



Collaborative Climate Community Data and Processing Grid (C3-Grid)

CDO or NCO applications on selected data

Work Package:	cross-WP - WG Workflows
Author:	M. Stockhause
Version:	V0.2/2008
Publishing Date:	05/2008
Pages:	5
WP Coordination:	-
Partners:	MPI-M, IFM-GEOMAR, FUB, ZAIK, DLR, AWI, PIK
Contact:	M. Stockhause
Email:	martina.stockhauseATzmaw.de



Content

1	Functionality	3
1.1	Overview.....	3
1.2	Usage of cdoexe.....	3
1.3	Input Data	4
1.4	Outlook	4
2	Installation at CP site.....	4
2.1	Scripts.....	4
2.2	Tools.....	4
3	References	5



1 Functionality

This little python script provides a flexible tool for a user defined cdo or nco call; principally, shell commands or other executables can be accessed as well. Multiple tool calls, i.e. a complex workflow application, is possible by several accesses of this script.

1.1 Overview

The tool `cdoexe.py` provides a low level tool for scientists to perform additional processings on selected data [1], e.g. comparisons or ensemble analyses. It can use tools from the packages `cdo` or `nco` as well as own executables or scripts which fulfil the below specified interface.

Therefore the user needs a certain level of knowledge of the data he or she processes as well as of the chosen tool.

Along with the selected data a rough metadata file in the C3Grid profile of ISO 19115 / 19139 format is prepared [2]. Since the processing is user specific, adaptations have to be made afterwards by the user. The script uses the agreed environment variables for the C3Grid workspace [3].

1.2 Usage of `cdoexe`

usage: <code>python cdoexe.py -i <inputfiles> -l <logfile> -m <metainfiles> -c <cdo/nco calls> -f <outfile></code>

- c cdo/nco/other commands enclosed by “, therein separated by “;”
\$1,\$2,... **can** be used for input files (list position 1,2,... of infiles)
EXAMPLE: `-c '\cdo -setcode,260 -add -selcode,142 $1 -selcode,143 $1 tmp1.grb;cdo -setcode,260 -add -selcode,142 $2 -selcode,143 $2 tmp2.grb;cdo -mergetime tmp1.grb tmp2.grb'`
- i absolute path to an input file; multiple options possible; position dependence!
- l absolute path to the logfile
- m absolute path to a metadata file; multiple options possible; position dependence (same order than of input files required)!
- f absolute path to the output file

Examples:

- **Using explicit names for input files:**
`python cdoexe.py -l /path/in cdo1.log -i /path/in/edu_in1.grb -m /path/in/edu_in1.xml
-i /path/in/edu_in2.grb -m /path/in/edu_in2.xml -f cdo_out1.grb
-c 'cdo -mergetime -selcode,182 edu_in1.grb -selcode,182 edu_in2.grb'`
- **Using \$1,\$2 for input files:**
`python cdoexe.py -l /path/in cdo1.log -i /path/in/edu_in1.grb -m /path/in/edu_in1.xml
-i /path/in/edu_in2.grb -m /path/in/edu_in2.xml -f cdo_out1.grb
-c 'cdo -mergetime -selcode,182 $1 -selcode,182 $2'`



The script `cdoexe.py` analyses the command line, and replaces the input (`$1`, `$2`, ...) flags by the names of the input parameters, if necessary. Finally, it appends the outputfile name given by the `-f` option. The specified command is executed.

Advices:

- Make sure that your specified command is valid before execution.
- Piping of `cdo` commands is better for the performance than the definition of multiple workflow tasks (calls of `cdoexe.py`)
- Adapt the delivered metadata file to your performed processing.

1.3 Input Data

The tool `cdoexe` can handle every kind of data, preferably `grb` for `cdo` calls and only `nc` for `nco` calls. It was especially created to run on C3Grid selected data (output of interface D – data request; [5]).

1.4 Outlook

The `cdoexe` script can be potentially used as a wrapper for user-defined scripts (written and uploaded by the user) on user defined data (uploaded by the user).

2 Installation at CP site

2.1 Scripts

There are two scripts needed that have to be placed in the same directory:

- `cdoexe.py`: execution script
- `meta_cdolSO.xsl`: metadata update script called by `cdoexe.py`,

and for error handling additionally:

- `printParam.py`
- `warn.lst` and `error.lst`.

The python version should be at 2.3 or newer plus an xml handling module: `lxml` oder `xml.etree.ElementTree`. The later is part of python since version 2.5.

2.2 Tools

The following tools have to be installed for `cdo/nco` applications:

- `cdo` (climate data operators - <http://www.mpimet.mpg.de/~cdo>)
- `nco` (climate data operators - <http://nco.sourceforge.net/>)
- `xmlstarlet`

These tools can be either installed in a directory in the search path or loaded by modules during execution from `os.environ['MODULESHOME']+'/init/python'`.



3 References

- [1] M. Stockhause:
Data Preparation (Preprocessing for Model Data) – Version P01/2008
<http://www.c3grid.de/documents/DataPreparation.pdf>

- [2] S. Kindermann:
C3Grid ISO Metadata Profile
http://www.c3grid.de/documents/c3grid_metadata.pdf
http://www.c3grid.de/documents/C3Datenkatalog-V_1_0.pdf
http://www.c3grid.de/documents/xml-schema_G0-C3GRID.xml

- [3] T. Langhammer, V. Achter, and V. Winkelmann:
Richtlinien für die Konfiguration des Workspaces [German]
http://www.c3grid.de/documents/Konfiguration_Workspace_deliverable.pdf

- [4] U. Schulzweida:
CDO user's Guide
<http://www.mpimet.mpg.de/~cdo/cdo.pdf>

- [5] T. Langhammer, F. Schnintke:
T5.1 (Supplement A): Interface Specification for C3-Grid Archives
http://www.c3grid.de/documents/interface_archive.pdf