NetCDF to/from CMC-RPN/CCCma format converter

There may be several tools that do CMC-RPN to/from NetCDF format conversions at UQÀM or at CMC/Dorval. The one described here is based on the Ouranos NetCDF to/from CCCma format converter, cdf2ccc. The resulting two programs, cdf2rpn and cdf2ccc, somewhat follow the 1.0 to 1.6 CF-Meta-data conventions that are recommended for climate and large-scale forecast data. The only difference between the two programs is that each one will by default expect its own file format. In fact, the two are hard-linked and the program's name at startup provides for the default behavior. For more info on CF-Meta-data conventions, please see

http://cfconventions.org/documents.html

Given the above, and the very "free" implementations often found in NetCDF files, *your mileage will vary*.

At UQÀM/SCA, the latest development version of the scripts that call the actual Linux binaries can usually be found under

${diag-tools}/r.diag_6.4.4-2.3.4_linux26-x86-64/bin

where diag-tools=/unique/armnssm/Ecssm/ssm-domains-base/tools/diag-tools and it is made available via the diagtools-alpha SSM bundle.

In Dorval, the different AIX (now deprecated) and Linux binaries used to be found under

${ARMNLIB}/modeles/diag/bin/${BASE_ARCH}

You will also need to define a UDUNITS2 environment variable, such as:

export UDUNITS2_XML_PATH=/opt/udunits2/share/udunits/udunits2.xml

You will also have to provide a CMC-RPN/NetCDF or CCCma/NetCDF dictionary that defines all of your name and unit conversions.

An example of this dictionary can be found in

${DIAGNOSTIQUE}/man/pdoc/attribut_netcdf.dat ** OR **
${DIAGNOSTIQUE}/src/cdf2ccc/attribut_netcdf.dat
where ${DIAGNOSTIQUE} is the actual location of the RDIAG/CDF package

Please note that the previous file actually holds some documentation as to its configuration and/or usage.

The converter's calling sequence will usually look something like this:

cdf2rpn -rpn your_fst_file -cdf your_netcdf_file -attr your_attribut_netcdf_file

Other important arguments include -*dir*, -*timdesc*, -invj, -*cell_method* and -*dtsize*. The -*dir* argument changes the conversion direction (default: from NetCDF); -*invj* reverses the order of the latitudes while converting them; -*timdesc* specifies time units (seconds, minutes, hours, days since ...) to be used in the NetCDF file; -*cell_method* overrides the default cell_method attribute definitions, i.e."time: mean" or "time: point" that may apply to time-mean or regular variables, respectively; finally -*dtsize* defines the implicit accumulator or time-mean period in hours to be used to define appropriate time bounds in NetCDF files.

The rest of the arguments are mainly required by the CCCma file conversions as this type of file does not contain several of the meta-data items needed by NetCDF. This information can generally be derived from the CMC-RPN internal file descriptors.

The converter will automatically generate time bounds information from CMC-RPN files when it detects the presence of multi-samples representing data such as the results of time averages, maximums, minimums, variances and standard deviations operations on a variable. This applies in particular to the output of the TIMAVG (TIMMAX, TIMMIN, ACCUMUL), TIMCOV (STDEV) and FSTMDIAG RDIAG commands. The '-dtsize' argument need not be used with such a variable. However, when converting from a NetCDF file, the '-dtsize' argument should be specified in order to reproduce the full CMC-RPN time-mean file information.

Current limitations of the converter:

- Only one set of multiple vertical levels is supported per file. And all of these must be of the same type.
- However, any number of single level variables are supported in the same file.
- Again, only one set of time-steps is supported for time-varying variables.

These constraints are mainly due to the use of NetCDF 3.x and its limited coordinate definitions; NetCDF version 4.x somewhat removes these limitations. But at this time this converter is still based on the NetCDF3 API, even though it is now generated using a 4.x library version. A NetCDF4-API-compliant version of the converter is nevertheless planned.

