

---

# **XBeachMI Documentation**

***Release 0.1***

**Bas Hoonhout**

June 28, 2017



<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	1. Download XBeach . . . . .	3
1.2	2. Download Python . . . . .	5
1.3	3. Download BMI and XBeach MI . . . . .	5
1.4	4. Install BMI and XBeach MI . . . . .	5
1.5	5. Test XBeach MI . . . . .	9
<b>2</b>	<b>Examples</b>	<b>13</b>
2.1	Sequential . . . . .	13
2.2	Parallel (MORMERGE) . . . . .	14
2.3	Nesting . . . . .	14
<b>3</b>	<b>Source code documentation</b>	<b>15</b>
3.1	model . . . . .	15
3.2	cmd . . . . .	15
<b>4</b>	<b>MPI</b>	<b>17</b>
4.1	Examples . . . . .	17
<b>5</b>	<b>Command-line tools</b>	<b>19</b>
5.1	xbeach-mi . . . . .	19
<b>6</b>	<b>Source code repository</b>	<b>21</b>
<b>7</b>	<b>Indices and tables</b>	<b>23</b>



XBeach MI (Multiple Instances) is a Python wrapper for XBeach that allows the user to run multiple instances of XBeach simultaneously. Each instance can have its own settings. Possible applications are:

1. Running XBeach in stationairy and instationairy mode alternating depending on the wave conditions
2. Running XBeach with multiple wind conditions and averaging the result (MORMERGE)

A BMI-compatible version of XBeach is needed to use with the XBeach MI Python wrapper. The wrapper itself is also BMI compatible.

Contents:



