
coastviewer Documentation

Release 0.1.12

Fedor Baart

Sep 28, 2018

Contents

1	Installation	3
1.1	Stable release	3
1.2	From sources	3
2	Usage	5
2.1	Introduction	5
2.2	Information	5
3	Contributing	9
3.1	Types of Contributions	9
3.2	Get Started!	10
3.3	Pull Request Guidelines	11
3.4	Tips	11
4	Credits	13
4.1	Development Lead	13
4.2	Contributors	13
5	History	15
5.1	0.1.0 (2017-07-13)	15
6	Indices and tables	17

Contents:

1.1 Stable release

To install coastviewer, run this command in your terminal:

```
$ pip install coastviewer
```

This is the preferred method to install coastviewer, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

1.2 From sources

The sources for coastviewer can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/openearth/coastviewer
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/openearth/coastviewer/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


2.1 Introduction

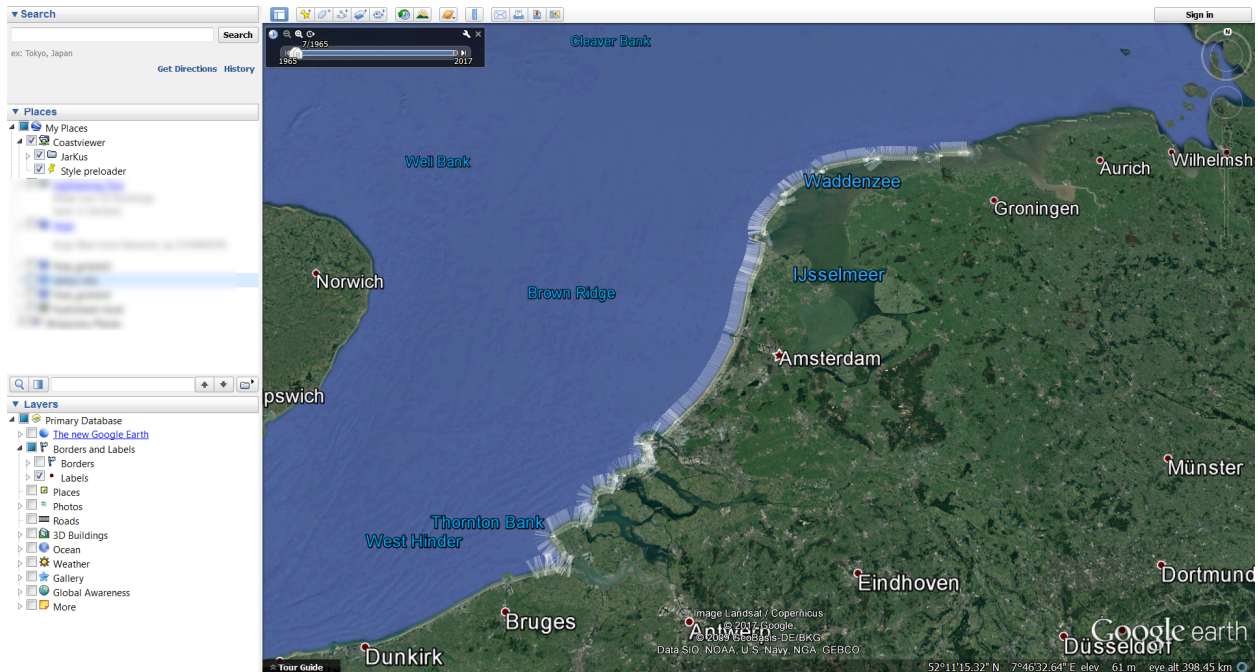
Obtained data from the entire Dutch coast is converted into one standardized, uniform format (NetCDF) and saved according to the [Open Earth](#) philosophy in an internet database. This data can be accessed through the OGC (Open Geospatial Consortium) KML feeds ([Google Earth](#) feeds), which can be openly accessed through the internet. This coastviewer provides these feeds. Through the KML file the data can be accessed and visualized on a map (Google Earth) more detailed information is given in graphs. This usage page describes how the data in the coastviewer can be accessed.

