

# TITAN 6.2.PLO CHANGE LOG

2017.05.22



- › Note: the list is not comprehensive; for all artifacts and bugs, see tables at the end.
- › See also  
[https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=ALL%20%3ATitan&list\\_id=15328562](https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=ALL%20%3ATitan&list_id=15328562)



- › New compiler options:
- › -J: Compiler (and xsd2ttcn, makefilegen) option to read input files list from a text file
- › -N: ignore UNTAGGED encoding instruction on top level unions (legacy behaviour)

# LIST OF MAIN NEW FEATURES AND FIXES (1)



## › Core:

- › **-support of encvalue/decvalue for ASN.1 types**
- › **-support for implicit call of PER codec external functions**
- › **-fixed: missing namespace for generated custom codec functions**
- › **-implemented: ports with translation capability**
- › **-support for concatenation of templates**
- › **-implemented 'any from' clause and index redirects with the use of the @index modifier (see standard , chapters 21-23)**
- › **-support for dynamic erroneous attributes**
- › **-implemented @fuzzy support**
- › **-support for external functions for decmatch and @decoded**

# LIST OF MAIN NEW FEATURES AND FIXES (2)



## › Core

- › -defining enumerations using binary or hexadecimal values instead of integer values
- › -using negations in template ranges
- › **-compiler option -J to read input files list from a text file**
- › -support of "\N{typereference}", "\N{reference}", "\q{group,plane,row,cell}" pattern notation (see standard B.1.5.4 Match a referenced character set)
- › -support of "-infinity", "infinity", "not-a-number" in arithmetic expressions
- › -reduce compilation times and produce cleaner code by removing or moving unnecessary includes from Titan headers
- › -suggest make clean when wrong compiler version detected
- › -support for use of constants in variant definitions
- › -support for indexing 'of' types or arrays with a record of or array of integer
- › -Raspberry GPIO TestPort
- › **-No support of Solaris binaries from Titan 6.2.0**
- › -'UNIVERSAL tagclass' warning changed to note
  
- › -fixed: compilation error for 'with { extension "done" };'
- › -fixed: Incorrect code generation of parameterized templates
- › -fixed: already mapped/connected ports should not be allowed to be connected/mapped
- › -fixed: it is possible to send a message to SUT without the port to be mapped previously if the port is declared with address
- › -fixed: Incorrect code generation of select statements with goto
- › -fixed: Prototype function value param does not defaults to 'in' param

# LIST OF MAIN NEW FEATURES AND FIXES (3)



## › makefilegen

- › **-makefilegen more restrictive on name attribute of the referenced project**
- › **-Makefilegen: remove generated headers dependency from all .c .cc files**
- › **(This will revert the following bugs:**
- › **Bug 499963 - The generated makefile does not make full build when -j switch is present**
- › **Bug 512688 - makefilegen: Incorrect .c and .cc compiling rule )**
  
- › -fixed: makefilegen segmentation fault error
- › -fixed: the Makefile generation from tpd should not depend on the order of the tpd
- › -fixed: makefilegen: type code splitting incorrect makefile generation
- › -fixed: makefilegen: possibly incorrect header name generation with centralstorage

## › XML

- › **-XER: allow anytype to be xer enc/decodable**
- › **-legacy switch -N in TITAN for the XER encoding of untagged top level unions**
- › -fixed: XER: untagged top level union should be encoded without a tag when indicated by UNTAGGEED
  
- › -fixed: xsd2ttn: translation error when minOccurs/maxOccurs and minLength/maxLength used to lists
- › -fixed: xsd2ttn: type aliases when restriction/extension has no real effect
- › -fixed: xsd2ttn: special float values mishandled in enumerations
- › -fixed: xsd2ttn: special fixed value of float elements are not handled correctly
- › -fixed: xsd2ttn: The type of the attributes does not follow the changed type name when name conversion is present
- › -fixed: XER: error when encoding union with useType
- › -fixed: XER: Incorrect decision implementation when to generate XER code

# LIST OF MAIN NEW FEATURES AND FIXES (4)



## › JSON

- › **-JSON 'as value' attribute extended for records/sets with one field and for the anytype**
- › -fixed: Printing type of JSON encoding through 'encvalue' is not set
- › -fixed: Erroneous decoding of binary strings encoded in JSON
- › -fixed: JSON encoding of field/element types
- › -fixed: Generated JSON coder functions cause unused parameter warning

## › Eclipse

- › **-"make archive" button in Eclipse**
- › **-support for "make port" command in Eclipse**
- › **-fixed: junitlogger does not escape the special characters in verdict reason**
- › **-fixed: failing license check in Eclipse with new Java versions**
- › **-upgrade to Jung 2.1**

# DETAILS





# ARTF618371 : ENCVVALUE/DECVALUE FOR ASN.1 TYPES



- › Tracker: Task
- › Title: encvalue/decvalue for ASN.1 types
- › Description:
  - › If a value of an ASN.1 type is used in an encvalue or decvalue function, compiler generates a "No coding rule specified
  - › for type <ASN.1 type> error" message
  - › See also: Bug 500221

# BUG 500221 - SUPPORT ENCVALUE()/DECVALUE() FOR ASN.1 TYPES



› [https://bugs.eclipse.org/bugs/show\\_bug.cgi?id=500221](https://bugs.eclipse.org/bugs/show_bug.cgi?id=500221)

- › Currently encvalue() and decvalue() supports TTCN-3 types only. 3GPP and ITS protocols are specified using ASN.1 and PER encoding; thus encvalue() and decvalue() are intensively used in both 3GPP and ETSI ITS test suites.
- › Because ASN.1 types can use different encodings, a similar support is needed, than for TTCN-3 types with user-defined encode attribute strings: defining an external function with bitstring return value (for encoding functions and bitstring inout and out parameters for decoding functions, and either with BER:BER|CER|DER or with PER encoding specification. When BER is specified, Titan built-in codec functions should be called implicitly. When PER is specified, a manually implemented external function should be called, just like user-defined encode strings.
- › It should work both for ASN.1-based values and ASN.1 fields of TTCN-3 types (both cases appear in the test suites).

In fact for PER encoding the codec functions are pretty standard: encode in BER using Titan, call OSS to decode and encode in PER or call OSS to decode PER and encode in BER, then decode BER with Titan. Therefore for PER encoding Titan could easily generate the .cc file containing the Titan BER part and a comment where to call the OSS functions. See the attached example.

# BUG 500221 - SUPPORT ENCVALUE()/DECVALUE() FOR ASN.1 TYPES



- › Implemented support for using encvalue, decvalue, decmatch and @decoded on ASN.1 types.
  - › An encoder external function (using a built-in codec) must be declared for an ASN.1 type for encvalue to work on its values. This function only serves to identify which codec to use when encoding values of the ASN.1 type.
  - › This encoder function can use any prototype, and uses the regular stream type associated with the codec (octetstring or charstring), even though the resulting encvalue call uses the prototype 'convert' and bitstring as stream.
  - › Using encvalue on a value of an ASN.1 type that has multiple encoder external functions with different codecs declared causes an error.
  - › Similarly a decoder external function must be declared for an ASN.1 type for decvalue, decmatch and @decoded to work on its values
  - › Regression test cases added under regression\_test/customEncoding.
- › Also, changed the way PER encoding is handled. It now works similarly to custom encoding, allowing the user to implement their own PER encoder & decoder external functions (see bug 508200).

# BUG 508200 - CALL EXTERNAL FUNCTIONS IMPLICITLY FOR PER ENCODING/DECODING



- › Current situation: when encvalue/decvalue/decmatch/@decoded is called for an ASN.1 instance, Titan will call the auto-generated BER encoder or decoder function with the type of coding specified in an external function declaration for the type. The return value of the encoding and the inout parameter of the decoding function shall be octetstring; e.g.:
- › ASN.1 definition:
- › CAM ::= SEQUENCE {
- ›     header ItsPduHeader,
- ›     cam CoopAwareness
- › }
- › External function declarations:
- › external function fx\_enc\_CAM (in CAM pdu) return octetstring
- ›     with {extension "prototype(convert) encode(BER:BER\_ENCODE\_DER)"}
- › external function fx\_dec\_CAM (inout octetstring stream, out CAM pdu) return integer with {extension "prototype(sliding) decode(BER:BER\_ACCEPT\_ALL)"}
  
- › New features requested:
- › 1) In the external function declarations also allow bitstring as return value and inout parameter; this will change the runtime behaviour only, i.e. add implicit octetstring/bitstring conversion, if the function is called directly as in case of encvalue/decvalue it is done already.
  
- › 2) If the encode and decode attribute value of the external function is "PER" for an ASN.1 type, there shall be a manually implemented C++ function for each such external function declaration. One ASN.1 type shall have at most one external function declaration with an encode and one with a decode attribute. No other attribute values than "BER:..." or "PER" shall be allowed.
- › Example:
- › external function fx\_enc\_CAM (in CAM pdu) return bitstring
- ›     with {extension "prototype(convert) encode(PER)"}
  
- › external function fx\_dec\_CAM (inout bitstring stream, out CAM pdu) return integer with {extension "prototype(sliding) decode(PER)"}
  
- › When an instance of an ASN.1 type is used in encvalue/decvalue/decmatch/@decoded, and the type has an external function declaration with the encode/decode attribute PER, the manually implemented external function shall be called implicitly, the same way, it is done for TTCN-3 types with user defined TTCN-3 encode attribute strings.

# BUG 508200 - CALL EXTERNAL FUNCTIONS IMPLICITLY FOR PER ENCODING/DECODING(2)



- › Both changes have been implemented (plus: the JSON codec can also be used for ASN.1 types, not just BER and PER).
- › Tests added under:
  - › regression\_test/customEncoding,
  - › function\_test/Semantic\_Analyser/encode.

# ARTF780946      DEFINING ENUMERATIONS USING BINARY OR HEXADECIMAL VALUES INSTEAD OF INTEGER VALUES



- › Tracker: DefectExt
- › Title: Defining Enumerations using binary or hexadecimal values instead of integer values:
- › Description:

› The following works:

```
› type enumerated BcmFlags_enum {  
›   e_CAN_BCM_SETTIMER      (1),  
›   e_CAN_BCM_RX_RTR_FRAME  (1024)  
› }
```

› , but the following additions all fail:

```
› type enumerated BcmFlags_enum {  
›   e_CAN_BCM_SETTIMER      (1),  
›   e_CAN_BCM_RX_RTR_FRAME  (1024),  
›   e_CAN_BCM_RX_RTR_FRAME_test1 ('0101'B),  
›   e_CAN_BCM_RX_RTR_FRAME_test2 (bit2int('0111'B)),  
›   e_CAN_BCM_RX_RTR_FRAME_test3 ('FFE'H),  
›   e_CAN_BCM_RX_RTR_FRAME_test4 (oct2int('FFF'H)),  
›   e_CAN_BCM_RX_RTR_FRAME_test5 ('FFFE'O),  
›   e_CAN_BCM_RX_RTR_FRAME_test6 (oct2int('FFFF'O))  
› }
```

› **Fixed as an extension to the standard. See RefGuide 4.34. Defining enumeration fields with values known at compile time**

# ARTF780941 : USING NEGATIONS IN TEMPLATE RANGES



- › Tracker: DefectExt
- › Title: using negations in template ranges
- › Description:
  
- › the following fails:
  - › template integer a:= (1 .. !10)
  - › template integer a:= (1 .. !v\_value10)

# ARTF455649 : COMPILER OPTION TO READ INPUT FILES LIST FROM A TEXT FILE



- › Tracker: Change Requests
- › Title: Compiler option to read input files list from a text file
- › Description:
  - › xsd2ttn has an option to read the list of input file from a simple text file. This comes handy in many cases, as
  - › filenames converted from XSD tend to be long and maximum command line length may be insufficient in some cases.
  
