```
#
# Generic config file for ABINIT (documented template)
#
# After editing this file to suit your needs, you may save it as
# "~/.abinit/build/<hostname>.ac" to keep these parameters as per-user
# defaults. Just replace "<hostname>" by the name of your machine,
# excluding the domain name.
#
# Example: if your machine is called "myhost.mydomain", you will save
#          this file as "~/.abinit/build/myhost.ac".
#
# You may put this file at the top level of an ABINIT source tree as well,
# in which case its definitions will apply to this particular tree only.
#
# Hint: If you do not know the name of your machine, just type "hostname".
#

#
# IMPORTANT NOTES
#
# 1. It may happen that this file is outdated. In such a case, the best is
#    to refer to the command-line options of the "configure" script. To
#    obtain a complete list of these options, just type "./configure --help"
#    from the top source directory of ABINIT.
#
# 2. Setting CPPFLAGS, CFLAGS, CXXFLAGS, or FCFLAGS manually is not
#    recommended and will override any other mechanism involved in their
#    configuration. A gentler way to do is to use the "--with-*-optflags"
#    command-line options of the "configure" script. See below for details.
#
# 3. Do not forget to remove the leading "#" on a line when you customize
#    an option.
#

# ---------------------------------------------------------------------- #
# Global build options                                                   #
# ---------------------------------------------------------------------- #

# Enable 64-bit compiler flags (default is no)
#
# Note:
#
#     * when necessary, fine-grained tuning may be achieved by setting
#       the CPPFLAGS_64BITS, CFLAGS_64BITS, CXXFLAGS_64BITS, and
#       FCFLAGS_64BITS variables manually
#
#enable_64bit_flags="yes"
enable_64bit_flags="yes"

# Specify where to apply 64-bit flags (default is unset)
#
# Allowed values:
#
#     * ar  : apply to archiver
#     * cpp : apply to preprocessor
#     * cc  : apply to C compiler
#     * cxx : apply to C++ compiler
#     * fc  : apply to Fortran compiler
#
# WARNING: this is not implemented yet
#
#with_64bit_flags="cc:cxx"
```

```
# Enable debug mode (default is yes)
#
# Allowed values:
#
#    * no       : strip debugging symbols
#    * yes      : keep debugging symbols and allow for user-defined flags
#    * enhanced : disable optimizations and debug verbosely
#    * paranoid : enhanced debugging with additional warnings
#    * naughty  : paranoid debugging with array bound checks
#
# Levels other than no and yes are "profile mode" levels in which
# user-defined flags are overriden and optimizations disabled (see
# below)
#
# Note: debug levels are incremental, i.e. the flags of one level are
# appended to those of the previous ones
#
#enable_debug="yes"

# Enable compiler hints (default is yes)
#
# Allowed values:
#
#    * no       : do not apply any hint
#    * yes      : apply all available hints
#
#enable_hints="no"

# Select optimization level whenever possible (default is standard,
# except when debugging is in profile mode - see above - in which case
# optimizations are turned off)
#
# Supported levels:
#
#    * no         : disable optimizations
#    * yes        : enable optimizations with user-defined flags
#    * safe       : build slow and very reliable code
#    * standard   : build fast and reliable code
#    * aggressive : build very fast code, regardless of reliability
#
# Levels other than no and yes are "profile mode" levels in which
# user-defined flags are overriden
#
# Note:
#
#    * this option is ignored when debugging is in profile mode
#
#enable_optim="aggressive"
enable_optim="yes"

# Where to install ABINIT (default is /usr/local)
#
#prefix="${HOME}/hpc"

# ---------------------------------------------------------------------- #
# Preprocessing                                                          #
# ---------------------------------------------------------------------- #

# C preprocessor (should not be set in most cases)
#
#CPP="/usr/bin/cpp"

# Set preprocessing options for debugging (overrides auto-detection)
```

```
#
#with_cpp_dbgflags="-DFOO"

# Set preprocessing options for optimizations (overrides auto-detection)
#
#with_cpp_optflags="-DBAR"

# ---------------------------------------------------------------------------- #
# C support                                                                    #
# ---------------------------------------------------------------------------- #

# C compiler
#
CC="/usr/vac/bin/xlc_r"

# Set debug flags for C source files (overrides auto-detection)
#
#with_cc_dbgflags="-g"

