

ERA5 monthly means of daily means retrieval efficiency

- How is the ERA5 monthly means of daily means (stream=moda) data organised in MARS?
- What would be the natural way to group requests?
- Web-API examples:
 - Example 1: A request for ERA5 monthly means of daily means (stream=moda) data, pressure levels

How is the ERA5 monthly means of daily means (stream=moda) data organised in MARS?

i In general it is organised, as a **huge tree**, with the indentation below, showing different levels down that tree:

- type of data: analysis
 - decade
 - type of level: model level
 - months, levels, parameters (all on the same tape file)★
 - type of level: potential temperature
 - months, levels, parameters (all on the same tape file)★
 - type of level: potential vorticity
 - months, levels, parameters (all on the same tape file)★
 - type of level: pressure level
 - months, levels, parameters (all on the same tape file)★
 - type of level: surface
 - months, parameters (all on the same tape file)★
- type of data: forecast
 - decade
 - type of level: model level
 - months, levels, parameters (all on the same tape file)★
 - type of level: pressure level
 - months, levels, parameters (all on the same tape file)★
 - type of level: surface
 - months, parameters (all on the same tape file)★

What would be the natural way to group requests?

i The idea is to request as much data as possible from the **same tape file** ★. The natural way to group requests would be: all parameters, all levels, all time-steps for all months of a decade.

! Note: 'all' means 'all' that the user wants. It doesn't have to be all available parameters.

Web-API examples:

Example 1: A request for ERA5 monthly means of daily means (stream=moda) data, pressure levels

- i**
- The objective of this example is to **demonstrate** how to iterate **efficiently** over some years, for a particular ERA5, monthly, pressure level request.
 - At this point you may wish to have a look on the [ERA5 data availability](#)
 - The request below can be used as a **starting point**, however you need to keep in mind that you have to adapt it to your needs eg to **set** the **keyword, values** according to your requirements (eg add or remove 'param', 'levtype', etc).
 - In this way you can **extend** this example to download a longer period or even the whole ERA5 dataset.
 - **!** Please note that at the moment **only decade 2010** is available.

```
#!/usr/bin/env python
import calendar
from ecmwfapi import ECMWFDataServer
server = ECMWFDataServer()

def retrieve_era5():
    """
    A function to demonstrate how to retrieve ERA5 monthly means data.
    Change the variables below to adapt to your needs.

    ERA5 monthly data is timestamped to the first of the month, hence dates
    have to be specified as a list in this format:
```

```
'19950101/19950201/19950301/.../20051101/20051201'.
```

Data is stored on one tape per decade, so for efficiency we split the date range into decades, hence we get multiple date lists by decade:

```
'19950101/19950201/19950301/.../19991101/19991201'  
'20000101/20000201/20000301/.../20051101/20051201'  
'20000101/20000201/20000301/.../20051101/20051201'
```

In the example below the output data are organised as one file per decade:

```
'era5_moda_1990'  
'era5_moda_2000'  
'era5_moda_2010'
```

Please note that at the moment only decade 2010 is available.

```
'''  
yearStart = 1995                # adjust to your requirements - as of 2017-07 only 2010-01-01  
onwards is available  
yearEnd = 2016                 # adjust to your requirements  
months = [1,2,3,4,5,6,7,8,9,10,11,12] # adjust to your requirements  
  
years = range(yearStart, yearEnd+1)  
print 'Years: ',years  
decades = list(set([divmod(i, 10)[0] for i in years]))  
decades = [x * 10 for x in decades]  
decades.sort()  
print 'Decades:', decades  
  
# loop through decades and create a month list  
for d in decades:  
    requestDates=''  
    for y in years:  
        if ((divmod(y,10)[0])*10) == d:  
            for m in months:  
                requestDates = requestDates+str(y)+(str(m)).zfill(2)+'01/'  
    requestDates = requestDates[:-1]  
    print 'Requesting dates: ', requestDates  
    target = 'era5_moda_%d.nc'% (d) # specifies the output file name  
    print 'Output file: ', target  
    era5_request(requestDates, d, target)  
  
# the actual data request  
def era5_request(requestDates, decade, target):  
    '''  
    Change the keywords below to adapt to your needs.  
    The easiest way to do this is:  
    1. go to http://apps.ecmwf.int/data-catalogues/era5/?class=ea  
    2. click through to the parameters you want, for any date  
    3. click 'View the MARS request'  
    '''  
    server.retrieve({  
        'class': 'ea',                # do not change  
        'dataset': 'era5',            # do not change  
        'expver': '1',                # do not change  
        'stream': 'moda',             # monthly means of daily means  
        'type': 'an',                 # analysis (versus forecast, fc)  
        'levtype': 'sfc',              # surface data (versus pressure level, pl, and model level, ml)  
        'param': '167.128/207.128',   # here: 2m temperature and 10m wind speed  
        "grid": "0.25/0.25",          # Optional for GRIB, required for NetCDF. The horizontal resolution  
in decimal degrees. If not set, the archived grid as specified in the data documentation is used.  
        'area': '75/-20/10/60',       # Optional. Subset (clip) to an area. Specify as N/W/S/E in  
Geographic lat/long degrees. Southern latitudes and western longitudes must be  
        # given as negative numbers. Requires "grid" to be set to a regular grid, e.  
g. "0.25/0.25".  
        'format': 'netcdf',           # Optional. Output in NetCDF format. Requires that you also specify  
'grid'. If not set, data is delivered in GRIB format, as archived.  
        'date': requestDates,         # dates, set automatically from above  
        'decade': decade,             # decade set automatically from above  
        'target': target              # output file name, set automatically from above  
    })  
if __name__ == '__main__':  
    retrieve_era5()
```

Merging files

The example script above creates one output file per decade of data. If you want a single output file:

For GRIB files, you can simply concatenate the files in Linux: `cat file1 file2 file3 > file4`

For NetCDF files (unsupported) see <https://code.mpimet.mpg.de/boards/1/topics/4446>

Useful links



- [ECMWF Web API Home](#)
- [What is ERA5](#)
- [How to access ERA5 data programmatically](#)
- [ERA5 data availability](#)
- [ERA5 data documentation](#)