



**ERICSSON**

# TITAN 10.1.0 CHANGE LOG (RELATIVE TO 10.0.0)

2024-04-25

# HIGHLIGHTS



- C++ compiler
  - Bugfixes ([592](#), [695](#), [703](#), [705](#), [707](#), [708](#), [709](#), [710](#), [711](#), [717](#), [722](#), [725](#))
  - Standard uplift ([600](#), [626](#), [634](#), [635](#), [639](#), [641](#), [645](#), [646](#), [648](#), [666](#), [668](#), [669](#), [670](#), [671](#), [675](#))
  - PER Encoder ([718](#), [719](#), [720](#), [721](#), [724](#))
    - Standalone ASN.1 encoder generation ([712](#), [714](#), [715](#), [723](#))
- Eclipse Plug-Ins
  - Bugfixes ([515](#), [516](#), [517](#))
- Language Server & NextGen Plug-Ins (VS Code, Theia, LS Eclipse)
  - Bugfixes ([12](#), [14](#), [15](#), [22](#), [28](#), [30](#), [31](#), [32](#), [33](#), [35](#), [37](#), [39](#), [40](#), [41](#), [42](#), [43](#), [45](#))
  - General stability and robustness improvements
- Usage of WSL2 as Windows alternative

# STANDARD UPLIFT



- This effort aims to uplift Titan's TTCN-3 standard compatibility to the latest version
- The current supported language standard is 12 years old
- New development would be hard as old standard versions lack the latest language features

# STANDALONE ASN.1 ENCODER



- Titan now supports native ASN.1 PER (Packed Encoding Rules) encoding.
- With this feature anyone can generate ASN.1 defined data structure encode/decode C/C++ code to use in any project requiring the functionality.
- Apart from the new PER encoder feature the other standard encoding rules (BER, OER, PER, XML, json) are also supported in standalone mode
- Next planned phase is to provide code generation for projects in other languages (e.g. Java, C#, etc.)



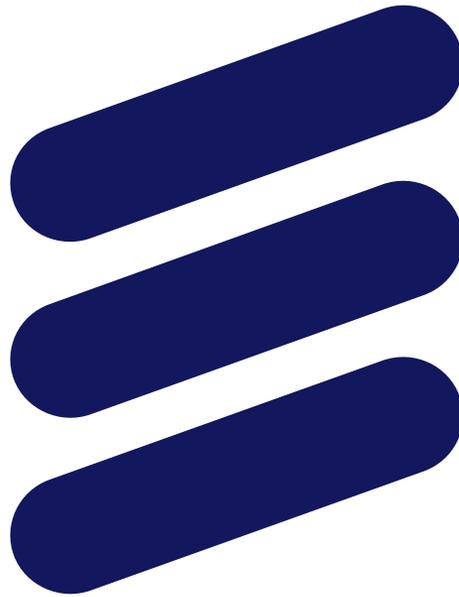
# USAGE OF WSL2 AS WINDOWS ALTERNATIVE

- Having the WSL extension for VS Code makes the work with the Titan toolset quite comfortable. The guide is available [here](#)
- According to the Managing extensions section, the Eclipse Titan extension be installed for WSL as well.
- The command line toolset is available directly via the Terminal of VS Code (make sure you have set the necessary environmental variables properly, such as TTCN3\_DIR).



# SELECTIVE CODE SPLITTING

- Made the -U option (option for code slitting) positing dependent for both makefile\_gen and compiler applications
- The ASN.1 or TTCN-3 modules listed after the -U option are split according to the specified mode
- Made the -U option compatible with the '-' (selective code generation feature)



**ERICSSON**