

TITAN 6.4.0 CHANGE LOG

2018.05.24

SUMMARY



- › This release is mainly a corrective release, it introduces no major new features. No backward incompatibilities are to be expected, with maybe one exception we see as a minor risk: the fix for

Bug 533767 - RAW encoder ALIGN(right) is working according of specification of ALIGN(left) (and vice versa) for octetstring

may induce under some unlikely circumstances an incompatible behaviour.

NEW FEATURES AND BUGFIXES



› Core:

- › Bug 527623 - Implement verdict redirect for 'done' statement
- › Bug 527847 - str2float should handle special float values
- › Bug 528082 - str2float("not_a_number") == not_a_number throws an exception in titan.core
- › Bug 527946 - RT2 record equality
- › Bug 529019 - Erroneous behaviour when default value of template parameter originates in an external function
- › Bug 529655 - string2ttcn to filter patterns of visible characters in octetstrings
- › Bug 529855 - coverage in 6.3.0
- › Bug 530502 - Code generation problems caused by the default template parameter modification
- › Bug 529892 - Syntax to bind a variant attribute to multiple encodings
- › Bug 531109 - TITAN build on Alpine Linux
- › Bug 532060 - 'done' operation on PTC causes internal error
- › Bug 532066 - Compiler performance issue (with OneM2M_Tester project)
- › Bug 532316 - Verdict redirects do not work on 'alive' components
- › Bug 531904 - new tpd tag disableUserInformation
- › Bug 534678 - Performance issues caused by template concatenation feature
- › Bug 535051 - runs on scope reduction (Titanium)

NEW FEATURES AND BUGFIXES



› **ASN.1**

- › Bug 521410 - ASN.1 named bit semantic check results false error
- › Bug 528313 - Type descriptor generation fault for ASN.1 open types
- › Bug 526554 - ASN.1 teletextstring value assignment error

› **Test ports, translation ports:**

- › Bug 531339 - Add discarding option to 'setstate' operation
- › Bug 532021 - Notify user if port is not mapped in translation mode
- › Bug 532451 - Port name clash error during unmapping
- › Bug 533006 - Implement reference to port in translation function
- › Bug 519945 - Determine the correct IPL4asp_User_CtrlFunctDef.cc template for translation ports
- › Bug 534266 - Cannot map translation ports on other components

NEW FEATURES AND BUGFIXES



› **xsd2ttcn:**

- › Bug 533543 - xsd2ttcn: incorrect code generated for user-defined type with built-in name
- › Bug 533242 - xsdtccn 6.3.0 error
- › Bug 534537 - xsd2ttcn: invalid subtype generated for restricted xs:long

› **Codecs:**

- › Bug 529017 - OER codec error
- › Bug 527653 - Implement extendable sequence coding in OER
- › Bug 533061 - OER dec: extension fields are not set to omit
- › Bug 534769 - OER dec: decoding default field causes DTE in RT2
- › **Bug 533767 - RAW encoder ALIGN(right) is working according of specification of ALIGN(left) (and vice versa) for octetstring**
- › Bug 533067 - RAW encoder froze in load test runtime
- › Bug 533487 - JSON attribute 'chosen' does not use default value
- › Bug 529888 - JSON variant attributes: use standard syntax
- › Bug 528465 - TAG and CROSSTAG for JSON encoder
- › Bug 527793 - TEXT codec bug

BUG 521410 - ASN.1 NAMED BIT SEMANTIC CHECK RESULTS FALSE ERROR



- › IEEE Security definitions ASN.1
- › ETSI is changing ITS security definitions to ASN.1, this problem come up during this work.
- › In ASN.1, named bits identify the position of the bit(s) within the instance, which shall be set to 1. All other bits in the instance are set to 0. But this doesn't influence the value space of the type.
- › For example in
- › `EndEntityType ::= BIT STRING {app (0), enrol (1) } (SIZE (8)) (ALL EXCEPT {})`
- › the actual value of app is: '10000000'B and the value of enroll is: '01000000'B
- › (and {app,enroll} would be '11000000'B)
- › Titan, when semantic checking the `_use_` (see attached file) of the above named bit 'app', gives the false error:
- › `../asn1/LibIts/Security/IEEE1609dot2.asn:293: error: '1'B is not a valid value for type `bitstring`
- › Named bit are specified in X.680(2008, 2015) clause 22.
- › Fixed.
- › Named bits are now extended (with zeros) to reach the minimum length restriction.

BUG 527623 - IMPLEMENT VERDICT REDIRECT FOR 'DONE' STATEMENT



- › The 'done' statement's value redirect is different in TITAN than in the TTCN-3 standard.
- › In TITAN it stores the behavior function's return value. It can only be used if the behavior function returns a value. The type of the variable in the value redirect must be the same as (or compatible with) the type returned by the behavior function. This type must also have the attribute 'extension "done"'. There must also be a matching template (of the same type as the return value) in the 'done' statement if the value redirect is used.
- › According to the TTCN-3 standard the 'done' statement's value redirect should store the local verdict on the component, after the behavior function has finished executing. Also, the 'done' statement cannot contain a matching template.
- › The standard compliant behavior should be added to TITAN in the following way:
 - › - if the 'done' statement contains a matching template, then the value redirect works as before (it stores the function's return value), and
 - › - if the 'done' statement doesn't contain a matching template, then the value redirect stores the component's local verdict (as per the standard).
- › Change implemented.
- › Regression tests added under regression_test/done.

BUG 527653 - IMPLEMENT EXTENDABLE SEQUENCE CODING IN OER



› Added in:

› <https://github.com/eclipse/titan.core/commit/6a82db01633db2428c8d29b7b87baf1976756ed7>

BUG 527793 - TEXT CODEC BUG



```
> I'm running a TEXT based MGCP decoder in a somewhat more complex parallel test configuration, and it fails with the following backtrace:
>
> Code: [Select all] [Show/ hide]
> ./IPA_Test: Abort was called
> /usr/lib/titan/libttn3-parallel-dynamic.so(_Z14signal_handleri+0xa3)[0x7f9a77b23813]
> /lib/x86_64-linux-gnu/libc.so.6(+0x33af0)[0x7f9a761e7af0]
> /lib/x86_64-linux-gnu/libc.so.6(gsignal+0x110)[0x7f9a761e7a70]
> /lib/x86_64-linux-gnu/libc.so.6(abort+0x17a)[0x7f9a761e919a]
> /lib/x86_64-linux-gnu/libc.so.6(regerror+0x75)[0x7f9a76286985]
> /usr/lib/titan/libttn3-parallel-dynamic.so(_ZNK11Token_Match11match_beginER11TTCN_Buffer+0x1b1)[0x7f9a77af1b11]
> ./MGCP_Types.so(_ZN11MGCP__Types11MgcpCommand11TEXT_decodeERK21TTCN_Typeddescriptor_iR11TTCN_BufferR16Limit_Token_Listbb+0x90)[0x7f9a7ea70904]
> ./MGCP_Types.so(_ZN11MGCP__Types11MgcpCommand6decodeERK21TTCN_Typeddescriptor_iR11TTCN_BufferN11TTCN_EncDec8coding_tEz+0x417)[0x7f9a7ea70229]
> ./MGCP_Types.so(_ZN11MGCP__Types16dec__MgcpCommandERK10CHARSTRING+0xbc)[0x7f9a7ea81323]
> ./IPA_Emulation.so(_ZN14IPA__Emulation17f__mgcp_to__userERK11OCTETSTRING+0x11b)[0x7f9a7fa0ad41]
> ./IPA_Emulation.so(_ZN14IPA__Emulation10ScanEventsEv+0x520)[0x7f9a7fa0b44d]
> ./IPA_Emulation.so(_ZN14IPA__Emulation12main__clientERK10CHARSTRINGRK7INTEGERS2_S5_+0xa1)[0x7f9a7fa0a8c4]
> ./IPA_Emulation.so(+0x21ba5)[0x7f9a7fa0cba5]
> /usr/lib/titan/libttn3-parallel-dynamic.so(_ZN11Module_List14start_functionEPKcS1_R8Text_Buf+0x2b)[0x7f9a77ad5d3b]
> /usr/lib/titan/libttn3-parallel-dynamic.so(_ZN12TTCN_Runtime14start_functionEPKcS1_R8Text_Buf+0x25)[0x7f9a77ae69e5]
> /usr/lib/titan/libttn3-parallel-dynamic.so(_ZN18TTCN_Communication13process_startEv+0x42)[0x7f9a77aad802]
> /usr/lib/titan/libttn3-parallel-dynamic.so(_ZN18TTCN_Communication23process_all_messages_tcEv+0x2f5)[0x7f9a77aae055]
> /usr/lib/titan/libttn3-parallel-dynamic.so(_ZN12TTCN_Runtime8ptc_mainEv+0xdff)[0x7f9a77aea50f]
> /usr/lib/titan/libttn3-parallel-dynamic.so(main+0x330)[0x7f9a77824da0]
> /lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf1)[0x7f9a761d4561]
> ./IPA_Test(+0xa1a)[0x564e293a0a1a]
> MC@nataraja: Unexpected end of PTC connection (6) from 127.0.0.1 [127.0.0.1].
```