2.2 Information

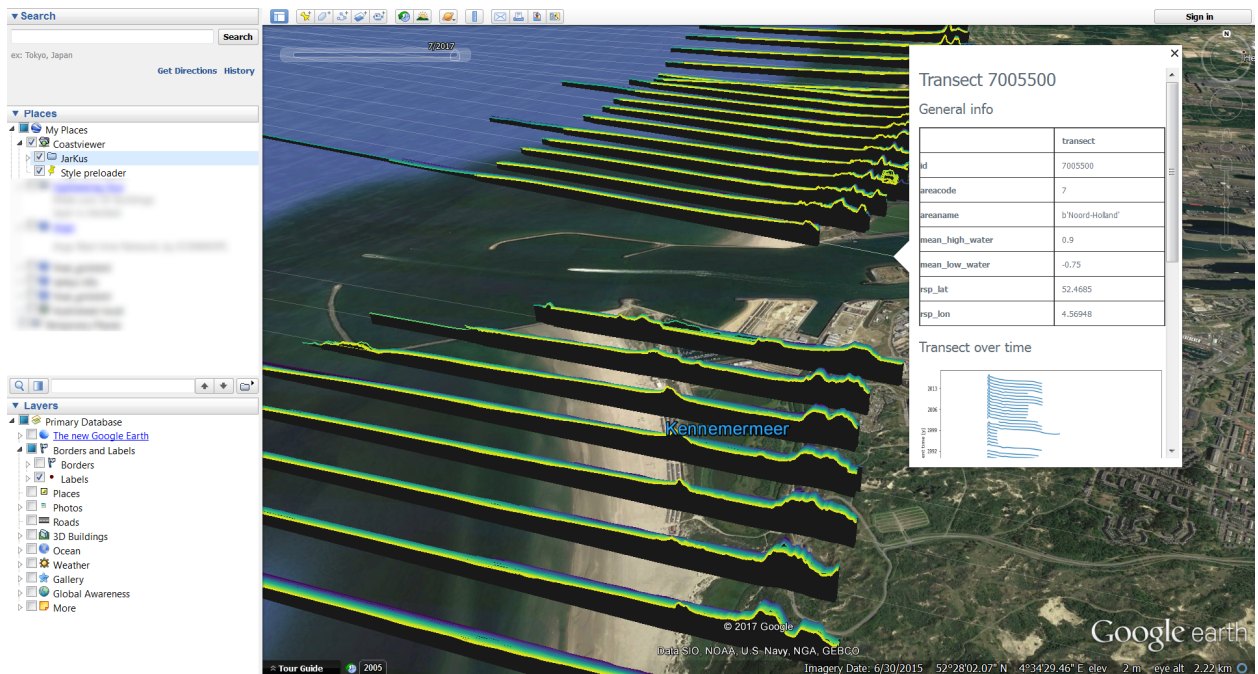
The visualization in Google Earth can be accessed in two ways, through installing the coastviewer with this package locally or by using the online version. Both methods provide a link to a KML file that can be opened in Google Earth.

- The coastviewer is a package that allows the user to run the program locally. In the [readme](#) the steps to install the coastviewer locally are described. The locally run package provides a link to the kml at: <http://localhost:5000/coastviewer/1.1.0/transects/kml>
- The network link of the online version: <http://coastal-dev.zdcxwh5vkz.eu-west-1.elasticbeanstalk.com/coastviewer/1.1.0/transects/kml>