---

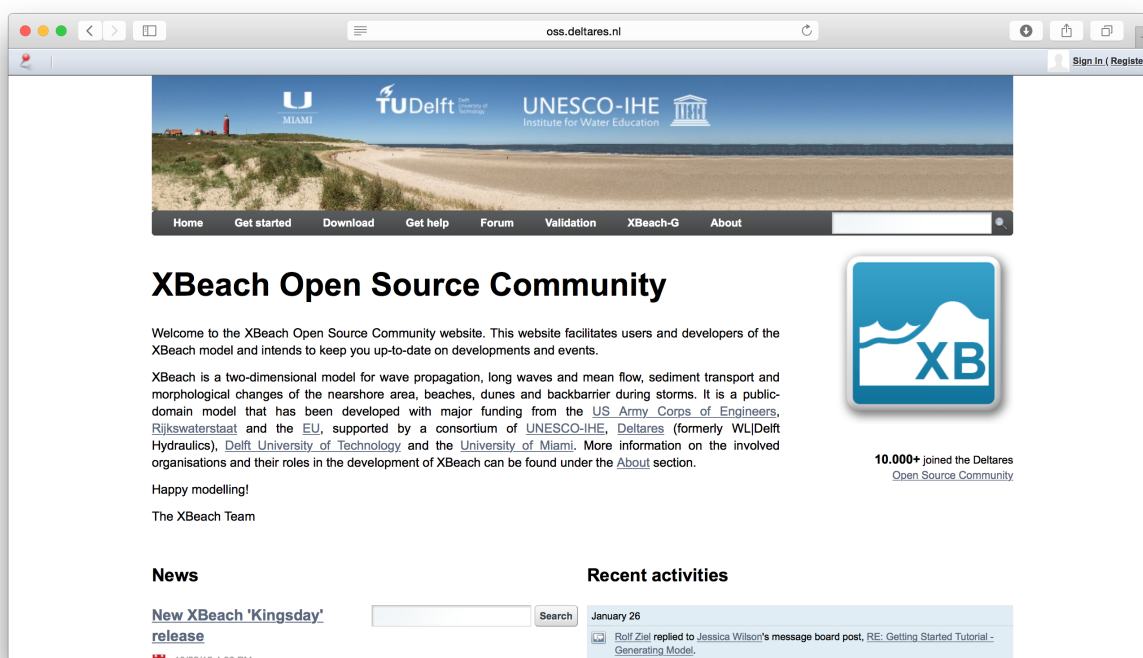
## Installation

---

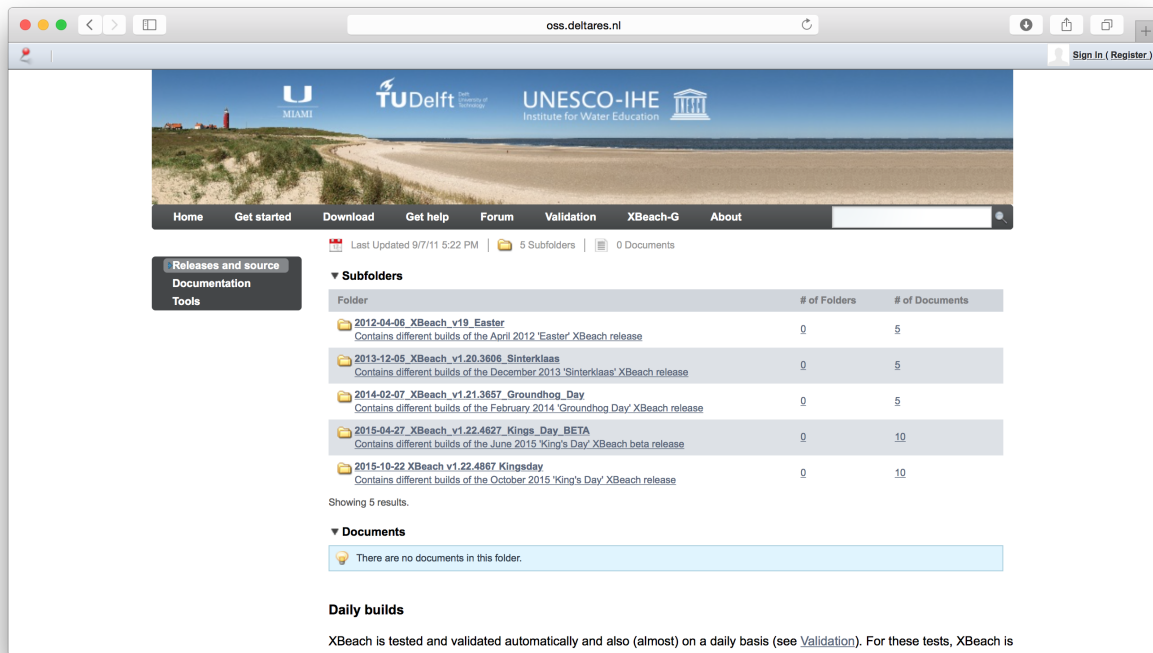
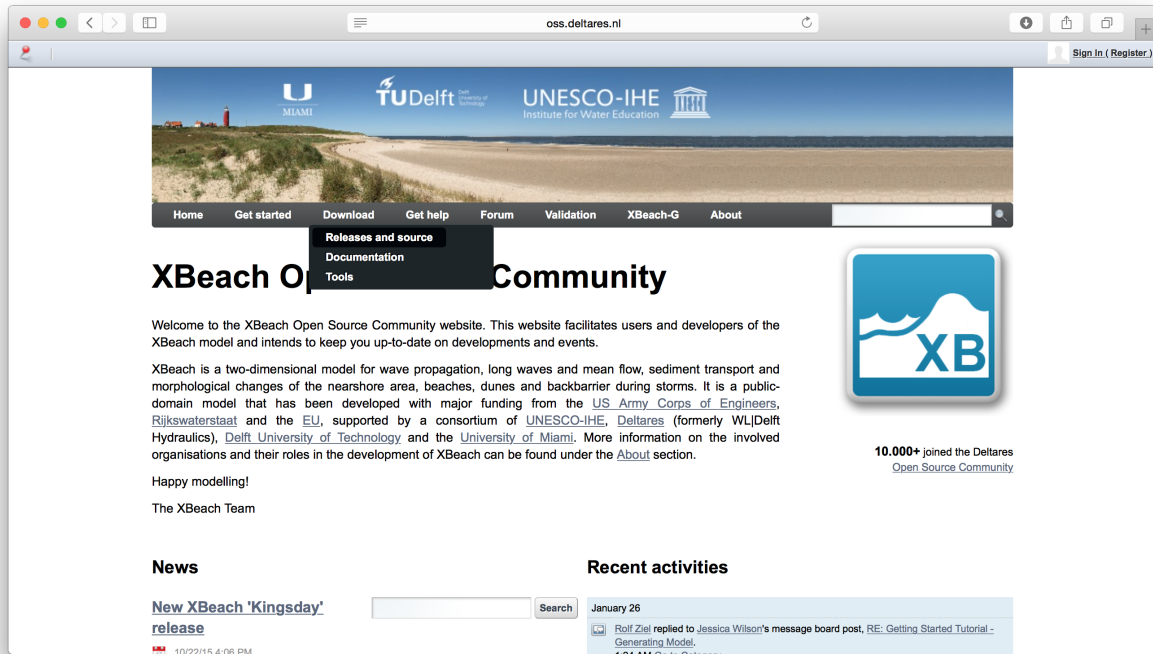
This document describes the installation steps needed to get the XBeach MI (Multiple Instances) framework to work on a Windows machine (the screenshots may deviate slightly as they are from a Mac OS X machine). If you have a working XBeach installation you can skip step 1. If you have a working Python 2.7 installation with packages numpy, netCDF4, multiprocessing and mako installed, you can skip step 2.

