SEAScope user manual

This document describes the configuration and user interface of the SEAScope viewer application.

It provides instructions to start the application, describes the graphical interface elements and the keyboard shortcuts that you can use to quickly display and browse your data in the 3D viewer.

More advanced subjects like configuration and extending capabilities using the SEAScope Python package are also covered in the second part of the document.

Other SEAScope-related subjects are covered by documents available on the SEAScope website:

- Installation instructions for the viewer and the Python package:

  `https://seascope.oceandatalab.com/docs/seascope_linux_install_manual_20190703.pdf`

  `https://seascope.oceandatalab.com/docs/seascope_macos_install_manual_20190703.pdf`

  `https://seascope.oceandatalab.com/docs/seascope_win32_install_manual_20190703.pdf`

- Intermediate data Format definition, general concept and glossary:

  `https://seascope.oceandatalab.com/docs/idf_specifications_1.2.pdf`

Contents

1) Basics

1.1 Starting the application

On Linux

In order to start the viewer, open a terminal, go to the directory where you unpacked the SEAScope release and run the following command:

```
./seascope
```

On macOS

The easiest way to start the viewer is by using Spotlight:

1. Hit CMD+Space to bring up Spotlight
2. Type SEAScope

Choose the SEAScope application (usually the first option)

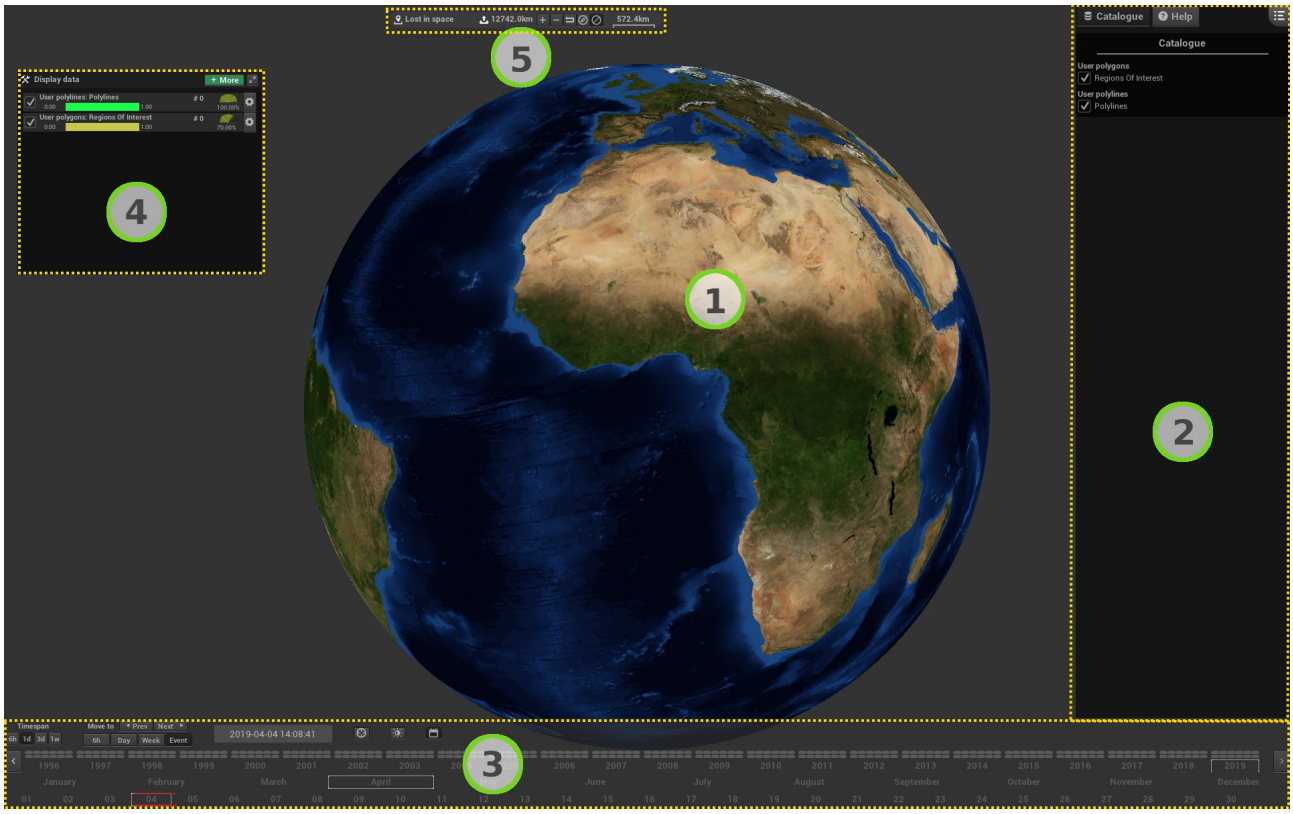
On Windows

In order to start the viewer, go to the directory where you unpacked the SEAScope release and double-click on `launchseascope.bat`.

1.2 Overview of the graphical interface

The interface has been designed to maximize the screen area dedicated to rendering the data and allow users to hide or reduce most of the controls when they don't need them.

