

The EwE-F User's Manual

for version 2.0

E. Akoglu^{1,2}, S. Libralato², B. Salihoglu¹, T. Oguz¹, and C.
Solidoro²

¹Middle East Technical University, Institute of Marine Sciences
(IMS-METU), Turkey

²Istituto Nazionale di Oceanografia e di Geofisica Sperimentale
(OGS), Italy

March 2019

Contents

1	Introduction	1
1.1	What is <i>EwE-F</i> ?	1
1.2	The motivation behind <i>EwE-F</i>	1
1.3	How to read this manual?	2
2	Instructions for UNIX-like systems	3
2.1	Prerequisites	3
2.2	Obtaining the source code of the <i>EwE-F</i>	4
2.3	Compiling and running the <i>EwE-F</i>	5
3	Instructions for Windows® systems	7
3.1	Prerequisites	7
3.1.1	Installing the HDF5 libraries from source	8
3.1.1.1	Configuring	9
3.1.1.2	Compiling	22
3.2	Obtaining the source code of the <i>EwE-F</i>	31
3.3	Setting up an <i>EwE-F</i> project in MS Visual Studio 2010	36
3.4	Compiling and running the <i>Ecosim-F</i>	48
4	The <i>EwE-F</i>: Details & Customisation	53
4.1	Folder hierarchy	53
4.2	The <i>Ecopath-F</i>	57
4.2.1	Files in the <i>src</i> directory	57
4.2.2	Setting up test cases for an <i>Ecopath-F</i> run	59
4.2.3	Setting up your model for an <i>Ecopath-F</i> run	59

4.2.4	Description of the input files	61
4.2.4.1	The main parameters input file: $<YourModelName>_{-}$ Scenario.csv	61
4.2.4.2	The diet composition input file: $<YourModelName>_{-}$ DC.csv	66
4.2.4.3	The detritus fate input file: $<YourModelName>_{-}$ DetFate.csv	68
4.2.4.4	The growth parameters input file: $<YourModelName>_{-}$ GrowthParam.csv . . .	69
4.2.5	Description of the <i>Ecopath-F</i> output (results) files . . .	70
4.3	The <i>Ecosim-F</i>	71
4.3.1	Files in the <i>src</i> directory	71
4.3.2	Setting up test cases for an <i>Ecosim-F</i> run	74
4.3.3	Setting up your model for an <i>Ecosim-F</i> run	75
4.3.4	Description of the input files	77
4.3.4.1	The group parameters input file: $<YourModelName>_{-}$ Scenario.csv	77
4.3.4.2	The vulnerability matrix input file: $<YourModelName>_{-}$ vul.csv	81
4.3.4.3	The forcing data input file: $<YourModelName>_{-}$ F.csv	83
4.3.4.4	The nutrient forcing data input file: $<YourModelName>_{-}$ NutForce.csv	85
4.3.4.5	The primary production forcing data input file: $<YourModelName>_{-}$ PrimaryProdForce.csv .	86
4.3.5	Description of the <i>Ecosim-F</i> output (results) files . .	86
4.4	The <i>Eospace-F</i>	88
4.4.1	Implementation	88
4.4.2	Files in the <i>src</i> directory	90
4.4.3	Setting up test cases for an <i>Eospace-F</i> run	90
4.4.4	Enabling <i>Eospace-F</i> when compiling <i>Ecosim-F</i>	91
4.4.5	Setting up your model for an <i>Eospace-F</i> run	92

4.4.6	Description of the input files	94
4.4.6.1	The spatial grid input file: $<YourModelName>_grid.csv$	94
4.4.6.2	The spatial distributions input files: "#.csv"	96
4.4.6.3	The advection input file: $<YourModelName>_Advection.csv$	96
4.4.7	Description of the <i>Ecospace-F</i> output (results) file	99

List of Figures

3.1	CMake: Main; specifying HDF5 source and build paths.	10
3.2	CMake: Configure; specifying compilers (1).	11
3.3	CMake: Configure; specifying compilers (2).	12
3.4	CMake: Configure; configuring (1).	13
3.5	CMake: Configure; configuring (2).	14
3.6	CMake: Configure; new configure options (in red).	15
3.7	CMake: Configure; selecting new configure options for Fortran and shared library support.	16
3.8	CMake: Configure; re-configuring (1).	17
3.9	CMake: Configure; new configure option for enabling Fortran 2003 support.	18
3.10	CMake: Configure; selecting new configure option for Fortran 2003 support.	19
3.11	CMake: Configure; re-configuring (2).	20
3.12	CMake: Generating.	21
3.13	Visual Studio: Building HDF5; opening the solution (1). . . .	23
3.14	Visual Studio: Building HDF5; opening the solution (2). . . .	24
3.15	Visual Studio: Building HDF5; opening the Configuration Manager.	25
3.16	Visual Studio: Building HDF5; setting options for the Config- uration Manager (1).	26
3.17	Visual Studio: Building HDF5; setting options for the Config- uration Manager (2).	27
3.18	Visual Studio: Building HDF5; building.	28

3.19	Visual Studio: Building HDF5; building completed.	29
3.20	Locating and selecting the HDF5 executable for installation.	30
3.21	GIT: opening GIT command window in the empty directory.	32
3.22	GIT: command window and entering command to download sources.	33
3.23	GIT: downloading sources done, the new <i>ecosim-f</i> directory.	34
3.24	The <i>ecosim-f</i> directory structure.	35
3.25	Visual Studio: <i>Ecosim-F</i> ; creating project (1).	38
3.26	Visual Studio: <i>Ecosim-F</i> ; creating project (2).	39
3.27	Visual Studio: <i>Ecosim-F</i> ; adding source files to the project (1).	40
3.28	Visual Studio: <i>Ecosim-F</i> ; adding source files to the project (2).	41
3.29	Visual Studio: <i>Ecosim-F</i> ; source files added to the project.	42
3.30	Visual Studio: <i>Ecosim-F</i> ; opening project properties.	43
3.31	Visual Studio: <i>Ecosim-F</i> ; setting additional include directories.	44
3.32	Visual Studio: <i>Ecosim-F</i> ; enabling preprocessor.	45
3.33	Visual Studio: <i>Ecosim-F</i> ; setting additional library directories.	46
3.34	Visual Studio: <i>Ecosim-F</i> ; setting additional dependencies.	47
3.35	Visual Studio: <i>Ecosim-F</i> ; building.	49
3.36	<i>Ecosim-F</i> : running the model (1).	50
3.37	<i>Ecosim-F</i> : running the model (2).	51
3.38	<i>Ecosim-F</i> : simulation completed.	52
4.1	Sample ASCII file containing the main input parameters matrix for Tampa Bay EwE model.	63
4.2	Sample ASCII file containing the diet composition matrix for Tampa Bay EwE model.	67
4.3	Sample ASCII file containing the detritus fate matrix Tampa Bay EwE model.	69
4.4	Sample ASCII file containing the growth parameters of multi-stanza groups for Tampa Bay EwE model.	70
4.5	Sample ASCII file containing the group input parameters matrix for Tampa Bay EwE model.	79

4.6	Sample ASCII file containing the vulnerability matrix for Tampa Bay EwE model.	82
4.7	Sample ASCII file containing the forcing time series data matrix for Generic 37 EwE model.	84
4.8	Sample ASCII file containing the nutrient forcing time series data matrix for Tampa Bay EwE model.	85
4.9	Sample ASCII file containing the primary production forcing time series data matrix for Generic 37 EwE model.	86
4.10	The movements of nektonic organisms to and from a grid cell in <i>Ecospace-F</i> . x and y denote latitude and longitude coordinates of the current cell respectively, B denotes biomass of an organism in cell x, y and m is the movement rate between grid cells. Movement between diagonal cells is not allowed.	88
4.11	Sample comma-separated ASCII file containing the <i>Ecospace-F</i> spatial grid for the Black Sea model.	95
4.12	Sample comma-separated ASCII file containing the relative spatial distribution of a functional group in an <i>Ecospace-F</i> hypothetical 10x10 spatial grid.	96
4.13	Sample comma-separated ASCII file containing directions of advective flows in an <i>Ecospace-F</i> grid.	98

Preface

We have been developing a version of Ecopath with Ecosim (EwE) in Fortran 95/2003 language (*EwE-F*) in order to facilitate the integration of this widely-used ecosystem model with existing oceanographic models which were mostly coded in Fortran. Primarily we developed it in order to conveniently couple it with available physical-biogeochemical models without using external couplers; however, practically *EwE-F* can be coupled with any other model.

EwE-F does not yet have the extensive analysis capabilities (e.g. network analysis, time series fitting) of Ecopath with Ecosim, but it is fully compliant with the EwE 6.6 fundamental equations (core routines of Ecopath and of Ecosim). Starting with version 2 of EwE-F, it incorporates primitive spatial simulation capabilities similar to, however, not fully compliant with Ecospace. Like its parent, *EwE-F* is also an open source software freely available under the GNU General Public License v2.

Under the umbrella of the Ecopath Consortium, we will provide further efforts to evolve *EwE-F* in line with the endeavours of Ecopath Consortium and the EwE user community.

The *EwE-F* development team:

Ekin Akoglu; OGS and IMS-METU
Baris Salihoglu, Temel Oguz; IMS-METU
Simone Libralato, Cosimo Solidoro; OGS

Chapter 1

Introduction

1.1 What is *EwE-F*?

EwE-F is the Fortran 95/2003 implementation of the widely-adopted ecological modelling tool for aquatic ecosystems; Ecopath with Ecosim (EwE) (Christensen and Walters, 2004; Christensen et al., 2005), which was written in Visual Basic .NET for Microsoft Windows platforms.

1.2 The motivation behind *EwE-F*

Being written in Visual Basic .NET, EwE is constrained strictly to the Microsoft Windows environment and this hampers the communicability of EwE with other oceanographic models majority of which were conventionally programmed in Fortran. This difficulty in communicability is crucial considering that nowadays ecological modelling is facing an important challenge to set basis for comprehensive description of marine ecosystems through what are called End-to-End (E2E) models (Fulton, 2010). This increasing requirement for setting up holistic (end-to-end) representations of ecosystems to explore ecosystem-based management options in marine and freshwater environments under changing climatological and anthropogenic conditions resulted in a pressing need for a Fortran implementation of such a popular higher-trophic-level (HTL) model that would be convenient for coupling

with numerous state-of-the-art hydrodynamic and lower-trophic-level (LTL) biogeochemical models. Hence, the availability of a Fortran version of EwE is an important step towards removing these obstacles mentioned above and help actuate the E2E ecosystem modelling research in aquatic ecosystems.

1.3 How to read this manual?

This manual is separated into four chapters. In Chapter 1 (this chapter), a brief introduction to *EwE-F* is given. In Chapter 2, obtaining, compiling and running *EwE-F* with its default configuration are introduced for UNIX-like systems (e.g. Linux, Mac OS, BSD, Sun/Oracle Solaris etc.). Similarly, in Chapter 3, obtaining, compiling and running *EwE-F* with its default configuration are introduced for Microsoft Windows systems. In the last chapter, (Chapter 4), structural details of the *EwE-F* are introduced under three separate sections; Section 4.2 is devoted to the *Ecopath-F*, Section 4.3 is devoted to the *Ecosim-F* and Section 4.4 is devoted to the *Eospace-F*. These three sections give detailed information on setting up *EwE-F* for your own custom ecosystem simulations by explaining the structure of the input files required by the *EwE-F* to run simulations as well as how to prepare them. In addition, the sections provide information about the output files created by the model. Chapter 4 also gives important information about the source code structures of *Ecopath-F*, *Ecosim-F* and *Eospace-F* in their respective sections.

You may start reading the manual by directly heading to the chapter related to your computing environment and skipping the other. However, Chapter 4 is a must-read for all *EwE-F* users in order to learn how to correctly prepare scenario files for the *EwE-F* simulations and how to interpret the results files output by the *EwE-F*. Further, the chapter provides detailed explanations of the source code files of the model for those users wishing to modify *EwE-F* to meet their specific needs.

Chapter 2

Instructions for UNIX-like systems

2.1 Prerequisites

Before downloading the source code of the *EwE-F*, make sure that the following software is installed on your system:

1. GIT distributed version control system (<http://git-scm.com/>) in order to download the source code of the *EwE-F*,
2. A Fortran compiler (e.g. GNU Fortran <http://gcc.gnu.org/fortran/>) in order to compile the source code,
3. HDF5 (<https://portal.hdfgroup.org/display/support/Downloads>) libraries in order to be able to store the *Ecopath-F* and *Ecosim-F* model results,



Important!

Be sure that you configure the HDF5 libraries with –enable-fortran switch.

4. NetCDF (<https://www.unidata.ucar.edu/downloads/netcdf/index.jsp>) C and Fortran libraries in order to be able to store the *Ecospace-F* model results.

2.2 Obtaining the source code of the *EwE-F*

After the installation of the prerequisites, follow the link <https://gitlab.com/ewe-f> to access the source code via the GitLab® web-based interface.

If you wish to access to and download the source code of the *EwE-F* via GIT version control software, the steps to follow are:

1. Create an empty directory for the files to be downloaded.
2. Open a terminal window and change (with command *cd*) to the empty directory you have just created.
3. Issue the commands below in the terminal / command prompt. For downloading the *Ecopath-F* sources do as shown in Listing 2.1.

Listing 2.1: Downloading *Ecopath-F*

```
git clone https://gitlab.com/ewe-f/ecopath-f.git
```

and for downloading the *Ecosim-F* sources, which also includes *Ecospace-F*, do as shown in Listing 2.2.

Listing 2.2: Downloading *Ecosim-F*

```
git clone https://gitlab.com/ewe-f/ecosim-f.git
```

If you completed the steps above successfully, you must have downloaded all the source files in your local directory. You will recognise that two directories named *ecopath-f* and *ecosim-f* are created in your local directory. These two directories have similar structure and contain a number of folders and files. For the explanation of the folders under these two directories see

Section 4.1 and for the explanation of the files see Section 4.2 for *Ecopath-F* and sections 4.3 and 4.4 for *Ecosim-F* and *Ecospace-F* respectively.

2.3 Compiling and running the *EwE-F*

Simply, there is a **Makefile** in each of the *ecopath-f* and *ecosim-f* directories. To compile the *Ecopath-F* or *Ecosim-F* model, just run *make* in the terminal under the model's respective root directory. For an *Ecosim-F* scenario with *Ecospace-F*, just run *ecospace=YES make*. The *make* command will create an executable named **ecoXXX.out** (where *XXX* is either *path* or *sim*) under *bin* directory. This binary file is the model. You can run it by typing *bin/ecoXXX.out* in the terminal. For detailed explanations utilising examples you may refer to sections 4.2.2, 4.3.2 and 4.4.3.

Important!

*You may need to edit the **Makefile** before you run *make* in order to correctly specify the command required by your Fortran compiler and set its respective compilation and linking flags and the path to the folder where HDF5 and NetCDF libraries are installed on your system. Once opened with a text editor, you will see that the file is quite self-explanatory and guides you through the sections that need modification. However, in a nutshell, you may need to edit lines 25, 42, 48 and the region between lines 59-73 and lines 87-88 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 and NetCDF libraries. The lines starting with a hash (#) are explanatory comment lines. Read them as you find your way through these lines.*

Whenever you wish to clean-up the *EwE-F/ecopath-f* or *EwE-F/ecosim-f* directories, you can simply run "make clean" in the terminal. For an *Ecosim-F* scenario set up with *Ecospace-F*, you need to run "ecospace=YES make clean". This will delete the object and module files created by *make* and the respective executable. Then you can type "make" again for *Ecopath-F*.

and *Ecosim-F* and "ecospace=YES make" for an *Ecosim-F* scenario with *Ecospace-F* if you wish to recompile the code.

 **Information**

EwE-F was compiled and tested against GNU FORTRAN compiler. Other compilers may or may not work.

Chapter 3

Instructions for Windows[®] systems

3.1 Prerequisites

To compile and run the *EwE-F* on a Microsoft[®] Windows[®] computer, the following software must be downloaded and installed in advance:

1. NSIS (Nullsoft[®] Scriptable Install System) from
<http://nsis.sourceforge.net/Download>,
2. CMake, the cross-platform make system from
<http://www.cmake.org/cmake/resources/software.html>,
3. HDF5 libraries and software from
<https://portal.hdfgroup.org/display/support/Downloads>,

Important!

As of version 1.8.11, pre-built HDF5 binaries for Windows available on HDF5 website include Fortran 2003 support. Hence, you may directly download and install pre-built HDF5 binaries from the HDF5 website (without installing 1 and 2) and skip Section 3.1.1 and proceed with Section 3.2. If you wish to use an older version of HDF5 (<

1.8.11), be sure that you compile and install the HDF5 libraries from source as detailed in Section 3.1.1 because the pre-built binaries for Windows do not include Fortran 2003 support, which is required by EwE-F.

4. NetCDF C and Fortran libraries and software from
<https://www.unidata.ucar.edu/downloads/netcdf/index.jsp>,

 **Important!**

NetCDF Fortran library isn't supported on Windows yet, however, it is speculated that it can be built: <https://www.unidata.ucar.edu/support/help/MailArchives/netcdf/msg14233.html>. This may limit Windows users not being able to utilise Ecospace-F.

5. Microsoft® Visual Studio Community Edition from
<https://visualstudio.microsoft.com/free-developer-offers/>,
6. Intel® Visual Fortran Compiler for Windows from
<https://software.intel.com/en-us/intel-visual-fortran-compiler-for-windows>,
7. GIT distributed version controlling system for MS Windows from
<http://git-scm.com/downloads>.

3.1.1 Installing the HDF5 libraries from source

 **Important!**

Be sure that you install CMake, NSIS, Microsoft® Visual Studio and Intel® Visual Fortran Compiler before proceeding to this section.

3.1.1.1 Configuring

Download the HDF5 sources from the URL given above and extract them into a suitable directory. Go into the extracted HDF5 folder and create an empty directory named *build*. Then launch the CMake program. A CMake window like the one in Figure 3.1 will appear. Provide the respective paths of the HDF5 source code and the folder to build the binaries in the top two entries. Set the build directory path to the *build* folder you have just created. Then press the *Configure* button located on bottom left. A pop-up window as shown in Figure 3.2 will appear asking you to specify the generator you would like to use to build the project. Select *Visual Studio 10* in the drop-down list and then select the radio button *Specify native compilers* and click *Next*. In the new window, provide the paths to your C, C++ and Fortran compilers as shown in Figure 3.3 and click *Finish*. The configure process will take some time as shown in Figure 3.4 and 3.5. Once completed, the new configure options will appear in red in the CMake window as shown in Figure 3.6. Check-mark the options named *HDF_BUILD_FORTRAN* and *BUILD_SHARED_LIBS* as shown in Figure 3.7 and click again the *Configure* button and wait until it is completed as shown in Figure 3.8. Once completed, another new option named *HDF5_ENABLE_F2003* will be shown in red (Figure 3.9). Check-mark this option as shown in Figure 3.10 and press *Configure* button for one last time (Figure 3.11). When *Configure* is completed, press the *Generate* button (Figure 3.12). When done, you may close the CMake program and proceed to compiling HDF5 in MS Visual Studio.

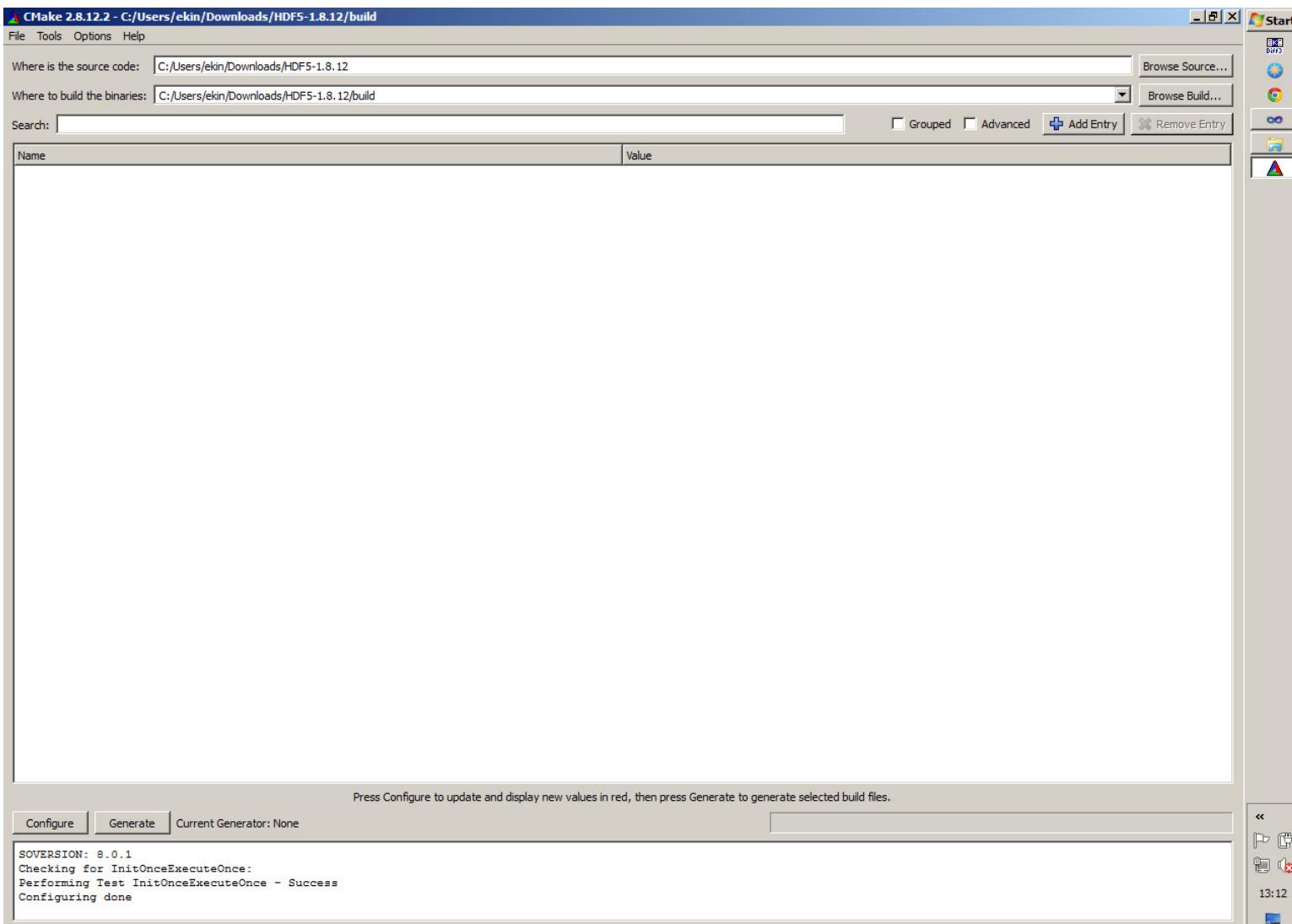


Figure 3.1: CMake: Main; specifying HDF5 source and build paths.

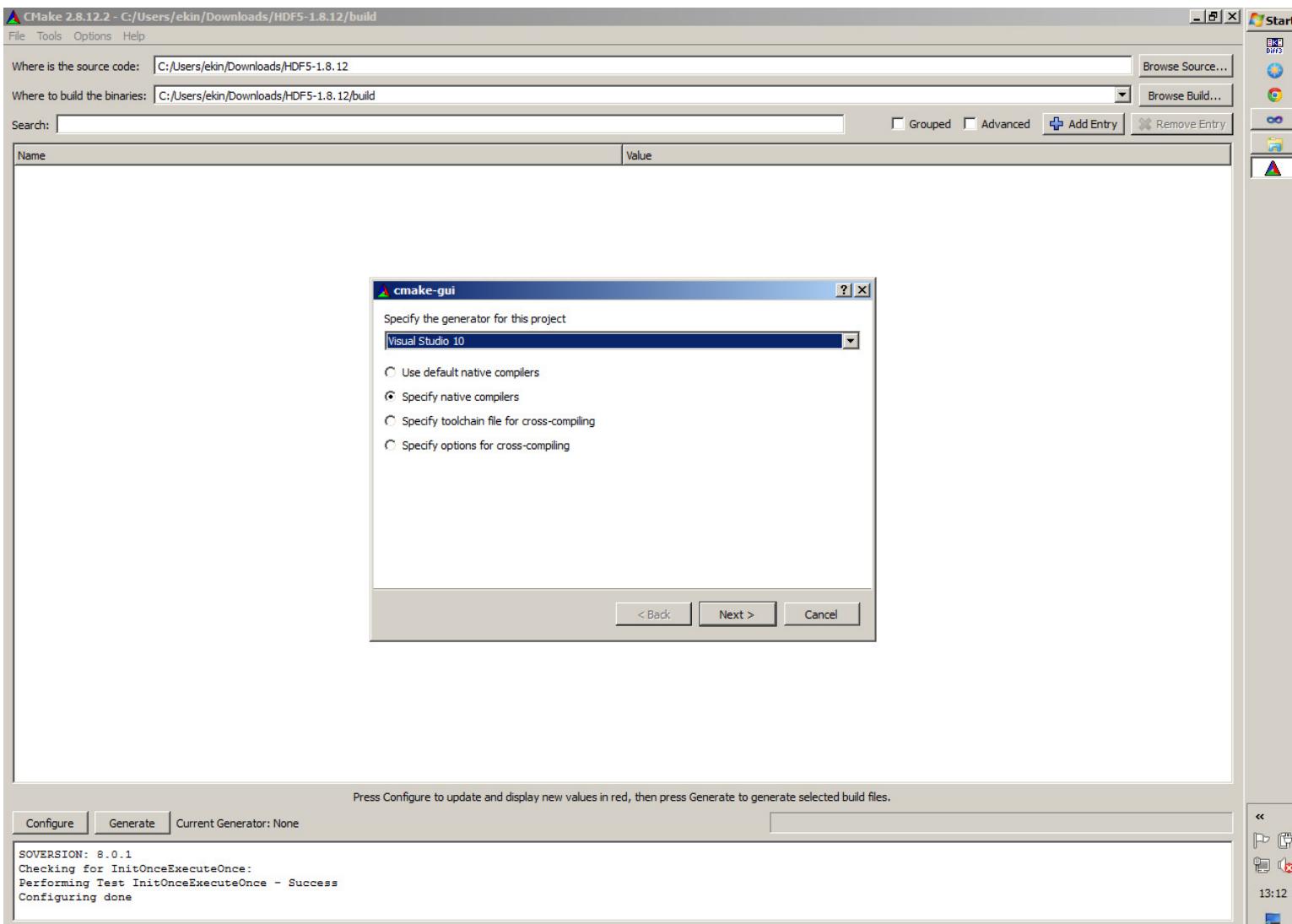


Figure 3.2: CMake: Configure; specifying compilers (1).

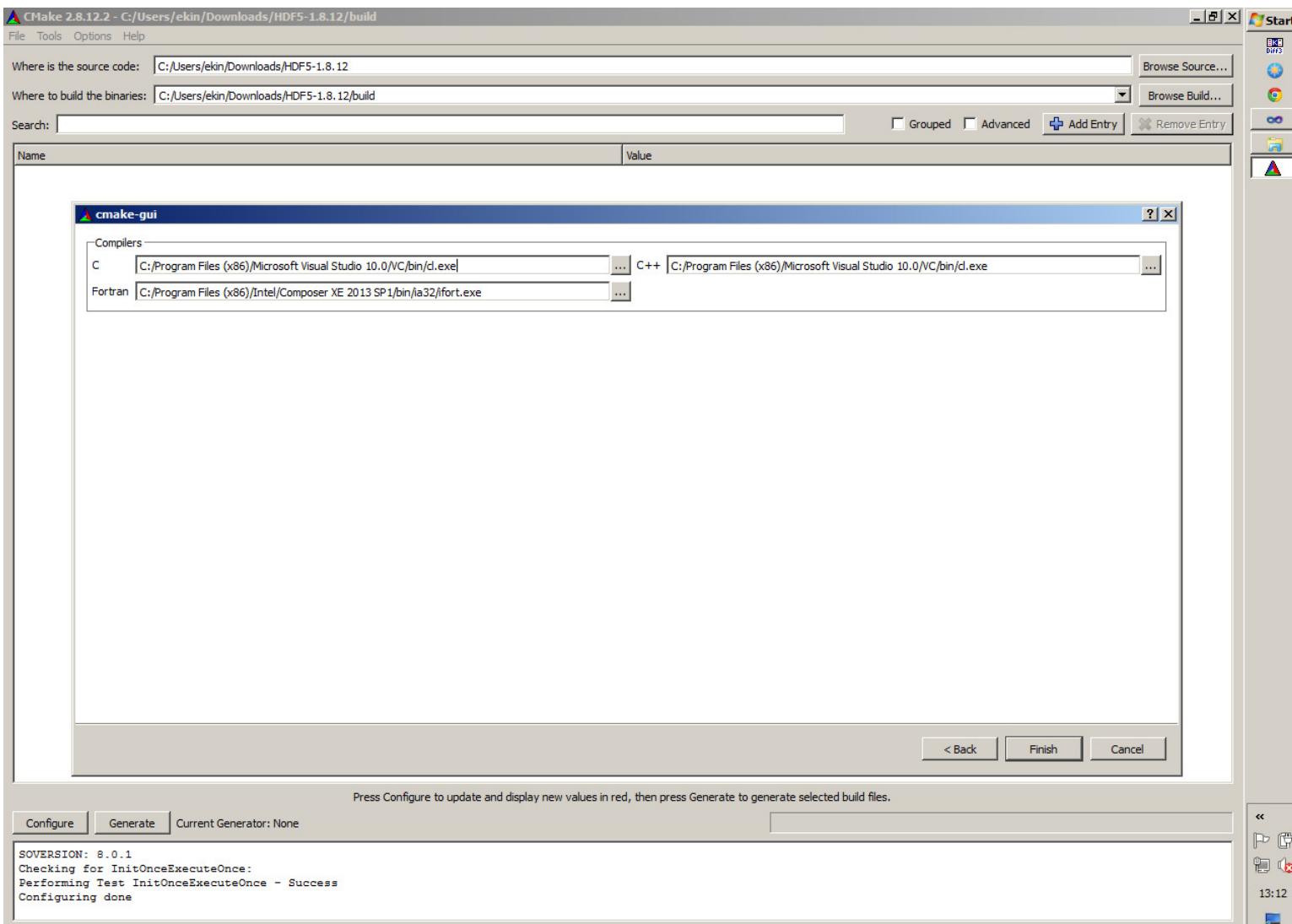


Figure 3.3: CMake: Configure; specifying compilers (2).