  - › The command line length in case of the Titan compiler is insufficient for example, when semantic checking 3GPP test
  - › suites, e.g. UTRAN\_IWD\_14wk49. It is simply impossible to pass the names of all the files in one command line, i.e. not
  - › possible to semantic check the test suite.
  
  - › The solution would be to allow the compiler reading the path/filename-s from a simple textual file, like xsd2ttn. For
  - › example the -F <filename> key could be used for this (in parallel -F could also be added to xsd2ttn as an alternative
  - › beside -f).
  
  - › **Comment:**
  
  - › **-J flag is used for this purpose in xsd2ttn, compiler and makefilegen.**
  
  - › **(the -J flag has been added in xsd2ttn for uniformity)**



# ARTF724182 : IMPLEMENT "\N{TYPEREFERENCE}", "\N{REFERENCE}", "\Q{GROUP,PLANE,ROW,CELL}" FOR PATTERN



- › Tracker: Task
- › Title: implement "\N{typerference}", "\N{reference}", "\q{group,plane,row,cell}" for pattern
- › Description:
  - › B.1.5.4 Match a referenced character set
  - › EXAMPLE:
    - › type charstring MyCharRange ("a".. "z");
    - › type charstring MyCharList ("a", "z");
    - › const MyCharRange myCharR := "r";
  
    - › template charstring myTempPatt1 := pattern "\N{myCharR}";
    - › // myTempPatt1 shall match the string "r" only
  
    - › template charstring myTempPatt2 := pattern "\N{MyCharRange}";
    - › // myTempPatt2 shall match any string containing a single character from a to z
  
    - › template MyCharRange myTempPatt3 := pattern "\N{MyCharList}";
    - › // myTempPatt3 shall match strings "a" or "z" only
  
    - › template universal charstring MyTemplate3:= pattern "{MyString}de\q{1, 1, 13, 7}";
    - › //matches any character string which consists of the characters "ab", followed by any
    - › //character, followed by the characters "de", followed by the character in ISO10646-1 with
    - › //group=1, plane=1, row=13 and cell=7.

# ARTF725754 : -INFINITY, INFINITY, NOT-A-NUMBER IN ARITHMETIC EXPRESSIONS



- › Comment:
  
- › see also refguide
  
- › 4.15 Special Float Values Infinity and not\_a\_number
- › The keyword infinity (which is also used to specify value range and size limits) can be used to specify the special float values -infinity and +infinity, these are equivalent to MINUS-INFINITY and PLUS-INFINITY used in ASN.1. A new keyword not\_a\_number has been introduced which is equivalent to NOT-A-NUMBER used in ASN.1. The special values cannot be used in arithmetic operations. The special value not\_a\_number cannot be used in a float range subtype because it's an unordered value, but can be added as a single value, for example subtype (0.0 .. infinity, not\_a\_number) contains all positive float values and the not\_a\_number value.
  
- › Comment: +-infinity and not\_a\_number can be used in +/- arithmetic expressions. Every operation with not\_a\_number will result with not\_a\_number

# ARTF721830 : REMOVE OR MOVE UNNECESSARY INCLUDES FROM TITAN HEADERS



- › Tracker: Task
- › Title: remove or move not necessary includes from TITAN headers
- › Description:
  - › This can speed up compilation time of code generated by TITAN.
  - › Some basic techniques are forward declaration of a classes, so the include is not necessary in the header file, and can be moved to the source file.
  - › In some cases like Integer.hh ASN\_NULL.hh and the similars, the Optional.hh can be moved to the .cc file, and a forward declaration can be used for the OPTIONAL class.
  - › In some cases like Float.hh the types from Error.hh are used only in one function, and that function can be moved to the .cc file with the Error.hh include.
  - › For example: Integer.hh includes Rint.hh and is not required to be included in the header, can be moved to the .cc.
  - › These changes may reduce compilation times, and produces cleaner code.

ARTF743325 : WHEN WRONG  
COMPILER VERSION DETECTED,  
SUGGEST MAKE CLEAN



# ARTF802981 : COMPILATION ERROR FOR WITH { EXTENSION "DONE" };



- > Tracker: DefectExt
- > Title: Compilation error for with { extension "done" };
- > Description:
  - > Error:
    - > g++ -c -DLINUX -I/proj/ims\_ttcn3/titan/5R6A.pre1/sles-11.3.x86\_64/include -DTITAN\_RUNTIME\_2 -DUSE\_SCTP -DIPL4\_USE\_SSL -I/app/vbuild/SLED11-x86\_64/openssl/1.0.1g/include -DUSE\_KAME\_IPSEC -DLKSCTP\_MULTIHOMING\_ENABLED -DAS\_USE\_SSL -DTARGET\_TEST -I. -I./src -Wall -o tinyxmlerror.o tinyxmlerror.cc
    - > IceClient\_definitions.hh:38: error: 'IceMessage\_Redirect\_Interface' in namespace 'Ice\_\_definitions' does not name a type
  - > :
  - > Workaround:
    - > The declaration
    - > type IceMessage IceMessageDone with { extension "done" };
    - > in
    - >
    - > IceClient\_definitions.ttcn
    - > Should be moved to the module where IceMessage is declared,
    - > That is , to Ice\_definitions.ttcn .
  - > Comment:
    - > Fixed.
    - > Fault: The namespace prefix was added to the generated value redirect interface class if the type returned by the 'done' statement was a subtype whose original type was defined in a different module.
    - > Solution: The type-specific value redirect interface classes were replaced by one interface class in the runtime library (as part of the 'any from' feature).

# BUG 505866 - IMPLEMENT MAKEFILEGEN PORT COMMAND IN ECLIPSE



# BUG 501007 - SUPPORT EXTERNAL FUNCTIONS FOR DECMATCH AND @DECODED



- › Comment: Currently decmatch and @decoded are working for TTCN-3 types and Titan built-in codecs only, but not for decoding by external functions and not for ASN.1 types. This strongly limits their usability.
- › For example ETSI ITS test suite uses lots of user defined encode strings (that are supported for TTCN-3 types and encvalue/decvalue), or see the email from Sverker Gustavsson dated den 7 september 2016 14:20.
- › decmatch and @decoded should be able to call external decoding functions and decoding functions for ASN.1 types (see proposal in related TR 500221).
- › **Comment: Implemented support for external decoding functions (custom decoding) in decmatch templates, in value redirects and parameter redirects with the @decoded modifier, and in encvalue\_unichar and decvalue\_unichar.**
- › **Tests have also been added for these cases under regression\_test/customEncoding.**
- › **Comment: Implemented support for using decmatch and @decoded on ASN.1 types.**
- › **For more information see bug 500221.**

# BUG 507957 - "MAKE ARCHIVE" BUTTON IN ECLIPSE



- › It will be a project properties menu item instead of a button
- › project properties -> Titan -> Make archive



# BUG 507352 - XSD2TTCN: TRANSLATION ERROR WHEN MINOCCURS/MAXOCCURS AND MINLENGTH/MAXLENGTH USED TO LISTS



› The problem originates from ETSI/oneM2M ATS development.

› When translating the XSD element:

```
› <xs:element name="roleIDs" minOccurs="0">
›   <xs:simpleType>
›     <xs:restriction>
›       <xs:simpleType>
›         <xs:list itemType="m2m:roleID" />
›       </xs:simpleType>
›       <xs:minLength value="1" />
›     </xs:restriction>
›   </xs:simpleType>
› </xs:element>
```

# BUG 507352 - XSD2TTCN: TRANSLATION ERROR WHEN MINOCCURS/MAXOCCURS AND MINLENGTH/MAXLENGTH USED TO LISTS(2)



> embedded in another element, the following TTCN-3 is generated:

```
> type record RequestPrimitive
> {
>   Operation operation,
>   XSD.AnyURI to_,
>   XSD.ID from_ optional,
>   RequestID requestIdentifier,
>   ResourceType resourceType optional,
>   PrimitiveContent primitiveContent optional,
>   record length(1 .. infinity) of RoleID roleIDs,
>   Timestamp originatingTimestamp optional,
>   AbsRelTimestamp requestExpirationTimestamp optional,
>   AbsRelTimestamp resultExpirationTimestamp optional,
>   AbsRelTimestamp operationExecutionTime optional,
>   ResponseTypeInfo responseType optional,
>   AbsRelTimestamp resultPersistence optional,
>   ResultContent resultContent optional,
>   EventCat eventCategory optional,
>   XSD.Boolean deliveryAggregation optional,
>   XSD.String groupRequestIdentifier optional,
>   FilterCriteria filterCriteria optional,
>   DiscResType discoveryResultType optional,
>   DynAuthJWT tokens optional,
>   TokenID tokenIDs optional,
>   record length(1 .. infinity) of XSD.NCName localTokenIDs,
>   XSD.Boolean tokenReqIndicator optional
> }
```

> In this particular case (minOccurs="0" maxOccurs="1" of the element roleIDs) the generated roleIDs field shall be optional.

> In the case when maxOccurs>1, the

> record length(1 .. infinity) of RoleID

> shall be embedded to an outer record of either without length restriction (minOccurs="0" maxOccurs="unbounded" case) or with an appropriate length restriction (see clause 7.1.4 of the Part-9 standard).

# BUG 507270 - XER: ERROR WHEN ENCODING UNION WITH USETYPE



- › When encoding a union which has useType encoding instruction, and it has a union field which has useUnion encoding instruction, the xmlns:xsi is inserted twice, and the xsi:type attribute has wrong value.
- › See last paragraph of: B.3.24 Use type

# ARTF724072 : IMPLEMENT @INDEX



- › Comment:
- › Implemented the 'any from' clause and index redirects (with the use of the @index modifier). See also Bug #498591

# BUG 498591 - IMPLEMENT THE 'ANY FROM' CLAUSE



- › TITAN does not currently support the 'any from' clause, referenced in the TTCN-3 standard in chapters 21-23.
- › The 'any from' feature has been implemented, together with index redirects (with the '@index' modifier).

# BUG 509146 - PRINTING TYPE OF JSON ENCODING THROUGH 'ENCVALUE' IS NOT SET



- › When using encvalue on a type with JSON encoding, the printing type (compact printing or pretty printing) is not specified to the codec, and the resulting JSON document may be compact or pretty at random.
- › This is because the Boolean argument of the 'encode' function is not set when generating code for an 'encvalue' expression, so the JSON codec just reads whatever is in the memory.
- › The printing when using 'encvalue' should be compact.
- › **Change implemented.**
- › **Tests adjusted accordingly.**

# BUG 500731 - MAKEFILEGEN SEGMENTATION FAULT ERROR



- › The attachment produces segmentation fault when executing:
  - › `makefilegen -f -FWrZH -t a.tpd`
  - › If the project hierarchy is circular then it will crash with segmentation fault.
- › Resolved.
- › Forbidden circular references when generating makefiles from TPD with improved linking method.