> So the generated code wants to do something with regular expressions and then that fails, hence it calls regerror() but this in turn crashes with an abort.

```
> In TEXT.cc of titan.core I can find
> Code: [Select all] [Show/ hide]
> int Token_Match::match_begin(TTCN_Buffer& buff) const
> {
>     int ret_val=-1;
>     int ret_val=-1;
>     if(!null_match){
>         if (TTCN_EncDec::get_error_behavior(TTCN_EncDec::ET_LOG_MATCHING)!=
>             TTCN_EncDec::EB_IGNORE){
>             char msg[ERRMSG_BUFSIZE2];
>             regerror(ret_val, &posix_regexp_begin, msg, ERRMSG_BUFSIZE2);
```

BUG 527793 - TEXT CODEC BUG



› Comment from Gabor Szalai:

› "

› The

› char msg[ERRMSG_BUFSIZE2];

› regerror(ret_val, &posix_regexp_begin, msg, ERRMSG_BUFSIZE2);

› shouldn't be there.

› “

› Faulty regerror() call removed.

BUG 527847 - STR2FLOAT SHOULD HANDLE SPECIAL FLOAT VALUES



- › Currently it does not handle the special float values like `-infinity` and `not_a_number`
- › Changed `str2float()` and `float2str()` to handle the special float values. The string equivalents for these values in both functions are: `"infinity"`, `"-infinity"` and `"not_a_number"`.
- › Regression tests added under `regression_test/predefFunction`.

BUG 528082 - STR2FLOAT("NOT_A_NUMBER") == NOT_A_NUMBER THROWS AN EXCEPTION IN TITAN.CORE



- › This source code finished by "pass" in ttcn2java but returns false in case of tita.core (ttcn2cpp):
- › testcase tc_predef_str2float_good4() runs on CT {
 - › var charstring vl_cs := "not_a_number";
 - › if(str2float(vl_cs) == not_a_number){
 - › setverdict(pass)
 - › } else {
 - › setverdict(fail,"expected not_a_number got:", str2float(vl_cs));
 - › }
 - › }
- › It is suggested to return true.
- › Fixed

BUG 528313 - TYPE DESCRIPTOR GENERATION FAULT FOR ASN.1 OPEN TYPES



› The XER descriptor (and the descriptors of other codecs) is not generated for open type fields, if the type is an enumerated declared in-line.

› Example:

› ERROR-CLASS ::= CLASS

```
› {  
›   &category      PrintableString(SIZE(1)),  
›   &code           INTEGER,  
›   &Type           DEFAULT INTEGER  
› }  
› WITH SYNTAX {&category &code [&Type]}
```

› ErrorSet ERROR-CLASS ::=

```
› {  
›   {"A" 1 [1] INTEGER} |  
›   {"A" 2 [1] VisibleString} |  
›   {"B" 1 [1] PrintableString} |  
›   {"B" 2 [1] BOOLEAN} |  
›   oneErrorObject  
› }
```

› oneErrorObject ERROR-CLASS ::= { "C" 3 ENUMERATED{foobar} } -- no XER descriptor

› ErrorReturn ::= SEQUENCE

```
› {  
›   errorCategory  ERROR-CLASS.&category ({ErrorSet}) OPTIONAL,  
›   errors          SEQUENCE OF SEQUENCE  
›   {  
›     errorCode     ERROR-CLASS.&code({ErrorSet}{@errorCategory}),  
›     errorInfo      [99] ERROR-CLASS.&Type({ErrorSet}{@errorCategory,@.errorCode})  
›   } OPTIONAL  
› }
```

› This only occurs with the new codec handling. If legacy codec handling is used, then this causes a fatal error in the compiler.

› These issues only occur if XER is enabled for ASN.1 types (compiler option -a).

› Fixed both issues. Added regression tests under regression_test/compileonly/openTypeXER.

BUG 527946 - RT2 RECORD EQUALITY



- › In RT2 the == operator (is_equal()) allows unbound fields of the record. But the standard states that the record must be completely initialized.
- › See Standard ETSI ES 201 873-1 V4.9.1 (2017-05):
- › 7.1.3 Relational operators
- › "Operands of equality (==) and non-equality (!=) shall be completely initialized values or ..."
- › Fixed.
- › Tests added under regression_test/recordOper.

BUG 529019 - ERRONEOUS BEHAVIOUR WHEN DEFAULT VALUE OF TEMPLATE PARAMETER ORIGINATES IN AN EXTERNAL FUNCTION



- › If for a template e.g. `m_denmMgmtCon` one relies on the default value of the parameter say `p_referenceTime` (which is the return value of `f_getCurrentTime()`) and invokes this parameter with "-",

- › like here :

```
› m_denmMgmtConTermination(  
›     m_tsActionId,  
›     -,  
›     c_transmissionInterval * 4 / 1000,  
›     c_transmissionInterval,  
›     -, //default:= f_getCurrentTime()  
›  
›                                     -, //default:=f_getCurrentTime()  
›                                     isCancellation  
›  
› )
```

- › then the value of the parameter stays the same after each invocation.

- › If instead of the "-" we use the function as parameter, as in:

```
› m_denmMgmtConTermination(  
›     m_tsActionId,  
›     -,  
›     c_transmissionInterval * 4 / 1000,  
›     c_transmissionInterval,  
›     f_getCurrentTime(),  
›  
›                                     f_getCurrentTime(),  
›                                     isCancellation  
›  
› )
```

- › then the parameter value changes accordingly at every new invocation.

- › This is a side effect of a speed optimization:

- › Titan calls the function referred as default parameter at the beginning of execution and then reuses the return value to save time;

- › This works well for deterministic functions , but causes problems with non-deterministic ones , as those that depend on time.

- › What we can offer as a solution is to implement the standard behaviour in runtime 2 (function test runtime) ,

- › but keeping the current behaviour in the default runtime (runtime 1 , load test runtime) .

- › Practically this means that users of runtime 1 will have to continue to use direct function references as parameters as you did.

- › Users of runtime 2 will have the choice of using the standard notation for default "-".

- Public | © Ericsson AB 2016 | 2016-10-14 | Page 16

BUG 529655 - STRING2TTCN TO FILTER PATTERNS OF VISIBLE CHARACTERS IN OCTETSTRINGS



- › Would it be possible to get either `log2str()` or `ttn2string()` to NOT output the user readable string when the octetstring happens to contain printable character values?
- › Ex:

```
› {  
›   data := {  
›     OctetString := '80'O  
›   }  
›   data := {  
›     OctetString := '60'O ("")  
›   }  
› }
```
- › I would like to get rid of the extra output (including the parentheses) to be able to feed the output of either `log2str` or `ttn3string` directly back into `string2ttn`.
- › (Or let `string2ttn` to accept (ignore) this extra information?)
- › Change implemented.
- › The operation `'ttn2string'` no longer displays the charstring equivalents of octetstrings. The operation `'log2str'` still displays the charstring equivalents of octetstrings as before.
- › Regression tests added under `regression_test/octetstrOper`.