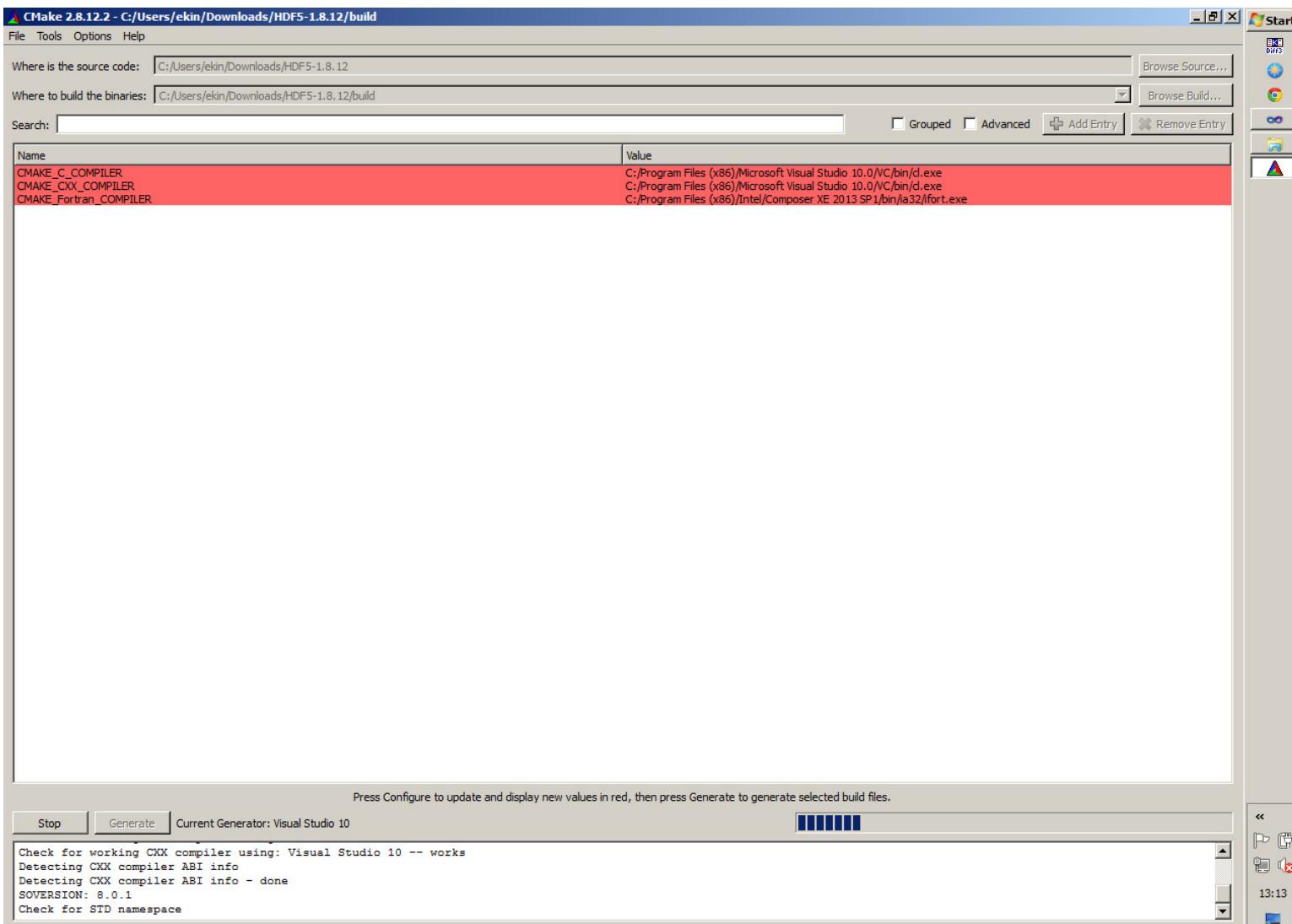


Figure 3.4: CMake: Configure; configuring (1).

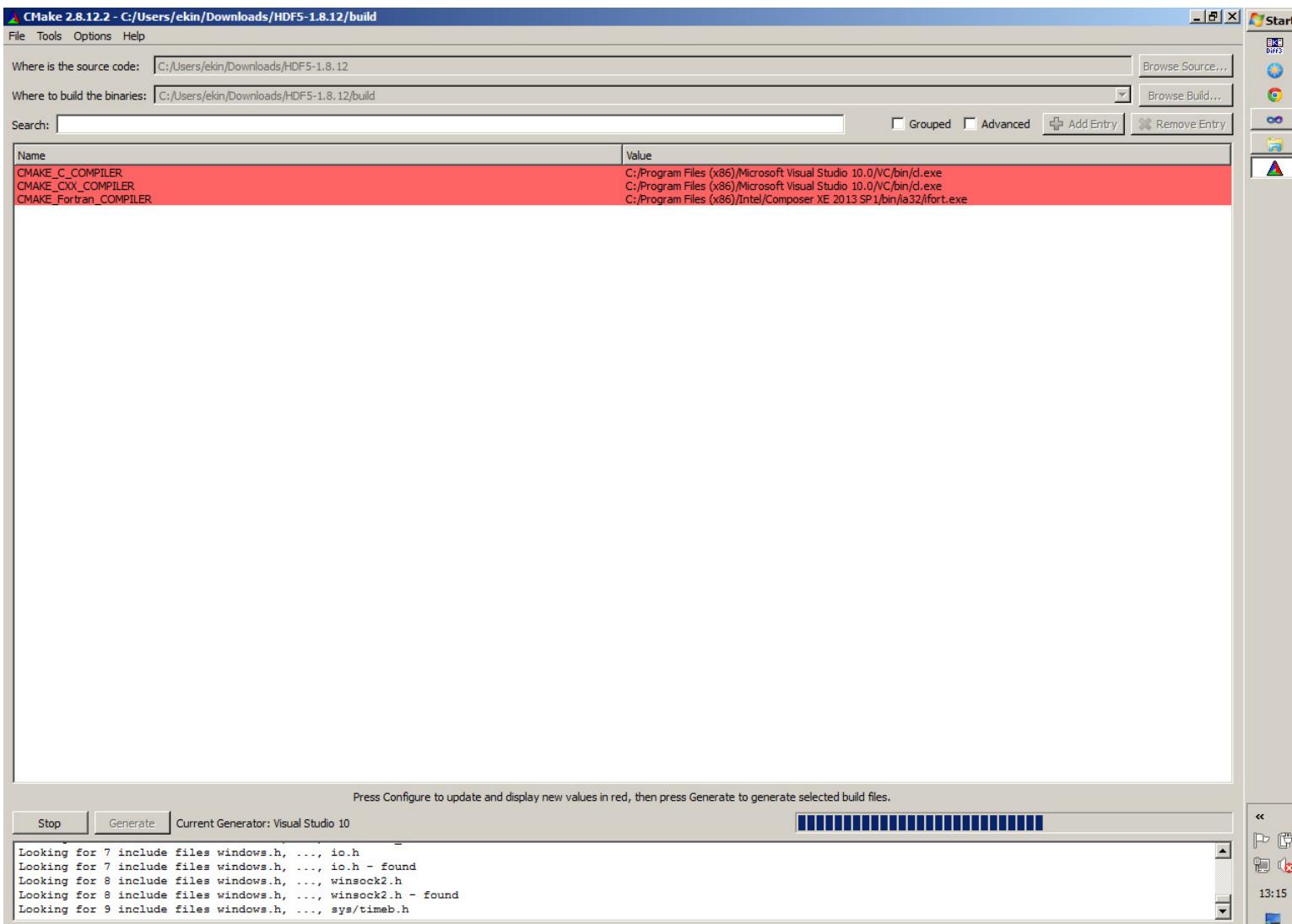


Figure 3.5: CMake: Configure; configuring (2).

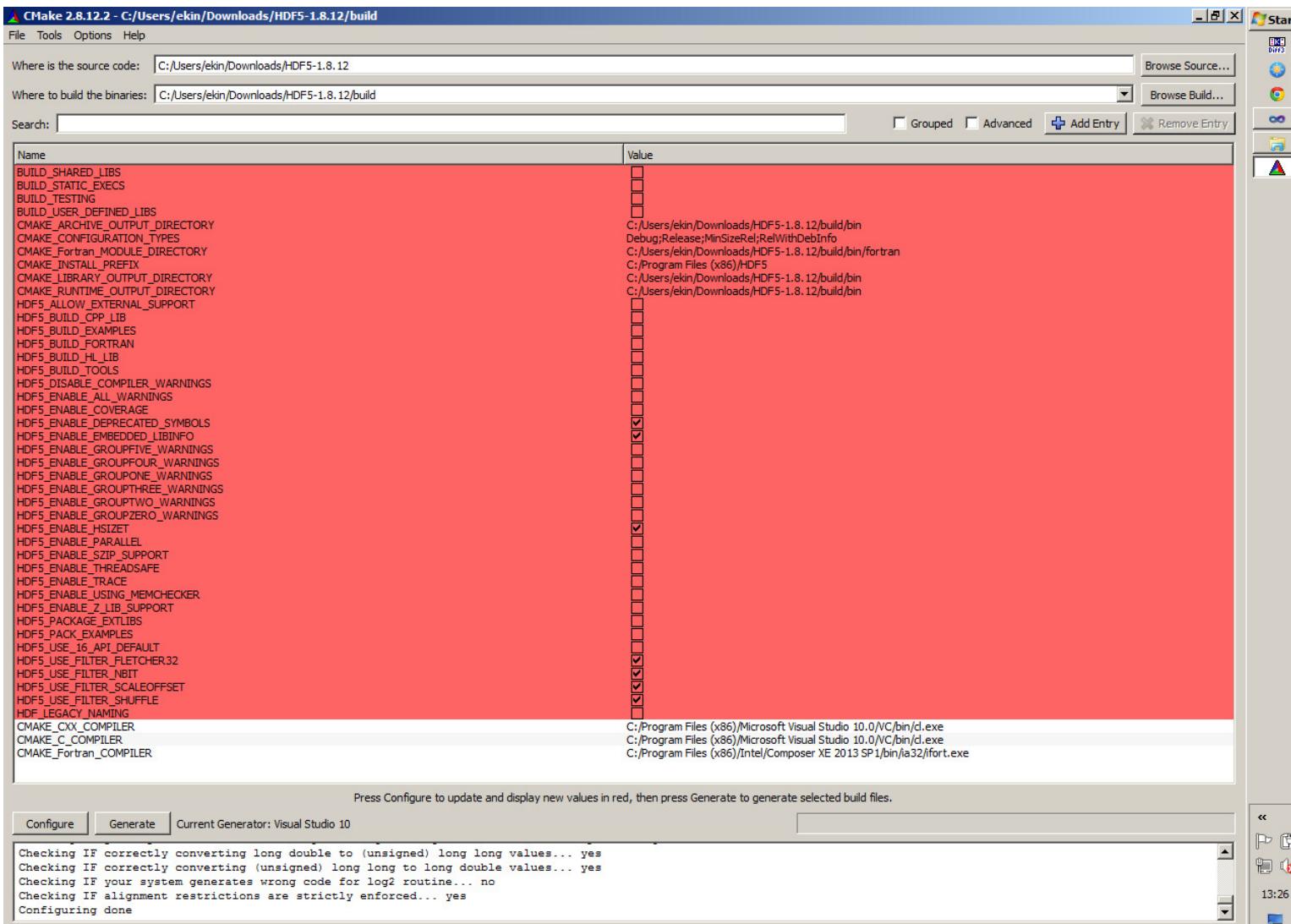


Figure 3.6: CMake: Configure; new configure options (in red).

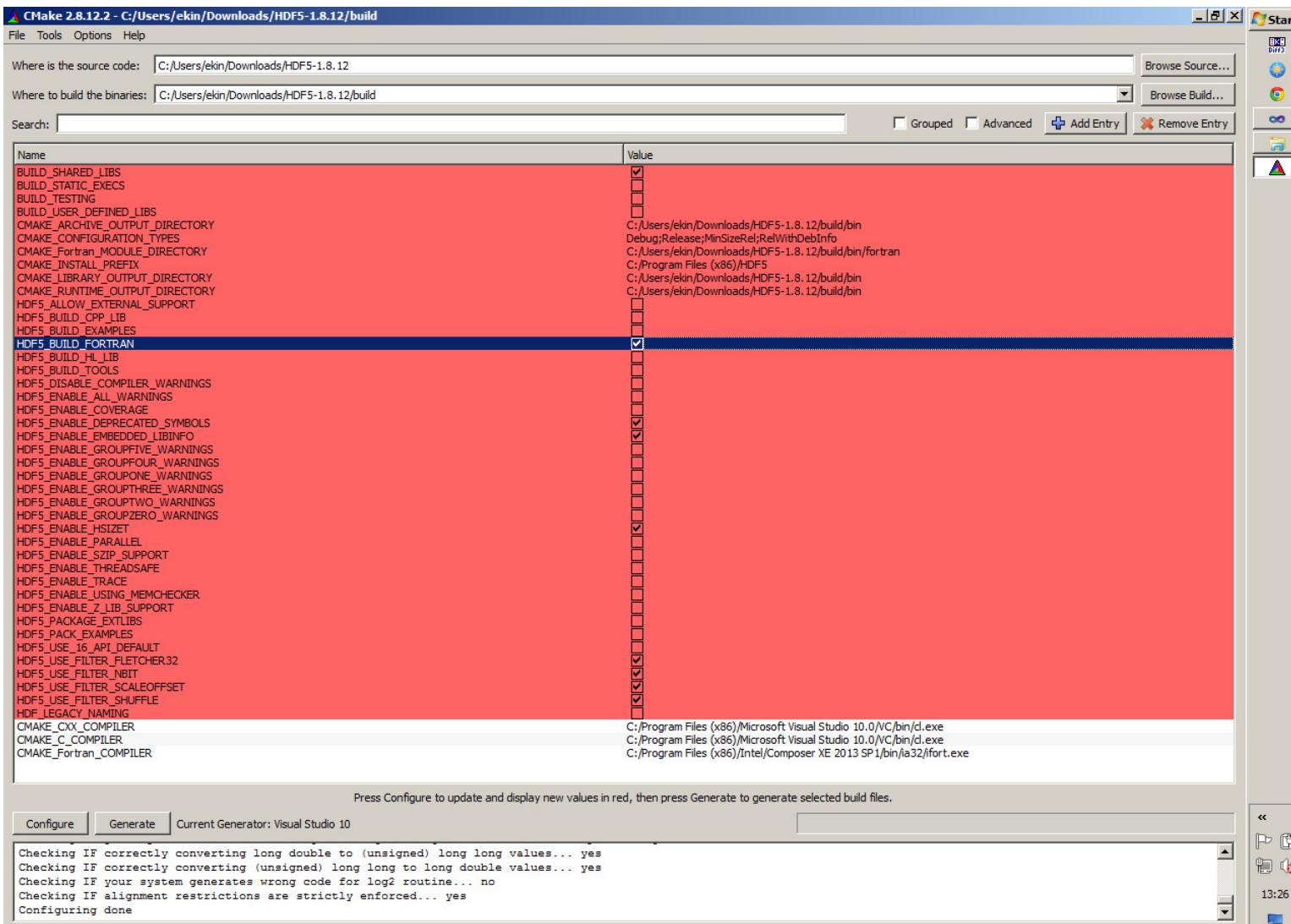


Figure 3.7: CMake: Configure; selecting new configure options for Fortran and shared library support.

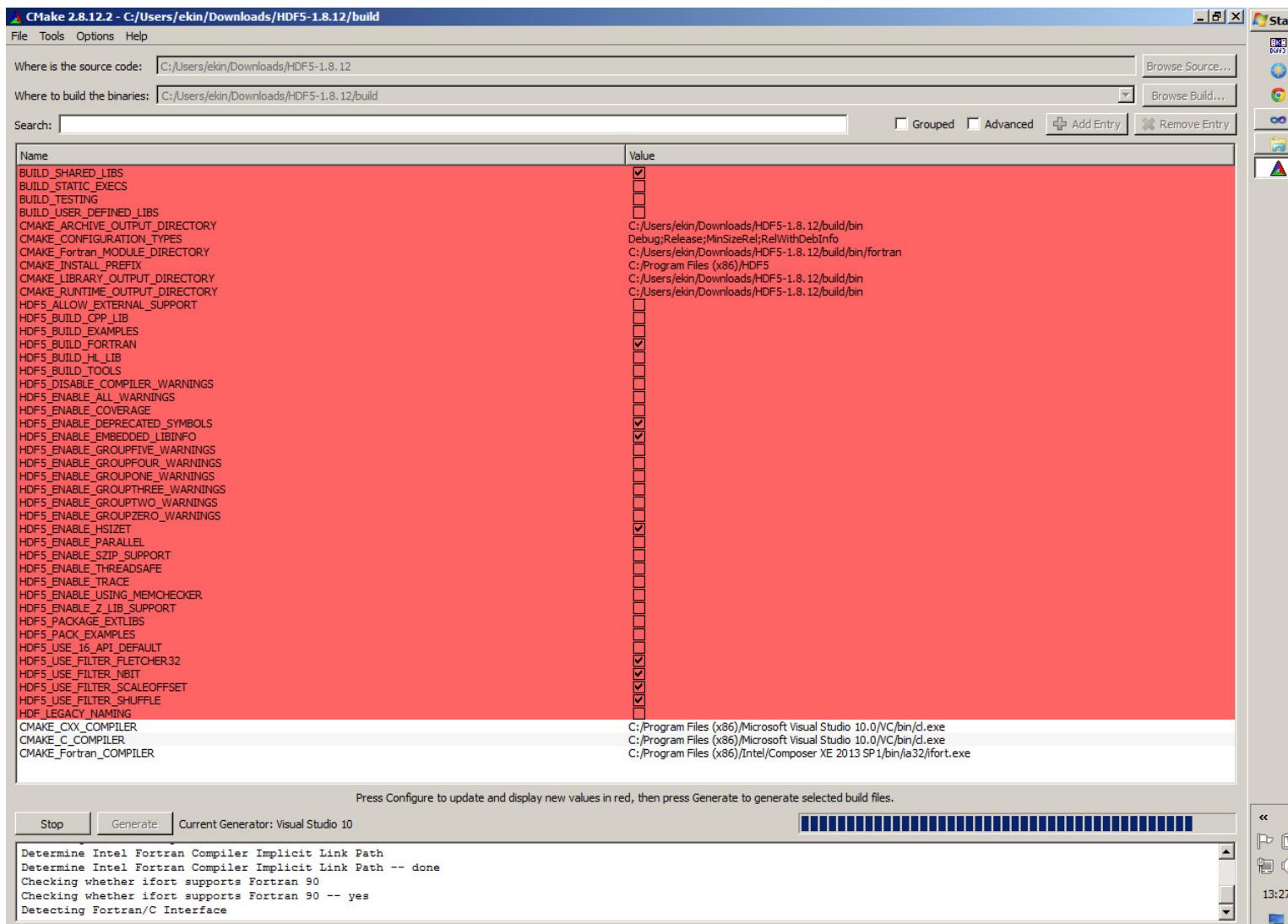


Figure 3.8: CMake: Configure; re-configuring (1).

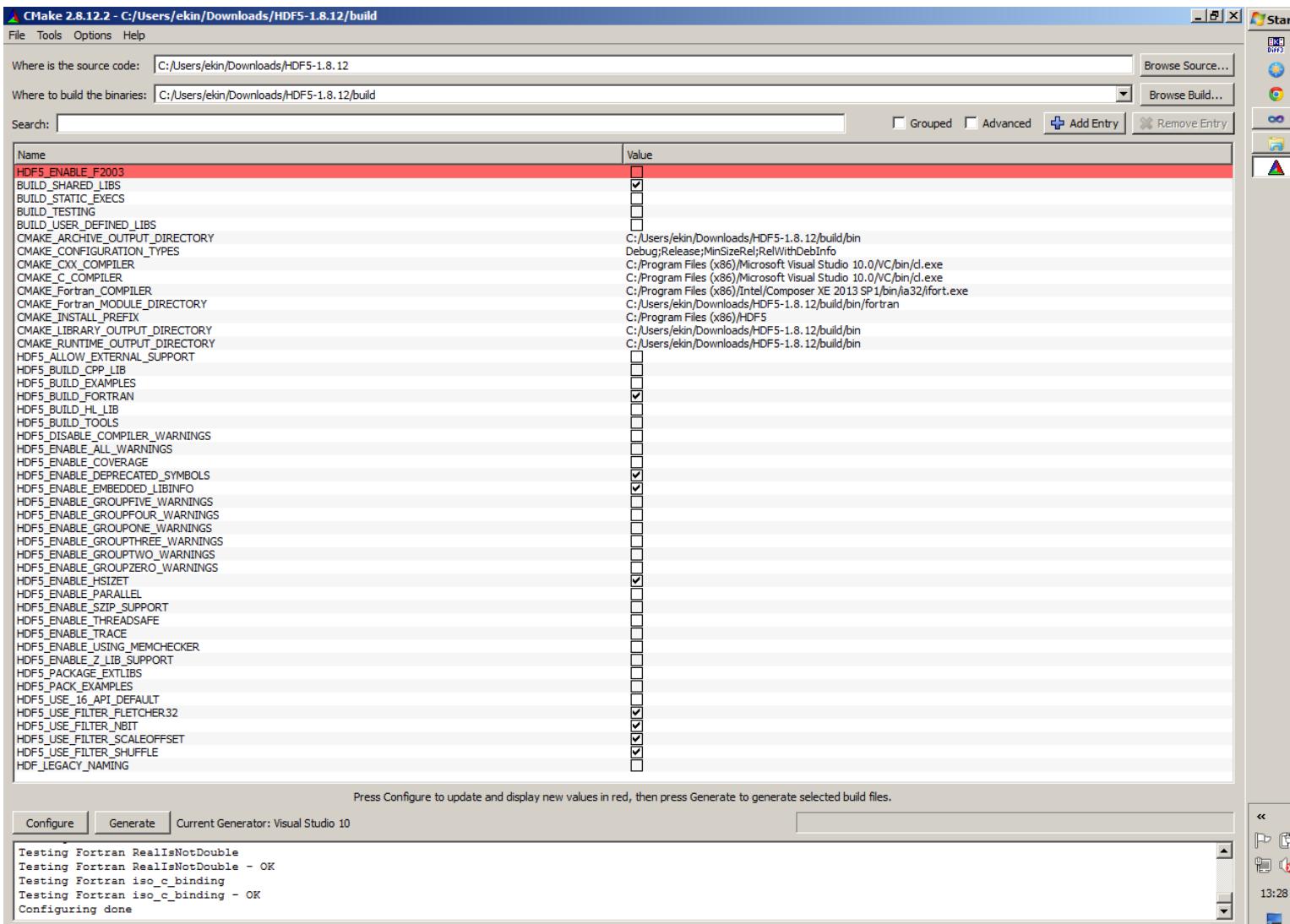


Figure 3.9: CMake: Configure; new configure option for enabling Fortran 2003 support.

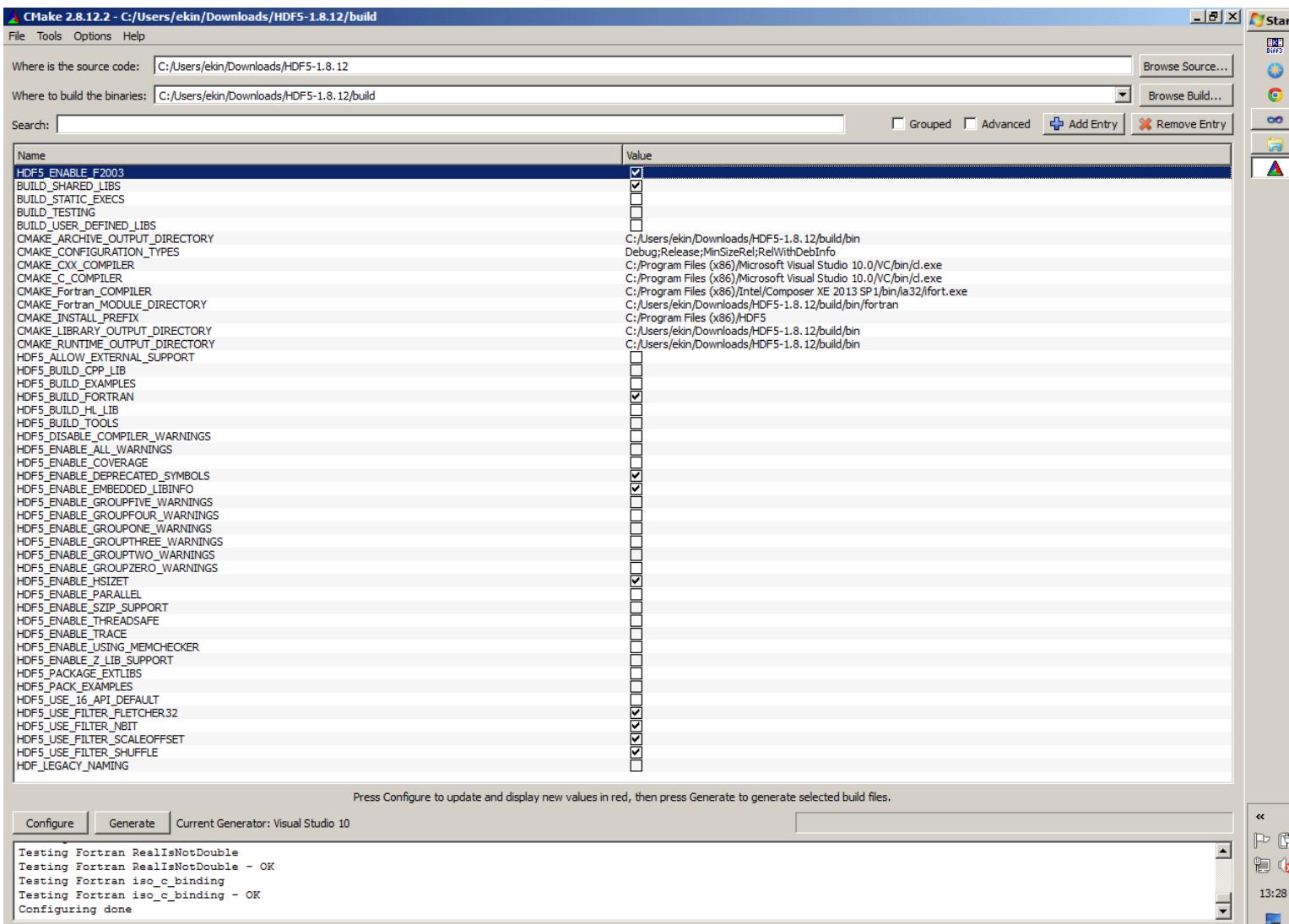


Figure 3.10: CMake: Configure; selecting new configure option for Fortran 2003 support.

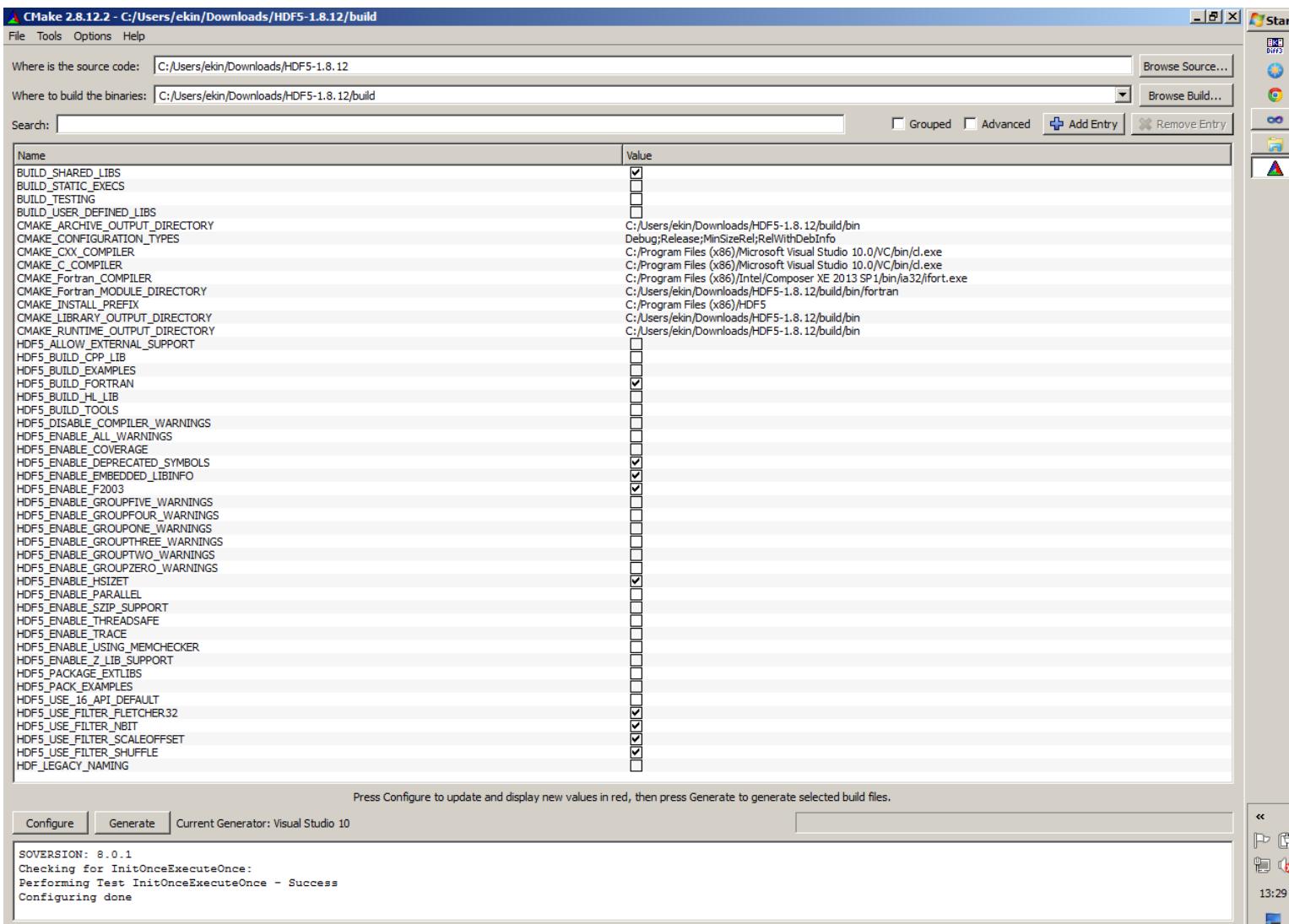


Figure 3.11: CMake: Configure; re-configuring (2).

