Package: asar (via r-universe)

June 5, 2026

Title Build NOAA Stock Assessment Report

Version 2.3.0

Description Build a full or update stock assessment report for any stock assessment model. Parameterization allows the user to call a template based on their regional science center, species, area, ect.

License MIT + file LICENSE

URL https://github.com/nmfs-ost/asar, https://noaa-fisheries-integrated-toolbox.r-universe.dev/asar

BugReports https://github.com/nmfs-ost/asar/issues

Depends R (>= 4.1.0)

Imports cli, data.table, dplyr, forstringr, fs, glue, gt, naniar, purrr, stats, stringi, stringr, svDialogs, tibble, tidyr, tinytex, utils

Suggests ggplot2, kableExtra, knitr, parallel, quarto, r4ss, rmarkdown, snowfall, stockplotr, testthat (>= 3.0.0), withr

Remotes nmfs-ost/stockplotr

Config/Needs/website quarto

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/testthat/parallel false

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev libx11-dev

Repository https://noaa-fisheries-integrated-toolbox.r-universe.dev

Date/Publication 2026-06-05 13:03:45 UTC

RemoteUrl https://github.com/nmfs-ost/asar

RemoteRef HEAD

RemoteSha 8522d6664c364a7230034a6aad3a9ecedff64d46

Contents

`add_accessibility` *Add Accessibility to .tex documents*

Description

Altering latex file of report to increase accessibility of the document.

Usage

```
add_accessibility(
  x = list.files(getwd())[grep("skeleton.tex", list.files(getwd()))],
  dir = getwd(),
  alttext_csv = file.path(getwd(), "captions_alt_text.csv"),
  compile = TRUE,
  rename = NULL
)
```

Arguments

x	.tex file containing report. Typically produced after initially rendering the skeleton made from create_template.
dir	directory where the tex file is located that will be edited
alttext_csv	File path for the csv file containing alternative text and captions generated when running stockplotr::exp_all_figs_tables
compile	Indicate whether the document (X) should be rendered after these files are changed. Default TRUE.
rename	Indicate a name for the new tex file produced from this function. There is no need to include ".tex" in the name. Defaults to current name and overwrites the current tex file.

Value

This function runs all functions from `asar` associated with accessibility and renders the final document. The document is tagged and includes alternative text from the captions_alt_text.csv produced from `stockplotr` package also available on GitHub.

Examples

```
## Not run:
create_template(
  new_template = TRUE,
  format = "pdf",
  office = "NWFSC",
  region = "U.S. West Coast",
  species = "Dover sole",
  spp_latin = "Microstomus pacificus",
  year = 2010,
  author = c("John Snow" = "AFSC", "Danny Phantom" = "NWFSC", "Patrick Star" = "SEFSC-ML"),
  model_results = output,
  model = "SS3",
  new_section = "an_additional_section",
  section_location = "after-introduction"
)

quarto::quarto_render(file.path(getwd(), "report", "SAR_USWC_Dover_sole_skeleton.qmd"))

add_accessibility(
  x = "SAR_USWC_Dover_sole_skeleton.tex",
  dir = file.path(getwd(), "report"),
  alttext_csv = file.path(getwd(), "captions_alt_text.csv"),
  compile = TRUE
)

## End(Not run)
```

add_alttext *Add alternative text into latex*

Description

Add alternative text into latex

Usage

```
add_alttext(
  x = list.files(getwd())[grep("skeleton.tex", list.files(getwd()))],
  dir = getwd(),
  alttext_csv = file.path(getwd(), "captions_alt_text.csv"),
  compile = TRUE,
  rename = NULL,
  tagged = TRUE
)
```

Arguments

x	.tex file containing report. Typically produced after initially rendering the skeleton made from create_template.
dir	directory where the tex file is located that will be edited
alttext_csv	File path for the csv file containing alternative text and captions generated when running stockplotr::exp_all_figs_tables
compile	Indicate whether the document (X) should be rendered after these files are changed. Default TRUE.
rename	Indicate a name for the new tex file produced from this function. There is no need to include ".tex" in the name. Defaults to current name and overwrites the current tex file.
tagged	Indicate if the input tex file from dir has the latex package, tagpdf, used so that tagging is present.

