

ICON NetCDF output to VTK converter

René Redler, MPI-M based on work by Moritz Hanke DKRZ.

This converter program is a byproduct of the YAC development done at DKRZ and MPI-M

<https://redmine.dkrz.de/doc/YAC/html/index.html>

Therefore it is currently required to download the whole package to build the executable. As it is quite fast to build the whole software stack this is not a strong limitation for the time being.

Installation

Required software:

- git
- autotools
- netcdf
- (g)make

To install:

```
mkdir <my_dir>
git clone --depth=1 git://redmine.dkrz.de/YAC.git <my_dir>
```

If all required software is installed on the default system paths

```
cd <my_dir>
autoreconf
configure
make
```

will be sufficient.

Installation on MPI PC:

```
module load autotools
autoreconf
```

```
configure \
--with-netcdf-include=/sw/lenny-x64/netcdf-4.1.3-gccsys/include \
--with-netcdf-lib=/sw/lenny-x64/netcdf-4.1.3-gccsys/lib
make
```

ICON2vtk uses some C99 standard which is not supported by all compiler. In this case you have to set CPPFLAGS -DANSI_C, and invoke the configure step again, followed by make.

Installation on Tornado

In this example, a YAC MPI-parallel version is compiled:

```
module load OpenMPI/1.4.3
```

```
autoreconf -i
configure CC=mpicc CFLAGS="-g -pg" FC=mpif90 \
--with-netcdf-root=/sw/sles10-x64/netcdf-4.1.3-static-gcc43 \
--with-hdf5-root=/sw/sles10-x64/hdf5-1.8.7-static \
--with-szip-root=/sw/sles10-x64/szip-2.1-static
```

In order to get a working version of the autotools you need to have

```
/sw/sles10-x64/autoconf-2.64/bin and /sw/sles10-x64/automake-1.11/bin
```

in your path.

Running the Program

At runtime the lib path has to be known, e.g.:

```
setenv LD_LIBRARY_PATH /sw/lenny-x64/netcdf-4.1.3-gccsys/lib
```

The ICON2vtk.x executable can be found in <my_dir>/examples.

ICON output files, e.g.

```
couple_atmo-ocean_ocn_iconR2B04-ocean_aqua_planet_0023.nc  
couple_atmo-ocean_ocn_iconR2B04-ocean_aqua_planet_0024.nc
```

ICON grid file on which the output data have been calculated, here:

```
iconR2B04-grid.nc
```

```
run ICON2vtk.x
```

```
> ICON2vtk.x -g iconR2B04-grid.nc \  
-b couple_atmo-ocean_ocn_iconR2B04-ocean_aqua_planet \  
-s 23 -e 24 -l 1 -t 1 -f T,S,u,v
```

or (all in one line)

```
> ICON2vtk.x --gridfile iconR2B04-grid.nc \  
--basename couple_atmo-ocean_ocn_iconR2B04-ocean_aqua_planet \  
--start 23 --end 24 --level 1 --timeincr 10 --field T,S,u,v
```

ICON2vtk.x --help or ICON2vtk.x -h provides a list of available options.

If compiled with **-DANSI_C** start ICON2vtk.x

```
> ICON2vtk.x
```

and follow the dialogue for the requested input, e.g.:

```
Enter ICON grid file: iconR2B04-grid.nc  
Enter data file base name <basename>_????.nc or restart_cpl :  
couple_atmo-ocean_ocn_iconR2B04-ocean_aqua_planet  
Enter start index: 23  
Enter time step increment: 10  
Enter end index: 24  
Enter field names separated by ',' :  
Enter field names separated by ',' : T,S,u,v  
Enter level: 1
```

As a result you will get a list of vtk files, named T_S_u_v_*.vtk running from *_0.vtk to *_N.vtk where N is the total number of timesteps -1 that are contained in the NetCDF input files.

For the time being it has only been tested for the extraction of horizontal levels out of 3d fields. Thus, a combination of 2d fields (like ELEV) and 3d fields (like T) does not work. Check beforehand what is in the NetCDF file and be reminded that field names are case sensitive.

ParaView – First Steps

From the VTK/ParaView page (<http://www.vtk.org/Wiki/ParaView>):

ParaView is an open-source, multi-platform application designed to visualize data sets of varying sizes from small to very large. The goals of the ParaView project include developing an open-source, multi-platform visualization application that supports distributed computational models to process large data sets. It has an open, flexible, and intuitive user interface. Furthermore, ParaView is built on an extensible architecture based on open standards. ParaView runs on distributed and shared memory parallel as well as single processor systems and has been successfully tested on Windows, Linux, Mac OS X, IBM Blue Gene, Cray XT3 and various Unix workstations and clusters. Under the hood, ParaView uses the Visualization Toolkit as the data processing and rendering engine and has a user interface written using the Qt cross-platform application framework.

