```
cmake_minimum_required(VERSION 3.0)

if(POLICY CMP0048)
  cmake_policy(SET CMP0048 NEW)
endif()

if(NOT PROJECT_NAME STREQUAL fabm)
  project(fabm VERSION 1.0.4 LANGUAGES Fortran)
endif()

# List of contributing institutes that should be included in compilation.
# When adding new institutes to the source tree, please add them here as well.
# You can exclude institute directories from compilation by commenting them out.
set(DEFAULT_INSTITUTES
    akvaplan     # Akvaplan-niva, Norway
    au           # University of Aarhus, Denmark
    bb           # Bolding & Bruggeman - formerly Bolding & Burchard
    csiro        # Commonwealth Scientific and Industrial Research Organisation, Australia
    ersem        # European Regional Seas Ecosystem Model
    examples     # Examples supplied with FABM itself
    gotm         # Models ported from original GOTM/BIO library
    iow          # Leibniz Institute for Baltic Sea Research, Germany
    jrc          # EC - Joint Research Centre, Ispra, Italy
    msi          # Marine Systems Institute, Tallinn University of Technology, Estonia
    nersc        # Nansen Environmental and Remote Sensing Center
    niva         # Norwegian Institute for Water Research, Norway
    pclake       # The PCLake model - reference implementation
    pml          # Plymouth Marine Laboratory, United Kingdom
    selma        # Simple EcoLogical Model for the Aquatic - PROGNOS
    su           # Swansea University, United Kingdom
    uhh          # University of Hamburg, Germany
)

# Ensure FABM_INSTITUTES tracks the up-to-date DEFAULT_INSTITUTES list unless the user manually customized FABM_INSTITUTES.
if(NOT FABM_DEFAULT_INSTITUTES OR NOT FABM_INSTITUTES)
  set(FABM_DEFAULT_INSTITUTES "${DEFAULT_INSTITUTES}" CACHE INTERNAL "Default institutes at the time FABM_INSTITUTES was initialized or updated automatically.")
endif()
set(FABM_INSTITUTES ${DEFAULT_INSTITUTES} CACHE STRING "Institute directories to include during compilation.")
if(FABM_INSTITUTES STREQUAL FABM_DEFAULT_INSTITUTES)
  # FABM_INSTITUTES matches its original default value; update both with the current DEFAULT_INSTITUTES list.
  set(FABM_INSTITUTES ${DEFAULT_INSTITUTES} CACHE STRING "Institute directories to include during compilation." FORCE)
  set(FABM_DEFAULT_INSTITUTES ${DEFAULT_INSTITUTES} CACHE INTERNAL "Default institutes at the time FABM_INSTITUTES was initialized or updated automatically.")
endif()

# Use solution folders in IDEs
set_property(GLOBAL PROPERTY USE_FOLDERS ON)

# Create fabm_library.F90 with all active institutes,
# and set final paths to institute source directories (FABM_${INSTITUTE_UC}_DIR)
set(FABM_LIBRARY_USE "")
set(FABM_LIBRARY_ADD "")
set(ACTIVE_INSTITUTES "")
foreach(institute ${FABM_INSTITUTES})
  # First retrieve the effective source directory
  set(DEFAULT_INSTITUTE_BASE "${CMAKE_CURRENT_LIST_DIR}/models/${institute}")
  string(TOUPPER ${institute} INSTITUTE_UC)
  find_path(FABM_${INSTITUTE_UC}_BASE
    NAMES CMakeLists.txt
    HINTS "${DEFAULT_INSTITUTE_BASE}"
    DOC "Path to source code of ${institute} models"
    NO_DEFAULT_PATH
    NO_CMAKE_FIND_ROOT_PATH
  )
  if(EXISTS "${DEFAULT_INSTITUTE_BASE}/CMakeLists.txt")
    mark_as_advanced(FABM_${INSTITUTE_UC}_BASE)
  endif()

  if(FABM_${INSTITUTE_UC}_BASE)
    # Either a custom institute directory has been specified, or the default institute directory exists.
    if (EXISTS "${FABM_${INSTITUTE_UC}_BASE}")
      # The institute directory exists
    set(FABM_LIBRARY_USE "${FABM_LIBRARY_USE}
   use ${institute}_model_library")
    set(FABM_LIBRARY_ADD "${FABM_LIBRARY_ADD}
      call self%add(${institute}_model_factory,'${institute}')")
    set(ACTIVE_INSTITUTES ${ACTIVE_INSTITUTES} ${institute})
    elseif(NOT FABM_${INSTITUTE_UC}_BASE STREQUAL DEFAULT_INSTITUTE_BASE)
      # The institute directory does not exist and it was explicitly specified (because different from default).
      message(FATAL_ERROR "Source directory ${FABM_${INSTITUTE_UC}_BASE} specified for institute ${institute} not found.")