Value

This function was made to help add in alternative text to latex documents generated from quarto. Quarto does not currently contain a way to add alternative text to PDF documents, so this function was developed as a work around. The addition of alternative text needs to be found in either the rda files produced from stockplotr::exp_all_figs_tables or in the captions_alt_text.csv also produced from the same function. Users not using this format should create a csv file with columns containing "label" and "alt_text" where the label column contains the exact label name when referencing the image/figure in text. The label is very important as it provides a way for the function to match where the alternative text gets placed. When compile is set to TRUE, the alternative text using this format will not be available and must be used in conjunction with `asar::add_tagging()` unless tagged is set to FALSE. Default is TRUE.

Examples

```
## Not run:
create_template(
  new_template = TRUE,
  format = "pdf",
  office = "NWFSC",
  region = "U.S. West Coast",
  species = "Dover sole",
  spp_latin = "Microstomus pacificus",
  year = 2010,
 authors = c("John Snow" = "AFSC", "Danny Phantom" = "NEFSC", "Patrick Star" = "SEFSC-ML"),
  model_results = "Report.sso",
  model = "SS3",
  new_section = "an_additional_section",
  section_location = "after-introduction",
  figures_dir = getwd()
)

quarto::quarto_render(file.path(getwd(), "report", "SAR_USWC_Dover_sole_skeleton.qmd"))

add_alttext(
  x = "SAR_USWC_Dover_sole_skeleton.tex",
  dir = file.path(getwd(), "report"),
  alttext_csv = "my_alttext_file.csv",
  compile = FALSE,
  rename = "SAR_Dover_sole_tagged"
)

## End(Not run)
```

---

add_authors *Format authors for skeleton*

Description

Format authors for skeleton

Usage

```
add_authors(authors, rerender_skeleton = FALSE, prev_skeleton = NULL)
```

Arguments

authors	A character vector of author names with their accompanying affiliations. For example, a Jane Doe at the NWFSC Seattle, Washington office would have an entry of c("Jane Doe"="NWFSC-SWA"). Information on NOAA offices is found in a database located in the package: `system.file("resources","affiliation_info.csv", package = "asar")`. Keys to the office addresses follow the naming convention

of the office acronym (ex. NWFSC) with a dash followed by the first initial of the city then the 2 letter abbreviation for the state the office is located in. If the city has 2 or more words such as Panama City, the first initial of each word is used in the key (ex. Panama City, Florida = PCFL)

`rerender_skeleton`
: Re-create the "skeleton.qmd" in your outline when changes to the main skeleton need to be made. This reproduces the yaml, output (if changed), preamble quantities, and restructures your sectioning in the skeleton if indicated. All files in your folder will remain as is.

`prev_skeleton` A character vector of the previous skeleton file read in through `readLines()`

Value

A list of authors formatted for a yaml in quarto. Viewable by running the return object inside of cat() for each part of the list.

Examples

```
## Not run:
add_authors(
 authors = c("Danny Phantom" = "SWFSC-LJCA", "John Snow" = "AFSC-ABL", "Jane Doe" = "NWFSC-SWA"),
  rerender_skeleton = FALSE
)

## End(Not run)
```

---

add_base_section *Add selected sections to outline*

Description

Add selected sections to outline

Usage

```
add_base_section(custom_sections = NULL)
```

Arguments

`custom_sections`
: List of existing sections to include in the custom template. Note: this only includes sections within list.files(system.file("templates", "skeleton", package = "asar")). The name of the section, rather than the name of the file, can be used (e.g., 'abstract' rather than '00_abstract.qmd'). If adding a new section, also use parameters 'new_section' and 'section_location'.

Value

Call and copy the sections in the package templates to create an outline for a stock assessment

Examples

```
add_base_section(c("executive summary", "assessment", "results"))
```

---

add_child *Write R Chunk to Add Child Document*

Description

Write R Chunk to Add Child Document

Usage

```
add_child(x, label = NULL)
```

Arguments

x	An additional section to add into the template. Options for additional sections are in the 'skeleton' folder. Appropriate files are .qmd files and are formatted as such: XX_section.qmd (i.e., not a, b, c... subfiles).
label	Description of the child document being added. It should be short- one or two words, maximum.

Value

Formatting R chunk for child document to add section into the template/skeleton. Utilize the cat() function to implement into readable text.

Examples

```
add_child("test_quarto.qmd", label = "test_doc")
```

---

add_chunk *Write R chunk to template*

Description

Write R chunk to template

Usage

```
add_chunk(
  x,
  label = NULL,
  add_option = TRUE,
  chunk_option = c("warnings: false", "eval: true"),
  rmark_option = NULL
)
```

Arguments

x	Content to be written within the R chunk. Wrap in quotation marks ("").
label	The name of the chunk in the 'label:' section of the R code chunk. This should be in snakecase (i.e., in which words are written in lowercase and connected by underscores).
add_option	TRUE/FALSE; Option to add additional chunk options. Default is false.
chunk_option	List of chunk options to add. For example: c("output: true", "error: false)
rmark_option	List of chunk options to add after indicating the language of the chunk as used in Rmarkdown.