# BUG 509304 - MISSING NAMESPACE FOR GENERATED CUSTOM CODER FUNCTIONS



- › The namespace prefix for isn't always generated for custom and PER coder functions, when generating code for `encvalue/decvalue/decmatch/@decoded`.
- › More specifically: the namespace prefix is added if the coded type and the coder function definitions are in separate modules, but it should be added when the external function definition and the `encvalue/decvalue/decmatch/@decoded` usage are in separate modules instead.
- › **Reworked the checking of coding methods for types. The coder function's namespace prefix is now determined during code generation, and is based on whether the coder function and the `encvalue/decvalue/decmatch/@decoded` usage are in the same module. Previously the generated code for the coder function call was determined during semantic analysis, and was calculated once per type (not once per `encvalue/decvalue/decmatch/@decoded` usage).**
- › **Test case added to module `regression_tests/customEncoding/Custom1.ttcn`.**



# BUG 509434 - KILL AND STOP OPERATIONS SOMETIMES CAUSE SEGFAULT



- › Performing a 'kill' or 'stop' operation on a component sometimes causes a segmentation fault during semantic analysis.
- › Example for 'kill':  
conformance\_test/core\_language\_test/positive\_tests/21\_configuration\_operations/2103\_test\_component\_operation/210305\_alive\_operation/NegSem\_210305\_alive\_operation\_002.ttcn.
- › Example for 'stop': regression\_test/config\_oper/TconfigOper.ttcn.
- › Fault description:
  - › The implementation of the 'any from' clause and the index redirects introduced new parameters to the 'killed' operation in the AST. Before this feature the operations 'killed', 'kill' and 'done' had the same parameters, and were often grouped together in switches. This caused the compiler to do null-pointer checks on memory garbage, since these new parameters were left uninitialized for operations 'stop' and 'kill'.
- › Solution description:
  - › Separated the 'killed' operation from the other two in all switches in the compiler code.

# ARTF817270 : ALREADY MAPPED/CONNECTED PORTS SHOULD NOT BE ALLOWED TO BE CONNECTED/MAPPED



- › An error will be thrown if an already mapped/connected port is attempted to be connected/mapped. This can potentially trigger backwards incompatible behaviour.
- › Re-mapping or reconnecting an already mapped/connected port is permitted, see Refguide 4.4 Ports and Test Configurations

# ARTF817538 : IT IS POSSIBLE TO SEND A MESSAGE TO SUT WITHOUT THE PORT TO BE MAPPED PREVIOUSLY IF THE PORT IS DECLARED WITH ADDRESS



```
> module NegSem_060212_AddressingEntitiesInsideSut_001_m {
>   type record SIP_address_type
>   {
>     charstring    host optional,      // hostname, IPv4 or IPv6
>     integer       portField optional, // represented as an integer
>     boolean       tcporudp optional // true if TCP false if UDP
>   }
>   type SIP_address_type address;
>   type port PortType1 message {      inout integer;  } with {extension "address"}
>   type component TestCaseComp {      port PortType1 p1;  }
>     testcase TC_NegSem_060212_AddressingEntitiesInsideSut_001() runs on TestCaseComp system TestCaseComp {
>       var address v_addr := {"Host", 4400,false}
>       // map(self:p1, system:p1) !!!!!
>       p1.send(5) to v_addr;
>       setverdict(pass);
>     }
>   control {
>     execute(TC_NegSem_060212_AddressingEntitiesInsideSut_001());
>   }
> }
```

# ARTF190392 : USE OF CONSTANTS IN VARIANT DEFINITIONS CR\_TR00019588



- › In the type definitions you can use the `defaultForEmpty` attribute to define the default values for the associated field
- › during the XER decoding. These values are usually present in form of constants. But in the attribute definition there is
- › no way to reuse these constants, and vice versa, there is no way to reuse the values of `defaultForEmpty` attributes in
- › constants. Therefore the default values must be defined twice. It causes deviations sooner or later.
  
- › There should be a method allowing to declare the default values once only.
  
- › variant “ `defaultForEmpty '5' //literal`
  
- › variant “ `defaultForEmpty myModule.myConstants //reference`

# ARTF640920 : XSD2TTCN: TYPE ALIASES WHEN RESTRICTION/EXTENSION HAS NO REAL EFFECT



- › According to an ETSI CR

<http://forge.etsi.org/mantis/view.php?id=6867> the converter should use type aliases when the restriction or extension tag just references the base type, but makes no change to the base type.

BUG 509855 - JUNITLOGGER DOES NOT  
ESCAPE THE SPECIAL CHARACTERS IN  
VERDICT REASON



# BUG 499534 - TPD BUILD CONFIG SELECTION



- › The Makefile generation from tpd should not depend on the order of the tpd's. If a tpd requests a specific build config of another tpd, that specific build config should be used.

# BUG 509867 - ERRONEOUS DECODING OF BINARY STRINGS ENCODED IN JSON



- › When the JSON string, representing a TTCN-3 binary string (tried with hexstring only), contains other characters than hexadecimal digits, these are "decoded" and inserted into the TTCN-3 value as extra digits. The digits inserted for the incorrect characters also depend if the decoding function has been called to decode another value(s) before.
- › Error code 2 is returned.
  
- › Proposed solution:
  - › - ignore any space (char(U20)), Horizontal tab (char(U9)), Line feed (char(UA)) and Carriage return (char(UD)) characters at decoding (this behaviour is proposed in the draft of the standard)
  - › - any other illegal character should cause a decoding error, i.e. it should cause error code 2, the out parameter should be returned unchanged (unbound if it was unbound before), and the whole input JSON value should be returned unchanged in the inout parameter.





# BUG 510562 - INCORRECT CODE GENERATION OF SELECT STATEMENTS WITH GOTO

› If a goto statement jumps over a select case statement then the generated code will not compile.

› Example:

```
› function f () runs on C
› {
›   var boolean v_result;
›   var charstring pl_messageType := "s";
›   v_result := true;
›   if (not v_result) { goto END_EXECUTION };

›   select (pl_messageType) {
›     case ("SS") {
›       v_result := false;
›     }
›   }
›   label END_EXECUTION;
› }
```

# BUG 510734 - XSD2TTN: SPECIAL FLOAT VALUES MISHANDLED IN ENUMERATIONS



- › XSD:
  - › `<simpleType name="Enum">`
  - ›     `<restriction base="float">`
  - ›         `<enumeration value="INF"/>`
  - ›         `<enumeration value="-INF"/>`
  - ›         `<enumeration value="NaN"/>`
  - ›     `</restriction>`
  - › `</simpleType>`
  
- › Wrong generation:
  - › type XSD.Float Enum ();
  
- › Correct generation:
  - › type XSD.Float Enum (-infinity, infinity, not\_a\_number);

# BUG 510589 - XSD2TTCN: SPECIAL FIXED VALUE OF FLOAT ELEMENTS ARE NOT HANDLED CORRECTLY



› `<xs:element name="FloatFixed3" type="xs:float" fixed="INF"/>`

› generates into:

› type XSD.Float FloatFixed3 (INF)

› with {

› variant "defaultForEmpty as c\_defaultForEmpty\_6";

› variant "element";

› };

› instead of:

› type XSD.Float FloatFixed3 (infinity)

› with {

› variant "defaultForEmpty as c\_defaultForEmpty\_6";

› variant "element";

› };

# BUG 510610 - XSD2TTCN: THE TYPE OF THE ATTRIBUTES DOES NOT FOLLOW THE CHANGED TYPE NAME WHEN NAME CONVERSION IS PRESENT



```
> <xs:element name="MyString_"/>
> <xs:simpleType name="MyString">
>     <xs:restriction base="xs:string">
>         <xs:pattern value="fix"/>
>     </xs:restriction>
> </xs:simpleType>
> <xs:complexType name="Complex">
>     <xs:sequence>
>     </xs:sequence>
>     <xs:attribute name="AttrfixedMyString" type="this:MyString"/>
> </xs:complexType>
> The Complex type generates into this:
> type XSD.String MyString_1 (pattern "fix")
> with {
>     variant "name as 'MyString'";
> };
> type record Complex {
>     MyString attrfixedMyString optional, }
> with {
>     variant (attrfixedMyString) "defaultForEmpty as c_defaultForEmpty_13";
>     variant (attrfixedMyString) "name as capitalized";
>     variant (attrfixedMyString) "attribute";
> };

> Instead of this:
> type XSD.String MyString_1 (pattern "fix") with { variant "name as 'MyString'"; };

> type record Complex {
>     MyString_1 attrfixedMyString optional, }
> with {
>     variant (attrfixedMyString) "defaultForEmpty as c_defaultForEmpty_13";
>     variant (attrfixedMyString) "name as capitalized";
>     variant (attrfixedMyString) "attribute";
> };
```

# BUG 511078 - SUPPORT FOR INDEXING 'OF' TYPES OR ARRAYS WITH A RECORD OF OR ARRAY OF INTEGER



- › The following should work:
- › type record of record of integer RoIRol;
  
- › var RoIRol v\_rec := {{0, 1}, {2, 3}};
- › var integer v\_index[1] := { 1 }
  
  
- › if (v\_rec[v\_index] == {2, 3}) {
- ›       setverdict(pass);
- › } else {
- ›       setverdict(fail);
- › }

# BUG 511903 - DYNAMIC ERRONEOUS ATTRIBUTES



- › Currently erroneous 'with' attributes (used by codecs for negative testing) may only be specified at the declaration of the values or templates they affect, and they may only contain literals or references to constants.
- › For the codecs in TITAN to be used for fuzzing, these erroneous attributes must be dynamically changeable.
- › Suggestion:
- › Implement a new statement called '@update' with the following syntax:
  - › @update(<value or template>) with { <attributes> }
- › This statement would set the erroneous attributes of the value or template parameter to the ones specified in the 'with' branch. Any erroneous attributes the value or template might have (from its declaration or from previous @update statements) will be overwritten.
- › The erroneous attributes in the '@update' statement may contain references to values that are not known at compile-time (e.g.: variables, function calls, etc.).
- › The statement's 'with' branch may not contain any other types of attributes other than erroneous attributes.
- › The 'with' branch may be empty or omitted, in which case the '@update' statement erases any previously set erroneous attributes from the value or template.

# BUG 51180 - RASPBERRY GPIO TESTPORT



ARTF824590 : NO SUPPORT  
SOLARIS BINARY FROM TITAN 6.2.0





# ARTF724042 : IMPLEMENT @FUZZY SUPPORT



- › Implemented the @fuzzy modifier to work on function/altstep/testcase parameters. Currently @fuzzy parameters have mostly
- › the same limitations as @lazy parameters.
- › Further limitations:
  - › - @fuzzy parameters cannot be used on the left hand side (their value cannot be changed). Assigning a new value to a @fuzzy parameter causes it to remain unchanged and retain its initial value. (@lazy parameters can be changed, but the newly assigned expression is evaluated immediately, thus losing the parameter's 'lazy'-ness).
  - › - If a @fuzzy parameter's expression contains a function call that has @fuzzy or @lazy parameters, then the internal function's parameters will behave as regular parameters, losing their 'fuzzy'-ness or 'lazy'-ness (this is also true for @lazy parameters' expressions).
  - › - Passing a @fuzzy parameter as another function's @lazy parameter causes the @fuzzy parameter to be 'converted' to @lazy. Similarly, a conversion is required when passing a @lazy parameter as another function's @fuzzy parameter. These conversions don't always work as intended, if the inner function call is inside an expression that also uses the @fuzzy/@lazy parameter (examples can be found in the newly added regression test folder 'fuzzy').
- › Regression tests added under 'regression\_test/fuzzy'.

# BUG 513762 - XER: INCORRECT DECISION IMPLEMENTATION WHEN TO GENERATE XER CODE



- › The function `XER_encode()` is incorrectly implemented.
- › When a field of a record is not xer encodable it is stored that the field is not xer encodable but it is not stored that the record is not xer encodable.
- › This leads to incorrect code generation which causes GCC to report compilation errors.

# BUG 513749 - CHANGE 'UNIVERSAL TAGCLASS' WARNING TO NOTE



- › The compiler displays a warning, when the UNIVERSAL tagclass is used in ASN.1. This should only cause a note to be displayed instead of a warning.

