European Network of Excellence
NANOQUANTA
Nanoscale Quantum Simulations for
Nanostructures and advanced Materials

Specification of file formats for NANOQUANTA Specification version 1.3 Final revision for this version 2 June 2006 (SpecFFNQ1.3final)

by the Integration Team 9 and collaborators

(X. Gonze, C.-O. Almbladh, A. Cucca, M. Marques,
C. Freysoldt, V. Olevano, Y. Pouillon, M. Verstraete)

Nanoquanta
Network of Excellence

Index

Abstract

In order to allow softwares to interact and exchange data, file format specifications are mandatory. Widely agreed file format specifications are still lacking in the field of first-principles calculations of material properties.
One of the (numerous) objectives of the European Network of Excellence "NANOQUANTA" is precisely to specify file formats, for the contents that are relevant to the scientific activity of its constituting nodes. After one year and a half of existence of NANOQUANTA, the present document describes the status of the corresponding work. It includes an inventory of the different contents, discussions of existing formats when relevant, as well as detailed (new) NetCDF specifications, for selected contents (density/wavefunctions). It is hoped that the new specifications will be implemented in many different softwares, or (at least) will be the basis of even better file format specifications.

Introduction

This document has two goals :
- to provide to the European Union *"A report of progress with achieving effective code interoperability and portability of essential data files, including specification of file formats for ETSF ...",* this report being a deliverable of the NANOQUANTA contract ;
- to inform the members of NANOQUANTA, as well as the electronic structure community at large, of the agreed specifications, in view of further discussions and implementations.

It is to be understood as the *first version of file specifications*, to be updated on a regular basis. Indeed, it reflects the status of the work on file format specification as of June 2006. It will be referred to as SpecFFNQ1 (The string "NANOQUANTA" will be abbreviated to NQ throughout the document). It is expected that the NQ nodes will rely on this document to implement the agreed file formats. Still, there are other file formats on which the specification work has to be done. In particular, it is planned that each year, a workshop should review the existing implementations, and discuss further the specifications. Even the few detailed file format specifications present in this document are subject to revision and improvements. An update of this document, to be named SpecFFNQ2, is thus expected around October 2006.

The document is organized in sections :
Section 1 presents general considerations concerning the present file format specifications.
Section 2 presents general specifications concerning NQ NetCDF file formats.
Section 3 deals with files containing density/potential, and presents a rather detailed NetCDF specification, ready for exchange of data among the NQ nodes.
Section 4 deals with files containing wavefunctions, and has reached the same level of detail.
Section 5 deals with pseudopotentials / PAW set up files. It presents the existing specifications, and summarizes the debates and conclusions reached during the Louvain-la-neuve mini-workshop (see below).
Section 6 presents briefly the situation concerning the standardization of files containing crystalline structure and atomic geometries.
Section 7 is an overview of the other contents relevant for NQ, and the status of file format specification.

In appendix A, the reader will find basic information about the European NQ Network of Excellence, as well as about the planned European Theoretical Spectroscopy Facility (ETSF). Ten groups (called nodes) are core members of this project (about one hundred researchers). NQ started in June 2004, thanks to funds from the European Union. It should last four years. One of the outcome of this project should be a virtual institute, ETSF, to be continued beyond the initial four-year period. Appendix B presents an historical overview of the efforts related to file standardization, focusing on those leading to the present document. The FSAtom project played a role of precursor. In the present NQ network, several preparatory discussions first took place, until a mini-workshop was organized, on 15-16 Nov 2005, in Louvain-la-neuve (Belgium), where the delegates from the ten nodes, helped by external invitees, issued *guidelines* (as well as, in some cases, *detailed prescriptions*), to achieve a specification of file formats. The Web site of the workshop is http://www.pcpm.ucl.ac.be/workshop_files. The members of the IT 9 (integration team 9, team in charge of the NQ code integration) and some collaborators worked out additional detailed prescriptions, and wrote the present document following these guidelines.

Section 1. General considerations concerning the present file format specifications.

One has to consider separately the set of data to be included in each of different types of files, from their representation. Concerning the latter, one encounters simple text files, binary files, XML-structured files, NetCDF files, etc … It was already decided previously (Nanoquanta meeting Maratea Sept. 2004) to evolve towards formats that deal appropriately with the self-description issue, i.e. XML and NetCDF. The inherent flexibility of these representations will also allow to evolve specific versions of each type of files progressively, and refine earlier working proposals. The same direction has been adopted by several groups of code developers that we know of.
Information on NetCDF and XML can be obtained from the official Web sites,
http://www.unidata.ucar.edu/software/netcdf/ and
http://www.w3.org/XML /
There are numerous other presentations of these formats on the Web, or in books.
The elaboration of file formats based on NetCDF has advanced a lot during the Louvain-la-neuve mini-workshop. There has been also some remarks about XML.

Concerning XML

(A) The XML format is most adapted for the structured representation of relatively small quantity of data, as it is not compressed.
(B) It is a very flexible format, but hard to read in Fortran (no problem in C, C++ or Python). Recently, Alberto Garcia has set up a XMLF90 library of routines to read XML from Fortran.
http://lcdx00.wm.lc.ehu.es/~wdpgaara/xml/index.html
Other efforts exists in this direction http://nn-online.org/code/xml/

Concerning NetCDF

(A) Several groups of developers inside NQ have already a good experience of using it, for the representation of binary data (large files).
(B) Although there is no clear advantage of NetCDF compared to HDF (another possibility for large binary files), this experience inside the NQ network is the main reason for preferring it. By the way, NetCDF and HDF are willing to merge (this might take a few years, though).
(C) File size limitations of NetCDF exist, see appendix F, but should be overcome in the future.

Thanks to the flexibility of NetCDF, the content of a NetCDF file format suitable for use for NQ softwares might be of four different types :
(1) The actual numerical data (that defines a file for wavefunctions, or a density file, etc ...), whose NetCDF description would have been agreed.
(2) The auxiliary data that are mandatory to make proper usage of the actual numerical data. The NetCDF description of these auxiliary data should also be agreed.
(3) The auxiliary data that are not mandatory, but whose NetCDF description has been agreed, in a larger context.
(4) Other data, typically code-dependent, whose existence might help the use of the file for a

specific code. The name of these variables should be different from the names chosen for agreed variables (1)-(3). Such other data might even be redundant with (1)-(3).

Such content is compatible with a file format being complete for use by many codes, though adapted for the specific usage by one code. The NQ file descriptions to be provided later (sections 2, 3 and 4) are based on this generic classification of data that can be integrated in such a NetCDF file.