The first time you start SEAScope, when you have not added any data yet, your screen should look like this:

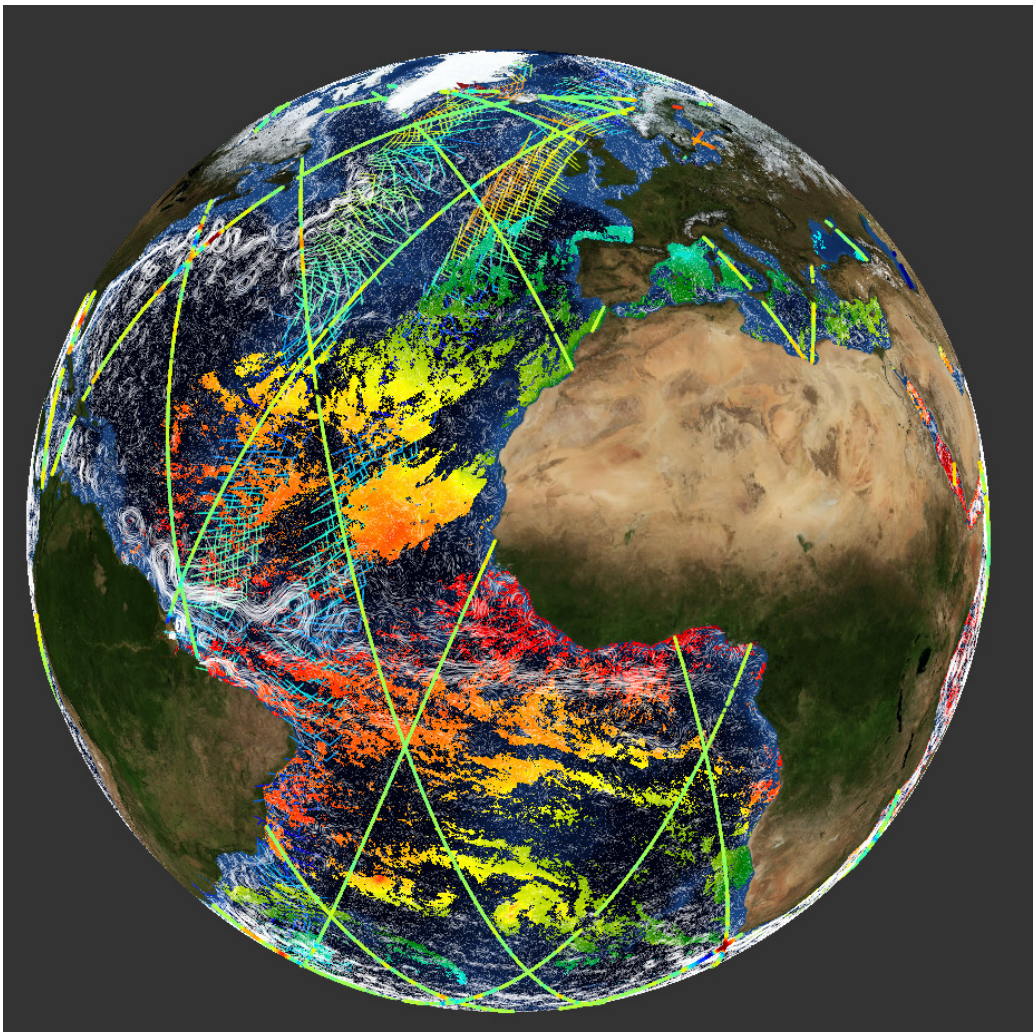


The viewer interface is composed of five main components that will be described in details in the next sections of the document:

1. 3D map
2. Side menu
3. Timeline control
4. Rendering control
5. Camera control

1.2.1 3D map

The 3D map is the canvas on which your data will be drawn and where you will be able to select granules, add custom shapes and make beautiful screenshots:



Navigation

Use your mouse to navigate on the map:

- Place the mouse cursor on the globe, press the left mouse button and move the mouse to rotate the camera. When you have reached the desired point of view, release the left mouse button.
- Use the mouse wheel to zoom in/out. This can also be done using the keyboard: press Page Down for zooming in and Page Up for zooming out.

User shapes

You can create and delete polylines and polygons on the globe to mark interesting phenomena.

In order to create a polyline:

- Keep Ctrl pressed
- Click with the right mouse button to place the points of the polyline on the globe. The viewer will draw segments between subsequent points.
- Release Ctrl
- Press the Enter key to finalize the polyline.

In order to create a polygon:

- Keep Ctrl pressed
- Click with the right mouse button to place the points of the polygon on the globe. The viewer will draw segments between subsequent points.
- Release Ctrl
- Press the Shift and Enter keys at the same time to finalize the polygon.

Press the Escape key to cancel the creation of a polyline or a polygon.
Finalizing a polyline or a polygon selects it by default and opens a dialog that allows users to interact with the created user shape.

Selection

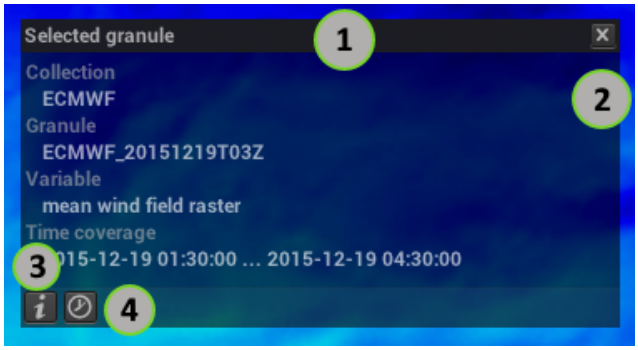
If you click on a polygon, the following contextual menu will be displayed:

1. Use drag'n'drop on the header to move the menu on the screen.
2. Click on the X button to close the menu. Pressing the Escape key has the same effect.
3. Clicking on this button extracts data that intersect the polygon and opens a dialog in the side menu.
4. The trash button deletes the polygon. Pressing the Del key achieves the same effect.
5. Polygon edition on/off. Draw an anchor on each polygon point. An anchor can be moved with the left mouse button and deleted with Del when hovered by the mouse cursor.