# Set optimizations for C source files (overrides auto-detection)
#
#with_cc_optflags="-O3"
with_cc_optflags="-qarch=pwr7 -qtune=pwr7 -q64 -O3"

# Prepend debug link flags for C binaries (default is unset)
#
#with_cc_ld_dbgflags=""

# Prepend optimization link flags for C binaries (default is unset)
#
#with_cc_ld_optflags="-Bstatic"

# Append debug libraries to link C binaries (default is unset)
#
#with_cc_ld_dbglibs=""

# Append optimization libraries to link C binaries (default is unset)
#
#with_cc_ld_optlibs="-lrt"

# ---------------------------------------------------------------------------- #
# C++ support                                                                  #
# ---------------------------------------------------------------------------- #

# C++ compiler
#
CXX="/usr/vacpp/bin/xlC_r"

# Set debug flags for C++ source files (overrides auto-detection)
#
#with_cxx_dbgflags="-g"

# Set optimizations for C++ source files (overrides auto-detection)
#
#with_cxx_optflags="-O3"
with_cxx_optflags="-qarch=pwr7 -qtune=pwr7 -q64 -O3"

# Prepend debug link flags for C++ binaries (default is unset)
#
#with_cxx_ld_dbgflags=""

# Prepend optimization link flags for C++ binaries (default is unset)
#
```

```
#with_cxx_ld_optflags="-Bstatic"

# Append debug libraries to link C++ binaries (default is unset)
#
#with_cxx_ld_dbglibs=""

# Append optimization libraries to link C++ binaries (default is unset)
#
#with_cxx_ld_optlibs="-lblitz"

# ---------------------------------------------------------------------- #
# Fortran support                                                        #
# ---------------------------------------------------------------------- #

# Fortran compiler
#
FC="/usr/bin/xlf90_r -F/home/jmht/xlf.cfg.61"

# Set debug flags for Fortran source files (overrides auto-detection)
#
#with_fc_dbgflags="-g"

# Set optimizations for Fortran source files (overrides auto-detection)
#
#with_fc_optflags="-O3"
with_fc_optflags="-qarch=pwr7 -qtune=pwr7 -q64 -O3 -qhot -lxlopt -qstrict -qcache=auto -bmaxdata:2000000000
-qmaxmem=-1 -qsuppress=1500-036"

# Prepend debug link flags for Fortran binaries (default is unset)
#
#with_fc_ld_dbgflags=""

# Prepend optimization link flags for Fortran binaries (default is unset)
#
#with_fc_ld_optflags="-Bstatic"

# Append debug libraries to link Fortran binaries (default is unset)
#
#with_fc_ld_dbglibs=""

# Append optimization libraries to link Fortran binaries (default is unset)
#
#with_fc_ld_optlibs="-lsvml"

# Use C clock instead of Fortran clock for timings (default is no)
#
#enable_cclock="yes"

# Choose whether to read file lists from standard input or "ab.files"
# (default is yes = standard input)
#
#enable_stdin="no"

# Wrap Fortran compiler calls (default is auto-detected)
# Combine this option with enable_debug="yes" to keep preprocessed source
# files (they are removed by default, except if their build fails)
#
#enable_fc_wrapper="yes"
enable_fc_wrapper="no"

# ---------------------------------------------------------------------- #
# Build customization                                                    #
# ---------------------------------------------------------------------- #
```

```
# Enable the build of macroave (default is yes)
#
#enable_macroave="no"

# Set per-directory Fortran optimizations (useful when a Fortran compiler
# crashes during the build)
#
# Note: this option is not available through the command line
#
#fcflags_opt_95_drive="-O0"

# ---------------------------------------------------------------------- #
# Libraries and linking                                                  #
# ---------------------------------------------------------------------- #

# Set archiver name
#
#AR="xiar"
AR="/usr/bin/ar"

# Set debug flags for archiver (default is unset)
#
#with_ar_dbgflags=""

# Set optimizations for archiver (default is unset)
#
with_ar_optflags="-X 64"

#jmht
NM="/usr/bin/nm -B -X 64"

# ---------------------------------------------------------------------- #
# MPI support                                                            #
# ---------------------------------------------------------------------- #

# Determine whether to build parallel code (default is unset)
#
#  * enable_mpi=unset  : let the configure script auto-detect MPI support
#  * enable_mpi=no     : disable MPI support (bypassing autodetection)
#  * enable_mpi=yes    : enable MPI support (bypassing autodetection)
#
# If left unset, the build system will take all appropriate decisions by
# itself, and MPI will be enabled only if the build environment supports
# it
#
enable_mpi="no"

# Activate parallel I/O (default is no)
# WARNING: this feature is still under development
#enable_mpi_io="yes"