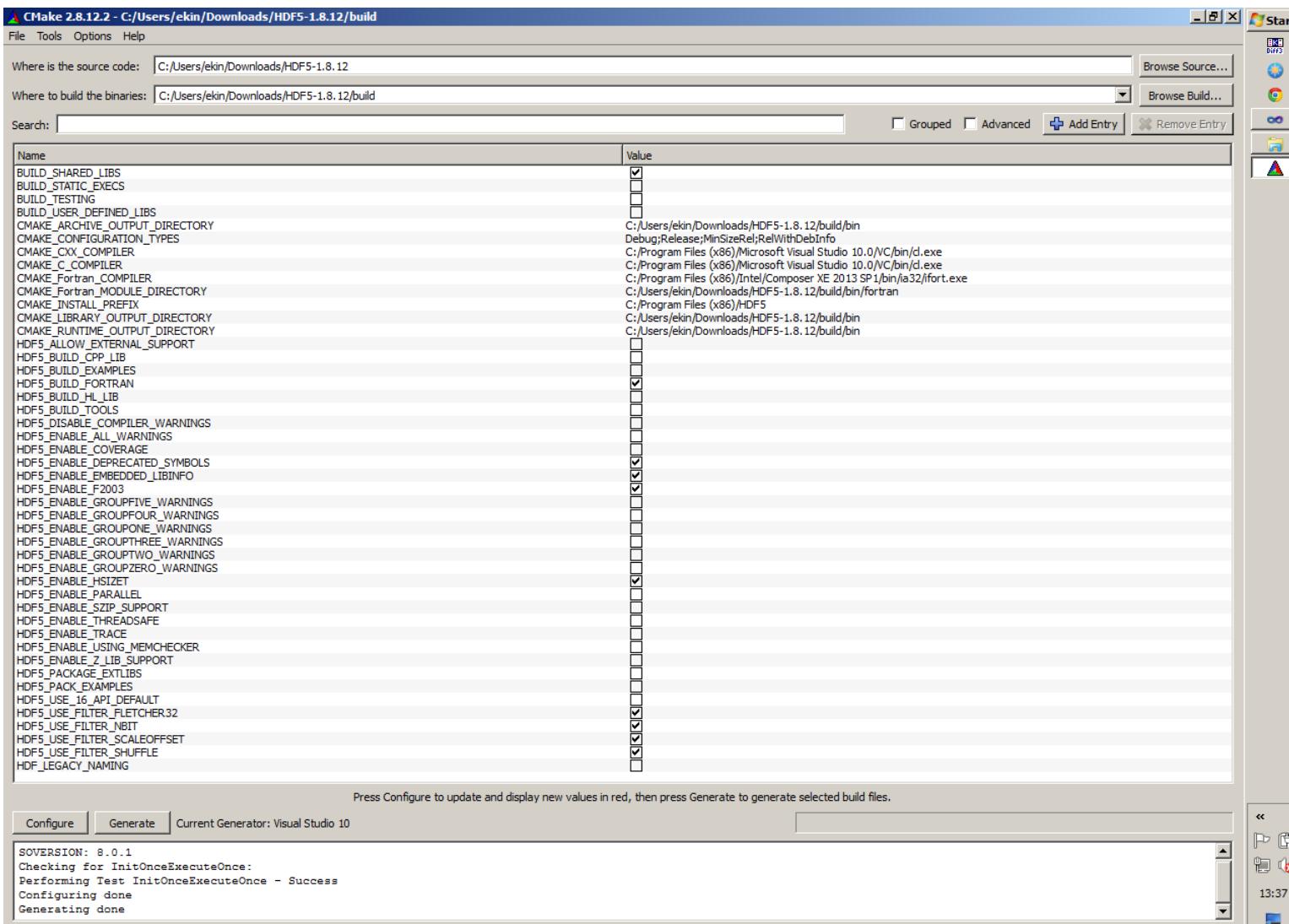


Figure 3.12: CMake: Generating.

3.1.1.2 Compiling

Now, you can launch the MS Visual Studio and open the HDF5 solution file you have just created with CMake. Launch the Visual Studio and from *File > Open > Project/Solution* (Figure 3.13), find and select the *HDF5.sln* file under the *build* folder in your HDF5 sources directory as shown in Figure 3.14. Once the solution is loaded, open the *Configuration Manager* from *Build > Configuration Manager...* as shown in Figure 3.15. Set the *Active solution configuration*: to *Release* in the opened window (Figure 3.16). Then from the project list (*Project contexts*) below the window, check-mark the *PACKAGE* project (Figure 3.17). Afterwards, you may close the *Configuration Manager*. After you finish with the *Configuration Manager*, start building HDF5 by selecting *Build > Build Solution* from the top menu (Figure 3.18). Wait until you receive the *Build succeeded* message on the bottom *Output* pane of Visual Studio (Figure 3.19). Once finished, you may close the Visual Studio and proceed to the installation of HDF5 libraries. Locate the *HDF5-1.8.XX-win32.exe* file created by the *build* process in Visual Studio under the *build* folder of your HDF5 sources directory and start the installation of the HDF5 libraries by double-clicking the *exe* file. Then follow the instructions to complete the installation. After the installation is finished, you may proceed to downloading the *EwE-F* source code (Section 3.2), setting up an *EwE-F* project in Visual Studio (Section 3.3), and then compiling and running the software (Section 3.4).

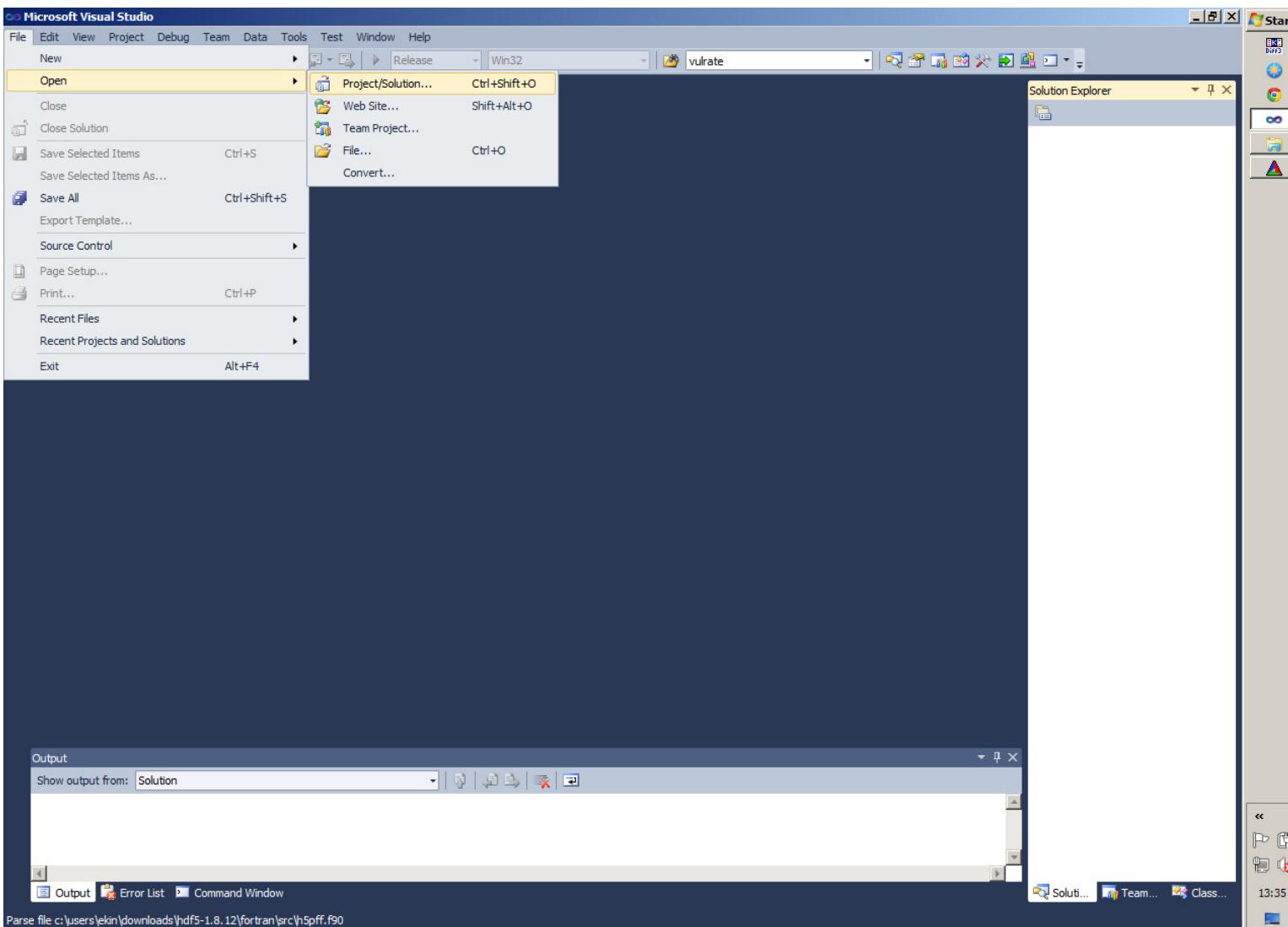


Figure 3.13: Visual Studio: Building HDF5; opening the solution (1).

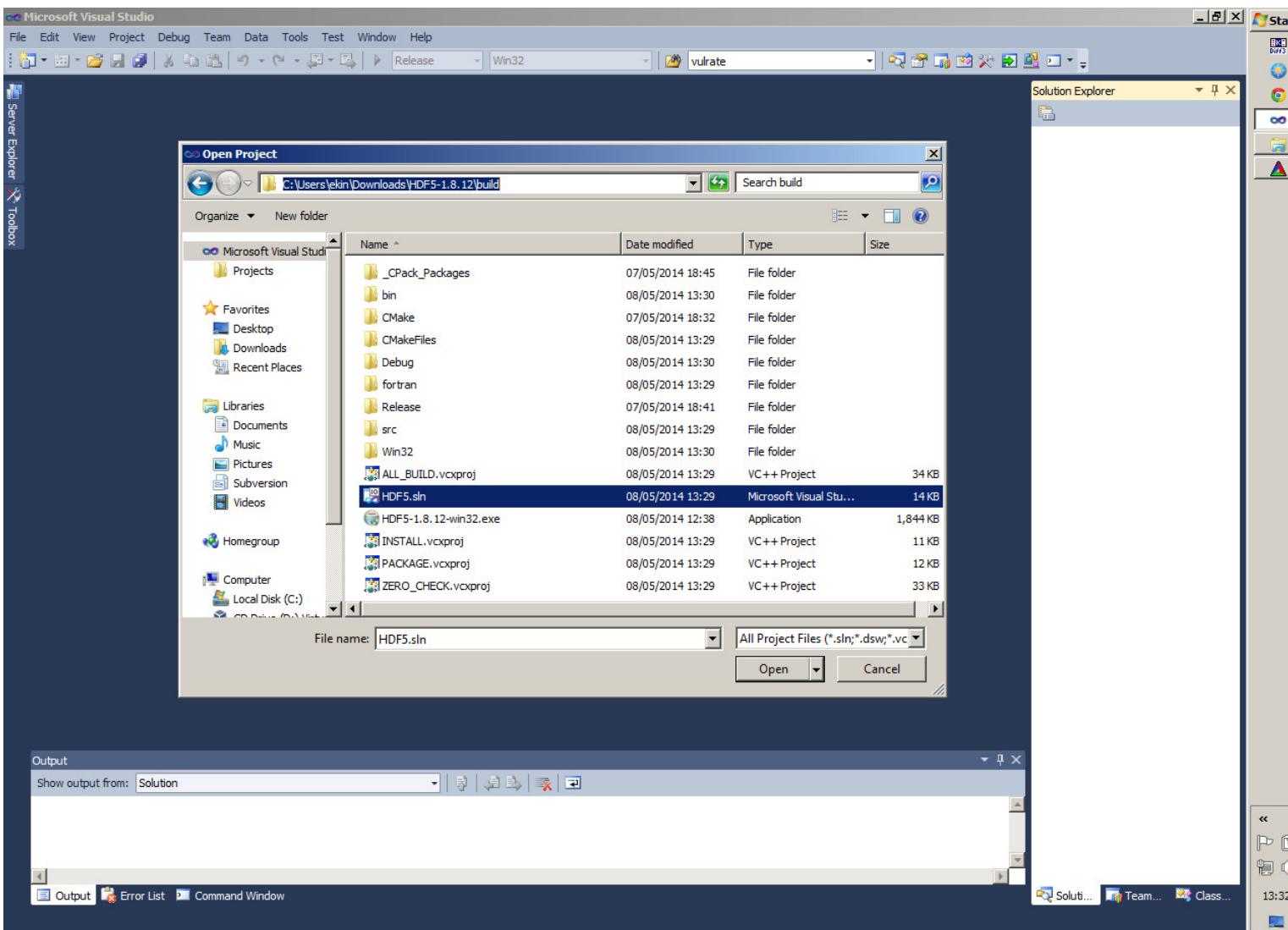


Figure 3.14: Visual Studio: Building HDF5; opening the solution (2).

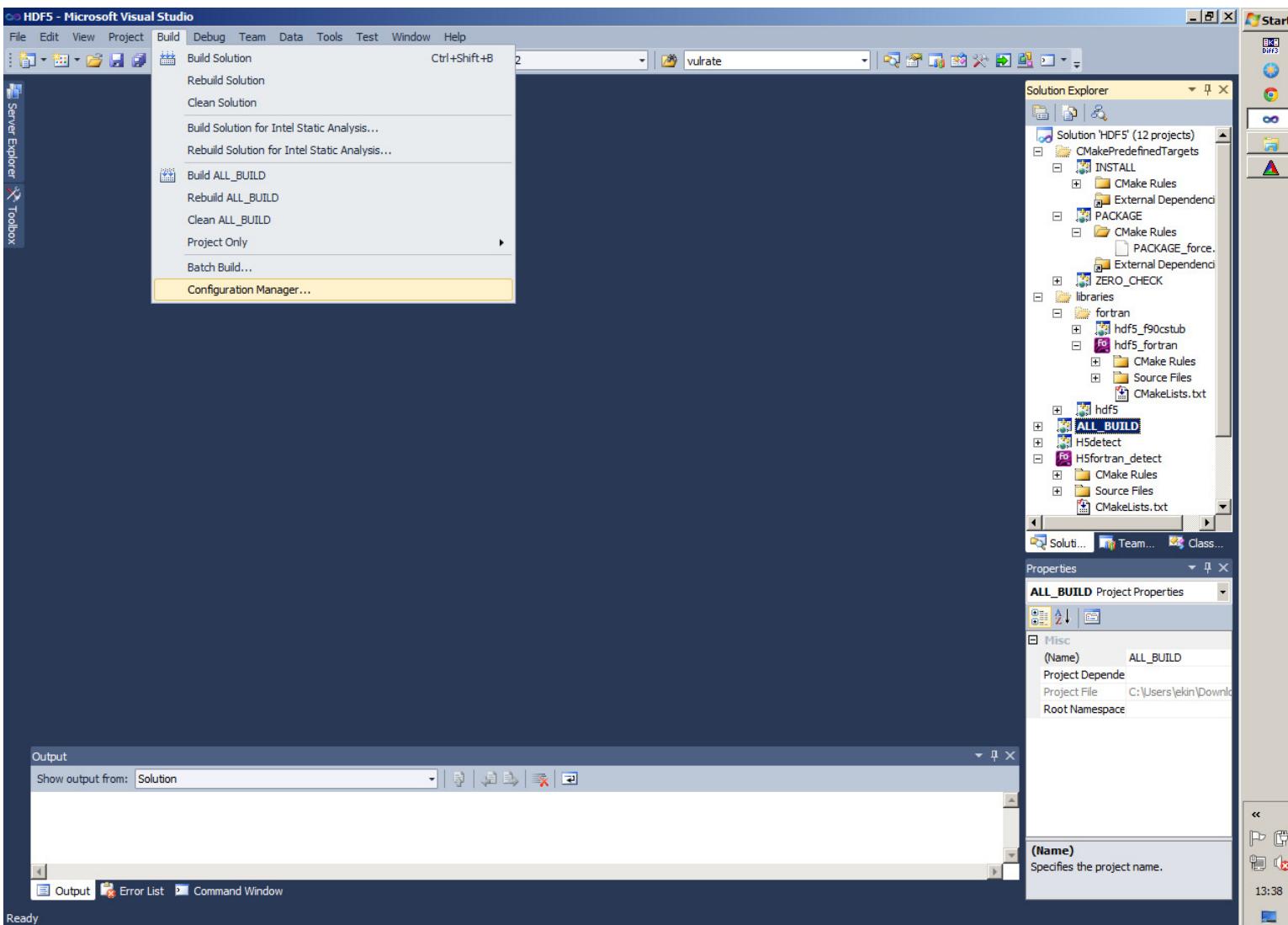


Figure 3.15: Visual Studio: Building HDF5; opening the Configuration Manager.

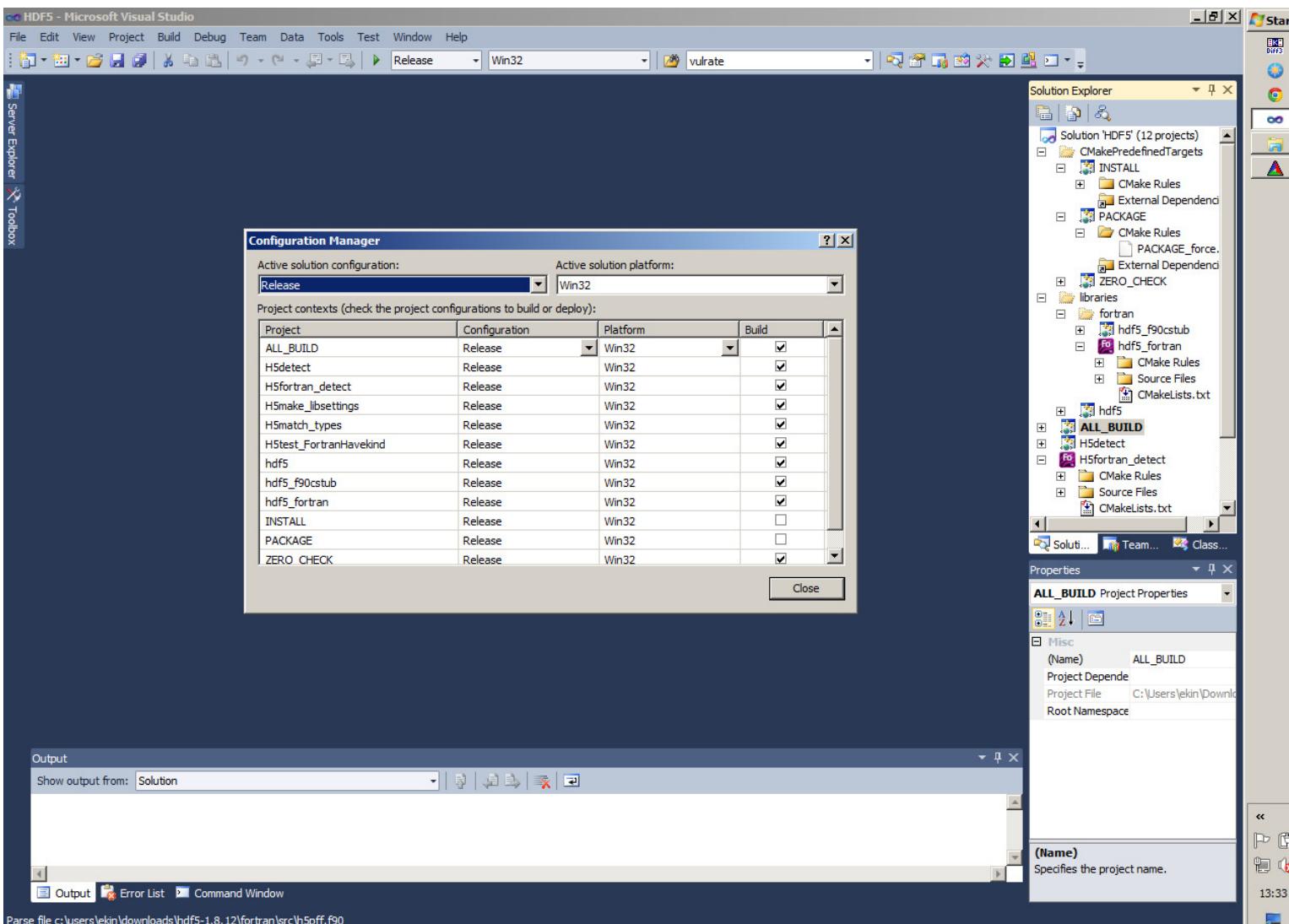


Figure 3.16: Visual Studio: Building HDF5; setting options for the Configuration Manager (1).

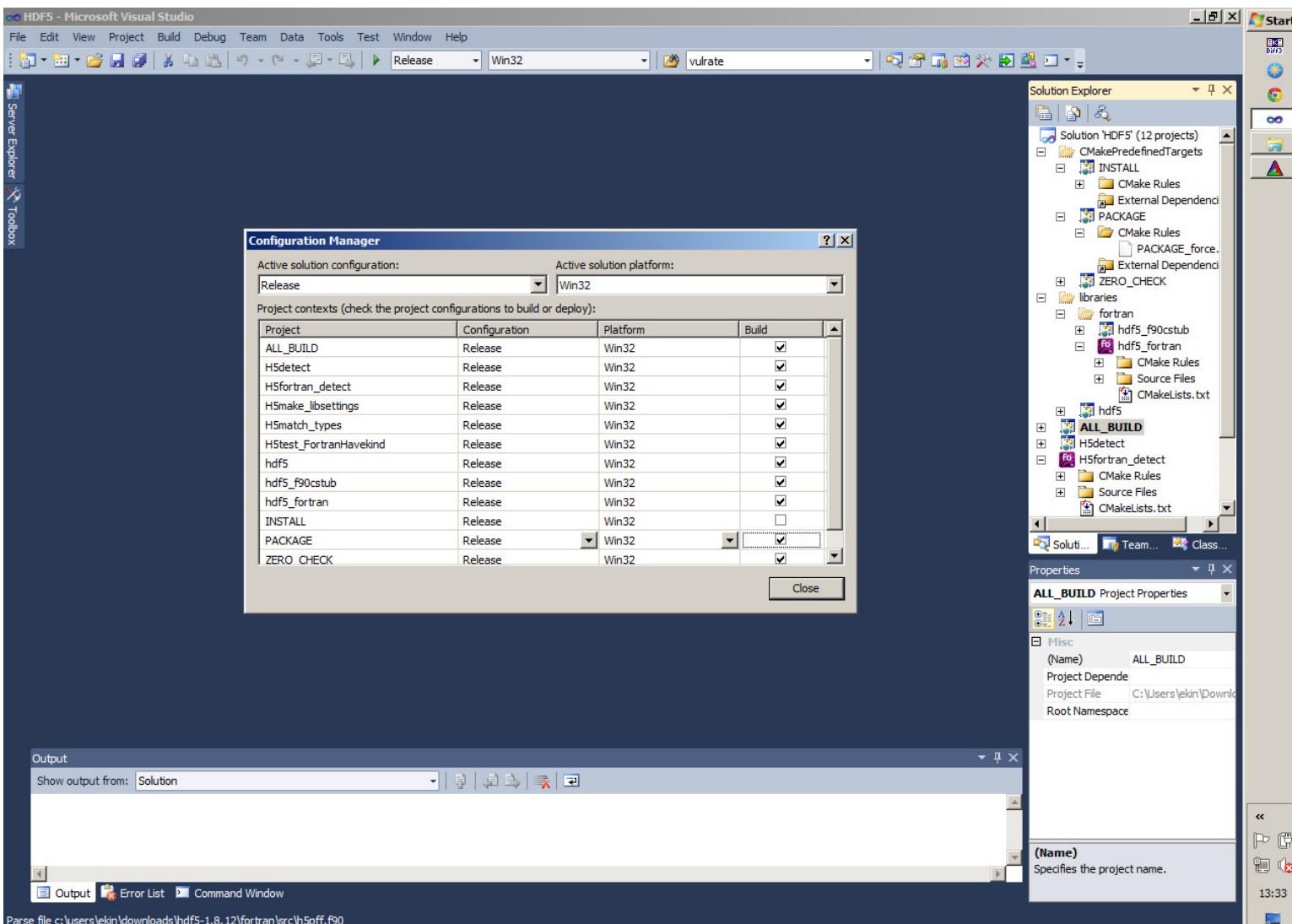


Figure 3.17: Visual Studio: Building HDF5; setting options for the Configuration Manager (2).

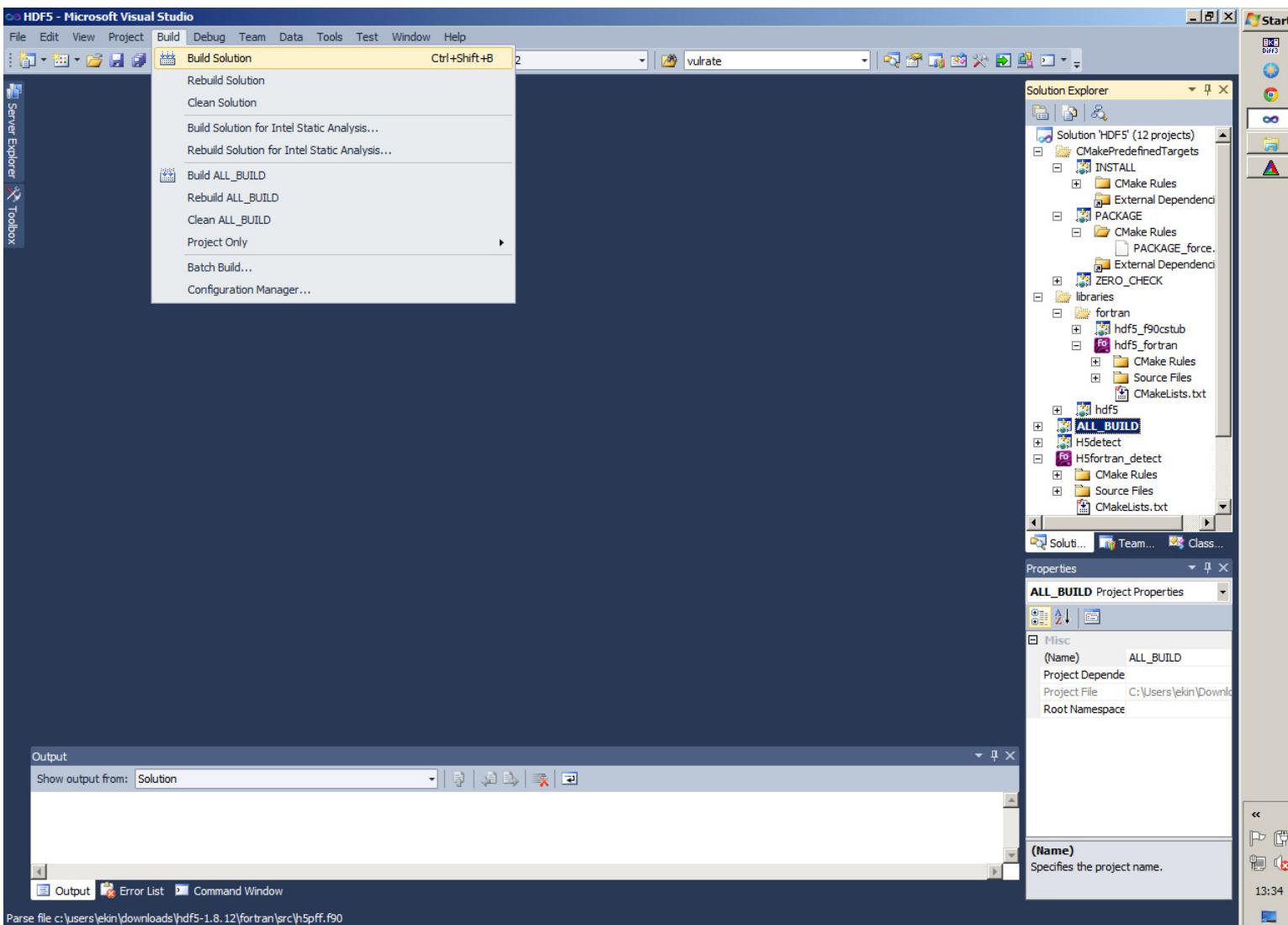


Figure 3.18: Visual Studio: Building HDF5; building.