BUG 529855 - COVERAGE IN 6.3.0



- › Hi all,
- › We have a problem with code coverage tool. When we collect and merge all .tcd files by using tcov2lcov, we get an error:
- › I/O warning : failed to load external entity ""
- › Schemas parser error : Failed to locate the main schema resource at '!'.
Error while validating XML/TCD files
- › Currently, we use lcov 1.12 on our machines.
- › The information of the tcov2lcov:
 - › TCD to LCOV Converter for the TTCN-3 Test Executor, version CRL 113 200/6 R3A
 - › Product number: CRL 113 200/6 R3A
 - › Build date: Nov 16 2017 19:46:23
 - › Compiled with: GCC 4.8.3
- › Copyright (c) 2000-2017 Ericsson Telecom AB
- › In version 6 R1A, the tcov2lcov run properly.
- › Best Regards
- › Huynh Thanh Tung
- › Fixed. (The XSD file name character pointer was set to a temporary string object's character pointer, which was already deleted when the program tried to open the XSD file).

BUG 530502 - CODE GENERATION PROBLEMS CAUSED BY THE DEFAULT TEMPLATE PARAMETER MODIFICATION



- › The change in the way default template parameters are handled caused C++ compilation errors in user projects (reported by Yann Garcia).
- › 1.) If a default template parameter is used in an 'encvalue' operation, where the type has a custom encoding set, then the default parameter's initialization code is not generated. Requires Runtime2 and legacy codec handling.
- › 2.) If a default value or template parameter is used in the default value of another template parameter, and it is used in a different module, then the first parameter's scope is not set correctly during code generation.
- › For examples see tests under regression_test/nondeterministicDefaultParam.

BUG 529888 - JSON VARIANT ATTRIBUTES: USE STANDARD SYNTAX



- › Currently Titan supports multiple encodings for a type, i.e. multiple encode variants can be attached to a type and the encode context can be (shall be) identified for the variant attributes.
- › But the syntax of the JSON variant attributes is still using the legacy syntax, e.g.
- › "JSON"."JSON:name as aE"
- › as opposed to the standard syntax
- › "JSON"."name as 'aE'".
- › This makes it impossible to apply the multiple encoding concept in standard test suites, like ETSI and oneM2M, as other TTCN-3 tools do not recognize this syntax, and also makes it difficult for the user to switch to the new syntax, as it probably will change again in the future.
- › Therefore it is anticipated to parse the standard syntax in Titan as soon as possible.
- › Ref.: ETSI ES 201 873-9 V4.8.3
- › Change implemented.
- › The existing JSON attributes can now be used with both the standard and the legacy syntax.
- › The functionality of JSON attributes has not been changed (any JSON attributes from the standard, that were previously not supported, are still not supported).

- › Related regression & function tests changed to test both the standard-compliant and the legacy syntax.

BUG 529892 - SYNTAX TO BIND A VARIANT ATTRIBUTE TO MULTIPLE ENCODINGS



- › The standard defines two syntax to bind a variant attribute to encoding(s):
 - › a) `"<encode>". "variant" //binding the variant attribute to one encoding`
 - › b) `{"<encode1>","<encode2>". "variant" //binding the variant attribute to multiple encodings`
- › Titan compiler currently supports syntax a) only.
- › Syntax b) would be very useful for some variants, especially for "name as ..." that is intensively used e.g. in the oneM2M test suite (which requires both XML and JSON encoding).
- › Beyond being a convenience feature to decrease code size, supporting syntax b) could also prevent typo errors, when e.g. different names are typed in "name as ..." for the different encodings unintentionally.
- › Change implemented.
- › Variant attributes can now have multiple encoding strings. Currently the "name as '..." variant attribute is the only one that can be applied to multiple encodings (XML and JSON). If the variant attribute cannot be applied to all of the specified (built-in) encodings, then the variant is considered erroneous. Variants attributes of custom encodings produce only warnings (as before).
- › Regression test case added under `regression_test/multipleEncodings`.



BUG 526554 - ASN.1 TELETExSTRING VALUE ASSIGNMENT ERROR

- › TeletexStringRestricted ::= TeletexString (SIZE(4))
- › var TeletexStringRestricted ts;
- › ts := "ttxx";
- › The assignment gives error:
- › In variable assignment:
 - › OER.ttcn:1595:10: error: Cannot use ISO-2022 string value in string context
 - › OER.ttcn:1595:10: error: "ttxx" is not a valid value for type `TeletexString' which has subtype length(4)
- › This error also applies to other ASN.1 string types like:
T61String,VideotexString,GraphicString,GeneralString
- › Fixed. The subtype check is now correctly performed on ISO-2022 string types, too.
- › Regression tests added under regression_test/compileonly/asnStringSubtypes.

BUG 531109 - TITAN BUILD ON ALPINE LINUX



- › TITAN should be buildable on Alpine Linux.
- › Added the makefile switch `ALPINE_LINUX`, that disables certain Linux-only features inside TITAN. (i.e. stack tracing for dynamic test case errors, segmentation faults and aborts).
- › Usage1: assign a value to `ALPINE_LINUX` in the file `Makefile.personal`.
- › E.g.: `ALPINE_LINUX := yes`
- › Usage2: assign a value to the `ALPINE_LINUX` environment variable in the command line.
- › E.g.: `export ALPINE_LINUX=yes`

BUG 531339 - ADD DISCARDING OPTION TO 'SETSTATE' OPERATION



- › Dual faced ports can discard messages of a certain type with a setting in the port's extension attribute.
- › This can only be done with the new translation ports by setting the port state to 2 (FRAGMENTED). A new option is needed for port.setstate, that discards the message, so it doesn't clash with actual fragmented messages.
- › Implemented state 4 (DISCARDED) for the 'setstate' operation.
- › Regression tests added under regression_test/portTranslation.

BUG 528465 - TAG AND CROSSTAG FOR JSON ENCODER



- › Hey,
- › I am implementing an OPC UA Protocol Module, and I am having issues with the JSON encoder. The OPC UA standard for JSON contains some structures, where the type of a field is determined by the value of another field. For example, the Node ID structure looks like the following:
 - › NodeId:
 - › integer IdType,
 - › - id,
 - › - integer Namespace
- › The IdType can contain values between 0 and 4, and the value determines the type of the ID. The id can be an integer, and 3 differently formatted string.
- › The current JSON encoder is not capable of dealing with this setup, so I would like to request a TAG, and CROSSTAG like addition for the JSON encoder.
- › Thanks,
- › Akos
- › Implemented the JSON attribute 'chosen', which behaves similarly to the RAW attribute 'CROSSTAG'. For details see the attribute's description in chapter 4.26.2 of the reference guide.
- › Regression tests added under regression_test/json and function_test/Semantic_Analyzer/json.

BUG 532021 - NOTIFY USER IF PORT IS NOT MAPPED IN TRANSLATION MODE



- › A note should be displayed by the compiler if a port with translation capabilities is mapped in direct mode (not in translation mode).

BUG 532060 - 'DONE' OPERATION ON PTC CAUSES INTERNAL ERROR



- › An internal error (DTE) occurs, when a 'done' operation on a PTC receives the status of a different PTC from the MC. This is because it attempts to extract the local verdict from the status message, which does not contain a verdict (the first segment of the return value is interpreted as the verdict, and the error occurs when the return value itself is being extracted).
- › Fixed.
- › Regression test case added to `regression_test/commMessage/TcommMessage.ttcn`.