If you click on a polyline, the following contextual menu will be displayed:

1. Use drag'n'drop on the header to move the menu on the screen.
2. Click on the X button to close the menu. Pressing the Escape key has the same effect.
3. Clicking on this button extracts data along the polyline and opens a dialog in the side menu.
4. The trash button deletes the polyline. Pressing the Del key achieves the same effect.
5. Polyline edition on/off. Draw an anchor on each polyline point. An anchor can be moved with the left mouse button and deleted with Del when hovered by the mouse cursor.

If you click on a granule, a contextual menu will appear:



1. Use drag'n'drop on the header to move the menu on the screen.
2. Click on the X button to close the menu. Pressing the Escape key has the same effect.
3. Clicking on this button displays information about the granule in the side menu.
4. The clock button centers the timeline on the granule's datetime.

Developer shortkeys

These keyboard shortcuts are only useful for developers but in case you activate them by mistake, here is a description of what they do:

Fill mode: press «F» to return to the default display mode

Point mode: press «P» to display only points on the map

Line mode: press «L» to display segments on the map

Picking mask: Press «O» to toggle the display of picking masks

Memory info: Press «C» to print information about allocated memory. Only works if SEAScope was started with the –debug option.

Brittany: press «B» to point the camera on Brittany

1.2.2 Side menu

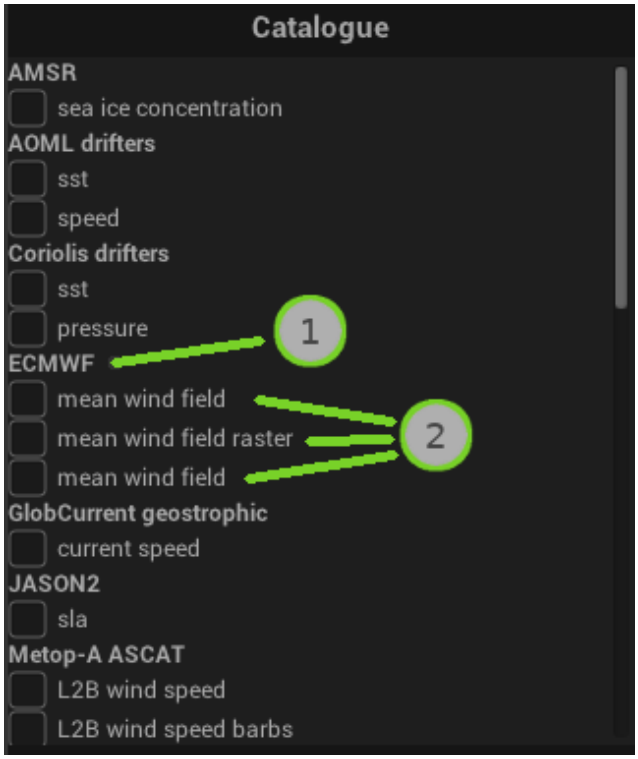
The side menu is where most textual information and dialogs will be displayed.

Topbar

The top bar contains buttons to configure the visual aspect of the menu.

1. This button toggles the display of the collections catalogue.
2. The Help button opens a popup providing ways to get help about the SEAScope viewer.
3. The menu anchor has two roles: toggling the side menu (left click on the anchor) and switching the side of the screen on which the side menu is displayed (Shift key and left click on the anchor at the same time).

Catalogue



The catalogue lists available variables for each collection.

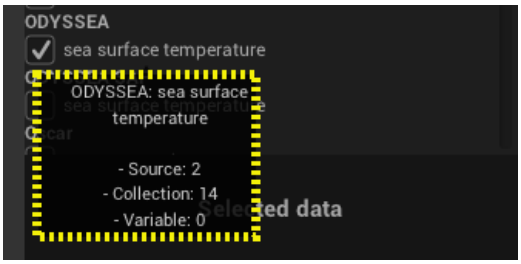
1. Label of the collection.
2. Variables of the collection that you can display on the 3D map.

Selecting a variable adds it to the rendering control and includes the datetime of the collection's granules in the timeline.

Maintaining the Shift key pressed while clicking on a checkbox unselect all variables but the one associated with the checkbox.

