

2nd Technical ESMValTool Workshop 2021

The Second Technical ESMValTool Workshop 2021 took place via video call from 23-25 November 2021 with 34 participants from BSC, DLR, Met Office, NLeSC, PML, SMHI, U Bremen and U Reading. The main goal of the workshop was to bring together the development community, discuss future strategies and provide updates on progress since the last workshop in May 2021. Specifically, the workshop included the following topics and side meetings:

- Latest developments: mamba Installation, highlights v2.4, automatic download of missing data, selection of time ranges
- Native model grids and diagnostics for monitoring model runs
- Support for regional models
- Testing recipes
- Non-backward compatible changes
- General discussion
- Side meetings:
 - a. user engagement team
 - b. cross team meeting
 - c. pinning dependencies



Figure 1 Participants of the Technical ESMValTool Workshop in November 2021.

In the following, thoughts and ideas discussed during the individual sessions are briefly summarized as bullet points. This summary is rather intended as a reminder of the sessions for the participants and therefore makes no claim of completeness.

Latest developments: mamba Installation, highlights v2.4, automatic download of missing data, selection of time ranges

- mamba + highlights v2.4 + automatic download: see GitHub and ReadTheDocs (V)
- Presentation on new feature: selecting time ranges in recipes (Saskia)

Native model grids and diagnostics for monitoring model runs

- Presentation: new diagnostics for monitoring (Javi)
- Iris UGRID for support of unstructured grids available soon (Klaus)

Support for regional models

- GitHub project on regional models: <https://github.com/orgs/ESMValGroup/projects/11>
- Proposal to add another team for regional model work (Klaus will create such a team on GitHub, see <https://github.com/orgs/ESMValGroup/teams/sig-regional>)
- CORDEX grids: work needed for regridding and spatial operators
- Regridding might be available in v2.5; several fixes for datasets might be needed though

Testing recipes

- Simulated data and testing framework introduced (Emma)
- Intermediate strategy for testing until a more comprehensive testing is available: small testing recipes working with a reduced set of real data and commonly used operations; a number of examples has been developed during the workshop:
 - https://github.com/ESMValGroup/ESMValTool/tree/workshop_test_recipes_al
 - Testing CMIP3, 5, 6 anomalies, horizontal maps, perfmetrics: https://github.com/ESMValGroup/ESMValTool/tree/test_recipe_lisa
 - Reduced version of existing recipe: https://github.com/ESMValGroup/ESMValTool/tree/fast_version_deangelis_for_test
 - https://github.com/ESMValGroup/ESMValTool/tree/recipe_eyring13jgr_fig12_test_recipe
 - https://github.com/ESMValGroup/ESMValTool/tree/Arctic_telecon
- These testing recipes are seen as a complimentary approach to the simulated data approach (Emma) until a more sophisticated testing framework is in place
- Proposal to approach (few) individual developers to provide a testing recipe from their own science recipes (e.g. ocean, land surface, etc.)
- Testing and reporting framework needed to run the testing recipes (use ESMValBot?)
- Report progress here: <https://github.com/ESMValGroup/ESMValCore/issues/1361>

Non-backward compatible changes

- Needed: common understanding / definition of non-backward compatible changes
 - Different classes of changes (core, diagnostics, dependencies)?
 - Anything that requires user to apply changes
 - Types of users are different which makes things more complicated
 - Using stable releases is strongly suggested for users / developers
 - Information on command line what has changed and what needs to be changed
- Support for “normal” users for fixes needed?
- Cleaning up log-files
- Changes to the tool sometimes hard to find in documentation; changelog is not sufficient for all users to know what they are expected to do; potentially add such a chapter to the tutorial
- Code maintenance is potentially a large overhead for scientists (tradeoff between time spent maintaining code and reinventing the wheel (writing own code))
- The overhead problem also applies to recipe owners
- Current main channel of communication: issues; does not seem to work too well
 - using mailing list more frequently
- public interfaces: recipe; public interfaces of preprocessor; described in the documentation
 - backward incompatible change: something that requires a user to apply changes to avoid breaking of a diagnostic / recipe
- important issue: scientific reproducibility; usage of containers?
- Current policy:
<https://docs.esmvaltool.org/projects/ESMValCore/en/latest/contributing.html?highlight=version#backward-compatibility>
- Use of semantic versioning (user might expect backward compatibility e.g. from v2.4 to v2.5 but not necessarily when going from v2.x to v3.0)
- Discussion continued in online discussion forum

General discussion

Notes available at: https://docs.google.com/document/d/11HqxVI4-akeu4NDyPKfc1GvitLYNmzdKuTbSxj_etU/edit#heading=h.rbxyhb2pxzmi

Handling of members (2-factor authentication, inactive members, communication)

2-factor authentication

- 2-factor authentication: encouraged but not enforced
- send email to all members with instructions

Inactive members

- Non-active private members cannot easily be removed
- Non-active members are problematic for maintenance of recipes and diagnostics
- Non-active members are authors forever?

