TITAN 7.2.0 CHANGE LOG (RELATIVE TO 7.1.0)

2020.11.18

HIGHLIGHTS

- Changes in Titan product code(Bug 565875)
- Changes in licensing(relevant for internal users only):
- -License ordering/extension page moved
- -License ordering/extension API to automate license renewal
- Changes in shipping libedit (Bug 565893)
- Titan image added to ADP marketplace(relevant for internal users only)
- Relaxing the C/C++ compiler version check for Titan 7.1.1 and upwards
- (Bug 564194)

HIGHLIGHTS(CONT.)

- Object-oriented language features (Bug 563718)
- Allow unsafe universal charstring to charstring conversion (Bug 564585)
- Support for @default in the compiler and in the xsd2ttcn converter (Bugs 564919, 564920)
- Java-based Main Controller implemented
- Java-based JSON codec implemented

CHANGES IN TITAN PRODUCT CODE

- To acknowledge that Titan is de facto an open-source product, hence it should be handled administratively as FOSS, the Titan product code has been changed from CRL 113 200 to CAX 105 7730.

- Accordingly, Titan 7.1.1 is designated as CRL 113 200/7 R1B, but 7.2.0 has become 7/CAX 105 7730 R2A.

- See https://bugs.eclipse.org/bugs/show_bug.cgi?id=565875

CHANGES IN TITAN PRODUCT CODE

- Accordingly, the product number in the product manifest, displayed with compiler -v:
- Changes from:
- TTCN-3 and ASN.1 Compiler for the TTCN-3 Test Executor
- Product number: **CRL 113 200/7 R1B**
- Version: 7.1.pl1
- To:
- TTCN-3 and ASN.1 Compiler for the TTCN-3 Test Executor
- Product number: **7/CAX 105 7730 R2A**
- Version: 7.2.pl0

CHANGES IN TITAN PRODUCT CODE

- Specifying minimum compiler version requirements accepts both the legacy format:
- Module mymodule {
- } extension "requiresTITAN **CRL 113 200/7 R1B**";

- And the new format:
- Module mymodule {
- } extension "requiresTITAN **7/CAX 105 7730 R2A**";

LICENSE PAGE MOVED

- The URL of the Titan license ordering page has changed
- The new URL is:
- **http://ttcn.ericsson.se/license/**

- (The old page https://tcc-licorder.rnd.ki.sw.ericsson.se/
- Displays the following message:
- Web site has been migrated.
  From: **http://tcc-licorder.rnd.ki.sw.ericsson.se**
  To: **http://ttcn.ericsson.se/license**
  Current link is: **http://ttcn.ericsson.se/license/**
  Please copy the link/s to the browsers adress field and save the bookmark.)

LICENSE ORDERING/EXTENSION API

- Call to this API will return a license file for hostid 11223344. If the license already exists, it will be extended.
- http://ttcn.ericsson.se/license/service.php?key=6a5e234a7890fd&hostid=11223344
-
- Extending license with license nr. license_num:
- http://ttcn.ericsson.se/license/service.php?id=<license_num>

- This API can be used to automate license renewal.

LICENSE ORDERING/EXTENSION API

- Usage examples:

- wget http://ttcn.ericsson.se/license/service.php?id=72
-   -O license_72.dat

- curl -L -o license_72.dat http://ttcn.ericsson.se/license/service.php?id=72

CHANGES IN SHIPPING LIBEDIT

- Libedit has been shipped as an embedded source code; the last version of Titan which includes Editline library in source was 7.1.1. In later versions, Titan will rely on the Editline library being installed on the Linux distribution it executes on. The Editline source code was removed from the repository.

- For details see:

- https://www.eclipse.org/forums/index.php/t/1105539/

TITAN IN ADP MARKETPLACE

- A Titan Docker image, built upon the Common Base OS (SLES 15) has been uploaded to the ADP Marketplace, together with a Helm Chart that can be used to deploy it in a Kubernetes cluster.

- See https://adp.ericsson.se/marketplace/titan

RELAXING THE C/C++ COMPILER VERSION CHECK FOR TITAN 7.1.1 AND UPWARDS

- For details,

- See https://www.eclipse.org/forums/index.php/t/1104359/

OBJECT ORIENTED LANGUAGE FEATURES

- These features are based on an extension of the core language standard, published in:
- Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Object-Oriented Features
- Baseline is the following version ETSI ES 203 790 V1.1.1 (2019-01)

OBJECT ORIENTED LANGUAGE FEATURES

- As these features are not part of the core language standard, they are optional and can be activated with a compiler switch:
- -k: enable object-oriented features
- C++ code skeletons are generated for external classes

with the same switch that has been used for test port skeleton generation:

-t: generate skeletons for test ports **and external classes**

OBJECT ORIENTED LANGUAGE FEATURES

- Implementation started with 7.1.0, see list of features:
- https://bugs.eclipse.org/bugs/show_bug.cgi?id=552011

- And continued in 7.1.1, 7.2.0 , see the new list:
- https://bugs.eclipse.org/bugs/show_bug.cgi?id=563718

OBJECT ORIENTED LANGUAGE FEATURES

- Implementation is not yet complete (e.g. exception handling has not yet been covered) and will continue in upcoming releases. Also the implementation will be aligned to the release 1.2.1 of the standard.
- Usage examples for data structures will be published.