BUG 532066 - COMPILER PERFORMANCE ISSUE (WITH ONEM2M_TESTER PROJECT)



- › The difference between the compiler's performance with two different versions of OneM2M_Types.ttcn is enormous.
- › The difference is in the time it takes to perform the semantic check: 9 seconds compared to 671 seconds.
- › The amount of memory allocations has also been multiplied by 100 (from 13 million to 1.2 billion).
- › Added a guard for `Type::chk_coding_attr`, which makes sure it is only executed once per type (except for recursive types).

BUG 532316 - VERDICT REDIRECTS DO NOT WORK ON 'ALIVE' COMPONENTS



- › If a component variable is created with the modifier 'alive', then redirecting its local verdict with the statement 'done' does not work (it always stores 'none' instead of the actual verdict).
- › Example:
 - › type component CT {}
 - › function f1() runs on CT {
 - › setverdict(pass);
 - › }
 - › function f2() runs on CT {
 - › log(getverdict); // pass
 - › }
 - › testcase tc() runs on CT {
 - › var CT ct := CT.create alive;
 - › ct.start(f1());
 - › var verdicttype vt := none;
 - › ct.done -> value vt;
 - › log(vt); // none
 - › ct.start(f2());
 - › ct.done -> value vt;
 - › log(vt); // none
 - › }

BUG 531904 - NEW TPD TAG DISABLEUSERINFORMATION



- › disableUserInformation is implemented by Jacek Klimkowicz in <https://review.gerrithub.io/#/c/401992/>
- › Documentation added to refguide

BUG 532451 - PORT NAME CLASH ERROR DURING UNMAPPING



- › The following line from the project 'oneM2MTester' causes an internal error (line 263 in OneM2M_Functions.ttcn):
 - › `unmap(vc_cse1:mccPort, system:mccPort);`
- › Output:
 - › Dynamic test case error: Internal error: There are more than one ports with name mccPort.
- › This is a code generation fault (the parameters of both calls to 'PORT::activate_port' are TRUE, which means both parameters of the 'unmap' statement are treated as ports in system components).
- › Fixed.

BUG 533006 - IMPLEMENT REFERENCE TO PORT IN TRANSLATION FUNCTION



- › A new (non-standard) language element should be added to TITAN, which references the translation port object in a translation function. This reference should be usable as the actual parameter of a function (where the corresponding formal parameter is of the translation port's type).
- › Syntax:
 - › `port.getref()`
- › Usage:
 - › `f_hasPortParameter(port.getref());`
 - ›
Language element implemented.
- › Tests added under `regression_test/portTranslation`.

BUG 533061 - OER DEC: EXTENSION FIELDS ARE NOT SET TO OMIT



› The extension fields in a record are not set to omit during decoding if the extension bit is unset. They remain unbound.

› Example:

› Stream:

› '038100400380442050800000200A0000004C5E0C14D2EA66B402E419F5EA0606244B6C0000000000000007D1000001020000279F3D4C40F9F35A60CE2DC3AD800200200030D41E8B00004002008D0000000098303D5F810101800300801A0CDDC8628A1410810C434552545F54535F415F4154000000000005A497A008410E001038001248104038300018001258104038300018002008D8104038300018080848E30E0535B8EC6BF19A68FEE1B97EC8C2AB56994177B3AC9C47BE89B20FA4599044E6227213C08CB3038ABB33357D4796BDF7939A3E36EE7F112001413257B238080B313E9614D3A9E8B62FBA832336068EF6820EC954655470BB40460EA2F1BFA9ADDC97632E53B2D1FDA4080D377ADF0D461AD51A28CB01C634B30B93AEA0564238080724AD2E23DF8D022333B664C9E4F4BB63DFB53EF7D1AE6AB068624C5AB9BB26690E1A2BC70F55374B8D149EEE33FFDDF3203670FA751631C07E9779503E712F3'O

› When decoded into an object of type Ieee1609Dot2Data (from IEEE1609dot2.asn) results in:

```
{
  protocolVersion := 3,
  content := {
    signedData := {
      hashId := sha256 (0),
      tbsData := {
        payload := {
          data := {
            protocolVersion := 3,
            content := {
              unsecuredData := '2050800000200A0000004C5E0C14D2EA66B402E419F5EA0606244B6C0000000000000007D1000001020000279F3D4C40F9F35A60CE2DC3AD800200200030D41E8B0000'O
            }
          },
          extDataHash := omit
        },
        headerInfo := {
          psid := 141,
          generationTime := 2553298271,
          expiryTime := omit,
          generationLocation := omit,
          p2pcdLearningRequest := omit,
          missingCrIIdentifier := omit,
          encryptionKey := omit,
          inlineP2pcdRequest := <unbound>,
          requestedCertificate := <unbound>
        }
      },
      signer := {
        ...
      }
    }
  },
  ...
}
```

› Change implemented.

› Record fields after the extension (...) are now set to omit during OER decoding, if the extension flag is not present in the stream.

BUG 533067 - RAW ENCODER FROZE IN LOAD TEST RUNTIME



```
> The following simplified code freezes in load test runtime but runs successfully in function test runtime (rt2). OS: win7/cygwin. (See also in attachment)
>
> module RAW_runtime_bug {
>
>   type component TempComp {}
>
>   type integer myint3
>   with {
>     variant "FIELDLENGTH(96)"
>     variant "COMP(signbit)"
>     variant "BITORDERINFIELD(lsb)"
>     variant "PADDING(dword32)"
>     encode "RAW" }
>
>   external function enc_raw3(in myint3 r) return octetstring
>   with { extension "encode(RAW) prototype(convert) errorbehavior(ALL:WARNING)" }
>   //external function dec_raw3(in octetstring r) return myint3
>   //with { extension "decode(RAW) prototype(convert) errorbehavior(ALL:WARNING)" }
>
>   testcase TC_encoding_decoding_of_big_integers() runs on TempComp{
>
>     var myint3 v12 := -2147483648 //->  '8000 0000 0000 0000 8000 0000'O //frozen
>     var octetstring o;
>     var octetstring expected12 := '0000 0080 0000 0000 0000 0080'O;
>
>     o := enc_raw3(v12); //freezing point
>     log("v12 test started");
>     // if ( o == expected12) { setverdict(pass) }
>     // else { setverdict(fail, "Encoding error 12. Expected: ", expected12, "; got: ", o)}
>     // if (v12 == dec_raw3(expected12)) {setverdict(pass)}
>     // else {setverdict(fail, "12. Decoding error of ", expected12, "; Expected: ", v12, " got ", dec_raw3(expected12) )}
>     // log("v12 test finished");
>   }
>
>   control {
>     execute( TC_encoding_decoding_of_big_integers());
>   }
>
> }
>
> Fixed.
> Changed the condition that identifies the lowest integer value, while RAW encoding, to use the global constant INT_MIN instead of checking for x == -x (whose behavior is undefined according to the C standard).
```

BUG 533487 - JSON ATTRIBUTE 'CHOSEN' DOES NOT USE DEFAULT VALUE



- › If one or more conditions in the JSON attribute 'chosen' refer to a field with a default value (set through the JSON attribute 'default'), then the condition is

evaluated before the default value is set for the field.

- › For example:
- › type record NodeId {
- › Byte IdType optional,
- › NodeIdType Id,
- › integer Namespace optional
- › } with {
- › variant (Id) "chosen (StringNodeId, IdType = 1;
- › GuidNodeId, IdType = 2;
- › ByteStringNodeId, IdType = 3;
- › UInt32NodeId, otherwise;)"
- ›
- › variant(IdType) "JSON : default (0)"
- › }

- › In this case the otherwise condition should be evaluated as 'true', if there is no data for field IdType in the JSON object.
- › Fixed.
- › Record/set fields with the attribute 'default' are now initialized with their default values at the beginning of decoding, instead of setting the default value at the end of decoding if there was no data for these fields in the JSON document.
- › This way the conditions in the attribute 'chosen' use the default value instead of the field's value before decoding (probably <unbound>).
- › This change is only made in the generated JSON decoder (used in RT1, and in RT2 if one of the fields has the attribute 'chosen'). The decoder in the class Record_Type (used in RT2 in all other cases) remains the same.
- › Regression tests added under regression_test/json.