### 1. Download XBeach

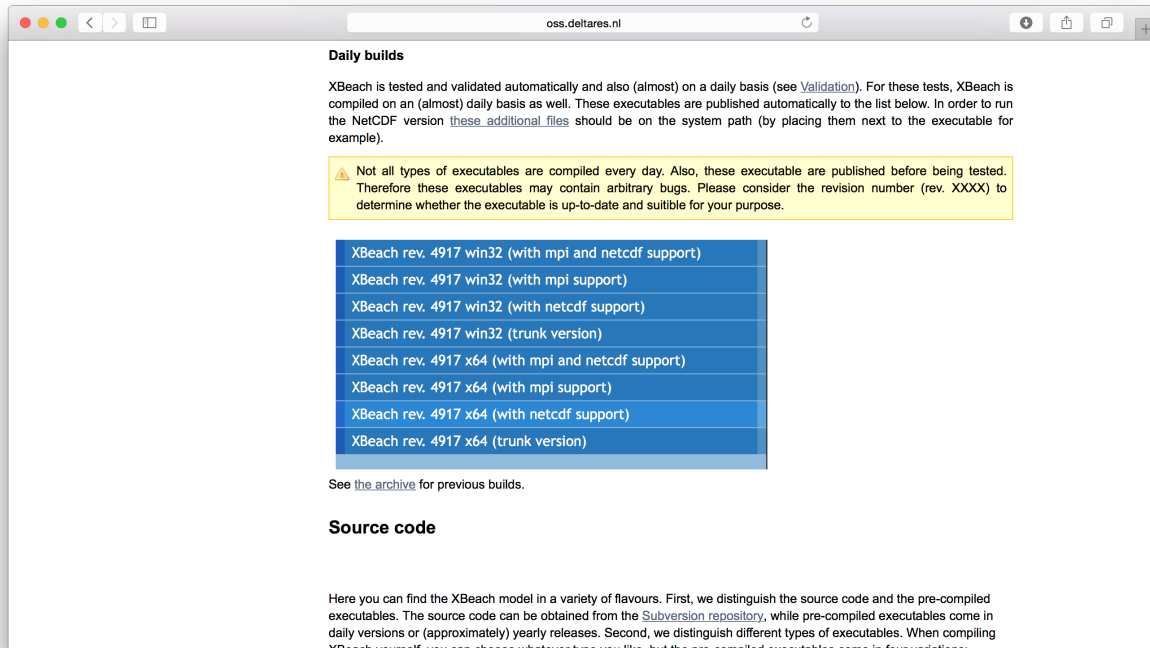
1. Go to <http://xbeach.org>.



1. Choose Downloads and Releases and source.
1. Go to Daily builds and download XBeach rev. XXXX (with netCDF support). You need a recent version of XBeach (rev.  $\geq 4748$ ) that implements the Basic Model Interface (BMI).







## 2. Download Python

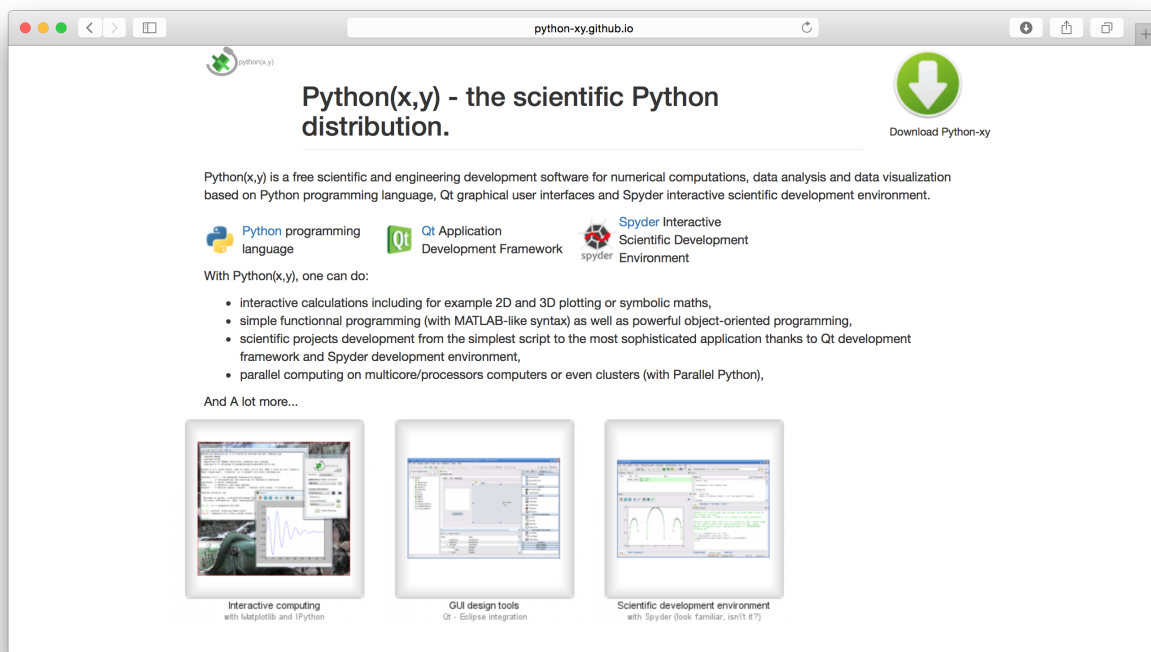
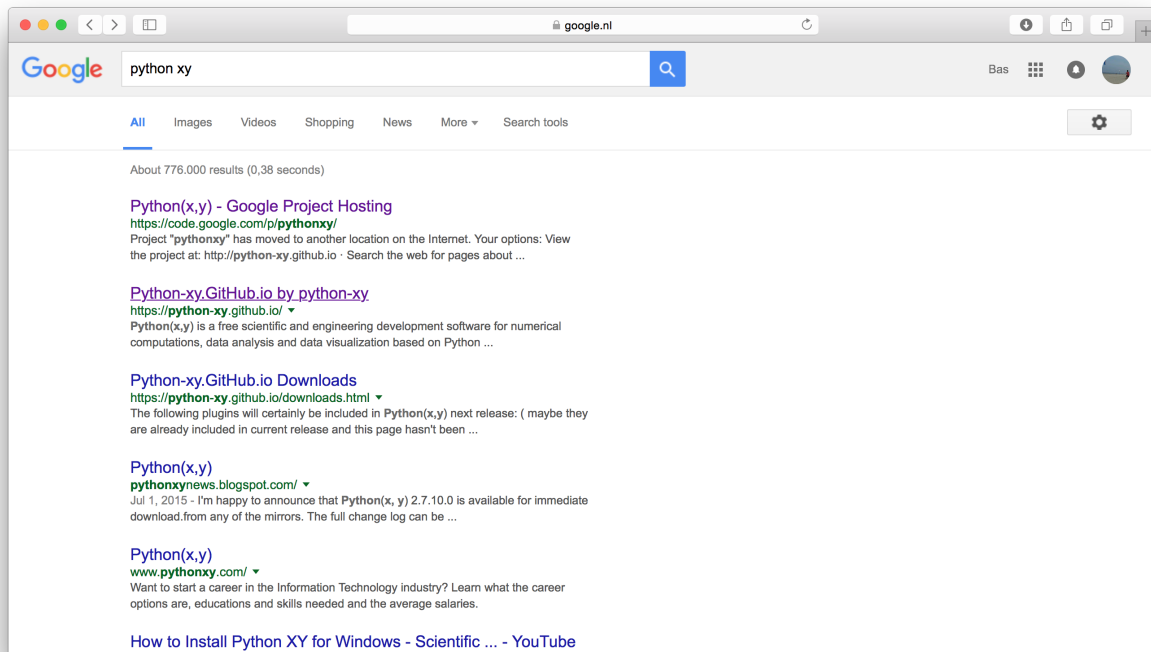
1. Google for Python XY (or Anaconda).
1. Follow the instructions to download Python XY (or Anaconda).
1. Follow the instructions to install Python XY (or Anaconda). Do not forget to choose Full install and not the default installation configuration.