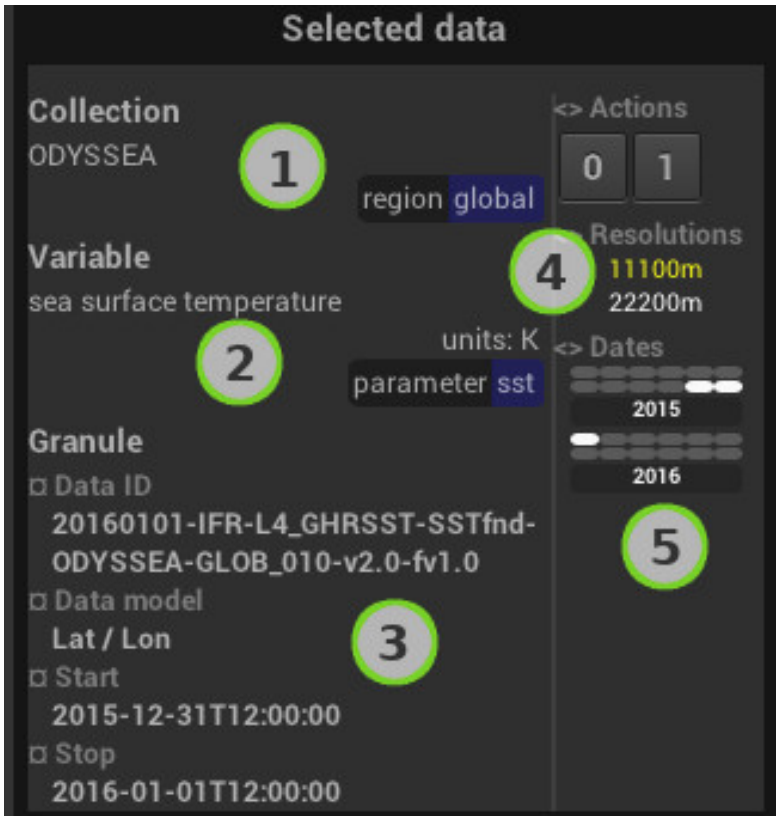
Place the mouse on a variable label to display a tooltip which gives information about the way SEAScope identifies the variable internally.

This information may be required if you want to write scripts that change the way a variable is displayed in the viewer.



Granule information

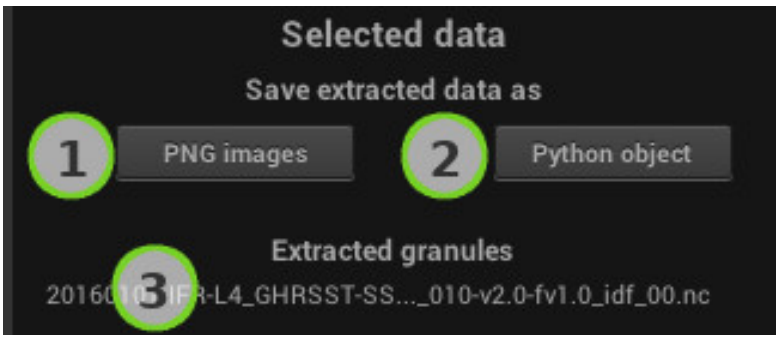
If you click on the information button of the contextual menu which pops after selecting a granule on the 3D map, you should see a new element in the side menu:



1. Collection details.
2. Variable details.
3. Granule details extracted from the attributes of the NetCDF files (cf. IDF specifications).
4. Available resolutions for the granule. The resolution written in yellow is the one used by the viewer when you clicked on the information button.
5. Years and months when the collection has granules. Clicking on a white dot centers the timeline on the associated month and year.

Extraction and transect dialogs

The following dialog appears when you click on the Extract button on the contextual menu which pops after selecting a polyline or a polygon on the 3D map.



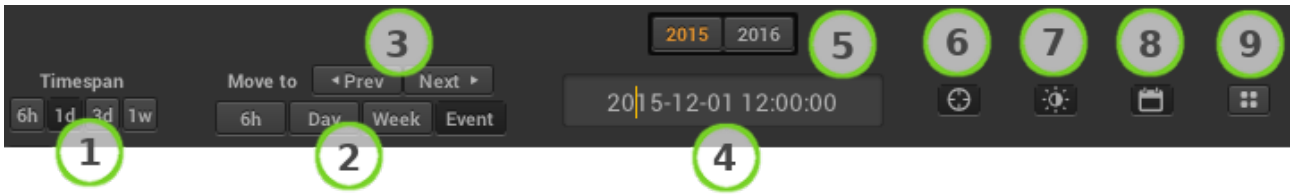
1. Save extracted data as PNG images.
2. Save extracted data as Python objects (pickle format).
3. List of granules from which data have been extracted.

The 1. and 2. buttons will only appear if the SEAScope processor option is enabled in the configuration file (`enabled = true` under the `[processor]` section).
Note that the 1. and 2. buttons will do nothing if the Python processing server is not running (cf. Advanced functionalities).

1.2.3 Timeline control

The timeline control lets you configure how the viewer chooses the granules to display.

Minimal view



1. Timespan: A timespan is a (potentially asymmetric) repartition of time around a reference datetime. The viewer applies the selected timespan to the current datetime to compute the time window granules must intersect in order to be displayed.

2. Timestep: A timestep is an amount of time by which the current datetime is decreased (resp. increased) when the user clicks on the Prev (resp. Next) button.

3. Click on the Prev/Next buttons to navigate through time by fixed amounts of time.

4. Datetime the timeline control considers as «current».

5. The timeline provides suggestions for dates while you edit the current datetime. Suggestions written in orange are years/months/days for which there is at least one granule of each selected collection available (temporal colocation).

6. «Nearest» mode: when this button is pressed, the viewer centers the timeline on the datetime of the closest granule.

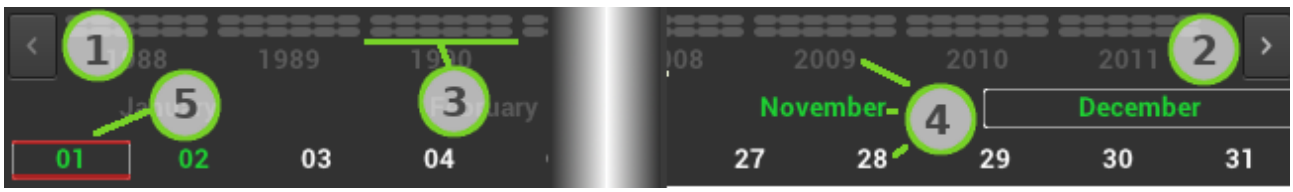
7. «Nearest in day» mode: when this button is pressed, the viewer centers the timeline on the datetime of the granule which is closest to 12 o'clock while staying in the selected day.