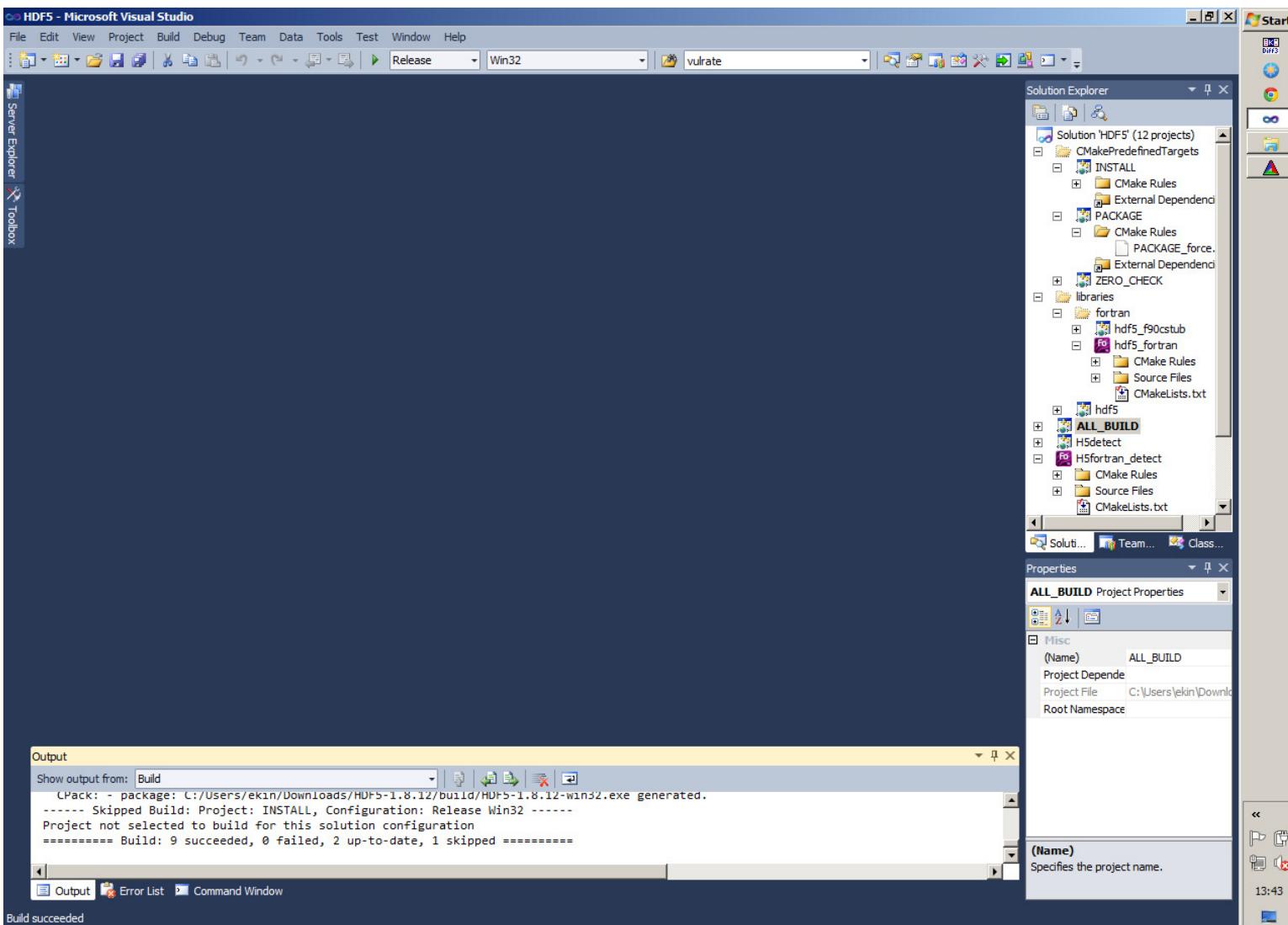


Figure 3.19: Visual Studio: Building HDF5; building completed.

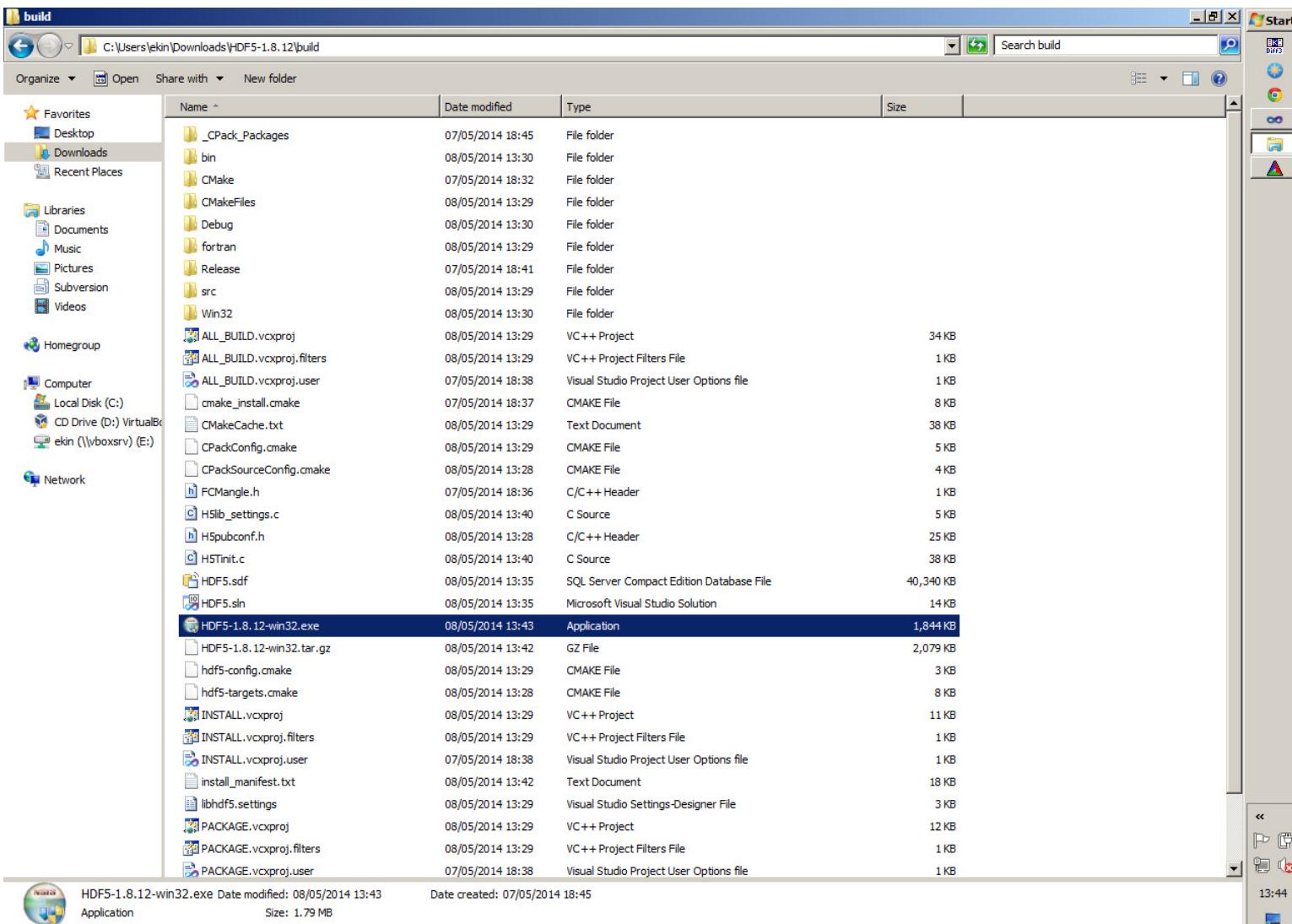


Figure 3.20: Locating and selecting the HDF5 executable for installation.

3.2 Obtaining the source code of the *EwE-F*

After the installation of the prerequisites, follow the link <https://gitlab.com/ewe-f/> to access the source code via the GitLab® web-based interface.

If you wish to access to and download the source code of the *EwE-F* via GIT version control software, first create an empty folder into which you would like to download the *EwE-F* source code. Then navigate to this directory with Windows Explorer and using your mouse right-click an empty area in the folder to open the context menu where you will find the GIT commands (Figure 3.21). Select *GIT Bash Here*. A GIT command window will be opened as in Figure 3.22. Type the command shown in Listing 3.1 in order to obtain *Ecopath-F* source code.

Listing 3.1: Downloading *Ecopath-F*

```
git clone https://gitlab.com/ewe-f/ecopath-f.git
```

Following, type the command shown in Listing 3.2 in order to obtain *Ecosim-F* source code, which also includes *Eospace-F*.

Listing 3.2: Downloading *Ecosim-F*

```
git clone https://gitlab.com/ewe-f/ecosim-f.git
```

The source code will be downloaded to your local directory (Figure 3.23). For the explanation of the folders see Section 4.1 and for the explanation of the files see Section 4.2 for *Ecopath-F* and sections 4.3 and 4.4 for *Ecosim-F* and *Eospace-F* respectively.

For this demonstration, we used the *Ecosim-F* source code as the example. Once finished, a folder named *ecosim-f* will be created in your local directory (Figure 3.23). This folder includes the full source code of the *Ecosim-F*. Once you navigate into it, you will see a folder structure similar to the one shown in Figure 3.24. This folder structure is fully explained in the *EwE-F User's Manual PDF* file that you may find under this folder. Completing steps above, you may proceed to Section 3.3.

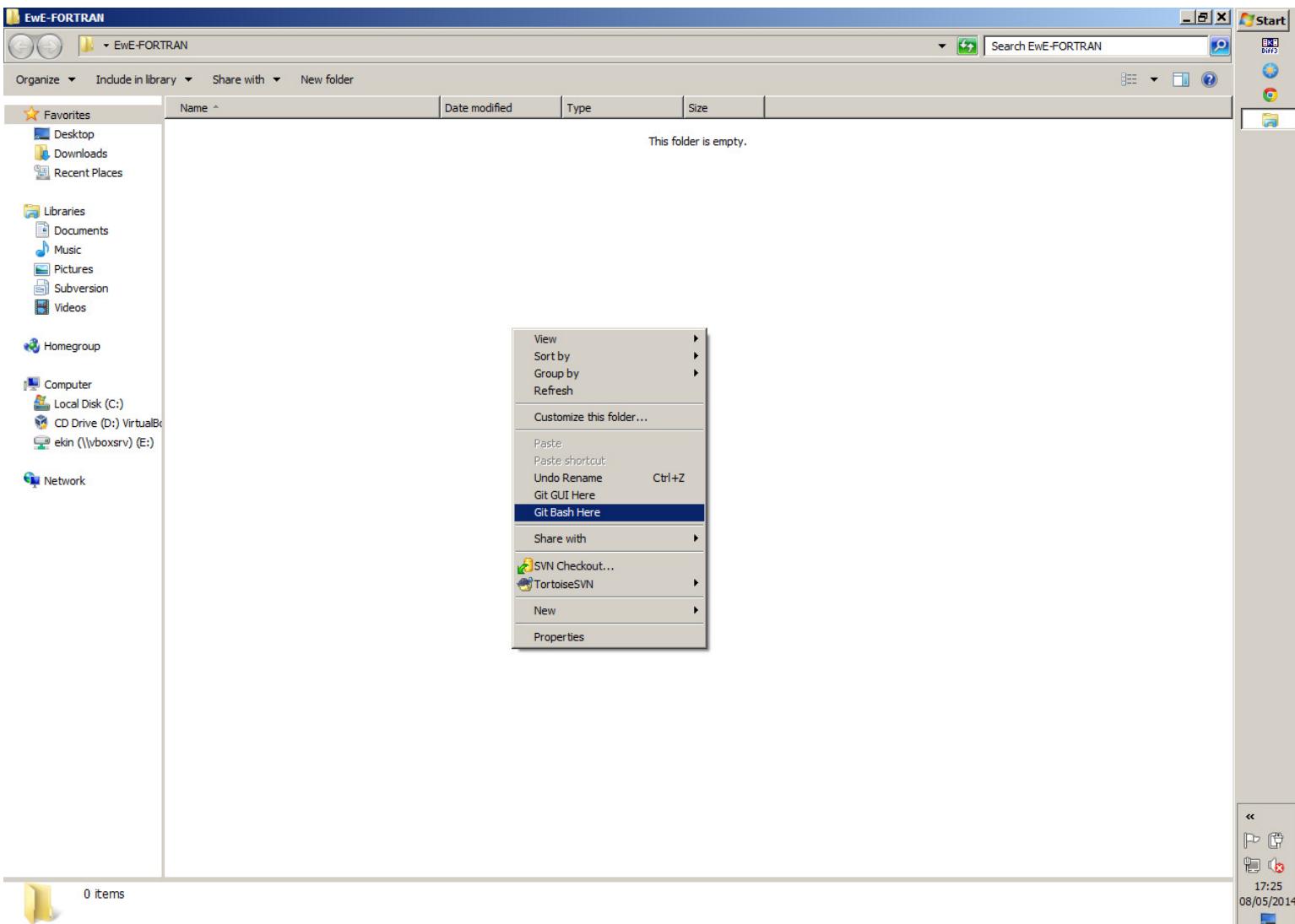


Figure 3.21: GIT: opening GIT command window in the empty directory.

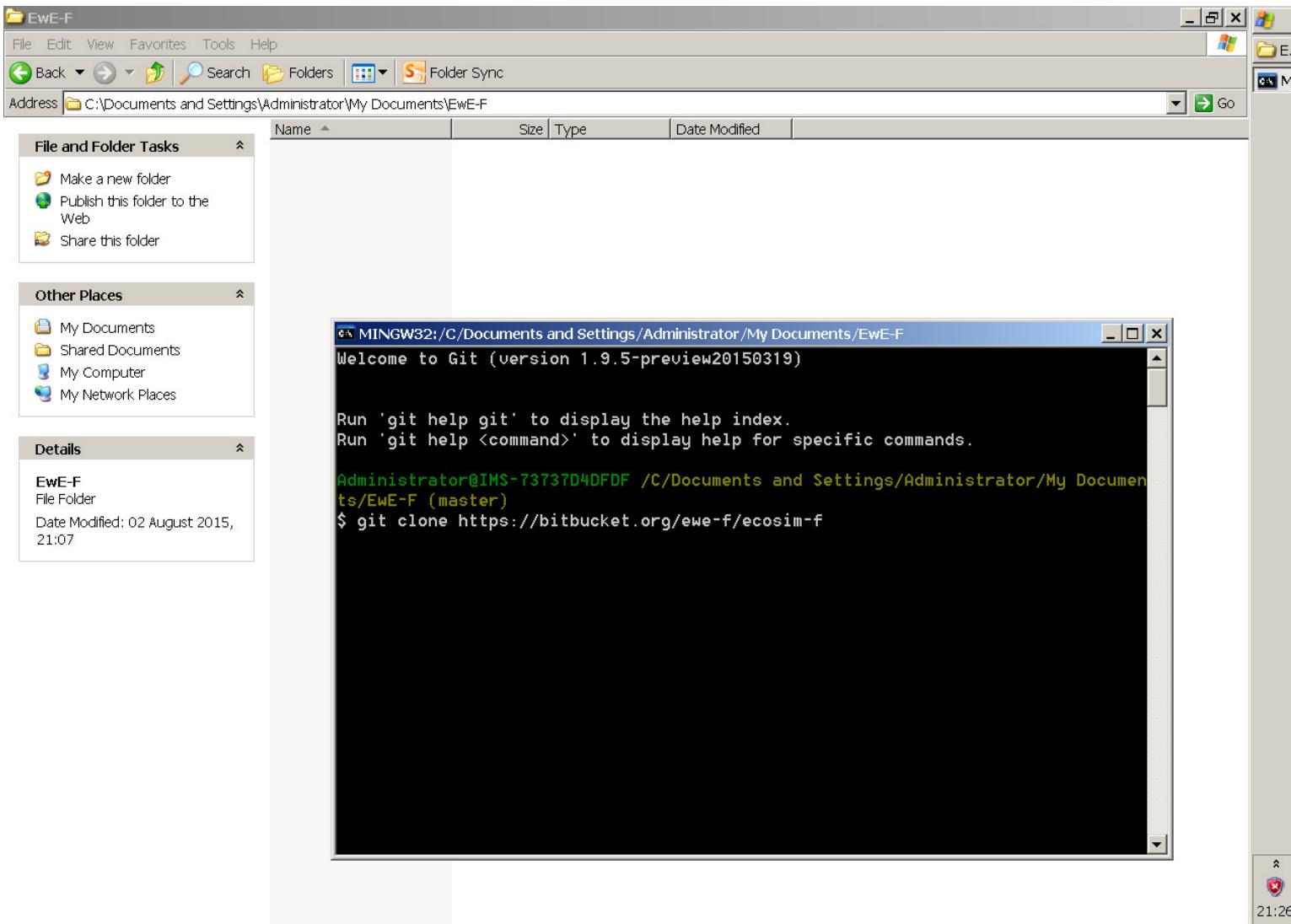


Figure 3.22: GIT: command window and entering command to download sources.

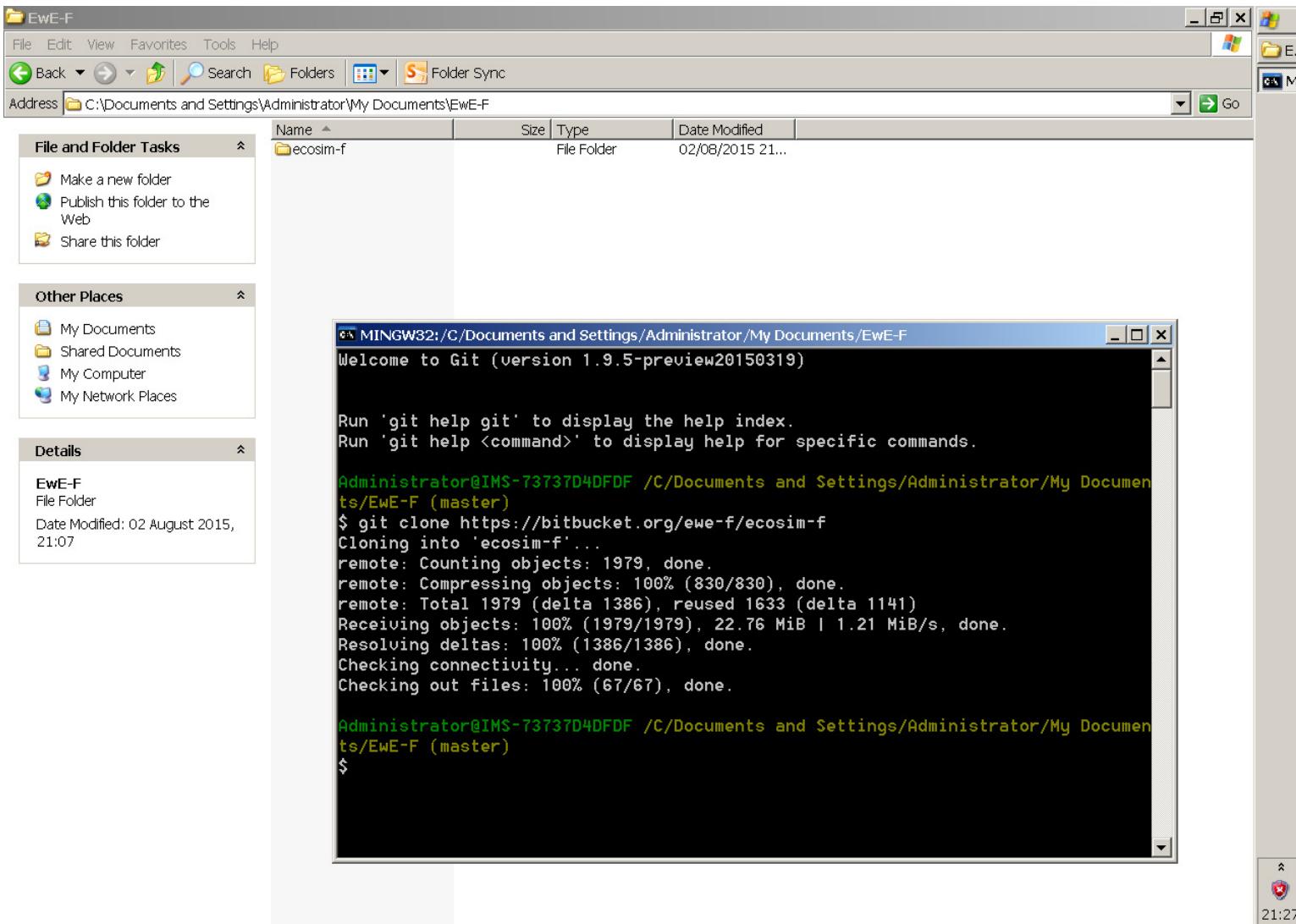


Figure 3.23: GIT: downloading sources done, the new *ecosim-f* directory.

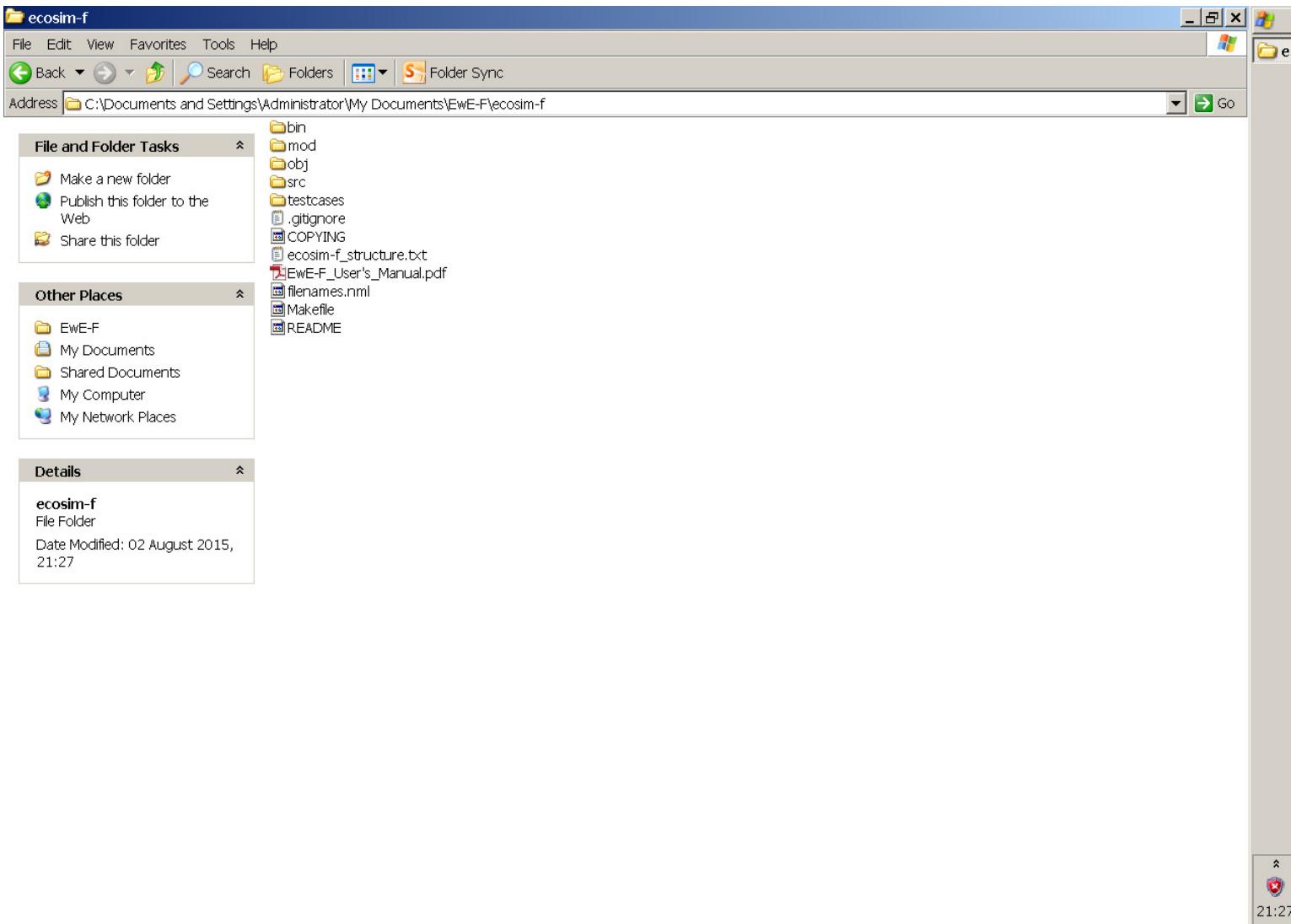


Figure 3.24: The *ecosim-f* directory structure.

3.3 Setting up an *EwE-F* project in MS Visual Studio 2010

First, launch the MS Visual Studio. Then, from the menu select *File > Open > Project...* (Figure 3.25). The *New Project* window will be opened. From the left pane of this window, select *Console Application* under the *Intel(R) Visual Fortran* tree item (Figure 3.26). In the middle pane, select *Empty Project*. Rename your project using the *Name* field located below the window. For this example, because we use the *Ecosim-F*, we named our *EwE-F* project as *ecosim-fortran*.) Then click *OK* push-button on the bottom right. A new project will be created. Following, from the *Solution Explorer* pane on the right side of the Visual Studio window (Figure 3.27), right-click *Source Files* item and from the context menu select *Add > Existing Item...* as shown in Figure 3.27. From the opened file explorer dialog box, navigate to the *src* directory in the folder where you downloaded the *Ecosim-F* sources. In the *src* directory, select all of the files with the *.F90* extension and then click the button *Add* on bottom right (Figure 3.28). After successfully completing this step, you must now see the *.F90* source files listed under the *Source Files* item in the *Solution Explorer* window as shown in Figure 3.29.

After this step, in the *Solution Explorer* pane, right-click on the name of your project (*ecosim-fortran*) as shown in Figure 3.30 and select *Properties* from the context menu. A new window with the project properties will be opened (Figure 3.31). In this window, using the tree view on the left pane, select *General* under the *Fortran* tree item. Then on the right pane, set the *Additional include directories* (highlighted blue in Figure 3.31) to the *include* directory of your HDF5 installation as shown. Next, select *Preprocessor* on the left pane and set *Preprocess source file* option on the right pane (highlighted blue in Figure 3.32) to *Yes (/fpp)* as shown. Following, select *General* under the *Linker* tree item on the left pane (Figure 3.33). Then on the right pane, set the *Additional library directories* (highlighted blue in Figure 3.33) to the *lib* directory of your HDF5 installation as shown. As the last step, now select *Input* under the *Linker* tree item on the left pane (Figure 3.34).

Then on the right pane, set the *Additional dependencies* (highlighted blue in Figure 3.34) to the *hdf5_fortran.lib* as shown in the figure. Now you may proceed to Section 3.4 to build and run the *Ecosim-F*.

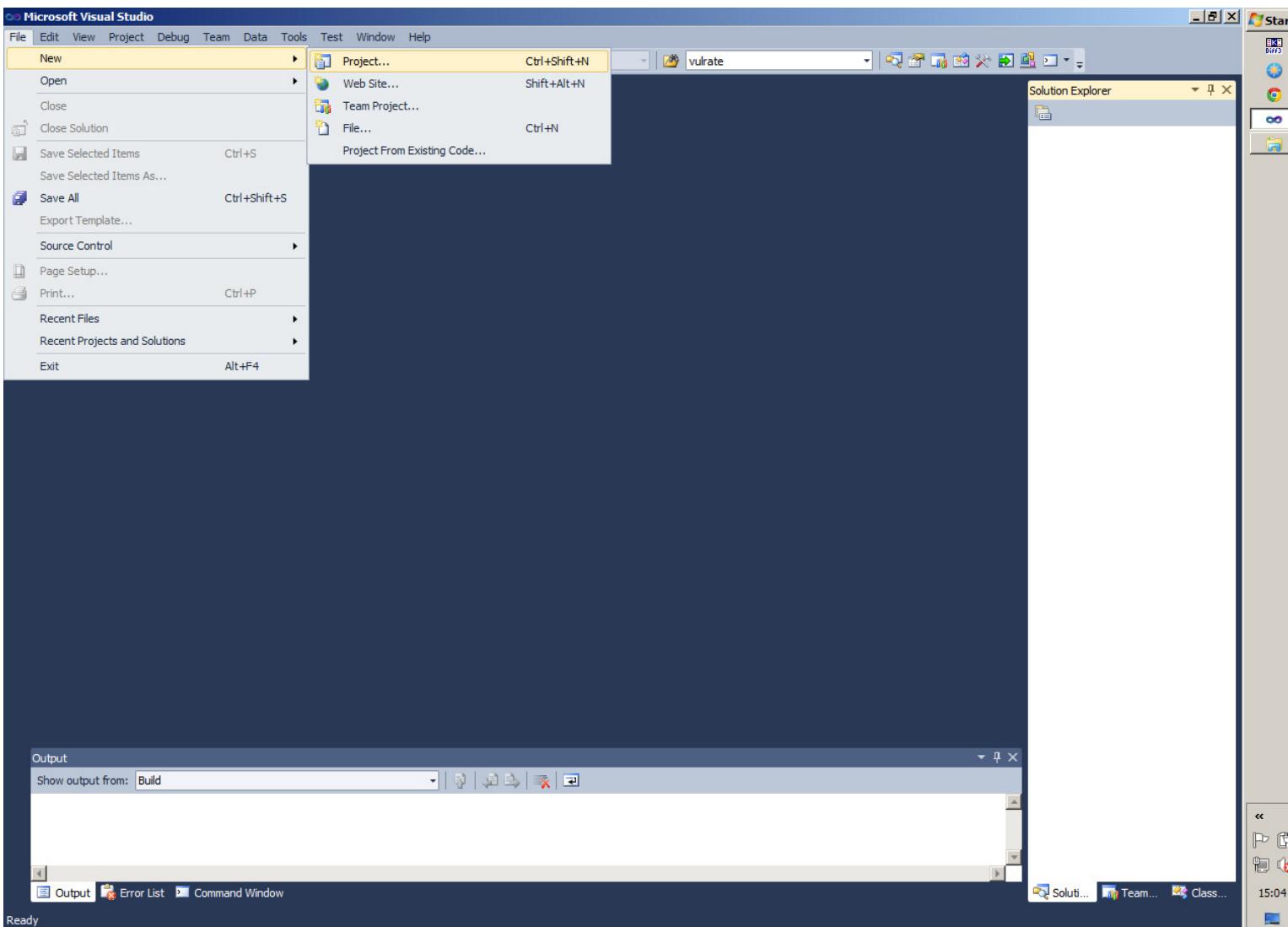


Figure 3.25: Visual Studio: *Ecosim-F*; creating project (1).