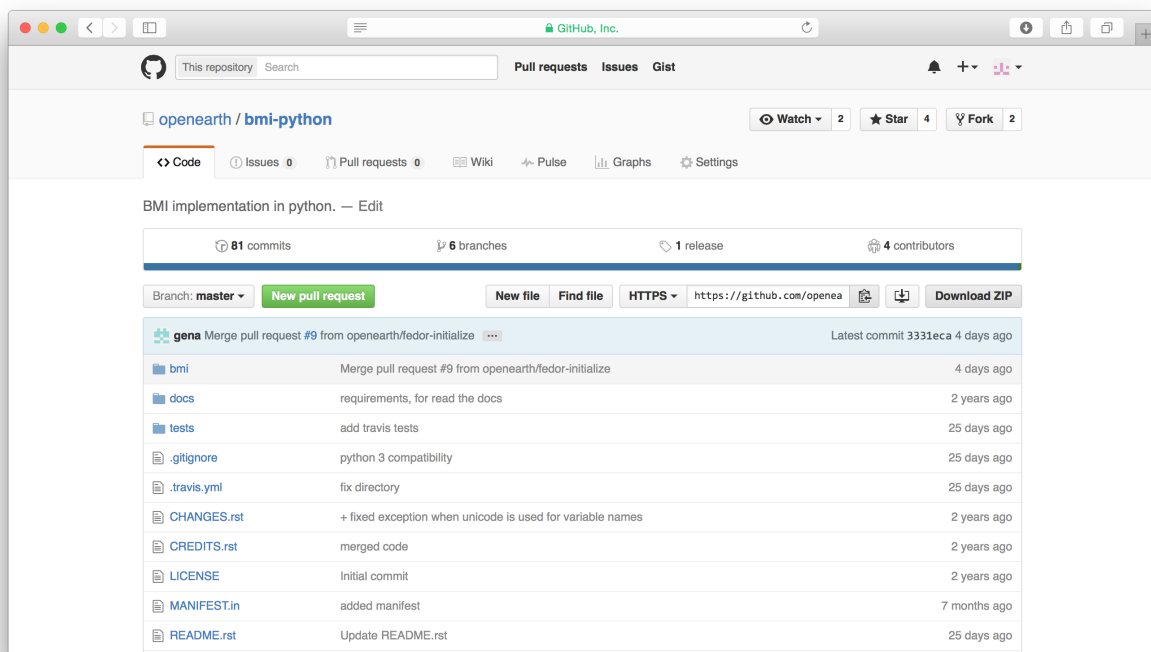
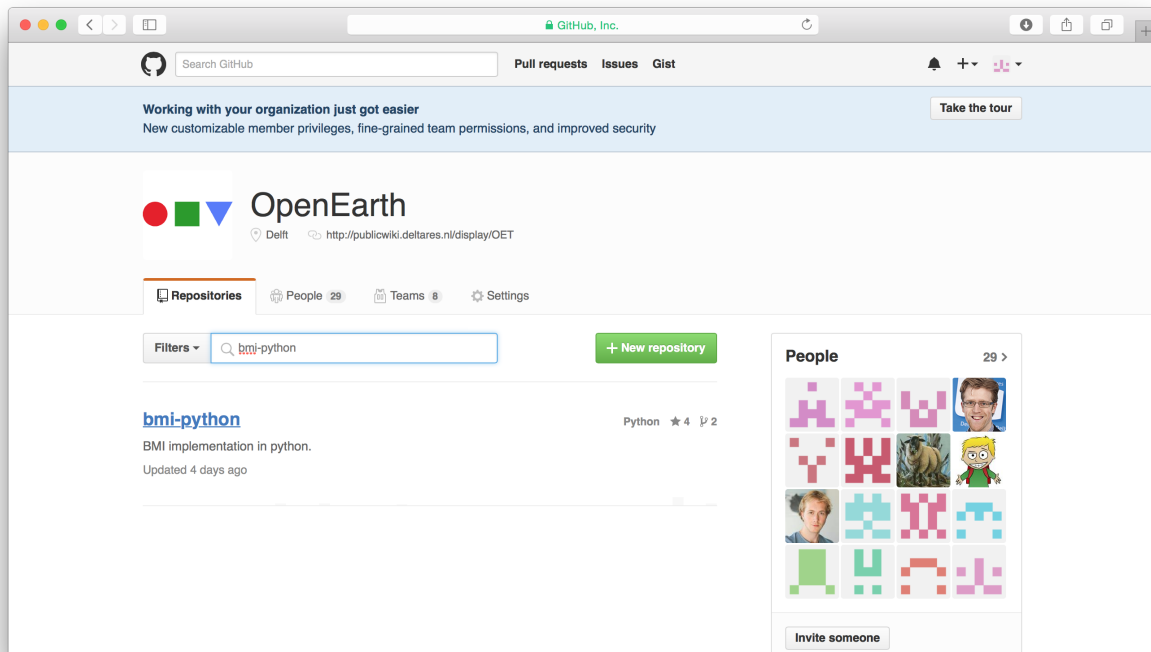
## 3. Download BMI and XBeach MI

