IS-ENES2 Datanode Administrator's Reference Manual

Editor: Prashanth Dwarakanath, NSC, Sweden

Version: 2014-1.15
April 26, 2014

Contents

1 Purpose and limitations

The purpose of this document is to serve as an unambigious single resource for reference by administrators of IS-ENES2 ESGF datanodes, to configure their datanodes and publish data in compliance with regulations discussed and adopted by all datanode managers. This document aggregates information from sources such as the Trieste meeting notes [2], Martin Juckes' 'CORDEX: ESGF Search Facet Mappings' document [1] and other discussions which have led to collective consensus. This document only contains information from the perspective of publishing/maintaining data on the ESGF datanode and may not be refered to for any other purpose.

2 Latest Version

The latest version of this document will always be available at:
https://github.com/snic-nsc/datanode-mgr-doc/raw/master/ro/Datanodemgr-doc.pdf
The entire repository, which includes the LaTeX source file can be cloned from:
https://github.com/snic-nsc/datanode-mgr-doc.git

3 IS-ENES2 ESGF datanode Search Facet Configuration

IS-ENES2 ESGF datanodes have some additional search facets pertaining to CORDEX. Here below are the entire list of facets used, on an IS-ENES2 ESGF datanode. This file is available in the '**configfiles**' directory in the repo.
Filename: `facets.properties`
Standard location: `/esg/config/facets.properties`

```
project=0:Project:optional_project_description
institute=1:Institute:optional_institute_description
model=2:Model:optional_model_description
source_id=3:Instrument:optional_instrument_description
experiment_family=4:Experiment Family:optional_experiment_family_description
experiment=5:Experiment:optional_experiment_description
time_frequency=6:Time Frequency:optional_time_frequency_description
product=7:Product:optional_product_description
realm=8:Realm:optional_realm_description
variable=9:Variable:optional_variable_description
variable_long_name=10:Variable Long Name:optional_variable_long_name_description
cmor_table=11:CMIP Table:optional_cmor_table_description
cf_standard_name=12:CF Standard Name:optional_cf_standard_name_description
ensemble=13:Ensemble:optional_ensemble_description
domain=14:Domain:optional_domain_description
driving_model=15:Driving Model:optional_driving_model_description
rcm_version=16:Downscaling realisation:optional_ds_description
data_node=17:Data Node:optional_data_node_description
```

Contents

4 ESGF Attribute Services

File name: `/esg/config/esgf_ats_static.xml`
For information about how to setup your datanode to correctly enforce restrictions on CORDEX data usage, refer to Section 14. This file is available in the '**configfiles**' directory in the repo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- File that may contain custom attribute service and registration endpoints
     in addition to those contained in the file esgf_ats.xml.
     This file is supposed to be maintained by the local system administrators,
     while the file esgf_ats.xml is dynamically generated by the node manager. -->
<ats_whitelist xmlns="http://www.esgf.org/whitelist">

    <!-- pcmdi9 Attribute and Registration services: it is included here to allow registration for "esgf-test" nodes,
         otherwise it should be contained in file esgf_ats.xml for nodes in the "esgf-prod" group -->
    <attribute type="CMIP5 Research"
               attributeService="https://pcmdi9.llnl.gov/esgf-idp/saml/soap/secure/attributeService.htm"
               description="Users of CMIP5 data for non-commercial research purposes only"
               registrationService="https://pcmdi9.llnl.gov/esgf-idp/secure/registrationService.htm"/>
    <attribute type="CMIP5 Commercial"
               attributeService="https://pcmdi9.llnl.gov/esgf-idp/saml/soap/secure/attributeService.htm"
               description="Users of CMIP5 data for commercial purposes"
               registrationService="https://pcmdi9.llnl.gov/esgf-idp/secure/registrationService.htm"/>