Value

Write an additional R chunk into the template using this function. The code can be written as usual, just remember to put it entirely in quotes for the function to render it properly

Examples

```
add_chunk("plot(cars$speed, cars$distance)")
```

---

add_section *Add New Section or Subsection to Template*

Description

Add New Section or Subsection to Template

Usage

```
add_section(
  subdir = NULL,
  custom_sections = NULL,
  new_section = NULL,
  section_location = NULL
)
```

Arguments

subdir	Directory where the new sections will be saved. In the create_template function, this defaults to the location where the template is saved.
custom_sections	List of existing sections to include in the custom template. Note: this only includes sections within list.files(system.file("templates", "skeleton", package = "asar")). The name of the section, rather than the name of the file, can be used (e.g., 'abstract' rather than '00_abstract.qmd'). If adding a new section, also use parameters 'new_section' and 'section_location'.

new_section — Names of section(s) (e.g., introduction, results) or subsection(s) (e.g., a section within the introduction) that will be added to the document. Please make a short list if >1 section/subsection will be added. The template will be created as a quarto document, added into the skeleton, and saved for reference.

section_location — Where new section(s)/subsection(s) will be added to the skeleton template. Please use the notation of 'placement-section'. For example, 'in-introduction' signifies that the new content would be created as a child document and added into the 02_introduction.qmd. To add >1 (sub)section, make the location a list corresponding to the order of (sub)section names listed in the 'new_section' parameter.

Value

Add an additional section or subsection to the report template if it is not already present in the default template. This provides the option to add it as a section before or after an existing section, or within a section as a child document. For developers: this function creates a list of sections that will be added to the skeleton file made from create_template.

Examples

```
add_section(
  new_section = "Ecosystem Considerations", section_location = "after-discussion",
  custom_sections = c("introduction.qmd", "model.qmd", "results.qmd", "discussion.qmd"),
  subdir = tempdir()
)
```

---

add_tagging *Add tagging structure to latex documents produced from quarto*

Description

Add tagging structure to latex documents produced from quarto

Usage

```
add_tagging(
  x = list.files(getwd())[grep("skeleton.tex", list.files(getwd()))],
  dir = getwd(),
  compile = TRUE,
  rename = NULL
)
```

Arguments

x	.tex file containing report. Typically produced after initially rendering the skeleton made from create_template.
dir	directory where the tex file is located that will be edited
compile	Indicate whether the document (X) should be rendered after these files are changed. Default TRUE.
rename	Indicate a name for the new tex file produced from this function. There is no need to include ".tex" in the name. Defaults to current name and overwrites the current tex file.

Value

This function was made to help add in latex packages and content associated with PDF tagging. Quarto does not allow the user to edit anything before documentclass, so this function alters the rendered .tex file. Flextable-based tables will not be tagged as flextable is not compatible with tagpdf.

Examples

```
## Not run:
create_template(
  new_template = TRUE,
  format = "pdf",
  office = "NWFSC",
  region = "U.S. West Coast",
  species = "Dover sole",
  spp_latin = "Microstomus pacificus",
  year = 2010,
 authors = c("John Snow" = "AFSC", "Danny Phantom" = "NEFSC", "Patrick Star" = "SEFSC-ML"),
  include_affiliation = TRUE,
  new_section = "an_additional_section",
  section_location = "after-introduction"
)

quarto::quarto_render(file.path(getwd(), "report", "SAR_USWC_Dover_sole_skeleton.qmd"))

add_tagging(
  x = "SAR_USWC_Dover_sole_skeleton.tex",
  dir = file.path(getwd(), "report"),
  compile = TRUE
)

## End(Not run)
```

create_citation *Generate Citation for Stock Assessment Report*

Description

Generate Citation for Stock Assessment Report

Usage

```
create_citation(
  authors = NULL,
  title = "[TITLE]",
  year = format(as.POSIXct(Sys.Date(), format = "%YYYY-%mm-%dd"), "%Y")
)
```

Arguments

authors	A character vector of author names with their accompanying affiliations. For example, a Jane Doe at the NWFSC Seattle, Washington office would have an entry of c("Jane Doe"="NWFSC-SWA"). Information on NOAA offices is found in a database located in the package: `system.file("resources","affiliation_info.csv", package = "asar")`. Keys to the office addresses follow the naming convention of the office acronym (ex. NWFSC) with a dash followed by the first initial of the city then the 2 letter abbreviation for the state the office is located in. If the city has 2 or more words such as Panama City, the first initial of each word is used in the key (ex. Panama City, Florida = PCFL)
title	A custom title that is an alternative to the default title (composed in asar::create_title()). Example: "Management Track Assessments Spring 2024".
year	Year the assessment is being conducted. Default is the year in which the report is rendered.