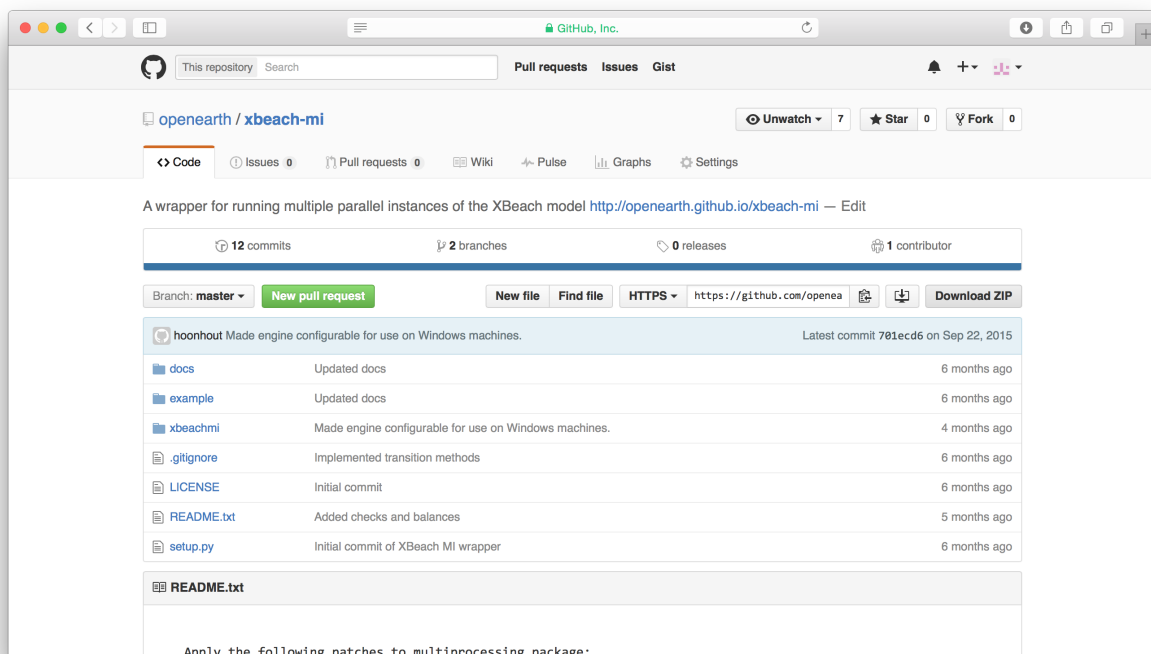
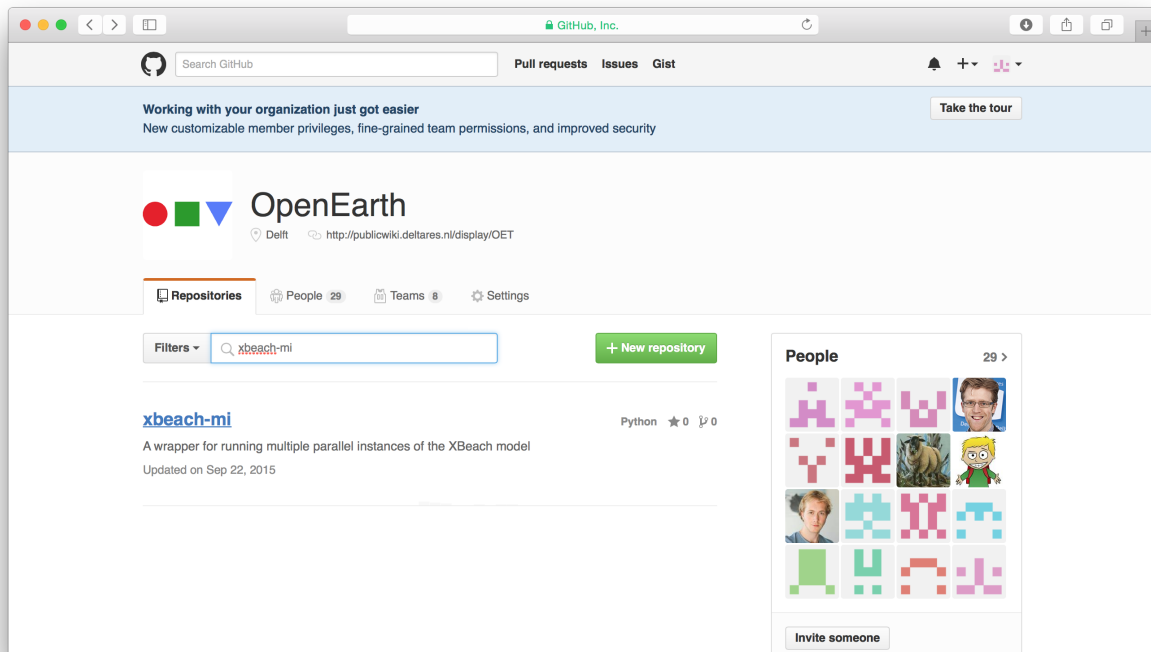
1. Go to <http://github.com/openearth> and search for bmi-python.
1. Choose Download ZIP.
1. Go back and search for xbeach-mi.
1. Choose Download ZIP.