BUG 533242 - XSDTCCN 6.3.0 ERROR



- › Created attachment 273440 [details]
- › xsd files
- › xsd2ttn 6.3.0 throws the following error with the attached files:
 - › xsd2ttn CDT-enumerationTypes-v2_14_0.xsd CDT-commonTypes-v2_14_0.xsd
 - › Notify: Checking documents...
 - › Notify: Parsing XML schema document `CDT-enumerationTypes-v2_14_0.xsd'...
 - › Notify: Parsing XML schema document `CDT-commonTypes-v2_14_0.xsd'...
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type from): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type from): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type holder): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type issuer): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type originator): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type dataContainerID): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type resourceID): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type resourceID): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type creator): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › ERROR:
 - › CDT-commonTypes-v2_14_0.xsd (in type resourceID): Reference for a non-defined element or simpleType or complexType type: http://www.onem2m.org/xml/protocols|ID
 - › Notify: 10 errors and no warnings were detected.
 - › This error does not appear with xsd2ttn earlier than 6.3.0 (6.2.0, 6.1.0, 5.5.0 tested)
- › The errors are no longer displayed for these types, but these types are converted to incorrect TTCN types (to XSD.ID instead of the user-defined ID type).
- › A new bug report has been created for this issue:
- › https://bugs.eclipse.org/bugs/show_bug.cgi?id=533543

BUG 519945 - DETERMINE THE CORRECT IPL4ASP_USER_CTRLFUNCTDEF.CC TEMPLATE FOR TRANSLATION PORTS



- › If the IPL4 testport is used with translation ports this code should be added to every function in the IPL4asp_User_CtrlFuncDef.cc.
- ›

```
PORT* p = portRef.get_provider_port();  
if (p == NULL) {  
    // Should be impossible  
}
```
- ›

```
IPL4asp__PortType::IPL4asp__PT_PROVIDER& provider_port = static_cast<IPL4asp__PortType::IPL4asp__PT_PROVIDER&>(*p);
```
- › But the "Should be impossible" part is not really impossible. So maybe this should be modified to this code:
- ›

```
PORT* p = portRef.get_provider_port();  
if (p == NULL) {  
    p = &portRef;  
}
```
- ›

```
IPL4asp__PortType::IPL4asp__PT_PROVIDER& provider_port = static_cast<IPL4asp__PortType::IPL4asp__PT_PROVIDER&>(*p);
```
- › IPL4asp_User_CtrlFunc.ttcn
- › IPL4asp_User_CtrlFuncDef.cc
- › added to IPL4asp demo/TranslationPortUserCtrl

BUG 533543 - XSD2TTCN: INCORRECT CODE GENERATED FOR USER-DEFINED TYPE WITH BUILT-IN NAME



- › If the name of a user-defined type in the XSD coincides with the name of a built-in XSD type, then the xsd2ttcn converter generates the TTCN equivalent of the built-in XSD type instead of the user-defined type.
- › Example:
 - › <?xml version="1.0" encoding="UTF-8"?>
 - › <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
 - › xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 - › xmlns:this="www.example.org/test"
 - › targetNamespace="www.example.org/test">
 - › <simpleType name="ID">
 - › <restriction base="xsd:anyURI" />
 - › </simpleType>
 - › <complexType name="TypeWithIDs">
 - › <sequence>
 - › <element name="id1" type="this:ID" />
 - › <element name="id2" type="xsd:ID" />
 - › </sequence>
 - › </complexType>
 - › </xsd:schema>
- › Converts to:
 - › ...
 - › type XSD.AnyURI ID;
 - ›
 - › type record TypeWithIDs
 - › {
 - › XSD.ID id1, // <-- this should use the local ID type, not XSD.ID
 - › XSD.ID id2
 - › };
 - › ..
 - › . Fixed.
 - › Changed the name-converter algorithm to also take the type's namespace into account.
 - › Regression test cases adjusted to test the new behavior.

BUG 533767 - RAW ENCODER ALIGN(RIGHT) IS WORKING ACCORDING OF SPECIFICATION OF ALIGN(LEFT) (AND VICE VERSA) FOR OCTETSTRING



- › Created attachment 273668 [details]
- › Corrected Annex_E_variants.ttcn

- › In regression_test/RAW/Annex_E_variants/Annex_E_variants.ttcn, there are testcases to be corrected. Their expectation were wrong, the ALIGN(left) and ALIGN(right) were changed with each other.
- › For example this is a corrected test and it fails after correction. (All test attached).
- › SRC:

- › type component TempComp {}

- › type octetstring RAW_PDU_18
- › with { variant "FIELDLENGTH(4)";
- › variant "BYTEORDER (first)";
- › encode "RAW"
- › };
- › external function enc_RAW_PDU_18(in RAW_PDU_18 pdu) return octetstring with { extension "prototype(convert) encode(RAW)" }
- › external function dec_RAW_PDU_18(in octetstring stream) return RAW_PDU_18 with { extension "prototype(convert) decode(RAW)" }

- › testcase TC_FIELDLENGTH_4_and_BYTEORDER_first_for_octetstring() runs on TempComp {
- › const RAW_PDU_18 i := '12345678'O
- › const octetstring o := '12345678'O

- › if ((enc_RAW_PDU_18(i) == o)and(dec_RAW_PDU_18(o) == i)) {setverdict(pass);}
- › else {setverdict(fail,"enc_RAW_PDU_18(i)= ", enc_RAW_PDU_18(i), "; dec_RAW_PDU_18(o)= ", dec_RAW_PDU_18(o));}
- › }

- › testcase TC_FIELDLENGTH_4_for_octetstring_3_octets_long() runs on TempComp {
- › const RAW_PDU_18 i := '123456'O
- › const octetstring o := '00123456'O

- › if ((enc_RAW_PDU_18(i) == o)and(dec_RAW_PDU_18(o) == '00'O & i)) {setverdict(pass);}
- › else {setverdict(fail,"enc_RAW_PDU_18(i)= ", enc_RAW_PDU_18(i), "; dec_RAW_PDU_18(o)= ", dec_RAW_PDU_18(o));}
- › }

BUG 533767 - RAW ENCODER ALIGN(RIGHT) IS WORKING ACCORDING OF SPECIFICATION OF ALIGN(LEFT) (AND VICE VERSA) FOR OCTETSTRING



- › LOG:
- › Test case TC_FIELDLENGTH_4_and_BYTEORDER_first_for_octetstring started.
- › setverdict(pass): none -> pass
- › Test case TC_FIELDLENGTH_4_and_BYTEORDER_first_for_octetstring finished. Verdict: pass
- › Test case TC_FIELDLENGTH_4_for_octetstring_3_octets_long started.
- › setverdict(fail): none -> fail reason: "enc_RAW_PDU_18(i)= '12345600'O; dec_RAW_PDU_18(o)= '00123456'O", new component reason: "enc_RAW_PDU_18(i)= '12345600'O; dec_RAW_PDU_18(o)= '00123456'O"
- › Test case TC_FIELDLENGTH_4_for_octetstring_3_octets_long finished. Verdict: fail reason: enc_RAW_PDU_18(i)= '12345600'O; dec_RAW_PDU_18(o)= '00123456'O
- › The list of testcases to be corrected in function_test/RAW:
 - › - TC_FIELDLENGTH_4_for_octetstring_3_octets_long
 - › - TC_FIELDLENGTH_5_for_octetstring_2_octets_long
 - › - TC_FIELDLENGTH_4_and_BYTEORDER_last_for_octetstring_3_octets_long
 - › - TC_ALIGN_right_for_octetstring
 - › - TC_ALIGN_left_for_octetstring
- › The bug has been corrected in titan.core. The function tests have been adjusted to test the new behavior.