8. This button toggles the calendars view.

9. This button only appears when the year of the current datetime is not visible in the calendars view. Clicking on this button forces the calendars view to display a years range containing the current datetime.

Calendars view

The calendars view is an extension of the timeline control which provides an easy way to select a date. It also displays information about data availability.



1. Display previous years range.

2. Display next years range.

3. Monthly data availability: a gray dot means that the associated month does not contain any suitable data, a white dot means that it contains data for at least one of the selected collections and a green dot means that there is at least one granule of each selected collection available during the associated month. Clicking on a white or green dot centers the timeline on the associated month.
4. Year, month and day selection buttons. These buttons use the same color scheme as the monthly availability dots.
5. The red lines show the time window granules must overlap for the viewer to display them on the 3D map.

1.2.4 Rendering control

Condensed view

The rendering menu is a draggable window which contains one rendering control for each variable selected in the catalogue.

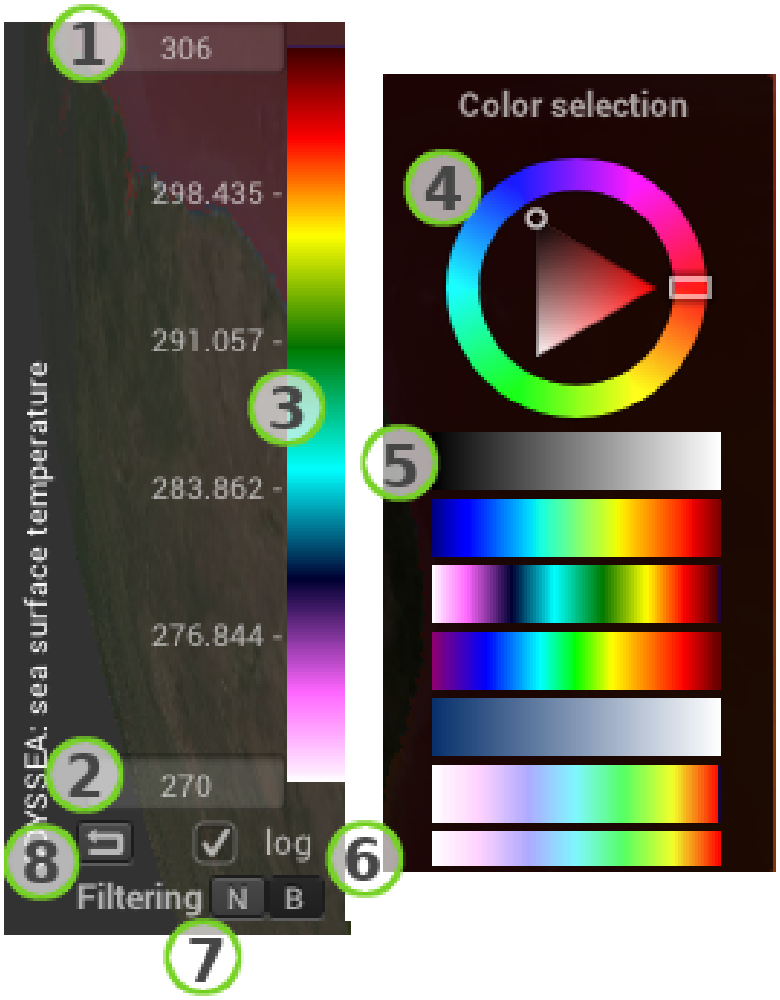
1. Shortcut to display the catalogue in the sidemenu
2. Minimize the rendering menu so that it looks like this: 
3. Visibility checkbox: if this box is unchecked, this associated data will be ignored when searching for granules to display.
4. Value associated with the first color of the colormap (min). Using the mouse wheel when the cursor is over this field changes the value.
5. Value associated with the last color of the colormap (max). Using the mouse wheel when the cursor is over this field changes the value.

6. Units of the variable.
7. Number of granules that contain the variable and are currently displayed on the 3D map.
8. The LOG flag is only visible if the mapping between data values and colors uses a logarithmic scale.
9. Opacity gauge. Using the mouse wheel when the cursor is over this field changes the value.
10. Clicking on this button toggles the advanced rendering control.

Note that you can reorder the list elements using drag'n'drop: the order of the list is also the order in which data are drawn on the 3D map. It means that if a variable «A» is above another variable «B» in the list, data from variable «A» will be displayed on top of data from variable «B» on the map.

Advanced rendering control

The advanced rendering menu provides a finer control over the rendering parameters of a given variable.