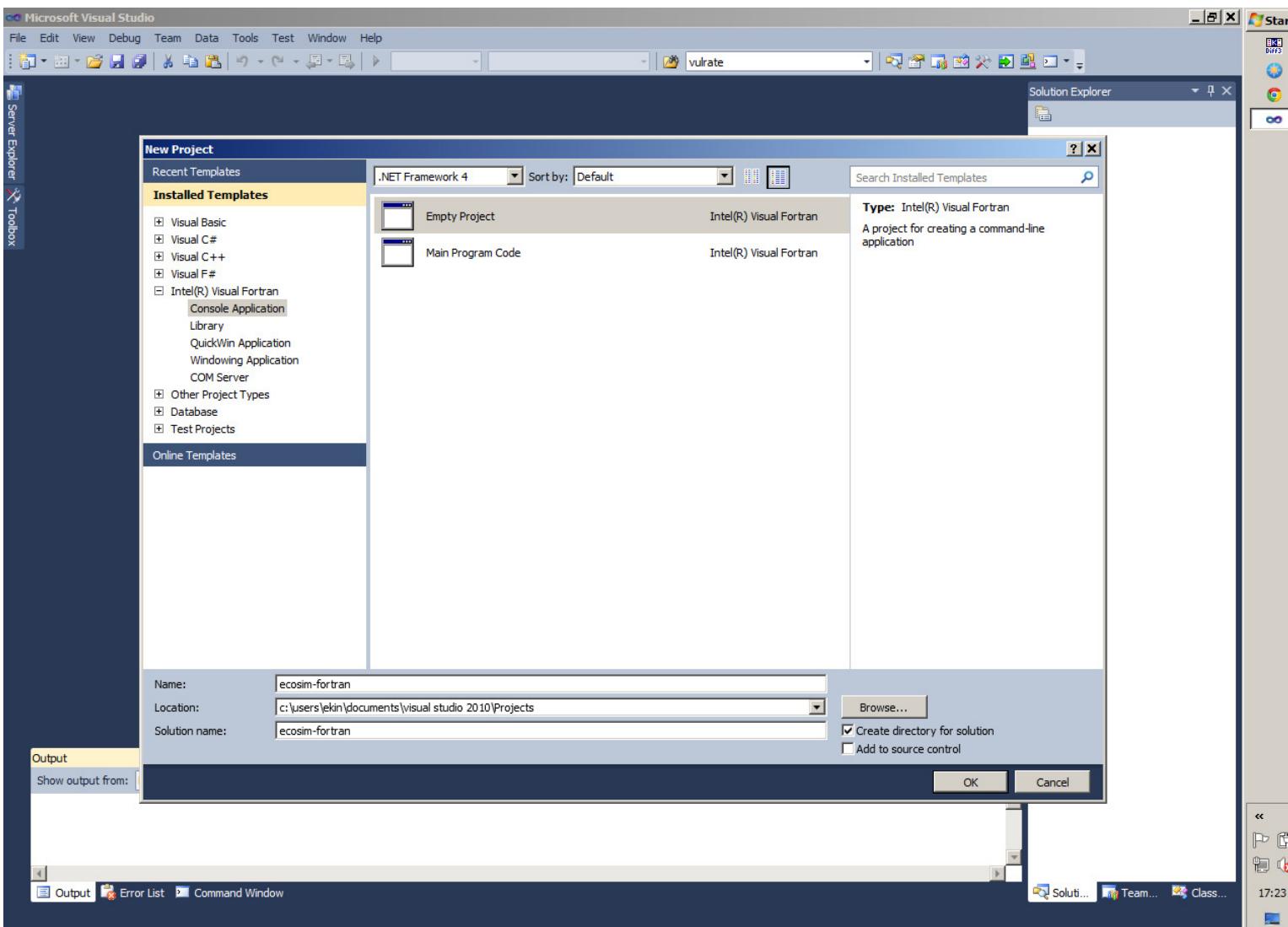


Figure 3.26: Visual Studio: *Ecosim-F*; creating project (2).

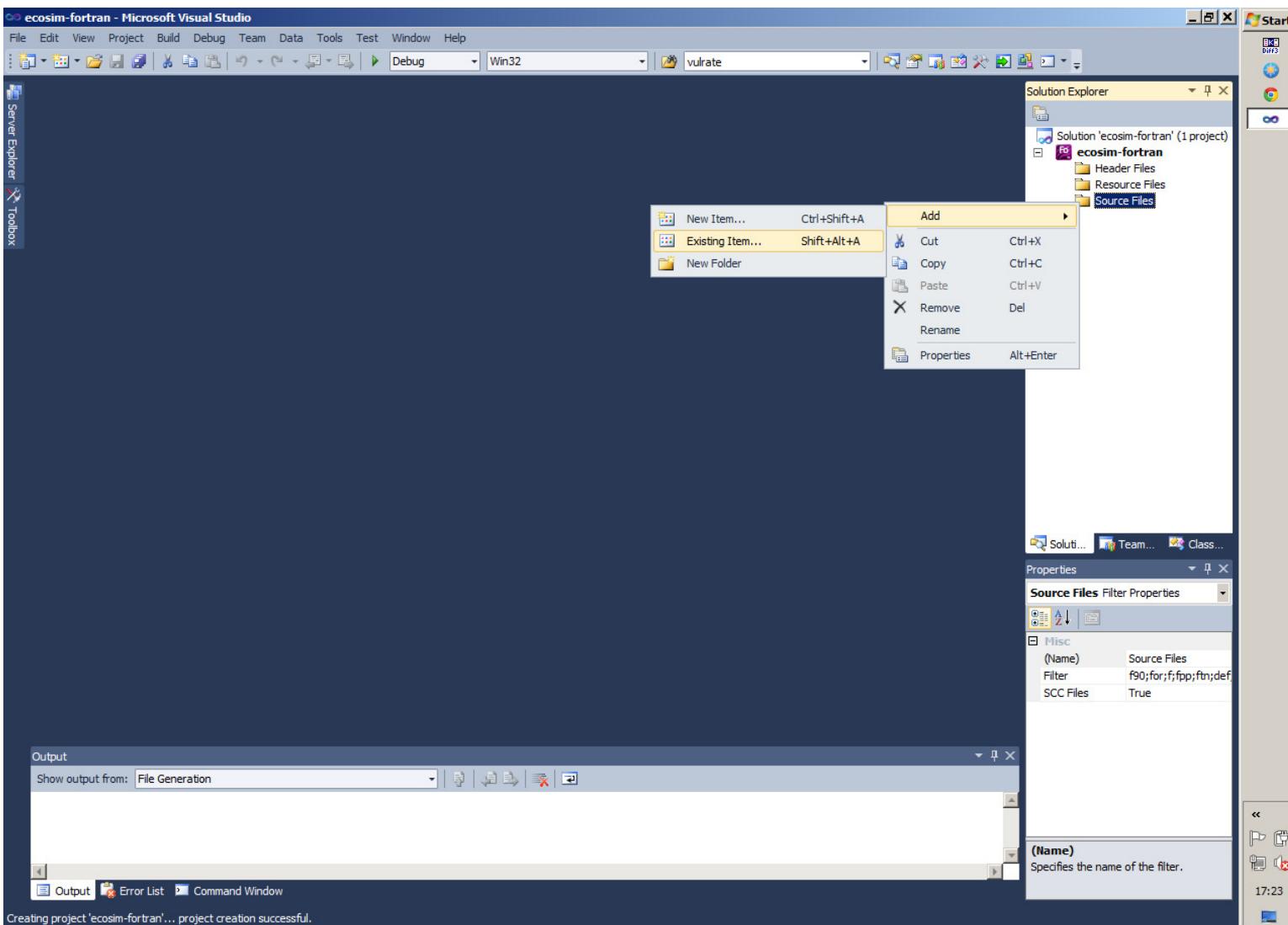


Figure 3.27: Visual Studio: *Ecosim-F*; adding source files to the project (1).

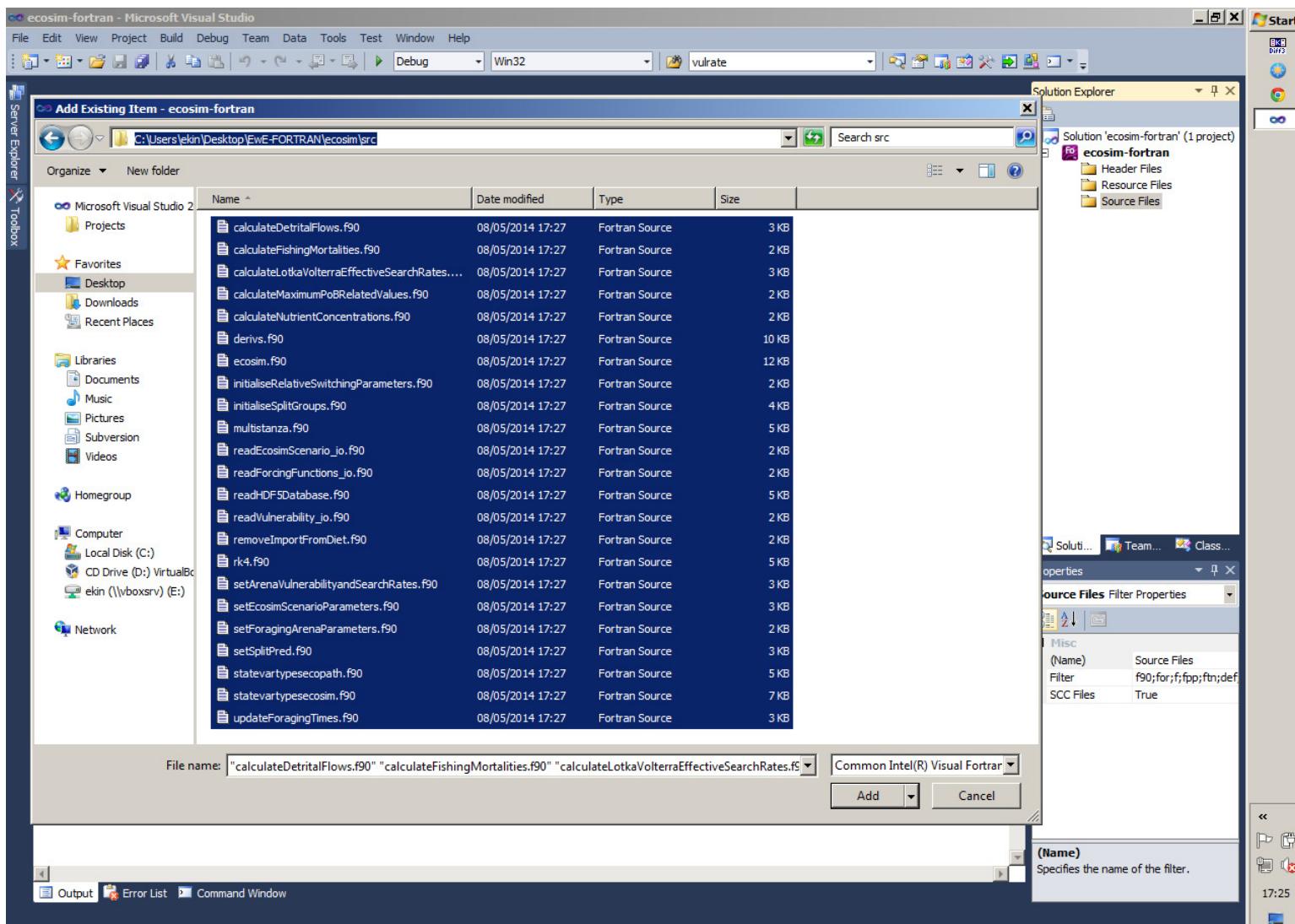


Figure 3.28: Visual Studio: *Ecosim-F*; adding source files to the project (2).

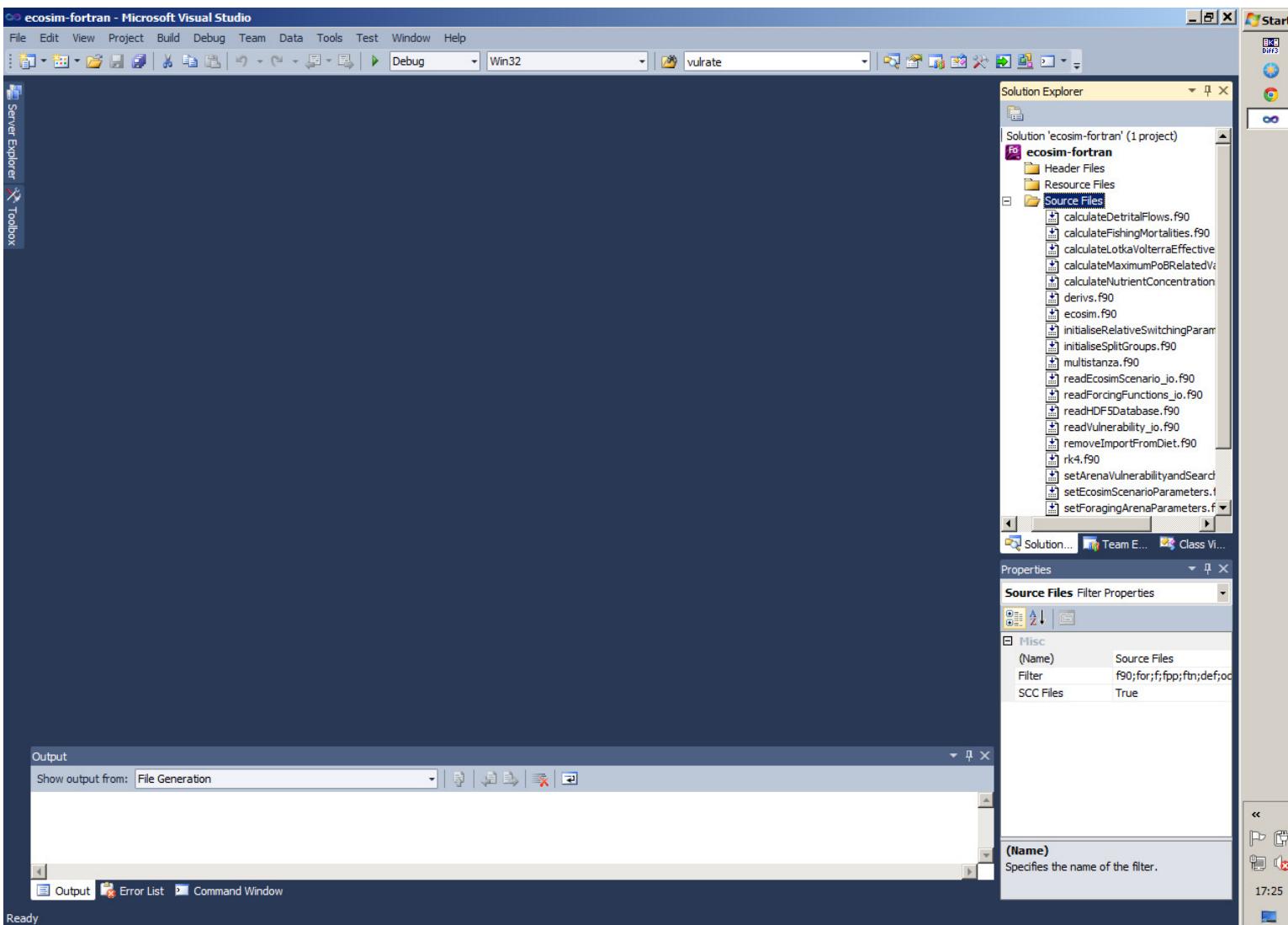


Figure 3.29: Visual Studio: *Ecosim-F*; source files added to the project.

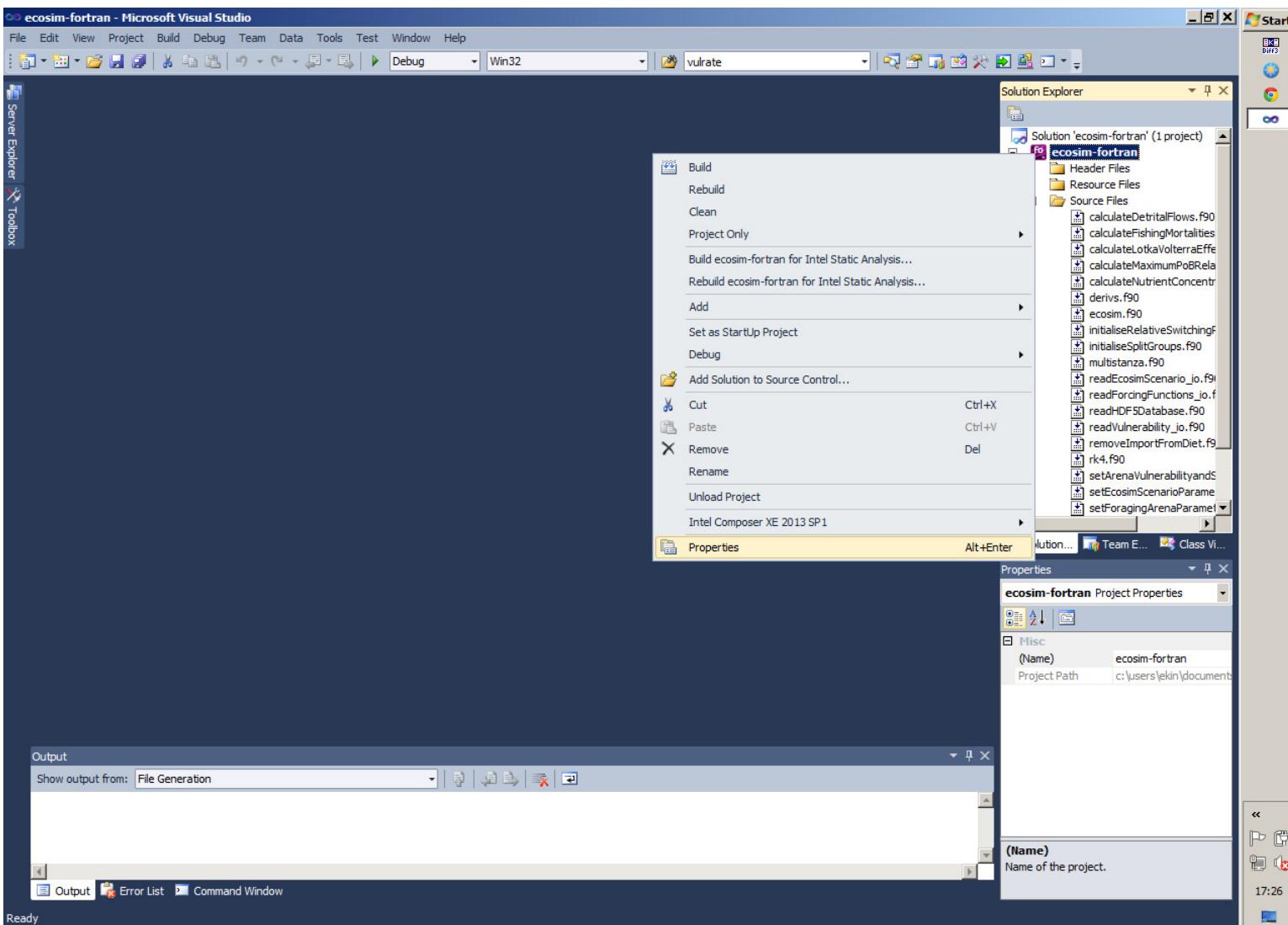


Figure 3.30: Visual Studio: *Ecosim-F*; opening project properties.

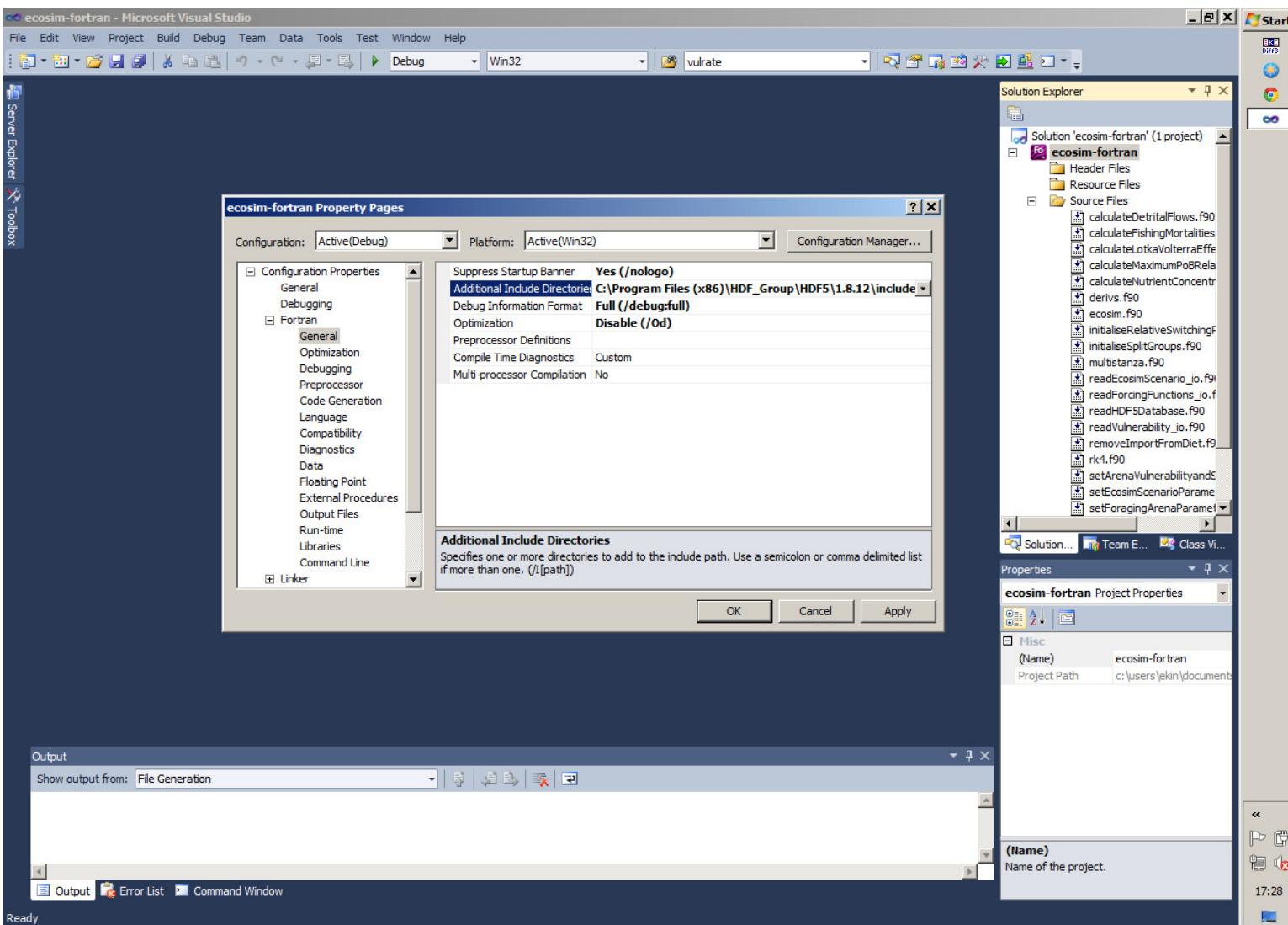
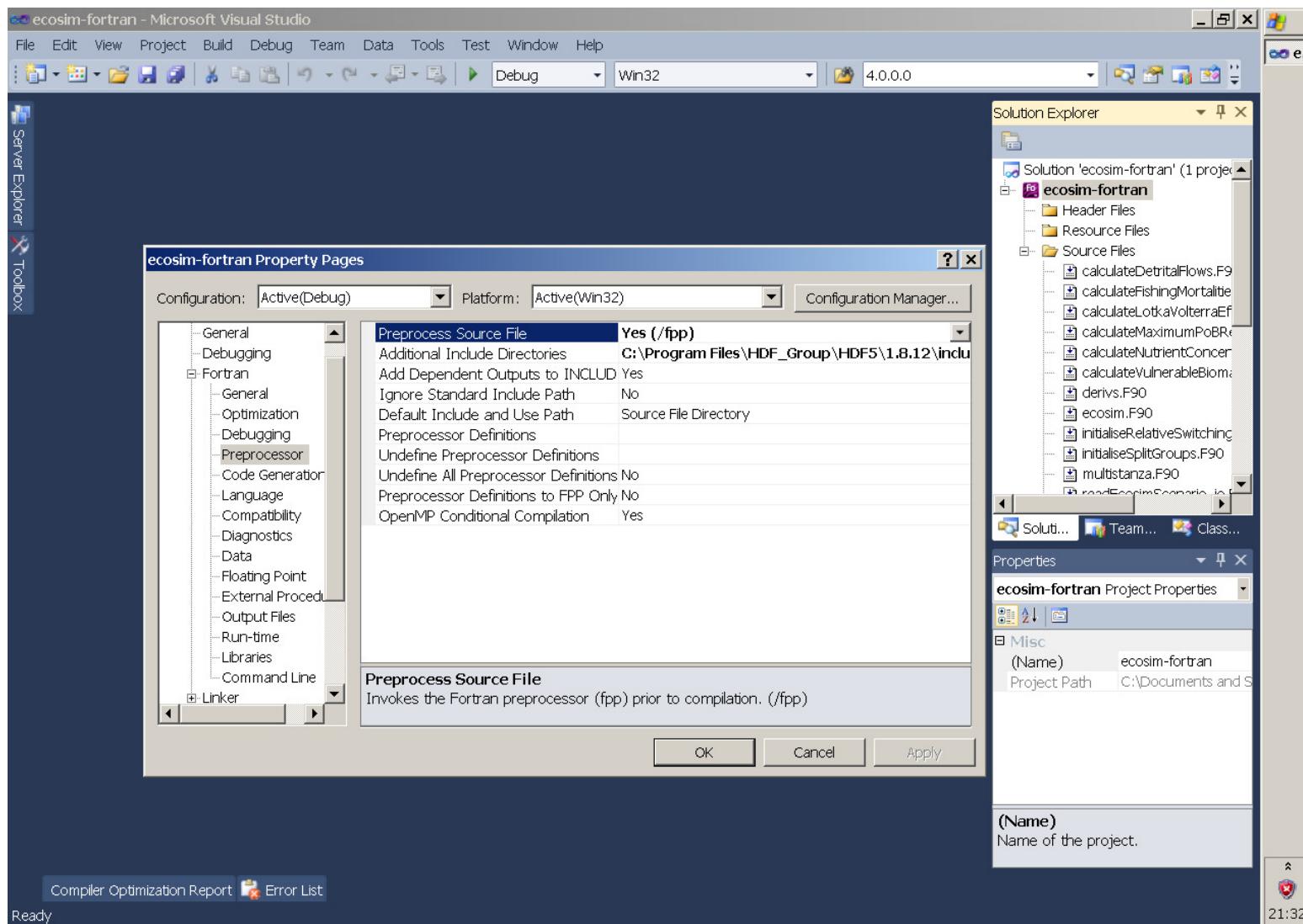


Figure 3.31: Visual Studio: *Ecosim-F*; setting additional include directories.

Figure 3.32: Visual Studio: *Ecosim-F*; enabling preprocessor.