Value

Generate a citation for use in publications and other references associated with the stock assessment report produced with `asar`.

Examples

```
## Not run:
create_citation(
  title = "SA Report for Jellyfish",
 authors = c("Danny Phantom" = "SWFSC-LJCA", "John Snow" = "AFSC-ABL", "Jane Doe" = "NWFSC-SWA"),
  year = 2024
)

## End(Not run)
```

create_figures_doc *Create Quarto Document of Figures*

Description

Create Quarto Document of Figures

Usage

```
create_figures_doc(subdir = getwd(), figures_dir = getwd())
```

Arguments

subdir	Location of subdirectory storing the assessment report template
figures_dir	The location of the "figures" folder, which contains figures files.

Value

A quarto document with pre-loaded R chunk that adds the stock assessment tables from the nmfs-ost/stockplotr R package. The quarto document will become part of the stock assessment outline.

Examples

```
## Not run:
create_figures_doc(
  subdir = getwd(),
  figures_dir = here::here()
)

## End(Not run)
```

create_inheader_tex *Create in-header latex document*

Description

Create in-header latex document

Usage

```
create_inheader_tex(species = NULL, year = NULL, subdir)
```

Arguments

species	common species name - used for footer
year	year assessment is conducted
subdir	directory where other files will be copied into

Value

Create an in-header latex document that dynamically changes based on the species and year along with other factors.

`create_tables_doc` *Create Quarto Document of Tables*

Description

Only tables in an rda format (e.g., my_table.rda) will be imported. Tables in other formats (e.g., .jpg, .png) are not supported; they lack text recognition. See the `asar` custom figures and tables vignette for more information about making .rda files with custom tables.

Usage

```
create_tables_doc(subdir = getwd(), tables_dir = getwd())
```

Arguments

`subdir`	Location of subdirectory storing the assessment report template
`tables_dir`	The location of the "tables" folder, which contains tables files.

Details

If your table is too wide to print on a portrait-oriented page, the page will be rotated to landscape view. If if is too wide to print in landscape view, it will be split into multiple tables. In this case, a new rda will be created and is identifiable by the phrase "split" in the filename (e.g., indices.abundance_table.rda will generate a new indices.abundance_table_split.rda file), and column 1 will be repeated across split tables. These tables will share the same caption. To specify a different repeated column(s), use asar::export_split_tbls with your preferred essential_columns value.

Value

Create a quarto document as part of a stock assessment outline with pre-loaded R chunks that add stock assessment tables from the nmfs-ost/stockplotr R package, or other tables in the same rda format.

Examples

```
## Not run:
create_tables_doc(
  subdir = getwd(),
  tables_dir = here::here()
)

## End(Not run)
```

create_template *Create Stock Assessment Report Template*

Description

To see templates included in the base skeleton, please run 'list.files(system.file('templates','skeleton', package = 'asar'))' in the console.

Usage

```
create_template(
  format = "pdf",
  type = "sar",
  office = c("AFSC", "PIFSC", "NEFSC", "NWFSC", "SEFSC", "SWFSC"),
  region = NULL,
  species = "species",
  spp_latin = NULL,
  year = format(as.POSIXct(Sys.Date(), format = "%YYYY-%mm-%dd"), "%Y"),
  authors = NULL,
  file_dir = getwd(),
  title = "[TITLE]",
  model_results = NULL,
  tables_dir = getwd(),
  figures_dir = getwd(),
  spp_image = "",
  bib_file = "asar_references.bib",
  new_template = TRUE,
  rerender_skeleton = FALSE,
  custom = FALSE,
  custom_sections = NULL,
  new_section = NULL,
  section_location = NULL,
  parameters = TRUE,
  param_names = NULL,
  param_values = NULL,
  ...
)
```