ALLOW UNSAFE UNIVERSAL CHARSTRING TO CHARSTRING CONVERSION

- See https://bugs.eclipse.org/bugs/show_bug.cgi?id=564585
- the default behavior is unchanged, a compiler error is thrown when a universal charstring value is assigned to a charstring:

- char2uni.ttcn:12.36-48: error: Type mismatch: a value of type `charstring' was expected instead of `universal charstring'

ALLOW UNSAFE UNIVERSAL CHARSTRING TO CHARSTRING CONVERSION

- -when using compiler option -h
- -h: allow unsafe universal charstring to charstring conversion

- assignment of universal charstring to charstring is permitted;
- if universal charstring consists of ISO 646 characters only, this will lead to a valid result,
- else a DTE is thrown:

- char2uni.ttcn:12 Dynamic test case error: Non-ASCII characters cannot be used to initialize a charstring, invalid character char(0, 0, 1, 113) at index 3.
-

- By using a compiler option, which requires an affirmation from the user, accidental unsafe usage of unichar 2 char assignment is prevented.

PART1: TITAN 7.1.1 CHANGE LOG(RELATIVE TO 7.1.0)

2020.06.28

SUMMARY

- This is a corrective release, with focus on 564194, and partly on 563718 (OO features)

BUG 540255 - MATCHING OPTIONAL FIELD HAVING VALUE "OMIT" WITH TEMPLATE ANYOROMIT (*) CAUSED DTE

- This code finished with DTE:

```
module ModuleparTestFloat {
type component CT{}
type record R {
  float f optional
}
testcase tc_modulepartemplate_float_anyoromit4()runs on CT {
  log("1");
  var R vl_r := { f:=omit }
  log("2")
  log(vl_r.f);
  log("3");
  template float t_f := *
  if( match(vl_r.f, t_f)) {
    setverdict(pass)
  } else {
    setverdict(fail)
  }

}
}
```

- **fixed, tested.**

BUG 563718 - OBJECT-ORIENTED FEATURES (CONTINUATION)

- Continuation of https://bugs.eclipse.org/bugs/show_bug.cgi?id=552011
- Stage 7:
- Implemented semantic checks for final methods (previously only final classes were checked).

- Implemented the 'select class' statement, the 'of' operator (dynamic class discrimination) and the casting operator ('=>’).
  Operator '=>' can now cast to abstract classes (including 'object’).
  (to be continued)

- Limitation: - subreferences cannot be applied to the result of casting (e.g. '(obj => ClassType).field').

BUG 564194 - ADAPT TITAN COMPILER VERSION CHECKING TO NEW GCC/CLANG VERSIONING SCHEME

- Titan demands major and minor levels of gcc/clang to be equal and does not consider the patch level.
- This practice was correct up to GCC4 being consistent with gcc versioning.
- https://gcc.gnu.org/develop.html

- From GCC5 gcc version numbering changed and uses only three numbers:

- <major_version>.<patch_level>.0

- The third number does not refer to patch level as before but to the version being a released or development one:
- 0 – released
- 1 – dev branch

- So for instance:
- 7.4.0 – released, 4th patch of the GCC7
- 7.4.1 – The dev branch during the development of the 4th patch of the GCC7, after release become 7.4.0

BUG 564194 - ADAPT TITAN COMPILER VERSION CHECKING TO NEW GCC/CLANG VERSIONING SCHEME

- Clang also dropped minor versions from Clang5.
- so in summary the version numberings are:
- GCC: <major>.<patch>.0
- Clang: <major>.0.<patch>

- Titan's compiler version checking should be adapted to this new versioning scheme.
- see also:

- https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=954999 and
- https://github.com/eclipse/titan.core/pull/128

- It appears that newer gcc compilers of versions generate compatible binaries;
- maybe we should relax compiler version checking and issue a warning only, instead of restricting linking.

BUG 564194 - ADAPT TITAN COMPILER VERSION CHECKING TO NEW GCC/CLANG VERSIONING SCHEME

- It has been tested:
-
- -for a gcc interval of versions: Titan (libraries) built with gcc 10.1 ,
- project built with gcc 5.5
- -for a clang interval of versions: Titan built with clang 10, tests built with clang 5
- -with static and dynamic linking
- -both with the gcc linker and gold linker

- clang and gcc-built libraries should not be mixed though: although technically possible, there's no visible advantage and should be avoided

- see also:

- https://gcc.gnu.org/onlinedocs/libstdc++/manual/abi.html
- https://gcc.gnu.org/onlinedocs/libstdc++/manual/using_dual_abi.html

- https://stackoverflow.com/questions/11682748/is-clang-abi-same-as-g

BUG 564441 - INCORRECT C++ COMPILER GENERATED TO MAKEFILES WHEN USING CLANG 7 OR NEWER

- The makefile generator always appends the major and minor release numbers to the C++ compiler, when using clang.
- e.g.: clang++-5.0

- This is good for older versions of clang, however, starting from clang 7 only the major number is present.
- e.g.: clang++-10
- Updated the makefilegen's generation of the C++ compiler accordingly.

