Ecosystem Status Report Best Practices Document

Gulf IEA

2025-11-17

Table of contents

1 Our Workflow

1.1 Purpose

The purpose of this living document is to ensure reproducibility and standardization of Gulf IEA Ecosystem Status Reports. This guidance document goes through all the steps for creating an ESR in quarto, from identifying indicators to writing the final report. A Kumu map of our workflow is below. The workflow and information in this document may change as iterations of ESRs are completed, so check back often for updates.

> **!** Important
>
> THIS DOCUMENT IS NOT YET COMPLETED. IT IS STILL IN DRAFT FORM.

> **i** Map Interaction Guide
>
> This map is fully interactive. Use the following steps to explore the system:
>
> 1. **Zoom and Pan:** Use your mouse scroll wheel/trackpad to zoom in and out. Click and drag the map background to pan.
> 2. **View Profiles: Click any element (node)** to open its profile on the side panel, which contains detailed information.
> 3. **Focus on a Node:** Use the **magnifying glass icon** on the map's toolbar to search for a specific element, or click the **Focus button**, which pops up on the lower right of the side panel when you click on an element, to highlight that element and its immediate connections.

1.2 Naming conventions

Add a table here with all of our outputs/products, how they are named, and an example

2 Start Here

If you are making an ESR from scratch, step one is to go to the ESR-Report-Template repository in the Gulf-IEA Github organization and click "use this template". Re-name the new repository xxx-ESR (e.g., Gulf-ESR, SA-ESR, Caribbean-ESR, Wind-ESR, etc.). You will also need to go to the projects tab for the Gulf-IEA organization and create a new project from the template "Indicator scoping" project. Name the project "Indicator scoping [report]" (e.g., "Indicator scoping Gulf", "Indicator scoping Caribbean", etc.) and link that project to your new repository.

If you are updating an existing ESR, go to the repository for that ESR (e.g., Caribbean-ESR), make sure the previous version was tagged by checking the tags in the main branch. From there, go to the Indicator scoping project linked to the repository. Change everything in the "selected" column to maybe for all listed metrics. Change everything in the "Code reviewer name", "Code reviewed", and "Production form submitted" columns back to blank.

> **Note**
>
> Note: Here we should add some screenshots showing where things are and what to select.

3 Indicator Scoping

3.1 Compiling ideas

If you think of ideas for a new metric or someone you talk to suggests a metric, go to the issues tab and create a new issue. Use the "Metric Idea Form" issue template. Fill out as much information as you can, leaving entries blank as needed. Anyone can fill this form out with an idea. If you already know the data do not exist or are extremely limited, you can note that. You should ideally know where the data exist so you can note that on the form before we decide to include the metric. If you identify data online, you should test that it can be downloaded and everything looks kosher. If you need to run any code just to check/download the data and take a preliminary look at what it is, you can make a script and put it in the sandbox/prelim_code folder. Make sure to link the script/code in the comments of the metric idea issue corresponding to that metric.

> **!** Important
>
> When you've finished filling out the Metric Idea Form, make sure you assign that issue to the "Indicator scoping xxx" project, so it automatically adds an entry to the project board.

3.2 Metric selection

Once the project board is filled with ideas and the appropriate people / teams have been consulted about metric ideas, the IEA team will meet to go over the list of metric ideas and decide which ones to include in the ESR and which ones not to. By the end of this meeting, all the metrics that will be included will have the selected category changed to "yes", everything else will be changed to "no" if the metric is not going to be used in the current report. Also, each metric will be assigned to an IEA POC by the end of the meeting. Make sure a root name has been created for each metric and documented in the project.

4 Data Processing

If an IEA POC has been assigned a metric, it is their job to either create the code, or find and adapt existing code, or re-run existing code to update for a new year. If it is important to consult with SMEs or data stewards, that is the responsibility of the IEA POC.

STEP 1: **CRUCIAL:** verify your R version. You must ensure your R studio session for this project is running the version that is listed in the renv.lock file in the main branch. If you really need to use a different R version, you will need to update the renv.lock file in the main branch, and all collaborators will need to use the same R version when working in the repo. Instructions for changing the R version in the main branch can be found HERE.

STEP 2: Either clone the repository to your local machine or if you already have it, make sure to run git pull from the main branch before starting work in the new branch to ensure everything is up to date. Then restart your R session. Read the renv.lock file from the main branch to ensure all the compatible packages are in the branch's local library using the following code:

```
renv::restore()
```

This step may take a while, since it will install all the packages and package dependencies used in the repo. The benefit of this is that you will not need to install any packages to run any of the code in the repo. You will just need to call packages using the library function. The renv.lock file ensures the code will always run with the R and package versions it was created with.

STEP 3: Once the main branch is up to date on your local machine, create a new branch and name it the root name for the metric (the root name should be designated in the metric idea form). Switch to that branch in R studio.

STEP 4: Identify the data type (automated, non-automated, or confidential)

STEP 5: Within the new branch, use the create_metric_code_file function in IEAnalyzeR to create an R script (or quarto doc if you prefer) within the appropriate scripts/metrics folder based on your data type. Use the appropriate template (metric_code_template.R or metric_code_template.qmd). To use the function, specify the metric name, format of the file (R or qmd), and data type. Best practice is to run the code for the function in your console. The new script file will be created in the appropriate folder. (If for some reason you need to edit the templates, they can be found in scripts/other/templates).

STEP 6: In your R script or qmd file, follow the template prompts to load your necessary packages and bring in the data/call the data from an online source, do whatever data manipulations/analysis are needed, and convert to the standard IEA data format (see example below). The template will prompt you to save the data in the appropriate format and location.