- How to identify non-active members?
- Some members may linger
- The list of authors is nice to keep track of contributions, also non-code contributions.
- There may be a difference between a maintainer not being there, and people having access at all.
- Suggestion: send a message to a maintainer to ask if they are still a maintainer.
- Also important for keeping tracking of number of users and developers
- Suggestion:
 - **AP, every year we contact inactive members and if no response move to an “inactive team” (and put a deadline on that and note that they are always happy to come back.**
- Perhaps we should add maintaining recipes to the consortium agreements
- Important to list duties of a maintainer (and that there is help)

Communication

- How do we communicate with users and developers (other than via GitHub). Also to gather some information on who is using ESMValTool for what projects.
- We need an announcement channel
- Perhaps used LinkedIn and/or Twitter
- **AP, we will use and promote the user/announcements mailinglist.**

Error “tolerance” of the tool

- Should ESMValTool be really pedantic about any errors, or should it be “garbage in garbage out”. How do we find a happy medium?
 - Warnings will most likely be ignored
- How about a scheme where the user can be more/less strict with warnings?
- Having a discussion in an issue on a specific example helps.
- Depends on the usecase.
- Stricter adherence to standards makes maintenance easier.
- The preprocessor also starts to malfunction if you give it non-standard input.
- Examples
 - Error when a recipe does not have a title
 - Almost-matching variable long names for multi-model statistics
- Iris is often very strict in this, perhaps we can relax this a bit. Causes problems with picking an output name.
- This is perhaps more an input data problem
- Should we ask the science team to check changes to ESMValCore?
- AP, we will create these “usability pedantic” issues by 20-01-2022
- AP, The Tech core team will look at the list to find commonalities

Teams, recipe maintainers, reviewers

- We have a hierarchy of teams that are not consistent
- What exactly are the tasks and duties of a recipe maintainer?
 - Author = person that created the recipe.
 - Maintainer = volunteer to ensure recipe will keep on being functional.
 - It works
 - Preprocessors used makes sense

- If a recipe breaks have a first look to see if you can fix it, otherwise ask for help.
 - Are recipe contributors always maintainers?
 - Do we have a supported status for a recipe?
- Are recipe contributors always maintainers?
- Do we have a supported status for a recipe?
- Could we split technical vs scientific maintenance, as in a change to a preprocessor breaks a recipe?
 - It's usually mostly science stuff.
- We need an adopt-a-recipe plan.
- AP, we should writeup the duties of a maintainer (see above)
- How do we get a new maintainer for a science field?
- Maybe we need maintainer and/or science training?
- What do we do with unmaintained recipes?
- We should communicate changes better with recipe maintainers
- We should update recipes that no longer function due to data not being available, and mark
- How we should handle legacy recipes?
- Old recipes and diagnostics are important also so others can learn

Optimization of the tool's performance

- there is still potential for improvements

Other topics

Recipe testing

- Plan is to create a handful of testing recipes
- Now we need a testing framework to run these recipes (e.g. once a month)

Cross team meeting results

- See "Side meeting: cross team meeting" below

Side meeting: user engagement team

The following topics were discussed:

- Discussion on how to keep the tutorial up to date with the developments and releases of the ESMValTool:
 - Versioning the tutorial
 - Updating tutorial with release candidates for ESMValTool/Core so that the tutorial gets updated as much as possible alongside changes in ESMValTool/Core
 - Creating a release schedule for the tutorial with a new version of the tutorial being released roughly two weeks after the ESMValTool release
 - Where possible request the Technical team to open an issue and tag the User Engagement Team when a new interesting feature is added to the tutorial.
- Current necessary updates to the tutorial for ESMValTool v2.4:

- fixes and updates (including suggestions from users for improvements) are worked on within the next few months
- Updates to the documentation: Student helper will start working at DLR for three months to work on this. Will be active on GitHub, so if any ideas suggestions from the User Engagement Team come up, please collect them that they can be considered.
- One member of the User Engagement Team will engage with FAIR4RS and participate as a representative of ESMValTool once they give us an idea of the time commitment required.
- A couple of ideas were floated to track and record funded projects and groups that use ESMValtool