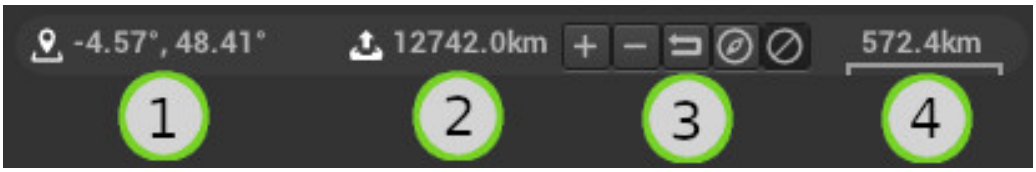
1. Value associated with the last color of the colormap (max). Using the mouse wheel when the cursor is over this field changes the value. If you edit the value yourself you need to press Enter to validate the new value.
2. Value associated with the first color of the colormap (min). Using the mouse wheel when the cursor is over this field changes the value. If you edit the value yourself you need to press Enter to validate the new value.
3. Colormap: clicking on the upper half changes the max value while clicking on the lower half changes the min value. Using drag'n'drop changes the value depending on mouse movements. Right-clicking on the colormap opens the color selection panel.
4. The color wheel can be used to select a uniform color.
5. Select the colormap to use for the selected variable.
6. Check this box if you want to map the data values range to the colormap using a logarithmic scale.

7. If the «N» button is selected, pixels will be displayed using the color of the nearest data value. If the «B» button is selected, the color of the pixel will be the result a bilinear interpolation filter applied on the surrounding data values.

8. Reset button which reinitialize the rendering parameters using the values defined in the configuration file of the collection.

1.2.5 Camera control

The camera control provides information about the parameters of the camera.



1. Location of the mouse cursor in terms of longitude and latitude (in decimal degrees).

2. Distance between the camera and the surface of the globe (in meters or kilometers depending on the altitude).

3. Zoom buttons. Use the «+» button to zoom in, the «-» button to zoom out and the reset button to reinitialize the camera parameters. The «compass» button realigns the vertical axis of the globe with the poles. The toggled button locks the camera movement with longitude / latitude axis. Unlock it to have free movement.

4. This gives an order of magnitude of the scale between pixels and geographical distances.

2) Application configuration

2.1 Finding the configuration file

SEAScope settings are set in a configuration file in INI format, each line containing either a section name between square brackets or a setting defined using the "key = value" syntax.This file is called "config.ini" and is located at the root of the SEAScope workspace:

- on Linux, where you extracted the seascope-viewer-20190628.tar.gz archive
- on macOS, at /SEAScope-workspace
- on Windows, where you extracted the seascope-viewer-20190628.zip archive

If this file does not exist, it will be generated automatically (with default settings) when you run SEAScope.
If you edit this file and make a mistake which prevents you from executing SEAScope, simply remove the config.ini file and start SEAScope again to regenerate a default config file.

2.2 Template

```
[paths]
plugins = value
colormaps = value
operators = value
data = value
state = value
index = value

[rendering]
glRenderer = value
glContextMajor = value
glContextMinor = value
windowWidth = value
windowHeight = value
antialiasing = value
antialiasingType = value
antialiasingFactor = value
fullscreen = value
vsync = value
minGranuleScreenEstate = value

[timeline]
minYear = value
maxYear = value
timespans = value
timesteps = value
defaultTimespan = value
defaultTimestep = value
useNearestMode = value
```

```
useNearestInDayMode = value

[misc]
worldMap = value
skipIndexCheck = value

[processor]
enabled = value
address = value
port = value
```

2.3 List of settings

There are five sections of settings in the configuration file: paths, rendering, timeline, misc and processor.
SEAScope will not work properly and might crash if there are undefined settings, so please make sure that your config file has all the settings described below, even if the associated value stays empty.

2.3.1 [paths] section

plugins

Path of the directory which contains the SEAScope plugins. Edit only if you move the SEAScope workspace to another location.

colormaps

Path of the directory which contains the colormaps available in the viewer.
A colormap is a text file which contains a list of RGB triplets and its filename must end with the .rgb extension. Each line of the colormap file must contain exactly three space-separated integer values between 0 and 255.

operators

Path of the directory which contains the definition files for operators. Not implemented yet. Edit only if you move the SEAScope workspace to another location.

data

Path of the directory which contains the data that will be available in the application, organized following a collection/granule hierarchy.

```
data
├── collection1
│   ├── config.ini
│   ├── granule1
│   │   ├── granule1_idf_00.nc
│   │   ├── granule1_idf_01.nc
│   │   └── ...
│   └── granule2
│       ├── granule2_idf_00.nc
│       ├── granule2_idf_01.nc
│       └── ...
└── collection2
    └── config.ini
```

```
granule3
    granule3_idf_00.nc
    granule3_idf_01.nc
    ...
```

state

Path of the file where the state of the application will be stored. Edit only if you move the SEAScope workspace to another location.

index

Path of the file where the application will index the contents of the data directory.
In order to provide users with a way to explore data smoothly, the viewer needs to list the available IDF files and to extract some metadata from each of them. Doing so dynamically would not yield satisfactory performances since the time required to crawl the data directory would grow as new granules are added.
The crawling step may take a long time therefore the listing and extracted metadata are stored in an index file. When the viewer starts, it checks the availability of the index file:

- if it exists, it loads its contents and skips the crawling step completely.
- otherwise it performs the index step and creates the index file.

You have to be patient the first time you start the viewer as it will have to build the index. Each time you add/remove/change files in the data directory, SEAScope will detect a change and ask you if you want to build a new index.
You can use the `skipIndexCheck` entry (`[misc]` section) to prevent SEAScope from asking you about index updates when you are modifying files that will have no effect on the index (changing a label in the config file of a collection for example). In this case SEAScope will not rebuild the index unless you delete it or disable the `skipIndexCheck` entry.