In order to address the 2 GB limit (see Appendix F), as well as the use of NetCDF files for parallel calculations, one file can actually be split into several partial files. Selected variables should describe the differing content of each of them. As an example, in section 4, a file containing a set of wavefunctions can be split in different files containing selected bands and/or k-points, however being exactly similar in every other respect.
Some technical details concerning the use of NetCDF files will apply to all formats specified in the NQ network :
(1) concerning the variable names, long names should be chosen, as close as possible to natural language (so inherently self-descriptive).
(2) all variable names are lower case, except "Conventions" - a name agreed by the NETCDF community
(3) underscores are used to replace blanks separating words
(4) in the tables, the slow indices are left-most, and the fast indices are right-most, so that the order of indices has to be reversed in FORTRAN

Section 2. General specifications for NQ NETCDF files.

2.1 Global attributes of NQ NetCDF files

Global attributes are used for a general description of the file, mainly the file format convention. Important data is not contained in attributes, but rather in variables.

The length of character attributes is the maximum length this attribute may take. This is relevant for reading, where sufficient space must be provided. In writing, the defined length may be reduced to the real length of the attribute.

Table 2.1 gather specifications for required attributes in any NQ NETCDF files. Table 2.2 presents optional attributes for NQ NETCDF files.

Attributes	Type (length)	Notes
file_format	char (80)	"ETSF Nanoquanta"
file_format_version	real	1.1 or 1.2 or 2.0 ...
Conventions	char (80)	"http://www.etsf.eu/fileformats"

Table 2.1 Mandatory global attributes for NQ NetCDF files.

Attributes	Type (length)	Notes
history	char (1024)	
title	char (80)	

Table 2.2 Optional global attributes for NQ NetCDF files.

Detailed description (tables 2.1 and 2.2)

file_format Name of the file format for ETSF wavefunctions.
file_format_version Real version number for file format (only one period, e.g. 1.2).
Conventions NetCDF recommended attribute specifying where the conventions for the file can be found on the Internet. Note that at the time of writing, the conventions are not yet available at http:// www.etsf.eu/fileformats , but at http:// www.cmt.york.ac.uk/nanoquanta/docs/specifications .

history NetCDF recommended attribute : each code modifying/writing this file is encouraged to add a line about itself in the history attribute. char(1024) allows for 12 additions of at most 80 characters.
title Short description of the content (system) of the file.

2.2 Generic attributes of variables in NQ NetCDF files

A few attributes might apply to a large number of variables. They are gathered in Table 2.3 .

Attributes	Type (length)	Notes
units	char (80)	required for variables that carry units
scale_to_atomic_units	double	required for units other than "atomic units"

Table 2.3 Optional generic attributes for variables in NQ NetCDF files.

Detailed description (table 2.3)

units It is one of the NetCDF recommended attributes, but it only applies to a few variables in our case, since most are dimensionless. The use of atomic units (corresponding to the string "atomic units") is advised throughout for portability. If other units are used, the definition of an appropriate scaling factor to atomic units is mandatory. Actually, the definition of the name "units" in the NQ files is only informative : the "scale_to_atomic_units" information should be the only one used to read the file by machines.

scale_to_atomic_units If "units" is something other than atomic units (Hartree for energies, Bohr for lengths) we request the definition of an appropriate scaling factor. The appropriate value in atomic units is obtained by **multiplying** the number found in the variable by the scaling factor. Examples:
units="eV" => scale_to_atomic_units = 0.036749326
units="angstrom" => scale_to_atomic_units = 1.8897261
units="parsec" => scale_to_atomic_units = 5.8310856e+26
This can be used to deal with unknown units.
Note that the recommended values for the fundamental constants can be found at http://physics.nist.gov/cuu/Constants/index.html .

2.3 Flag-like attributes

"Flag-like" attributes can take the values "yes" and "no". When such attributes are written, they should be written in full length and small letters. When they are read, only the first character needs to be checked (i.e. "y" or "n" -- this simplifies life a lot).

2.4 Dimensions

Dimensions are used for one- or multidimensional variables. It is very important to remember that the netCDF interface adapts the dimension ordering to the programming language used. The notation here is C-like, i.e. the last index varies fastest. In Fortran, the order is reversed. When implementing new reading interfaces, the dimension names can be used to check the dimension ordering. The dimension names help also to identify the meaning of certain dimensions in cases where the number alone is not sufficient.

The list of variables that specify dimensions in NQ NetCDF files is given in Table 2.4 and 2.5. Table 2.4 list the dimensions that are not supposed to lead to a splitting, while table 2.5 list the dimensions that might be used to define a splitting (e.g. in case of parallelism).

Dimensions	Type (index order as in C)	Notes
character_string_length	integer	Always ==80
real_or_complex	integer	Either ==1 (real) or ==2 (complex)
number_of_cartesian_directions	integer	Always ==3
number_of_reduced_dimensions	integer	Always ==3
number_of_vectors	integer	Always ==3
number_of_spinor_components	integer	Either ==1 or 2
number_of_symmetry_operations	integer	
number_of_atoms	integer	
number_of_atom_species	integer	
symbol_length	integer	Always ==2

Table 2.4 Variables that specify dimensions in NQ NetCDF files (no splitting case).

Detailed description (table 2.4)

character_string_length The maximum length of string variables (attributes may be longer).
real_or_complex Complex numbers are stored as arrays of 2 real numbers, namely the real and the imaginary part.
number_of_cartesian_directions Used for absolute coordinates.
number_of_reduced_dimensions Used for reduced (also called relative) coordinates in reciprocal or real space.
number_of_vectors Used to distinguish the vectors when defining their relative/reduced coordinates.
number_of_spinor_components For non-spinor wavefunctions, this dimension must be present and equal 1. For spinor wavefunctions this dimension must equal to 2.
number_of_symmetry_operations The number of symmetry operations.
number_of_atoms The number of atoms in the unit cell.
number_of_atom_species The number of different atom species in the unit cell.
symbol_length Maximum number of characters for the chemical symbols

Dimensions	Type (index order as in C)	Notes
max_number_of_states	integer	
number_of_kpoints	integer	
number_of_spins	integer	
number_of_components	integer	
max_number_of_coefficients	integer	
number_of_grid_points_vector1	integer	
number_of_grid_points_vector2	integer	
number_of_grid_points_vector3	integer	

Table 2.5 Variables that specify dimensions in NQ NETCDF files (possible splitting case). For the auxiliary variables needed in case of splitting, see Table 2.6

Detailed description (table 2.5)

max_number_of_states The maximum number of states
number_of_kpoints The number of kpoints
number_of_spins Used to distinguish collinear spin-up and spin-down components :
1 for non-spin-polarized or spinor wavefunctions
2 for collinear spin (spin-up and spin-down)
number_of_components Used for the spin components of spin-density matrices :
1 for non-spin-polarized
2 for collinear spin (spin-up and spin-down)
4 for non-collinear spin (average density, then magnetization vector in cartesian coordinates x,y and z)
max_number_of_coefficients The (maximum) number of coefficients for the basis functions at each k-point, except in the case of real space grids (see next lines)
number_of_grid_points_vector1 The number of grid points along direction 1 in the unit cell in real space, for dimensioning the wavefunction coefficients (an alternative to max_number_of_coefficients)
number_of_grid_points_vector2 Same as number_of_grid_points_vector1, for the second direction.
number_of_grid_points_vector3 Same as number_of_grid_points_vector1, for the third direction.