ParaView is shipped with several Linux distributions, among them Ubuntu and Kubuntu. It is available on the MPI PCs and on the Tornado cluster.

Here we give a first very brief introduction on how to visualise the newly generated vtk file with ParaView. As ParaView is a quite powerful tool it is strongly recommended to have a look into the ParaView user manual to learn about the richness of functionality that is provided.

<http://www.paraview.org/files/v1.6/ParaViewUsersGuide.PDF>

http://paraview.org/Wiki/ParaView/Users_Guide/Table_Of_Contents

On the MPI PCs it is necessary to load the respective module:

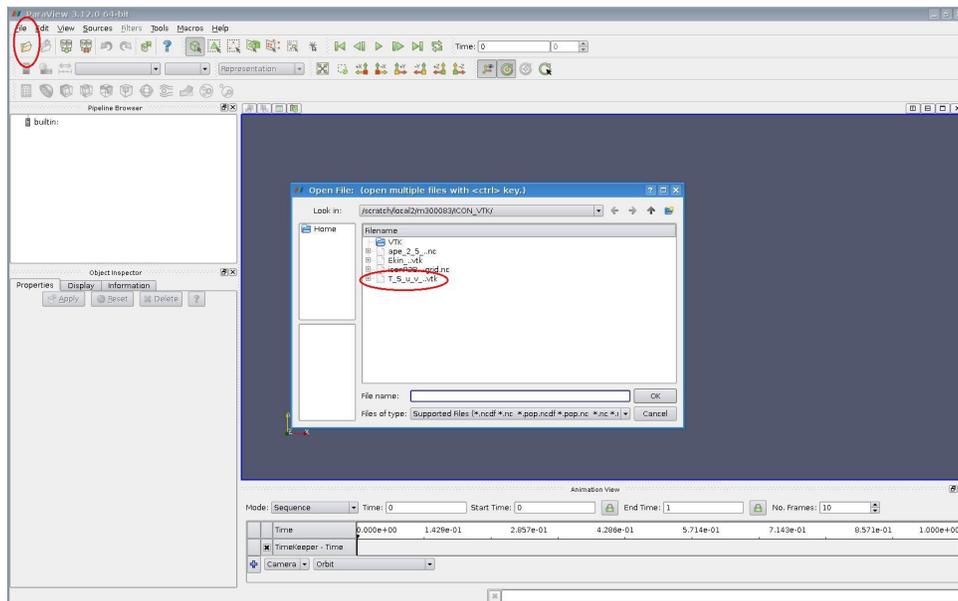
1.) module load paraview

Depending on your software installation this first step may be skipped.

2) Start paraview:

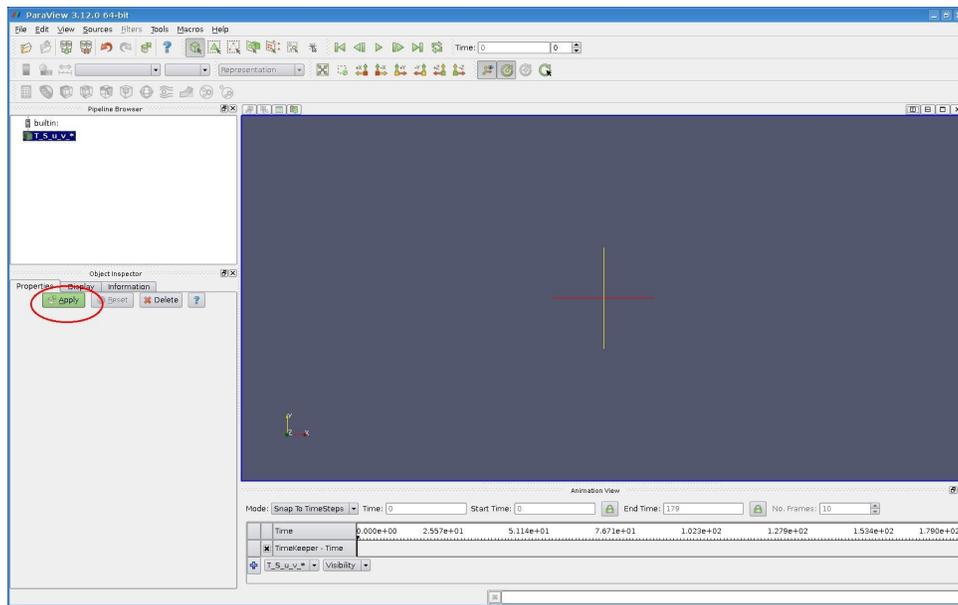
> paraview &

3) Load the vtk file(s)