    <attribute type="CORDEX_Commercial"
                attributeService="https://esg-dn1.nsc.liu.se/esgf-idp/saml/soap/secure/attributeService.htm"
                description="User group for possible commercial users of CORDEX data"
                registrationService="https://esg-dn1.nsc.liu.se/esgf-idp/secure/registrationService.htm"/>
    <attribute type="CORDEX_Research"
                attributeService="https://esg-dn1.nsc.liu.se/esgf-idp/saml/soap/secure/attributeService.htm"
                description="User group only for non-commercial users of CORDEX data"
                registrationService="https://esg-dn1.nsc.liu.se/esgf-idp/secure/registrationService.htm"/>
</ats_whitelist>
```

5 ESGF IDP Whitelist settings

File name: `/esg/config/esgf_idp_static.xml`. This file is available in the '**configfiles**' directory in the repo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<idp_whitelist xmlns="http://www.esgf.org/whitelist">

    <value>https://adm07.cmcc.it/esgf-idp/idp/openidServer.htm</value>
    <value>https://ceda.ac.uk/openid/Provider/server</value>
    <value>https://ceda.ac.uk/OpenID/Provider/server</value>
    <value>https://cordexesg.dmi.dk/esgf-idp/idp/openidServer.htm</value>
    <value>https://dev.esg.anl.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg01.nersc.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg2.e-inis.ie/esgf-idp/openid/openidServer.htm</value>
    <value>https://esg2.nci.org.au/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg.bnu.edu.cn/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg.ccs.ornl.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg.pik-potsdam.de/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg-datanode.jpl.nasa.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://esg-dn1.nsc.liu.se/esgf-idp/idp/openidServer.htm</value>
    <value>https://esgf-data.dkrz.de/esgf-idp/idp/openidServer.htm</value>
    <value>https://esgf.nccs.nasa.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://esgf-node.ipsl.fr/esgf-idp/idp/openidServer.htm</value>
    <value>https://euclipse1.dkrz.de/esgf-idp/idp/openidServer.htm</value>
```

Contents

```
    <value>https://hydra.fsl.noaa.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://noresg.norstore.uio.no/esgf-idp/idp/openidServer.htm</value>
    <value>https://pcmdi9.llnl.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://pcmdi11.llnl.gov/esgf-idp/idp/openidServer.htm</value>
    <value>https://www.earthsystemgrid.org/openid/provider.htm</value>

</idp_whitelist>
```

6 ESGF Search Shard configuration settings

File name: `/esg/config/esgf_shards_static.xml`. This file is available in the '**configfiles**' directory in the repo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<shards xmlns="http://www.esgf.org/whitelist">
    <value>localhost:8983/solr</value>
    <!-- US -->
    <value>pcmdi9.llnl.gov:8983/solr</value>
    <value>pcmdi11.llnl.gov:8983/solr</value>
    <value>esg-datanode.jpl.nasa.gov:8983/solr</value>
    <value>esg.ccs.ornl.gov:8983/solr</value>
    <value>esgf.nccs.nasa.gov:8983/solr</value>
    <value>esg01.nersc.gov:8983/solr</value>
    <value>esgdata.gfdl.noaa.gov:8983/solr</value>
    <value>hydra.fsl.noaa.gov:8983/solr</value>

    <!-- Europe -->
    <value>esgf-index1.ceda.ac.uk:8983/solr</value>
    <value>esgf-data.dkrz.de:8983/solr</value>
    <value>esg-dn1.nsc.liu.se:8983/solr</value>
    <value>adm07.cmcc.it:8983/solr</value>
    <value>esgf-node.ipsl.fr:8983/solr</value>
    <value>noresg.norstore.uio.no:8983/solr</value>
    <value>cordexesg.dmi.dk:8983/solr</value>
    <value>esg.pik-potsdam.de:8983/solr</value>
    <value>esg2.e-inis.ie:8983/solr</value>