Side meeting: pinning dependencies

- Move to conda-forge for dependencies makes sense as we moved from conda to conda-forge for the ESMValTool installation
- requires putting some Python packages not yet available on conda-forge to conda-forge
- this would add some work to maintain these packages
- Klaus compiles a list of such packages so we can get an overview on the additional maintenance work

Side meeting: cross team meeting

(Alistair, Ranjini, Birgit, Klaus, Axel)

Action items from last meeting

Ranjini

- User Engagement Team is fine with sharing notes publicly
- Google doc for people to comment on backward incompatible changes setup

Alistair

- Agree framing of discussion: leave open as there is more work to do
- schedule meeting in early-mid December: maybe not needed; move meeting to January (new AI for Alistair)

Next steps for backward compatibility

- would be good to get a discussion document out to workshop participants and wider ESMValTool community
- Outlining the problem before addressing solutions, maybe different sections in discussion document: (1) definition (version stability, reproducibility, maintenance), (2) solutions, (3) related issues
- Dependencies require an extra section
- Important link to testing strategy; possible proposal: work on automatic testing could become a priority
- Strategy for maintaining recipes --> discussion in the afternoon including tasks expected from recipe maintainers
- Discussion will be restarted from time to time
- Communicating / requesting input to document:

- User Engagement Team
- mailing lists + issue / discussion

Decide whether we have capacity yet to start an organised discussion on another issue

- Rather not, but including topics that touch the non-backward compatibility issue (e.g. communication, testing)

Gather some views on the desirability (or not) of having an interim steering group

- Difficult topic as long as institutes did not commit with resources
- “interim committee” could propose tasks and be in charge of getting things moving but no official power to make decisions
- Topic will be brought up in general discussion

Agenda Technical ESMValTool Workshop

(via video call, all times are given in CEST)

23 November 2021 - Day 1 (Tuesday)

10.15 am – 10.20 am Introduction, welcome and workshop agenda (Birgit)

10.20 am – 10.30 am Update and current status of the consortium agreement (Alistair)

10.30 am – 12.30 pm ESMValTool v2.4 and new features (Klaus)

- a. Introduction to v2.4 (Klaus)
- b. Downloading of missing files & reusing preprocessor output (V)
- c. Installation using mamba (V)
- d. Using X first/last years (Saskia)

12.30 pm – 01.30 pm Lunch break (opportunity to virtually hang out with fellow workshop participants)

01.30 pm – 03.30 pm Development of scientific test recipe(s) (including writing the test recipes and setting up testing infrastructure) (Axel, Birgit)

- a. Introduction prototype using dummy data (Emma)

02.30 am – 03.30 pm Side meeting: User Engagement Team (Ranjini)

03.30 pm – 04.00 pm Coffee break (opportunity to virtually hang out with fellow workshop participants)

04.00 pm – 04.30 pm Summary and wrap up of first day (Birgit)

24 November 2021 - Day 2 (Wednesday)

10.30 am – 11.30 pm Generic diagnostics and reading / regridding of native model output (Manuel)

11.30 am – 12.30 pm Side meeting: How to pin dependencies (Klaus)

12.30 pm – 01.30 pm Lunch break (opportunity to virtually hang out with fellow workshop participants)

01.30 pm – 02.30 pm Strategy for improved support of regional models (Klaus)

02.30 pm – 03.00 pm Coffee break (opportunity to virtually hang out with fellow workshop participants)

03.00 pm – 04.00 pm Discussion of preliminary test recipes (Birgit, Axel)

04.00 pm – 04.30 pm Summary and wrap up of second day (Axel)

25 November 2021 - Day 3 (Thursday)

10.30 am – 12.00 am Strategy for handling and communicating implementation of non-backward compatible changes with the aim of minimizing unnoticed breaking of recipes new changes (Alistair)

12.00 am – 01.00 pm Cross team meeting (Alistair)

01.00 pm – 02.00 pm Lunch break (opportunity to virtually hang out with fellow workshop participants)

02.00 pm – 03.00 pm General discussion (Niels)

- a. Handling of members (2-factor authentication + inactive members + communication)
- b. Error “tolerance” of the tool
- c. Teams, recipe maintainers, reviewers
- d. Optimization of the tool’s performance

03.00 pm – 03.15 pm Coffee break (opportunity to virtually hang out with fellow workshop participants)

03.15 pm – 04.15 pm General discussion - continued (Niels)

04.15 pm – 04.30 pm Wrap up (Alistair, Birgit)

The workshop was funded by the EU Horizon 2020 research and innovation programme under the grant agreement No 824084 (IS-ENES3 project).