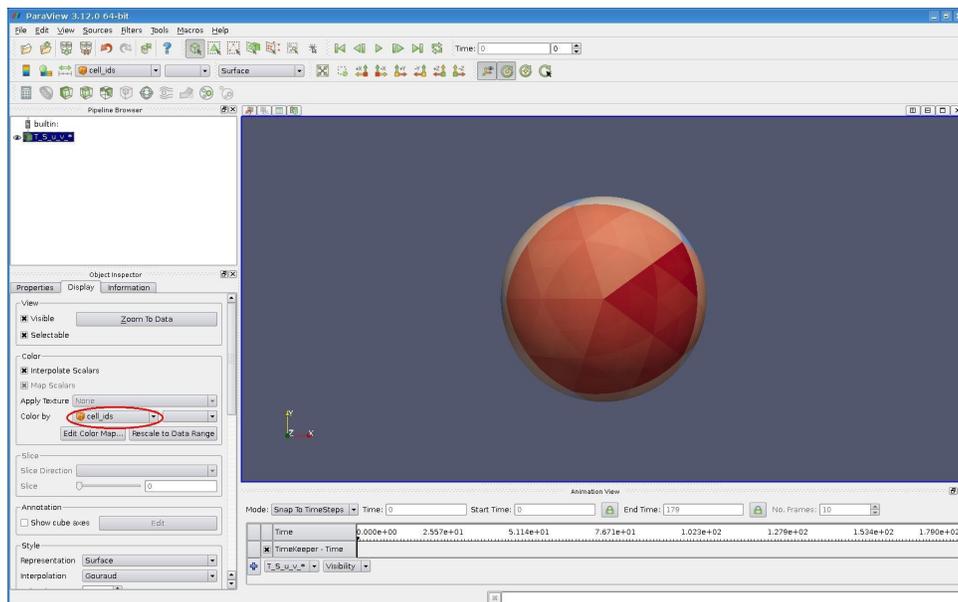


In the example above the files T_S_u_v_*.vtk will be grouped and can be loaded with one click.

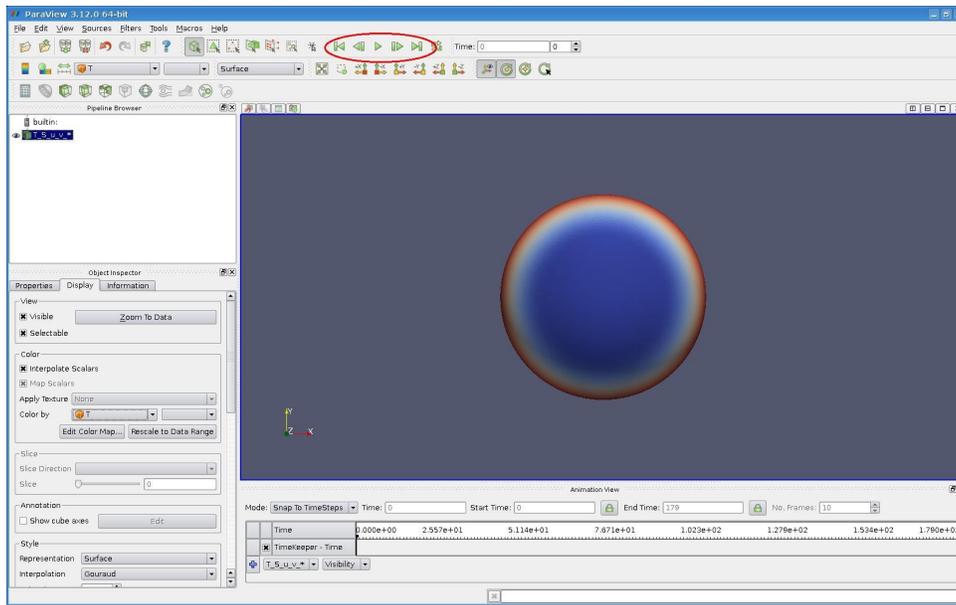
4) Apply selected data file



5) Select field



6) Display



Use the mouse to move and rotate the object. For time steering use the green buttons.