Marine user guide

Release v1.0

IPG

Jul 28, 2020

CONTENTS:

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

TWO

INTRODUCTION

This project contains the necessary code to produce the data summaries that are included in the Marine User Guide. These help document the status of the marine in situ data in the CDS after every new data release. The marine data available in the CDS is the result of a series of data releases that are stored in the marine data file system in different directories. This project uses the data in the marine file system, rather than accessing the CDS data.

Additionally, the tools employed to create the individual source deck reports are also available in this project. These can be created for a single data release or for the combination of releases included in a Marine User Guide version.

This manual has two main independent sections dedicated to the Marine User Guide and to the individual source deck reports:

- *Marine User Guide*
- *Individual source-deck reports*

Every new data release can potentially be created with a different version of the marine processing software. The current version of this project is compatible with the glamod-marine-processing code up to version v1.1 the HEAD of the repository as of August 2020.

The configuration files needed to run this project are maintained in a separate git repository (https://git.noc.ac.uk/iregon/glamod-marine-config.git), within this repository.

CHAPTER

THREE

TOOL SET-UP

3.1 Code set up

The repository was tagged to v1.1 after C3S data release_2.0 and Marine User Guide v4. However changes were added to the project following the job scheduler change in CEDA-JASMIN. These changes have not been tagged yet, to clone the latest available:

```
git clone git@git.noc.ac.uk:iregon/marine-user-guide.git
```

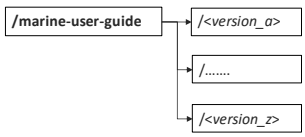
Build the python environment using the requirements.txt file in marine-user-guide/env. This step system dependent. The following code block described the steps to follow in CEDA JASMIN, using the Jaspy toolkit.

```
cd marine-user-guide/env
module load jaspy/3.7
virtualenv --system-site-packages mug_env
source mug_env/bin/activate
pip install -r requirements.txt
```

3.2 Data directory setup

The data the tools in this project use and the products created are stored in the marine-user-guide data directory[1]. This directory does not contain the actual data files, but links to the files in the data releases' directories. This approach greatly simplifies the configuration of the different scripts and is followed even if a given Marine User Guide version is made up of a single data release.

The marine-user-guide data directory is then split in directories to host subsequent versions of the Marine User Guide.



This general directory needs to be created before starting using the tool.

```
cd <parent_data_directory>
mkdir marine-user-guide
```

[1] When producing data summaries and figures of individual source-decks of a single release, the data is accessed directly from the release data directory.

3.3 Paths setup

Edit file marine-user-guide/setpaths.sh and modify as needed the following fields:

- code_directory: parent path of the repository installation.
- data_directory: parent path to the data release directories.
- mug_code_directory: marine user guide code directory installation.
- mug_data_directory: marine user guide data directory path.

CHAPTER

FOUR

MARINE USER GUIDE

Every C3S Marine User Guide version includes a series of figures that describe the marine in situ data holdings in the CDS. The following sections explain how these figures are created for every new version of the Marine User Guide.

4.1 Initializing a new user guide

Every new version of the Marine User Guide (MUG) needs to be initialised in the tools' data directory as shown in the figure.

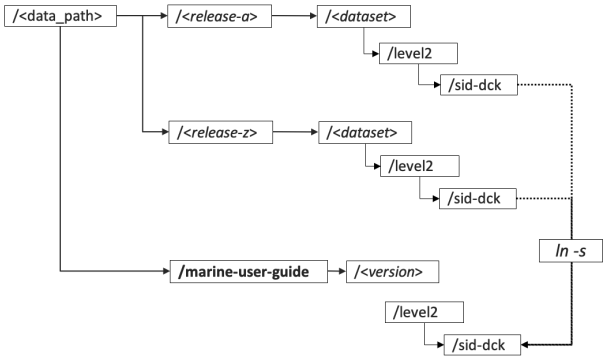


Fig. 1: Marine User Guide data directory and its relation to the individual data releases' directories.

These steps initialize a new version:

1. Create the data configuration file (*mug_file*, *Marine user guide data configuration file*) by merging the level2 configuration files of the different data releases included in the new version (*level2 file format*).

```
source marine-user-guide/setpaths.sh
source marine-user-guide/setenv.sh
python marine-user-guide/init_version/init_config.py
```

2. Use the sid-dck keys of the *mug_config* file to create a simple ascii file with the full list of source-deck IDs of the release merge (*mug_list*).