Arguments

format	Rendering format (pdf, html, or docx).
type	Type of report to build. Default is "sar" (NOAA Fisheries Stock Assessment Report).
office	Regional Fisheries Science Center producing the report (i.e., AFSC, NEFSC, NWFSC, PIFSC, SEFSC, SWFSC).

region — Full name of region in which the species is evaluated (if applicable). If the region is not specified for your center or species, do not use this variable.

species — Full common name for target species. Split naming with a space and capitalize first letter(s). Example: "Dover sole".

spp_latin — Latin name for the target species. Example: "Pomatomus saltatrix".

year — Year the assessment is being conducted. Default is the year in which the report is rendered.

authors — A character vector of author names with their accompanying affiliations. For example, a Jane Doe at the NWFSC Seattle, Washington office would have an entry of c("Jane Doe"="NWFSC-SWA"). Information on NOAA offices is found in a database located in the package: `system.file("resources","affiliation_info.csv", package = "asar")`. Keys to the office addresses follow the naming convention of the office acronym (ex. NWFSC) with a dash followed by the first initial of the city then the 2 letter abbreviation for the state the office is located in. If the city has 2 or more words such as Panama City, the first initial of each word is used in the key (ex. Panama City, Florida = PCFL)

file_dir — Location of stock assessment files produced by this function. Default is the working directory.

title — A custom title that is an alternative to the default title (composed in asar::create_title()). Example: "Management Track Assessments Spring 2024".

model_results — Path to standard output file made from `stockplotr::convert_output()`

tables_dir — The location of the "tables" folder, which contains tables files. Default is the working directory.

figures_dir — The location of the "figures" folder, which contains figures files. Default is the working directory.

spp_image — File path to the species' image if not using the image included in the project's repository.

bib_file — File path to a .bib file used for citing references in the report

new_template — TRUE/FALSE; Create a new template? If true, will pull the last saved stock assessment report skeleton. Default is false.

rerender_skeleton — Re-create the "skeleton.qmd" in your outline when changes to the main skeleton need to be made. This reproduces the yaml, output (if changed), preamble quantities, and restructures your sectioning in the skeleton if indicated. All files in your folder will remain as is.

custom — TRUE/FALSE; Build custom sectioning for the template, rather than the default for stock assessments in your region? Default is false.

custom_sections — List of existing sections to include in the custom template. Note: this only includes sections within list.files(system.file("templates", "skeleton", package = "asar")). The name of the section, rather than the name of the file, can be used (e.g., 'abstract' rather than '00_abstract.qmd'). If adding a new section, also use parameters 'new_section' and 'section_location'.

new_section — Names of section(s) (e.g., introduction, results) or subsection(s) (e.g., a section within the introduction) that will be added to the document. Please make a short list if >1 section/subsection will be added. The template will be created as a quarto document, added into the skeleton, and saved for reference.

section_location — Where new section(s)/subsection(s) will be added to the skeleton template. Please use the notation of 'placement-section'. For example, 'in-introduction' signifies that the new content would be created as a child document and added into the 02_introduction.qmd. To add >1 (sub)section, make the location a list corresponding to the order of (sub)section names listed in the 'new_section' parameter.

parameters — TRUE/FALSE; For parameterization of the script. Default is true.

param_names — List of parameter names that will be called in the document. Parameters automatically included: office, region, species (each of which are listed as individual parameters for this function, above).

param_values — List of values associated with the order of parameter names. Parameters automatically included: office, region, species (each of which are listed as individual parameters for this function, above).

... — Additional arguments passed into functions used in create_template such as `create_citation()`, `format_quarto()`, `add_chunk()`, ect

Value

Create template and pull skeleton for a stock assessment report. Function builds a YAML specific to the region and utilizes current resources and workflows from different U.S. Fishery Science Centers. General sections are called as child documents in this skeleton and each of the child documents should be edited separately.

