



COP Viewer

COP Viewer INSTALLATION GUIDE

Pedro Montero, Garbiñe Ayensa, Silvia Allen-Perkins, Alberto Gómez



Co-funded by the European Union Civil
Protection

ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective “Enhancing prevention and protection from the effects of maritime disasters” under priority 1: “Developing response capacity for marine pollution”.

DISCLAIMER

The content of this document represents the views of the author only and is his/her sole responsibility; it cannot be considered to reflect the views of the European Commission and/or the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) or any other body of the European Union. The European Commission and the DG-ECHO is not responsible for any use that may be made of the information it contains.



Project Acronym	MANIFESTS
Project Full Title	MANaging risks and Impacts From Evaporating and gaseous Substances To population Safety
Gant Agreement Nr.	101004912
Project Website	https://www.manifests-project.eu/

Deliverable Nr.	D5.2 MANIFESTS DSS Installation guide
Status (Final/Draft/Revised)	Draft
Work Package	5
Task Number	5.4
Responsible Institute	INTECMAR
Author/s	P. Montero, G. Ayensa, S. Allen-Perkins, A. Gómez
Recommended Citation	
Dissemination Level	

Document History			
Version	Date	Modification Introduced	
		Modification Reason	Modified by
1.0	31/01/2023		

Content

1. Background.....	6
2. Administration guide.....	7
2.1. Map configuration (basemap.js)	7
2.2. Default application settings (cfg.js)	7
2.3. Layer configuration (eg test.xml)	8
2.4. Translation files into different languages.....	9
3. Installation guide.....	10

1. Background

In the event of a maritime accident involving HNS, maritime authorities must take numerous decisions to organize the best response strategy, i.e., one that minimizes risks to human health (including incident response teams, crew members and coastal communities), to the marine environment, for maritime safety and for socio-economic activities and facilities. While many key decisions and considerations are prescribed in national or regional contingency plans, operational response activities will generally need ongoing adjustment or review to reflect the most recent information available as the contamination event evolves. In such a rapidly changing situation, an efficient exchange of information between competent decision-making authorities and response teams on the ground can greatly facilitate both decision-making processes and organizational processes.

The purpose of work package 5 of the Manifests project (Manifests decision support system) is to develop an efficient information system that helps (1) decision makers understand the situation at stake and its likely evolution in the coming hours and days; (2) identify the population, ecosystems and socio-economic assets at risk and (3) share useful information with response teams deployed at sea, in the air or on the coast.

Building on the experience gained and development carried out during the previous HNS-MS and MARINER projects, the MANIFESTS decision support system (DSS) will integrate several services, including the DSS Common Operational Picture (COPtool).

This COPtool refers to a system designed so that during a contingency, the exchange of information that occurs between the maritime authorities and the different response teams (sea, coast, air) is carried out in the most efficient way possible, ensuring that all actors involved in the crisis committee and response teams can access the same data. These can be standard reports (such as the Standard Pollution Observation Report of the Bonn Agreement), images, videos and any other georeferenced data collected by response teams, as well as satellite observations, model simulation results, exclusion areas, location of response media, requests for new response actions shared by the crisis.

The COPtool documentation is composed by:

- Installation guides (this document).
- User Manual
- DSS Implementation report.

2. Administration guide

The viewer has several configuration files adjustable to the needs of the users who manage them:

2.1. Map configuration (basemap.js)

File .js in which you define, in JSON format, the different layers or services that will be used as basemap in the viewer. The properties available in each of them are the following:

- **name.** The name by which the layer will be recognized and displayed in the layer selector of the viewer. Example: 'IGN-BASE'.
- **URL.** Address of the service to load the map.
- **layers.** Identifiers of the layers to be recovered.
- **CRS.** Coordinate Reference System for this map.
- **version.** Version of the layer.
- **attribution.** Attribution of the map.
- **default.** Possible values are *True* or *False* and indicates whether the layer is enabled by default when you start the map. Only one of the basemaps can be set to *True*.

2.2. Default application settings (cfg.js)

File .js in which the default configuration of the application is defined, in JSON format. The available properties are as follows:

- **profile.** By default, the profile will be 'default'.
- **proxy.** This field indicates the file where the *proxy* used by the application is located.
- **mapProxy.** Possible values are *True* or *False*.
- **layerProxy.** Possible values are *True* or *False*.
- **encoder.** This field indicates the file where the *encoder* used in the application is located.
- **defaultLocale.** Default language of the application.

- **availablePremises.** Array with the different languages available in the application.
- **CRS.** Coordinate Reference System used by Leaflet by default in the application.
- **initialZoom.** The zoom or distance with which the viewer is loaded by default.
- **minZoom.** Maximum distance to which we can move away in the viewer view.
- **maxZoom.** Maximum distance to which we can approach in the viewer view.
- **initialCenter.** The coordinates of the point where the view will focus when you first load your view.
- **ignoreData.** Possible values are *True* or *False*. Indicates, if *True*, that there is no initial data to load.
- **layersXML.** List of different XML from which the cartography layers will be retrieved.
- **toponymURL.** URL of the toponymy service.
- **toponymField.** Name of the toponymy field.
- **toponymName.** Name of toponymy.
- **toponymFeatName.** The name of the toponymy feature.
- **bitacoraUrl.** Bitacora URL.
- **episodes.** URL of the episodes.

2.3. Layer configuration (eg test.xml)

Files .xml in which you can define the different layers that can be loaded into the viewer, as well as add new ones. To do this, these files must be referenced in the **cfg.js** configuration file, specifically in **layersXML**.

The elements that make up these XML are as follows:

- **EpisodeId.** The ID of the episode to which the layers belong.
- **BeginTime.** Start date.
- **EndTime.** End date.
- **LatLonBoundingBox.** Coordinates that delimit the maximum surface area to be covered by the layers to be loaded.
- **Panel.** A panel under which a series of layer groups or layers are grouped. It has a **label** attribute, which is the name (or key to translate using the translation files) that it is in the panel.

- **LayerGroup.** A group of layers under which different layers are grouped. It has a `group` attribute responsible for naming that group, in the same way as the **label** attribute in a **Panel** element.
 - **Layer.** Each of the layers that can be loaded into the application viewer.
 - **Id.** The ID of the layer.
 - **Name.** The name of the layer.
 - **Title.** The title of the layer displayed in the layer collector in the viewer.
 - **LatLonBoundingBox.** Coordinates that delimit the maximum surface area covered by the layer.
 - **Style.** The style of the layer.
 - **Filter.** The filter of the layer.
 - **LayerType.** The type of the layer.
 - **Url.** URL from which the layer is retrieved.

2.4. Translation files into different languages

Files .js, one for each of the languages available in the application, both for the "main" application and for the Log, which serve as a kind of dictionary, in which a certain key is assigned a value, which corresponds to the text we want to use in the application in the language that interests us. Example:

```
...  
'campanha': 'Campaign',  
'bathymetry': 'Bathymetry (m)',  
...
```

To change the translation of a certain text is as simple as modifying the second field. If you want to add a new translation, simply add a new key-value pair in each of the languages.

3. Installation guide

To deploy the project on the server and ensure its proper functioning, it is only necessary to have **php** (version 7.3 or higher) installed on the server. In addition, it is necessary to have the **php cURL** library (usually already is installed by default; if this is not the case, it will be necessary to install it).

If the above requirements are met, it is sufficient to decompress or clone the source code of the project (if obtained through a ZIP or from a Git repository, respectively), and paste the source code into the server.

IMPORTANT: The source code must be copied to the server maintaining the exact structure of the application to avoid problems and communication errors between the components of the application (such as Log).