To clarify the interplay between number_of_spins, number_of_components, and number_of_spinor_components, note the different following magnetic or non-magnetic cases:
Non-spin-polarized :
number_of_spins=1 , number_of_spinor_components=1, number_of_components=1
Collinear spin-polarized :
number_of_spins=2, number_of_spinor_components=1, number_of_components=2
Non-collinear spin-polarized :
number_of_spins=1, number_of_spinor_components=2, number_of_components=4

We now turn to the specification of the (optional) splitting of files in partial files. Such splitting might be done in many different ways. In order to allow for very general, flexible, splittings, but still rely on a simple system, we set up different pairs of variables, one for each possible splitting. These pairs of variables are described in Table 2.6. If a software cannot cope with the file splitting, it should simply check that no file splitting is done, and if the contrary happens, it should stop.

Let us work out an example.
Suppose we split the file according to the kpoints. The full set might have 10 kpoints, of which 3 kpoints (number 1, 2 and 5) might be contained in a first file, 3 other kpoints (number 3, 6 and 9) might be contained in a second file, and the 4 remaining kpoints (number 4, 7, 8 and 10) might be contained in the third file.
Then, the first file will contain :
number_of_kpoints = 10 , my_number_of_kpoints = 3 , my_kpoints=(1,2,5)
The second file will contain :
number_of_kpoints = 10 , my_number_of_kpoints = 3 , my_kpoints=(3,6,9)
The third file will contain :
number_of_kpoints = 10 , my_number_of_kpoints = 4 , my_kpoints=(4,7,8,10)

If more than one splitting is done, the file will contain the intersection of the split data.
As an example, suppose we split the file according to the kpoints and the spins. The full set of kpoints might have 4 kpoints, and there would be two spins. We perform two splittings, one separating kpoints 1 and 2 from kpoints 3 and 4, and one separating the spins.
The first file might contain :
number_of_kpoints = 4 , my_number_of_kpoints = 2 , my_kpoints=(1,2)
number_of_spins = 2 , my_number_of_spins = 1 , my_spins=(1)
The second file might contain :
number_of_kpoints = 4 , my_number_of_kpoints = 2 , my_kpoints=(3,4)
number_of_spins = 2 , my_number_of_spins = 1 , my_spins=(1)
The third file might contain :
number_of_kpoints = 4 , my_number_of_kpoins = 2 , my_kpoints=(1,2)
number_of_spins = 2 , my_number_of_spins = 1 , my_spins=(2)
The fourth file might contain :
number_of_kpoints = 4 , my_number_of_kpoins = 2 , my_kpoints=(3,4)
number_of_spins = 2 , my_number_of_spins = 1 , my_spins=(2)

Different variables might change their sizes when splitting is used. The list of variables whose size might change compared to non-split files will have to be specified.

Dimensions	Type (index order as in C)	Notes
my_max_number_of_states	integer	At least 1, at most number_of_states
my_number_of_kpoints	integer	At least 1, at most number_of_kpoints
my_number_of_spins	integer	At least 1, at most number_of_spins
my_number_of_components	integer	At least 1, at most number_of_components
my_number_of_grid_points_vector1	integer	At least 1, at most number_of_grid_points_vector1
my_number_of_grid_points_vector2	integer	At least 1, at most number_of_grid_points_vector2
my_number_of_grid_points_vector3	integer	At least 1, at most number_of_grid_points_vector3
my_number_of_coefficients	integer	At least 1, at most number_of_coefficients

Table 2.6a Dimensions of variables to specify the (optional) splitting of one file in different partial files. These dimensions and associated variables (see Table 2.6b) are defined by pair (one integer, and one integer array). Any one of these pairs can be used to split the files, and several of these pairs can be used as well. In case several pairs are used, the content of the file is defined by the intersection of the different integer arrays. The detailed description of these variables is induced from the one of the corresponding Table 2.5 variables.

Variables	Type (index order as in C)	Notes
my_states	integer[my_max_number_of_states]	
my_kpoints	integer[my_number_of_kpoints]	
my_spins	integer[my_number_of_spins]	
my_components	integer[my_number_of_components]	
my_grid_points_vector1	integer[my_number_of_grid_points_vector1]	
my_grid_points_vector2	integer[my_number_of_grid_points_vector2]	
my_grid_points_vector3	integer[my_number_of_grid_points_vector3]	
my_coefficients	integer[my_number_of_coefficients]	

Table 2.6b Variables to specify the (optional) splitting of one file in different partial files. See the explanation in Table 2.6a. The detailed description of these variables is induced from the one of the corresponding Table 2.5 variables.

2.5 Optional variables

In order to avoid the "divergence of the formats in the additional data" (Olevano), we propose names and formats for some information that is likely to be written to the files. This section will grow in future format versions. Please report any variable you miss here, so we can add it to the list. Nothing of these data is mandatory for the file formats to be described later. Some of the proposed variables contain redundant information.

All optional variables must be defined **BEFORE** the largest size array of the file, otherwise this array will be restricted to 4GB. Examples of such arrays are coefficients_of_wavefunctions or real_space_wavefunctions (see later).

Tables 2.7 to 2.9 present these optional variables, grouped with respect to their physical relevance : atomic structure, electronic structure, and reciprocal space.

Variables	Type (index order as in C)	Notes
space_group	integer	Between 1 and 232
atom_species	integer[number_of_atoms]	Between 1 and "number_of_atom_species"
reduced_atom_positions	double[number_of_atoms] [number_of_reduced_dimensions]	
valence_charges	double[number_of_atom_species]	
atomic_numbers	double[number_of_atom_species]	
atom_species_names	char[number_of_atom_species] [character_string_length]	
chemical_symbols	char[number_of_atom_species] [symbol_length]	
pseudopotential_types	char[number_of_atom_species] [character_string_length]	

Table 2.7 Optional variables : atomic structure

Variables	Type (index order as in C)	Notes
number_of_electrons	integer	
exchange_functional	char [character_string_length]	
correlation_functional	char [character_string_length]	
fermi_energy	double	Units attribute needed
smearing_scheme	char [character_string_length]	
smearing_width	double	Units attribute needed

Table 2.8 Optional variables : electronic structure

Variables	Type (index order as in C)	Notes
kinetic_energy_cutoff	double	Units attribute needed
kpoint_grid_shift	double [number_of_reduced_dimensions]	
kpoint_grid_vectors	double[number_of_vectors] [number_of_reduced_dimensions]	
monkhorst_pack_folding	integer [number_of_vectors]	

Table 2.9 Optional variables : reciprocal space

Detailed description (tables 2.7 to 2.9)

space_group Space group number according to international tables of crystallography (from 1 to 232)

atom_species Types of each atom in the unit cell. Note that the first type of atom has number "1", and the last type of atom has number "number_of_atom_species".