The full list of arguments and their description follows in the following tables. Assuming that …

1) "clés" is the actual argument name
2) "def1" is the corresponding default argument value when the argument itself is not specified
3) "def2" is its secondary default value when it is specified, but without a specific value
4) "?" denotes no values

clés(1) = 'cdf'	def1(1) = '?'	def2(1) = '?'
clés(2) = 'ccc'	def1(2) = '?'	def2(2) = '?'
clés(3) = 'dir'	def1(3) = 'def'	def2(3) = 'netcdf'
clés(4) = 'leap'	def1(4) = 'no'	def2(4) = 'yes'
clés(5) = 'dateo'	def1(5) = '?'	def2(5) = '0'
clés(6) = 'dt'	def1(6) = '?'	def2(6) = '0.0'
clés(7) = 'tlbl'	def1(7) = 'no'	def2(7) = 'yes'

clés(8) = 'lev'	def1(8) = '?'	def2(8) = '?'
clés(9) = 'tm'	def1(9) = '220.0'	def2(9) = '?'
clés(10) = 'ht'	def1(10) = '?'	def2(10) = '?'
clés(11) = 'grid'	def1(11) = '?'	def2(11) = '?'
clés(12) = 'ni'	def1(12) = '?'	def2(12) = '?'
clés(13) = 'nj'	def1(13) = '?'	def2(13) = '?'
clés(14) = 'pi'	def1(14) = '?'	def2(14) = '?'
clés(15) = 'pj'	def1(15) = '?'	def2(15) = '?'
clés(16) = 'dgrw'	def1(16) = '?'	def2(16) = '?'
clés(17) = 'd60'	def1(17) = '?'	def2(17) = '?'
clés(18) = '0lon'	def1(18) = 'GLOBAL'	def2(18) = '?'
clés(19) = '0lat'	def1(19) = '?'	def2(19) = '?'
clés(20) = 'dlon'	def1(20) = '?'	def2(20) = '?'
clés(21) = 'dlat'	def1(21) = '?'	def2(21) = '?'
clés(22) = 'invj'	def1(22) = 'yes'	def2(22) = 'no'
clés(23) = 'npack'	def1(23) = '999'	def2(23) = '?'
clés(24) = 'lalo'	def1(24) = 'no'	def2(24) = 'yes'
clés(25) = 'attr'	def1(25) = file_attr	def2(25) = local
clés(27) = 'fill_ccc'	def1(27) = '?'	def2(27) = 'ERR'
clés(28) = 'cle_nhem'	def1(28) = '?'	def2(28) = '?'
clés(29) = 'udunits'	def1(29) = udunit_def	def2(29) = '?'
clés(30) = 'rlonoff'	def1(30) = '?'	def2(30) = '?'
clés(31) = 'hyb_pt'	def1(31) = '?'	def2(31) = '?'
clés(32) = 'hyb_pref'	def1(32) = '?'	def2(32) = '?'
clés(33) = 'hyb_r'	def1(33) = '?'	def2(33) = '?'
clés(34) = 'rpn'	def1(34) = '?'	def2(34) = '?'
clés(35) = 'phis'	def1(35) = '?'	def2(35) = '?'
clés(36) = 'timdesc'	def1(36) = 'hours'	def2(36) = '?'
clés(37) = 'nongeog'	def1(37) = 'oui'	def2(37) = 'non'
clés(38) = 'xcoord'	def1(38) = '!@#$%^&'	def2(38) = '!@#$%^&'
clés(39) = 'ycoord'	def1(39) = '!@#$%^&'	def2(39) = '!@#$%^&'
clés(40) = 'zcoord'	def1(40) = '!@#$%^&'	def2(40) = '!@#$%^&'
clés(41) = 'tcoord'	def1(41) = '!@#$%^&'	def2(41) = '!@#$%^&'
clés(42) = 'dtsize'	def1(42) = '0.0'	def2(42) = '?'
clés(43) = 'calendar'	def1(43) = 'gregorian'	def2(43) = '?'
clés(44) = 'gribcode'	def1(44) = '?'	def2(44) = '?'
clés(45) = 'cell_method'	def1(45) = '?'	def2(45) = '?'
clés(46) = 'title'	def1(46) = ' '	def2(46) = ' '
clés(47) = 'typvar'	def1(47) = 'NC'	def2(47) = ' '
clés(48) = 'etiket'	def1(48) = 'Netcdf2RPN'	def2(48) = ' '

FORTRAN Data type : (C) character - (I) integer - (R) real

description(1) = (C) Nom du fichier netCDF
description(2) = (C) Nom du fichier CCCma
description(3) = (C) Convertir vers un des formats (*cccma* ou *rpncmc*) ou bien vers *netcdf*

