abinit.org

Specifications for the Nanoquanta-ETSF NetCDF file format

V. Olevano, M. Verstraete,
C. Freysoldt, Y. Pouillon,
X. Gonze, A. Cucca and D. Caliste

Nanoquanta
Network of Excellence

ETSF

File -etsf.nc

- It can contain:
  - **Crystallographic Data**
  - **Density**
  - **Potentials**
  - **Electronic Structure (Wavefunctions + Energies)**
- Purpose:
  - Exchange between DFT codes
  - DFT and else (postprocessing on density, TDDFT, BSE, GW, Quantum Transport)
  - Even between ab initio codes and non ab initio ones (Tight Binding, DMFT)

Why NetCDF?

- **High Portability**: little/big-endian problem free; machine precision free;
- **Versatile**: readable from Fortran, C, Java, Python, Perl, ...
- **Efficiency** (or better, efficiency in perspective): however, first save human time, and then computer;
- **Incremental**: you can add further data and the old reading interfaces still work -> High backcompatibility.

-etsf.nc characteristics

- It can be **Compact**: all the informations (chrystallographic data, density, potentials and wavefunctions) about a system in a **single file** (sparse information in many files, in many directories, usually leads to human errors in managing).
- But you can also split information however you want -> parallelization (**VO & MV do not advise**)
- Versatility and Redundancy: You can write only the **mandatory data** (advised), or also the **optional**
- If your code does not contain enough physics to write a mandatory data, we however encourage you to write a -etsf.nc file by using the alternative optional data. This will rely on the reading code; hoping that it will contain enough physics to anyway interpret your information.

NetCDF philosophy: writing

```
use netcdf

nf90_create(“foo-etsf.nc”,nf90_clobber,ncid)                                  ! opening

nf90_put_att(ncid,nf90_global,”title”,”Silicon bulk, Si 1s corehole, etc.”)    ! global attribute

nf90_def_dim(ncid,”number_of_kpoints”,4,nkdimid)                              ! Dimensions decl.

nf90_def_var(ncid,”kpoint_weights”,nf90_double, (/nkdimid /), kwid)           ! Variables decl.

nf90_enddef(ncid)

nf90_put_var(ncid,kwid, (0.125, 0.25, 0.25, 0.375) )                          ! write Variables

nf90_close(ncid)                                                              ! closing
```

NetCDF philosophy: reading

```
use netcdf

nf90_open("foo-etsf.nc",nf90_nowrite,ncid)                          ! opening

s= nf90_inquire_attribute(ncid,nf90_global,"title",len=tl)          ! inquire global attribute
if(s /= nf90_noerr) nf90_get_attribute(ncid,nf90_global,"title",filetitle)   ! get global attribute

s= nf90_inq_dimid(ncid,"number_of_kpoints",nkdimid)                 ! check if dimension exists
if(s/=nf90_noerr) nf90_inquire_dimension(ncid,nkdimid,len=nk)       ! read dimension

s= nf90_inq_varid(ncid,"reduced_coordinates_of_kpoint",kvid)        ! check if Variable exists
if(s/=nf90_noerr) nf90_get_var(ncid,kvid,kpoints)                   ! read Variable
else try_to_manage_if_this_does_not_exist(....)                     ! or try to read other ...

nf90_close(ncid)                                                    ! closing
```

Hence, if you caught the right philosophy behind NetCDF, what is correct to do and what not

- Once agreed on the tags' names, **never change them!** Nor change the rank and the order of the dimensions in arrays (neither for aesthetic reasons nor alphabetic or whatever) -> lost of back-compatibility.
- You can write data in the order you wish!
- If a mandatory data doesn't apply for an unpredicted reason, **better not to write it** than to invent a new convention for it. -> rely on the reading code escape capability.
- **You can add further data/tags**, it does not interfere with the others. But for every code being able to use them, you need to make them agreed in the standards.

.etsf-nc structure

Intro

- **General Info** (NetCDF global attributes):
  - title, history
- **Dimensions**:
  - number_of_atom_species, number_of_atoms
  - number_of_symmetry_operations
  - max_number_of_states, number_of_kpoints, max_number_of_coefficients
  - number_of_spins, numbers_of_spinor_components, number_of_components
  - number_of_grid_points_vector1, ...

Data: Crystallographic File

- **Crystal Structure**:
  - primitive_vectors
  - reduced_symmetry_matrices, reduced_symmetry_translations
  - atom_species, atomic_numbers, atom_species_names, chemical_symbols
  - reduced_atom_positions
  - space_group

Data: Density File or Potential File

- **Density**:
  - density[number_of_components, number_of_grid_points_vector1, ..., real_or_complex_density]
- **Potentials**:
  - exchange_potential[number_of_components, number_of_grid_points_vector1, ..., real_or_complex_potential]
  - correlation_potential[...]
  - exchange_correlation_potential[...]

Data: Electronic Structure

- **Brillouin Zone**:
  - reduced_coordinates_of_kpoints[number_of_kpoints,number_of_reduced_dimensions]
  - kpoints_weights[number_of_kpoints]
- **Energies, Occupations**:
  - number_of_states[number_of_spins,number_of_kpoints]
    - k_dependent (flag, attribute)
  - eigenvalues[number_of_spins,number_of_kpoints,max_number_of_states]
  - occupations[number_of_spins,number_of_kpoints,max_number_of_states]
- **Wavefunctions**:
  - basis_set (= ”plane_waves” or ...)
  - number_of_coefficients[number_of_kpoints]
  - reduced_coordinates_of_plane_waves [number_of_kpoints,max_number_of_coefficients,number_of_reduced_dimensions]
  - coefficients_of_wavefunctions[...]
- **Real Space Wavefunctions**:
  - real_space_wavefunctions[...]

Data: Optional

- **Atomic Data:**
  - valence_charges, pseudopotential_types, number_of_electrons
- **Brillouin Zone:**
  - kpoint_grid_shift, kpoint_grid_vectors, monkhorst_pack_folding
- **Convergency:**
  - kinetic_energy_cutoff, smearing_width, smearing_scheme, fermi_energy
  - exchange_functional, correlation_functional

Data: Electronic Structure

- **GW/BSE/TDDFT data**:
  - gw_corrections[...]
  - kb_formfactors[...]

Asked Questions and Open Points

- **Dielectric Function (Screening)**?

.etsf-nc at present

- -etsf.nc f90 interface (VO & Matthieu Verstraete) + Damien Caliste interface.
- **kss2etsf** conversion utility from ABINIT **_KSS** format to **.etsf-nc**;
- DP (TDDFT) code used to test ETSF NetCDF interface: it works fine!
- EXC (BSE) can also be released with a NetCDF interface, but we would like to solve the last point, the **screening** -> agreement with the community.

.etsf-nc in future

- -etsf.nc: integration within ABINIT, DP, EXC, STGW, SPHINGx (?), WANT.
- Converter NetCDF ETSF <-> IOTK Espresso?
- What about VASP, Siesta, Espresso?