# Activate untested features of parallel I/O (default is no)
# WARNING: USE AT YOUR OWN RISKS!
#enable_mpi_io_test="yes"

# Activate MPI time tracing (default is no)
# WARNING: this feature is still under development
#enable_mpi_trace="yes"

# Include flags for the MPI library (default is unset)
#
#with_mpi_includes="-I/usr/local/include"
```

```
# Set MPI standard level (default is auto-detected)
# Note: usually 1 for MPICH or LAM, 2 for OpenMPI
#
#with_mpi_level="2"

# Link flags for the MPI library (default is unset)
#
#with_mpi_libs="-L/usr/local/lib -lmpi"

# Set MPI Runner for the tests (default is unset)
#
#with_mpi_runner="/usr/local/bin/mpiexec"

# Convenience option: try to set MPI parameters by looking into the
# specified directory
#
# Note 1: the build system expects to find subdirectories named bin/, lib/,
# include/ under the prefix
#
# Note 2: this option is mutually exclusive with the other with_mpi_*
# options
#
#with_mpi_prefix="/usr/local/openmpi-gcc"

# ------------------------------------------------------------------------ #
# External libraries                                                       #
# ------------------------------------------------------------------------ #

# Enable FFTW library support (default is no)
#
#enable_fftw="yes"

# Enable threaded FFTW library support (default is no)
#
# Note: will be ignored if "enable_fftw" is set to "no"
#
#enable_fftw_threads="yes"

# Include flags for the FFTW library (default is unset)
#
#with_fftw_includes="-I/usr/local/include/fftw"

# Link flags for the FFTW library (default is unset)
#
#with_fftw_libs="-L/usr/local/lib/fftw -lfftw3"

                # ------------------------------ #

# Enable GSL support (default is no)
#
#enable_gsl="yes"

# Include flags for the GSL (default is unset)
#
#with_gsl_includes="-I/usr/local/include/gsl"

# Link flags for the GSL (default is unset)
#
#with_gsl_libs="-L/usr/local/lib/gsl -lgsl"

                # ------------------------------ #
```

```
# Link flags for BLAS and LAPACK linear algebra libraries (default is unset)
#
# Note: internal versions will be built if left unset
#
#with_linalg_libs="-lessl /home/jmht/abinit-6.0.4/build/prereqs/linalg/lapack/liblapack.a"
#with_linalg_libs=""

# Type of linear algebra libraries to use (default is unset)
# Under development - USE AT YOUR OWN RISKS!
#
# Supported libraries:
#
#    * acml      (not implemented)
#    * asl       (not implemented)
#    * atlas     (untested)
#    * cxml      (not implemented)
#    * essl      (should work)
#    * mkl       (not implemented)
#    * mlib      (not implemented)
#    * sgimath   (not implemented)
#    * sunperf   (not implemented)
#
#with_linalg_type="atlas"

# Enable SCALapack support (default is no)
#
#enable_scalapack="yes"

# Include flags for the SCALapack library (default is unset)
#
#with_scalapack_includes="-I/usr/local/include/scalapack"

# Link flags for the SCALapack library (default is unset)
#
#with_scalapack_libs="-L/usr/local/lib/scalapack -lscalapack -lblacs"

                    # ------------------------------ #

# Enable PAPI support (default is no)
#
#enable_papi="yes"

# Include flags for the PAPI library (default is unset)
#
#with_papi_includes="-I/usr/local/include/papi"

# Link flags for the PAPI library (default is unset)
#
#with_papi_libs="-L/usr/local/lib/papi -lpapi"

# ---------------------------------------------------------------------------- #
# Plug-ins                                                                     #
# ---------------------------------------------------------------------------- #

#
# IMPORTANT NOTE: the source tarballs of plug-ins are downloaded from
#                 internet into ~/.abinit/tarballs
#

# Look for the plug-in components in <prefix>/bin, <prefix>/include
# and <prefix>/lib (default is unset)
#
#with_plugins_prefix="/opt/etsf/abinit-plugins"
```

```
# Set local repository where to look for plug-in tarballs (default is
# ~/.abinit/tarballs)
#
#with_plugins_tardir="/opt/etsf/abinit-plugins/tarballs"

                # ------------------------------ #

# Enable BigDFT support (default is yes)
#
#enable_bigdft="no"
