
list[Timeseries]: The most recent versions of the variables in this dataset

versions
list[*Version*]: Available versions of this dataset, in release order

class ARCCSSive.model.cmip5.**Version**(***kwargs*)
A version of a ESGF dataset

Over time files within a dataset get updated, due to bug fixes and processing improvements. This results in multiple versions of files getting published to ESGF

dataset
Dataset: Dataset associated with this version

files
list[*File*]: Files belonging to this dataset version

is_latest
boolean: True if this is the latest version available

open()
Open all variables in the dataset

override
VersionOverride: Errata information for this version

version_number
str: Version number

warnings
list[*Warning*]: Warnings attached to the datset by users

class ARCCSSive.model.cmip5.**File**(***kwargs*)
A CMIP5 output file's attributes

Relationships:

attribute:: dataset *Dataset*: The dataset this file is part of

attribute:: version *Version*: This file's dataset version

attribute:: warnings [*Warning*]: Warnings associated with this file

attribute:: timeseries Timeseries holding all files in the dataset with the same variables

Attributes:

attribute:: experiment_id attribute:: frequency attribute:: institute_id attribute:: model_id attribute:: modeling_realm attribute:: product attribute:: table_id attribute:: tracking_id attribute:: version_number attribute:: realization attribute:: initialization_method attribute:: physics_version

open()
Open the file

CHAPTER 6

CMIP5 Errata

class ARCCSSive.model.cmip5.**VersionOverride**(***kwargs*)

Errata for a CMIP5 dataset version, for cases when the published version_id is unset or incorrect

Editing this table will automatically update the corresponding *Version*.

v = session.query(Version).first() v.override = VersionOverride(version_number='v20120101') session.add(v)

is_latest

boolean: True if this is the latest version available

version_number

str: New version number

class ARCCSSive.model.cmip5.**Warning**(***kwargs*)

added_by

str: Who added thge warning

added_on

str: Date the warning was added

warning

str: Warning text

CHAPTER 7

Administration

— Making a new release —

Use the Github interface to create a new relase with the version number, e.g. '1.2.3'. This should use semantic versioning, if it's a minor change increase the third number, if it introduces new features increase the second number and if it will break existing scripts using the library increase the first number.

After doing this the following will happen:

- Travis-ci will upload the package to PyPI
- CircleCI will upload the package to Anaconda
- The conda update cron job at NCI will pick up the new version overnight

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index

M

N

O

P

Q

R

S

T

V

W