2.3.2 [rendering] section

glRenderer

Identifier of the OpenGL renderer used by the viewer.
This is a developer setting, do not change it unless you know what you are doing.

glContextMajor

Major version number of the GL context to create.
This is a developer setting, do not change it unless you know what you are doing.

glContextMinor

Minor version number of the GL context to create.
This is a developer setting, do not change it unless you know what you are doing.

windowWidth

Width in pixels of the viewer window. Should be an integer value.
Ignored if fullscreen is enabled.

windowHeight

Height in pixels of the viewer window. Should be an integer value.
Ignored if fullscreen is enabled.

antialiasing

Enable/disable antialiasing filter. Should be a boolean value (true|false)

antialiasingType

Identifier of the antialiasing filter to use. Leave it empty or use the default value (msaa)

antialiasingFactor

Parameter for the msaa antialiasing filter. Should be an integer value.Higher values generate smoother images but require a more powerful GPU.

fullscreen

Open the viewer in fullscreen mode. Should be a boolean value (true|false)

vsync

Enable/disable vertical synchronization. Should be a boolean value (true|false)Vertical synchronization prevents the GPU from generating output at a frequency higher than the screen refresh rate. We recommend to enable this option unless you are performing a benchmark.

minGranuleScreenEstate

Minimal value the (granule area / screen area) ratio must reach for the viewer to actually load data.Some high resolution granules may only be displayed as as bunch of pixels at low zoom levels. This setting tells the viewer to skip data loading in these cases, thus saving processing power and improving fluidity.

2.3.3 [timeline] section

minYear

Lower bound (inclusive) for the years list available in the timeline.

maxYear

Upper bound (exclusive) for the years list available in the timeline.

timespans

This setting defines a list of timespans the user will be able to choose from in the viewer.A timespan is a (potentially asymmetric) distribution of time around a reference datetime.
The viewer applies the selected timespan to the current datetime to compute the time window granules must intersect in order to be displayed.
Timespans are displayed as buttons in the timeline control.
You must provide four elements to define a timespan:

- a label for the button (keep it short, 3-4 characters)
- a tooltip for the button
- the size, in seconds, of the left part (before current datetime) of the time window
- the size, in seconds, of the right part (after current datetime) of the time window

These four elements must be separated by pipe characters. Timespan definitions must be separated by commas.
For example, in order to create two buttons:

- one for a [T-3h, T+3h] time window
- one for a [T-30min, T+30min] time window

The value associated to the `timespans` key in the configuration file would be:

```
6h|6hours timespan|10800|10800,1h|1hour timespan|1800|1800,
```

timesteps

This setting defines a list of timesteps the user will be able to choose from in the viewer.
Timesteps are displayed as buttons in the timeline control.
A timestep is a number of seconds by which the current datetime is decreased (resp. increased) when the user clicks on the Prev (resp. Next) button.
You must provide three elements to define a timestep:

- a label for the button (keep it short, 3-4 characters)
- a tooltip for the button
- the size of the step (in seconds)

These three elements must be separated by pipe characters. Timestep definitions must be separated by commas.
Note that by defining a timestep of zero second, you will create a button which has a special behavior: if it is selected, instead of decreasing/increasing the current datetime by a fixed number of seconds, it will tell the viewer to use the datetime of the closest granule in the past/future as current datetime.
For example, in order to create three buttons:

- one for a one-hour step
- one for a three-days step
- one for the nearest granule behavior

The value associated to the `timesteps` key in the configuration file would be:

```
1h|Move by one hour|3600,1d|Move by one day|86400,Event|Move to closest data|0,
```

defaultTimespan

Label of the timespan button which is selected by default in the viewer.

defaultTimestep

Label of the timestep button which is selected by default in the viewer.

useNearestMode

Default state (enabled|disabled) of the the «nearest» mode button in the timeline. Should be a boolean value (true|false).
When the «nearest» mode button is selected, the viewer looks for the closest datetime for which a granule is available and use it as current datetime.

useNearestInDayMode

Default state (enabled|disabled) of the «nearest in day» mode button in the timeline. Should be a boolean value (true|false).
When the «nearest in day» mode button is selected, the viewer looks for the granule whose datetime is within the selected day and is the closest to 12:00:00 to use it as current datetime.

2.3.4 [misc] section

worldMap

SEAScope has three predefined worldmap backgrounds:

- highres: The worlmap with the most details, close to the size limit of current graphics cards (not provided in the SEAScope package).
- lowres: A worldmap with coarse details but very light in memory
- black: A black worldmap

If you want to use your own worldmap, you can use a path to your PNG file here.

skipIndexCheck

Skip the index rebuild if SEAScope detects a change.

2.3.5 [processor] section

enabled

Show the processing options in the SEAScope menus. Should be a boolean value (true|false)

address

IP address of the SEAScope processor. Use 127.0.0.1 if the processor runs on the same machine as the SEAScope viewer.

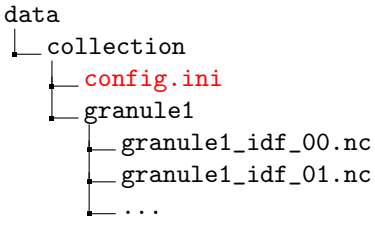
port

Port that the SEAScope processor listens to (default value: 53450)

3) Collections configuration

3.1 Configuration file

Each collection must be defined using a configuration file in INI format, each line containing either a section name between square brackets or a setting defined using the "key = value" syntax.
This file is called `config.ini` and is located in the directory of the collection, at the same level as the directories that contain the IDF files.