    endif()
  else()
    #message(STATUS "${institute} will not be compiled because FABM_${INSTITUTE_UC}_BASE is not set. Please visit http://fabm.net/${institute} for more information.")
  endif()
endforeach(institute)
configure_file(${CMAKE_CURRENT_LIST_DIR}/fabm_library.F90.in
               ${CMAKE_CURRENT_BINARY_DIR}/fabm_library.F90)

# Specify default build type for single-type systems (not VS)
if(NOT CMAKE_BUILD_TYPE AND NOT CMAKE_CONFIGURATION_TYPES)
  set (CMAKE_BUILD_TYPE "Release" CACHE STRING "Choose the type of build." FORCE)
  set_property(CACHE CMAKE_BUILD_TYPE PROPERTY STRINGS "Debug" "Release" "MinSizeRel" "RelWithDebInfo")
endif()
```

```

# Customize compiler flags
if(${CMAKE_Fortran_COMPILER_ID} STREQUAL "GNU")
  add_compile_options(-ffree-line-length-none)
elseif(${CMAKE_Fortran_COMPILER_ID} STREQUAL "Intel")
  if(WIN32)
    add_compile_options("$<$<CONFIG:DEBUG>:/Od>")
  endif()
elseif(${CMAKE_Fortran_COMPILER_ID} STREQUAL "Cray")
  add_compile_options(-N 1023)
endif()

if(FABM_FORCED_HOST)
  # FABM_HOST was specified by a higher level CMakeLists.txt
  unset(FABM_HOST CACHE)
  set(FABM_HOST ${FABM_FORCED_HOST})
else()
  # FABM_HOST is user-configurable.
  # Make sure FABM_HOST is set to a subdir of src/drivers
  FILE(GLOB HOSTNAMES RELATIVE "${CMAKE_CURRENT_LIST_DIR}/drivers" "${CMAKE_CURRENT_LIST_DIR}/drivers/*")
  set (FABM_HOST "gotm" CACHE STRING "Host that FABM should be compiled for")
  set_property(CACHE FABM_HOST PROPERTY STRINGS ${HOSTNAMES})
  message(STATUS "FABM host: ${FABM_HOST}")
endif()

# Use host-prescribed real kind if set.
if(FABM_FORCED_REAL_KIND)
  # FABM_REAL_KIND was specified by a higher level CMakeLists.txt
  unset(FABM_REAL_KIND CACHE)
  set(FABM_REAL_KIND ${FABM_FORCED_REAL_KIND})
else()
  # FABM_REAL_KIND is user-configurable [as advanced variable]
  set(FABM_REAL_KIND "selected_real_kind(13)" CACHE STRING "Fortran real kind for host API")
  mark_as_advanced(FABM_REAL_KIND)
endif()
set(FABM_REAL_KIND_EXTERNAL ${FABM_REAL_KIND})
set(FABM_REAL_KIND_INTERNAL ${FABM_REAL_KIND_EXTERNAL} CACHE STRING "Fortran real kind for biogeochemistry.")
mark_as_advanced(FABM_REAL_KIND_INTERNAL)

option(FABM_USE_IEEE_ARITHMETIC "Use IEEE_ARITHMETIC module" ON)
mark_as_advanced(FABM_USE_IEEE_ARITHMETIC)
if(NOT FABM_USE_IEEE_ARITHMETIC)
  add_definitions(-D_NO_IEEE_ARITHMETIC_)
endif()

option(FABM_USE_DO_CONCURRENT "Use DO CONCURRENT statement (Fortran 2008)" ON)
mark_as_advanced(FABM_USE_DO_CONCURRENT)
if(NOT FABM_USE_DO_CONCURRENT)
  add_definitions(-D_NO_DO_CONCURRENT_)
endif()

option(FABM_USE_CONTIGUOUS "Use CONTIGUOUS attribute (Fortran 2008)" ON)
mark_as_advanced(FABM_USE_CONTIGUOUS)
if(NOT FABM_USE_CONTIGUOUS)
  add_definitions(-D_NO_CONTIGUOUS_)
endif()

set_property(DIRECTORY APPEND PROPERTY COMPILE_DEFINITIONS $<$<NOT:$<CONFIG:DEBUG>>:NDEBUG>)

# Use use position-independent code (-fPIC) everywhere if building shared libraries
if(BUILD_SHARED_LIBS)
  set(CMAKE_POSITION_INDEPENDENT_CODE ON)
endif()

