



ERICSSON

TITAN 9.0.0 CHANGE LOG (RELATIVE TO 8.3.0)

2023-05-17

HIGHLIGHTS



- C++ compiler
 - Bugfixes ([584](#), [591](#), [601](#), [604](#), [608](#), [611](#), [612](#), [636](#), [673](#), [676](#), [680](#), [688](#), [689](#), [690](#), [692](#), [696](#), [697](#))
- Eclipse Plug-Ins
 - Bugfixes ([507](#), [508](#), [509](#), [510](#))
 - IDE enhancements ([417](#), [427](#), [439](#), [458](#), [477](#), [480](#), [484](#), [488](#), [489](#), [491](#))
- New EXPERIMENTAL features
 - Language Server
 - NextGen Plug-Ins: VS Code and Theia extensions

TITAN IDE REWORK



- As You may have noticed on the previous page the former two main product lines are being reworked into a more versatile structure
- The Titan Core (C++ compiler) is still with us featuring C++ based compilation and fast runtime environment
- The former Eclipse Plug-Ins IDE environment is now decoupled from the monolithic Eclipse IDE environment
 - The business logic (source parsing, syntax tree building, marker reporting, Titanium, etc...) will be lifted out to the Language Server supporting LSP protocol
 - Thanks to the versatility and quasi-standard position of the LSP protocol plug-ins can be provided to many IDEs
 - We provide official LSP-based plugins for Eclipse IDE, Visual Studio Code
 - The same plug-in will be used to build an Eclipse Theia based specialized Titan IDE, expected in the next release in Autumn 2023
 - The current monolithic Eclipse Plug-Ins is expected to be discontinued by the next release

BACKWARD INCOMPATIBLE CHANGE REGARDING OPTIONAL FIELDS IN ASN.1



[Issue 696](#) introduced a backward incompatible change. Extension additions are now treated as optional fields, when imported into TTCN-3 modules, and in the generated C++ code. The ASN.1-based codecs (i.e. BER, OER and PER) still only treat extension additions as optional, if they are declared so in the ASN.1 type definition (i.e. encoding/decoding extension additions will have the same results as before).



ERICSSON