3.2 Template

```
[general]
label = value
xSeamless = value
ySeamless = value
mustBeCurrent = value
tags = value
variables = value
defaulVariable = value
NEWSAligned = value

[VARID_N]
label = value
units = value
tags = value
fields = value
defaultRendering = value
logscale = value
min = value
max = value
opacity = value
zindex = value
particlesCount = value
streamlineLength = value
colormap = value
color = value
filterMode = value
```

3.3 List of settings

The configuration file has a mandatory section called `[general]` and one additional section for each variable that should appear in the data catalogue of the viewer.
An incomplete configuration file can cause errors and prevent SEAScope from completing the indexation process, so please make sure that your config files have all the settings described below, even if the associated value remains empty.

3.3.1 [general] section

label

Name used in the viewer to designate the collection.

xSeamless

Tell the viewer that the granules of the collection have a 360° longitudinal cover. Should be a boolean value (true|false).

ySeamless

Tell the viewer that the granules of the collection have a 180° latitudinal cover. Should be a boolean value (true|false).

mustBeCurrent

This setting controls the viewer behavior when it searches for granules to display and uses temporal constraints to filter the results. Should be a boolean value (true|false).
When this setting is set to true, the viewer will ignore granules whose time range does not contain the current datetime. It is the recommended behavior for collections whose granules cover the whole globe.
When it is set to false, the viewer displays the granules of the collection as long as their time range intersects the current time window (cf. timespans in the list of settings for the application configuration file). Use this behavior for collections of granules with a smaller spatial footprint or a sparse temporal coverage.

tags

List of tags associated with the collection.
Tags are shown in the viewer when the detailed information of a granule is displayed.
Tags are defined with a key:value syntax. Tag definitions must be separated by commas.

variables

Comma-separated list of identifiers for the variables which can be extracted from the granules of the collection and be displayed in the viewer.
Identifiers must be unique within a collection and can be chosen arbitrarily. In addition to the `[general]` section, the configuration file of the collection must contain one section per variable identifier in this list.

defaultVariable

Identifier of one variable of the collection. Not used at the moment but must be set anyway.

NEWSAligned

Tell the viewer that the axes of the data matrix are aligned with the South-North and West-East axes.

3.3.2 Variable sections

label

Name used in the viewer to designate the variable.

units

Units of the values contained in the variable.

tags

List of tags associated with the variable.
Tags are shown in the viewer when the detailed information of a granule is displayed.
Tags are defined with a key:value syntax. Tag definitions must be separated by commas.

fields

Comma-separated list of NetCDF variables to extract from the IDF files in order to compute the SEAScope variable.
In some cases it may be necessary to extract more than one NetCDF variable to compute a SEAScope variable. For example, displaying vectorfield as arrows/streamlines requires u and v components of the vectors.
Another example would be a variable composed from three fields which are interpreted as R, G and B channels to create an image.

defaultRendering

Rendering method used to display the variable. Must be one of the following:

- RASTER
- ARROWS
- BARBS
- TRAJECTORIES
- STREAMLINES
- RAWRGB

logscale

Use a logarithmic scale for the palette. Should be a boolean value (true|false).

min

Minimum value for the color palette applied to the representation of the variable. Should be a float value.

max

Maximum value for the color palette applied to the representation of the variable. Should be a float value.

opacity

Opacity of the representation of the variable. Should be a float value between 0.0 (transparent) and 1.0 (completely opaque).

zindex

Z-indices are used to define the order in which the representations are drawn on the screen. Should be a float value.
Representations of a variable with a low z-index are drawn first and may be hidden by the representations of another variable with a higher z-index if they overlap.
You can choose an arbitrary range to define the z-indices, but z-indices must be unique within a collection AND between collections.

particlesCount

Size of the pool of animated particles. Should be an integer value.
This value controls the density of the particles when using the streamlines rendering method. Increasing the density adds a toll on the GPU and may significantly reduce fluidity.

streamlineLength

Number of segments of a streamline. Should be an integer value.

colormap

Identifier of the colormap applied to the representation of the variable.
A colormap identifier is only valid if there is a file in the colormaps directory whose name is the identifier suffixed by the .rgb extension.
For example, you can use:

```
colormap = rainbow
```

only if there is a file named rainbow.rgb in the colormaps directory.
If both a uniform color and a colormap are defined for a variable, the colormap will be used.

color

Uniform RGB color applied to the representation of the variable. Should be a comma-separated list of three integer values in the [0, 255] range.
If both a uniform color and a colormap are defined for a variable, the colormap will be used.

filterMode

Method used by SEAScope to find the value associated with a pixel. Either NEAREST to use the nearest data value or BILINEAR to use the result of a bilinear interpolation applied on surrounding values.

4) Advanced functionalities

4.1 Data extraction

The viewer provides buttons to request the serialization of extracted data to PNG images or Python objects (cf. Side menu – extraction and transect dialogs).
The serialization is not handled by the viewer and requires a program which is part of the SEAScope Python package.
In order to enable the serialization capabilities, you have to:

- enable the processor in the SEAScope configuration file («enabled» entry set to «true» under the [processor] section)
- start the Python processing server by opening a terminal and running the following command: `seascope-processor`

4.2 Remote control

It is possible to control the viewer using Python scripts if the SEAScope Python package is installed.
Using the -l option when starting the viewer, you can define the IP address and port SEAScope will listen to for incoming remote commands. It means that you can run the viewer on a machine and control it from another machine as long as they can reach each other on the network.
See tutorials on the SEAScope website for more information about this functionality: `https://seascope.oceandatalab.com/tutorials.html`