# Set default installation prefix.
if(WIN32)
  if(DEFINED ENV{LOCALAPPDATA})
    set(DEFAULT_PREFIX "$ENV{LOCALAPPDATA}/fabm/${FABM_HOST}")
  else()
    set(DEFAULT_PREFIX "$ENV{APPDATA}/fabm/${FABM_HOST}")
  endif()
else()
  set(DEFAULT_PREFIX "$ENV{HOME}/local/fabm/${FABM_HOST}")
endif()

# Global include directories
include_directories("${CMAKE_CURRENT_LIST_DIR}/drivers/${FABM_HOST}"
                    "${CMAKE_CURRENT_LIST_DIR}/../include"
                    "${CMAKE_CURRENT_BINARY_DIR}/modules"
                   )

set(GIT_COMMIT_ID unknown)
set(GIT_BRANCH_NAME unknown)
configure_file(${CMAKE_CURRENT_LIST_DIR}/fabm_version.F90.in ${CMAKE_CURRENT_BINARY_DIR}/fabm_version.F90)
option(FABM_EMBED_VERSION "Embed FABM version information" OFF)
if(FABM_EMBED_VERSION)
  set(GET_GIT_INFO_SCRIPT "${CMAKE_CURRENT_LIST_DIR}/cmake/Modules/GetGitInfo.cmake")
  add_custom_target(fabm_version
      ${CMAKE_COMMAND} -DINFILE=${CMAKE_CURRENT_LIST_DIR}/fabm_version.F90.in -DOUTFILE=${CMAKE_CURRENT_BINARY_DIR}/fabm_version.F90 -P ${GET_GIT_INFO_SCRIPT}
      WORKING_DIRECTORY ${CMAKE_CURRENT_LIST_DIR}
      COMMENT "Retrieving description of last FABM commit..."
      SOURCES ${CMAKE_CURRENT_LIST_DIR}/fabm_version.F90.in ${GET_GIT_INFO_SCRIPT}
      VERBATIM
  )
  set_property(TARGET fabm_version PROPERTY FOLDER fabm)
endif()

```

```
configure_file(${CMAKE_CURRENT_LIST_DIR}/fabm_parameters.F90.in ${CMAKE_CURRENT_BINARY_DIR}/fabm_parameters.F90)

add_library(fabm_base OBJECT
            ${CMAKE_CURRENT_BINARY_DIR}/fabm_parameters.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_driver.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_standard_variables.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_properties.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_types.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_particle.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_expressions.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/constant.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/reduction.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/sum.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/scale.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/source.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/time_filter.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/depth_integral.F90
            ${CMAKE_CURRENT_LIST_DIR}/builtin/relaxation.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_builtin_models.F90
            ${CMAKE_CURRENT_LIST_DIR}/../include/fabm.h
            ${CMAKE_CURRENT_LIST_DIR}/../include/fabm_private.h
            ${CMAKE_CURRENT_LIST_DIR}/../include/fabm_version.h
            ${CMAKE_CURRENT_LIST_DIR}/drivers/${FABM_HOST}/fabm_driver.h
           )
source_group(builtin REGULAR_EXPRESSION /builtin/)

# Create a list of institute-specific object libraries.
foreach(institute ${ACTIVE_INSTITUTES})
  string(TOUPPER ${institute} INSTITUTE_UC)
  if(NOT DEFINED FABM_${INSTITUTE_UC}_BASE)
    set(FABM_${INSTITUTE_UC}_BASE "${CMAKE_CURRENT_LIST_DIR}/models/${institute}")
  endif()
  set(CMAKE_FOLDER fabm/models)
  add_subdirectory(${FABM_${INSTITUTE_UC}_BASE} ${CMAKE_CURRENT_BINARY_DIR}/models/${institute})
  set_property(TARGET fabm_models_${institute} PROPERTY Fortran_MODULE_DIRECTORY "${CMAKE_CURRENT_BINARY_DIR}/models/${institute}")
  target_include_directories(fabm_models_${institute} PUBLIC "${CMAKE_CURRENT_BINARY_DIR}/models/${institute}")
  list(APPEND MODEL_OBJECTS $<TARGET_OBJECTS:fabm_models_${institute}>)
  set_property(TARGET fabm_models_${institute} PROPERTY FOLDER fabm/models)
endforeach(institute)

# Add YAML parsing library
if(NOT TARGET yaml)
  add_subdirectory("${CMAKE_CURRENT_LIST_DIR}/yaml" yaml EXCLUDE_FROM_ALL)
  set_property(TARGET yaml PROPERTY FOLDER fabm/yaml)
  install(DIRECTORY ${CMAKE_CURRENT_BINARY_DIR}/yaml/modules/\${BUILD_TYPE}/ DESTINATION include/yaml)
endif(NOT TARGET yaml)

add_library(fabm
            $<TARGET_OBJECTS:fabm_base>
            ${YAML_SOURCES}
            ${MODEL_OBJECTS}
            ${CMAKE_CURRENT_BINARY_DIR}/fabm_library.F90
            ${CMAKE_CURRENT_BINARY_DIR}/fabm_version.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_job.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_task_order.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_graph.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_coupling.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_schedule.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_config.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_debug.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_work.F90
            ${CMAKE_CURRENT_LIST_DIR}/fabm_v0_compatibility.F90
           )
add_dependencies(fabm yaml)

if(FABM_EMBED_VERSION)
  add_dependencies(fabm fabm_version)
endif()