BUGZILLA JUNE

540255	Titan	Core	jeno.balasko@ericsson.com	CLOSED	FIXED	Matching optional field having value "omit" with template anyOrOmit (*) caused DTE	6/8/2020 1:16
564031	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	TitaniumRefactoring menu shows up for logviewer testcases	6/9/2020 11:25
564023	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	emergency logging causes memory leak in Java runtime during execution	6/9/2020 11:26
543091	Titan	Plug-ins	titan-inbox@eclipse.org	RESOLVED	FIXED	incorrect error message on variant specifications.	6/9/2020 11:27
547163	Titan	Website	lenard.nagy@ericsson.com	CLOSED	FIXED	titan github contributor workflow fails to mention hooks/commit-msg	6/10/2020 3:23
526159	Titan	Core	lenard.nagy@ericsson.com	RESOLVED	FIXED	Write html help to "extends"	6/11/2020 7:28
564235	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	incorrect package info header generated	6/12/2020 4:38
564236	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	sometime an error is reported by Eclipse for not existing package, in builds after cleans	6/12/2020 4:42
563718	Titan	Core	botond.baranyi@ericsson.com	ASSIGNED	---	Object-oriented features (continuation)	6/16/2020 8:34
564441	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Incorrect C++ compiler generated to Makefiles when using clang 7 or newer	6/19/2020 6:50
564555	Titan	Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	Semantic checking can deadlock without reporting error	6/23/2020 5:15
564194	Titan	Core	titan-inbox@eclipse.org	RESOLVED	FIXED	Adapt Titan compiler version checking to new gcc/clang versioning scheme	6/25/2020 9:37

PART 2: TITAN 7.2.0 CHANGE LOG(RELATIVE TO 7.1.1)

2020.11.19

BUG 565952 - FALSE COMPILER ERROR FOR 'ACTIVATE' PARAMETER

- The compiler displays a false error for the parameter of an 'activate' operation, if it refers to an altstep from another module (i.e. 'activate(mod.as(...))').
- Fixed. (References are now checked before determining whether they have parameters or not.)

BUG 564919 - SUPPORT FOR @DEFAULT IN THE COMPILER

- Support for @default in unions as per para. 6.2.5.0

- Implemented the '@default' modifier with the clarifications/limitations described in section 4.39 of the reference guide.

BUG 564920 - SUPPORT FOR @DEFAULT IN XSD2TTCN

- Support for @default when converting XSD Schemas to TTCN-3 ,see examples in ETSI ES 201 873-9 V4.11.1 (2020-05)
  Note: Support for @default has to be implemented first in the compiler, see https://bugs.eclipse.org/bugs/show_bug.cgi?id=564919
- Change implemented.
- The union types generated for element substitutions and type substitutions now have the '@default' modifier set for their first (head) field.

BUG 563718 - OBJECT-ORIENTED FEATURES (CONTINUATION)

- Continuation of https://bugs.eclipse.org/bugs/show_bug.cgi?id=552011
- Stage 7:
- Implemented semantic checks for final methods (previously only final classes were checked).
- Implemented the 'select class' statement, the 'of' operator (dynamic class discrimination) and the casting operator ('=>’).
- Operator '=>' can now cast to abstract classes (including 'object').

BUG 563718 - OBJECT-ORIENTED FEATURES (CONTINUATION)

- Limitation:
- - subreferences cannot be applied to the result of casting (e.g. '(obj => ClassType).field’).
- Fixed usage of standalone 'this' as actual parameter to a function call.
- Fixed code generated for aliases of class types.

- Added support for the following module declaration:
- module <name> "TTCN-3:2018 Object-Oriented" { ... }

BUG 563718 - OBJECT-ORIENTED FEATURES (CONTINUATION)

- Implemented the external class skeleton generator. It uses the same command line option as the test port skeleton generator ('-t’).
- Class members can no longer be public.
- Fixed class method calls in 'in' actual parameters.
- Hotfix: When checking whether a reference can be generated as a single expression, the subreferences are taken into account, too.
- Added semantic error for using class method return values as 'out' / 'inout' parameters.

BUG 563718 - OBJECT-ORIENTED FEATURES (CONTINUATION)

- Removed the extra semantic checks on class method return values that did not affect 'out' and 'inout' parameters.
- Updated OOP limitations and clarifications in the reference guide.
- Documented the external class skeleton generator in the API guide.

BUG 563718 - OBJECT-ORIENTED FEATURES (CONTINUATION)

- This bug report only details the parts of the feature, that were implemented between TITAN version 7.1.pl0 and version 7.2.pl0.
- Continuation: https://bugs.eclipse.org/bugs/show_bug.cgi?id=568899

BUG 568774 - TIMESTAMP PROBLEM AFTER 2038 FOR TESTCASE START TIMES

- The issue that produced bug563289, is also present with testcase start times.
- They also need to be treated like 64 bit numbers.

- fixed/closed

BUG 568592 - UNDER SPECIAL CIRCUMSTANCES, COMPILATION GETS STUCK AT "CHECKING MODULES..."

- We used xsd2ttcn to generate code from ETSI specification for X1 types and ran into the problem that compilation would get stuck forever at "Checking modules...".
- A minimal example:
- type XSD.AnyURI TELURI (pattern "[a-zA-Z\q{0, 0, 0, 38}&-;]");
- There was an issue while merging set elements in patterns. It has been corrected.