BUG 534266 - CANNOT MAP TRANSLATION PORTS ON OTHER COMPONENTS



- › Mapping ports in translation mode does not work if they are on a different component, than the one the map statement is initiated from.
- › Example:

```
type port P5 message {  
  in integer  
  out charstring  
} with {  
  extension "provider"}
```
- ›

```
type port PT3 message map to P5 {  
  in integer from integer with int_to_int()  
  out charstring to charstring with char_to_char()  
}
```
- ›

```
type component MyComp {  
  port PT3 pt3;  
}
```
- ›

```
type component System {  
  port P5 p5;  
}
```
- › function f_map() runs on MyComp2 system System {

```
  var MyComp ct := MyComp.create alive;  
  map(ct:pt3, system:p5);  
}
```
- › Currently this causes a DTE, stating that port p5 does not exist. This is because the ports of the system component have not been initialized and started on the newly created PTC's (ct's) process.
- › Unmapping similarly doesn't work, but it doesn't display a DTE (the MC only sends the unmap request to the PTC if the mapping is actually found in the MC's database).

BUG 534266 - CANNOT MAP TRANSLATION PORTS ON OTHER COMPONENTS



- › Change implemented.
- › Moved the activation and start of the system port inside the runtime library's mapping and unmapping functions ('map_port' and 'unmap_port'), (these are no longer generated in the C++ code before each mapping and unmapping). This required the generation of a new function that activates and starts a given port in the module.
- › Moved the 'add_port' and 'remove_port' function calls to the runtime library's mapping and unmapping functions (these were also generated in the C++ code, after the mapping or unmapping call). The prototypes of these functions have been changed. There is now only one 'add_port' and 'remove_port' function in each port class, whose argument is a pointer of the built-in port type (PORT*), and it is later converted to the appropriate type before used.
- › Regression tests added under regression_test/portTranslation.

BUG 534537 - XSD2TTCN: INVALID SUBTYPE GENERATED FOR RESTRICTED XS:LONG



- › The XSD converter doesn't merge the type restrictions of a simpleType with the restrictions of the built-in type 'long'.
- › Example:
 - › `<xs:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.onem2m.org/xml/protocols"`
 - › `xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xs="http://www.w3.org/2001/XMLSchema"`
 - › `elementFormDefault="unqualified">`
 - › `<xs:simpleType name="longMin-1">`
 - › `<xs:restriction base="xs:long">`
 - › `<xs:minInclusive value="-1" />`
 - › `</xs:restriction>`
 - › `</xs:simpleType>`
 - › `</xs:schema>`
- › Generated type:
 - › type XSD.Long LongMin_1 (-1 .. infinity)
 - › with {
 - › variant "name as 'longMin-1'";
 - › };
- › Expected type:
 - › type XSD.Long LongMin_1 (-1 .. 9223372036854775807)
 - › with {
 - › variant "name as 'longMin-1'";
 - › };

BUG 534678 - PERFORMANCE ISSUES CAUSED BY TEMPLATE CONCATENATION FEATURE



- › The template concatenation feature causes C++ compilation performance issues in certain cases, notably in `UTRAN_RRC_ASN1_Definitions.asn` (attached).
- › With TITAN 6/R1A the code generated from the ASN.1 file (split into 8 pieces) compiles in about 15-20 minutes, in Runtime 2 (using `make -j4`). With TITAN 6/R2A (where the feature was implemented) the first 7 segments take 10+ hours to compile, and the 8th segment takes days.
- › Changed the handling of optional fields in template concatenations. Now the compiler generates extra code to convert references to optional fields (which would use the C++ template class `OPTIONAL<...>`) into values or templates of that type instead. All concatenation operator functions that take an `OPTIONAL<...>` as argument have been removed from the code and from the generated code. (These were previously needed, because otherwise the C++ compiler couldn't determine which type to convert the `OPTIONAL` value to.)
- › With these changes the compilation time of the mentioned ASN.1 file is down to 22-24 minutes with TITAN 6/R4A (the performance issues were likely caused by the arguments of type `OPTIONAL` generated for the concatenation of record of/set of types).

BUG 534769 - OER DEC: DECODING DEFAULT FIELD CAUSES DTE IN RT2



- › In Runtime2 the OER decoder tries to set fields with default value to 'omit', if their presence bits are unset. This causes a DTE (Internal error: trying to set a non-optional field to OMIT).

BUG 535051 - RUNS ON SCOPE REDUCTION



- › I propose a new code smell to be detected by **Titanium**.
- › In many cases we have seen that the "runs on" components on functions is not "tight". Meaning that no definition is actually used from the explicitly named component.
- › It would be great if Titanium could be extended with detecting and reporting the cases when the component a function/altstep/testcase runs on can be "lowered" (in the sense of replacing the reference with a reference to a "parent" component in it's extension hierarchy).
- › And possibly also give some hints on which might be the best component to preference based on the contents of the function/altstep/testcase.
- › The feature is now available under the name "runs on scope reduction"

BUGZILLA NOVEMBER- DECEMBER



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
521410	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	ASN.1 named bit semantic check results false error	11/23/2017 4:28
527623	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Implement verdict redirect for 'done' statement	11/28/2017 9:32
527793	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	TEXT codec bug	11/30/2017 7:08
527653	Titan	Core	bence.janos.szabo@ericsson.com	CLOSED	FIXED	Implement extendable sequence coding in OER	12/5/2017 4:13
527847	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	str2float should handle special float values	12/8/2017 5:28
528082	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	str2float("not_a_number") == not_a_number throws an exception in titan.core	12/8/2017 10:42
528313	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Type descriptor generation fault for ASN.1 open types	12/11/2017 3:36

BUGZILLA NOVEMBER- DECEMBER



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
526094	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	semantic analyzer: NPE at getting location in a not ready right hand side	11/20/2017 9:33
526515	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	designer NPE if the right side of an assignment is empty	11/20/2017 9:36
525623	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty semantic error markers on variable of type of setof if "not used symbols" were used	11/20/2017 9:42
527905	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Semantic analyzer error: Not accepted strings in str2float(): "infinity", "-infinity", "not_a_number"	11/29/2017 7:08
528897	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing error marker on an argument of replace()	12/20/2017 9:05

BUGZILLA JANUARY



527946	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	RT2 record equality	1/5/2018 4:38
529019	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Erroneous behaviour when default value of template parameter originates in an external function	1/17/2018 10:27
529017	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	OER codec error	1/18/2018 4:22
529855	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	coverage in 6.3.0	1/18/2018 10:33
529655	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	string2ttcn to filter patterns of visible characters in octetstrings	1/18/2018 10:36
530502	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Code generation problems caused by the default template parameter modification	1/30/2018 7:00
529359	Titan	Plug-ins	arpad.lovassy@semcon.com	CLOSED	FIXED	NPE when deleting from a char quadruple	1/3/2018 6:15
529424	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	unichar2int(arg): arg universal charstring initialized with charstring triggers an NPE in semantic checker	1/5/2018 9:27
529015	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Missing error marker on empty list as first argument of substr()	1/5/2018 9:28
529399	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Wrong universal charstring triggers an NPE in semantic checker	1/5/2018 9:29
529636	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Semantic analyzer: NPE at string concatenation if second operand is missing	1/30/2018 3:42
529455	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty semantic error marker on argument of sizeof if the argument is func parameter value	1/30/2018 3:43
530487	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	ConsoleTimeStampFormat in cfg file freezes the running of the executable	1/30/2018 8:50