# Store FABM *.mod in separate directory, so these files can later be installed.
set_property(TARGET fabm_base fabm PROPERTY Fortran_MODULE_DIRECTORY "${CMAKE_CURRENT_BINARY_DIR}/modules")
target_include_directories(fabm_base PUBLIC
  $<BUILD_INTERFACE:${CMAKE_CURRENT_BINARY_DIR}/modules>
  $<BUILD_INTERFACE:${CMAKE_CURRENT_LIST_DIR}/drivers/${FABM_HOST}>
  $<BUILD_INTERFACE:${CMAKE_CURRENT_LIST_DIR}/../include>
  $<INSTALL_INTERFACE:include>
)
target_include_directories(fabm PUBLIC
  $<BUILD_INTERFACE:${CMAKE_CURRENT_BINARY_DIR}/modules>
  $<BUILD_INTERFACE:${CMAKE_CURRENT_LIST_DIR}/drivers/${FABM_HOST}>
  $<BUILD_INTERFACE:${CMAKE_CURRENT_LIST_DIR}/../include>
  $<INSTALL_INTERFACE:include>
)

# Add directories with model-specific *.mod to include directories of main FABM library.
target_include_directories(fabm PRIVATE $<TARGET_PROPERTY:yaml,INTERFACE_INCLUDE_DIRECTORIES>)
foreach(institute ${ACTIVE_INSTITUTES})
  target_include_directories(fabm PRIVATE $<TARGET_PROPERTY:fabm_models_${institute},INTERFACE_INCLUDE_DIRECTORIES>)
endforeach(institute)

option(FABM_BUILD_C_API "Build C interfaces" OFF)
mark_as_advanced(FABM_BUILD_C_API)
if(FABM_BUILD_C_API)
  add_subdirectory("${CMAKE_CURRENT_LIST_DIR}/c" c)
endif()

```

```
# Tests
add_executable(test_host EXCLUDE_FROM_ALL
               ${CMAKE_CURRENT_LIST_DIR}/test/host.F90
               ${CMAKE_CURRENT_LIST_DIR}/test/models.F90
              )
target_link_libraries(test_host PRIVATE fabm)
if(YAML_SOURCES)
  # YAML is integrated into FABM library; we only need to set the proper include directory
  target_include_directories(test_host PRIVATE $<TARGET_PROPERTY:yaml,INTERFACE_INCLUDE_DIRECTORIES>)
else()
  # YAML library is stand-alone - target_link_libraries will add both include directory and library
  target_link_libraries(test_host PRIVATE yaml)
endif()

# Place projects in specific solution folders (Visual Studio only).
set_property(TARGET fabm_base fabm test_host PROPERTY FOLDER fabm)

if(NOT FABM_FORCED_HOST)
  # Re-initialize installation prefix if needed.
  if(CMAKE_INSTALL_PREFIX_INITIALIZED_TO_DEFAULT OR CMAKE_INSTALL_PREFIX STREQUAL PREVIOUS_DEFAULT_PREFIX)
    get_filename_component(DEFAULT_PREFIX "${DEFAULT_PREFIX}" ABSOLUTE)
    set(CMAKE_INSTALL_PREFIX "${DEFAULT_PREFIX}" CACHE PATH "Directory to install FABM in" FORCE)
    set(PREVIOUS_DEFAULT_PREFIX "${DEFAULT_PREFIX}" CACHE INTERNAL "Default directory to install FABM in")
  endif()
  message(STATUS "Installation prefix: ${CMAKE_INSTALL_PREFIX}")
endif()

# Install FABM library and include files, unless the host overruled this.
if (NOT DEFINED FABM_NO_LIBRARY_INSTALL)
  install(TARGETS fabm
    EXPORT fabmConfig
    DESTINATION lib)
  install(DIRECTORY ${CMAKE_CURRENT_BINARY_DIR}/modules/\${BUILD_TYPE}/
    DESTINATION include)
  install(FILES
    ${CMAKE_CURRENT_LIST_DIR}/../include/fabm.h
    ${CMAKE_CURRENT_LIST_DIR}/../include/fabm_version.h
    ${CMAKE_CURRENT_LIST_DIR}/drivers/${FABM_HOST}/fabm_driver.h
    DESTINATION include)
endif()

export(EXPORT fabmConfig)
install(EXPORT fabmConfig DESTINATION cmake)
```