

MEMO: Proposal for a light Taskforce revision process

Motivation

Although the work of the Task Force on Detector Layout has been concluded with the delivery of the output documents and of the external review report, some minor adjustment of the output documents can be foreseen following the interaction with stakeholders, in order to fix residual errors and improve the quality and clarity of the output documents for their use in the engineering studies on infrastructure. Major revisions of the Task Force output documents are not foreseen. A more detailed and advanced technical design of the ET detector will be the subject of forthcoming activities to be organized in a different structure.

Revision intervals

Following a 'snowball' release scheme, errors and small changes to the subject matter would be accumulated and revised until the batch of changes is deemed large enough to warrant an intermediate release. Could be implemented with more frequent releases starting out, then modifying to suit project needs and rate of error/change discovery.

Note: The current 'FINAL' release (Jun 27th, 2025) would be considered release no. 1 – as it is the baseline work.

Release practical considerations

Release package and document naming

For the chosen 'snowball' release scheme, an incremental version suffix (_Vxx; _Rxx) seems the most convenient choice. _Rxx would be used for 'live' docs/annexes, _R01, _R03 etc. for released versions. Examples:

Package: [Package name]_R04

Annexes: TAB9_ET-2L Optical Layout output table_Rxx.pdf, ET Triangle Detector Layout_R02.dwg

Package publication

Packages are released on EDMS, once the first version has been uploaded, with a release notification sent to the ET-ALL mailing list.

Documents affected

Documents subject to revision include the *ET Baseline Detector Layout (Task Force main document)*, *Supporting Document for the ET Baseline Detector Layout (Task Force supporting document)* and the bundle of technical annexes: output tables, technical drawings, risk tables, etc. Full list of supporting documents mentioned in Main and Supporting documents.

Change documentation

Release notes/logbook should be added as a simple document to the release package. Proposed to compile a simple logbook as text document from Gitlab (see next sections). Full traceability logged by Gitlab.

Revision process using Gitlab

It is proposed to base the light revision process around the (pre-existing) Gitlab environment:

- change proposals will be collected from both
 - o internal input: Task Force members identify bugs or inconsistencies in the current version;
 - o stakeholders: consistency issues identified by Local Teams in the use of Baseline Detector Layout for infrastructure engineering, and addressed during periodic meetings between Local Teams and ETO Engineering Department in the presence of ETO Technical Coordinator; issues extracted from the report of the ETC review team; issues arising from the use of Baseline Detector Layout within CERN engineering contracts, and collected by the ETO ED.

- changes will only be implemented for either solving inconsistencies and minor typos, or to substantially improve the efficiency in the use of technical information in engineering studies by stakeholders. Design choices are not subject to the light review process.
- The Task Force should then perhaps process only change proposals that are posted in GitLab (like <https://gitlab.et-gw.eu/et/eto/eto-task-force-detector-layout-2025/-/issues>, possibly splitting the internal and external spaces).
- Small textual/consistency/naming errors should not be traced individually, but bucketed into general modification statements – it should be possible for these to be processed independently by section/table/drawing owners.
- Larger changes in consistency/traceability by preference to be documented independently as errors
- Add Gitlab labels for 'writing/consistency error', 'technical amendment' or similar to distinguish feedback on subject matter from purely writing/consistency errors.
- Add Gitlab labels with origin of change might be useful (e.g. 'documentation', 'Annex: output table' etc.) to help localize the issue.
- Use already existing Gitlab domain labels to categorize also expertise domains (easier assignment)?
- Drawing, document section, (output) table owners accept/reject change proposal with rationale through GitLab (regular process also used for OPT/DET layout exercises)
- Modification to document/technical annex done.
- End of cycle: GitLab logbook printed into release notes: 'Subject Matter changes' and 'Textual/consistency changes'.
- Proposed no contributions to the logbook from Overleaf and Google sheets logs – too distributed and extensive (any 'space' added for example already adds to the logbook).

Practical: how to print Gitlab logbook for a new revision – full cycle

- 1) After previous release, make a new release on Gitlab (<https://docs.gitlab.com/user/project/releases/>)
- 2) Transfer open issues from old to new Gitlab revision (in part this should happen automatically)
- 3) Throughout cycle: close issues while resolving.
- 4) Agree new release: check all open issues, close finished ones
- 5) At end of cycle: go to Issues tab → filter for **closed** issues in **current** release (see Fig. 1), if useful add filter here for labels 'writing/consistency error' and 'subject matter feedback' for easy separation in logbook

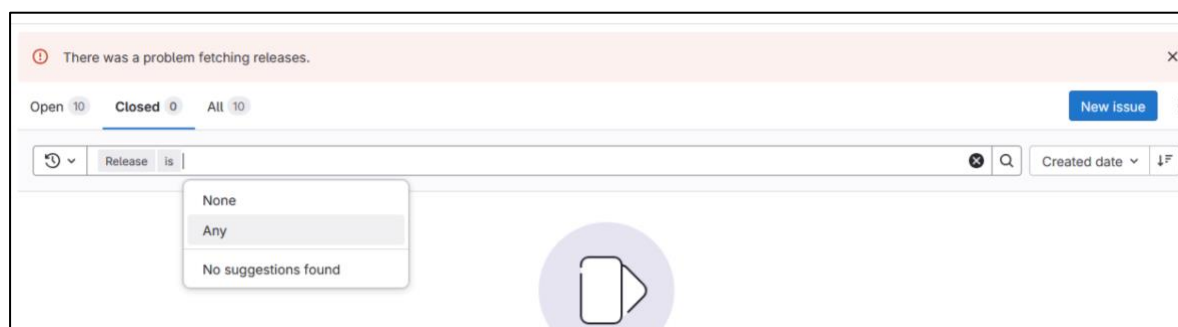


Fig. 1: Gitlab filter settings in 'Closed' issues tab for compiling release logbook. In the dropdown, revision numbers should show up once releases are made following point 1).

- 6) Click dots next to 'New issue': export as CSV. E-mail with CSV will be sent to the person compiling.

- 7) Open CSV in Notepad and trim to retain only the minimum useful info for the logbook: Issue title, closing statement (*need to add to template*), possibly manual # (no need for additional info here as traceability is anyways retained in Gitlab).
- 8) Paste into logbook that is supplied within release package