BUG 514249 - XER: ALLOW  
ANYTYPE TO BE XER  
ENC/DECODABLE



# BUG 514253 - XER: UNTAGGED TOP LEVEL UNION SHOULD BE ENCODED WITHOUT A TAG



› Follow-up on Bug 514249

# BUG 515436 - CREATE A LEGACY SWITCH IN TITAN FOR THE XER ENCODING OF UNTAGGED TOP LEVEL UNIONS



- › -N: ignore UNTAGGED encoding instruction on top level unions (legacy behaviour)

# BUG 515449 - EXTEND JSON 'AS VALUE' ATTRIBUTE FOR RECORDS/SETS WITH ONE FIELD AND FOR THE ANYTYPE



## › **Use case:**

Values of the anytype should be encodable as a JSON value (without the name of the selected field and without the braces). This can be done through the JSON attribute 'as value'. Since the anytype cannot have variant attributes, this could only be done through subtyping (by creating a type alias for the anytype or by using it as a field or element of a structure).

- › Currently the JSON codec only takes the variant attributes of base types into account when coding (subtypes are only considered for records, sets, record ofs, set ofs in runtime 2). This needs to be extended, at least for the 'as value' attribute.
- › The 'as value' attribute should also work on records and sets with one field. In this case the record's/set's encoding should be equal to the field's encoding (the field name and the braces should be omitted).

## › **Change implemented.**

- › The 'as value' coding instruction has been extended to the anytype, and to records and sets with one field.
- › The JSON codec now checks for the attribute 'as value' at runtime, instead of at compile-time. This allows the use of this attribute on subtypes and on fields and elements of structured types.
- › The other JSON attributes are still checked only at compile-time.

# BUG 515584 - JSON ENCODING OF FIELD/ELEMENT TYPES



- › When declaring JSON coder functions for subtypes or the field or element types of structures, the compiler only checks the 'encode' attributes of the base, when trying to determine whether the type supports JSON encoding (variant attributes are checked for the type itself).

› Example:

```
› module json_descr {  
›   type record of anytype Stuff  
›   with {  
›     //variant ([-]) "untagged";  
›     encode ([-]) "JSON";  
›   }  
›   external function f_enc(in Stuff[-] x) return bitstring  
›   with { extension "prototype(convert) encode(JSON)" };  
›   control {  
›     var Stuff x := { { integer := 3 } };  
›     action(bit2oct(f_enc(x[0])));  
›   }}  
›   with {  
›     extension "anytype integer, charstring";  
›   }  
› }
```

- › Here the compiler displays an error, that the input type of 'f\_enc' does not support JSON encoding, even though it has the 'encode "JSON"' attribute (since 'anytype' indeed doesn't support JSON encoding).
- › Giving an XML variant to the type lets it pass the compiler's checks, but a dynamic test case error is issued, when attempting to use the encoder function ('No JSON descriptor available for type ...').  
Updated the JSON encoding check and type descriptor generation. Now 'encode' attributes of subtypes, type aliases and fields/elements of structures are also considered when determining whether a type supports JSON encoding.



# BUG 514156 - MAKEFILEGEN: MORE RESTRICTION ON NAME ATTRIBUTE OF THE REFERENCED PROJECT



- › The name attribute of the referenced project should always be equal to the project name in the referenced .tpd, else an error is issued.

# BUG 514156 - MAKEFILEGEN: MORE RESTRICTION ON NAME ATTRIBUTE OF THE REFERENCED PROJECT (CONT.)



## > Example : EPTF\_Applib\_SDP\_CNL113744.tpd references EPTF\_CLL\_FBQ.tpd:

```
> <?xml version="1.0" encoding="UTF-8"?>
> <TITAN_Project_File_Information version="1.0">
>   <ProjectName>EPTF_Applib_SDP_CNL113744</ProjectName>
>   <ReferencedProjects>
>     :
>     <ReferencedProject name="EPTF_CLL_DataSource" projectLocationURI="../EPTF_Core_Library_CNL113512/src/DataSource/EPTF_CLL_DataSource.tpd"/>
>     <ReferencedProject name="EPTF_CLL_FreeBusyQueue"
> projectLocationURI="../EPTF_Core_Library_CNL113512/src/FreeBusyQueue/EPTF_CLL_FBQ.tpd"/>
>     <ReferencedProject name="EPTF_CLL_HashMap" projectLocationURI="../EPTF_Core_Library_CNL113512/src/HashMap/EPTF_CLL_HashMap.tpd"/>
>     :
>     :
>     <ReferencedProject name="TCCUsefulFunctions_CNL113472_Maths" projectLocationURI="../TCCUsefulFunctions_CNL113472/TCCUsefulFunctions_CNL113472_Maths.tpd"/>
>   </ReferencedProjects>
>   <Files>
>     <FileResource projectRelativePath="EPTF_SDP_Definitions.ttcn" relativeURI="src/LoadGen/EPTF_SDP_Definitions.ttcn"/>
>     <FileResource projectRelativePath="EPTF_SDP_Functions.ttcn" relativeURI="src/LoadGen/EPTF_SDP_Functions.ttcn"/>
>   </Files>
>   <ActiveConfiguration>Default</ActiveConfiguration>
>   <Configurations>
>     <Configuration name="Default">
>       <ProjectProperties>
>         <MakefileSettings>
>           <generateInternalMakefile>true</generateInternalMakefile>
>           <GNUMake>true</GNUMake>
>           <incrementalDependencyRefresh>true</incrementalDependencyRefresh>
>           <targetExecutable>bin/EPTF_Applib_SDP_CNL113744</targetExecutable>
>           <buildLevel>Level 3 - Creating object files with dependency update</buildLevel>
>         </MakefileSettings>
>         <LocalBuildSettings>
>           <workingDirectory>bin</workingDirectory>
>         </LocalBuildSettings>
>       </ProjectProperties>
>     </Configuration>
>   </Configurations>
> </TITAN_Project_File_Information>
```

# BUG 514156 - MAKEFILEGEN: MORE RESTRICTION ON NAME ATTRIBUTE OF THE REFERENCED PROJECT (CONT.)



## › EPTF\_CLL\_FBQ.tpd:

```
› <?xml version="1.0" encoding="UTF-8"?>
› <TITAN_Project_File_Information version="1.0">
›   <ProjectName>EPTF_CLL_FreeBusyQueue</ProjectName>
›   <ReferencedProjects>
›     <ReferencedProject name="EPTF_CLL_Common" projectLocationURI="../Common/EPTF_CLL_Common.tpd"/>
›     <ReferencedProject name="EPTF_CLL_NQueue" projectLocationURI="../NQueue/EPTF_CLL_NQueue.tpd"/>
›   </ReferencedProjects>
›   <Files>
›     <FileResource projectRelativePath="EPTF_CLL_FBQ_Definitions.ttcn" relativeURI="EPTF_CLL_FBQ_Definitions.ttcn"/>
›     <FileResource projectRelativePath="EPTF_CLL_FBQ_Functions.ttcn" relativeURI="EPTF_CLL_FBQ_Functions.ttcn"/>
›   </Files>
›   <ActiveConfiguration>Default</ActiveConfiguration>
›   <Configurations>
›     <Configuration name="Default">
›       <ProjectProperties>
›         <MakefileSettings>
›           <generateInternalMakefile>true</generateInternalMakefile>
›           <GNUMake>true</GNUMake>
›           <targetExecutable>bin/EPTF_FreeBusyQueue</targetExecutable>
›           <buildLevel>Level3-Creatingobjectfileswithdependencyupdate</buildLevel>
›         </MakefileSettings>
›         <LocalBuildSettings>
›           <workingDirectory>bin</workingDirectory>
›         </LocalBuildSettings>
›       </ProjectProperties>
›     </Configuration>
›   </Configurations>
› </TITAN_Project_File_Information>
```

› **<Project name> has to be the same as <ReferencedProject> in the referencing .tpd !!!!**

# BUG 514742 - CHANGE THE OUTPUT OF THE 'MAKE INSTALL', MAKE IT MORE USER FRIENDLY



- › At the end of the 'make install', the 'hello world' mini test project finishes its run by "overall verdict: fail". That concludes misunderstanding. Change this smoke test to be successful. Additionally finish the install with a direct 'Install successfully done' to make users well informed.

The new log of the 'make install' is this:

- › MTC@HU00078244: Verdict statistics: 0 none (0.00 %), 2 pass (100.00 %), 0 inconc (0.00 %), 0 fail (0.00 %), 0 error (0.00 %).
- › MTC@HU00078244: Test execution summary: 2 test cases were executed. Overall verdict: pass
- › MC@HU00078244: MTC terminated.
- › MC2> exit
- › MC@HU00078244: Shutting down session.
- › MC@HU00078244: Shutdown complete.
- › cd /cygdrive/c/Users/ETHBAAT/git/titan.core/Install/demo && make clean
- › make[2]: Entering directory '/cygdrive/c/Users/ETHBAAT/git/titan.core/Install/demo'
- › rm -f MyExample.exe libMyExample.a MyExample.o hello\_world.o PCOType.o MyExample.hh hello\_world.hh \  
MyExample.cc hello\_world.cc compile PCOType.d MyExample.d hello\_world.d \  
tags \*.log
- › make[2]: Leaving directory '/cygdrive/c/Users/ETHBAAT/git/titan.core/Install/demo'
- › make[1]: Leaving directory '/cygdrive/c/Users/ETHBAAT/git/titan.core/hello'
- › chmod -R +r,go-w ./Install
- › \*\*\*\*\*
- › Installation successfully completed
- › \*\*\*\*\*

# BUG 511453 - FIXED: INCORRECT CODE GENERATION OF PARAMETERIZED TEMPLATES



- › The code generation of `isbound()` and `ispresent()` operations are incorrect when the template is parameterized or the operation is executed on a field of the template.

# [BUG 516022](#) - MAKEFILEGEN: REMOVE GENERATED HEADERS DEPENDENCY FROM ALL .C .CC FILES



- › Remove GENERATED\_HEADERS dependency from all .c and .cc files because it possibly slows down the compilation process in some cases.
- › Revert to old behaviour where the **make dep** command should be invoked before **make** to be sure that all generated files are ready.
- › This will revert the following bugs:
  - › [Bug 499963](#) - The generated Makefile does not trigger a full build with **make** when -j switch is present
  - › [Bug 512688](#) - makefilegen: Incorrect .c and .cc compiling rule

# BUG 516022 - MAKEFILEGEN: REMOVE GENERATED HEADERS DEPENDENCY FROM ALL .C .CC FILES (CONT.)



- › “**make dep**” has to be used before “**make**”;
- › Else the first “**make**” will just create the dependencies and has to be issued again for the build to take place

# BUG 512878 - CONCATENATION FOR TEMPLATES



- › Implemented template concatenation with the following restrictions:
- › - templates can only be concatenated in the Function Test runtime (runtime 2);
- › - valid concatenation operands: specific value (i.e. literal value), any value ('?') with no length restriction or with a fixed length restriction, any value or none ('\*') with a fixed length restriction, or references to constants, templates, variables or template variables;
- › - operands of charstring and universal charstring template concatenation cannot contain matching mechanisms, only specific values and references (concatenation of charstring/universal charstring patterns has not been changed);
- › - reference operands of binary string template concatenation can also refer to binary string templates with wildcards in addition to the template types listed as valid operands (these cannot be used in template concatenations directly, because of parser limitations);
- › - similarly, reference operands of 'record of' or 'set of' template concatenation can also refer to template lists containing matching mechanisms (but these cannot appear in template concatenations directly due to parser limitations);
- › - when checking '?' and '\*' operands, a range length restriction, whose upper and lower bounds are equal, is also considered as a 'fixed' length restriction
- › (e.g. ? length(2..2) is a valid operand, but ? length(2..3) is not);
- › - the first operand of a 'record of' or 'set of' template concatenation can only be a reference (because of parser limitations);
- › - template module parameters cannot be concatenated in the configuration file;
- › - template concatenations of string types are evaluated at compile-time whenever possible, 'record of' and 'set of' template concatenations are not (same as it was in value concatenations).