reduced_atom_positions Positions of the different atoms in the unit cell in relative/reduced coordinates.

valence_charges Ionic charges for each atom species.

atomic_numbers Atomic number for each atom species. If not appropriate (fractional charge atoms or similar), zero may be used.

chemical_symbols Chemical symbol for each atom species (as in periodic table). If not appropriate (fractional charge atoms or similar), "X" may be used.

atom_species_names Descriptive name for each atom species = "H" "Ga" plus variants like "Ga-semicore" "C-1s-corehole" "C-sp2" "C1"

number_of_electrons Number of electrons in the elementary cell.

fermi_energy Fermi energy corresponding to occupation numbers.

smearing_scheme Smearing scheme used for metallic or finite temperature occupation numbers = "gaussian", "fermi-dirac", "cold-smearing", "methfessel-paxton-n" for n=1 ... 10

smearing_width Smearing width used with scheme above.

kinetic_energy_cutoff Cutoff used to generate the plane wave basis set.

kpoint_grid_vectors Basis vectors for kpoint grid if it is homogeneous.

kpoint_grid_shift Shift for offset of grid of kpoints. Used with both kpoint_grid_vectors and monkhorst_pack_folding.

monkhorst_pack_folding This indicates the ``folding" for regular kpoint grids (e.g. Monkhorst-Pack Phys. Rev. B 13, 5188 (1976)). An alternative to kpoint_grid_vectors.

pseudopotential_types Type of pseudopotential scheme
= "bachelet-hamann-schlueter", "troullier-martins", "hamann", "hartwigsen-goedecker-hutter", "goedecker-teter-hutter" ...
A standardized list should be found or established.

exchange_functional String describing the functional used for exchange: names should be taken from the Nanoquanta XC library specifications (at present, under construction).

correlation_functional String describing the functional used for correlation: Lee Yang Parr or Colle-Salvetti etc... names should be taken from the Nanoquanta XC library specifications.

2.6 Naming conventions

NetCDF files, that respect the specifications described in the present document, should be easily recognized. We suggest to append, in their names, the string "-etsf.nc" .

The appendix ".nc" is a standard convention for naming NetCDF files, see http://www.unidata.ucar.edu/software/netcdf/docs/faq.html#filename . Some filesystems are case-insensitive, and this motivates the lower-case choice. Finally, a dash is to be preferred to an underscore to allow the files references by a Web search engine.

Section 3. Specification for files containing a density or a potential

3.1 Specification

A NQ NetCDF file for a density should contain the following set of mandatory information :
(1) The three attributes defined in Table 2.1
(2) The following dimensions from Table 2.4 (dimensions that do not lead to a splitting) :
- number_of_cartesian_directions
- number_of_vectors
- real_or_complex

(3) The following dimensions from Table 2.5 (dimensions that might lead to a splitting) :
- number_of_components
- number_of_grid_points_vector1
- number_of_grid_points_vector2
- number_of_grid_points_vector3

(4) The primitive vectors of the cell, as defined in Table 3.1
(5) The density or potential, as defined in Table 3.2 or 3.3 . This variable must be the last, in order not to be limited to 4 GB.

As mentioned in section 1 and 2, such file might contain additional information agreed within NQ, such as any of the variables specified in section 2. Such file might also contain additional information specific to the software that generated the file. It is not expected that this other software-specific information might be used by another software.

The information might distributed among different files, thanks to the use of splitting of data for variables :
- number_of_components
- number_of_grid_points_vector1
- number_of_grid_points_vector2
- number_of_grid_points_vector3

In case the splitting related to one of these variables is activated, then the corresponding variables in Table 2.6 must be defined. Accordingly, the dimensions of the variables in Table 3.2 and 3.3 will be changed, to accommodate only the segment of data effectively contained in the file.

A NQ NetCDF exchange, correlation, or exchange-correlation potential file should contain at least one variable among the three presented in Table 3.3 in replacement of the specification of the density. The type and size of such variables are similar to the one of the density. The other variables required for a density are also required for a potential file. Additional NQ or software-specific information might be added, as described previously.

In case the splitting related to one of these variables is activated, then the corresponding variables in Table 2.6 must be defined. Accordingly, the dimensions of the variables in Tables 3.2, and/or 3.3 might have to be changed, to accommodate only the segment of data effectively contained in the file.

Variables	Type (index order as in C)	Notes
primitive_vectors	double[number_of_vectors] [number_of_cartesian_directions]	By default, given in Bohr

Table 3.1 The primitive vectors, mandatory to have a density file.

Variables	Type (index order as in C)	Notes
density	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex]	This is a pseudo-density. Note in case of PAW, the augmentation contribution is missing. By default, the density is given in atomic units, that is, number of electrons per Bohr^3 The "units" attribute is needed.

Table 3.2 The specification of the density (the last of the variables)

Variables	Type (index order as in C)	Notes
correlation_potential	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex]	Note in case of PAW, the augmentation contribution is missing. Units attribute needed.
exchange_potential	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex]	Note in case of PAW, the augmentation contribution is missing. Units attribute needed.
exchange_correlation_potential	double[number_of_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex]	Note in case of PAW, the augmentation contribution is missing. Units attribute needed.

Table 3.3 The specification of exchange, correlation, and exchange-correlation potentials.

3.2. Comments

(1) A density in such a format (represented on a 3D homogeneous grid) is suited for the representation of smooth densities, as obtained naturally from pseudopotential calculations using plane waves.
(2) The definition of names for other types of potentials meet some problems, that should be examined in further NQ meetings. In the absence of a specification of name, the NQ groups are nevertheless encouraged temporarily to define potentials using the same types and sizes, as well as to choose long names.

3.3. Discussion and future

(1) This specification should be extended to the PAW as well as LAPW methods. We suggest to supplement the present set of variables with other variables, not fixed at present.

(2) The specification of a density, Table 3.2, can accommodate the response densities of Density-Functional Perturbation Theory.

Section 4. Specification for files containing the wavefunctions.

4.1 Specification

A NQ NETCDF file "containing the wavefunctions" should contain at least the information needed to build the density from this file. Also, since the eigenvalues are intimately linked to eigenfunctions, it is expected that such a file contain eigenvalues. Of course, files might contain less information than the one required, but still follow the naming convention of NQ. It might also contain more information, of the kind specified in other tables of the present document.