enable_bigdft="no"

# Include flags for the BigDFT library (default is unset)
#
#with_bigdft_includes="-I/usr/local/include/bigdft"

# Link flags for the BigDFT library (default is unset)
#
#with_bigdft_libs="-L/usr/local/lib/bigdft -lbigdft -lpoissonsolver"

                # ------------------------------ #

# Enable FoX support (default is no)
#
#enable_fox="yes"

# Include flags for the FoX library (default is unset)
#
#with_fox_includes="-I/usr/local/include/fox"

# Link flags for the FoX library (default is unset)
#
#with_fox_libs="-L/usr/local/lib/fox -lFoX_wxml -lFoX_wcml -lFoX_utils -lFoX_sax -lFoX_common -lfsys"

                # ------------------------------ #

# Enable Nanoquanta/ETSF I/O library support (default is yes)
#
#enable_etsf_io="no"
enable_etsf_io="no"

# Include flags for the Nanoquanta/ETSF I/O library (default is unset)
#
#with_etsf_io_includes="-I/opt/etsf/include"

# Link flags for the Nanoquanta/ETSF I/O library (default is unset)
#
#with_etsf_io_libs="-L/opt/etsf/lib -letsf_io"

                # ------------------------------ #

# Enable Nanoquanta/LibXC library support (default is yes)
#
#enable_libxc="no"
enable_libxc="no"

# Include flags for the Nanoquanta/LibXC library (default is unset)
#
#with_libxc_includes="-I/opt/etsf/include"

# Link flags for the Nanoquanta/LibXC library (default is unset)
#
```

```
#with_libxc_libs="-L/opt/etsf/lib -llibxc"

                # ------------------------------ #

# Enable NetCDF support (default is yes)
#
#enable_netcdf="no"
enable_netcdf="no"

# Include flags for the NetCDF library (default is unset)
#
#with_netcdf_includes="-I/usr/local/include/netcdf"

# Link flags for the NetCDF library (default is unset)
#
#with_netcdf_libs="-L/usr/local/lib/netcdf -lnetcdf"

                # ------------------------------ #

# Enable Wannier90 support (default is yes)
#
#enable_wannier90="no"
enable_wannier90="no"

# Full path for the Wannier90 binary (default is unset)
#
#with_wannier90="/usr/local/bin/wannier90"

# Include flags for the Wannier90 library (default is unset)
#
#with_wannier90_includes="-I/usr/local/include/wannier90"

# Link flags for the Wannier90 library (default is unset)
#
#with_wannier90_libs="-L${HOME}/lib/wannier90 -lwannier90"

# ---------------------------------------------------------------------------- #
# Developer options                                                            #
# ---------------------------------------------------------------------------- #

# Note: all the following options are disabled by default
#       (i.e. they have the complementary values to those displayed)

# Enable bindings (caliste)
#
#enable_bindings="yes"

# Enable CUDA (mancini)
#enable_cuda="yes"

# Enable error handlers (gmatteo)
#
#enable_error_handlers="yes"

# Enable exports (pouillon)
#
#enable_exports="yes"

# Enable double precision for GW calculations (gmatteo)
#
#enable_gw_dpc="yes"

# ---------------------------------------------------------------------------- #
```

```
# Maintainer options                                                    #
# ---------------------------------------------------------------------- #

# Activate support for all plug-ins (default is unset)
#
# Warning: setting this option to "no" will disable plug-in support
#
#enable_all_plugins="yes"

# Enable dependency tracking (Automake feature)
# Though the exact outcome of this is not known at present, you may
# turn it on if you have a taste for adventure ;-)
#
#enable_dependency_tracking="yes"

# Activate nightly build support (default is no)
#
#enable_nightly="yes"

# Activate test abirules (default is no)
#
#enable_test_abirules="yes"

# Set timeout for automatic tests (default is 600)
#
# Note: this option requires Nightly to be activated
#
#with_timeout="1200"

# Fortran compiler vendor to be used by the build system (default is unset)
#
# Note: do not use if you don't know what it is about
#
#with_fc_vendor="dummy"
with_fc_vendor="ibm"

# Fortran compiler version to be used by the build system (default is unset)
#
# Note: do not use if you don't know what it is about
#
#with_fc_version="0.0"
```