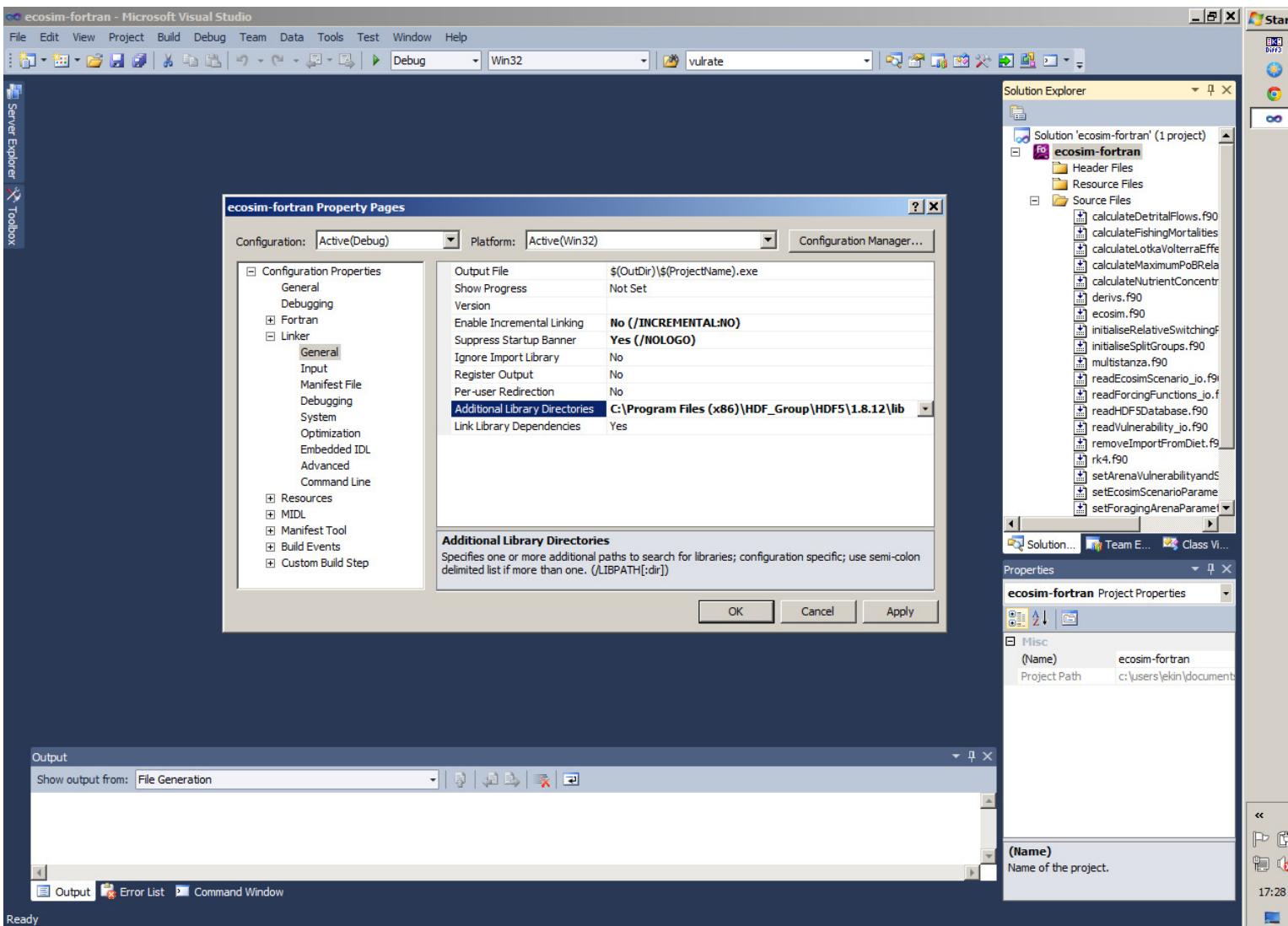
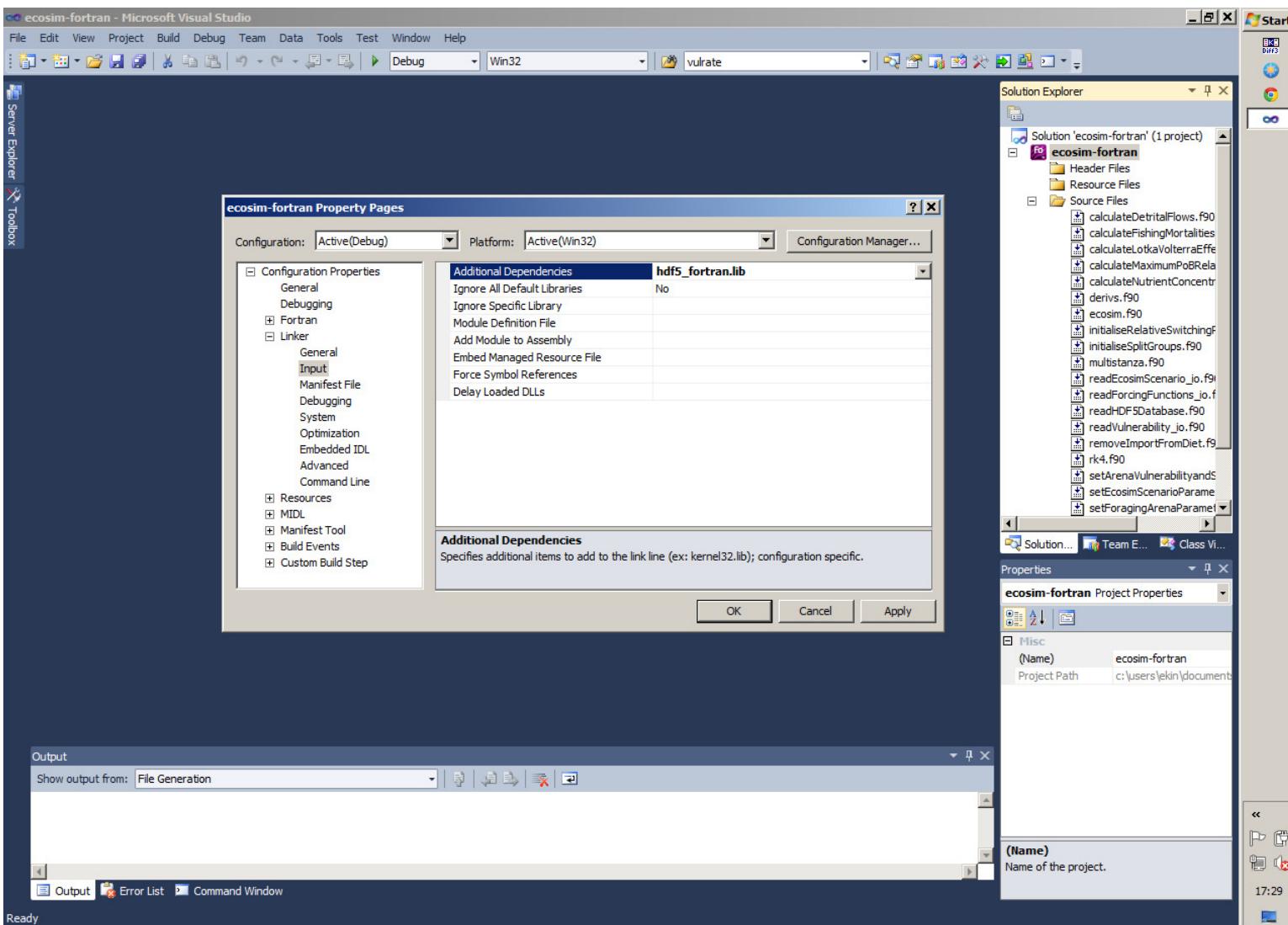


Figure 3.33: Visual Studio: *Ecosim-F*; setting additional library directories.

Figure 3.34: Visual Studio: *Ecosim-F*; setting additional dependencies.

3.4 Compiling and running the *Ecosim-F*

In order to build the *Ecosim-F*, select *Build Solution* from the *Build* menu on the top toolbar. Once completed successfully (Figure 3.35), your *Ecosim-F* model executable must be ready. You may close the MS Visual Studio now.

Your *Ecosim-F* model (*ecosim-fortran.exe*) is under the

```
C:\Users\<User_Name>\Documents\Visual Studio 2010\Projects\  
ecosim-fortran\ecosim-fortran\Debug\
```

directory. Copy this executable file to the directory named *ecosim-f*, the directory into which you have downloaded the *Ecosim-F* sources with GIT. Afterwards, you can run the model first by opening a GIT bash shell in the same directory (Figure 3.36) and then typing as shown in Listing 4.10 (Figure 3.37).

Listing 3.3: Running the model

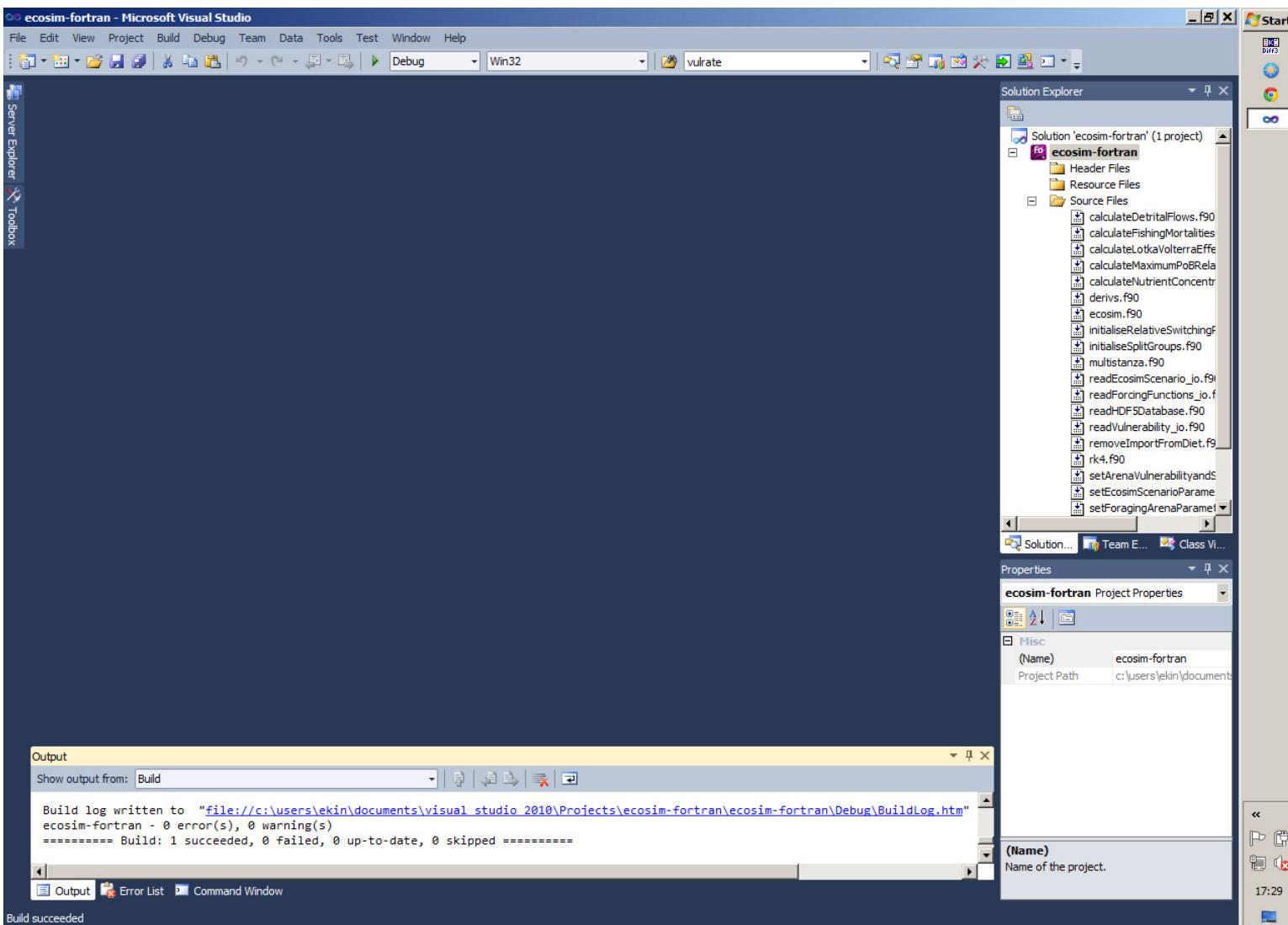
```
./ecosim-fortran.exe
```

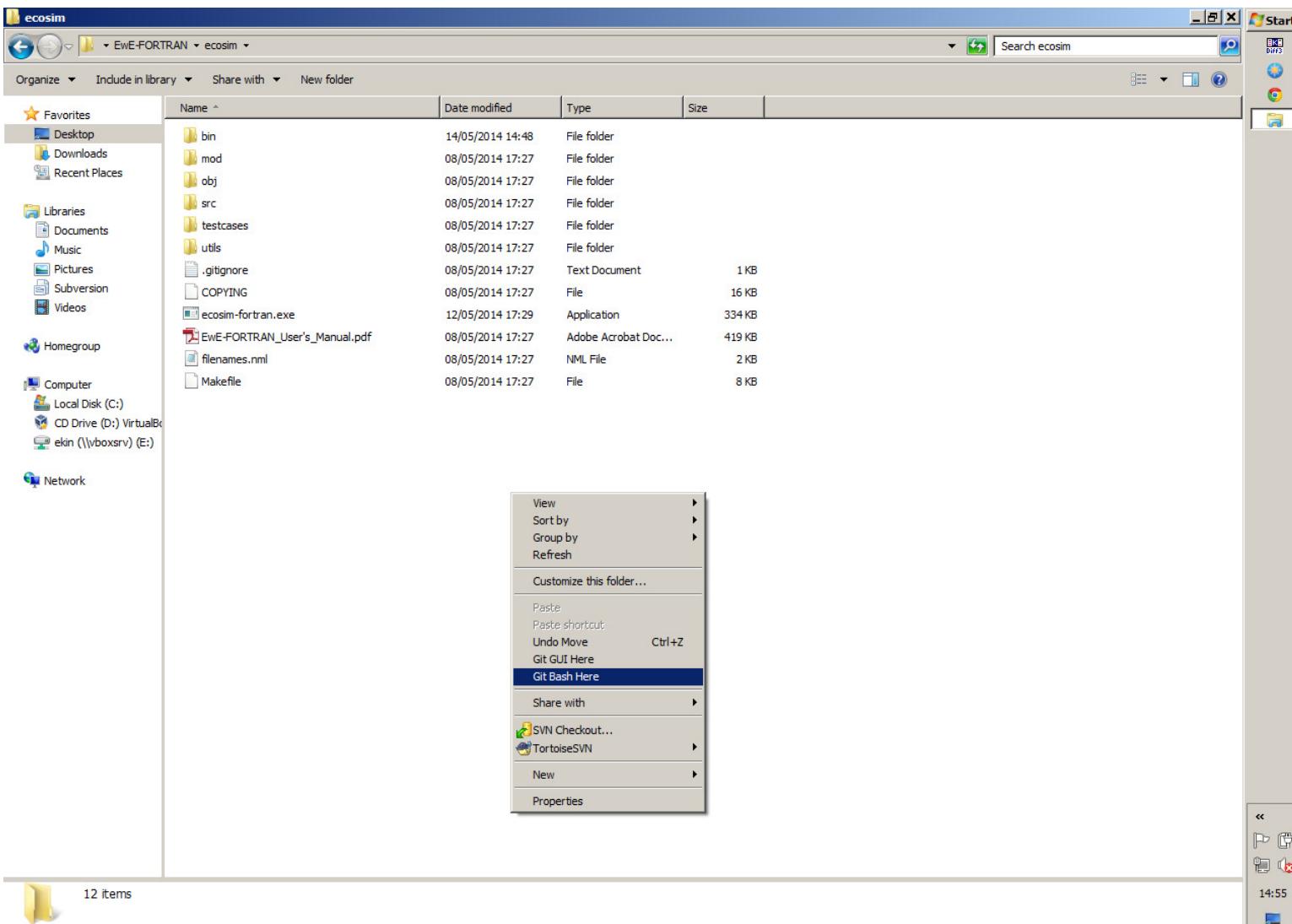
You will see lots of numbers scrolling in the shell command window while the simulation is running. Once it is completed, you will receive the *Simulation ended successfully*. message in the GIT shell (Figure 3.38).

Information

Similarly, you may follow the steps detailed in sections 3.3 and 3.4 to compile, set up and run the Ecopath-F.

For detailed explanations utilising examples you may refer to sections 4.2.2, 4.3.2 and 4.4.3.

Figure 3.35: Visual Studio: *Ecosim-F*; building.

Figure 3.36: *Ecosim-F*: running the model (1).

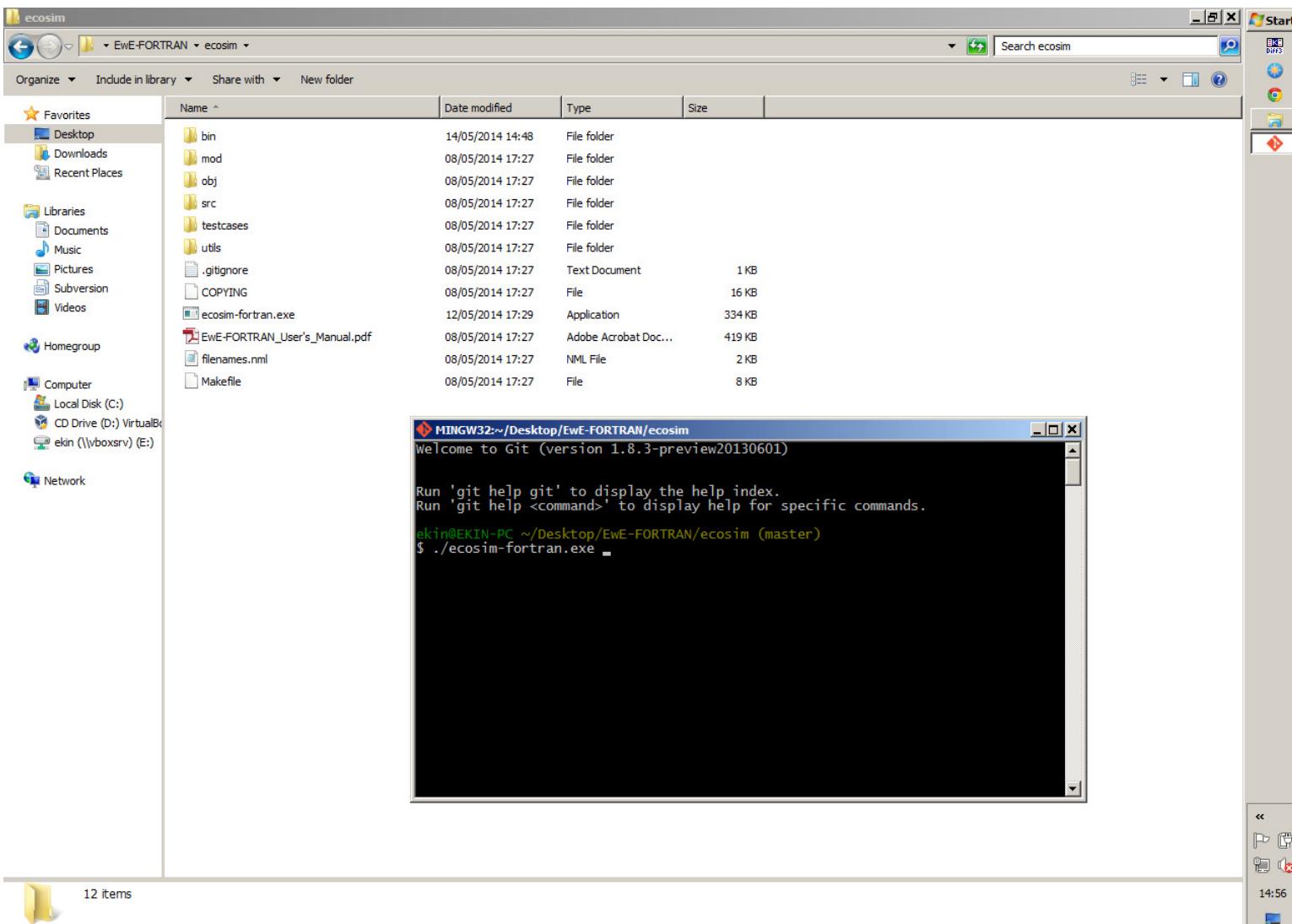
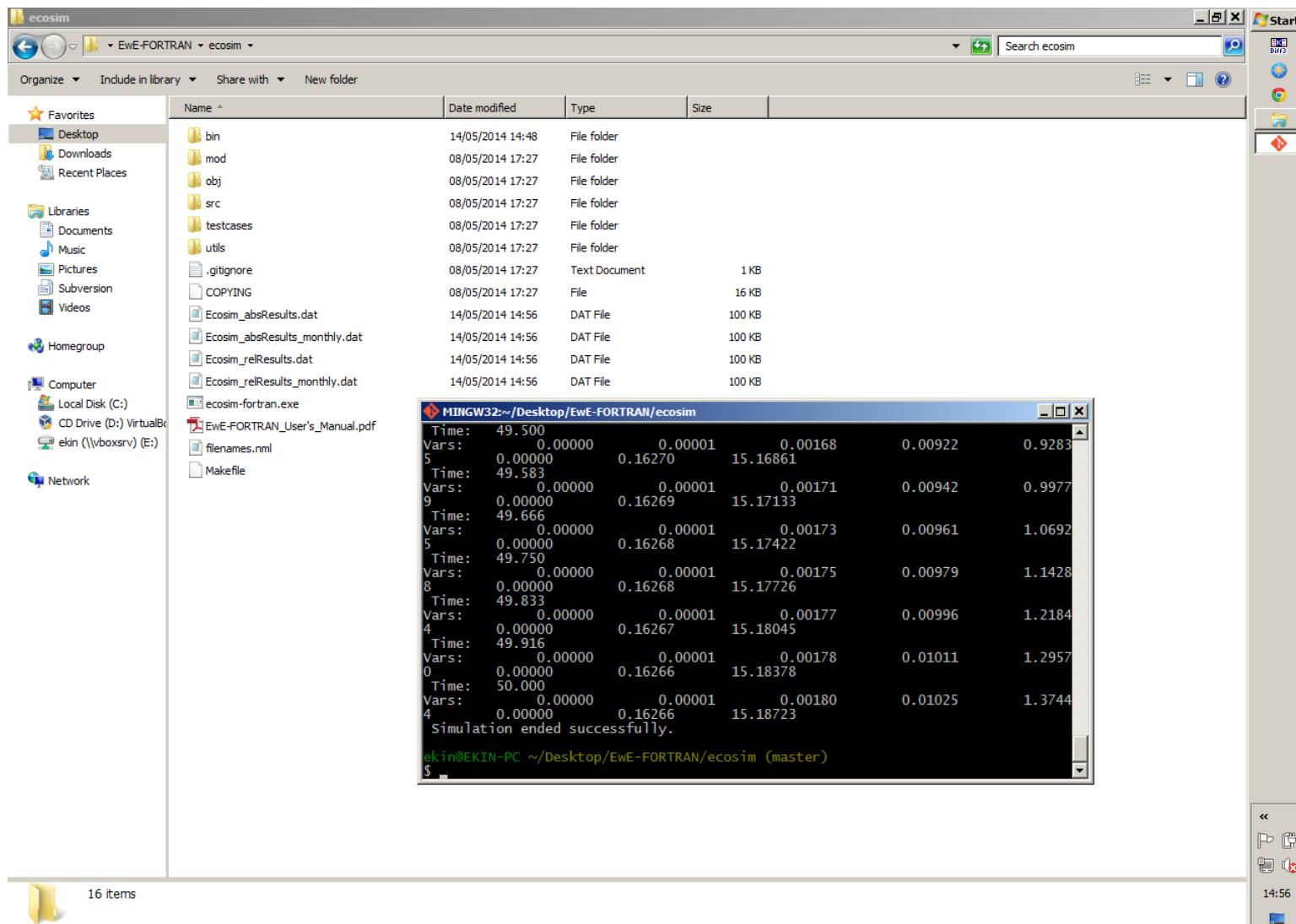


Figure 3.37: *Ecosim-F*: running the model (2).

Figure 3.38: *Ecosim-F*: simulation completed.

Chapter 4

The *EwE-F*: Structural details, custom scenario set-up and preparation of input files

4.1 Folder hierarchy

Both the *Ecopath-F* and the *Ecosim-F* root directories have the same folder and file structure as summarised below:

bin: This is the directory in which the *EwE-F* executable (*ecopath.out* for the *Ecopath-F* and *ecosim.out* for the *Ecosim-F*) will be created once you compile the program¹.

mod: This directory will include the module files necessary for the *EwE-F* once you compile the program¹.

obj: This directory will include the object files created during the compilation of the *EwE-F*¹.

src: This directory contains all the source code files with the extension *.F90*. These files are explained in detail under Section 4.2.1 for the

¹Only on Unix-like systems.

Ecopath-F, Section 4.3.1 for the *Ecosim-F* and Section 4.4.2 for the *Ecospace-F* in the manual.

testcases: This directory contains three sub-directories each for different test cases that you may use to test the *EwE-F*. For the *Ecopath-F*, under each test case directory, there are three (if the test case model does not include multi-stanza groups) or four (if the test case model includes one or more multi-stanza groups) ASCII encoded comma-separated text (CSV; **Comma-Separated Values**) input files containing the necessary input data required to run the model. There is also one additional HDF5 binary input file for the test case, which could, if desired, be used instead of the ASCII files. Please refer to section 4.2.2 on how to set up test cases for an *Ecopath-F* run.

For the *Ecosim-F*, there are at least three ASCII encoded comma-separated text (CSV; **Comma-Separated Values**) input files containing the necessary input data to run the simulation plus an additional HDF5 database file including the *Ecopath-F* results for the test case. Please refer to section 4.3.2 on how to set up test cases for an *Ecosim-F* run and section 4.4.3 on how to set up test cases for an *Ecospace-F* run.

All comma-separated input files are explained in detail under Section 4.2.4 for the *Ecopath-F*, Section 4.3.4 for the *Ecosim-F* and Section 4.4.6 for the *Ecospace-F* in the manual.

Furthermore, under each test case folder within the *ecosim-f* directory, there are two NCL² (NCAR Command Language) script files (*compar_box.ncl* and *compar_box_catch.ncl*) that can be used to draw box-whisker type residual plots of the simulated biomass and catch results of the *EwE-F* simulation for the test case models against the original EwE simulation results for the same test case. You can use these scripts only if NCL is installed on your system. When run as

² The NCAR Command Language (<http://www.ncl.ucar.edu>), a product of the Computational & Information Systems Laboratory at the National Centre for Atmospheric Research (NCAR) and sponsored by the National Science Foundation, is a free interpreted language designed specifically for scientific data processing and visualisation.

```
ncl compar_box.ncl  
or  
ncl compar_box_catch.ncl
```

the NCL scripts produce PDF plots of the residuals from the comparison of the biomasses/catches of the test case respectively in the script's own directory. As default, each NCL script produces a plot utilising the data from two formerly prepared results files; one EwE-F output file (*EwE-F_<TestCaseName>_Biomass.txt* for biomass or *EwE-F_<TestCaseName>_Yield.txt* for catches) and one EwE 6.6 output (*EwE-6.6_<TestCaseName>_Biomass.txt* for biomass or *EwE-6.6_<TestCaseName>_Yield.txt* for catches) file in the directory.

The test case folders within the *ecosim-f* directory also include one prescribed Fortran NML file and one *Makefile* with required parameters already set for their respective model scenarios. These files are named as *<TestCaseName>_filenames.nml* and *<TestCaseName>_Makefile* respectively and can be copied to the model's root directory and renamed as *filenames.nml* for the former and *Makefile* for the latter and then could be used to compile and test the *EwE-F* with one of these test cases.

Finally, in each folder under the test case directory within the *ecosim-f* directory, there is an EwE MS ACCESS database file for the respective test case model, which could be used with EwE 6.6 to reproduce the simulations and produce home-brewed results to compare against the *EwE-F*.

Makefile: This file includes the commands and definitions of environmental variables and preprocessor MACROs that are required to compile the *EwE-F* using the *make* command¹. Before compiling the model, you need to edit this file so as to match the specifications of your system. Once opened with a text editor, the file is quite self-explanatory and provides instructions to the user about how and which sections of the file must be edited. However, for detailed instructions

you may refer to sections 4.2.2 for *Ecopath-F*, 4.3.2 for *Ecosim-F* and 4.4.3 for *Ecospace-F*.

filenames.nml: This is a typical Fortran namelist file that includes the paths and names of the *EwE-F* input files. Once you compile and run the *EwE-F*, this file is read to locate the files that contain data for the model scenario. For details about this file, please refer to sections 4.2.3 for *Ecopath-F*, 4.3.3 for *Ecosim-F* and 4.4.5 for *Ecospace-F*.

ecoXXX-f_structure.txt: (where *XXX* is either *path* or *sim*) This file includes an ASCII diagram of the hierarchical tree view of the *Ecopath-F/Ecosim-F* subroutines in the order they are executed.

4.2 The *Ecopath-F*

4.2.1 Files in the *src* directory

The *Ecopath-F* source code is split into seventeen Fortran source files. The hierarchical structure of the source code can be seen in the "ecopath-f_structure.txt" file in the root directory of *Ecopath-F*. The names and corresponding descriptions of source files are as follows:

- **statevartypesecopath.F90:** Global variables and derived variable types are defined in this module file. All other model routines require this module.
- **ecopath.F90:** This is the main *Ecopath-F* model source file. All other subroutines are called from this file.
- **readEcopathScenarioParameters_io.F90:** This subroutine reads the main Ecopath scenario ASCII input file which comprises the matrix of basic input parameters for the state variables.
- **setEcopathScenarioParameters.F90:** This subroutine allocates the variables utilising data read from the ASCII input file by *readEcopath-ScenarioParameters.F90*.
- **readDietComposition_io.F90:** This subroutine reads the ASCII input file which comprises the diet composition matrix of the state variables.
- **readDetritusFate_io.F90:** This subroutine reads the ASCII input file which comprises the detritus fate matrix of the state variables.
- **readHDF5Database.F90:** If *isASCIIinputFile* is set to *false*. in the *Ecopath-F* namelist file (see Section 4.2.3), bypassing the above mentioned ASCII input files, this subroutine reads an HDF5 database file prepared for the respective Ecopath model scenario.

- **calculateMultistanzaParameters.F90:** This subroutine is called before mass-balance calculations only if there are multi-stanza groups in the model. It calculates the parameters of the multi-stanza groups using their respective growth parameters.
- **calculateSolvingOrderOfLinearEquations.F90:** This subroutine computes the precedence of the state variables from the least-predated to the most-predated and aligns the primary producers and detritus groups last in the order. This precedence determines the order which linear equations are solved.
- **calculatePredation.F90:** This subroutine calculates the predation pressure on each state variable.
- **calculateDetritalFlows.F90:** This subroutine calculates flows to detritus compartments from living groups.
- **calculateDetritusFate.F90:** This subroutine calculates flows between detritus groups only if there is more than one detritus group defined in the model.
- **calculateBAofDetritus.F90:** This subroutine calculates biomass accumulation for detritus groups.
- **calculateEEofDetritus.F90:** This subroutine calculates the ecotrophic efficiencies of detritus groups.
- **calculateEnergyBalance.F90:** After mass-balance calculations are finalised, this subroutine solves the second master equation of Ecopath; the energy balance equation for each state variable.
- **writeHDF5ResultsFile.F90:** This subroutine writes the results of the *Ecopath-F* model run to an HDF5 database file.
- **warnSanityChecks.F90:** Before completing the model run, this subroutine checks the computed results for any inconsistencies (e.g. Eco. Eff. > 1 or Respiration < 0) and prints an error message to warn the user in such case.