# BUG 512878 - CONCATENATION FOR TEMPLATES(2)



- › This change has altered the order in which elements of the AST are checked in some cases, which sometimes causes different error messages to appear in RT2 than in RT1 for the same fault. Tests have been adjusted accordingly.
- › Template concatenation regression tests added under regression\_test/templateConcat.
- › Negative tests for the semantic analysis of template concatenations have been added under function\_test/Semantic\_Analyser/template\_concat.
  
- › Added an error message in case two length restrictions are used one after the other without separating brackets (i.e. when the 2nd concatenation operand and the concatenation result both have a length restriction).
- › For example:
  - › `template octetstring t := 'ABCD'O & ? length(2) length(4);`
  - › now causes a semantic error (for conformance to the TTCN-3 standard).
  - › Previously it was accepted by the compiler and it was equal to:
    - › `template octetstring t := ('ABCD'O & ? length(2)) length(4);`

# BUG 512359 - IMPLEMENT PORTS WITH TRANSLATION CAPABILITY



- › **The Testing and Test Control Notation version 3;**
- › **TTCN-3 Language Extensions:**
- › **Configuration and Deployment Support**
  
- › type port PortTypeIdmessage
- › [map to {OuterPortType[";"+ ]
- › [connect to {OuterPortType[";"+ ]} {"
- › {
- › (in {InnerInType [from {OuterInType with InFunction("[";"+ ][";"+ ]}+ |
- › out {InnerOutType [to {OuterOutType with OutFunction("[";"+ ][";"+ ]}+ |
- › inout {InOutType[";"+ ] |
- › address AddrType [to {OuterAddrType with AddrOutFunction("[";"+ ]}+ ]
- › [from { OuterAddrType with AddrInFunction("[";"+ ]}+ ] |
- › map param "{ { FormalValuePar [";"+ ]}+ " )"
- › unmap param "{ { FormalValuePar [";"+ ]}+ " )"
- › VarInstance) ";"
- › }+
- › "}"

# BUG 512359 - IMPLEMENT PORTS WITH TRANSLATION CAPABILITY(CONT.)



- › For limitations see ref.guide 4.35 **Ports with translation capability**
- › • A difference from the dual faced test ports is that the ports with translation capability can work in two modes, in normal mode and in translation mode. In normal mode, the port behaves as a standard messaging port, while in translation mode it works as it is defined in the ES 202 781 standard clause 5.2 ([21]). **A test port skeleton must be generated for the port type with the translation capability**, to ensure that the port can work in both modes.
- › **When used in standard mode, the skeleton has to be replaced by a real test port.**

# BUG 516296 - GENERATED JSON CODER FUNCTIONS CAUSE UNUSED PARAMETER WARNING



- › The type descriptor parameter 'p\_td' is always generated for JSON encoder and decoder functions of record types with 2 or more fields and is never used. This causes a warning when compiling the generated code.
- › Noticeable in the regression test folder regression\_test/XML/XER.
- › The warning only appears in Runtime 1.
- › Fixed.
- › The condition for generating the parameter name had not taken the number of record fields into account. Now it is only generated if the record only has 1 field.

# BUG 516237 - FAILING LICENSE CHECK IN ECLIPSE WITH NEW JAVA VERSIONS



- › It was brought to our attention that since the 1.8.121 version of java the verification of dsa signatures has been improved ... causing our license checking to fail.
- › The problem was caused by the trailing 0 -s of the dsa signature.
- › our license generator creates 48 bytes long dsa signatures ... or if it is smaller fills up the remaining space with 0 -s.
- › This was no problem so far .. but since this improved security check we need to cut down the trailing 0-s before passing the byte array to the JVM for verification.
- › This fix seems to solve the problem as far as I could test it.

# BUG 514753 - UPGRADE TO JUNG 2.1



- › Recently Jung has made a new release (2.1) that involves several new features (including efficiency improvements) that Titanium would benefit from.
- › This also means replacing apache commons with google guava 21.0 as Jung depends on this library (already available from eclipse orbit).
- › + the source code of Titanium needs some minor modifications in this upgrade.



# BUG 516871 - PROTOTYPE FUNCTION VALUE PARAM DOES NOT DEFAULTS TO 'IN' PARAM

- › Fixed.
- › For example:
- › `function bad_int_to_char2(octetstring a, out charstring b) {`
- ›  `} with {`
- ›  `extension "prototype(fast)"`
- ›  `}`
- › The a param should be defaulted to a in param implicitly.

# BUG 511966 - MAKEFILEGEN: POSSIBLY INCORRECT HEADER NAME GENERATION WITH CENTRALSTORAGE



- › Fixed.
- › When centralstorage is used then the `BASE_GENERATED_HEADERS` variable will contain wrong header files if the base project has TTCN or ASN1 module which has a different modulename than the filename.
- › This issue is also present in the eclipse makefilegen generator.



# BUG 512695 - MAKEFILEGEN: TYPE CODE SPLITTING INCORRECT MAKEFILE GENERATION



- › When the type codesplitting is used TITAN will generate only one header file.
- › But the makefilegen generates the makefile in a way to expect \*\_seq.hh etc. header files is.
- › This induces bad behaviour, like running the compiler 30 times etc.
- › The problem occurs when gnu make flag is used.

# E-FORGE NOVEMBER



Artifact ID	Title	Assigned To	Last Modified
artf506595	report some helpfull information when the VM runs out of memory	Nagy Lenard	11/30/2016 17:37
artf758526	"Make archive" button in Eclipse	Árpád Lovassy	11/22/2016 15:55
artf780940	enums in ranges	Bence Janos Szabo	11/21/2016 16:41
artf780941	using negations in template ranges	Bence Janos Szabo	11/23/2016 15:06
artf780933	enum in const and var definition	Bence Janos Szabo	11/22/2016 8:43
artf780937	enum in const and var definition 2	Bence Janos Szabo	11/22/2016 8:47
artf780945	Matching received enumeration variables according to a specific enumeration value range:	Bence Janos Szabo	11/21/2016 16:31
artf724215	After decoding failure all new bitstrings are initialized by '0'	Botond Baranyi	11/30/2016 13:08
artf503420	Obscure compiler error message	Botond Baranyi	11/30/2016 13:11
artf804312	xsd2ttcn issue	Bence Janos Szabo	11/30/2016 13:22

# BUGZILLA NOVEMBER



Bug ID	Product	Component	Status	Resolution	Summary	Changed	Reporter
495150	Titan	Plug-ins	CLOSED	FIXED	local variable smell ussue	11/7/2016 4:47	zsolt.szalai@ericsson.com
505866	Titan	Plug-ins	CLOSED	FIXED	Implement makefilegen port command in eclipse	11/7/2016 9:40	bence.janos.szabo@ericsson.com
506141	Titan	Plug-ins	CLOSED	FIXED	Ttcn3Parser: adding missing built-in functions: hostid, istemplatekind, any2unistr	11/7/2016 9:41	arpad.lovassy@ericsson.com
506448	Titan	Plug-ins	CLOSED	FIXED	Ttcn3Parser: support select union (syntactic and semantic check)	11/7/2016 9:42	arpad.lovassy@ericsson.com
506545	Titan	Plug-ins	CLOSED	FIXED	Ttcn3Parser: support checkstate port operation (syntactic and semantic check)	11/7/2016 9:42	arpad.lovassy@ericsson.com
501007	Titan	Core	CLOSED	FIXED	Support external functions for decmatch and @decoded	11/17/2016 3:04	gyorgy.rethy@ericsson.com
507957	Titan	Plug-ins	CLOSED	FIXED	"Make archive" button in Eclipse	11/22/2016 9:56	arpad.lovassy@ericsson.com
507958	Titan	Plug-ins	CLOSED	FIXED	NPE org.eclipse.titan.designer.parsers.ProjectConfigurationParser.fileBasedAnalysis(Pro	11/22/2016 10:12	arpad.lovassy@ericsson.com
507352	Titan	Core	CLOSED	FIXED	xsd2ttcn: translation error when minOccurs/maxOccurs and minLength/maxLength used t	11/23/2016 4:19	gyorgy.rethy@ericsson.com
506917	Titan	Core	CLOSED	FIXED	XER: union fields are encoded incorrectly	11/23/2016 4:20	bence.janos.szabo@ericsson.com
507930	Titan	Plug-ins	CLOSED	FIXED	Too much circular reference marker (duplicated)	11/30/2016 10:03	jeno.balasko@ericsson.com
507684	Titan	Plug-ins	CLOSED	FIXED	Infinite loop and stack overflow in semantic analyzer	11/30/2016 10:04	jeno.balasko@ericsson.com
495397	Titan	Plug-ins	CLOSED	FIXED	Documentation: correct the Titan Actions on the Toolbar	11/30/2016 10:07	jeno.balasko@ericsson.com
500559	Titan	Plug-ins	CLOSED	FIXED	Titan specific preference exporting	11/30/2016 10:09	kristof.szabados@ericsson.com
507836	Titan	Plug-ins	RESOLVED	FIXED	import via tpd fails if the tpd contains virtual folder	11/20/2016 13:51	jeno.balasko@ericsson.com

# E-FORGE DECEMBER



Artifact ID	Title	Assigned To	Last Modified
artf743325	when wrong compiler version detected, suggest make clean	Bence Janos Szabo	12/8/2016 10:13
artf635316	infinity	Bence Janos Szabo	12/13/2016 9:00
artf399982	JSON enc/dec: default values for empty fields- problems on Solaris only	Botond Baranyi	12/6/2016 12:03
artf677608	Xmlworkflow testcase unification	Bence Janos Szabo	12/15/2016 12:06
artf780943	Initialization of an enumeration variable with a specific enumeration value	Bence Janos Szabo	12/6/2016 9:56
artf721830	remove or move not necessary includes from TITAN headers	Bence Janos Szabo	12/13/2016 9:02
artf724072	implement @index	Botond Baranyi	12/6/2016 12:05
artf724182	implement "\N{typereference}", "\N{reference}", "\q{group,plane,row,cell}" for pattern	Bence Janos Szabo	12/8/2016 12:34
artf725754	"-infinity, infinity, not-a-number in arithmetic expressions"	Bence Janos Szabo	12/9/2016 13:22
artf729005	charstring in match	Bence Janos Szabo	12/15/2016 12:23
artf802981	Compilation error for with { extension "done" };	Botond Baranyi	12/6/2016 12:05
artf805060	Look through the TTCN3+ study	Bence Janos Szabo	12/1/2016 9:36
artf227790	Makefilegen support for ttcn-based XML workflow	Bence Janos Szabo	12/16/2016 9:19
artf190389	More info about XER enc/dec problem CR_TR00019566	Bence Janos Szabo	12/19/2016 9:14
artf455649	Compiler option to read input files list from a text file	Bence Janos Szabo	12/19/2016 9:13

# BUGZILLA DECEMBER



Bug ID	Product	Component	Status	Resolution	Summary	Changed	Reporter
507270	Titan	Core	CLOSED	FIXED	XER: encoding union with useType error	12/6/2016 2:19	bence.janos.szabo@ericsson.com
498591	Titan	Core	CLOSED	FIXED	Implement the 'any from' clause	12/6/2016 6:11	botond.baranyi@ericsson.com
509153	Titan	Plug-ins	CLOSED	FIXED	TTCN-3 code completion does not work	12/14/2016 10:23	arpad.lovassy@ericsson.com
508200	Titan	Core	CLOSED	FIXED	Call external functions implicitly for PER encoding/decoding	12/15/2016 4:13	gyorgy.rethy@ericsson.com
500221	Titan	Core	CLOSED	FIXED	Support encvalue()/decvalue() for ASN.1 types	12/15/2016 4:16	gyorgy.rethy@ericsson.com
509146	Titan	Core	CLOSED	FIXED	Printing type of JSON encoding through 'encvalue' is not set	12/15/2016 4:16	botond.baranyi@ericsson.com
509294	Titan	Plug-ins	CLOSED	FIXED	NPE if extension attribute string is empty	12/15/2016 10:59	arpad.lovassy@ericsson.com
508808	Titan	Plug-ins	CLOSED	FIXED	Incremental parser: commentLocation is not filled correctly	12/15/2016 11:01	arpad.lovassy@ericsson.com
500731	Titan	Core	CLOSED	FIXED	makefilegen segmentation fault error	12/16/2016 3:07	bence.janos.szabo@ericsson.com
509304	Titan	Core	CLOSED	FIXED	Missing namespace for generated custom coder functions	12/19/2016 4:27	botond.baranyi@ericsson.com
509434	Titan	Core	CLOSED	FIXED	Kill and stop operations sometimes cause segfault	12/19/2016 7:29	botond.baranyi@ericsson.com
500334	Titan	Core	CLOSED	FIXED	#include header file to the generated file containing encvalue/decvalue	12/20/2016 7:29	gyorgy.rethy@ericsson.com