> **Important**
>
> Be cognizant of others. Annotate your code. You will be expected to add a plain language version of your code to the production form to be included in the methods document. Be kind to your future self and do it now.

STEP 7: When you finish working on your branch and are ready to push it to main, you first need to check if you installed any additional packages that are not already in the renv.lock file (run the code below to check). If so, you will need to snapshot the new dependencies and then commit and push before you initiate a pull request.

```
#First check if there are packages in your branch that are not in the lock file:
renv::status()

#If there are new packages, you need to snapshot the dependencies:
renv::snapshot()

#Accept the prompt to update the lock file. Commit and push the new lock file.
```

> **Tip**
>
> You might run into an issue with the renv/activate.R file when trying to push to github. This file ensures your R session is configured correctly to work with renv. However, it can sometimes be modified when a local renv version is updated, leading to a Git conflict. Since it is just an autoloader script, it is usually safe to overwrite your local version with the version from Github. To do so, just run the following in your Git terminal to discard the local changes:
>
> ```
> git checkout renv/activate.R
> ```

> **Note**
>
> Once your code is done be sure to check off the "coded" column in the indicator scoping project.

> **Note**
>
> include an example here of the appropriate data format.

STEP 8: One you are satisfied with your code and you have ensured the renv.lock file is updated, do a pull request and tag another IEA POC as a reviewer. The reviewer will pull the branch, go through the script, test that the code works, check for errors, and accept the pull request once everything looks good. If there are issues, contact the IEA POC.

> **Note**
>
> Once the code is checked and the branch is merged into the main branch, check off the "code checked" column in the indicator scoping project.

STEP 9: The IEA POC now fills out the metric production form for each of their assigned metrics. Some of the information can be copied from the metric idea form.

> **Note**
>
> Make sure you go back to the indicator scoping project and check off the column called "Production form submitted".

5 Report Writing

STEP 1: The IEA indicator development team (IEA lab group) meets to go over all the metrics and decide how to turn the metrics into indicators and how to structure the report (sections, order, etc.)

STEP 2: Send the Report Text Google Form to SMEs / data stewards / whoever is best suited to writing about the metric/indicator and interpreting the plot.

STEP 3: The Quarto Book Master creates the quarto book files, updates the YAML, etc. Run the template R code to pull the text data from the google form and combine it with the code chunks to create the indicator plots in the qmd files.

6 Glossary

- **Automated**
  - Data that can be downloaded using an API or online database.

- **Coded**
  - An R script (or Python) that can be used to download the data and has been successfully tested.

- **Code Review**
  - During the metric idea process, an additional IEA team member is needed to review the code and ensure it meets the requirements. This will be initiated as a pull request in GitHub, and a reviewer will be assigned. The reviewer will need to test the code and approve the branch merge. The reviewer also needs to make the indicator plot and review the plot as a gut check.

- **Confidential**
  - Data containing sensitive personal or financial information that requires aggregation before public release; any person working with unaggregated data must have a non-disclosure agreement on file.

- **Data Access Point**
  - Person or website service where data can be accessed or downloaded (e.g., ERDDAP)

- **Data Acquired**
  - You have downloaded all the data necessary to create the metric for full production.

- **Data Identified**

  - You have found a data source (e.g., API, data POC, or other) but have not downloaded the data; however, you may have attempted to access or downloaded some of the data.

- **Data Source**
  - Program, lab, or person that produces the data (e.g., NASA).

- **Data Steward**
  - Person most directly associated with the data; can be an external SME, but not necessarily.

- **Data type**
  - Categories of data based on ease of download or level of confidentiality.

- **External SME**
  - Non-IEA member subject matter expert.

- **IEA POC / Responsible Member**
  - IEA member overseeing the indicator/metric.

- **Indicator**
  - Usually, the metric, but it could have multiple metrics.

- **Indicator Directory**
  - A list of all indicators used by the SE-IEA for all past ESRs. This is called the 'Indicator Catalog' by the NEFSC.

- **Methods Document**
  - Specific to each ESR produced, and contains the methods for each data, metric, and indicator used. The information will come from the Metric Production Form. This is called the 'Technical Document' by the NEFSC.

- **Metric**

  - The smallest unit of an indicator or analysis.

- **Metric Idea Form**
  - Issue template to be filled out in GitHub repository for current ESR to scope out new metric for upcoming ESR; to be filled out by IEA POC (or potentially external SME if GitHub proficient); once completely filled out, then it can be transferred into the production form.

- **Metric Production Form**
  - Issue template to be filled out in GitHub repository for current ESR to document metric for upcoming ESR; to be filled out by IEA POC; this will directly feed into the Methods Document.

- **Non-automated**
  - Data that you have to email someone to get; this data, once received, goes to the unformatted data folder.

- **Quarto Book Master**
  - The one person who is responsible for the report creation and the one who is creating and updating the Quarto files. Only one person will be doing this, so there are no merge conflicts. This person will still call on others to help in report creation.

- **Sandbox**
  - Folder in ESR repo that holds preliminary code or other code in progress that don't fit elsewhere; you will find some poop and an old action figure missing a limb or two.

- **Scouted**
  - Part of the metric ideation process, the IEA POC, or the person who initially had the idea for the metric and filled out the metric idea form, will have done some initial research on the data and outlined the details (e.g., data availability, data download websites, external POCs).

- **Unformatted data**
  - Data that is downloaded from an online source, database, or email as is. Data will not be processed for this.