

```

1 cmake_minimum_required(VERSION 3.0)
2
3 if(POLICY CMP0048)
4   cmake_policy(SET CMP0048 NEW)
5 endif()
6
7 # Use solution folders in IDEs
8 set_property(GLOBAL PROPERTY USE_FOLDERS ON)
9
10 # Create main project.
11 project(fabm0d Fortran)
12
13 # Find GOTM directory.
14 # This directory will also be configurable in the CMake GUI or interactive mode.
15 find_path(GOTM_BASE
16     NAMES src/gotm/gotm.F90
17     HINTS ENV GOTM_DIR
18     PATHS ${PROJECT_SOURCE_DIR}/../../../../../ ${PROJECT_SOURCE_DIR}/../../..
19     PATH_SUFFIXES GOTM/gotm-git gotm-git gotm
20     DOC "Directory with GOTM source code (www.gotm.net)")
21
22 # Make NetCDF usage a configurable option.
23 option(FABM_USE_NetCDF "Enable NetCDF output" ON)
24
25 # Specify additional compiler flags
26 if(${CMAKE_Fortran_COMPILER_ID} STREQUAL "GNU")
27   set(CMAKE_Fortran_FLAGS "${CMAKE_Fortran_FLAGS} -ffree-line-length-none")
28 elseif(${CMAKE_Fortran_COMPILER_ID} STREQUAL "Intel")
29   if(WIN32)
30     set(CMAKE_Fortran_FLAGS_DEBUG "${CMAKE_Fortran_FLAGS_DEBUG} /Od")
31   endif()
32 endif()
33
34 set(GOTM_USE_NetCDF ${FABM_USE_NetCDF} CACHE BOOL "Enable output in NetCDF format" FORCE)
35 set(GOTM_USE_FABM OFF CACHE BOOL "use FABM" FORCE)
36 set(GOTM_USE_SEAGRASS OFF CACHE BOOL "use seagrass" FORCE)
37 set(GOTM_USE_STIM OFF CACHE BOOL "use STIM" FORCE)
38 mark_as_advanced(GOTM_USE_FABM GOTM_USE_NetCDF GOTM_USE_SEAGRASS GOTM_USE_STIM)
39 add_subdirectory(${GOTM_BASE} gotm EXCLUDE_FROM_ALL)
40
41 # Force FABM_HOST to 0d and include FABM CMakeLists.txt.
42 set(FABM_FORCED_HOST 0d)
43 add_subdirectory(${PROJECT_SOURCE_DIR}/../.. fabm EXCLUDE_FROM_ALL)
44
45 # Build 0d executable.
46 add_executable(fabm0d
47     main.F90
48     fabm0d.F90
49     shared.F90
50     output.F90
51     fabm_0d.h
52 )
53
54 # Link with FABM library, field/output manager and GOTM.
55 target_link_libraries(fabm0d PRIVATE fabm output_manager field_manager airsea input_manager util)
56
57 # Set module directory (output of *.mod)
58 set_property(TARGET fabm0d PROPERTY Fortran_MODULE_DIRECTORY ${CMAKE_CURRENT_BINARY_DIR}/modules)
59
60 # Set include directories with modules (*.mod)
61 target_include_directories(fabm0d PRIVATE "${CMAKE_CURRENT_BINARY_DIR}/modules")
62
63 # Set include directories with header files (*.h) for GOTM - version.h, cppdefs.h, ode_solver_template.F90.
64 target_include_directories(fabm0d PRIVATE "${GOTM_BASE}/include")
65 target_include_directories(fabm0d PRIVATE "${GOTM_BASE}/src/util")
66
67 # Set include directories with header files (*.h) - our own and FABM's.
68 target_include_directories(fabm0d PRIVATE "${PROJECT_SOURCE_DIR}")
69 target_include_directories(fabm0d PRIVATE "${PROJECT_SOURCE_DIR}/../../.. /include")
70 target_include_directories(fabm0d PRIVATE $<TARGET_PROPERTY:yaml,INTERFACE_INCLUDE_DIRECTORIES>)
71
72 # Set default installation prefix.
73 if(WIN32)
74   if(DEFINED ENV{LOCALAPPDATA})
75     set(DEFAULT_PREFIX "$ENV{LOCALAPPDATA}/fabm/0d")
76   else()
77     set(DEFAULT_PREFIX "$ENV{APPDATA}/fabm/0d")
78   endif()
79 else()
80   set(DEFAULT_PREFIX "$ENV{HOME}/local/fabm/0d")
81 endif()
82
83 # Set cache variable with installation prefix.
84 if(CMAKE_INSTALL_PREFIX_INITIALIZED_TO_DEFAULT)
85   # None set - set our default prefix.
86   set(CMAKE_INSTALL_PREFIX "${DEFAULT_PREFIX}" CACHE PATH "Directory to install FABM's 0d driver in" FORCE)
87 else()
88   # User specified a prefix - just update the variable description.
89   set(CMAKE_INSTALL_PREFIX "${DEFAULT_PREFIX}" CACHE PATH "Directory to install FABM's 0d driver in")
90 endif()
91 message(STATUS "Installation prefix: ${CMAKE_INSTALL_PREFIX}")
92
93 install(TARGETS fabm0d
94     DESTINATION bin)

```