# E-FORGE JANUARY



Artifact ID	Title	Assigned To	Last Modified
artf190392	Use of constants in variant definitions CR_TR00019588	Bence Janos Szabo	1/23/2017 14:57
artf695299	Designer false error: incomplete template and variable definitions	Balasko, Jeno	12/16/2016 9:09
artf786863	Debug mode and memory leaks	Botond Baranyi	1/26/2017 9:49
artf378938	Encoding in oct2unichar, unichar2oct, encvalue_unichar, decvalue_unichar	Bence Janos Szabo	1/3/2017 10:17
artf640920	xsd2ttcn: type aliases when restriction/extension has no real effect	Bence Janos Szabo	1/5/2017 13:34
artf817270	Already mapped/connected ports should not be allowed to be connected/mapped	Bence Janos Szabo	1/31/2017 12:16
artf817538	it 'is possible to send a message to SUT without the port to be mapped previously if the port is declared with address	Bence Janos Szabo	1/31/2017 15:56

# BUGZILLA JANUARY/1



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter
509855	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	junitlogger does not escapes the special characters in verdict reason	1/5/2017 9:53	bence.janos.szabo@ericsson.com
499534	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	TPD build config selection	1/6/2017 7:33	gabor.szalai@ericsson.com
509523	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Segfault caused by debugger function call storage	1/11/2017 11:41	botond.baranyi@ericsson.com
492432	Titan	Core	titan-inbox@eclipse.org	CLOSED	WONTFIX	Inconsistent assignment handling	1/12/2017 4:51	gabor.szalai@ericsson.com
509867	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Erroneous decoding of binary strings encoded in JSON	1/16/2017 11:25	gyorgy.rethy@ericsson.com
510562	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Incorrect code generation of select statements with goto	1/17/2017 9:51	bence.janos.szabo@ericsson.com
499151	Titan	Core	jeno.balasko@ericsson.com	RESOLVED	FIXED	Documentation:Titan installation prerequisites are unclear in installation guide	1/18/2017 9:05	gyorgy.rethy@ericsson.com
510734	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	xsd2ttn: special float values mishandled in enumerations	1/20/2017 7:55	bence.janos.szabo@ericsson.com
510589	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	xsd2ttn: special fixed value of float elements are not handled correctly	1/20/2017 7:57	bence.janos.szabo@ericsson.com
510610	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	xsd2ttn: The type of the attributes does not follow the changed type name when nam	1/20/2017 8:01	bence.janos.szabo@ericsson.com
510757	Titan	Core	titan-inbox@eclipse.org	CLOSED	DUPLICATE	xsd2ttn: Type name conversion does not follows to the type of the attributes	1/20/2017 8:01	bence.janos.szabo@ericsson.com
511078	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	'of' types or array can be indexed with a record of or array of integer	1/26/2017 8:08	bence.janos.szabo@ericsson.com
511180	Titan	TestPorts	lenard.nagy@ericsson.com	CLOSED	FIXED	Raspberry GPIO TestPort	1/27/2017 6:52	lenard.nagy@ericsson.com

# BUGZILLA JANUARY/2



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter
495127	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	headless mode reports generation	1/6/2017 7:35	leela.illa@wipro.com
509442	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	There is no error marker on not existing timer in altstep	1/6/2017 7:47	jeno.balasko@ericsson.com
509444	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing error marker in parameter assignment of getreply if the variable is not defined	1/6/2017 7:58	jeno.balasko@ericsson.com
509453	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing marker on argument of activate	1/6/2017 7:59	jeno.balasko@ericsson.com
509446	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing error markers on not existing port	1/6/2017 7:59	jeno.balasko@ericsson.com
509449	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing marker if function definition is missing in argument of refers	1/6/2017 8:00	jeno.balasko@ericsson.com
502170	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Parser doesn't recognize renaming a definition in a group	1/6/2017 8:04	jeno.balasko@ericsson.com
502537	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Find and replace in ttcn field name cause NPE	1/6/2017 8:04	jeno.balasko@ericsson.com
505795	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Makefile contains [UNRESOLVED_PATH_VAR] which cause make error	1/6/2017 8:05	jeno.balasko@ericsson.com
507836	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	import via tpd fails if the tpd contains virtual folder	1/6/2017 8:05	jeno.balasko@ericsson.com
495934	Titan	Plug-ins	kristof.szabados@ericsson.com	CLOSED	FIXED	Multiple reactions to single editing when the file is open in several editors	1/6/2017 8:06	kristof.szabados@ericsson.com
498195	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Check initial value of the variable definition for each possible type	1/6/2017 8:07	jeno.balasko@ericsson.com
509441	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing error markers on arguments of function decvalue	1/6/2017 8:21	jeno.balasko@ericsson.com
498267	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Check template ref type against expected type	1/6/2017 8:22	jeno.balasko@ericsson.com
498592	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	bin directory generated in wrong location (Artifact artf760105)	1/6/2017 8:22	jeno.balasko@ericsson.com
499599	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	NPE while processing linker option [OPENSSL_DIR]/lib	1/6/2017 8:23	jeno.balasko@ericsson.com
509269	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Tpd should accept relativeURI as virtual folder	1/6/2017 8:24	jeno.balasko@ericsson.com
510037	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	some code smells were measured twice	1/6/2017 8:34	kristof.szabados@ericsson.com
499888	Titan	Plug-ins	lenard.nagy@ericsson.com	CLOSED	FIXED	incorrect error reported for implicitly omitted field	1/9/2017 7:36	kristof.szabados@ericsson.com
495785	Titan	Plug-ins	lenard.nagy@ericsson.com	CLOSED	FIXED	Empty asn1 file is not handled	1/9/2017 8:12	jeno.balasko@ericsson.com
506684	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Improve incremental parsing using these files	1/11/2017 9:52	jeno.balasko@ericsson.com
508462	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False error marker in ASN1 file in case of named values	1/11/2017 10:00	jeno.balasko@ericsson.com
510369	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty error marker on template parameter of type TYPE_ASN1_ENUMERATED	1/12/2017 10:35	jeno.balasko@ericsson.com
498868	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	call setIsErroneous(true) after reportingError	1/17/2017 8:31	jeno.balasko@ericsson.com
510262	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty error marker on template parameter referenced from log statement	1/18/2017 7:32	jeno.balasko@ericsson.com
510035	Titan	Plug-ins	titan-inbox@eclipse.org	RESOLVED	FIXED	add extra checks for unused parameter detection in template definitions	1/18/2017 9:49	kristof.szabados@ericsson.com
510590	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Document make archive	1/19/2017 8:20	arpad.lovassy@ericsson.com
508115	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	DUPLICATE	Incremental parser: marker remains in commented code	1/19/2017 8:24	arpad.lovassy@ericsson.com
511184	Titan	Plug-ins	jeno.balasko@ericsson.com	RESOLVED	FIXED	NPE in headless mode because of trying to open dialogs	1/27/2017 8:13	jeno.balasko@ericsson.com
511318	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False error markers on logged timer value	1/31/2017 4:30	jeno.balasko@ericsson.com



# E-FORGE FEBRUARY



Artifact ID	Title	Assigned To	Last Modified
artf824590	No Solaris binary for Titan 6.2.0	Elemer Lelik	2/20/2017 9:25
artf657110	Code completion bug in case of functin return value	Árpád Lovassy	2/6/2017 10:29
artf724042	implement @fuzzy support	Botond Baranyi	3/1/2017 10:09
artf802978	RAW codec error	Elemer Lelik	2/20/2017 15:20
artf824206	Analyze Titan code with code checker	Bence Janos Szabo	2/17/2017 9:42

# BUGZILLA FEBRUARY



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter
510868	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen generation from TPD buildconfig issue	2/1/2017 8:18	bence.janos.szabo@ericsson.com
511756	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Modifying the template restrictions of modified templates should be allowed	2/7/2017 8:01	bence.janos.szabo@ericsson.com
511826	Titan	Core	lenard.nagy@ericsson.com	CLOSED	FIXED	Value template of empty record is unbound	2/7/2017 8:26	lenard.nagy@ericsson.com
511816	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	compiler -v should show openssl version regardless licensing	2/7/2017 9:52	bence.janos.szabo@ericsson.com
511764	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Using indexing short hand notation does not works with constants	2/7/2017 9:52	bence.janos.szabo@ericsson.com
511912	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	xsd2ttn: enumeration in unions are not converted correctly	2/9/2017 9:39	bence.janos.szabo@ericsson.com
511952	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen: should call TTCN3_DIR/Install/bin/xsd2ttn instead of xsd2ttn	2/9/2017 9:40	bence.janos.szabo@ericsson.com
507197	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen: Code splitting modes from TPD does not work	2/9/2017 9:43	bence.janos.szabo@ericsson.com
511976	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	modulepar with refers produces fatal error	2/9/2017 9:59	bence.janos.szabo@ericsson.com
511918	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen: codesplitting with -g switch unwanted behaviour	2/10/2017 3:42	bence.janos.szabo@ericsson.com
511838	Titan	Plug-ins	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Implement code splitting for internal makefile generator	2/13/2017 7:33	jeno.balasko@ericsson.com
511842	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Timeout field is too "narrow"	2/13/2017 7:34	jeno.balasko@ericsson.com
511521	Titan	Plug-ins	arpad.lovassy@ericsson.com	CLOSED	FIXED	Select union does not work for case(m1,m2)	2/13/2017 7:36	jeno.balasko@ericsson.com
512119	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	TITAN is not compatible with OpenSSL 1.1.0d 26 Jan 2017	2/14/2017 7:44	bence.janos.szabo@ericsson.com
511438	Titan	Plug-ins	kristof.szabados@ericsson.com	CLOSED	FIXED	False syntactic error marker on semicolon in function body in case of incremental parsing	2/14/2017 7:49	jeno.balasko@ericsson.com
511733	Titan	Plug-ins	arpad.lovassy@ericsson.com	CLOSED	FIXED	Select union does not work if expression is anytype	2/14/2017 7:50	arpad.lovassy@ericsson.com
512174	Titan	Other	bence.janos.szabo@ericsson.com	CLOSED	FIXED	hello example project incorrect logfile generation	2/14/2017 8:08	bence.janos.szabo@ericsson.com
508116	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Incremental parser: marker remains in commented code	2/16/2017 7:29	arpad.lovassy@ericsson.com
512346	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Increase year to 2017 in copyright message	2/17/2017 4:29	bence.janos.szabo@ericsson.com
500919	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Cannot access parameters after altstep deactivates itself	2/17/2017 6:16	botond.baranyi@ericsson.com
512030	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Code splitting error markers on projects are not removed	2/17/2017 10:40	bence.janos.szabo@ericsson.com
506451	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Test Bug 506448 - Tttn3Parser: support select union (syntactic and semantic check)	2/17/2017 10:41	arpad.lovassy@ericsson.com
512488	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	usage stats send sometimes crashes on ubuntu 16.04	2/23/2017 2:49	bence.janos.szabo@ericsson.com
512409	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty error marker on enum value	2/24/2017 7:56	jeno.balasko@ericsson.com
511184	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	NPE in headless mode because of trying to open dialogs	2/24/2017 8:10	jeno.balasko@ericsson.com
512688	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen: Incorrect .c and .cc compiling rule	2/27/2017 10:39	bence.janos.szabo@ericsson.com
512353	Titan	Plug-ins	lenard.nagy@ericsson.com	CLOSED	FIXED	Update copyright headers to 2017	2/28/2017 13:08	jeno.balasko@ericsson.com
511453	Titan	Core	bence.janos.szabo@ericsson.com	RESOLVED	FIXED	Incorrect code generation of parameterized templates	2/1/2017 10:01	bence.janos.szabo@ericsson.com
512698	Titan	Plug-ins	titan-inbox@eclipse.org	RESOLVED	FIXED	Faulty semantic error marker on enum value embedded in function return value	2/24/2017 10:19	jeno.balasko@ericsson.com
512757	Titan	Plug-ins	jeno.balasko@ericsson.com	RESOLVED	FIXED	False semantic error marker on component variable assignment with enum type	2/27/2017 11:10	jeno.balasko@ericsson.com
512793	Titan	Core	botond.baranyi@ericsson.com	RESOLVED	FIXED	JSON decoder can't decode zero length bit/hex/octetstring	2/28/2017 5:00	gabor.szalai@ericsson.com
511903	Titan	Core	botond.baranyi@ericsson.com	RESOLVED	FIXED	Dynamic erroneous attributes	2/28/2017 6:03	botond.baranyi@ericsson.com