BUGZILLA FEBRUARY



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
529888	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	JSON variant attributes: use standard syntax	2/7/2018 8:00
529892	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Syntax to bind a variant attribute to multiple encodings	2/7/2018 8:01
526554	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	ASN.1 teletextstring value assignment error	2/8/2018 9:32
531109	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	TITAN build on Alpine Linux	2/19/2018 7:58
531339	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Add discarding option to 'setstate' operation	2/20/2018 8:11
530541	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	ConsoleTimeStamp is not filtered out from TITAN notification	2/2/2018 5:33
530221	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Install Guide: Minimal Eclipse version shall be changed for 4.2.0, suggested version: 4.7.1a	2/8/2018 9:06
521502	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty error marker 'Invalid working directory' while opening a project	2/12/2018 10:42
530656	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	ConsoleTimeStamp is not supported by the cfg file editor	2/12/2018 10:44
531391	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty property change handling in builder	2/20/2018 10:01
531416	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty type check for formal parameters of type 'sequence of'	2/21/2018 2:55
531569	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Faulty semantic error marker on match(vl_integers,?)	2/23/2018 2:53
526425	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Designer: Starting functions with 'out' and 'inout' parameters	2/23/2018 8:58

BUGZILLA MARCH



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
531904	Titan	Core	jeno.balasko@ericsson.com	CLOSED	FIXED	Document new tpd tag disableUserInformation in the referenceguide.doc	3/2/2018 3:16
528465	Titan	Core	titan-inbox@eclipse.org	RESOLVED	FIXED	TAG and CROSSTAG for JSON encoder	3/2/2018 5:41
532021	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Notify user if port is not mapped in translation mode	3/6/2018 6:30
532060	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	'done' operation on PTC causes internal error	3/6/2018 8:29
532066	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Compiler performance issue (with OneM2M_Tester project)	3/8/2018 8:22
532316	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Verdict redirects do not work on 'alive' components	3/14/2018 5:41
532451	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Port name clash error during unmapping	3/28/2018 10:11
533006	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Implement reference to port in translation function	3/29/2018 7:41
509270	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Run button is missing in the Executor perspective	3/2/2018 3:10
531933	Titan	Plug-ins	jeno.balasko@ericsson.com	RESOLVED	FIXED	Symlink creation should be forced for external Makefile generation	3/2/2018 10:11
531906	Titan	Plug-ins	jeno.balasko@ericsson.com	RESOLVED	FIXED	Update Designer to support the new tpd element 'disableUserInformation'	3/2/2018 10:12
531343	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	WONTFIX	Stack overflow for long string concatenation	3/6/2018 6:52

BUGZILLA APRIL



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
533061	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	OER dec: extension fields are not set to omit	4/4/2018 6:21
533067	Titan	Core	botond.baranyi@ericsson.com	CLOSED	FIXED	RAW encoder froze in load test runtime	4/13/2018 6:49
533487	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	JSON attribute 'chosen' does not use default value	4/13/2018 7:03
533242	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	xsd2tcn 6.3.0 error	4/13/2018 7:14
533543	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	xsd2tcn: incorrect code generated for user-defined type with built-in name	4/18/2018 10:40
533767	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	RAW encoder ALIGN(right) is working according of specification of ALIGN(left) (and vica versa) for octetstring	4/25/2018 11:42
531906	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Update Designer to support the new tpd element 'disableUserInformation'	4/18/2018 10:32
531933	Titan	Plug-ins	jeno.balasko@ericsson.com	CLOSED	FIXED	Symlink creation should be forced for external Makefile generation	4/18/2018 10:34
533419	Titan	Plug-ins	arpad.lovassy@semcon.com	CLOSED	FIXED	StringIndexOutOfBoundsException in PredefFunc.is_utf8()	4/18/2018 10:35
519945	Titan	TestPorts	titan-inbox@eclipse.org	CLOSED	FIXED	Determine the correct IPL4asp_User_CtrlFuncDef.cc template for translation ports	4/3/2018 7:44

BUGZILLA MAY



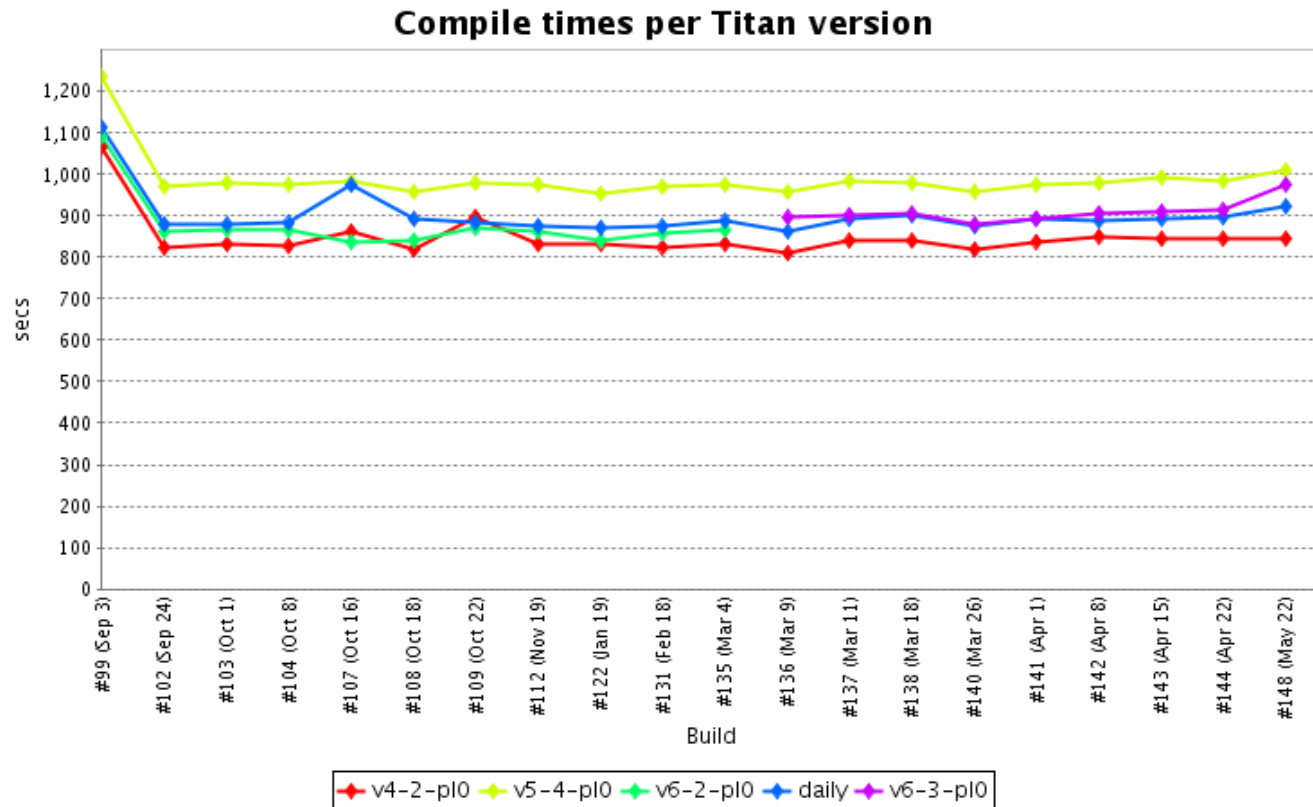
Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
534266	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Cannot map translation ports on other components	5/10/2018 9:11
534537	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	xsd2ttcn: invalid subtype generated for restricted xs:long	5/11/2018 4:01
534769	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	OER dec: decoding default field causes DTE in RT2	5/17/2018 8:52
534678	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Performance issues caused by template concatenation feature	5/17/2018 8:54
528465	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	TAG and CROSSTAG for JSON encoder	5/17/2018 8:55
534431	Titan	Plug-ins	jeno.balasko@ericsson.com	RESOLVED	FIXED	Semantic analyzer: NPE at the first re-analyzing	5/7/2018 9:15
535051	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	runs on scope reduction	5/24/2018 3:26