3. Create the directory tree for the version in the in the marine-user-guide data directory.

```
source marine-user-guide/setpaths.sh
source marine-user-guide/setenv.sh
python /marine-user-guide/init_version/create_version_dir_tree.py mug_path␣
↪version mug_config
```

where:

- mug_path: full path to the marine-user-guide data directory
- version: tag to use for Marine User Guide version
- mug_config: path to *mug_config* file

4. Populate it with a view of the merged data releases: rather than copying all the files, this is done by linking the corresponding files from the releases' directories to the marine-user-guide data directory. Data linked is the level2 data files and level1a and level1c quicklook json files.

A launcher bash script configures a SLURM job for each *sid-dck* data partition and logs to level2/log/*sid-dck*/merge_release_data.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/init_version/merge_release_data.slurm version mug_config␣
↪mug_list
```

where:

- version: tag to use for Marine User Guide version
- mug_config: path to *mug_config* file
- mug_list: path to *mug_list* file.

5. Check that the copies really reflect the merge of the releases. Edit the following script to add the corresponding paths and run. If any does not match, it will prompt an error.

```
./marine-user-guide/init_version/merge_release_data_check.sh
```

4.2 Data summaries

The data summaries are monthly aggregations over all the source-deck ID partitions in the data. These aggregations are on the data counts and observation values and on some relevant quality indicators and are the basis to then create the time series plots and maps included in the MUG.

4.2.1 Monthly grids

Aggregations in monthly lat-lon grids. The CDM table determines what aggregations are applied:

- header table: number of reports per grid cell per month.
- observations tables: number of observations and mean observed_value per grid cell per month.

Each aggregation is stored in an individual netcdf file.

All the aggregations are configured in a common configuration file, monthly_grids.json (*Monthly grids*). The current configuration for the MUG excludes reports not passing all the quality checks. The same tool can be used to produce data summaries with different filter criteria, but modifying the filter values in the configuration file.

A launcher bash script configures a SLURM job for each CDM table and logs to level2/log/*sid-dck*/*config_file-table*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/data_summaries/monthly_grids.slurm version monthly_grids.json
```

where:

- version: tag to use for Marine User Guide version

- config_file: path to the monthly grids configuration file

4.2.2 Monthly time series of selected quality indicators

Monthly time series of quality indicators' value counts aggregated over all the source-deck partitions. These are additionally, split in counts by main platform types (ships and buoys) and include the total number of reports. They are stored in ascii pipe separated files.

The configuration file monthly_qi.json (*Quality indicators time series*), includes very limited parameterization, basically the data paths. The python script only works on the CDM header table quality indicators.

A launcher bash script configures a SLURM job for each quality indicator summarized (currently only report_quality and duplicate_status) and logs to level2/log/*sid-dck*/*config_file-qi*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/data_summaries/monthly_qi.slurm version config_file
```

where:

- version: tag to use for Marine User Guide version
- config_file: path to the monthly grids configuration file

4.3 Figures

The data summaries generated are used to create the maps and time series plots included in the Marine User Guide. The following sections give the necessary directives to create them, with references to the configuration files used.

4.3.1 Number of reports time series plot

- Data summary used: *Monthly time series of selected quality indicators* (report_quality counts file: total number of reports field only)
- Configuration file: nreports_ts_plot.json (*Number of reports time series plot*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/nreports_ts_plot.py nreports_ts_plot.json
```

4.3.2 Duplicate status time series plot

- Data summary used: *Monthly time series of selected quality indicators* (duplicate_status file)
- Configuration file: duplicate_status_ts_plot.json (*Duplicate status time series plot*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/duplicate_status_ts_plot.py duplicate_status_ts_
↪plot.json
```

4.3.3 Report quality time series plot

- Data summary used: *Monthly time series of selected quality indicators* (report_quality file)
- Configuration file: report_quality_ts_plot.json (*Report quality time series plot*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/report_quality_ts_plot.py report_quality_ts_
↪plot.json
```

4.3.4 Number of reports Hovmöller plots

- Data summary used: *Monthly grids* (report counts files: header and observation tables)
- Configuration file: nreports_hovmoller_plot.json (*Number of reports Hovmöller plots*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/nreports_hovmoller_plot.py nreports_hovmoller_
↪plot.json
```

4.3.5 ECV coverage time series plot grid

- Data summary used: *Monthly grids* (report counts files: header and observation tables)
- Configuration file: ecv_coverage_ts_plot_grid.json (*ECV coverage time series plot grid*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/ecv_coverage_ts_plot_grid.py ecv_coverage_ts_
↪plot_grid.json
```

