**3GPP Conference Call on 3GPP Spec Modernization #1 6GSM-250041**

**Electronic, 6th August 2025, 13:00-15:00 UTC**

**Source: Huawei**

**Title: pCR on advantages of current tools (clause 4.1)**

**Document for: Approval**

**Agenda item: 5.1**

**Spec: 3GPP TR 21.802**

**Version: TR version 0.0.0**

**Work Item: FS\_6GSpecs**

**Comments**

This pCR provides text proposals for creating an annex of draft TR 21.802. This annex will include a survey of specification formats, tools and processes in use as of 2025 by 3GPP WGs for specifications and CRs.

**Proposed Changes**

\* \* \* Start of Changes \* \* \* \*

Annex B:  
Survey of specification formats, tools, and CR processes in use by 3GPP WGs in 2025In 2025, 3GPP WGs are using Word for writing specifications and for working on CRs/pCRs in meetings. Additional tools (e.g. Visio, MSC generator, PlantUML) are used for crafting objects that are then embedded in Word. For some WGs, such as RAN1 and RAN2, specifications are published only using Word with embedded objects. In order to address specific needs of certain WGs, some WGs use additional file formats for writing annexes or attachments to the specifications (e.g. YAML file/codec codes).

Table B.1 below summarizes the various tools in use as of 2025 in 3GPP WGs. When only Word is mentioned, it is implied that built-in tools for handling equations, tables, and for drawing figures are also commonly being used. The table indicates when parts of the specifications are also stored on 3GPP Forge (https://forge.3gpp.org), while when not indicated the storage of the specifications is only on 3gpp.org/ftp/Specs also accessible via the 3GPP portal 3GU (https://portal.3gpp.org/).

Table B.1 – Survey of formats and tools in use as of 2025 by 3GPP WGs for specifications and CRs