BUG 568000 - C++17 COMPATIBILITY

- The default c++ standard of the upcoming GCC11 will be c++17
- Currently the Titan can't be compiled according to the c++17
- problems:
- 1. dynamic exception specifications
- deprecated in c++11 completely removed later
- Example:
- new.cc:21:33: error: ISO C++1z does not allow dynamic exception specifications
- void *operator new(size_t size) throw (std::bad_alloc)
- Solution remove the throw() from declaration.
- Backward compatibility note: The feature was not used. Doesn't really limited the possible exceptions.
- 2. register
- deprecated in c++11, removed in c++17.
- See: https://en.cppreference.com/w/cpp/language/storage_duration
- It was a hint for the c++ compiler, but not guarantees anything. Any optimizer ignore it.
- Can be safely removed.
- Solution: remove register keyword
- Both solution are backward compatible.

BUG 565852 - ISSUES WITH REGULAR EXPRESSIONS

- Dear all,

- Working with v6.5pl1 (Windows version) we have found some problems we'd like to share.
- 1.- Crash at runtime.
- Titan does not support the notation "#()" but silently crashes at runtime => "#()" needs to be replaced by alternative natation "#(0,)" or "#(,)"
- Can you please check whether the issue still exists with your current compiler release (you can use the attached example XmlElementContentTest.ttcn and change "#(0,)" to "#()") and confirm it crashes or let us know whether it is solved on a newer version.
- I have also checked with Titan 7.1.0 built for Ubuntu 18.04 and the messages logged are the same as shown on the Windows version though I can only check on syntax analysis so I can't confirm whether it actually crashes the tool on runtime.
- 2.- Titan seems to handle charstring and universal charstring differently in regexp:
- It seems that with universal charstring expressions like "[^a]#(1,)" do not work anymore or get ignored.
- => in our TTCN we have universal charstring just because of XSD.ttcn
- => as workaround a universal charstring parameter can be converted into a charstring variable which then is handed over to regexp.
- Please find attached an example illustrating the issue (TitanIssue_Regexp.ttcn)

- Best Regards,
- Carlos

BUG 565852 - ISSUES WITH REGULAR EXPRESSIONS

- Fixed both issues.
- Also fixed a code generation fault when 'regexp' has a charstring input parameter and a universal charstring expression parameter.

BUG 564585 - INCOMPATIBILITY OF CHARSTRING AND UNIVERSAL CHARSTRING

- The below TTCN code causes Titan to raise an error even though according to the TTCN core language (ES 201 873-1 clause 6.3.1) the assignment of a universal charstring to a charstring is allowed in general.
- function f_StringTest(universal charstring p_UCharString) return charstring
- {
- var charstring v_CharString := p_UCharString; // assignment causes an error even though being allowed according to ES 201 873-1 clause 6.3.1
- return v_CharString;
- }

BUG 564585 - INCOMPATIBILITY OF CHARSTRING AND UNIVERSAL CHARSTRING

- we have amended the compiler as below:

- -the default behavior is unchanged, a compiler error is thrown when a universal charstring value is assigned to a charstring:
- char2uni.ttcn:12.36-48: error: Type mismatch: a value of type `charstring' was expected instead of `universal charstring'

- -when using compiler option -h
- -h: allow unsafe universal charstring to charstring conversion

- assignment of universal charstring to charstring is permitted;
- if universal charstring consists of ISO 646 chareters only, this will lead to a valid result,
- else a DTE is thrown:
- char2uni.ttcn:12 Dynamic test case error: Non-ASCII characters cannot be used to initialize a charstring, invalid character char(0, 0, 1, 113) at index 3.
-
- By using a compiler option, which requires an affirmation from the user, accidental unsafe usage of unichar 2 char assignment is prevented.

BUG 568552 - UNEXPECTED BEHAVIOR OF PORT.HALT

- see https://www.eclipse.org/forums/index.php/t/1105772/
- From TTCN-3 standards:
- Quote:
- 22.5.4 The Halt port operation
- The halt operation is comparable to the stop operation, but allows entries being already in the queue to be processed with receiving operations
- [...]
- Examples
- Code: [Select all] [Show/ hide]
- myPort.halt;
- // No sending allowed on myPort from this moment on;
- // processing of messages in the queue still possible.

- myPort.receive (mw_myTemplate1);
- // If a message was already in the queue before the halt
- // operation and it matches mw_myTemplate1, it is processed; // otherwise the receive operation blocks

BUG 568552 - UNEXPECTED BEHAVIOR OF PORT.HALT

- I interpreted this to mean that I could halt my port, then receive what has already arrived on the port and been enqueued. No more messages would be enqueued on the port if they arrive.
- However, when inside an alt statement, if a message arrives on the test port after halting it, I get a dynamic test case error saying "Port <name> is not started by a message has arrived on it".
- Is this optimal behavior and what was intended by the standard? It seems to me that this makes port.halt way less useful, since it risks raising an error if used after halting?

Changed the handling of incoming messages on halted ports to display a warning and discard the message instead of causing a dynamic test case error.