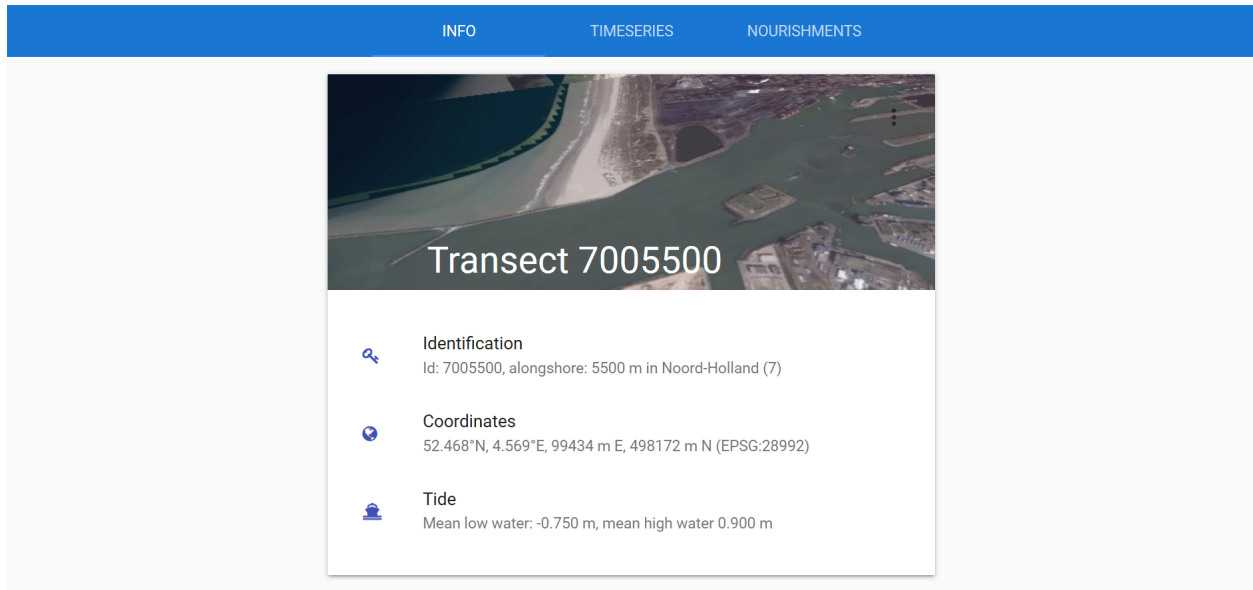
Within Google Earth click on Add -> Network Link. Provide a name and paste the network link. The JarKus transects are shown along the entire Dutch coast.



These transects can be analysed in more detail. When zooming into a section of the coast the individual depth profiles become visible. In the top left of the screen a time bar is shown. This bar can be used to select a time span of data to be shown. This gives a good overview of the overall dynamics of the coast profile. More information per transect is given when clicking on a transect. A small pop-up will appear containing general information and two transect plots over time. For even more information a link is provided when scrolling down in the pop-up called <details>. When clicked upon you are redirected to a webpage.



The webpage provides three tabs with information concerning the selected transect. The first tab 'INFO' provides general information about the transect.



The second tab 'TIMESERIES' provides two timeserie plots of the transect. The uppermost plot gives the cross-shore profile of the transect in time. The second plot does the same, but visualizes it in a topographic map and combines the mean high/low waterline contours. Both figures can be exported (png, pdf or svg format). The third tab 'NOURISHMENTS' shows three plots with indicators for the coastal state.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at <https://github.com/openearth/coastviewer/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

3.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

3.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

3.1.4 Write Documentation

coastviewer could always use more documentation, whether as part of the official coastviewer docs, in docstrings, or even on the web in blog posts, articles, and such.

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/openearth/coastviewer/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

3.2 Get Started!

Ready to contribute? Here's how to set up *coastviewer* for local development.

1. Fork the *coastviewer* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/coastviewer.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv coastviewer
$ cd coastviewer/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 coastviewer tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/openearth/coastviewer/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ py.test tests.test_coastviewer
```


4.1 Development Lead

- Fedor Baart <fedor.baart@deltares.nl>

4.2 Contributors

None yet. Why not be the first?

5.1 0.1.0 (2017-07-13)

- First release on PyPI.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`