Examples

```
## Not run:
create_template(
  new_section = "a_new_section",
  section_location = "before-introduction"
)


create_template(
  new_template = TRUE,
  format = "pdf",
  office = "NWFSC",
  species = "Dover sole",
  spp_latin = "Microstomus pacificus",
  year = 2010,
  authors = c(
    "John Snow" = "AFSC",
    "Danny Phantom" = "NEFSC",
    "Patrick Star" = "SEFSC-ML"
  ),
```

```
  model_results = here::here("folder", "std_output.rda"),
  figures_dir = here::here(),
  tables_dir = here::here("tables_folder_location"),
  new_section = "an_additional_section",
  section_location = "after-introduction"
)

asar::create_template(
  new_template = TRUE,
  format = "pdf",
  office = "PIFSC",
  species = "Striped marlin",
  spp_latin = "Kajikia audax",
  year = 2018,
  authors = c("John Snow" = "AFSC"),
  new_section = c("a_new_section", "another_new_section"),
  section_location = c("before-introduction", "after-introduction"),
  custom = TRUE,
  custom_sections = c("executive_summary", "introduction")
)

create_template(
  new_template = TRUE,
  format = "pdf",
  office = "NWFSC",
  region = "my_region",
  species = "Bluefish",
  spp_latin = "Pomatomus saltatrix",
  year = 2010,
  authors = c("John Snow", "Danny Phantom", "Patrick Star"),
  title = "Management Track Assessments Spring 2024",
  parameters = TRUE,
  param_names = c("region", "year"),
  param_values = c("my_region", "2024"),
  model_results = here::here("folder", "std_output.rda"),
  new_section = "an_additional_section",
  section_location = "before-discussion",
  type = "sar",
  custom = TRUE,
  custom_sections = c("executive_summary", "introduction", "discussion"),
  spp_image = "dir/containing/spp_image"
)

## End(Not run)
```

`create_title` *Write Stock Assessment Report Title*

Description

Write Stock Assessment Report Title

Usage

```
create_title(
  type = "skeleton",
  office = "",
  species = "species",
  spp_latin = NULL,
  region = NULL,
  year = format(Sys.Date(), "%Y"),
  complex = NULL
)
```

Arguments

type	Type of report to build. Default is "sar" (NOAA Fisheries Stock Assessment Report).
office	Regional Fisheries Science Center producing the report (i.e., AFSC, NEFSC, NWFSC, PIFSC, SEFSC, SWFSC).
species	Full common name for target species. Split naming with a space and capitalize first letter(s). Example: "Dover sole".
spp_latin	Latin name for the target species. Example: "Pomatomus saltatrix".
region	Full name of region in which the species is evaluated (if applicable). If the region is not specified for your center or species, do not use this variable.
year	Year the assessment is being conducted. Default is the year in which the report is rendered.
complex	TRUE/FALSE; Is this a species complex? Default is false.

Value

Return a string containing a title for a NOAA Fisheries stock assessment report.

Examples

```
create_title(
  type = "SAR", office = "SEFSC", species = "Red Snapper",
  spp_latin = "Lutjanus campechanus", region = "South Atlantic",
  year = 2024
)
```

---

create_titlepage_tex *Create a title page latex document*

Description

Create a title page latex document

Usage

```
create_titlepage_tex(office = "", subdir, species = "")
```

Arguments

office	primary science center writing the document
subdir	directory where files are going to be held
species	target species for assessment

Value

Create a _titlepage.tex document that contains formatting options for a cover page. The only thing that changes currently is the primary author's fishery science center.

---

create_yaml *Create string for yml header in quarto file*

Description

Create string for yml header in quarto file

Usage

```
create_yaml(
  format = "pdf",
  office = NULL,
  region = NULL,
  species = "species",
  spp_latin = NULL,
  spp_image = "",
  year = NULL,
  bib_name = NULL,
  bib_file = "asar_references.bib",
  author_list = NULL,
  title = "[TITLE]",
  rerender_skeleton = FALSE,
  prev_skeleton = NULL,
  prev_format = NULL,
  parameters = TRUE,
  param_names = NULL,
  param_values = NULL,
  type = "SAR"
)
```

Arguments

format	Rendering format (pdf, html, or docx).
office	Regional Fisheries Science Center producing the report (i.e., AFSC, NEFSC, NWFSC, PIFSC, SEFSC, SWFSC).
region	Full name of region in which the species is evaluated (if applicable). If the region is not specified for your center or species, do not use this variable.
species	Full common name for target species. Split naming with a space and capitalize first letter(s). Example: "Dover sole".
spp_latin	Latin name for the target species. Example: "Pomatomus saltatrix".
spp_image	File path to the species' image if not using the image included in the project's repository.
year	Year the assessment is being conducted. Default is the year in which the report is rendered.
bib_name	Name of a bib file being added into the yaml. For example, "asar.bib".
bib_file	File path to a .bib file used for citing references in the report
author_list	A list of strings containing pre-formatted author names and affiliations that would be found in the format in a yaml of a quarto file when using base R function `cat()`.
title	A custom title that is an alternative to the default title (composed in asar::create_title()). Example: "Management Track Assessments Spring 2024".
rerender_skeleton	Re-create the "skeleton.qmd" in your outline when changes to the main skeleton need to be made. This reproduces the yaml, output (if changed), preamble quantities, and restructures your sectioning in the skeleton if indicated. All files in your folder will remain as is.
prev_skeleton	Vector of strings containing all the lines of the previous skeleton file. File is read in using the function readLines from base R.
prev_format	The format that the previous skeleton was directed to render to. Parameter is inherited from create_template.
parameters	TRUE/FALSE; For parameterization of the script. Default is true.
param_names	List of parameter names that will be called in the document. Parameters automatically included: office, region, species (each of which are listed as individual parameters for this function, above).
param_values	List of values associated with the order of parameter names. Parameters automatically included: office, region, species (each of which are listed as individual parameters for this function, above).
type	Type of report to build. Default is "sar" (NOAA Fisheries Stock Assessment Report).