BUG 566094 - MAP PARAM COMPILER ERROR, FUNCTION WITH RETURN VALUE

- The compiler incorrectly reports error:
- Cannot determine system component in `map' operation with `param' clause
- if the function has return value
- Additional bug:
- The parameter is not passed to the test port if mtc is used as component ref

- function t2_init() runs on CT system CTS {
-   map(mtc:p1,system :p2) param (21) // error, parameter is not passed to user_map

- }

BUG 566094 - MAP PARAM COMPILER ERROR, FUNCTION WITH RETURN VALUE

- Fixed the first issue.
- Fixed the 2nd issue as well.

BUG 565893 - LIBEDIT SHIPPED IN OTHER FORM THAN EMBEDDED SOURCE

- The source code of 'editline' (libedit) has been removed from the TITAN repository. Instead, the 'editline' library is now linked to the 'mctr_cli' executable. It is also linked to the executable built from the generated C++ code, if TITAN was built using the 'ADVANCED_DEBUGGER_UI' option (which is disabled by default).
- A new TITAN build option can now be set in Makefile.personal, called 'OLD_LIBEDIT'. It must be set to 'yes' if the version of the 'editline' library is 0:53:0 or older (i.e. the first line of the 'readline.h' include file contains 'v1.34' or lower).

BUG 568645 - JSON DECODER BUG/INCONSISTENCY

- The JSON decoder wrongly decodes the union member:
- The JSON decoder ignores extra fields, but only if the extra fields are either number or string.

- Fixed.

BUG 568340 - CHARSTRING PATTERN RESTRICTION AND EMPTY STRING

- The following pattern subtyping not matches to ""

- type charstring api_prefix (pattern "(/?+)#(0,1)")

- var api_prefix vl_p1 := "/a/b/c/d" // OK
- var api_prefix vl_p2 := "" // error: "" is not a valid value for type `charstring' which has subtype pattern ((/?+)#(,1))

- But the pattern allows ""

- log(match("", pattern "(/?+)#(0,1)"))
- ->
- "" with pattern "(/?+)#(,1)" matched

- Fixed.

BUG 567844 - LOGGING ERROR WHEN MATCHING WITH A UNION TEMPLATE IS UNSUCCESSFUL

- When logging the result of an unsuccessful matching operation with a union template, the matching info is not properly cleaned up, and can cause future logged matchings of the same type to also display leftover data.

- Fixed.
- Regression test case added to regression_test/templateUnion

BUG 566908 - CHANGE DISPLAY CATEGORY OF RND() TO DEBUG

- The execution of rnd() is currently logged for other categories than DEBUG;
- when called repeatedly, the log is filled with irrelevant log entries as below:

- ......Function rnd() returned 0.202676.

- The printout category of rnd() execution should be changed to DEBUG to avoid this.

- Change implemented. The messages produced by 'rnd()' are now in the 'DEBUG_UNQUALIFIED' subcategory.

- Note: the logging subcategory 'FUNCTION_RND' is now unused.

BUG 566562 - IMPLEMENT NEGATIVE TESTING FOR THE RAW ATTRIBUTE 'LENGTHTO'

- Currently the RAW variant attribute 'LENGTHTO' (or any other variant) does not work if any of the fields mentioned in the variant have erroneous attributes.
- If the erroneous attributes change or omit the value of the field, then a coding error is displayed.
- If the erroneous attributes add new data before or after the field, then this data is not considered when calculating the length.

- How it should work:
- The length should be calculated based on the final value of the mentioned fields (after they have been altered by erroneous attributes).
- Fields omitted by erroneous attributes are ignored when calculating length (i.e. their length is zero).
- New data added before or after any of the mentioned fields by erroneous attributes should also be added to the length calculation.

- Implemented.
- Tests added to regression_test/negativeTest (and old tests updated).

BUG 568518 - RAW DECODER: SET WITH RECORD/SET OF FIELD AND REPEATABLE(YES)

- The RAW decoder incorrectly decodes the set with a record of type field if:
- - The record of field has a variant: REPEATABLE(yes)
- - It is a "catch all unknonw IE" list

- The problem is that the "catch all unknonw IE" list field can consume all Ies
- Solution:
- If the record/set of field of the set has variant "REPEATABLE(yes)" decode 1 element of the field at a time.
- tested

BUG 568755 - OOP-FIELDS CANNOT BE DECLARED PUBLIC

- Change implemented: public fields are no longer allowed in classes.

BUG 565875 - TITAN PRODUCT CODE CHANGE

- As the Titan product code has changed from CRL 113 200 (signifying an internal product) to CAX 105 7730 (meaning a 3rd party product):
- -product references in the documentation have to be changed accordingly
- -product references in the code, e.g. the product number in the product manifest, displayed with compiler -v:
- have to be changed accordingly
- -document numbers, having internal significance only, will have to be erased from the Titan documentation

BUG 566029 - DTE IN STRING2TTCN

- The string2ttcn can't parse the string produced by the ttcn2string with float values:

```
var float fl:=1.99887E-9
var charstring ch

log(fl)
ch:=ttcn2string(fl)
log(ch)
string2ttcn(ch,fl)
log(fl)
```