    <!-- Australia -->
    <value>esg2.nci.org.au:8983/solr</value>

    <!-- Asia -->
   <value>esg.bnu.edu.cn:8983/solr</value>
</shards>
```

7 Publication Version

It was decided at the Trieste meet that all data published on IS-ENES2 datanodes will clearly specify the version number which is the date of the publication, expressed in the format **v***yyyymmdd*. This requires the creation of directory with that name, in the physical directory structure. This directory has to be created after the 'Variable name' directory. Examples:

`/datapool1/cordexgeneral/cordex/output/MNA-22/SMHI/ECMWF-ERAINT/evaluation/r0i0p0/SMHI-RCA4/v1/fx/orog/v20131101`

`/datapool1/cordexgeneral/cordex/output/ARC-44/SMHI/NCC-NorESM1-M/historical/r0i0p0/SMHI-RCA4/v1/fx/sftlf/v20140123`

To get this version number correctly, the procedure is to append a `--new-version <versionnum>` to the `esgpublish` command.

Contents

8 Directory Structure

The path to the directory tree containing the data shall have `Project/Product` followed by the directory tree containing the data.
Given below are examples of valid and invalid directory structures.

`/cordex/output/...` ✓
`/localfs/localpath/cordex/output/...` ✓[1,2]
`/corddata/output/...` ✗ //non-standard name corresponding to 'Project'.
`/cordex/AFR-44/...` ✗ //there is no directory corresponding to 'Product'. Here is a complete `directory_format` line, for reference:

```
directory_format = /localpath/cordex/%(product)s/%(domain)s/%(institute)s/\
%(driving_model)s/%(experiment)s/%(ensemble)s/%(rcm_model)s/%(rcm_version)s/\
%(time_frequency)s/%(variable)s/v%(version)s
```

9 Variables to be excluded during publish: CORDEX

The following declaration inside `/esg/esgcet/esg.ini` should be used to exclude certain variables from the THREDDS catalogues generated by `esgpublish`. Note that this differs from the default value created by previous versions of `esgsetup`; in particular managers should ensure that the variable `basin` is NOT excluded.

```
thredds_exclude_variables = a,a_bnds,alev1,alevel,alevhalf,alt40,b,b_bnds,bnds,\
bounds_lat,bounds_lon,dbze,depth,depth0m,depth100m,depth_bnds,geo_region,height,\
height10m,height2m,heightv,Lambert_Conformal,lat,lat_bnds,lat_bounds,\
lat_vertices,latitude,latitude_bnds,layer,lev,lev_bnds,location,lon,lon_bnds,\
lon_bounds,lon_vertices,longitude,longitude_bnds,olayer100m,olevel,oline,p0,\
p220,p500,p560,p700,p840,plev,plev3,plev7,plev8,plev_bnds,plevs,pressure1,region,\
rho,rlat,rlat_bnds,rlon,rlon_bnds,rotated_pole,Rotated_Pole,scatratio,sdepth,\
sdepth1,sza5,tau,tau_bnds,time,time1,time2,time_bnds,vegtype,x,y
```

10 Checking for variables that need to be skipped

We saw in Section 9 the compiled list of variables that need to be present in the `thredds_exclude_variables` list, prior to data publication. However, there might well be variables in your data which need to be similarly excluded, but are not part of this list yet. This has caused problems in the past. Here's a simple script that can be used to inspect the data prior to publication. It reports variables which **may** need to be added to the exclude list. It also logs potential problems. The script uses the `ncdump` utility which is shipped with **uvcdat/cdat** on ESGF datanodes. This script is available in the '**scripts**' directory in the repo.

[1] Some sites use the lower-case 'cordex' while some use 'CORDEX'; While there is no rule, the lower-case 'cordex' may be considered as the prefered option.

[2] 'output' is the value of the 'Product' facet option here. It may take other values that are applicable to the 'Product' facet in the future.