A NQ NETCDF file "containing the wavefunctions" should contain the following set of mandatory information :
(1) The three attributes defined in Table 2.1
(2) The following dimensions from Table 2.4 (dimensions that do not lead to a splitting) :
- character_string_length
- number_of_cartesian_directions
- number_of_vectors
- real_or_complex
- number_of_spinor_components
- number_of_symmetry_operations
- number_of_reduced_dimensions

(3) The following dimensions from Table 2.5 (dimensions that might lead to a splitting) :
- max_number_of_states
- number_of_kpoints
- number_of_spins

(4) In case of a real-space wavefunctions, the following dimensions from Table 2.5 :
- number_of_grid_points_vector1
- number_of_grid_points_vector2
- number_of_grid_points_vector3

or, in case of a wavefunction given in terms of a basis set, the following dimensions from Table 2.5:
- max_number_of_coefficients

(5) The primitive vectors of the cell, as defined in Table 3.1
(6) The symmetry operations, as defined in Table 4.1
(7) The information related to each kpoint, as defined in Table 4.2
(8) The information related to each state (including eigenenergies and occupation numbers), as defined in Table 4.3
(9) In case of basis set representation, the information related to the basis set, and the variable coefficients_of_wavefunctions , as defined in Table 4.4
(10) In case of real-space representation, the variable real_space_wavefunctions, see Table 4.5.
The variables coefficients_of_wavefunctions or real_space_wavefunctions must be the last one, in order not to be limited to 4 GB.

As mentioned in section 1 and 2, such file might contain additional information agreed within NQ, such as any of the variables specified in section 2. Such file might also contain additional

information specific to the software that generated the file. It is not expected that this other software-specific information might be used by another software.

The information might be distributed among different files, thanks to the use of splitting of data for variables :
- max_number_of_states
- number_of_kpoints
- number_of_spins

And, either
- number_of_grid_points_vector1
- number_of_grid_points_vector2
- number_of_grid_points_vector3

or
- max_number_of_coefficients

In case the splitting related to one of these variables is activated, then the corresponding variables in Table 2.6 must be defined. Accordingly, the dimensions of the variables in Tables 4.2, 4.3, 4.4, 4.5 might have to be changed, to accommodate only the segment of data effectively contained in the file.

Variables	Type (index order as in C)	Notes
reduced_symmetry_matrices	integer[number_of_symmetries] [number_of_reduced_dimensions] [number_of_reduced_dimensions]	The "symmorphic" attribute is needed.
reduced_symmetry_translations	double[number_of_symmetries] [number_of_reduced_dimensions]	The "symmorphic" attribute is needed.
Attributes	Type	Notes
symmorphic	char(80)	flag-type attribute

Table 4.1 Variables and attributes to specify the symmetry operations.

Detailed description (table 4.1)

Symmetry operations are defined in real space, with reduced coordinates.
A symmetry operation in real space sends the input point r to the output point r', with

$$r'^{red}_{\alpha} = \sum_{\beta} S^{red}_{\alpha\beta} r^{red}_{\beta} + t^{red}_{\beta}$$

The matrix S, in reduced coordinates, is contained in the array **reduced_symmetry_matrices** of Table 4.1, while the vector t, in reduced coordinates, is contained in the array **reduced_symmetry_translations** of the same Table. There might be a confusion between the two dimensions "number_of_reduced_dimensions" of this variable. In the C ordering, the last one corresponds to the beta index in the above-mentioned formula.

The first symmetry operation must always be unity with translation vector (0,0,0). If all translations are zero, the attribute symmorphic for reduced_symmetry_matrices should be set to "yes".

Variables	Type (index order as in C)	Notes
reduced_coordinates_of_kpoints	double[number_of_kpoints] [number_of_reduced_dimensions]	See possible changes for split files in Table 4.6
kpoint_weights	double[number_of_kpoints]	See description of density construction in section 4.2 See also possible changes for split files in Table 4.6

Table 4.2 Variables that specify the k points

Detailed description (table 4.2)

reduced_coordinates_of_kpoints k-point in relative/reduced coordinates

kpoint_weights k-point integration weights. The weights must sum to 1. See the description of the density construction, section 4.2.

Variables	Type (index order as in C)	Notes
number_of_states	integer[number_of_kpoints]	the attribute "k_dependent" must be defined
eigenvalues	double[number_of_spins] [number_of_kpoints] [max_number_of_states]	The "units" attribute needs to be defined. See also possibles changes for split files in Table 4.6
occupations	double[number_of_spins] [number_of_kpoints] [max_number_of_states]	See also possibles changes for split files in Table 4.6
Attributes	Type (index order as in C)	
k_dependent	char(80)	attribute to number_of_states, flag-type

Table 4.3 Specifications related to each state : occupation numbers and eigenvalues

Detailed description (table 4.3)

number_of_states Number of states for each kpoint, if varying (the attribute **k_dependent** must be set to yes). Otherwise (the attribute **k_dependent** must be set to no), might not contain any information, the actual number of states being set to max_number_of_states.

eigenvalues One-particle eigenvalues/eigenenergies. Should be 0 if unknown.

occupations Occupation numbers. Full occupation for spin-unpolarized cases (number_of_spins = 1 AND number_of_spinor_components = 1) is 2, otherwise it is 1. See section 4.2 .

Variables	Type (index order as in C)	Notes
basis_set	char(character_string_length)	"plane_waves" if a plane wave basis set is used
number_of_coefficients	integer[number_of_kpoints]	The attribute "k_dependent" must be defined (see Table 4.3). Possible splitting, see Table 4.6
reduced_coordinates_of_plane_waves	integer[number_of_kpoints] [max_number_of_coefficients] [number_of_reduced_directions]	The attribute "k_dependent" must be defined (see Table 4.3). See possible modifications for split files in Table 4.6
coefficients_of_wavefunctions	double [number_of_spins] [number_of_kpoints] [max_number_of_states] [number_of_spinor_components] [max_number_of_coefficients] [real_or_complex]	Normalization : 1 per unit cell, see section 3.2 See also possible modifications for split files in Table 4.6

Table 4.4 Specification of wavefunctions in a plane wave basis. Needed only in case "coefficients_of_wavefunctions" will be the array containing the wavefunctions.

Detailed description (table 4.4)

basis_set Type of basis set used if not in a real-space grid. The default and most used value should be ``plane waves".

number_of_coefficients Number of basis function coefficients for each kpoint, if varying (the attribute **k_dependent** must be set to yes). Otherwise (the attribute **k_dependent** must be set to no), might not contain any information, the actual number of states being set to max_number_of_coefficients.

reduced_coordinates_of_plane_waves Plane-wave G-vectors in relative/reduced coordinates. If the attribute **k_dependent** is set to no, then the dimension [number_of_kpoints] must be omitted.

coefficients_of_wavefunctions Wavefunction coefficients. The wavefunctions must be normalized to 1, i.e. the sum of the absolute square of the coefficients of one wavefunction must be 1. See section 4.2 .

Variables	Type (index order as in C)	Notes
real_space_wavefunctions	double [number_of_spins] [number_of_kpoints] [max_number_of_states] [number_of_spinor_components] [number_of_grid_points_vector3] [number_of_grid_points_vector2] [number_of_grid_points_vector1] [real_or_complex]	Normalization : 1 per unit cell, see section 4.2 See also possible modifications for split files in Table 4.6

Table 4.5 Specification of wavefunctions in real space.