# E-FORGE MARCH



Artifact ID	Title	Assigned To	Last Modified
artf768630	Test the new code splitting mode	Balasko, Jenő	2/27/2017 17:14
artf430279	Eclipse executor crashes on template module parameter	Balasko, Jenő	2/28/2017 11:38

# BUGZILLA MARCH



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter
512793	Titan	Core	botond.baranyi@ericsson.com	CLOSED	FIXED	JSON decoder can't decode zero length bit/hex/octetstring	3/1/2017 10:11	gabor.szalai@ericsson.com
512757	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False semantic error marker on component variable assignment with enum type	3/15/2017 18:30	jeno.balasko@ericsson.com
512698	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	Faulty semantic error marker on enum value embedded in function return value	3/15/2017 18:31	jeno.balasko@ericsson.com
511903	Titan	Core	botond.baranyi@ericsson.com	CLOSED	FIXED	Dynamic erroneous attributes	3/16/2017 7:36	botond.baranyi@ericsson.com
513762	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	XER: Incorrect decision implementation when to generate XER code	3/17/2017 5:07	bence.janos.szabo@ericsson.com
513749	Titan	Core	botond.baranyi@ericsson.com	CLOSED	FIXED	Change 'UNIVERSAL tagclass' warning to note	3/28/2017 9:43	botond.baranyi@ericsson.com
513765	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False positive error marker in ASN1 in the ROSE PM	3/31/2017 7:16	jeno.balasko@ericsson.com
514249	Titan	Core	bence.janos.szabo@ericsson.com	RESOLVED	FIXED	XER: allow anytype to be xer enc/decodable	3/27/2017 9:33	bence.janos.szabo@ericsson.com
513731	Titan	Plug-ins	titan-inbox@eclipse.org	RESOLVED	FIXED	False error markers in open type used in ttcn3 module	3/31/2017 7:47	jeno.balasko@ericsson.com

# BUGZILLA APRIL



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter
514253	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	XER: untagged top level union should be encoded without a tag	4/3/2017 6:20	bence.janos.szabo@ericsson.com
514180	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Select union code generation sometimes produces incorrect c++ code	4/3/2017 6:22	bence.janos.szabo@ericsson.com
514742	Titan	Core	jeno.balasko@ericsson.com	CLOSED	FIXED	Change the output of the 'make install', make it more user friendly	4/5/2017 2:46	jeno.balasko@ericsson.com
515193	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Write an example about code splitting in the reference guide	4/12/2017 9:46	bence.janos.szabo@ericsson.com
515229	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Update titanver to work with clang	4/13/2017 3:46	bence.janos.szabo@ericsson.com
514156	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen: More restriction on name attribute of the referenced project	4/19/2017 9:46	bence.janos.szabo@ericsson.com
515449	Titan	Core	botond.baranyi@ericsson.com	CLOSED	FIXED	Extend JSON 'as value' attribute for records/sets with one field and for the anytype	4/20/2017 8:48	botond.baranyi@ericsson.com
515436	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Create a legacy switch in TITAN for the XER encoding of untagged top level unions	4/25/2017 6:08	bence.janos.szabo@ericsson.com
515230	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Tests are failing with clang	4/27/2017 8:53	gergely.pilisi@ericsson.com
515569	Titan	Core	botond.baranyi@ericsson.com	CLOSED	FIXED	Value redirect in 'done' operation crashes on RHEL	4/27/2017 10:03	botond.baranyi@ericsson.com
515584	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	JSON encoding of field/element types	4/28/2017 4:58	botond.baranyi@ericsson.com
515592	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Any product using TSTLogger plugin?	4/28/2017 5:27	francisco.javier.jimenez@ericsson.com
515104	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	junitlogger creates an <error> tag outside the <testcase> tag	4/28/2017 7:46	botond.baranyi@ericsson.com
514249	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	XER: allow anytype to be xer enc/decodable NPE while parsing git\titan.EclipsePlug-ins\Semantic_Analyzer\Tests\src\Unstructured_tests\SemanticErrors3.asn	4/3/2017 6:19	bence.janos.szabo@ericsson.com
507368	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False error markers in open type used in ttcn3 module	4/3/2017 8:56	jeno.balasko@ericsson.com
513731	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	False error markers in open type used in ttcn3 module	4/3/2017 8:57	jeno.balasko@ericsson.com
514683	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False positive error marker on field if its type is ASN1 OpenType/enum and dot notation is applied	4/6/2017 6:47	jeno.balasko@ericsson.com
514750	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False positive error marker if the same Open_Type possibility is referred from more objects	4/6/2017 6:51	jeno.balasko@ericsson.com
510035	Titan	Plug-ins	kristof.szabados@ericsson.com	CLOSED	FIXED	add extra checks for unused parameter detection in template definitions	4/6/2017 8:33	kristof.szabados@ericsson.com
515168	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	ASN1 Occurrence marker freezes if click after "]"	4/20/2017 8:50	jeno.balasko@ericsson.com
515509	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	False positive error marker if the type of the Open_Type has inline def without name	4/26/2017 2:32	jeno.balasko@ericsson.com
515799	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	False positive error marker in ASN1 related with the MAP-ReturnError-v3	4/27/2017 5:20	jeno.balasko@ericsson.com
508528	Titan	Plug-ins	arpad.lovassy@ericsson.com	CLOSED	FIXED	Parse tree logging	4/27/2017 8:50	arpad.lovassy@ericsson.com
515740	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Add the new -N flag to the plugins	4/28/2017 4:35	bence.janos.szabo@ericsson.com
515942	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	Activate Debug makefilegen flag shall be implemented	4/28/2017 4:51	jeno.balasko@ericsson.com
515427	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	TITAN plugin wrongly emits error message with list templates	4/28/2017 9:22	bence.janos.szabo@ericsson.com

# BUGZILLA MAY



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter			
511453	Titan	Core	bence.jan	CLOSED	FIXED	Incorrect code generation of parameterized templates	5/2/2017 6:09	bence.janos.szabo@ericsson.com			
516022	Titan	Core	bence.jan	CLOSED	FIXED	Makefilegen: remove generated headers dependency from all .c .cc files	5/3/2017 10:32	bence.janos.szabo@ericsson.com			
512878	Titan	Core	botond.ba	CLOSED	FIXED	Concatenation for templates	5/5/2017 10:05	gyorgy.rethy@ericsson.com			
512359	Titan	Core	bence.jan	CLOSED	FIXED	Implement ports with translation capability	5/8/2017 3:53	bence.janos.szabo@ericsson.com			
499151	Titan	Core	jeno.bal	CLOSED	FIXED	Documentation: Titan installation prerequisites are unclear in installation guide	5/8/2017 4:43	gyorgy.rethy@ericsson.com			
516296	Titan	Core	botond.ba	CLOSED	FIXED	Generated JSON coder functions cause unused parameter warning	5/8/2017 4:50	botond.baranyi@ericsson.com			
516107	Titan	Plug-ins	jeno.bal	CLOSED	FIXED	Wrong circular import markers	5/5/2017 6:53	jeno.balasko@ericsson.com			
516193	Titan	Plug-ins	jeno.bal	CLOSED	FIXED	Error in exporting project: rawURI == null is not handled	5/5/2017 6:54	jeno.balasko@ericsson.com			
515799	Titan	Plug-ins	jeno.bal	CLOSED	FIXED	False positive error marker in ASN1 related with the MAP-ReturnError-v3	5/5/2017 6:55	jeno.balasko@ericsson.com			
492363	Titan	Plug-ins	jeno.bal	CLOSED	FIXED	Rename refactoring does not work properly for fields in records	5/5/2017 9:25	lenard.nagy@ericsson.com			
501459	Titan	Plug-ins	titan-inbo	CLOSED	FIXED	Titanium warning in code smell detector - improvement	5/8/2017 4:26	bengt.miksits@ericsson.com			
516237	Titan	Plug-ins	titan-inbo	CLOSED	FIXED	failing license check with new Java versions	5/8/2017 4:27	kristof.szabados@ericsson.com			
514753	Titan	Plug-ins	kristof.sza	CLOSED	FIXED	upgrade to Jung 2.1	5/8/2017 4:29	kristof.szabados@ericsson.com			
512816	Titan	Plug-ins	titan-inbo	CLOSED	INVALID	Java out of Memory error	5/8/2017 4:48	erik.a.sjoberg@ericsson.com			
495776	Titan	Plug-ins	titan-inbo	CLOSED	WORKSFORME	Outline window is not refreshing after rename refactoring	5/8/2017 4:51	jeno.balasko@ericsson.com			

# BUGZILLA MAY (CONT.)



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed	Reporter
514249	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	XER: allow anytype to be xer enc/decodable	5/2/2017 6:08	bence.janos.szabo@ericsson.com
516022	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Makefilegen: remove generated headers dependency from all .c.cc files	5/8/2017 6:29	bence.janos.szabo@ericsson.com
511193	Titan	Core	jeno.balasko@ericsson.com	CLOSED	FIXED	Improve Titan install guide	5/8/2017 8:29	jeno.balasko@ericsson.com
512351	Titan	Core	jeno.balasko@ericsson.com	CLOSED	FIXED	Update copyright headers to 2017	5/8/2017 10:04	jeno.balasko@ericsson.com
516871	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Prototype function value param does not defaults to 'in' param	5/18/2017 9:01	bence.janos.szabo@ericsson.com
511966	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	makefilegen: possibly incorrect header name generation with centralstorage	5/18/2017 9:02	bence.janos.szabo@ericsson.com
512695	Titan	Core	bence.janos.szabo@ericsson.com	RESOLVED	FIXED	makefilegen: type code splitting incorrect makefile generation	5/18/2017 9:02	bence.janos.szabo@ericsson.com
495573	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	Issue in fetching code smells into single .xls sheet in headless mode	5/10/2017 2:44	kamanuri.swathi@wipro.com
516463	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Charstring_Value cannot be cast to org.eclipse.titan.designer.AST.TTCN3.values.UniversalCharstring_Value	5/11/2017 10:06	oakimk@gmail.com