4.3.6 Number of reports and number of months maps

- Data summary used: *Monthly grids* (report counts files: header and observation tables)
- Configuration file: nreports_and_nmonths_maps.json (*Number of reports and number of months maps*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/nreports_and_nmonths_maps.py nreports_and_
↪nmonths_maps.json
```

4.3.7 Mean observed value maps

- Data summary used: *Monthly grids* (mean files: observation tables)
- Configuration file: mean_observed_value_maps.json (*Mean observed value maps*)
- Command:

```
source /marine-user-guide/setpaths.sh
source /marine-user-guide/setenv.sh
python /marine-user-guide/figures/mean_observed_value_maps.py mean_observed_value_
↪maps.json
```

CHAPTER

FIVE

INDIVIDUAL SOURCE-DECK REPORTS

First see *Appendix 2. Individual source-deck reports configuration files* for configuration file and options.

The source_deck_list is a simple ascii file with the list of source ID - deck ID pairs to process by the launcher script.

5.1 Reports on a release merge

To create the individual source-deck reports on a merge of data releases, steps in *Initializing a new user guide* first need to be followed, so that the input data and required directory structure is ready in the marine-user-guide data directory.

5.2 Data data summaries

The data summaries are monthly aggregations of report counts, observation values and additional CDM fields of the individual source-deck's table files.

5.2.1 Monthly grids

Aggregations in monthly lat-lon grids. The CDM table determines what aggregations are applied:

- header table: number of reports per grid cell per month.
- observations tables: number of observations and mean, max and min observed_value per grid cell per month.

Each aggregation is stored in an individual netcdf file.

All the aggregations are configured in a common configuration file. There are currently two configurations that need to be run to create the data summaries needed: one using the full dataset and another one using the optimal dataset (all quality control checks passed).

A launcher bash script configures a SLURM job for each *sid-dck* data partition and each table and logs to *log_dir*/*sid-dck*/*config_file*-*table*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/data_summaries_sd/monthly_grids_sd.slurm log_dir config_file␣
↪source_deck_list
```

where:

- log_dir: the logging directory is assumed to be split in the source-deck data partitions. It is normally */level[1e|2]/log* in the directory where the data is. It needs to be input to the launcher as this script can be run either on the individual release directories or on the marine-user-guide data directories.
- config_file:

  - Full dataset: monthly_grids_sd_all.json (*Monthly grids (full dataset)*)
  - Optimal dataset: monthly_grids_sd_optimal.json (*Monthly grids (optimal dataset)*)
- source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.2.2 Monthly time series of selected quality indicators

Monthly time series of quality indicators' value counts for every *sid-dck* data partition. These are additionally, split in counts by main platform types (ships and buoys) and include the total number of reports. They are stored in ascii pipe separated files.

The configuration file monthly_qi_sd.json (*qi_counts*), includes very limited parameterization, basically the data paths. The python script only works on the CDM header table quality indicators.

A launcher bash script configures a SLURM job for each *sid-dck* data partition and each quality indicator (currently only report_quality and duplicate_status) and logs to *log_dir*/*sid-dck*/*config_file-qi*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/data_summaries_sd/monthly_qi_sd.slurm log_dir monthly_qi_sd.json␣
↪source_deck_list
```

where:

- log_dir: the logging directory is assumed to be split in the source-deck data partitions. It is normally */level[1e|2]/log* in the directory where the data is. It needs to be input to the launcher as this script can be run either on the individual release directories or on the marine-user-guide data directories.
- source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.2.3 Monthly time series with source to C3S IO flow

Collection of monthly time series that describe the main report IO flow, from the initial reports in the source dataset to those finally delivered to C3S for every *sid-dck* data partition.

A launcher bash script configures a SLURM job for each *sid-dck* data partition and logs to *log_dir*/*sid-dck*/*config_file*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/data_summaries_sd/report_io_sd.slurm log_dir report_io_sd.json␣
↪source_deck_list
```

where:

- log_dir: the logging directory is assumed to be split in the source-deck data partitions. It is normally */level[1e|2]/log* in the directory where the data is. It needs to be input to the launcher as this script can be run either on the individual release directories or on the marine-user-guide data directories.
- report_io_sd.json: *Monthly time series with source to C3S IO flow*
- source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.3 Figures