Les arguments 4 à 10 décrivent les paramètres de base d'un fichier source en format CCCma

description(4) = (C) Année bissextile (off / no / non / on / yes / oui)
description(5) = (I) Date de départ de la simulation (AAAAMMJJHH)
description(6) = (R) Pas de temps (en secondes)
description(7) = (C) Format temporel CCCma = AAAAMMDDHH ?
description(8) = (C) Type de niveaux verticaux. Les seules valeurs reconnues sont *Pressure Levels*, *Sigma Levels*, *Gal-Chen Levels*, *10 m*, *2 m*, *Surface*, *Sea Level*, *Hybrid Levels*, *Log Pressure Hybrid Levels*, *Arbitrary Levels*, *Height*, *Top of Atmosphere*, *Soil Layers* et *Hybrid Height.* Voir la Note #3 pour plsu de détails
description(9) = (R) Variable TMOYEN de PARAMÈTRES
description(10) = (R) Hauteur du toit du modèle en mètres

description(11) = (C) Type de projection. Les seules valeurs reconnues sont *polar_stereographic,* l*on/lat*, *gaussian*, *rotated_latitude_longitude*, *rotated_pole* et *unknown.* Cet argument est surtout utilisé pour décrire les fichiers CCCma, mais il peut aussi l'être si le fichier source NetCDF ne contient pas toutes les informations requises.

Les arguments 12 à 17 décrivent les détails d'une projection *polar_stereographic*

description(12) = (I) Nombre de points de grille en X
description(13) = (I) Nombre de points de grille en Y
description(14) = (R) Coordonnée selon X du pôle (nombre de Δx)
description(15) = (R) Coordonnée selon Y du pôle (nombre de Δy)
description(16) = (R) Angle entre le méridien de Greenwich et l'axe X (degrés ouest)
description(17) = (R) Longueur de la maille vraie à 60 degrés (en mètres)

Les arguments 18 à 21 décrivent les détails d'une projection *lon/lat*

description(18) = (R) Longitude d'origine (degrés ouest)
description(19) = (R) Latitude d'origine (degrés nord)
description(20) = (R) Longueur de la maille selon les longitudes (degrés)
description(21) = (R) Longueur de la maille selon les latitudes (degrés)

Les arguments 22 à 35 sont de divers types

description(22) = (C) Inverser l'ordre de l'indice 'j' à la sortie
description(23) = (I) Densité de compression 0,1,2,4,-64,-32,-16
description(24) = (C) Écrire les latitudes et longitudes (off / no / non / on / yes / oui)
description(25) = (C) Nom du fichier dictionnaire. Les valeurs par défaut primaire et secondaire, respectivement, sont file_attr = /LOGICIELS/cdf2ccc/etc/attribut_netcdf.dat
local = ./attribut_netcdf.dat
description(27) = (R) Valeur de remplissage dans les sorties (*FILL VALUE*)
description(28) = (I) 0=Global, 1=Hémisphère Nord, 2=Hémisphère Sud (pour grilles PS ou Lat/Lon)
description(29) = (C) Chemin complet du fichier *udunits2.xml*

description(30) = (R) Déplacement des longitudes d'une grille tournée
description(31) = (R) Pression au toit de la coordonnée hybride (Pa)
description(32) = (R) Pression de référence pour une coordonnée hybride
description(33) = (R) Exposant pour une coordonnée hybride
description(34) = (C) Nom du fichier CMC-RPN
description(35) = (C) Nom du fichier PHIS (Option de niveaux *Gal-Chen*)

Les arguments 36 à 43 ne concernent que les fichiers NetCDF

description(36) = (C) Unités associées à la variable temporelle. Les seules valeurs reconnues sont *seconds*, *minutes*, *hours*, *days*, *months* et *years* (*since ...*)
description(37) = (C) Convertir les variables non-géographiques
description(38) = (C) Nom de la coordonnée en X du fichier NetCDF (voir Notes #2,3)
description(39) = (C) Nom de la coordonnée en Y du fichier NetCDF " " "
description(40) = (C) Nom de la coordonnée en Z du fichier NetCDF " " "
description(41) = (C) Nom de la coordonnée en T du fichier NetCDF " " "
description(42) = (R) Intervalle d'accumulation temporelle en heures
description(43) = (C) Nom du calendrier (pour plus de détails voir Note #4)

Les arguments 44 à 48 ne concernent que les fichiers CMC-RPN et NetCDF

description(44) = (I) Code GRIB pour une grille Lambert Conforme Conique
description(45) = (C) "Cell Method" utilisée dans les calculs temporels
description(46) = (C) Optionnel, "title" for meta-data
description(47) = (C) Optionnel, TYPVAR (défaut = "NC")
description(48) = (C) Optionnel, ETIKET (défaut = "Netcdf2RPN")