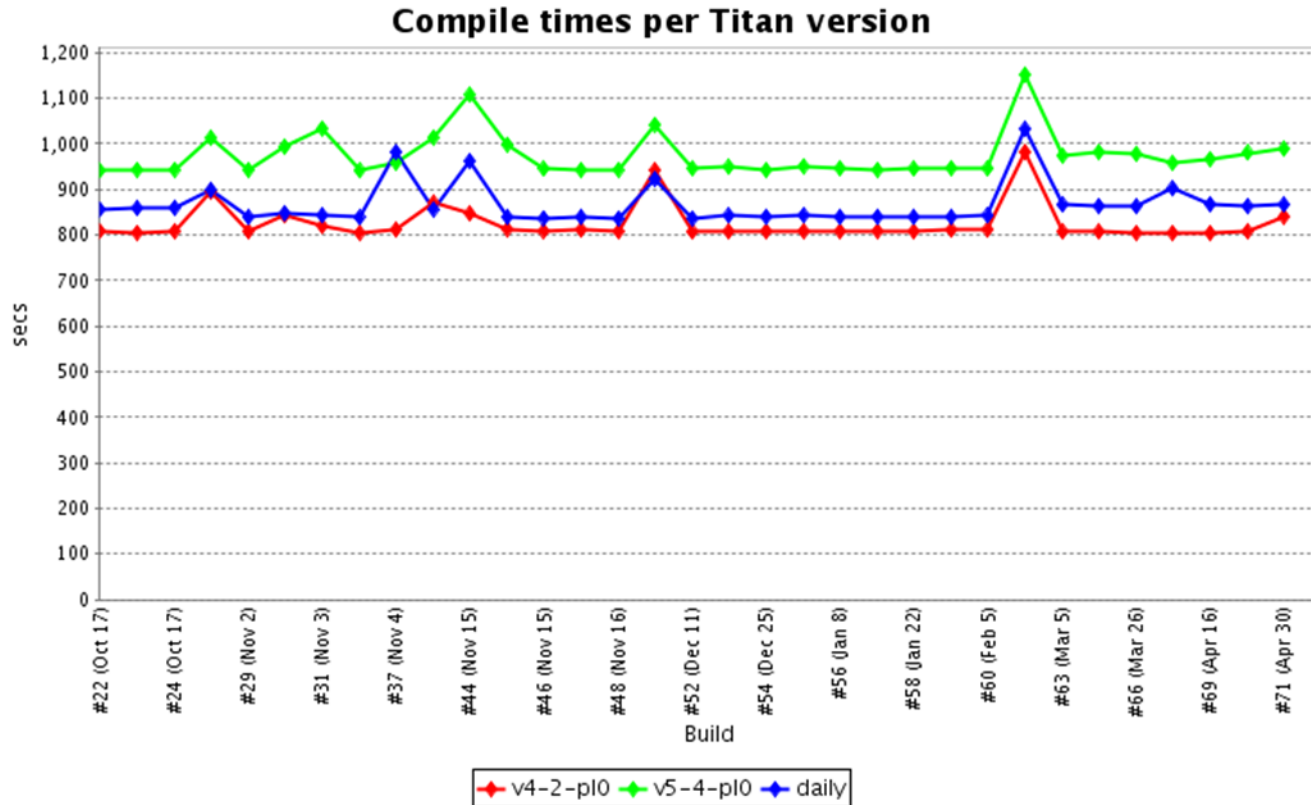
# DOCUMENTATION



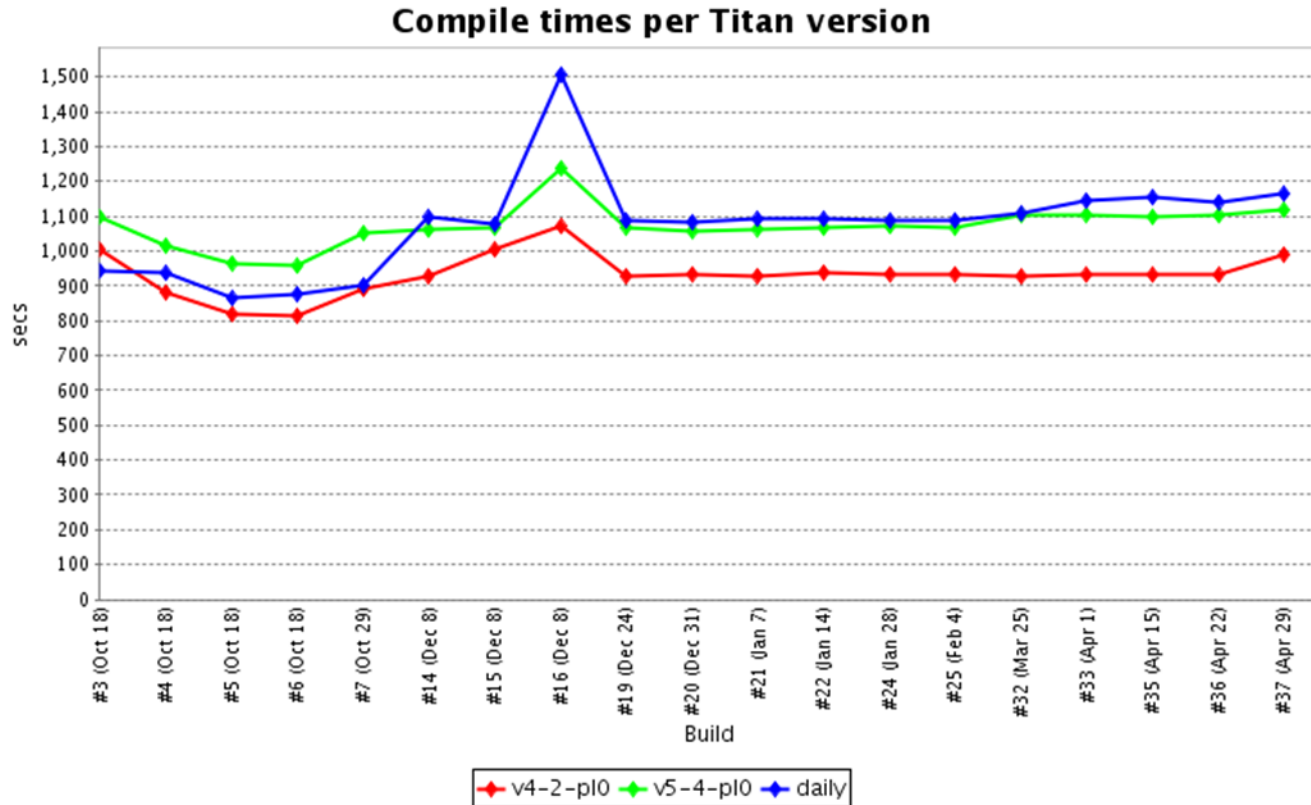
<b>CRL 113 200/6 R2A</b>		
109 21	CRL 113 200/6 2Uen Rev.A	PRI
109 47	CRL 113 200/6 Uen Rev. B	Release Notes
1/1531	CRL 113 200/6 Uen Rev.B	Install. Guide for....Test Executor
3/1531	CRL 113 200/6 Uen Rev.B	Install. Guide for....TITAN Designer and..... Eclipse IDE
1551	CRL 113 200/6 Uen Rev.B	Titanium Description
1/174 02	CRL 113 200/6 Uen Rev.B	Statement of Compliance for Eclipse Titan
2/174 02	CRL 113 200/6 Uen Rev.B	Statement of Compliance for use of XML schema in Eclipse Titan
1/198 17	CRL 113 200/6 Uen Rev.B	UG for Titan TTCN-3 Test Executor
2/198 17	CRL 113 200/6 Uen Rev.B	Programmers Tech. Reference Guide for the TITAN executor
4/198 17	CRL 113 200/6 Uen Rev.B	UG for the TITAN Designer for the Eclipse IDE
5/198 17	CRL 113 200/6 Uen Rev.B	UG of the TITAN Executor for the Eclipse IDE plug-in
6/198 17	CRL 113 200/6 Uen Rev.B	API Technical Reference for the TITAN executor
8/198 17	CRL 113 200/6 Uen Rev.B	Titan Executor API User Guide
9/198 17	CRL 113 200/6 Uen Rev.B	Programmers Tech. Reference Guide for Titanium



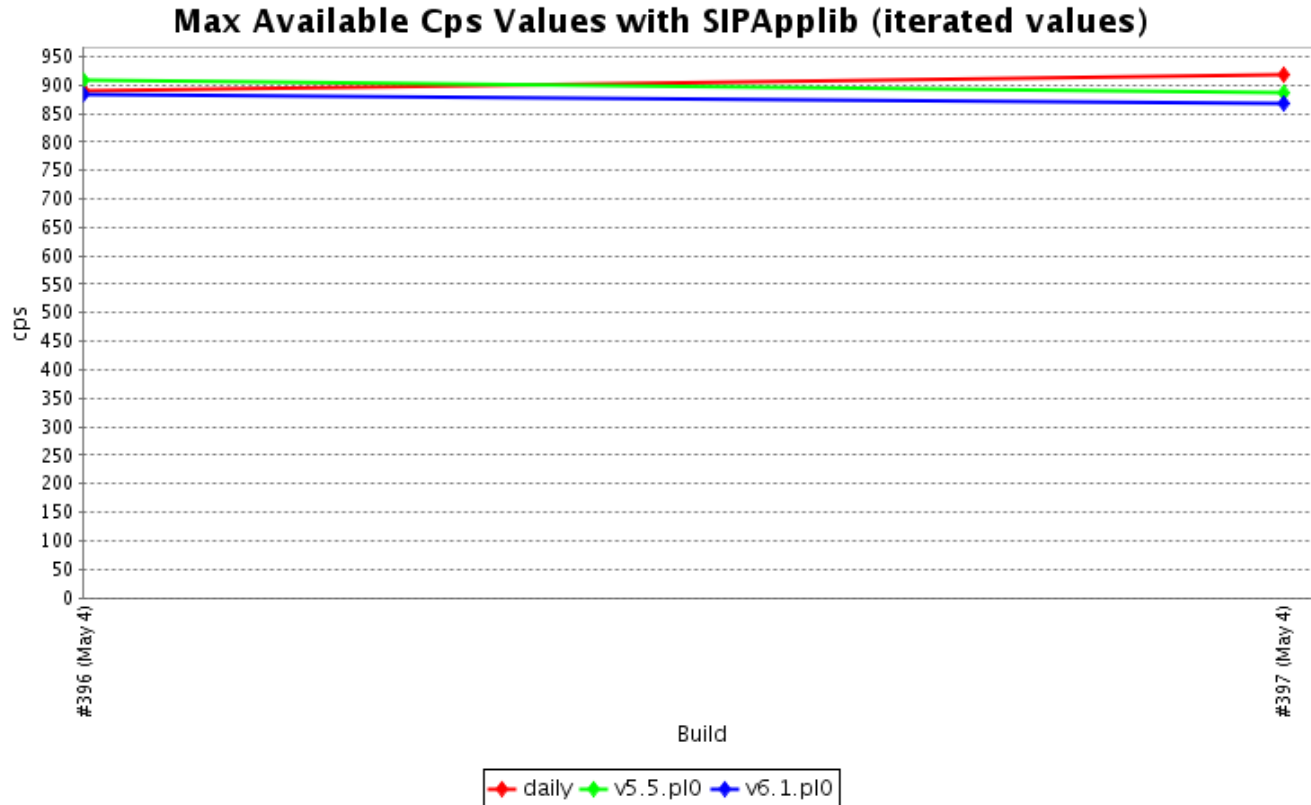
# COMPILE TIMES RT1



# COMPILE TIMES RT2



# RUNTIME CAPACITY VARIATION





**ERICSSON**