4.2.2 Setting up test cases for an *Ecopath-F* run

First, copy the template "filenames.nml" under the test case directory that you would like to run to the *Ecopath-F*'s root directory. For instance, to set up *Ecopath-F* to run Generic_37 test case scenario, open a terminal in *Ecopath-F*'s root directory and issue the command as shown in Listing 4.1 under a UNIX-like system.

Listing 4.1: Copying necessary files

```
cp testcases/Generic_37/Generic_37_filenames.nml ./filenames.nml
```

Then, if necessary, edit the *Makefile* to match your system's Fortran compiler settings. The *Makefile* is self-explanatory, but in a nutshell you may need to edit lines 25, 36, 42 and the region between lines 53-67 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 libraries. The lines starting with a hash (#) are explanatory comment lines. Read them as you find your way through these lines. Once finished, compile the model as shown in Listing 4.4.

Listing 4.2: Compiling the model

```
make
```

Then you may run the model as shown in Listing 4.5.

Listing 4.3: Running the model

```
bin/ecopath -f
```

4.2.3 Setting up your model for an *Ecopath-F* run

In order to be able to run the model with a specific Ecopath scenario, one must edit the **filenames.nml** Fortran namelist file. The explanations of the fields in the namelist file are as follows:

1. *InputData_fname*: Enter the path and name of the file that comprises the basic input parameters for the *Ecopath-F* model scenario (see 4.2.4.1).

2. *DietComp_fname*: Enter the path and name of the file that comprises the diet composition matrix for the *Ecopath-F* model scenario (see 4.2.4.2).
3. *DetFate_fname*: Enter the path and name of the file that comprises the detritus fate matrix for the *Ecopath-F* model scenario (see 4.2.4.3).
4. *GrowthParam_fname*: Enter the path and name of the file that comprises growth parameters of multi-stanza groups for the *Ecopath-F* model scenario. If there is no multi-stanza group in your model scenario, leave this field empty (see 4.2.4.4).
5. *Results_fname*: Enter the path and name of the HDF5 database file that you want the outputs of the *Ecopath-F* run to be written to.
6. *HDF5_fname*: If using binary input for the *Ecopath-F* (see the following item), enter the path and name of the HDF5 database file from which the *Ecopath-F* input parameters are going to be read. Otherwise, leave this field empty. This feature is highly **EXPERIMENTAL**.
7. *isASCIIinputFile*: If using ASCII input files for the data input to the *Ecopath-F*, set this Boolean value to *.true.* (default). If you wish to use a binary HDF5 Ecopath input file for better floating-point precision, set this value to *.false..*

 **Important!**

*All the field values of the fields specified above (except for field *isASCIIinputFile*) must be enclosed in double quotes.*

Edit the *Makefile* to match your system's Fortran compiler settings. The *Makefile* is self-explanatory, but in a nutshell you may need to edit lines 25, 36, 42 and the region between lines 53-67 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 libraries. The lines starting with a hash (#) are explanatory comment lines.

Read them as you find your way through these lines. Once finished, compile the model as shown in Listing 4.4.

Listing 4.4: Compiling the model

```
make
```

Then you may run the model as shown in Listing 4.5.

Listing 4.5: Running the model

```
bin/ecopath -f
```

4.2.4 Description of the input files

All *EwE-F* input files are comma-separated (CSV) ASCII input files. These files could easily be prepared using common spreadsheet software such as LibreOffice Calc or Microsoft® Excel and then saved as CSV using "Save as..." under *File* menu.

4.2.4.1 The main parameters input file:

<YourModelName>_Scenario.csv



Important!

In your main parameters input file, group your state variables, i.e. the order of rows, so that detritus groups come last, primary producers are located above them and the consumer groups are atop of the file. In the remaining input files, follow the same row order unless stated otherwise.

This file is a comma-separated ASCII file. The first three rows and their respective first two columns are reserved for; i) the specification of the number of state variables, ii) number of parameters (currently 15) related to each state variable of the input data matrix, and iii) the number of multi-stanza groups (number of split groups) (Figure 4.1). If there is no multi-stanza group in the model, then a value of 0 must be entered. The fourth row and its respective first column in the file are reserved for the specification

of the names of the state variables. On the fifth row and its respective second column, the data matrix starts. The descriptions of the columns are as follows:

Number of groups	52														
Number of parameters	15														
Number of split groups	7														
Groups	B	P/B	Q/B	EE	P/Q	Unass. Q	Detritus Import	Landings	Discards	Biomass Accumulation	Organism Type	Split Group?	Stanza No	Age Start	Leading stanza?
0-12 Snook	-999	5	-999	-999	-999	0.2	-999	0	0	0	2	1	1	0	0
3-12 Snook	-999	2	-999	-999	-999	0.2	-999	0	0	0	2	1	1	3	0
12-48 Snook	-999	0.9	-999	-999	-999	0.2	-999	0	0.02	0	2	1	1	12	0
48-90 Snook	-999	0.62	-999	-999	-999	0.2	-999	0.035	0	0	2	1	1	48	0
90+ Snook	0.0202	0.6	1.289	-999	-999	0.2	-999	0	0	0	2	1	1	90	1
0-3 Red Drum	-999	8	-999	-999	-999	0.2	-999	0	0	0	2	1	2	0	0
3-8 Red Drum	-999	3.5	-999	-999	-999	0.2	-999	0	0	0	2	1	2	3	0
8-18 Red Drum	-999	1.1	-999	-999	-999	0.2	-999	0	0	0	2	1	2	8	0
18-36 Red Drum	-999	0.6	-999	-999	-999	0.2	-999	0.005	0	0	2	1	2	18	0
36+ Red Drum	0.3004	0.55	0.979	-999	-999	0.2	-999	0.0005	0	0	2	1	2	36	1
0-3 Sea Trout	-999	6	-999	-999	-999	0.2	-999	0	0	0	2	1	3	0	0
3-18 Sea Trout	-999	1.4	-999	-999	-999	0.2	-999	0	0	0	2	1	3	3	0
18+ Sea Trout	0.2201	0.7	1.6	-999	-999	0.2	-999	0.05	0	0	2	1	3	18	1
0-3 Sand Trout	-999	5	-999	-999	-999	0.2	-999	0	0	0	2	1	4	0	0
3-12 Sand Trout	-999	1.2	-999	-999	-999	0.2	-999	0	0	0	2	1	4	3	0
12+ Sand Trout	0.1	0.7	2.7	-999	-999	0.2	-999	0.0165	0	0	2	1	4	12	1
0-6 Mullet	-999	6.7	-999	-999	-999	0.2	-999	0	0	0	2	1	5	0	0
6-18 Mullet	-999	1.8	-999	-999	-999	0.2	-999	0.25	0	0	2	1	5	6	0
18+ Mullet	2.8006	0.8	7.999	-999	-999	0.2	-999	0.5	0	0	2	1	5	18	1
Mackrel 0-3	-999	4	-999	-999	-999	0.2	-999	0	0	0	2	1	6	0	0
Mackrel 3+	0.0183	0.5	5.99	-999	-999	0.2	-999	0.005	0	0	2	1	6	3	1
Ladyfish 0-10	-999	2.8	-999	-999	-999	0.2	-999	0	0	0	2	1	7	0	0
Ladyfish 10+	0.089	1.6	6	-999	-999	0.2	-999	0.01	0	0	2	1	7	10	1
Jacks	-999	0.6	-999	0.6	0.3	0.2	-999	0	0	0	2	0	-999	-999	-999
Bay Anchovy	-999	2.53	14	0.6	-999	0.2	-999	0.02	0	0	2	0	-999	-999	-999
Pin Fish	0.32	1.019	8	-999	-999	0.2	-999	0.001	0	0	2	0	-999	-999	-999
Spot	0.69	1.1	12	-999	-999	0.2	-999	0	0	0	2	0	-999	-999	-999
Silver Perch	0.09	1.4	9	-999	-999	0.2	-999	0.001	0	0	2	0	-999	-999	-999

Figure 4.1: Sample ASCII file containing the main input parameters matrix for Tampa Bay EwE model.

Groups: This column includes the names of the functional groups in the model. It is used to entitle respective rows of Ecopath-F's output file for the sake of user friendliness.

B (Biomass): This column contains the biomass values of the state variables. If biomass is to be estimated, the value **-999** must be entered; otherwise, the amount of biomass is entered.

P/B (Production/Biomass): This column contains the production to biomass ratios of the state variables. If **-999** is entered, P/B value is estimated by the model for living groups. For detritus groups, **-999** must be entered.

Q/B (Consumption/Biomass): This column contains the consumption to biomass ratios of the state variables. If **-999** is entered, Q/B value is estimated for consumer groups by the model. For producer and detritus groups, **-999** must be entered.

Important!

For multi-stanza groups, only the Q/B value of the leading stanza must be entered. The Q/B value of other stanzas must be entered as -999 so that they are calculated once the Ecopath-F is run.

EE (Ecotrophic Efficiency): This column contains the ecotrophic efficiency values of the state variables. If **-999** is entered, EE value is estimated by the model. For detritus groups, an EE value of **-999** must be entered.

P/Q (Production/Consumption): This column contains the production to consumption ratios of the state variables. If **-999** is entered, P/Q value is estimated by the model. For producer and detritus groups, a value of **-999** must be entered.

Unass./Q (Unassimilated food/Consumption): This is the ratio of unassimilated food to the consumption of the state variables. A value between **0** and **1** must be entered. For producer and detritus groups, a value of **-999** must be entered.

Det. Import (Detritus Import): This is the detritus import value for detritus groups. This value is entered only for the detritus groups. For other state variables, a value of -999 must be entered.

Landings: This is the landed part of the fisheries catch. Enter 0 if the group is not fished.

Discards: This is the discarded part of the fisheries catch. Enter 0 if the group is not fished or not discarded.

Biomass Accumulation: Biomass accumulation for state variables are specified. Enter 0 if there is no biomass accumulation for group.

Org. Type (Organism Type): This column specifies the organism type; enter 1 for producers, 2 for consumers and 0 for detritus groups.

Multi-stanza: This column specifies if the group is a multi-stanza group or not. 1 is entered for multi-stanza groups and 0 is entered otherwise.

Important!

If there is no multi-stanza group in the model, the last three columns in the input parameters file; i.e. the ones listed below and named as Stanza No, Age Start, and Leading Stanza, must be entered as -999.

Important!

*Be sure to organise your multi-stanza groups so that in each group the stan-
zas are ordered starting from the youngest stanza atop to the leading stanza
at the bottom.*

Stanza No: This is the identifier of the each unique multi-stanza group. All members of the same multi-stanza group must have the same identifier number. It starts from 1 and may go to infinity.

Age Start: This is the starting age in months of each stanza in the multi-stanza group.

Leading Stanza: This column specifies if the stanza in the multi-stanza group is the leading stanza or not. *1* is entered for the leading stanza and *0* is entered otherwise.

4.2.4.2 The diet composition input file:

<YourModelName>_DC.csv

Similar to the input file comprising the main input parameters matrix, this file is also a comma-separated ASCII file. The first two rows and their respective first two columns are reserved for; i) the specification of the number of state variables plus 1 row for diet import, ii) the number of consumer groups in the diet data matrix as shown in Figure 4.2. The third row and its respective first column in the file are reserved for the specification of the names of the state variables. On the fourth row and its respective second column, the data matrix starts. The prey groups are located row-wise and the predators are located column-wise. The last row is reserved for the imports in the diet composition matrix. If there is no predation between a prey and predator pair, a value of *0* must be entered. Otherwise, the ration of the prey in the predator's diet is entered. Just like in EwE, each column must sum up to unity.

CHAPTER 4. THE EWE-F: DETAILS & CUSTOMISATION

67

Number of groups incl. import	53																		
Number of predators	48																		
Prey/Pred	0-12 Snook	3-12 Snook	12-48 Snook	48-90 Snook	90+ Snook	0-3 Red Drum	3-8 Red Drum	8-18 Red Drum	18-36 Red Drum	36+ Red Drum	0-3 Sea Trout	3-18 Sea Trout	18+ Sea Trout	0-3 Sand Trout					
0-12 Snook	0	0.001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3-12 Snook	0	0	0	0.0020002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12-48 Snook	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48-90 Snook	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90+ Snook	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-3 Red Drum	0	0	9.9989E-06	0.00010001	0	0	0	0	0.0010001	0	0	0	0	0.0001	0	0	0	0	0
3-8 Red Drum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8-18 Red Drum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18-36 Red Drum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36+ Red Drum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-3 Sea Trout	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3-18 Sea Trout	0	0	0	0.02480248	0.00184466	0	0	0	0	0	0	0	0	0	0.003	0	0	0	0
18+ Sea Trout	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-3 Sand Trout	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3-12 Sand Trout	0	0	0	0	0.00834951	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12+ Sand Trout	0	0	0	0	0.00368932	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0-6 Mullet	0	0.005	0.0009999	0.0020002	0.04621358	0	0	0.006	0.0060006	0.0033	0	0	0.120012	0	0	0	0	0	0
6-18 Mullet	0	0	0	0	0.00834951	0	0	0.013	0.0110011	0.045795	0	0	0.001	0	0	0	0	0	0
18+ Mullet	0	0	0	0	0.03796116	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mackrel 0-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mackrel 3+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ladyfish 0-10	0	0	0.0009999	0	0.02912621	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ladyfish 10+	0	0	0	0	0.00368932	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jacks	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bay Anchovy	0	0	0.0009999	0.09910991	0.19252424	0	0	0.019	0.0569057	0.040296	0	0.0498	0.156216	0	0	0	0	0	0
Pin Fish	0	0.005	0.05009499	0.2280228	0.01941747	0	0	0.006	0.0280028	0.023198	0	0	0.10111	0	0	0	0	0	0
Spot	0	0	0.019998	0.03070307	0.00368932	0	0	0.032	0.0280028	0.015998	0	0	0.008001	0	0	0	0	0	0
Silver Perch	0	0	0.0049995	0.03070307	0.04436892	0	0	0.1289	0.0060006	0.005499	0	0.0498	0.030003	0	0	0	0	0	0

Figure 4.2: Sample ASCII file containing the diet composition matrix for Tampa Bay EwE model.

4.2.4.3 The detritus fate input file:

<YourModelName>_DetFate.csv

This input file contains the detritus fate for each group. In other words, how much of the flows originating from each group ends up in each detritus compartment. Similar to the input file containing the main input parameters matrix, this file is also a comma-separated ASCII file. As shown in Figure 4.3, the first three rows and their respective first two columns are reserved for; i) the specification of the number of state variables, ii) the number of detrital groups plus 1 for detritus export, and finally iii) the number of the detritus groups. The fourth row and its respective first column in the file are reserved for the specification of the names of the state variables. On the fifth row and its respective second column, the data matrix starts. If there is only one detritus group, this file contains two columns; one for the fraction of flows going to the detritus compartment and one for the amount of flow that is exported instead of ending up in detritus. If there is more than one detritus group, the columns must be replicated so that the number of columns equals to the sum of the number of detritus groups plus one column for the exported fraction.

Number of groups	52	
Number of detrital groups incl. export	2	
Number of detrital groups	1	
Groups	Detritus	1 Export
0-12 Snook	1	0
3-12 Snook	1	0
12-48 Snook	1	0
48-90 Snook	1	0
90+ Snook	1	0
0-3 Red Drum	1	0
3-8 Red Drum	1	0
8-18 Red Drum	1	0
18-36 Red Drum	1	0
36+ Red Drum	1	0
0-3 Sea Trout	1	0
3-18 Sea Trout	1	0

Figure 4.3: Sample ASCII file containing the detritus fate matrix Tampa Bay EwE model.

4.2.4.4 The growth parameters input file:

<YourModelName>_GrowthParam.csv

This input file is only needed if there are multi-stanza groups in the model. The order of the rows must represent the order of the multi-stanza groups, i.e. multi-stanza group 1 should be listed first and multi-stanza group 2 should be listed second, and so on. Similar to the input file containing the main input parameters matrix, this file is also a comma-separated ASCII file. The first two rows and their respective first two columns are reserved for; i) the specification of the number of split groups, and ii) the number of the growth parameters in the growth parameter data matrix as shown in Figure 4.4. The third row is reserved for the indication of the names of the parameters in the file and its respective first column in the file is reserved for the specification of the names of the state variables. On the fourth row and its respective second column, the data matrix starts. The descriptions of the columns are as follows:

K (von Bertalanffy Growth Coefficient): This is the VBGF (von

Bertalanffy Growth Function) K value of the multi-stanza group.

Recruit. Power (Recruitment Power): This is the recruitment power of the multi-stanza group.

BA/B (Relative Biomass Accumulation Rate): This is the ratio of BA (Biomass Accumulation) to B (Biomass) of the multi-stanza group.

Wmat/Winf (Weight at maturity age / Weight at infinite age): This is the ratio of weight at maturity age to weight at infinite age.

Number of split groups	7			
Number of growth parameters	4			
Groups	K	Rec. Pow	BA/B	Wmaturity/Winf
Snook	0.35	1	0	0.25
Red Drum	0.323	1	0	0.2
Sea Trout	0.32	1	0	0.25
Sand Trout	0.3	1	0	0.35
Mullet	0.35	1	0	0.35
Mackerel	0.35	1	0	0.45
Ladyfish	0.28	0.7	0	0.4

Figure 4.4: Sample ASCII file containing the growth parameters of multi-stanza groups for Tampa Bay EwE model.

4.2.5 Description of the *Ecopath-F* output (results) files

Once the model finishes running, one comma-separated (CSV) ASCII file named **Ecopath_Results.dat** and an HDF5 database file are created under the directory where the model is run and in the user-specified location respectively. The HDF5 file comprises the *Ecopath-F* run results as well as the input parameters and is required by the *Ecosim-F* to run an Ecosim simulation. The ASCII file is a short summary of the *Ecopath-F* estimated parameters and the input parameters. Every row in the ASCII file corresponds to a state variable in compliance with the order in the main scenario input file and columns are named in the first row of the file. You can open the file with a spreadsheet program like LibreOffice Calc or Microsoft® Excel.

4.3 The *Ecosim-F*

4.3.1 Files in the *src* directory

The *Ecosim-F* source code is split into thirty-three Fortran source files, six of them are related to *Ecospace-F* and are explained in Section 4.4.2. The hierarchical structure of the source code can be seen in the "ecosim-f_structure.txt" file in the root directory of *Ecosim-F*. The names and corresponding descriptions of these source files are as follows:

- **statevartypesecopath.F90:** Global variables and derived variable types for the *Ecopath-F* are defined in this module file. Many other model routines require this module.
- **statevartypesecosim.F90:** Global variables and derived variable types for the *Ecosim-F* are defined in this module file. All other model routines use this module.
- **ecosim.F90:** This is the main *Ecosim-F* model source file. Unless explicitly specified otherwise, all other subroutines are called from this file.
- **readEcosimScenario_io.F90:** This subroutine reads the basic scenario parameters of the *Ecosim-F* model scenario from its respective ASCII file.
- **setEcosimScenarioParameters.F90:** This subroutine allocates the scenario variables utilising data read from the ASCII input file by *read-EcosimScenario_io.F90*.
- **readHDF5Database.F90:** This module includes a subroutine to read the *Ecopath-F* outputs from its corresponding HDF5 database file.
- **readVulnerability_io.F90:** This subroutine reads the vulnerability data matrix of the *Ecosim-F* model scenario from its respective ASCII file.

- **setForagingArenaParameters.F90:** This subroutine allocates the foraging arena parameters of the *Ecosim-F* model in the memory.
- **readForcingFunctions_io.F90:** This subroutine reads the time series of forcing functions (fishing mortality or effort) of the *Ecosim-F* model scenario from its respective ASCII file.
- **readPrimaryProdForcingFunction_io.F90:** This subroutine reads the time series of primary production forcings for the primary producer groups of the *Ecosim-F* model scenario from its respective ASCII file.
- **readNutrientForcingFunction_io.F90:** This subroutine reads the time series of nutrient forcings of the *Ecosim-F* model scenario from its respective ASCII file.
- **calculateMaximumPoBRelatedValues.F90:** This subroutine calculates the maximum P/B ratios of the state variables.
- **calculateNutrientConcentrations.F90:** This subroutine calculates the initial nutrient concentration in the system.
- **removeImportFromDiet.F90:** This subroutine accounts for and removes the import fraction in the diet compositions of the state variables.
- **initialiseSplitGroups.F90:** This subroutine calculates the initial conditions for the multi-stanza groups before running the *Ecosim-F* simulation only if there are multi-stanza groups in the model scenario.
- **setSplitPred.F90:** This function calculates the abundances for each multi-stanza group and its corresponding stanzas only if there are multi-stanza groups in the model scenario.
- **calculateLotkaVolterraEffectiveSearchRates.F90:** This subroutine calculates the initial values of prey search rates for consumer groups.

- **initialiseRelativeSwitchingParameters.F90:** This subroutine calculates the initial prey switching parameters for consumer groups.
- **setArenaVulnerabilityandSearchRates.F90:** This subroutine calculates the vulnerability rates and search rates between predator-prey pairs in each foraging arena.
- **setRelativeSwitchingParameters.F90:** This subroutine sets the prey switching parameters for consumer groups at each time step. This subroutine is called by the *derivs.F90* subroutine.
- **calculateVulnerableBiomasses.F90:** This subroutine calculates vulnerable biomass values of preys in a given arena at each time step. This subroutine is called by the *derivs.F90* subroutine.
- **updateForagingTimes.F90:** This subroutine updates the feeding times of consumer groups.
- **multistanza.F90:** This subroutine carries out the multi-stanza calculations only if there are multi-stanza groups in the model.
- **derivs.F90:** This is the derivative function that calculates the sources and sinks for the state variables in the model and called by the Runge-Kutta 4th order integrator routine *rk4.F90*.
- **calculateFishingMortalities.F90:** This subroutine calculates the fishing mortality values of harvested state variables based on the density-dependent catchability values and times series of fishing mortalities.
- **rk4.F90:** This is the Runge-Kutta 4th order integrator subroutine.
- **calculateDetritalFlows.F90:** This subroutine carries out the detrital flow calculations. This subroutine is called by the *derivs.F90* subroutine.

4.3.2 Setting up test cases for an *Ecosim-F* run

First, copy the template "filenames.nml" and "Makefile" under the test case directory that you would like to run to the *Ecosim-F*'s root directory. For instance, to set up *Ecosim-F* to run Generic_37 test case scenario, open a terminal in *Ecosim-F*'s root directory and issue the command as shown in Listing 4.6 under a UNIX-like system.

Listing 4.6: Running the model

```
cp testcases/Generic_37/Generic_37_filenames.nml ./filenames.nml
cp testcases/Generic_37/Generic_37_Makefile ./Makefile
```

Then, if necessary, edit the *Makefile* to match your system's Fortran compiler settings. The *Makefile* is self-explanatory, but in a nutshell you may need to edit lines 25, 42, 48 and the region between lines 59-73 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 libraries. The lines starting with a hash (#) are explanatory comment lines. Read them as you find your way through these lines. Once finished, compile the model as shown in Listing 4.9.

Listing 4.7: Compiling the model

```
make
```

Then you may run the model as shown in Listing 4.10.

Listing 4.8: Running the model

```
bin/ecosim-f
```



Important!

*When switching test cases or models, it is **strongly** advised to do a "make clean" for Ecosim-F and a "ecospace=YES make clean" for Ecosim-F with Ecospace-F in order to clean up the directories of EWE-F. For more details, please refer to 2.3.*

4.3.3 Setting up your model for an *Ecosim-F* run

In order to be able to run the model with a specific Ecosim scenario, one must edit the **filenames.nml** Fortran namelist file. The explanations of the fields in the namelist file are as follows:

1. *HDF5_fname*: Enter the path and name of the HDF5 database file that comprises the results from the *Ecopath-F* run of the *Ecosim-F* scenario.
2. *GroupInfo_fname*: Enter the path and name of the file that comprises the group parameters for the *Ecosim-F* model scenario (see 4.3.4.1).
3. *Vulnerability_fname*: Enter the path and name of the file that comprises the vulnerability data matrix for the *Ecosim-F* model scenario (see 4.3.4.2).
4. *Forcing_fname*: Enter the path and name of the file that includes the times series of forcing functions (fishing mortality or effort) of groups for the *Ecosim-F* model scenario (see 4.3.4.3).
5. *NutrientForcing_fname*: Enter the path and name of the file that includes the times series of nutrient forcings for the *Ecosim-F* model scenario. If no nutrient forcing is applied in your model scenario, leave this field empty (see 4.3.4.4).
6. *PrimaryProdForcing_fname*: Enter the path and name of the file that includes the times series of primary production forcings of the primary producer groups for the *Ecosim-F* model scenario. If no primary production forcing is applied in your model scenario, leave this field empty (see 4.3.4.5).



Important!

All the field values of the fields specified above (fields from 1 to 6) must be enclosed in double quotes.

7. *tf*: Enter the number of years to run the model.
8. *StepsPerMonth*: Enter the number of time steps per month (default = 1).
9. *NutBaseFreeProp*: Enter the base proportion of free nutrients (default = 0.9999).
10. *NutPBmax*: Enter the maximum P/B rate due to the nutrient concentration (default = 1.5).
11. *relax*: Enter the relaxation parameter for the *rk4.F90* integrator (default = 0).

You may need to edit the **Makefile** before you run *make* in order to enable your time series by defining respective macros. For primary production forcing, uncomment line 34 in **Makefile** as shown below:

```
# Uncomment below if there is primary production forcing
CPPDEFS += -D_ForcePrimaryProd_
```

And for nutrient forcing, uncomment line 31 in **Makefile** as shown below:

```
# Uncomment below if there is nutrient forcing
CPPDEFS += -D_ForceNutrient_
```

Further, edit the *Makefile* to match your system's Fortran compiler settings. The *Makefile* is self-explanatory, but in a nutshell you may need to edit lines 25, 42, 48 and the region between lines 59-73 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 libraries. The lines starting with a hash (#) are explanatory comment lines. Read them as you find your way through these lines. Once finished, compile the model as shown in Listing 4.9.

Listing 4.9: Compiling the model

```
make
```

Then you may run the model as shown in Listing 4.10.

Listing 4.10: Running the model

```
bin/ecosim-f
```

 **Important!**

*When switching test cases or models, it is **strongly** advised to do a "make clean" for Ecosim-F and a "ecospace=YES make clean" for Ecosim-F with Ecospace-F in order to clean up the directories of EwE-F. For more details, please refer to 2.3.*

4.3.4 Description of the input files

All *EwE-F* input files are comma-separated (CSV) ASCII input files. These files could easily be prepared using common spreadsheet software such as LibreOffice Calc or Microsoft® Excel and then saved as CSV using "Save as..." under *File* menu.

4.3.4.1 The group parameters input file:

<YourModelName>_Scenario.csv

 **Important!**

In your group parameters input file, group your state variables, i.e. the order of rows, so that detritus groups come last, primary producers are located above them and the consumer groups are atop of the file (EXACTLY THE SAME ORDER you used in the Ecopath-F main parameters input file). In the remaining input files, follow the same row order you used in this file unless stated otherwise.

This file is a comma-separated ASCII file. The first three rows and their respective first two columns are reserved for the specification of the size (#rows (equal to the number of state variables) * #columns (9)) of the input data matrix and the number of multi-stanza groups (#stanzas) (Figure 4.5). If there is no multi-stanza group in the model, then a value of 0 must be entered. The fourth row and its respective first column in the file are reserved for the specification of the names of the state variables. On the fifth row and its respective second column, the data matrix starts. The descriptions of the columns are as follows:

Number of groups	52										
Number of parameters	9										
Number of split groups	7										
Groups	Max rel. P/B	Feeding time	Max rel. feeding time	Feeding time adjust rate [0,1]	Fraction of other mortality sens. to changes in feeding time	Predator effect on feeding time [0,1]	Density-dep. catchability: Qmax/Qo [>=1]	QBmax/ QBo (for handling time) [>1]	Switching power parameter [0,2]	Advection?	
0-12 Snook	2	1	2	0.5		1	0	1	1000	0	0
3-12 Snook	2	1	2	0		1	0	1	1000	0	0
12-48 Snook	2	1	2	0		1	0	1	1000	0	0
48-90 Snook	2	1	2	0		1	0	1	1000	0	0
90+ Snook	2	1	2	0		1	0	1	1000	0	0
0-3 Red Drum	2	1	2	0.5		1	0	1	1000	0	0
3-8 Red Drum	2	1	2	0		1	0	1	1000	0	0
8-18 Red Drum	2	1	2	0		1	0	1	1000	0	0
18-36 Red Drum	2	1	2	0		1	0	1	1000	0	0
36+ Red Drum	2	1	2	0		1	0	1	1000	0	0
0-3 Sea Trout	2	1	2	0.5		1	0	1	1000	0	0
3-18 Sea Trout	2	1	2	0		1	0	1	1000	0	0
18+ Sea Trout	2	1	2	0		1	0	1	1000	0	0
0-3 Sand Trout	2	1	2	0.5		1	0	1	1000	0	0
3-12 Sand Trout	2	1	2	0		1	0	1	1000	0	0
12+ Sand Trout	2	1	2	0		1	0	1	1000	0	0
0-6 Mullet	2	1	2	0.5		1	0	1	1000	0	0
6-18 Mullet	2	1	2	0		1	0	1	1000	0	0
18+ Mullet	2	1	2	0		1	0	1	1000	0	0
Mackrel 0-3	2	1	2	0.5		1	0	1	1000	0	0
Mackrel 3+	2	1	2	0		1	0	1	1000	0	0
Ladyfish 0-10	2	1	2	0.5	0.1	0	1	1000	0	0	0
Ladyfish 10+	2	1	2	0.5		1	0	1	1000	0	0
Jacks	2	1	2	0		1	0	1	1000	0	0
Bay Anchovy	2	1	2	0		1	0	1	1000	0	0
Pin Fish	2	1	2	0		1	0	1	1000	0	0
Spot	2	1	2	0		1	0	1	1000	0	0
Silver Perch	2	1	2	0		1	0	1	1000	0	0

Figure 4.5: Sample ASCII file containing the group input parameters matrix for Tampa Bay EwE model.