```
#!/bin/bash
scripthome=/root/scripts
ncdumplocation=/middleware/uvcdat/1.4.0/bin
outputfile=/tmp/ncchecks/exclvars
tmpfileone=/tmp/ncchecks/tmpexclvars1
tmpfiletwo=/tmp/ncchecks/tmpexclvars2
notfounds=/tmp/ncchecks/notfound
suspects=/tmp/ncchecks/suspects
excllist=$scripthome/excl_cordex

if [ $# -ne 1 ]; then
    echo "bailing out";
    exit -1;
fi
ncfile=$1;
mkdir -p /tmp/ncchecks;

if [ ! -e $tmpfileone ]; then
    touch $tmpfileone;
fi
if [ ! -e $outputfile ]; then
    touch $outputfile;
fi
if [ ! -e /tmp/ncchecks/sessionfilecount ]; then
    echo "1" >/tmp/ncchecks/sessionfilecount;
fi
fcount=`cat /tmp/ncchecks/sessionfilecount`;
echo "$fcount";
fcount=`expr $fcount + 1`;
echo $fcount >/tmp/ncchecks/sessionfilecount

varname=`basename $ncfile|cut -d '_' -f1`;
#assumes that your file is named as <var>_something.nc
$ncdumplocation/ncdump -h $ncfile|sed -n "/variables/,/global attributes/p"\
|grep -v ':'|awk 'NF'|cut -d ' ' -f2|cut -d '(' -f1 > $tmpfileone;
grep $varname $tmpfileone >/dev/null;
if [ $? -ne 0 ]; then
    echo "Could not find var $varname in file $ncfile";
    echo "Could not find var $varname in file $ncfile" >>$notfounds;
fi
cat $tmpfileone|grep -v "$varname" >$tmpfiletwo
if [ -s $tmpfiletwo ]; then
    while read ln; do
        grep $ln $outputfile >/dev/null;
        if [ $? -ne 0 ]; then
            echo $ln >>$outputfile;
        fi
        grep $ln $excllist >/dev/null;
        if [ $? -ne 0 ]; then
            echo "line in question: $ln"
            echo "Found variable $ln in file $ncfile";
            echo "Found variable $ln in file $ncfile" >>$suspects;
        fi

    done <$tmpfiletwo;
fi
```

How to use

1. Ensure that you set the correct path to the variable 'ncdumplocation'.
2. Choose a location for 'scripthome'. Copy the `exclvarcheck.sh` and `excl_cordex`[3] files to that location.

[3]available in the '**scripts**' directory in the repo

3. If you have run the script before, remember to clear the files in `/tmp/ncchecks`
4. To run, simply **cd** to the directory where your datafiles reside and run:

   ```
   find . -name '*.nc' -exec bash <script home>/exclvarcheck.sh {} \;
   ```

5. When it completes, inspect the file `/tmp/ncchecks/exclvars` to see if you might need to add any variables.
6. The file `/tmp/ncchecks/suspects` lists the files which produced the additions.
7. The file `/tmp/ncchecks/notfound` will list cases, if any, where a variable after which the file is named, isn't present in the file itself!
8. If you indeed find variables that need to be added to the list of excluded variables, please let me know

11 Value for the 'Model' facet

It was decided that the value of the 'Model' facet should NOT contain the institute information, as this information is already captured and presented by the 'Institute' facet. However, the directory corresponding to the 'Model' contains the name of the institute too, along with the model name, as stipulated by the CORDEX archive specifications [4]. This results in the requirement for some special handling.

The easiest way to handle this is by creating a substitution map for the variable.

1. Under the options for `[project:cordex]`, find the configuration line that says `maps`
2. Edit the line to say the following:
   `maps = model_map,institute_map, las_time_delta_map, domain_map`
3. Create a new map 'model_map' and populate it with entries that correspond to the models that you handle, leaving out the institute part in the last field.
4. Look at the example below for reference:

   ```
   model_map = map(project,rcm_model : model)
       cordex |SMHI-RCA4| RCA4
       cordex |SMHI-RCA4-SN| RCA4-SN
   ```

5. Use the regex 'model' in the place of the directory corresponding to the model directory, in the `dataset_id` string.

   ```
   dataset_id = cordex.%(product)s.%(domain)s.%(institute)s.%(driving_model)s.\
   %(experiment)s.%(ensemble)s.%(model)s.%(rcm_version)s.%(time_frequency)s.\
   %(variable)s
   ```

[4] "RCMModelName is an alphanumeric identifier chosen by the modeling group; it should consist of an institute acronym and a model acronym, connected by a dash, e.g., DMI-HIRHAM5 or SMHI-RCA3."[3]

11.1 Complete model_map

The current and comprehensive list of CORDEX models may be obtained from: http://cordex.dmi.dk/joomla/images/CORDEX/RCMModelName.txt.
Given below is a script that can generate a complete model_map table, that could then be pasted into the ini file. This script is available in the '**scripts**' directory in the repo. Also produced here is the full output of the script.