Value

Create a string indicating the important formatting pieces for a quarto file for a stock assessment report.

Examples

```
## Not run:
my_author_list <- paste(
  "  - name: 'Patrick Star'",
  "    affiliations:",
  "      - name: 'NOAA Fisheries Southeast Fisheries Science Center'",
  "        address: '75 Virginia Beach Drive'",
  "        city: 'Miami'",
  "        state: 'FL'",
  "        postal-code: '33149'",
  sep = "\n"
)
create_yaml(
  rerender_skeleton = FALSE,
  prev_skeleton = NULL,
  title = "My title",
  author_list = my_author_list,
  author = c("Patrick Star" = "SEFSC"),
  office = "SEFSC",
  add_author = NULL,
  spp_image = NULL,
  species = "",
  spp_latin = NULL,
  region = NULL,
  format = "pdf",
  parameters = TRUE,
  param_names = NULL,
  param_values = NULL,
  bib_file = "path/asar_references.bib",
  bib_name = "asar_references.bib",
  year = 2025
)

## End(Not run)
```

---

export_glossary *Create and export glossary*

Description

Create and export glossary

Usage

```
export_glossary()
```

`export_split_tbls` *Split extra-wide tables into smaller tables and export*

Description

Split extra-wide tables into smaller tables and export

Usage

```
export_split_tbls(
  tables_dir = NULL,
  plot_name = NULL,
  essential_columns = NULL
)
```

Arguments

`tables_dir`	The location of the "tables" folder, which contains tables files.
`plot_name`	Name of the .rda file containing the table
`essential_columns`	The columns that will be retained between the split tables, formatted as a sequence (e.g., 1:2 for columns 1-2, or 1 for a single column). Example: for the indices table, this could be the year column.

Value

The number of split tables

Examples

```
## Not run:
export_split_tbls(
  tables_dir = here::here(),
  plot_name = "bnc_table.rda",
  essential_columns = 5
)

export_split_tbls(
  tables_dir = getwd(),
  plot_name = "indices.abundance_table.rda",
  essential_columns = 1:2
)

## End(Not run)
```

format_quarto *Add Formatting Arguments for YAML Header*

Description

Add Formatting Arguments for YAML Header

Usage

```
format_quarto(format = "pdf", type = "sar")
```

Arguments

format	Rendering format (pdf, html, or docx).
type	Type of report to build. Default is "sar" (NOAA Fisheries Stock Assessment Report).

Value

This function returns part of a quarto YAML header involved in formatting the document during rendering.

Examples

```
## Not run:
format_quarto(format = "pdf")

## End(Not run)
```

gt_split *Split a table into a group of tables (a* gt_group*)*

Description

With a **gt** table, you can split it into multiple tables and get that collection in a gt_group object. This function is useful for those cases where you want to section up a table in a specific way and print those smaller tables across multiple pages (in RTF and Word outputs, primarily via gtsave, or, with breaks between them when the output context is HTML.

Usage

```
gt_split(data, row_every_n = NULL, row_slice_i = NULL, col_slice_at = NULL)
```

Arguments

`data` *The gt table data object*
`obj:<gt_tbl>` // **required**
This is the **gt** table object that is commonly created through use of the `gt::gt()` function.

`row_every_n` *Split at every n rows*
`scalar<numeric|integer>` // *default:* `NULL` (`optional`)
A directive to split at every *n* number of rows. This argument expects a single numerical value.

`row_slice_i` *Row-slicing indices*
`vector<numeric|integer>` // *default:* `NULL` (`optional`)
An argument for splitting at specific row indices. Here, we expect either a vector of index values or a function that evaluates to a numeric vector.