- result:

```
10:14:02.550964 proba.ttcn:64 1.998870e-09
10:14:02.551000 proba.ttcn:67 "1.998870e-09"
10:14:02.551161 proba.ttcn:68 Dynamic test case error: Parse error in line 1, at or before token `9': syntax error, unexpected integer value, expecting $end
10:14:02.551203 proba.ttcn:68 Performing error recovery.
```

- Fixed

BUG 566028 - WARNING: LEADING ZERO DIGIT WAS DETECTED IN THE ARGUMENT OF FUNCTION STR2FLOAT()

- The str2float function produces unnecessary leading zero warning. The TTCN standard explicitly permits the leading zero for the argument of the str2float:
- "C.1.29 Character string to float
- str2float(in charstring invalue) return floatThis function converts a charstringcomprising a number into a float value. The format of the number in the charstring shall follow rules in clause 6.1.0, items a) or b) with the following exceptions:
- •leading zeros are allowed; "
- Moreover the warning is issue if the result of the float2str is passed to the str2float:
-     var float fl:=1.99887E-9
-     log(fl)
-     var charstring ch:=float2str(fl)
-     log(ch)
-     fl:=str2float(ch)
-     log(fl)
- result:
- 09:41:55.316406 proba.ttcn:59 1.998870e-09
- 09:41:55.316530 proba.ttcn:62 "1.998870e-09"
- 09:41:55.316582 proba.ttcn:63 Warning: Leading zero digit was detected in the argument of function str2float(): "1.998870e-09".
- 09:41:55.316627 proba.ttcn:64 1.998870e-09

BUGZILLA JUNE-JULY

Component	Assignee	Status	Resolution	Summary	Changed
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	incorrect error message on variant specifications.	6/25/2020 11:28
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	holding on the consoles for too long might create deadlocks	6/26/2020 7:56
Plug-ins	arpad.lovassy@semcon.com	CLOSED	FIXED	Locale dependent decimal point in JSON encoded float	7/3/2020 4:58
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Adapt Titan compiler version checking to new gcc/clang versioning scheme	7/8/2020 6:41
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	JSON encoding/decoding fails some of the tests on my machine, encounters dynamic testcase error in other	7/8/2020 7:03
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	config editor throws exception when opening historical files	7/8/2020 7:04

BUGZILLA AUG-SEPT

Component	Assignee	Status	Resolution	Summary	Changed
Plug-ins	balaskojenoattila@gmail.com	CLOSED	FIXED	Semantic analysis error for permutation-all_from-array	8/4/2020 4:44
Core	titan-inbox@eclipse.org	CLOSED	FIXED	False compiler error for 'activate' parameter	8/10/2020 9:30
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Support for @default in the compiler	8/12/2020 11:26
Plug-ins	titan-inbox@eclipse.org	CLOSED	INVALID	Generated java file contains error whenever an ttcn 3 module added with port definition	8/28/2020 3:12
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Appending '00'B to a bit-string produces unexpected '...11'B result	9/2/2020 0:47
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	unable to connect executable from both Java and C side to the same main controller	9/2/2020 8:05
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Document non-optional semicolon in 'alt' statements	9/15/2020 9:56
Core	balaskojenoattila@gmail.com	CLOSED	FIXED	The item with_pwtc is missing from the main table and the reference for it in 'with.html' is wrong	9/23/2020 9:22

BUGZILLA OCT-NOV

Component	Assignee	Status	Resolution	Summary	Changed
Core	balaskojenoattila@gmail.com	CLOSED	FIXED	Modify the install guide according to the behavior of the new AFS service on E2C	10/7/2020 5:01
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Titan as a microservice	10/21/2020 8:08
Plug-ins	balaskojenoattila@gmail.com	CLOSED	FIXED	IndexOutOfBoundsException in map and unmap handling in tc_single	10/27/2020 7:20
Plug-ins	balaskojenoattila@gmail.com	CLOSED	FIXED	False positive test in semantic error "Cannot determine system component" in map and unmap st.	10/27/2020 7:21
Plug-ins	balaskojenoattila@gmail.com	CLOSED	FIXED	false positive semantic error in unmap parameter evaluation: "Too few parameters"	10/27/2020 7:22
Core	titan-inbox@eclipse.org	CLOSED	FIXED	timestamp problem after 20138 for testcase start times	11/13/2020 9:22
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Under special circumstances, compilation gets stuck at "Checking modules..."	11/17/2020 11:01
Core	botond.baranyi@ericsson.com	CLOSED	FIXED	Object-oriented features (continuation)	11/17/2020 11:10
Core	titan-inbox@eclipse.org	CLOSED	FIXED	c++17 compatibility	11/17/2020 11:11
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Issues with regular expressions	11/17/2020 11:11
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Incompatibility of charstring and universal charstring	11/17/2020 11:12
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Unexpected behavior of port.halt	11/17/2020 11:13
Core	titan-inbox@eclipse.org	CLOSED	FIXED	map param compiler error, function with return value	11/17/2020 11:13
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Investigate if libedit can be shipped in other form than embedded source	11/17/2020 11:13