```
#!/bin/bash
outp='curl -s 'http://cordex.dmi.dk/joomla/images/CORDEX/RCMModelName.txt'|grep -v '#'|grep 'confirmed'|tr -s ' ''';
echo "model_map = map(project,rcm_model : model)"
echo "$outp" |while read ln; do
        inst='echo $ln|cut -d' ' -f2';
        res='echo $ln| sed "s/$inst-\(.*\) \(.*\) confirmed/\tcordex| $inst-\1| \1/"'
        echo -e "$res"
done
```

```
model_map = map(project,rcm_model : model)
        cordex| AUTH-LHTEE-WRF321B| WRF321B
        cordex| AUTH-Met-WRF331A| WRF331A
        cordex| AWI-HIRHAM5| HIRHAM5
        cordex| BCCR-WRF331C| WRF331C
        cordex| CCCma-CanRCM4| CanRCM4
        cordex| CHMI-ALADIN52| ALADIN52
        cordex| CLMcom-CCLM4-8-17| CCLM4-8-17
        cordex| CNRM-ALADIN52| ALADIN52
        cordex| CNRM-ARPEGE52| ARPEGE52
        cordex| CRP-GL-WRF331A| WRF331A
        cordex| CUNI-RegCM4-2| RegCM4-2
        cordex| DHMZ-RegCM4-2| RegCM4-2
        cordex| DMI-HIRHAM5| HIRHAM5
        cordex| ENEA-RegCM4-3| RegCM4-3
        cordex| HMS-ALADIN52| ALADIN52
        cordex| ICTP-RegCM4-3| RegCM4-3
        cordex| IDL-WRF331D| WRF331D
        cordex| IPSL-INERIS-WRF331F| WRF331F
        cordex| KNMI-RACMO21P| RACMO21P
        cordex| KNMI-RACMO22T| RACMO22T
        cordex| MIUB-WRF331A| WRF331A
        cordex| MOHC-HadGEM3-RA| HadGEM3-RA
        cordex| MOHC-HadRM3P| HadRM3P
        cordex| MPI-CSC-REMO2009| REMO2009
        cordex| NUIM-WRF331F| WRF331F
        cordex| SMHI-RCA4| RCA4
        cordex| SMHI-RCA4-SN| RCA4-SN
        cordex| SMHI-RCAO| RCAO
        cordex| SMHI-RCAO-SN| RCAO-SN
        cordex| UCAN-WRF331G| WRF331G
        cordex| UCAN-WRF350I| WRF350I
        cordex| UCLM-PROMES| PROMES
        cordex| UHOH-WRF331H| WRF331H
        cordex| UQAM-CRCM5| CRCM5
```

12 esgcet_models_table.txt

Apart from the model map, another map that lists models and their parent organizations is the /esg/config/esgcet_models_table.txt. After making any changes to it, one needs to execute esginitialize -c, to update it, and if that doesn't work, you may need to 'downgrade' the database by excuting esginitialize --d0 and then executing esginitialize -c.