DOCUMENTATION

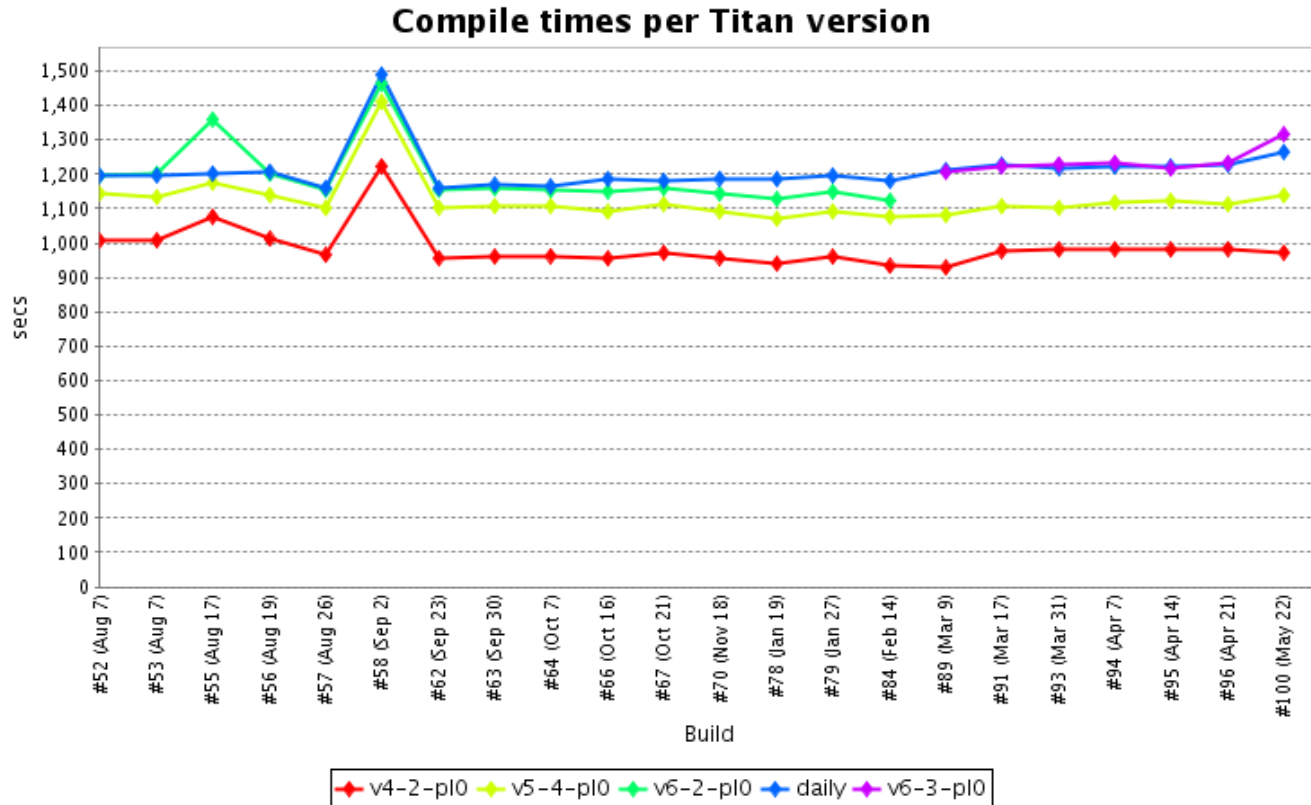


CRL 113 200/6 R4A		
109 21	CRL 113 200/6 4Uen Rev.A	PRI
109 47	CRL 113 200/6 Uen Rev. D	Release Notes
1/1531	CRL 113 200/6 Uen Rev.D	Install. Guide for....Test Executor
3/1531	CRL 113 200/6 Uen Rev.D	Install. Guide for....TITAN Designer and..... Eclipse IDE
1551	CRL 113 200/6 Uen Rev.D	Titanium Description
1/174 02	CRL 113 200/6 Uen Rev.D	Statement of Compliance for Eclipse Titan
2/174 02	CRL 113 200/6 Uen Rev.D	Statement of Compliance for use of XML schema in Eclipse Titan
1/198 17	CRL 113 200/6 Uen Rev.D	UG for Titan TTCN-3 Test Executor
2/198 17	CRL 113 200/6 Uen Rev.D	Programmers Tech. Reference Guide for the TITAN executor
4/198 17	CRL 113 200/6 Uen Rev.D	UG for the TITAN Designer for the Eclipse IDE
5/198 17	CRL 113 200/6 Uen Rev.D	UG of the TITAN Executor for the Eclipse IDE plug-in
6/198 17	CRL 113 200/6 Uen Rev.D	API Technical Reference for the TITAN executor
8/198 17	CRL 113 200/6 Uen Rev.D	Titan Executor API User Guide
9/198 17	CRL 113 200/6 Uen Rev.D	Programmers Tech. Reference Guide for Titanium

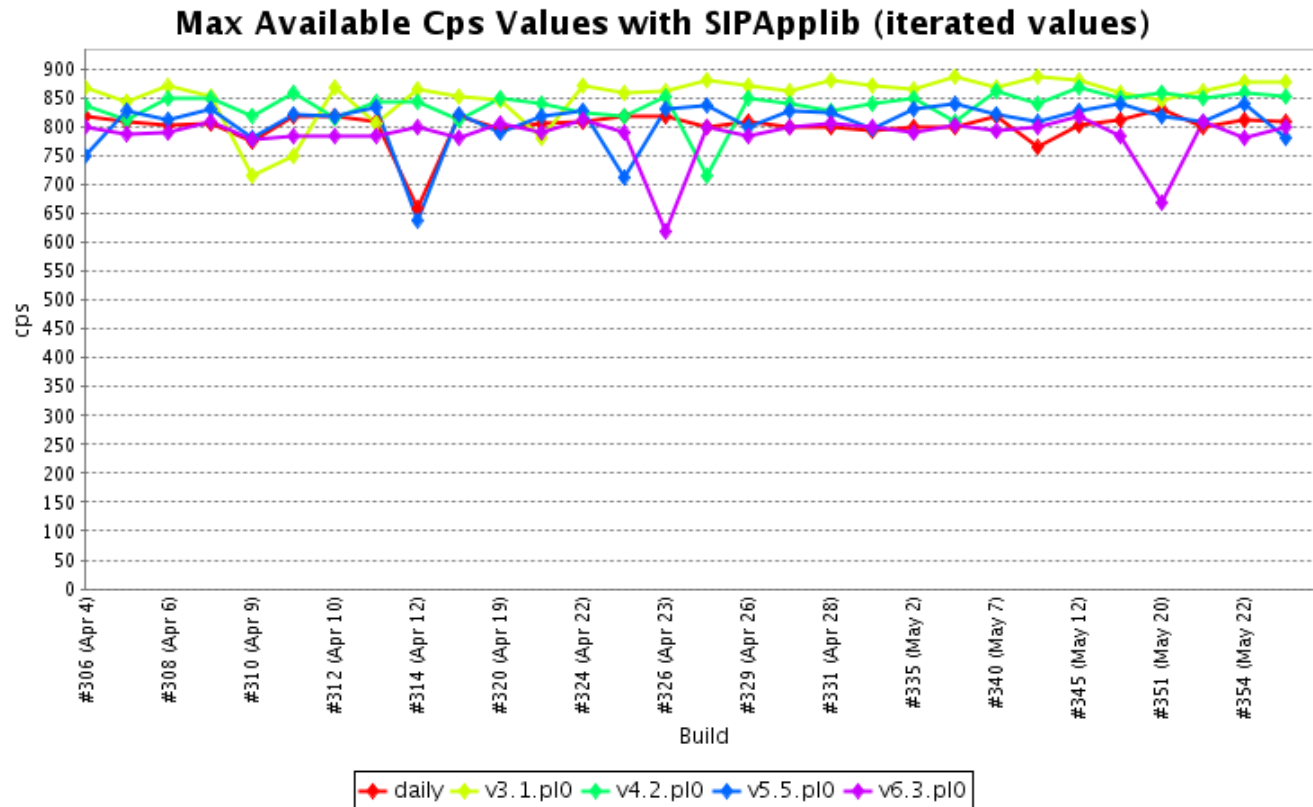
COMPILE TIMES RT1



COMPILE TIMES RT2



EXECUTION PERFORMANCE RT1





ERICSSON