## 4. Install BMI and XBeach MI

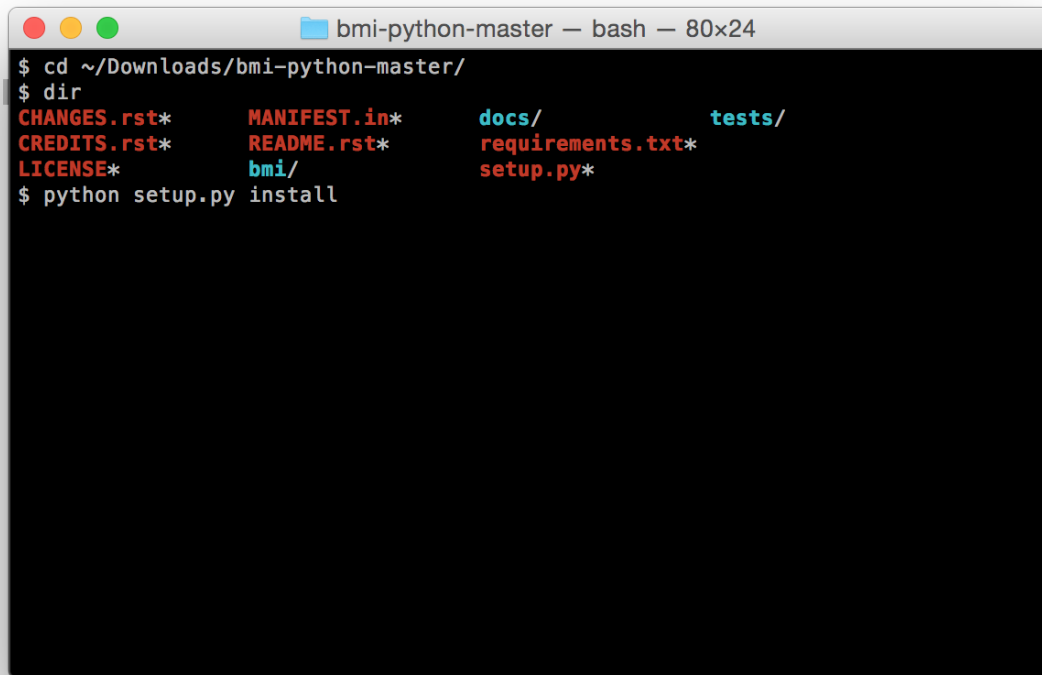
1. Unzip the downloaded bmi-python and xbeach-mi packages.
2. Go to the command line (Start > cmd).
3. Go to the download directory of the bmi-python package.
4. Go to the directory that contains the setup.py file.