Detailed description (table 4.5)

real_space_wavefunctions Wavefunction coefficients. The wavefunctions must be normalized to 1, i.e. the sum of the absolute square of the coefficients of one wavefunction must be 1. See section 4.2 .

Different variables see their dimensions modified, in case the file is split, as described in Section 2.4 (see Tables 2.5 and 2.6). In Table 4.6, we have gathered the variables whose dimensions will change. We have also dimensioned them as if the splitting was done on all the possible dimensions. This will rarely be the case, but intermediate situations can easily be deduced from the data gathered in this table.

Variables	Type (index order as in C)	Notes
reduced_coordinates_of_kpoints	double[my_number_of_kpoints] [number_of_reduced_directions]	
number_of_coefficients	integer[my_number_of_kpoints]	
kpoint_weights	double[my_number_of_kpoints]	
occupations	double[my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states]	
eigenvalues	double[my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states]	Units attribute needed.
real_space_wavefunctions	double [my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states] [number_of_spinor_components] [my_number_of_grid_points_vector1] [my_number_of_grid_points_vector2] [my_number_of_grid_points_vector3] [real_or_complex]	
coefficients_of_wavefunctions	double [my_number_of_spins] [my_number_of_kpoints] [my_max_number_of_states] [number_of_spinor_components] [my_max_number_of_coefficients] [real_or_complex]	
reduced_coordinates_of_plane_waves	integer[my_number_of_kpoints] [number_of_reduced_directions]	

Table 4.6 Variables whose sizes are modified, in case of split files (assuming all splittings have been activated).

Dimensions	Type (index order as in C)	Notes
max_number_of_angular_momenta	integer	
max_number_of_projectors	integer	
Variables	Type (index order as in C)	Notes
gw_corrections	double[number_of_spins] [number_of_kpoints] [max_number_of_states] [real_or_complex]	The "units" attribute needs to be defined. Possibles changes for split files, as in Table 4.6
kb_formfactor_sign	integer[number_of_atomic_species] [max_number_of_angular_momenta] [max_number_of_projectors]	
kb_formfactors	double[number_of_atomic_species] [max_number_of_angular_momenta] [max_number_of_projectors] [number_of_kpoints] [max_number_of_coefficients]	Possible changes for split files, as in Table 4.6
kb_formfactor_derivative	double[number_of_atomic_species] [max_number_of_angular_momenta] [max_number_of_projectors] [number_of_kpoints] [max_number_of_coefficients]	Possible changes for split files, as in Table 4.6

Table 4.7 Optional dimensions and variables that might be needed for some GW/BSE softwares

The variables mentioned in Table 4.7 are optional. Note : they have been introduced in the present specification in prevision of use by some GW/BSE softwares, and might be subject to (heavy ?) revisions in future versions of the specification.

Detailed description (table 4.7)

gw_corrections GW-corrections to one-particle eigenvalues (see Table 4.6). Imaginary part (originating from the non-hermiticity) is optional. Should be 0 if unknown.

max_number_of_angular_momenta The maximum number of angular momenta to be considered for non-local Kleinman-Bylander separable norm-conserving pseudopotentials. If there is no non-local part, set it to 0. If the s channel is the highest angular momentum channel over all atomic species, then set it to 1. If the p channel (resp. d or f) is the highest, set it to 2 (resp. 3 or 4).

max_number_of_projectors The maximum number of projectors for non-local Kleinman-Bylander separable norm-conserving pseudopotentials, over all angular momenta and all atomic species. If there is no non-local part, set it to 0. Most separable norm-conserving pseudopotentials have only one projector per angular momentum channel.

kb_formfactor_sign An array of integers whose value depend on the specific atomic species, angular momentum, and projector. It can have three values : when 0, it means that there is no projector defined for that channel. When +1 or -1, it gives the sign of the Kleinman-Bylander projector for that channel, as explained in section 4.2 .

kb_formfactors
kb_formfactor_derivatives

Kleinman-Bylander form factors in reciprocal space, and their derivative, as explained in section 4.2 .

4.2. Comments

(1) On the density, kpoint weights and occupation numbers.
Supposing $\rho_{n,k}(r)$ to be the partial density at point r (in real space, using reduced coordinates) due to band n at k-point k (in reciprocal space, using reduced coordinates), then the full density at point is obtained thanks to

$$\rho(r_\alpha^{red}) = \sum_{s \in sym} \sum_k w_k \sum_n f_{n,k} \rho_{n,k} \left(S_{s,\alpha\beta}^{red} (r_\beta^{red} - t_{s,\beta}^{red}) \right)$$

where w_k is contained in the array "kpoint_weights" of Table 4.2, and
$f_{n,k}$ is contained in the array "occupations" of Table 4.3 .
This relation generalizes to the collinear spin-polarized case, as well as the non-collinear case by taking into account the "number_of_components" defined in Table 2.5 , and the direction of the magnetization vector.

(2) On the Kleinman-Bylander form factors.
One can always write the non-local part of Kleinman-Bylander pseudopotential (reciprocal space) in the following way :

$$\mathrm{v}_{\mathrm{nonloc}}^{\mathrm{KB}}(\vec{K},\vec{K}') = \sum_s \left[\sum_{a(s)} e^{-i(\vec{K}-\vec{K}')\vec{\tau}_a} \right] \left[\sum_{lp} P_l(\hat{K}.\hat{K}') F_{slp}^*(K) S_{slp} F_{slp}(K') \right]$$

with $\vec{K} = \vec{k} + \vec{G}$, $\vec{k}$ is one of the kpoints (see Table 4.2), $\vec{G}$ is a vector of the reciprocal lattice, the list of reduced coordinates of which can be found in the variable reduced_coordinates_of_plane_waves of Table 4.4. K is the module of $\vec{K}$ and $\hat{K}$ its direction. $\vec{\tau}_a$ is the atomic position of atom a belonging to species s. $P_l(x)$ is the Legendre polynomial of order l . $F_{slp}(K)$ is the Kleinman-Bylander form factor for species s, angular momentum l, and number of projector p . S_{slp} is the sign of the dyadic product $F_{slp}^*(K)F_{slp}(K')$. The sum on a (s) runs over all atoms of atomic species s, l runs over all the pseudopotential angular momentum components of the atomic species s, and p runs over the number of projectors allowed for a specific angular momentum channel of atomic species s. The additional variable kb_formfactor_derivative is equal to $\dfrac{dF_{slp}(K)}{dK}$

4.3. Discussion and future

(1) This specification should be extended to the PAW as well as LAPW methods. We suggest to supplement the present set of variables (so, using the same names) with other variables to be fixed later.
(2) The specification of wavefunctions, Table 4.4 as well as 4.5, can accommodate the response wavefunctions of Density-Functional Perturbation Theory. On the contrary, the response eigenenergies (actually a hermitian matrix of Lagrange multipliers) cannot be accommodated by the "eigenvalues" array of Table 4.3 .
(3) Additional information has to be provided to deal with magnetic symmetry operations, in both collinear and non-collinear cases.