```
  test | test | http://www-pcmdi.llnl.gov | Test
```

Contents

```
test | ncar_ccsm3_0 | http://www.ccsm.ucar.edu| NCAR Community Climate System Model, CCSM 3.0
cordex | RCA4 | SMHI | www.smhi.se
cordex | RCA4-SN | SMHI | www.smhi.se
cordex | RCAO | SMHI | www.smhi.se
cordex | RCAO-SN | SMHI | www.smhi.se
```

13 Displaying the project name in upper case

Though the project name is always expressed in lower case in catalogs and metadata, it is displayed in the upper-case in the web frontend. This requires setting a simple substitution string. Simply add the name of the project, first in lower case and then in upper case, separated by a colon. The file into which this string goes in is:

```
/usr/local/tomcat/webapps/esg-search/WEB-INF/classes/esg/search/config/projects.properties
cmip5:CMIP5
obs4mips:obs4MIPs
cssef:CSSEF
tamip:TAMIP
lucid:LUCID
test:TEST
pmip3:PMIP3
geomip:GeoMIP
euclipse:EUCLIPSE
cordex:CORDEX
```

14 Enforcing group restrictions on CORDEX data

CORDEX data published on the ESGF datanodes in the federation are made available only to those who apply for membership to one of the two groups associated with CORDEX data. These groups, apart from restricting who can access these datasets can also serve as a mechanism to specify additional terms of data access. The `CORDEX_RESEARCH` group is for individuals who wish to download and use the data only for non-commercial purposes whereas `CORDEX_COMMERCIAL` is for those individuals who may wish to use the data for commercial purposes. CORDEX data which is open for unrestricted use is made available to both groups whereas data which is meant to be only used for non-commercial use is only made accessible to members of the `CORDEX_RESEARCH` group. Unless otherwise specified by the data-provider, all CORDEX datasets should be accessible by members of both `CORDEX_RESEARCH` and `CORDEX_COMMERCIAL` groups. Attribute management for these CORDEX groups is managed on the `esg-dn1.nsc.liu.se` datanode and for configuring your datanode to use this attribute service, refer to Section 4.

14.1 Ensure presence of license files

If you are running the latest version of the middleware (1.6.x), you may skip to Section 14.2. If you are running an older release, check whether the following files are present, on your datanode:

1. $tomcatdir/webapps/esg-orp/licenses/CordexResearchLicense.xml
2. $tomcatdir/webapps/esg-orp/licenses/CordexCommercialLicense.xml

If the above listed files are NOT present:

1. git clone the repository containing this document, along with the license files from https://github.com/snic-nsc/datanode-mgr-doc.git
2. Copy the license files present in the `cordexlicensefiles` directory over to their respective target locations on the datanode(specified in file 'filelocations', also in the same directory).
3. Ensure that you replace the default 'registration-request.jsp' file with the one present in the `cordexlicensefiles` directory, as this file activates the usage of the CORDEX license files.
4. Restart `esg-node`

14.2 Segregating data

The ESGF attribute service can be used to restrict access to data by creating different policies for different file paths. This means that data with different levels of access restrictions ought to be in distinct directory heirarchies. This needs some conscious planning by datanode managers, preferably prior to data publication, as it may be inconvenient to move data directories later. Planning is required to setup unambigious and intuitive directory trees which will then have different restriction policies applied on them. For the purpose of reducing publication time confusions and or possibility of errors, it is strongly recommended to set up entirely seperate directory trees, rather than having a mix of the two types under the same tree, that is, under distinct `thredds_dataset_roots`.

14.3 Caveat

Unlike most commercial scenarios where a paying or 'commercial' customer gets additional features/privileges, in the CORDEX sense, a commercial user is one who has fewer datasets he/she can possibly access; This is because datasets which are meant for non-commercial access would not be available for these users. What this means is that naming a top-level directory/dataset_root as `Commercial` or similar, would be counter-intuitive as it would be available for all users. It is however beneficial to create a directory/dataset_root called `Non-Commercial`, as this would clearly indicate that it's only for non-commercial use, that is, it's only available for users belonging to the `CORDEX_RESEARCH` group.

14.4 Paths and regexes

The ESGF attribute service sees paths as presented to it by thredds. You can use that to design the regex that you need. Ensure that you don't design a regex which gets triggered by

unintended elements in the path, including the hostname of the node itself! While configuring the attribute service on the DMI datanode, the hostname of the node, `cordexesg.dmi.dk` was triggering the regex match for the expression `.*cordex.*` causing every url to match!!

14.5 Setting up the `esgf_policies_local.xml`

Let's consider the following configuration lines:

```
<policy resource=".*fileServer.*cordexnoncommercial.*" attribute_type="CORDEX_Research" attribute_value="user" action="Read"/>
<policy resource=".*fileServer.*cordexgeneral.*" attribute_type="CORDEX_Research" attribute_value="user" action="Read"/>
<policy resource=".*fileServer.*cordexgeneral.*" attribute_type="CORDEX_Commercial" attribute_value="user" action="Read"/>
<policy resource=".*fileServer.*cord.*" attribute_type="wheel" attribute_value="super" action="Write"/>
```

These lines indicate that thredds urls containing the element `cordexnoncommercial` are only accessible to members of `CORDEX_RESEARCH` group whereas urls containing `cordexgeneral` are accessible by all CORDEX data users. We can also see that `Write` or `Publish` access is only provided to users of group `wheel` with attribute `super`. This would allow the special user account `rootAdmin` to be used for all publication activities.

14.6 Corresponding `thredds_dataset_roots` entries

The `thredds_dataset_roots` entries can be set up in many ways. Let's consider two cases.

1. Both non-commercial and general data being under the same dataset_root:

   ```
   thredds_dataset_roots =
   esg_dataroot1| /data
   ```

   Here, the non-commercial data would be placed under `/data/cordexnoncommercial` whereas the general data would be under `/data/cordexgeneral`.

2. Non-commercial and general data being under different dataset_roots:

   ```
   thredds_dataset_roots =
   esg_cordexnoncommercial| /dir1/cordex
   esg_cordexgeneral| /dir2/cordex
   ```

Caution!! The part of the path specified as the `thredds_dataset_root` would be subsituted by the name associated with the dataset_root in the thredds filename. This means that if your `thredds_dataset_root` value reads thus: `esg_data| /partion1/noncommercial`, the '`partition1/noncommercial`' part of the path will be substituted by `esg_data` in the thredds url and hence would not match the regex you'd planned to capture 'noncommercial'. It is therefore preferred to simply use the name of the `thredds_dataset_root` as the regex match.

14.7 Data restricted to 'Non-Commercial usage only', by site

Sl	Site	Data
1.	BADC	None
2.	DKRZ	CLMcom data
3.	DMI	HMS data
4.	IPSL	None
5.	LIU-NSC	None
6.	UIO	None
7.	UNICAN	All

Table 0.2: Data restricted to 'non-commercial usage only', by site

15 Enabling Gridftp and OPeNDAP access

In order to enable OPeNDAP access, simply ensure that the following lines are present in your `esg.ini` file:

```
thredds_file_services =
 HTTPServer | /thredds/fileServer/ | HTTPServer | fileservice
 GridFTP | gsiftp://<nodename>:2811/ | GRIDFTP | fileservice
 OpenDAP | /thredds/dodsC/ | OpenDAP | fileservice
```

To allow incoming OpenDAP requests, you should also ensure that access is provided to CORDEX group members. It's done by putting these lines in the `/esg/config/esgf_policies_local.xml` file:

```
<policy resource=".*dodsC.*cordex.*" attribute_type="CORDEX_Research" attribute_value="user" action="Read"/>
<policy resource=".*dodsC.*cordex.*" attribute_type="CORDEX_Commercial" attribute_value="user" action="Read"/>
<policy resource=".*cordex.*aggregation.*" attribute_type="CORDEX_Research" attribute_value="user" action="Read"/>
<policy resource=".*cordex.*aggregation.*" attribute_type="CORDEX_Commercial" attribute_value="user" action="Read"/>
```

If you intend to offer gridftp, don't forget to allow inbound access to TCP port 2811

16 Handy pre-publish tips

16.1 Adding checksums

It's a good practice to publish data along with their checksums, so that silent corruption or download errors don't get away unnoticed. Generating checksums at publish time may drastically slow down a publication, so it's advisable to precompute them and then insert them into the mapfile. There are several ways in which one could generate checksums. I prefer to use gnu parallel to speed things up.

```
time(find . -name '*.nc' -print| parallel sha256sum) >/unsorted-sha256sums \
2>checksumout;
cat unsorted-sha256sums|sort -k 2,2 >sha256sums;
```

Generate the map file using the standard `esgscan_directory` and then use the following python script to insert the checksums. This script is available in the '**scripts**' directory in the repo.

```
#!/usr/bin/python

import os,sys
mydict=dict()
try:
    checksumfile=sys.argv[1]
    inpfile=sys.argv[2]
    outfile=sys.argv[3]
    fp=open(checksumfile,'r')
    fp2=open(inpfile,'r')
    fp3=open(outfile,'w')
    cklines=fp.readlines()
    fp.close()
    for line in cklines:
        lna=line.split(' ')
        cksum=lna[0]
        ffn=lna[len(lna)-1]
        ffna=ffn.split('/')
        fn=ffna[len(ffna)-1].split('\n')[0]
        mydict[fn]=cksum

    inplines=fp2.readlines()
    fp2.close()
    for line in inplines:
        ffn=line.split(' ')[2]
        ffna=ffn.split('/')
        fn=ffna[len(ffna)-1]
        cksum=mydict[fn]
        mystr=line.split('\n')[0]
        mystr=mystr+' | checksum='+cksum+' | checksum_type=sha256\n'
        fp3.write(mystr)
except:
    print "Error accessing file or with finding precomputed checksum"
    exit()
fp3.close()
```

Simply call the script like this:

```
python addchecksum.py <checksumfile> <mapfile> <outputfile>
```

17 Acknowledgments

Many people have contributed to this document, pointing out errors and suggesting improvements. Thanks in particular to Hans Ramthun, Katharina Berger and Stephanie Legutke of the DKRZ, Stephen Pascoe of the BADC, and Jose Carlos Blanco of UNICAN, for their suggestions and help. Together, we strive to make the task of datanode administration a bit less of a hopeless task!

Changelog

Version 2014-1.15

1. Added info about lines to be added to `esgf_policies_local.xml` file, to allow OpenDAP access. See Section 15 for details.

2. Noted DMI's 'non-commercial usage only' dataset.

Version 2014-1.14

1. Script to check for skipped variables in exclude list.

2. Added variables **Rotated_Pole** and **heightv** to the `thredds_exclude_variables` list.

3. All config files and scripts now added to the repo itself, eliminating the need to copy/paste.

4. Updated 'non-commercial usage only' data table.

5. Added 'Changelog' appendix to this document.

References

[1] Martin Juckes. CORDEX: ESGF Search Facet Mappings. URL: https://github.com/snic-nsc/datanode-mgr-doc/raw/master/ro/cordexSearchFacets_v5_20140210.doc (visited on 01032014).

[2] Martin Juckes. Trieste Meeting Notes. URL: https://docs.google.com/document/d/1rRXn4pyOlb95K9mYqxpIdJaxDf-xoZrIlc9XlWxn4HA (visited on 01032014).

[3] O.B. Christensen et al. CORDEX Archive Design. URL: https://verc.enes.org/data/projects/documents/cordex-archive-design (visited on 01032014).