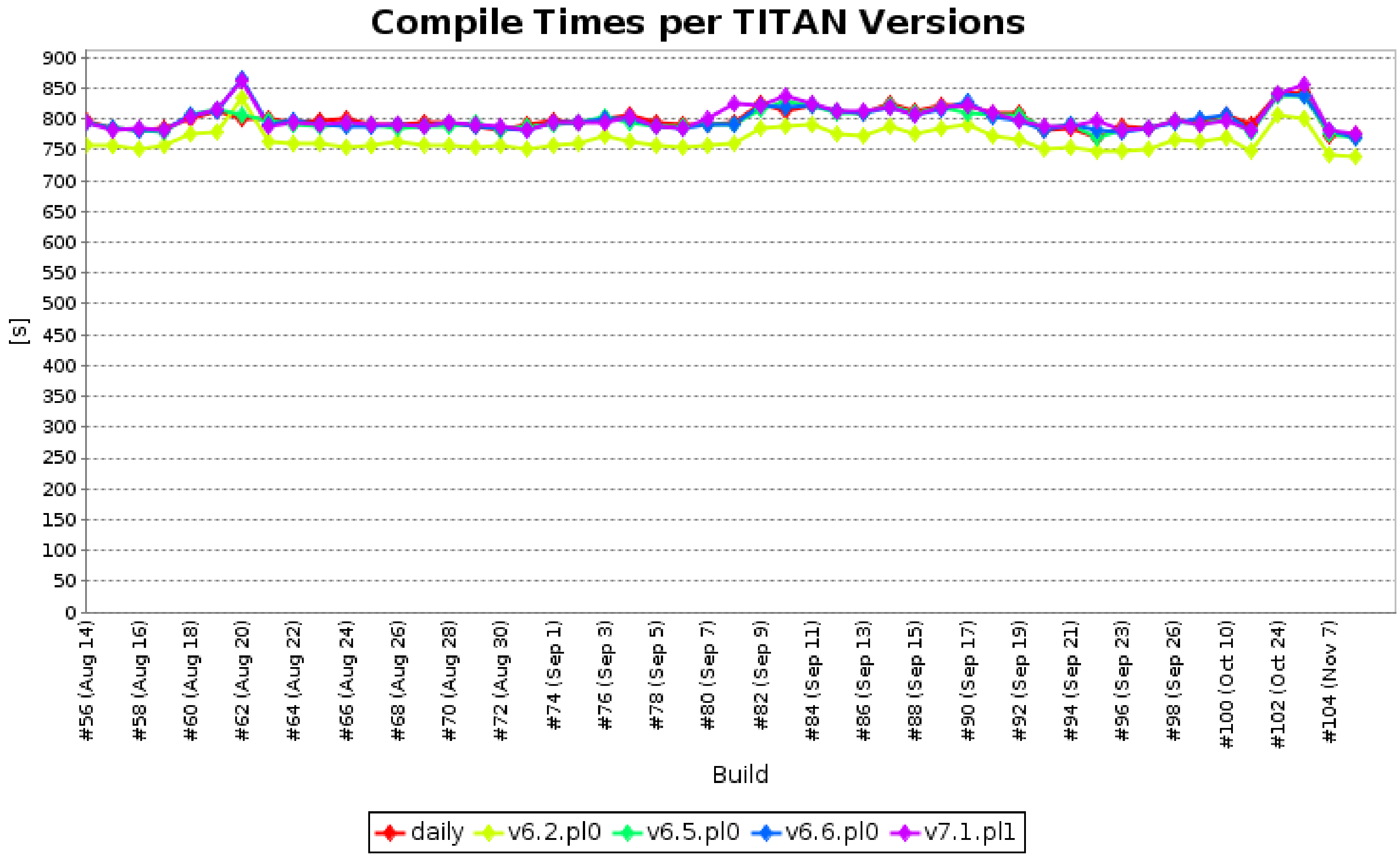
BUGZILLA NOV

Component	Assignee	Status	Resolution	Summary	Changed
Core	titan-inbox@eclipse.org	CLOSED	FIXED	JSON decoder bug/inconsistency	11/17/2020 11:14
Core	titan-inbox@eclipse.org	CLOSED	FIXED	charstring pattern restriction and empty string	11/17/2020 11:15
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Logging error when matching with a union template is unsuccessful	11/17/2020 11:15
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Change display category of rnd() to DEBUG	11/17/2020 11:15
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Implement negative testing for the RAW attribute 'LENGTHTO'	11/17/2020 11:16
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Support for @default in xsd2ttcn as per ETSI ES 201 873-9 V4.11.1 (2020-05)	11/17/2020 11:16
Core	gabor.szalai@ericsson.com	CLOSED	FIXED	RAW decoder: Set with record/set of field and REPEATABLE(yes)	11/17/2020 11:18
Core	titan-inbox@eclipse.org	CLOSED	FIXED	OOP-fields cannot be declared public	11/17/2020 11:21
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Titan product code change	11/18/2020 3:04
Core	balaskojenoattila@gmail.com	CLOSED	FIXED	Documentation on "isbound" differs from actual execution	11/18/2020 4:05
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	update json default attribute behavior to the one agreed on by the standardization committee	11/18/2020 4:07
Plug-ins	titan-inbox@eclipse.org	CLOSED	FIXED	CfgParseTreePrinter results empty string	11/18/2020 4:08
Core	titan-inbox@eclipse.org	CLOSED	FIXED	DTE in string2ttcn	11/18/2020 9:42
Core	titan-inbox@eclipse.org	CLOSED	FIXED	Warning: Leading zero digit was detected in the argument of function str2float()	11/18/2020 9:42
Core	balaskojenoattila@gmail.com	CLOSED	FIXED	Document usage of OLD_LIBEDIT	11/18/2020 10:29