Section 5. Pseudopotential / PAW set up files

At the Louvain-la-neuve meeting, three formats based on XML have been reported, as well as other text formats. A combined strategy has been decided: writing a conversion utility to cope with existing files (this is the short-term action); then writing routines to facilitate the use of XML-based formats, in view of hoped unification (this is the long-term action).

The different formats can be found on the Web site
http://www.pcpm.ucl.ac.be/workshop_files/file_format

In particular, note :

(1) the XML format, for PAW set up files (a reduced set of these data is able to specify ultra-soft as well as norm-conserving pseudopotentials) from Jens Mortensen :
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-mortensen_1.ps
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-mortensen_2.ps

(2) the XML format, for norm-conserving pseudopotential files, from J. Junquera and A. Garcia :
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-garcia.txt ,
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-garcia.tgz ,
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-junquera_v2.ppt ,
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-xml-junquera.txt

(3) the "XML-like" UPF format presented by P. Gianozzi, used in the ESPRESSO package :
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-upf-giannozzi.txt
http://www.pcpm.ucl.ac.be/workshop_files/file_format/psps-upf-giannozzi.f90

Section 6. Files containing crystalline structure and atomic geometries

They were not discussed in the Louvain-la-Neuve meeting.
We nevertheless give a list of Web sites, for further work ...

(1) CML (Chemical Markup Language) is a metalanguage, based on XML, that allows to specify crystalline structures.
Original site : http://www.xml-cml.org
CMLCore : http://wwmm.ch.cam.ac.uk/moin/CmlCore

(2) OpenBabel is a tool to convert files describing chemical species from one format to another.
http://openbabel.sourceforge.net/wiki/index.php/Main_Page
Many different I/O format for chemical species are mentioned at
http://openbabel.sourceforge.net/wiki/index.php/Babel

Section 7. Files with other contents relevant to NANOQUANTA.

Files containing the following information are also relevant to the NANOQUANTA activities:
- dielectric matrices (frequency-dependent), susceptibility matrices, screening matrices (typically for GW calculations)
- derivatives of the total energy, typically obtained from Density-Functional Perturbation Theory (dynamical matrices, Born effectives charges, elastic constants, piezoelectric constants, etc ...).
- electron-phonon coupling elements

Some of the existing file formats to contain these data were presented at the Louvain-la-Neuve meeting. The different formats can be found on the Web site
http://www.pcpm.ucl.ac.be/workshop_files/file_format

Appendix A. Presentation of Nanoquanta and the ETSF

Nanoquanta is a Network of Excellence funded by the European Commission's Sixth Framework Programme (FP6). The Network performs fundamental physics research within the Third Thematic Priority of the *"Integrating and Strengthening the European Research Area"* activity of the FP6: nanotechnologies and nanosciences, knowledge-based multifunctional materials and new production processes and devices, known as "NMP".

The Network is intended to operate from 1 June 2004 to 31 May 2008 and consists of 10 nodes and over 100 researchers.

Network Objectives

The *Nanoquanta* Network of Excellence integrates and develops the research capabilities of ten European teams in the field of the fundamental science of nanoscale systems and advanced materials, exploiting the powerful combination of quantum-mechanical theory and computer simulation to make contact with nanoscience experimental studies and also directly with technologically relevant electronic, dynamic and optical processes.

Through scientific publications, workshops and conferences, training, and contact with experimental and applied research groups in European universities, research institutions and other organizations, *Nanoquanta's* ongoing work is disseminated widely.

The Network is intended to evolve into a European Theoretical Spectroscopy Facility (ETSF), with strong links with a wide range of research groups and available for collaboration with users from across science and industry in projects concerned with nanoscale systems and advanced materials. Further information can be obtained from the Web site http://www.cmt.york.ac.uk/nanoquanta/

Partners in the Network

University of York (Team Leader and network coordinator: Prof. R Godby, Department of Physics)

Fritz-Haber-Institut, Berlin (Team Leaders: Prof. Matthias Scheffler, Dr. A. Schindlmayr and Dr. P. Rinke, Theory Department)

Freie Universität, Berlin (Team Leader: Prof. E.K.U. Gross, Department of Physics)

Friedrich-Schiller-Universität, Jena (Team Leader: Prof. F. Bechstedt and Dr. J Furthmüller, Institut für Festkörpertheorie und Theoretische Optik)

Université Catholique de Louvain (Team Leader: Prof. X. Gonze, Unité de Physico-Chimie et de Physique des Matériaux)

Lunds Universitet (Team Leaders: Prof. C.-O. Almbladh and Prof. Ulf von Barth, Department of Solid-State Theory)

Universitá degli Studi di Milano (Team Leader: Prof. G. Onida, Department of Physics)

Laboratoire des Solides Irradiés (Paris) (Team leader and network deputy coordinator: Dr. L. Reining)

Consiglio Nazionale delle Ricerche - Istituto Nazionale per la Fisica della Materia (Rome) (Team Leader: Prof. R. Del Sole, Department of Physics, University of Rome *Tor Vergata*)

Universidad del País Vasco / Euskal Herriko Unibertsitatea (San Sebastián) (Team Leaders: Prof. P. Echenique and Prof. A. Rubio, Facultad de Quimicas and Donostia International Physics Center)

Appendix B. Historical overview

The idea of standardization of file formats is not new in the Electronic Structure community, but proved difficult to realize without a formal organization gathering code developers. As an early attempt, it was explicitly stated in 2002, in the manifest of the "Free software project for atom-scale simulations" (FSAtom) (http://www.fsatom.org/index.php):
" <The FSAtom should> through workgroups, organize the collaboration between developers: file exchange, code testing, definition of objects, exchange of development tools, exchange of expertise ... ;".
The most noticeable outcome of the FSAtom along this line is the creation of a discussion group on pseudopotential file formats, that was quite active in 2003. Two file formats (one for PAW set-up and one for norm-conserving pseudopotentials) have been proposed as a consequence of these discussions.

In the framework of NANOQUANTA, the standardization of file formats is an explicit goal of the project, to be achieved in the framework of the Work Package 9 "Integration of theory and code development", lead by the Integration Team 9.

Towards this goal, several preliminary steps have been achieved :
- At the Maratea meeting of the NANOQUANTA network (September 2004), a discussion session on the objectives of Work Package 9 "Theory and code integration" was lead, with preliminary discussion of coding rules, and the set-up of a list of interested persons, and one contact person per node.
- A list of the programs used by NANOQUANTA groups has been established, with explicit mention of their licensing scheme, and with the mention of the control that NANOQUANTA members have on their development.
- A draft list of coding rules for NANOQUANTA has been written, and is available on the nanoquanta_admin smartgroup, as well as on the Web site of the Louvain-la-Neuve mini-workshop
http://www.pcpm.ucl.ac.be/workshop_files/Guidelines_v2.pdf

In particular, in the latter document, the use of NetCDF for large files is recommended, instead of binary (unformatted) files, that are not portable across platforms and not flexible.