5.3.1 ECV reports time series plots

- Data summary used: *Monthly grids* (counts, header and observation tables)
- Launcher script: configures a SLURM job for each *sid-dck* data partition and logs to *log_dir/sid-dck/*config_file*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/figures_sd/ecv_reports_ts_plot_grid_sd.slurm log_dir config_
↪file source_deck_list
```

where:

  - log_dir: the logging directory is assumed to be split in the source-deck data partitions. It is normally */level[1e|2]/log* in the directory where the data is. It needs to be input to the launcher as this script can be run either on the individual release directories or on the marine-user-guide data directories.
  - config_file: ecv_reports_ts_plot_grid_sd-all.json (*ECV reports time series plots (all data)*), ecv_reports_ts_plot_grid_sd-optimal.json (*ECV reports time series plots (optimal dataset)*)
  - source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.3.2 Observed parameters latitudinal time series

- Data summary used: monthly grids (counts, min, max, counts from observation tables). All data and optimal dataset summaries.
- Launcher script: configures a SLURM job for each *sid-dck* data partition and each *mode* (all data and optimal dataset) logs to *log_dir/sid-dck/*config_file-mode*.*ext*, with *ext* being *ok* or *failed* depending on job termination status.

```
./marine-user-guide/figures_sd/param_lat_bands_ts.slurm log_dir config_file␣
↪source_deck_list
```

where:

  - log_dir: is created by the launcher script if does not exist
  - config_file: *Observed parameters latitudinal time series plot*
  - source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.3.3 Duplicate status time series plot

- Data summary used: duplicate_status quality indicators time series.
- Command:

```
./marine-user-guide/figures_sd/nreports_dup_ts_sd.slurm log_dir config_file␣
↪source_deck_list
```

where:

  - log_dir: is created by the launcher script if does not exist
  - config_file: *Duplicate status time series plot*

  - source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.3.4 Report quality time series plot

- Data summary used: report_quality quality indicators time series.
- Command:

```
./marine-user-guide/figures_sd/nreports_qc_ts_sd.slurm log_dir config_file source_
↪deck_list
```

where:

- log_dir: is created by the launcher script if does not exist
- config_file: *Report quality time series plot*
- source_deck_list: ascii file with a list of the *sid-dck* partitions to process

5.3.5 Monthly time series with source to C3S IO flow

- Data summary used: monthly time series with IO flow
- Command:

```
./marine-user-guide/figures_sd/report_io_sd.slurm log_dir config_file source_deck_
↪list
```

where:

- log_dir: is created by the launcher script if does not exist
- config_file: *Data IO flow plot*
- source_deck_list: ascii file with a list of the *sid-dck* partitions to process

CHAPTER

SIX

APPENDIX 1. MARINE USER GUIDE CONFIGURATION FILES

The configuration files needed to run this project are maintained in a separate git repository (https://git.noc.ac.uk/iregon/glamod-marine-config.git). Every Marine User Guide version has a dedicated directory within this repository. This appendix provides the files (or an extract of them) used to run the fourth version of the Marine User Guide.

6.1 level2 file format

Listing 1: Release_2.0 level2 file extract

```
{
    "001-110": {
        "year_init": 1945,
        "year_end": 1949,
        "exclude": false,
        "params_exclude": []
    },
    "001-116": {
        "year_init": 1945,
        "year_end": 1949,
        "exclude": false,
        "params_exclude": []
    },
    "171-711": {
        "year_init": 1889,
        "year_end": 1899,
        "exclude": false,
        "params_exclude": []
    },
    "params_exclude": ["observations-wbt"],
    "year_init": 1851,
    "year_end": 1949
}
```

Listing 2: r092019 level2 file extract

```
{
     "year_init": 1950,
     "year_end": 2010,
     "params_exclude": [
        "observations-wbt"
    ],
    "001-110": {
```

(continues on next page)

(continued from previous page)

```
        "exclude": false,
        "params_exclude": [],
        "year_end": 1951,
        "year_init": 1950
    },
    "001-116": {
        "exclude": false,
        "params_exclude": [],
        "year_end": 1963,
        "year_init": 1950
    },
    "001-999": {
        "exclude": false,
        "params_exclude": [],
        "year_end": 1969,
        "year_init": 1967
    }
}
```

6.2 Marine user guide data configuration file

Listing 3: Marine User Guide v4 configuration file extract