Notes regarding some of the above arguments:

1) When the program fails to convert a NetCDF file, a first step is to look at the the content of this file using the command "*ncdump filename.nc | more*" where *filename.nc* is the actual NetCDF file name. This will display the file dimensions, followed the included variables headers, each with their particular attributes. A global attributes section should be displayed at the end of this section. A (very large) data section will be displayed after the global descriptors. Adding the *-h* argument to the *ncdump* command will prevent the display of this large data section. Normally, each of the declared dimensions should also be extensively described in the variable section of a file. A data section relating to each of these coordinates should also be found following the header sections of a NetCDF file. The converter may attempt to supply default values when any of these conditions are not met, **but it is more than likely that it will FAIL to do so**.
2) The *-xcoord*, *-ycoord*, *-zcoord* and *-tcoord* arguments may be used when the program fails to recognize any of the existing x, y, z or t dimension names, for example, if the unlimdimid dimension is called ***lev*** rather than ***time*** or ***t***. Furthermore, the time coordinate can be associated to the unlimdimid dimension.
3) Assuming the vertical coordinate is recognized but its values are either missing or not appropriate for conversion to a CMC-RPN file format, it is possible to input an alternative set of values to be written in the CMC-RPN file. Given a NetCDF vertical dimension name of ***lev***, and given that a file called ***lev_remplacement.txt*** exists in the current working directory, the converter will attempt to read this file with a (BN,I10) FORTRAN I/O format (one 10 character integer "lev" value per line). The values thus retrieved will be assumed to be already coded and will be used "AS IS" when writing the CMC-RPN file. Any *lev* values to be found in the NetCDF files will then be ignored.
4) The recognized -calendar arguments are ***gregorian*** (or ***standard***), ***proleptic_gregorian***, ***365_day*** (or ***noleap***) and ***360-day***. The two *gregorian* options only differ before 1582-10-15: the *standard* one then follows the Julian 365.25-day year, while the *proleptic* extends the 365.2425-day year backward in time.

UDUNITS2 follows the *gregorian* calendar. This last argument is mainly used when converting to the NetCDF format, as a calendar attribute should always be found in the time coordinate description of CF-Meta-data compliant NetCDF files.

5) Again, a large number of the previous arguments exist to account for the very summary file format descriptors that holds with CCCma files. They thus are generally ignored when dealing with CMC-RPN files.

Finally, as this converter shares much of its low-level I/O routines with the RDIAG toolbox, the following arguments are also relevant:

clés(R1)= 'help'	def1(R1) = 'non'	def2(R1) = 'oui'
clés(R2) = 'info'	def1(R2) = 'non'	def2(R2) = 'oui'
clés(R3) = 'ipktyp'	def1(R3) = ' '	def2(R3) = ' '
clés(R4) = 'opktyp'	def1(R4) = ' '	def2(R4) = ' '
clés(R5) = 'input'	def1(R5) = '****'	def2(R5) = '****'
clés(R6) = 'output'	def1(R6) = '****'	def2(R6) = '****'
clés(R7) = 'date'	def1(R7) = ' '	def2(R7) = ' -1'
clés(R8) = 'singlz'	def1(R8) = ' '	def2(R8) = ' -1'
clés(R9) = 'seq'	def1(R9) = 'rnd'	def2(R9) = 'seq'
clés(R10) = 'vers'	def1(R10) = 'non'	def2(R10) = 'oui'
clés(R11) = 'na'	def1(R11) = '****'	def2(R11) = ' -1'
clés(R12) = 'keepip2'	def1(R12) = 'non'	def2(R12) = 'oui'
clés(R13) = 'mvalue'	def1(R13) = 'none'	def2(R13) = ' -1'
clés(R14) = 'mvalue'	def1(R14) = 'none'	def2(R14) = ' -1'
clés(R15) = 'bisect'	def1(R15) = 'oui'	def2(R15) = 'non'

This last set of arguments is documented at the end of Diag_Config.html which can be accessed via the RPN utilities RDIAG web page. Note that this page usually resides in the same directory as the current file.

Maintained by: B.Dugas, ESCER/UQAM (Dugas.Bernard@uqam.ca)
Latest revision: July 2019