`col_slice_at` *Column-slicing locations*
`<column-targeting expression>` // *default:* `NULL` (`optional`)
Any columns where vertical splitting across should occur. The splits occur to the right of the resolved column names. Can either be a series of column names provided in `c()`, a vector of column indices, or a select helper function (e.g. starts_with, ends_with, contains, matches, num_range, and everything).

Value

An object of class `gt_group`.

Examples

Use a subset of the `gt::gtcars` dataset to create a **gt** table. Format the `msrp` column to display numbers as currency values, set column widths with cols_width, and split the table at every five rows with `gt_split()`. This creates a `gt_group` object containing two tables. Printing this object yields two tables separated by a line break.

```
gtcars |>
  dplyr::slice_head(n = 10) |>
  dplyr::select(mfr, model, year, msrp) |>
  gt() |>
  fmt_currency(columns = msrp) |>
  cols_width(
    year ~ px(80),
    everything() ~ px(150)
  ) |>
  gt_split(row_every_n = 5)
```

Use a smaller subset of the `gt::gtcars` dataset to create a **gt** table. Format the `msrp` column to display numbers as currency values, set the table width with `gt::tab_options()` and split the table at the `model` column This creates a `gt_group` object again containing two tables but this time we get a vertical split. Printing this object yields two tables of the same width.

```
gtcars |>
  dplyr::slice_head(n = 5) |>
  dplyr::select(mfr, model, year, msrp) |>
  gt() |>
  fmt_currency(columns = msrp) |>
  tab_options(table.width = px(400)) |>
  gt_split(col_slice_at = "model")
```

Function ID

14-2

Function Introduced

`v0.9.0` (Mar 31, 2023)

Note

This function is a temporary export of asar, but all development and rights belong to `rstudio/gt`. This function provides a fix to the function introduced by a bug in gt v1.3.0. Until this is corrected in the package, we are using the function here. Once this bug is patched, we will deprecate and remove this function from asar and direct users to use the gt package version of this function.

`ID_tbl_length_class` *Identify table length class*

Description

Identify table length class

Usage

```
ID_tbl_length_class(tables_dir, plot_name)
```

Arguments

`tables_dir`	The location of the "tables" folder, which contains tables files.
`plot_name`	Name of the .rda file containing the table

Value

The length class of a table: regular or long. The result will determine whether the table can be rendered on a page as-is, or if it needs to be split across multiple pages.

Examples

```
## Not run:
ID_tbl_length_class(
  plot_name = "indices.abundance_table.rda",
  tables_dir = here::here()
)

## End(Not run)
```

`ID_tbl_width_class` *Identify table width class*

Description

Identify table width class

Usage

```
ID_tbl_width_class(tables_dir, plot_name, portrait_pg_width)
```

Arguments

`tables_dir`	The location of the "tables" folder, which contains tables files.
`plot_name`	Name of the .rda file containing the table
`portrait_pg_width`	The amount of space between the margins of a portrait-oriented page, in inches. Represents the threshold for the maximum width of a table that can be rendered on a portrait page before it needs to be resized, rotated, and/or split across multiple pages.

Value

The width class of a table: regular, wide, or extra-wide. The result will determine whether the table can be rendered on a portrait page as-is, or if it needs to be resized, rotated, and/or split across multiple pages.

Examples

```
## Not run:
ID_tbl_width_class(
  plot_name = "indices.abundance_table.rda",
  tables_dir = here::here(),
  portrait_pg_width = 5
)

## End(Not run)
```

render_lg_table *Split an extra-wide table into multiple tables*

Description

Split an extra-wide table into multiple tables

Usage

```
render_lg_table(report_gt, essential_columns, tables_dir, plot_name)
```

Arguments

report_gt	The extra-wide gt table.
essential_columns	The columns that will be retained between the split tables, formatted as a sequence (e.g., 1:2 for columns 1-2, or 1 for a single column). Example: for the indices table, this could be the year column.
tables_dir	The location of the "tables" folder, which contains tables files.
plot_name	Name of the .rda file containing the table

Value

A list of the split tables.

Examples

```
## Not run:
render_lg_table(
  report_gt = indices_table,
  essential_columns = 1,
  tables_dir = here::here(),
  plot_name = "indices.abundance_table.rda"
)

render_lg_table(
  report_gt = important_table,
  essential_columns = 1:3,
  tables_dir = "data",
  plot_name = "bnc_table.rda"
)

## End(Not run)
```

Index