|  |  |  |
| --- | --- | --- |
|  | **Specification formats/tools** | **Formats/tools/processes used for CRs** |
| CT1 | Word (Visio and MSC-GEN for figures). XML files are included as annex in the Word file of a TS. OpenAPI files are included as .yaml files (Notepad++) in the ZIP file of a TS, and the OpenAPI text is also copied into the Word doc of that TS in annexes. A YAML file is also stored as a fixed branch in Git on 3GPP Forge. These 3 versions must be identical.  3GPP Forge for OpenAPI YAML files: OpenAPI descriptions are extracted from the annex of the 3GPP Technical Specifications and made available as stand-alone YAML files, identified by a file name composed of the API name prefixed by the TS number of the specification containing the OpenAPI description. All these files are then stored in a common repository managed by Gitlab on the 3GPP Forge for testing.  YAML text parser (Notepad++) to generate YAML file (note that new swagger version does not support saving of YAML file and cross checking).  OpenAPI YAML syntax checker (Swagger tool)  XML syntax checker/validator.  CDDL and JSON syntax checker/validator. | Word + Visio  regex101: used to build, test, and debug regex, to check regular expressions online.  CRs are submitted in 3GU, stored on 3gpp.org/ftp, and after the meeting agreed CRs are tested using Git in 3GPP forge:  1. A company submits a CR for an OpenAPI using Word in 3GU, showing track changes to the YAML text copied from the Word annex of a TS with OpenAPI  2. CT1 discusses CRs for that OpenAPI during the meeting. The CRs agreed during the CT1 meeting don’t include a separate .yaml file but just a Word file.  3. After the CT1 meeting, the TS rapporteur merges all agreed CRs for that OpenAPI into a .yaml file (using NotePad++), as part of drafting the TS update.  4. The TS rapporteur checks for syntax errors in Swagger, and provides the checked .yaml file to 3GPP Forge and on FTP draft folder (and also use Github e.g. to detect issues with APIs referring to each other). If CR implementation errors are found the TS Rapporteur proposes a correction and asks Source companies if they agree and asks for a revision to next meeting/plenary.  5. Delegates review the comments from rapporteur, .yaml files and draft TS for correctness and provide feedback on the email reflector.  6. For CRs which are agreed by CT1 and need a correction, the source companies provide a company revision to next plenary with an indication corrected due to errors found during pre-implementation and checking.  7. Once the CRs (including Word docs and .yaml files) are approved in plenary, TS rapporteur provide final version of TS and .yaml files. MCC is doing a final check (e.g. running their Macros) and TS is uploaded to ftp/Specs and the Git branch is fixed and cannot be updated anymore by delegates/rapporteurs. MCC is creating a new draft branch for next plenary for testing.  Similar is valid for correcting or changing a specification’s text, with the difference that Git is not used for text other than .yaml files. |
| CT3 | See CT1 | See CT1 |
| CT4 | See CT1 | See CT1  In deviation to CT1-Step1 as described above, forge is already used when drafting a CR. The author shall check and correct errors early, list the impacted API.  In CT4, the use of the field function provided by Word and foreseen in the template is forbidden. |
| CT6 | Word (Visio for figures) | Word + Visio  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| RAN1 | Word, Excel (in some cases for TRs)  Excel (included as attachment in a single zip file for some TRs). The formula tool built in Excel is used for calculating e.g. link budgets. | Word  CRs are submitted in 3GU, stored on 3gpp.org/ftp  In RAN1 (and some other RAN WGs such as RAN2), TS rapporteurs are tasked at certain stages with providing a CR for their responsible TS which merges together all or many of the changes agreed during a RAN1 meeting relating to a given release, and these editor’s CRs are reviewed by delegates before submission to RAN plenary. TS rapporteurs are not tasked with providing a draft TS merging all approved CRs for their TS. This is done by the MCC officer (RAN1 secretary) after RAN plenary for all TSs under RAN1 responsibility. Draft TSs are provided for review by delegates after which the new version of the TSs is published. |
| RAN2 | Word (Visio and MSC-GEN for figures)  ASN.1 syntax checker (not included in spec) | Word (Visio and MSC-GEN for figures)  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| RAN3 | Word (Visio and MSC-GEN for figures)  ASN.1 syntax checker (not included in spec) | Word (Visio and MSC-GEN for figures)  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| RAN4 | Word, Excel (in some cases)  At least one TR (37.941) includes multiple Excel spreadsheets in separate files in the same zip file | Word, Visio  CRs are submitted in 3GU, stored on 3gpp.org/ftp  RAN4 use separate tool for the CA/DC band combinations requests (Excel so far; database under construction) |
| RAN5 | Word, Excel, Visio  Excel and Visio (included as attachment in a single zip file, and implemented as part of TR 38.903/38.905). The formula tool built-in Excel is used for calculating e.g. uncertainty values.  Machine processable files specified in TTCN-3 language (.ttcn) are provided as attachments in the zip file of a TS, and are not embedded in the Word doc of a TS. Several TTCN-3 compilers are used for syntax checking the TTCN-3 files. | Word  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| SA1 | Word | Word  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| SA2 | Word, Visio | Word, Visio  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| SA3 | Word  Machine deliverable parts (.asn, .xsd) are provided as attachments in the zip file of a TS, and are not embedded in the Word doc of a TS. | Word  CRs are submitted in 3GU, stored on 3gpp.org/ftp  SA3-LI SWG:   * Word + Visio * 3GPP Forge for ASN.1, following these instructions: <https://forge.3gpp.org/rep/sa3/li/-/wikis/How-To/Create-a-CR>   A summary of the process used with 3GPP Forge by SA3-LI is provided below:  Changes to 3GPP specifications still have to go through the 3GPP change control procedure, even if the changes are held on the Forge. A CR can refer to the Forge for any changes to the machine deliverable parts, rather than writing them out in a change-marked Word document.  SA3LI puts each CR and each plenary meeting in its own branch. A CR entered as a branch in 3GPP Forge requires to first obtain a CR number from the 3GPP Portal, and to finally be submitted to a WG meeting with a proper CR form. A CR branch follows the branching convention and a name of the form cr/{deliverable/{CR number}, and it is placed in the branch for the plenary meeting that might eventually approve the CR. Revisions of a CR are possible during a WG meeting, and the corresponding commits must be updated along with increasing the CR revision.  One CR branch can include one or more commits, each with a commit hash, which is a unique and durable identifier of the changes in the Forge. This hash (or the URL associated with the commit hash) is to be copied in the "Comments" section of the CR, along with the merge request number.   * 3GPP Forge for storing Visio diagrams and their revisions, following these instructions: <https://forge.3gpp.org/rep/sa3/li_diagrams>   Visio diagrams are embedded in the specification Word file. 3GPP Forge is used only as a repository for Visio diagrams that can be downloaded for editing. There is no preview for this type of file within 3GPP Forge. A naming convention is defined allowing for referencing each Visio diagram to its specification and specification release and version. |
| SA4 | Word (Visio and MSC-GEN for figures)  3GPP Forge for stage 3 openAPI (YAML file/codec codes maintenance) - cf further details as described for CT1 | Word  CRs are submitted in 3GU, stored on 3gpp.org/ftp |
| SA5 | Word (PlantUML for figures)  3GPP Forge for stage 3 openAPI - cf further details as described for CT1 | Word (PlantUML for figures)  CRs are submitted in 3GU, stored on 3gpp.org/ftp  3GPP Forge for stage 3 openAPI, XSD, andYANG data models. SA5 YANG data models and APIs are validated by an automated pipeline that runs for every push or merge-request event. This pipeline is composed of two stages, validation and generation. Generation stage runs an automated Word CR text generation for the corresponding merge-request.  The generated Word CR text is a changed marked word document that includes all changes made by the associated merge-request. Each modified file (YANG model, OpenApi or XSD) will be included as a change in the Word CR text. This can be included in the Word Change Request document. The Word CR text can be downloaded from the merge-request webpage or the pipeline-page as described above for the detailed log output files. |
| SA6 | Word | Word  CRs are submitted in 3GU, stored on 3gpp.org/ftp |

In addition, MCC provides a number of file templates that are to be used by delegates when drafting CRs, specifications and other document types that are prepared in Word format. These templates are regularly updated and available in meeting-specific folders such as <https://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_108/Templates>. Some templates are also available at <https://www.3gpp.org/ftp/Information/All_Templates>.

MCC also provides a number of tools (Macros) to help with using 3GPP Styles, available at the same link as above. "Unofficial" macros are also used by some groups for specific purposes e.g. ASN.1 review in RAN2

For specific purposes, such as for the collection of evaluation results in an excel worksheet, other templates may be provided by the rapporteur in the course of a study or work item.

\* \* \* End of Changes \* \* \* \*