```
{
    "release_names": [
        "r092019",
        "release_2.0"
    ],
    "dataset_names": {
        "r092019": "ICOADS_R3.0.0T",
        "release_2.0": "ICOADS_R3.0.0T"
    },
    "year_init": 1851,
    "year_end": 2010,
    "params_exclude": [
        "observations-wbt"
    ],
    "001-110": {
        "year_init": {
            "r092019": 1950,
            "release_2.0": 1945
        },
        "year_end": {
            "r092019": 1951,
            "release_2.0": 1949
        },
        "exclude": false,
        "params_exclude": []
    },
    "063-714": {
        "year_init": {
            "r092019": 1978
        },
        "year_end": {
```

(continues on next page)

(continued from previous page)

```
            "r092019": 2010
        },
        "exclude": false,
        "params_exclude": [
            "observations-ws",
            "observations-dpt",
            "observations-at",
            "observations-wd"
        ]
    },
    "152-721": {
        "year_init": {
            "release_2.0": 1851
        },
        "year_end": {
            "release_2.0": 1868
        },
        "exclude": false,
        "params_exclude": []
    }
}
```

6.3 Monthly grids

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2
↪",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2/
↪reports",
  "start":1851,
  "stop":2010,
  "id_out":"optimal",
  "region":"Global",
  "resolution":"lo_res",
  "header":
{
  "filter_by_values":{"header.report_quality":[0]}
},
  "observations-at":
{
  "filter_by_values":{"header.report_quality":[0],"observations-at.quality_flag":[0]}
},
"observations-dpt":
{
  "filter_by_values":{"header.report_quality":[0],"observations-dpt.quality_flag":[0]}
},
"observations-slp":
{
  "filter_by_values":{"header.report_quality":[0],"observations-slp.quality_flag":[0]}
},
"observations-sst":
{
  "filter_by_values":{"header.report_quality":[0],"observations-sst.quality_flag":[0]}
},
"observations-wd":
```

(continues on next page)

(continued from previous page)

```
{
  "filter_by_values":{"header.report_quality":[0],"observations-wd.quality_flag":[0]}
},
"observations-ws":
{
  "filter_by_values":{"header.report_quality":[0],"observations-ws.quality_flag":[0]}
}
}
```

6.4 Quality indicators time series

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2
↪",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2/
↪reports",
  "start":1851,
  "stop":2010
}
```

6.5 Number of reports time series plot

```
{ "file_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports/report_quality-ts.psv",
  "file_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports/nreports-ts.png"
}
```

6.6 Duplicate status time series plot

```
{ "file_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports/duplicate_status-ts.psv",
  "file_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports/duplicate_status-ts.png"
}
```

6.7 Report quality time series plot

```
{ "file_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports/report_quality-ts.psv",
  "file_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports/report_quality-ts.png"
}
```

6.8 Number of reports Hovmöller plots

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2/
↪reports",
  "file_in_id" : "-reports_grid_ts-optimal",
  "file_out_id_sea":"nreports_hovmoller_seasonal.png",
  "file_out_id_mon":"nreports_hovmoller_monthly.png",
  "start":1851,
  "stop":2010,
  "tables":["header","observations-at","observations-sst","observations-slp",
                     "observations-dpt","observations-wd","observations-ws"],
  "colorbar":"magma"
  }
```

6.9 ECV coverage time series plot grid

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2/
↪reports",
  "file_in_id" : "-reports_grid_ts-optimal",
  "file_out":"ecv_coverage_ts_grid.png",
  "start":1851,
  "stop":2010
  }
```

6.10 Number of reports and number of months maps

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪level2/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/level2/
↪reports",
  "projection" : "Mollweide",
  "colorpalette_nobs":"magma",
  "colorpalette_nmonths":"magma",
  "colorbar_title_nobs":"# reports",
  "colorbar_title_nmonths":"# months",
  "normalization_nobs":"log",
  "normalization_nmonths":"log",
  "colorbar_orien":"h",
  "tables":
  {
    "header":
      {
        "nc_file":"header-reports_grid_ts-optimal.nc",
        "out_file_nobs":"header-reports-map-optimal.png",
```

(continues on next page)