5. Run the command `python setup.py install`.

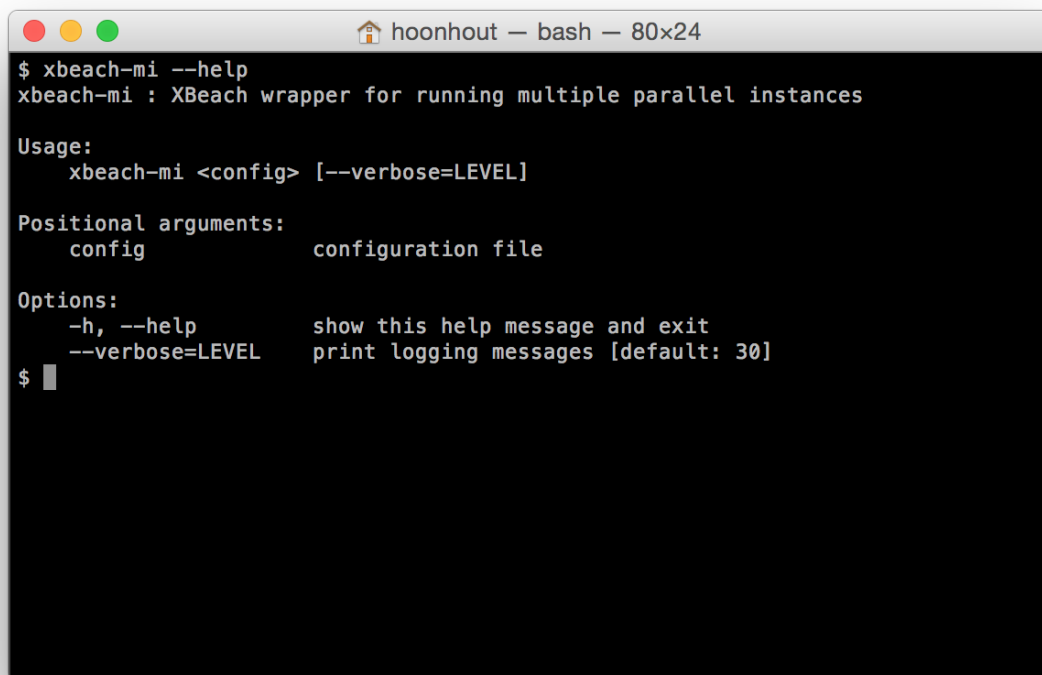


```
bmi-python-master — bash — 80x24
$ cd ~/Downloads/bmi-python-master/
$ dir
CHANGES.rst*  MANIFEST.in*  docs/  tests/
CREDITS.rst*  README.rst*  requirements.txt*
LICENSE*  bmi/  setup.py*
$ python setup.py install
```

1. Go to the download directory of the `xbeach-mi` package that contains the `setup.py` file.
2. Again, run the command `python setup.py install`.

## 5. Test XBeach MI

1. Test the installation by running the command `xbeach-mi --help`.
1. Go to the `example` directory in the `xbeach-mi` download directory.
2. Run XBeach MI with the provided configuration file.
1. Details on configuring and running XBeach MI can be found on the documentation website: [Examples](#)



A terminal window titled 'hoonhout — bash — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal displays the command `$ xbeach-mi --help` and its output. The output describes `xbeach-mi` as an 'XBeach wrapper for running multiple parallel instances'. It shows the usage as `xbeach-mi <config> [--verbose=LEVEL]`. It lists a positional argument 'config' as the 'configuration file'. It also lists two options: `-h, --help` to 'show this help message and exit', and `--verbose=LEVEL` to 'print logging messages [default: 30]'. The prompt `$` is followed by a cursor.

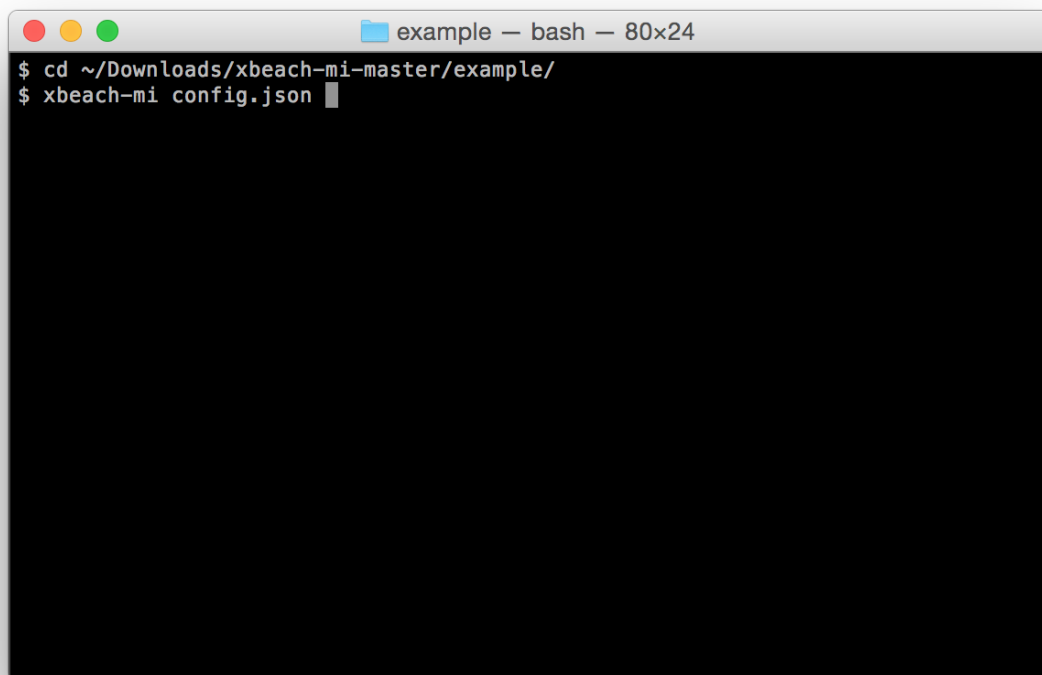
```
$ xbeach-mi --help
xbeach-mi : XBeach wrapper for running multiple parallel instances

Usage:
  xbeach-mi <config> [--verbose=LEVEL]

Positional arguments:
  config          configuration file

Options:
  -h, --help      show this help message and exit
  --verbose=LEVEL print logging messages [default: 30]

$
```