Groups: This column includes the names of the functional groups in the model. It is used to entitle respective rows of Ecosim-F's output files for the sake of user friendliness.

Maximum Relative P/B: This column contains the maximum relative P/B (Production/Biomass) ratio of the state variables. For consumer groups, the default value of *2* is entered. For producers, enter the value specific to your model. For detritus groups, this parameter must be set to *-999*.

Feeding Time: This column contains the feeding time parameters for the state variables. Regardless of being consumer, producer or detritus, the default value of *1* must be entered.

Maximum Relative Feeding Time: This column contains the maximum relative feeding times of consumer groups. For producer and detritus groups, this value must be set to *-999*.

Feeding Time Adjustment Rate: This column contains the adjustment rate of the changes in the feeding time of consumer groups. For producer and detritus groups, this value must be set to *-999*.

Fraction of Other Mortality Sensitive to Changes in Feeding Time: This column contains the parameter values only for consumer groups and indicates the fraction of predation mortality caused by means of changes in the feeding time of the consumer groups. This value must be set to *-999* for producer and detritus groups.

Predator Effect on Feeding Time: This column contains the parameters that specify how the feeding times of consumer groups change depending on the abundance of their predators in the foraging arena. This value must be set to *-999* for producer and detritus groups.

Density Dependent Catchability (Qmax/Q0): This value must be set to *-999* for producer and detritus groups. Otherwise, set accordingly.

Maximum Relative Consumption: This column contains the parameters that specify how much the consumption rates of predators can increase when there is abundant prey. This value must be set to *-999* for producer and detritus groups.

Prey Switching Power Parameter: This column contains the values that specify how the preferences of consumer groups for specific types of prey

change. This value must be set to *-999* for producer and detritus groups.

Advect $_?$: This column applies to only *Ecospace-F*. It must be specified regardless of *Ecospace-F* is being used. It indicates whether the functional group is subject to advection due to currents. *1* must be entered if the group is subject to advective motion, e.g. phytoplankton, otherwise *0* must be entered.

4.3.4.2 The vulnerability matrix input file:

<YourModelName>_vul.csv

Similar to the input file containing the group information matrix, this file is also a comma-separated ASCII file. The first two rows and their respective two columns are reserved for the specification of the size (#rows (equal to the number of state variables) * #columns (equal to the number of consumer groups)) of the vulnerability data matrix as shown in Figure 4.6. The fourth row and its respective first column in the file are reserved for the specification of the names of the state variables. On the fourth row and its respective second column, the data matrix starts. The prey groups are located row-wise and the predators are located column-wise. If there is no prey-predator interaction (no foraging arena) between two given groups, a value of *-999* must be entered. Otherwise, the corresponding vulnerability value specific to that foraging arena must be entered.

Number of groups	52													
Number of predators	48													
Prey/Pred	0-12 Snook	3-12 Snook	12-48 Snook	48-90 Snook	90+ Snook	0-3 Red Drum	3-8 Red Drum	8-18 Red Drum	18-36 Red Drum	36+ Red Drum	0-3 Sea Trout	3-18 Sea Trout	18+ Sea Trout	0-3 Sand Trout
0-12 Snook	-999	2	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
3-12 Snook	-999	-999	-999	1	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
12-48 Snook	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
48-90 Snook	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
90+ Snook	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
0-3 Red Drum	-999	-999	1	1	-999	-999	-999	-999	1	-999	-999	-999	1	-999
3-8 Red Drum	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
8-18 Red Drum	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
18-36 Red Drum	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
36+ Red Drum	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
0-3 Sea Trout	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
3-18 Sea Trout	-999	-999	-999	1	1	-999	-999	-999	-999	-999	-999	-999	1	-999
18+ Sea Trout	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
0-3 Sand Trout	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
3-12 Sand Trout	-999	-999	-999	-999	1	-999	-999	-999	-999	-999	-999	-999	-999	-999
12+ Sand Trout	-999	-999	-999	-999	1	-999	-999	-999	-999	-999	-999	-999	-999	-999
0-6 Mullet	-999	1	1	1	2	-999	-999	1	1	1	-999	-999	1	-999
6-18 Mullet	-999	-999	-999	-999	1	-999	-999	1	1	1	-999	-999	1	-999
18+ Mullet	-999	-999	-999	-999	1	-999	-999	-999	-999	-999	-999	-999	-999	-999
Mackrel 0-3	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
Mackrel 3+	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
Ladyfish 0-10	-999	-999	1	-999	2	-999	-999	-999	-999	-999	-999	-999	-999	-999
Ladyfish 10+	-999	-999	-999	-999	1	-999	-999	-999	-999	-999	-999	-999	-999	-999
Jacks	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
Bay Anchovy	-999	-999	1	2	1	-999	-999	1	1	1	-999	2	1	-999
Pin Fish	-999	1	1	1	1	-999	-999	1	1	1	-999	-999	1	-999
Spot	-999	-999	1	1	1	-999	-999	1	1	1	-999	-999	1	-999
Silver Perch	-999	-999	1	1	1	-999	-999	1	1	1	-999	2	2	-999

Figure 4.6: Sample ASCII file containing the vulnerability matrix for Tampa Bay EwE model.

4.3.4.3 The forcing data input file:

<YourModelName>_F.csv

This file is a comma-separated ASCII file. The first two rows and their respective two columns are reserved for the specification of the size (#rows (equal to the simulation time in months) * #columns (equal to the number of state variables)) of the forcing data matrix. The third row is reserved for the group numbers and the fourth row is reserved for the forcing type (Figure 4.7). Currently, only fishing mortality and fishing effort forcing types are enabled with type numbers of 4 and 5 respectively. The forcing time series must be prescribed MONTHLY. The fifth row is reserved for the specification of the names of the state variables. The first column of the sixth row in the file denotes months in the *Ecosim-F* simulation time span. On the sixth row and its respective second column, the data matrix starts.

 **Important!**

Forcing type 5, i.e. fishing effort time series, could only be used provided that only one single fleet exploits the stocks. It is not possible to read in fishing effort time series in case of multiple fleets. In such case of multiple fleets, the Ecosim-F requires that the time series of fishing mortalities are calculated externally using the time series of fishing efforts and the Ecopath-calculated initial exploitation rates by fleet. These are simple calculations that will result in the fishing mortality time series by species to be used in the Ecosim-F run.

If no forcing is applied to the corresponding state variable, a value of -999 must be entered. After the reserved first five rows, each row in the file corresponds to a month. The forcing time series data must be monthly, therefore, the number of rows must equal to the number of months in the simulation time frame, i.e. if the simulation is to be run for 51 years, then the forcing data matrix must include 612 rows (51 years * 12 months), hence, each row corresponding to a month.

Number of months	120												
Number of groups	37												
Group number	1	2	3	4	5	6	7	8	9	10	11	12	
Forcing type	4	4	4	4	4	4	4	4	4	4	4	4	
Group name	Baleen whales	Toothed whales	Seals	Birds	Sharks, large	Sharks, small medium	Rays, large	Rays, small medium	Pelagics, large	Pelagics, medium	Pelagics, small, carniv.	Pelagics, small, herbiv.	
Month 1	-999	-999	-999	-999	0.1	0.133333325	0.12	0.06666667	0.125	0.3333333	0.02	0.04	
Month 2	-999	-999	-999	-999	0.1	0.133333325	0.12	0.06666667	0.125	0.3333333	0.02	0.04	
Month 3	-999	-999	-999	-999	0.1	0.133333325	0.12	0.06666667	0.125	0.3333333	0.02	0.04	
Month 4	-999	-999	-999	-999	0.1	0.133333325	0.11964	0.06666667	0.125	0.3333333	0.02	0.04	
Month 5	-999	-999	-999	-999	0.0997	0.132933319	0.11964	0.06646667	0.124625	0.3323333	0.01994	0.03988	
Month 6	-999	-999	-999	-999	0.1125	0.149999991	0.135	0.075	0.140625	0.375	0.0225	0.045	
Month 7	-999	-999	-999	-999	0.1272	0.16959998	0.15264	0.0848	0.159	0.424	0.02544	0.05088	
Month 8	-999	-999	-999	-999	0.1437	0.1916	0.17244	0.0958	0.179625	0.479	0.02874	0.05748	
Month 9	-999	-999	-999	-999	0.1437	0.1916	0.17244	0.0958	0.179625	0.479	0.02874	0.05748	
Month 10	-999	-999	-999	-999	0.1565	0.208666667	0.1878	0.1043333	0.195625	0.5216666	0.0313	0.0626	
Month 11	-999	-999	-999	-999	0.1638	0.218399987	0.19656	0.1092	0.20475	0.5459999	0.03276	0.06552	
Month 12	-999	-999	-999	-999	0.1711	0.228133321	0.20532	0.1140667	0.213875	0.5703333	0.03422	0.06844	
Month 13	-999	-999	-999	-999	0.1803	0.240399987	0.21636	0.1202	0.225375	0.601	0.03606	0.07212	
Month 14	-999	-999	-999	-999	0.1949	0.259866655	0.23388	0.1299333	0.243625	0.6496666	0.03898	0.07796	
Month 15	-999	-999	-999	-999	0.1949	0.259866655	0.23388	0.1299333	0.243625	0.6496666	0.03898	0.07796	
Month 16	-999	-999	-999	-999	0.2022	0.2696	0.24264	0.1348	0.25275	0.674	0.04044	0.08088	
Month 17	-999	-999	-999	-999	0.2114	0.28186667	0.25368	0.1409333	0.26425	0.7046667	0.04228	0.08456	
Month 18	-999	-999	-999	-999	0.2279	0.303866655	0.27348	0.1519333	0.284875	0.7596666	0.04558	0.09116	
Month 19	-999	-999	-999	-999	0.2279	0.303866655	0.27348	0.1519333	0.284875	0.7596666	0.04558	0.09116	
Month 20	-999	-999	-999	-999	0.2279	0.303866655	0.27348	0.1519333	0.284875	0.7596666	0.04558	0.09116	
Month 21	-999	-999	-999	-999	0.2425	0.3233333	0.291	0.1616666	0.303125	0.8083333	0.0485	0.097	
Month 22	-999	-999	-999	-999	0.2425	0.3233333	0.291	0.1616666	0.303125	0.8083333	0.0485	0.097	
Month 23	-999	-999	-999	-999	0.2425	0.3233333	0.291	0.1616666	0.303125	0.8083333	0.0485	0.097	
Month 24	-999	-999	-999	-999	0.2517	0.3356	0.30204	0.1678	0.314625	0.8389999	0.05034	0.10068	
Month 25	-999	-999	-999	-999	0.2517	0.3356	0.30204	0.1678	0.314625	0.8389999	0.05034	0.10068	

Figure 4.7: Sample ASCII file containing the forcing time series data matrix for Generic 37 EwE model.

4.3.4.4 The nutrient forcing data input file:

<YourModelName>_NutForce.csv

This file is a comma-separated ASCII file that includes nutrient forcings to be applied in the nutrient-specific calculations in the *Ecosim-F* scenario. This file is strictly limited to contain only one column of time series data and the first two rows and their respective first two columns are reserved for the specification of the size (#rows (equal to the simulation time in months) * #columns (1)) of the forcing data matrix (Figure 4.8). The third row is reserved for the specification of the name of the nutrient time series. The first column of the fourth row in the file denotes months in the *Ecosim-F* simulation time span. On the fourth row and its respective second column, the data matrix starts. **Thereafter, each row in the file corresponds to a month.** The nutrient forcing time series data must be monthly, therefore, the number of rows must equal to the number of months in the simulation time frame, i.e. if the simulation is to be run for 51 years then the nutrient forcing data must include 612 rows (51 years * 12 months), each row corresponding to a month. If no forcing is applied to the nutrients, you won't need to prepare this file.

Number of months	612
Number of nutrients	1
Group name	Nutrients1
Month 1	0.189
Month 2	0.204
Month 3	0.217
Month 4	0.247
Month 5	0.252
Month 6	0.257
Month 7	0.283
Month 8	0.333
Month 9	0.386
Month 10	0.413
Month 11	0.41
Month 12	0.404
Month 13	0.398
Month 14	0.394

Figure 4.8: Sample ASCII file containing the nutrient forcing time series data matrix for Tampa Bay EwE model.

4.3.4.5 The primary production forcing data input file: $<\text{YourModelName}>_{_}\text{PrimaryProdForce.csv}$

This file is a comma-separated ASCII file. The first two rows and their respective first two columns are reserved for the specification of the size (#rows (equal to the simulation time in months) * #columns (equal to the number of primary producer groups)) of the forcing time series. The third row is reserved for the specification of the names of the state variables that the time series will be applied to. The first column of the fourth row in the file denotes months in the *Ecosim-F* simulation time span. On the fourth row and its respective second column, the data matrix starts. **Thereafter, each row in the file corresponds to a month.** If no primary production forcing is applied to primary producer groups, you won't need to prepare this file. In case of some of the primary producer groups are forced, the time series for the ones that are not forced must be set to unity (Figure 4.9).

Number of months	120	
Number of producers	2	
Group name	Phytoplankton	Benthic plants
Month 1	1	1
Month 2	1	1
Month 3	1	1
Month 4	1	1
Month 5	1	1
Month 6	1	1
Month 7	1	1
Month 8	1	1
Month 9	1	1
Month 10	1	1
Month 11	1	1
Month 12	1	1
Month 13	1	1
Month 14	1	1

Figure 4.9: Sample ASCII file containing the primary production forcing time series data matrix for Generic 37 EwE model.

4.3.5 Description of the *Ecosim-F* output (results) files

Once the model finishes running, five comma-separated (CSV) ASCII files are created under the directory where the model was run; two files

named **Ecosim_absResults.dat** and **Ecosim_relResults.dat** that comprise simulated absolute and relative biomass values at each time step respectively, two files named **Ecosim_absResults_monthly.dat** and **Ecosim_relResults_monthly.dat** which comprise simulated absolute and relative monthly averaged biomass values respectively, and another file named **Ecosim_absCatches_monthly.dat** which comprises monthly catch values of fished groups. Every column in these files corresponds to a state variable in compliance with the order of rows in the main scenario input file. You can open the files with a spreadsheet program like LibreOffice Calc or Microsoft® Excel.

4.4 The *Ecospace-F*

The implementation of Ecospace as *Ecospace-F* somewhat differs from stock EwE as detailed in Section 4.4.1.

4.4.1 Implementation

Considering nektonic organisms, *Ecospace-F* extends the *Ecosim-F* scheme by addition of an exchange term as depicted in Figure 4.10 ((Walters et al., 1999)).

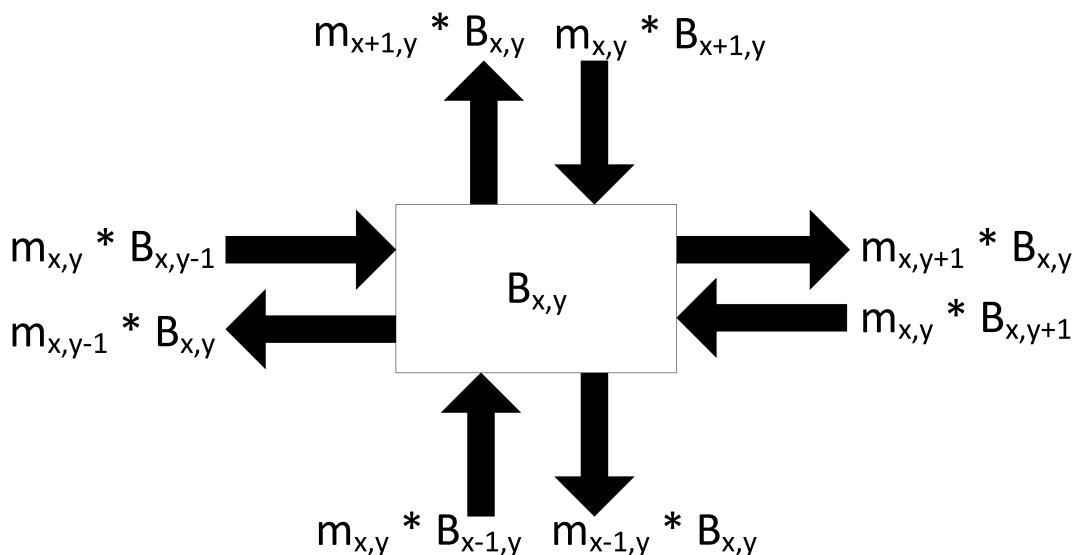


Figure 4.10: The movements of nektonic organisms to and from a grid cell in *Ecospace-F*. x and y denote latitude and longitude coordinates of the current cell respectively, B denotes biomass of an organism in cell x, y and m is the movement rate between grid cells. Movement between diagonal cells is not allowed.

The movement rate (m) shown in Figure 4.10 is a function of risk ratio, i.e. the probability of being eaten by predators in the cell to the probability of finding enough resources for consumption in the very same cell. This risk ratio is calculated for all the grid cells available for movement from the

current cell. Following this concept, *Ecospace-F* formulates the risk ratio (R) as in Equation 4.1.

$$R_{(x,y)} = \sum_{j=1}^n Q_{ij} / \sum_{j=1}^n Q_{ji} \quad (4.1)$$

where $\sum Q_{ij}$ is the sum of the predation on group i by all of its predators in grid cell x, y and $\sum Q_{ji}$ is the sum of the consumptions of group i over all of its preys in grid cell x, y . The movement rate between two adjacent cells, e.g. the current cell (x, y) and the cell right of it $(x, y + 1)$, is calculated as in Equation 4.2.

$$m_{x,y} = 1 - \frac{R_{x,y+1}}{R_{x,y} + R_{x,y+1}} \quad (4.2)$$

If $m_{x,y}$ is negative, then the risk ratio is higher in the adjacent cell compared to the current grid cell where the organism resides, and hence, no movement occurs toward that cell. Otherwise, a predetermined fraction of the organism moves to that cell. The risk ratios and movement rates are calculated separately for each organism in the grid cell.

Ecospace-F does not actively move planktonic organisms but let them advect with the prescribed currents, hence, *Ecospace-F* does not calculate risk ratios and movement rates for these type of organisms. Instead, planktonic organisms are advected with the prescribed currents. The prescription of currents is done by means of specifying net flow directions in degrees (see Section 4.4.6.3) for each grid cell in the model domain and flow speeds currently cannot be prescribed. During model simulation, each grid cell is evaluated for the overall flow direction in the cell and planktonic organisms are advected to any of the neighbouring grid cells in line with the net flow direction. At the land-sea boundary, if the net flow direction is towards the land, the planktonic organisms stay in their present grid cell. Unlike nektonic movement, transport of organisms to diagonal cells is allowed.

4.4.2 Files in the *src* directory

Unlike *Ecopath-F* and *Ecosim-F*, *Ecospace-F* is not a standalone application but an extension of *Ecosim-F* which enables the spatial simulation capabilities. The *Ecospace-F* source code resides in the source code directory of *Ecosim-F* and is split into six Fortran source files. *Ecospace-F* is enabled at compilation time within the *Makefile*. The list and corresponding descriptions of these source files are as follows:

- **statevartypesecospace.F90:** Global variables and derived variable types for the *Ecospace-F* are defined in this module file. All other model routines use this module.
- **ecospace.F90:** This is the main *Ecospace-F* model source file which carries out the spatial simulation within *Ecosim-F*.
- **readGridFile_io.F90:** This subroutine reads the spatial grid of the model domain for *Ecospace-F* scenario from its respective comma-delimited ASCII file. It is called by *ecosim.F90*.
- **readSpatialDistribution_io.F90:** This subroutine reads the relative spatial distributions of the model groups across the model grid cells for *Ecospace-F* scenario from its respective comma-delimited ASCII file. It is called by *ecosim.F90*.
- **readAdvectionFile_io.F90:** This subroutine reads the compass directions of advective flows in the model grid cells for *Ecospace-F* scenario from its respective comma-delimited ASCII file. It is called by *ecosim.F90*.
- **writenetCDFfile.F90:** This subroutine writes the results of the *Ecospace-F* model run to a NetCDF database file.

4.4.3 Setting up test cases for an *Ecospace-F* run

First, copy the template "filenames.nml" and "Makefile" under the test case directory that you would like to run to the *Ecosim-F*'s root directory.

Currently, only the Black_Sea test case is available for an *Ecospace-F* run. To set up *Ecospace-F* to run Black_Sea test case scenario, open a terminal in *Ecosim-F*'s root directory and issue the command shown in Listing 4.11 under a UNIX-like system.

Listing 4.11: Copying necessary files

```
cp testcases/Black_Sea/Black_Sea_filenames.nml ./filenames.nml
cp testcases/Black_Sea/Black_Sea_Makefile ./Makefile
```

Then, if necessary, edit the *Makefile* to match your system's Fortran compiler settings. The *Makefile* is self-explanatory, but in a nutshell you may need to edit lines 25, 42, 48, the region between lines 59-73 and the region between lines 87-88 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 and NetCDF libraries. The lines starting with a hash (#) are explanatory comment lines. Read them as you find your way through these lines. Once finished, compile the model as shown in Listing 4.16.

Listing 4.12: Compiling the model

```
ecospace=YES make
```

Then you may run the model as shown in Listing 4.17.

Listing 4.13: Running the model

```
bin/ecosim-f
```

4.4.4 Enabling *Ecospace-F* when compiling *Ecosim-F*

You may need to edit the **Makefile** before you run *make* in order to enable *Ecospace-F* simulation by defining the *_Ecospace_* macro as shown below:

```
# Uncomment below if there is geospatial simulation
CPPDEFS += -D_Ecospace_
```

Afterwards, just compile the models as shown in Listing 4.14.

Listing 4.14: Compiling the model

```
ecospace=YES make
```

The *make* command will create an executable named **ecosim.out**. This binary file is the model with *Ecospace-F* enabled. You can run it by typing *bin/ecosim.out* in the terminal.

Whenever you wish to clean-up the *EwE-F/ecosim-f* directories, you need to run *make clean* as shown in Listing 4.15.

Listing 4.15: Cleaning up the model

```
ecospace=YES make clean
```

This will delete the object and module files created by *make* and the respective executable.

Information

EwE-F was compiled and tested against GNU FORTRAN compiler. Other compilers may or may not work.

4.4.5 Setting up your model for an *Ecospace-F* run

In order to be able to run the model with a specific Ecospace scenario, one must edit the **filenames.nml** Fortran namelist file and append the corresponding parameters to the file in addition to the parameters detailed in

Section 4.3.3 for *Ecosim-F*. The explanations of the additional parameters in the namelist file are as follows:

1. *SpatialGrid_fname*: Enter the path and name of the CSV (**Comma-Separated Values**) file that comprises the spatial grid of the model domain for *Eospace-F* scenario (see 4.4.6.1).
2. *SpatialDistribution dirname*: Enter the path and name of the directory that comprises the spatial distributions of model groups across the grid cells in the model domain for *Eospace-F* model scenario. This directory contains one CSV (**Comma-Separated Values**) file for each model group which defines its relative distribution across the grid cells in the model's geographic domain (see 4.4.6.2).
3. *Advection_fname*: Enter the path and name of the CSV (**Comma-Separated Values**) file that comprises the advective flow directions for grid cells in the model domain for *Eospace-F* scenario (see 4.4.6.3).

 **Important!**

All the field values of the fields specified above must be enclosed in double quotes.

Edit the *Makefile* to match your system's Fortran compiler settings. The *Makefile* is self-explanatory, but in a nutshell you may need to edit lines 25, 42, 48, the region between lines 59-73 and the region between lines 87-88 to match your Fortran compiler's settings and correctly point the compiler to the location of installed HDF5 and NetCDF libraries. The lines starting with a hash (#) are explanatory comment lines. Read them as you find your way through these lines. Once finished, compile the model as shown in Listing 4.16.

Listing 4.16: Compiling the model

```
ecospace=YES make
```

Then you may run the model as shown in Listing 4.17.

Listing 4.17: Running the model

```
bin/ecosim-f
```

4.4.6 Description of the input files

All *EwE-F* input files are comma-separated (CSV) ASCII input files. These files could easily be prepared using common spreadsheet software such as LibreOffice Calc or Microsoft® Excel and then saved as CSV using "Save as..." under *File* menu.

4.4.6.1 The spatial grid input file:

<YourModelName>_grid.csv

This file is a comma-separated ASCII file. The first two rows and their respective first two columns are reserved for the specification of the size (#rows (equal to the number of latitude points) * #columns (equal to the number of longitude points)) of the grid space in the spatial domain of the *Ecospace-F* scenario (Figure 4.11). Afterwards, there comes the specification of the model domain; "1" indicates "sea" regions and "0" indicates "land" areas.

4.4.6.2 The spatial distributions input files: "#.csv"

These files are comma-separated ASCII files. The naming of the files should start from "1" up to the number of groups (n groups): "1.csv", "2.csv", ..., "10.csv", "11.csv"..., "n.csv". The number should match the group order in the *Ecosim-F* scenario file (see 4.3.4.1). The first two rows and their respective two columns in the files are reserved for the specification of the size (#rows (equal to the number of latitude points) * #columns (equal to the number of longitude points)) of the grid space in the spatial domain of the *Ecospace-F* scenario (Figure 4.12). Afterwards, there comes the specification of the relative distributions of functional groups in the model domain; "0" indicates an uninhabited cell and a real number between "0" and "1" indicates the relative proportion of the biomass of the group within that cell with respect to the *Ecopath-F* biomass. The sum of all rows and columns in the files (excluding the first two rows) must be equal to "1".

```
"Number of latitude points",10
"Number of longitude points",10
0,0,0,0,0,0,0,0
0,0,01152,0,00699,0,01627,0,00398,0,01615,0,01755,0,01845,0,01737,0
0,0,02494,0,02012,0,00562,0,03866,0,03499,0,01101,0,02753,0,00113,0
0,0,01546,0,03039,0,03194,0,01149,0,00333,0,03217,0,02274,0,03642,0
0,0,02428,0,03068,0,00637,0,01660,0,03431,0,01766,0,0,01596,0
0,0,02184,0,00444,0,03387,0,0,00584,0,00459,0,0,03166,0
0,0,03329,0,00247,0,03430,0,0,00711,0,03387,0,0,02202,0
0,0,03496,0,0,0,0,0,02468,0,00449,0,0,00012,0
0,0,02689,0,0,0,0,0,01903,0,01804,0,0,03443,0■
```

Figure 4.12: Sample comma-separated ASCII file containing the relative spatial distribution of a functional group in an *Ecospace-F* hypothetical 10x10 spatial grid.

4.4.6.3 The advection input file:

<YourModelName>_Advection.csv

The first two rows and their respective first two columns in the file are reserved for the specification of the size (#rows (equal to the number of latitude points) * #columns (equal to the number of longitude points)) of the grid space in the spatial domain of the *Ecospace-F* scenario (Figure 4.11).

Afterwards, there comes the specification of the directions of the advective flows in each grid cell of the model domain; 180 indicated eastward flow, -180 indicates westward flow, -90 indicates southward flow, 90 indicates northward flow, -45 indicates southeastward flow, 45 indicates northeastward flow, -135 indicates southwestward flow and 135 indicates northwestward flow (Figure 4.13). For land parts of the grids and grids that correspond to sea areas where there is no advective flow direction, a value of 0 is entered.

Information

Currently only advective flow directions are incorporated in Ecospace-F and flow speeds ARE NOT implemented. As in line with the implementation of directional movements of nekton in Ecospace-F, it is assumed that advective flows can transfer the advected organisms only to adjacent grid cells (however, unlike nektonic movement, this time including grid cells located in the diagonals of the evaluated grid cell) in one time step of EwE-F.

4.4.7 Description of the *Ecospace-F* output (results) file

Once the model finishes running, one NetCDF file is created under the directory where the model was run; named **ecospace_4D.nc**. The file comprises only the biomasses (but not catches) of the state variables in their respective variable; *Biomass*. It has 4 dimensions: *nFunctionalGroup* x *nTime* x *nLongitude* x *nLatitude*. The variable has only one attribute *_FillValue* which is set to -999 .

When *Ecospace-F* is used, the comma-separated ASCII output files of *Ecosim-F*, which are described in section 4.3.5, do not contain any data from the simulation results, hence, cannot be utilised.

Bibliography

- Christensen, V., Walters, C. J., 2004. Ecopath with Ecosim: methods, capabilities and limitations. *Ecological Modelling* 172, 109–139.
- Christensen, V., Walters, C. J., Pauly, D., November 2005. Ecopath with Ecosim: A User's Guide. Fisheries Centre, University of British Columbia, Vancouver, Canada, 154 p.
- Fulton, E. A., 2010. Approaches to end-to-end ecosystem models. *Journal of Marine Systems* 81, 171–183.
- Walters, C., Pauly, D., Christensen, V., 1999. Prediction of mesoscale spatial patterns in trophic relationships of exploited ecosystems, with emphasis on the impacts of marine protected areas. *Ecosystems* 2(6), 539–554.