(continued from previous page)

```
        "out_file_nmonths":"header-months-map-optimal.png"
      },
  "observations-at":
    {
      "nc_file":"observations-at-reports_grid_ts-optimal.nc",
      "out_file_nobs":"observations-at-reports-map-optimal.png",
      "out_file_nmonths":"observations-at-months-map-optimal.png"
    },
    "observations-sst":
      {
        "nc_file":"observations-sst-reports_grid_ts-optimal.nc",
        "out_file_nobs":"observations-sst-reports-map-optimal.png",
        "out_file_nmonths":"observations-sst-months-map-optimal.png"
    },
    "observations-dpt":
      {
        "nc_file":"observations-dpt-reports_grid_ts-optimal.nc",
        "out_file_nobs":"observations-dpt-reports-map-optimal.png",
        "out_file_nmonths":"observations-dpt-months-map-optimal.png"
    },
    "observations-slp":
      {
        "nc_file":"observations-slp-reports_grid_ts-optimal.nc",
        "out_file_nobs":"observations-slp-reports-map-optimal.png",
        "out_file_nmonths":"observations-slp-months-map-optimal.png"
    },
    "observations-ws":
      {
        "nc_file":"observations-ws-reports_grid_ts-optimal.nc",
        "out_file_nobs":"observations-ws-reports-map-optimal.png",
        "out_file_nmonths":"observations-ws-months-map-optimal.png"
    },
    "observations-wd":
      {
        "nc_file":"observations-wd-reports_grid_ts-optimal.nc",
        "out_file_nobs":"observations-wd-reports-map-optimal.png",
        "out_file_nmonths":"observations-wd-months-map-optimal.png"
    }
  }
}
```

6.11 Mean observed value maps

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪quicklooks",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/marine-user-guide/v4/
↪quicklooks",
  "projection" : "Mollweide",
  "tables":
  {
  "observations-at":
    {
      "nc_file":"observations-at-mean_grid_ts-optimal.nc",
```

(continues on next page)

(continued from previous page)

```
      "out_file":"observations-at-mean-map-optimal.png",
      "scale":1,
      "offset":-273.15,
      "colorpalette":"jet",
      "colorbar_title":"AT ($^\\circ$C)",
      "colorbar_orien":"h",
      "cmax_value":35,
      "cmin_value":-10
    },
    "observations-sst":
      {
        "nc_file":"observations-sst-mean_grid_ts-optimal.nc",
        "out_file":"observations-sst-mean-map-optimal.png",
        "scale":1,
        "offset":-273.15,
        "colorpalette":"jet",
        "colorbar_title":"SST ($^\\circ$C)",
        "colorbar_orien":"h",
        "cmax_value":35,
        "cmin_value":-5
    },
    "observations-dpt":
      {
        "nc_file":"observations-dpt-mean_grid_ts-optimal.nc",
        "out_file":"observations-dpt-mean-map-optimal.png",
        "scale":1,
        "offset":-273.15,
        "colorpalette":"jet",
        "colorbar_title":"DPT ($^\\circ$C)",
        "colorbar_orien":"h",
        "cmax_value":35,
        "cmin_value":-10
    },
    "observations-slp":
      {
        "nc_file":"observations-slp-mean_grid_ts-optimal.nc",
        "out_file":"observations-slp-mean-map-optimal.png",
        "scale":0.01,
        "offset":0,
        "colorpalette":"jet",
        "colorbar_title":"SLP (hPa)",
        "colorbar_orien":"h",
        "cmax_value":1030,
        "cmin_value":980
    },
    "observations-ws":
      {
        "nc_file":"observations-ws-mean_grid_ts-optimal.nc",
        "out_file":"observations-ws-mean-map-optimal.png",
        "scale":1,
        "offset":0,
        "colorpalette":"jet",
        "colorbar_title":"WS (ms$^{-1}$)",
        "colorbar_orien":"h",
        "cmax_value":15,
        "cmin_value":0
    },
```

(continues on next page)

(continued from previous page)