Appendix C. Methodology

One has to consider separately the set of data to be included in each of different types of files, from their representation. Concerning the latter, one encounters simple text files, as well as binary files, XML-structured files, NetCDF files, etc … It was already decided previously (Maratea Sept. 2004) to evolve towards formats that deal appropriately with the self-description issue, i.e. XML and NetCDF. The inherent flexibility of these representations will also allow to evolve specific versions of each type of files progressively, and refine earlier working proposals. The same direction has been adopted by other groups of code developers.

However, the real work to be done now involves the structuration of each set of data, and the definition of their content (minimal/maximal). We have to start from already existing (and used) types of files, and discuss better representations, that could be adopted by a large set of softwares. We might as well just choose one existing file format, or decide to write interconversion utilities, or a mix of these.

As a preparation to the Louvain-la-neuve mini-workshop, each existing type of files, specific of the NANOQUANTA domain of activity, was described (see the workshop Web site). In order to be accurate, each type of file was described thanks to a documented piece of **code**, either in Fortran, in C, or using NetCDF routine calls, or a DTD (for XML). Also, each variable was described properly (type and meaning).

The Louvain-la-neuve mini-workshop centered only on the files :
- that are created by a software, to be input to some other (or the same) software
- directly related to the "core business" of NANOQUANTA.

In particular, one discussed the files that contain :
(1) pseudopotentials or PAW atomic data
(2) wavefunctions and eigenenergies (esp. plane wave representation)
(3) density/potential
(4) susceptibility and dielectric matrices
(5) matrix elements, or derivatives of wavefunctions, of total energies, ...
This list is not exhaustive.
On the contrary, the participants of the Louvain-la-neuve mini-workshop did neither examine the input files (produced by humans), nor the files that should be read by some graphical software (outside the core business of NANOQUANTA). This might be the subject of another mini-workshop.

Appendix D.
Some information on the NetCDF size limitation

(The following information has been provided shortly after the workshop by Valerio Olevano)
To summarize :
1) The 2GB limit is firstly a FILE-SIZE limit of operating systems on 32-bits machine (and some non-updated 64-bits old-operating-systems). And this cannot be overcome, even splitting wavefunctions into nbands*nkpoints variables.
2) Assuming your machine can store >2GB files, the NetCDF has in general a limit of 4GB.
BUT even with the actual version you can store in NetCDF at least one variable (the last) up to Terabytes, and probably in future this will be extended to also the non last variables.

More detailed discussion :
4.4 NetCDF 64-bit Offset Format Limitations
Although the 64-bit offset format allows the creation of much larger NetCDF files than was possible with the classic format, there are still some restrictions on the size of variables.
It is important to note that without Large File Support (LFS) in the operating system, it is impossible to create any file larger than 2 GBytes. Assuming an operating system with LFS, the following restrictions apply to the NetCDF 64-bit offset format.
No fixed-size variable can require more than 2^32 - 4 bytes (i.e. 4GB - 4 bytes, or 4,294,967,292 bytes) of storage for its data, unless it is the last fixed-size variable and there are no record variables. When there are no record variables, the last fixed-size variable can be any size supported by the file system, e.g. terabytes.
A 64-bit offset format NetCDF file can have up to 2^32 - 1 fixed sized variables, each under 4GB in size. If there are no record variables in the file the last fixed variable can be any size.
No record variable can require more than 2^32 - 4 bytes of storage for each record's worth of data, unless it is the last record variable. A 64-bit offset format netCDF file can have up to 2^32 - 1 records, of up to 2^32 - 1 variables, as long as the size of one record's data for each record variable except the last is less than 4 GB - 4.
Note also that all netCDF variables and records are padded to 4-byte boundaries.

Appendix E.
List of changes of this version with respect to v1.0 of Feb 12, 2006

Changes from v1.2 to v1.3 :

Page 2 and 34 : add the Appendix F.

Page 12 and 13 : the indices to be used for "atom_species" have been specified (from 1 to "number_of_atom_species".

Tables 2.1, 2.2, 2.3, changed "Variables" to "Attributes" in the title of the first column

Tables 2.4, 2.5, changed "Variables" to "Dimensions" in the title of the first column and the table caption

Table 2.6, split the "Dimensions" from the "Variables", and changed the title of the first column accordingly

Page 13, table 2.9: "kinetic_energy_cutoff" has the "units" attribute.

Page 16, table 3.3: all variables have the "units" attribute.

Page 20, table 4.1: the "symmorphic" attribute appears in the notes of the first row.

Page 25, table 4.6, changed "number_of_coefficients" from double to integer ; unit attribute was needed for variable "eigenvalues".

Table 4.7, "max_number_of_angular_momenta" and "max_number_of_projectors" are dimensions, and not variables. This table has been split, and the title of the first column changed accordingly.

Prior changes (from 1.0 to 1.2) :

Page 2 : The first section was missing in the index. The index mentioned a non-existent Appendix E (for pseudopotentials / PAW).
The index now correctly refers to the present Appendix E, and includes correctly all the sections of the document.

Page 6 : Concerning the "Conventions" , the actual address where the specification is available has been mentioned.

Page 12 : Table 2.7 . Five occurrences of "number_of_atoms_species" have been replaced by the correct "number_of_atom_species"

Page 14 : Section 2.6 is new.

Page 16 : Table 3.1 . The order of dimensions of the "primitive_vectors" array was incorrect, and has been corrected.

Page 20 : Detailed description of Table 4.1, 6th line . "reduced_symmetry_vectors" has been replaced by the correct "reduced_symmetry_translations".

Page 22 : Table 4.4 . The dimension "max_number_of_coefficients" was missing for the "reduced_coordinates_of_plane_waves".

Page 22 : Table 4.4 . Concerning the array " reduced_coordinates_of_plane_waves", the role of the attribute "k_dependent" has been specified.

Page 22 : Detailed description of the array " reduced_coordinates_of_plane_waves", the role of the attribute "k_dependent" has been specified.

Page 24 : Table 4.6 . "occupation_numbers" has been replaced by the correct "occupations".

Page 24 : Table 4.6 . In the description of the variable "corefficients_of_wavefunctions", a typo has been corrected : "corefficients" has been replaced by "coefficients".

Page 25 : Table 4.7, the note about "kb_formfactor_sign" has been removed.

The present Appendix E has been created.

Then, in the whole document, dozen of typos have been corrected. The occurences of "NETCDF" or "netcdf" have been normalized to "NetCDF". Modifications of the text (style, additional comments) have also been done, without affecting the actual specification. Reference to a specific version of the present document have been eliminated, except in the title page, to ease the maintenance of the document.

Appendix F.
List of things to be debated at the NQ meeting in September 2006

Define "primitive_vectors" in section 2 instead of 3.
Define a NQ/ETSF NetCDF format for crystalllographic data files.