A terminal window with a title bar containing three colored buttons (red, yellow, green) and a title "example — bash — 80x24". The terminal has a black background with white text. The first line shows a shell prompt "\$" followed by the command "cd ~/Downloads/xbeach-mi-master/example/". The second line shows the same prompt "\$" followed by the command "xbeach-mi config.json" and a cursor.

```
example — bash — 80x24
$ cd ~/Downloads/xbeach-mi-master/example/
$ xbeach-mi config.json
```





---

## Examples

---

---

**Note:** In order to run XBeach MI you need a compiled XBeach library with BMI interface (bmi branch or trunk rev  $\geq 4748$ ) that is in your path.

---

**Note:** You can find the full examples described here in the GIT repository <https://github.com/openearth/xbeach-mi/tree/master/example>

---

For an XBeach MI run you need at least two files:

1. Regular `params.txt` file for XBeach
2. A XBeach MI JSON configuration file

In the following sections two example setups for XBeach MI are explained. The first is a sequential run in which multiple XBeach instances run in sequence (e.g. surfbeat for storm conditions and stationary for average conditions). The second is a parallel run in which multiple XBeach instances are run in parallel, while their bathymetry is averaged (MORMERGE). It is possible to combine both approaches in a single setup.

## Sequential

The XBeach MI configuration file describes the communication between the different instances. It is in JSON format as shown below. Two sections are particularly important for a sequential run: `scenario` and `exchange`.

`scenario` describes what instances should be activated at what point in time. For a sequential run only one instance runs at a time. Note that it is possible to provide a list of instances, which would enable the `parallel` mode. Also note that all instances are activated by default at the start of the simulation.

`exchange` provides a list with all variables that are being exchanged between instances in case the model switches from one instance to another. For most sequential runs this list needs to contain the majority of the XBeach output variable for the best results as it ensures that the full model state is copied from one instance to another.

Note that the XBeach MI configuration file references a single `params.txt` file through the `params_file` keyword. Consequently, all instances share the same `params.txt` file and no differences between the instances exist. Therefore, it is possible to add templating makers to the `params.txt` file to make it instance dependent. In the example below, the instance `instat` defines surfbeat boundary conditions, while the instance `stat` defines stationary boundary conditions. Please refer to the Mako templating engine for all the possible templating options.

---

**Note:** Note that the grids of all instances should be equal, so it is not allowed to define different values for `nx` and `ny` between instances.

---

## Parallel (MORMERGE)

For a parallel run instances don't need to be defined by a scenario (although it is possible), but can be defined by a simple list using the `instances` keyword. The `exchange` list is generally much smaller than for a sequential run and typically only holds the `zb` variable.

In addition, a keyword `aggregate` can be defined that specifies how data from the different instances need to be aggregated. The `method` keyword defines the methodology or a reference to a custom Python function. The `options` keyword holds key/value pairs that are passed as options to the aggregation function. The `interval` defined the interval in seconds when the data between instances should be aggregated and exchanged.

Also a parallel run uses a single `params.txt` file that uses Mako templating differentiate between instances. In this example both instances run in surfbeat mode, but with different boundary conditions.

## Nesting

Not supported.

---

## Source code documentation

---

**model**

**cmd**



---

**MPI**

---

An MPI version of XBeach MI is being developed. This version can be ran within an MPICH2 or OpenMPI wrapper. Individual XBeach models can then be distributed over multiple cores for optimal performance.

The beta version of the MPI version can be found in a branch of the GitHub repository:

<https://github.com/openearth/xbeach-mi/tree/mpi>

## Examples

```
mpirun -n 8 xbeach-mi xbeachmi.json
```



---

## **Command-line tools**

---

The XBeach MI wrapper can be executed from the command-line using the “xbeach-mi” command. See for more information the *-help* option.

### **xbeach-mi**





---

## Source code repository

---

The XBeach MI source code can be downloaded from the OpenEarth GitHub repository:  
<https://github.com/openearth/xbeach-mi/>.



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*