```
    "observations-wd":
      {
        "nc_file":"observations-wd-mean_grid_ts-optimal.nc",
        "out_file":"observations-wd-mean-map-optimal.png",
        "scale":1,
        "offset":0,
        "colorpalette":"twilight_shifted",
        "colorbar_title":"WD (degrees)",
        "colorbar_orien":"h",
        "cmax_value":360,
        "cmin_value":0
    }
  }
}
```

CHAPTER

SEVEN

APPENDIX 2. INDIVIDUAL SOURCE-DECK REPORTS CONFIGURATION FILES

The scripts for the individual source-deck reports can run both on a data release directory or on a release merge under the marine-user-guide data directory.

For individual data releases the following will apply for file paths in the configuration files:

- *dir_data*: <marine_data_directory>/*release*/*dataset*/level[1el2]
- *dir_out*: <marine_data_directory>/*release*/*dataset*/level[1el2]/reports

with corresponding main paths for other input data required.

For release merges the following will apply for file paths in the configuration files:

- *dir_data*: <marine_data_directory>/marine-user-guide/*version*/level2
- *dir_out*: <marine_data_directory>/marine-user-guide/*version*/level2/reports

with corresponding main paths for other input data required.

The sample files provided can be found in the /marine-user-guide/release_demo directory of the configuration repository. They correspond to reports on the level1e data of an individual data release.

7.1 Monthly grids (full dataset)

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "start":1851,
  "stop":2010,
  "id_out":"all",
  "region":"Global",
  "resolution":"lo_res"
}
```

7.2 Monthly grids (optimal dataset)

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "start":1851,
  "stop":2010,
  "id_out":"optimal",
  "region":"Global",
  "resolution":"lo_res",
  "header":
{
  "filter_by_values":{"header.report_quality":[0]}
},
  "observations-at":
{
  "filter_by_values":{"header.report_quality":[0],"observations-at.quality_flag":[0]}
},
"observations-dpt":
{
  "filter_by_values":{"header.report_quality":[0],"observations-dpt.quality_flag":[0]}
},
"observations-slp":
{
  "filter_by_values":{"header.report_quality":[0],"observations-slp.quality_flag":[0]}
},
"observations-sst":
{
  "filter_by_values":{"header.report_quality":[0],"observations-sst.quality_flag":[0]}
},
"observations-wd":
{
  "filter_by_values":{"header.report_quality":[0],"observations-wd.quality_flag":[0]}
},
"observations-ws":
{
  "filter_by_values":{"header.report_quality":[0],"observations-ws.quality_flag":[0]}
}
}
```

7.3 qi_counts

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "start":1851,
  "stop":2010
}
```

7.4 Monthly time series with source to C3S IO flow

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "dir_level1a_ql":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_
↪R3.0.0T/level1a/quicklooks",
  "dir_level1c_ql":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_
↪R3.0.0T/level1c/quicklooks",
  "periods_file":"/gws/smf/j04/c3s311a_lot2/code/marine_code/glamod-marine-config/obs-
↪suite/release_demo-000000/ICOADS_R3.0.0T/source_deck_periods.json"
}
```

7.5 ECV reports time series plots (all data)

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "file_in_id" : "-reports_grid_ts-all",
  "file_out":"ecv_reports_ts_grid-all.png",
  "start":1851,
  "stop":2010
  }
```

7.6 ECV reports time series plots (optimal dataset)

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "file_in_id" : "-reports_grid_ts-optimal",
  "file_out":"ecv_reports_ts_grid-optimal.png",
  "start":1851,
  "stop":2010
  }
```

7.7 Observed parameters latitudinal time series plot

```
{
  "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports",
  "counts_all_id": "no_reports_grid_ts-all",
```

(continues on next page)

(continued from previous page)

```
  "counts_optimal_id": "no_reports_grid_ts-optimal",
  "mean_all_id": "mean_grid_ts-all",
  "mean_optimal_id": "mean_grid_ts-optimal",
  "max_all_id": "max_grid_ts-all",
  "max_optimal_id": "max_grid_ts-optimal",
  "min_all_id": "min_grid_ts-all",
  "min_optimal_id": "min_grid_ts-optimal",
  "start":1851,
  "stop":2010,
  "obs_tables":["observations-at","observations-sst","observations-slp",
                    "observations-dpt","observations-wd","observations-ws"]
  }
```

7.8 Duplicate status time series plot

```
{ "dir_data": "/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "file_data": "duplicate_status-ts.psv",
  "dir_out": "/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "file_out": "duplicate_status-ts.png"
}
```

7.9 Report quality time series plot

```
{ "dir_data": "/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "file_data": "report_quality-ts.psv",
  "dir_out": "/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "file_out": "report_quality-ts.png"
}
```

7.10 Data IO flow plot

```
{ "dir_data":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.
↪0T/level1e/reports",
  "dir_out":"/group_workspaces/jasmin2/glamod_marine/data/release_demo/ICOADS_R3.0.0T/
↪level1e/reports"
}
```