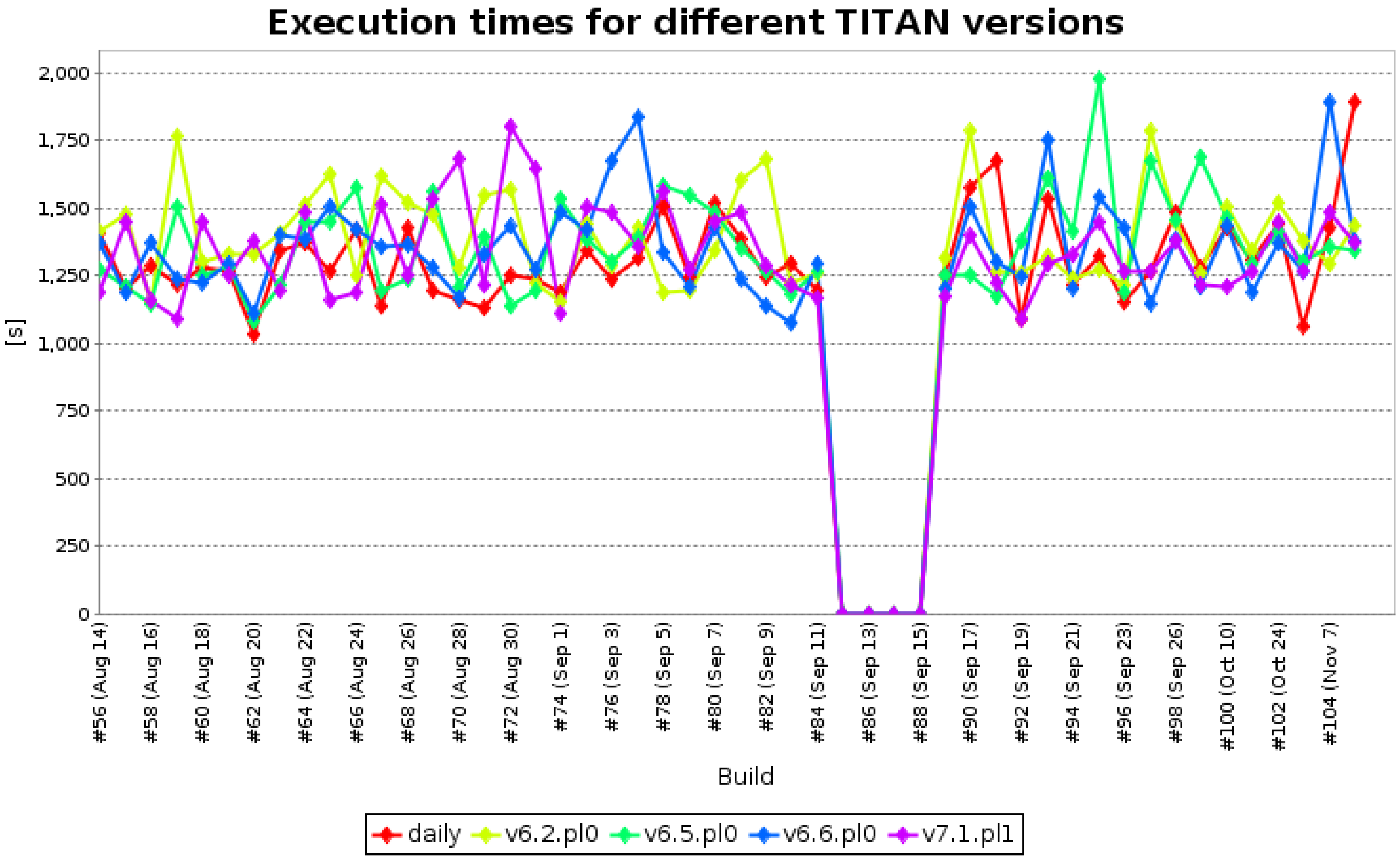
DOCUMENTATION

Title	Version
Release Notes for TITAN TTCN-3 Test Executor	7.2.0
Installation guide for TITAN TTCN-3 Test Executor	7.2.0
Installation Guide for TITAN Designer and TITAN Executor for the Eclipse IDE	7.2.0
User Guide for TITAN 7.2.0	7.2.0
Programmers Technical Reference for TITAN TTCN-3 Test Executor	7.2.0
User Guide for the TITAN Designer for the Eclipse IDE	7.2.0
User Guide for the TITAN Executor for the Eclipse IDE	7.2.0
API Technical Reference for TITAN TTCN-3 Test Executor	7.2.0
Titan Executor API user guide	7.2.0
Programmers Tech. Reference Guide for Titanium	7.2.0
Programmers Tech. Reference Guide for the Java Codegenerator	7.2.0
Titanium Refactoring Description	7.2.0
Titanium Description	7.2.0
Statement of Compliance for Eclipse Titan	7.2.0
Statement of Compliance for use of XML schema in Eclipse Titan	7.2.0

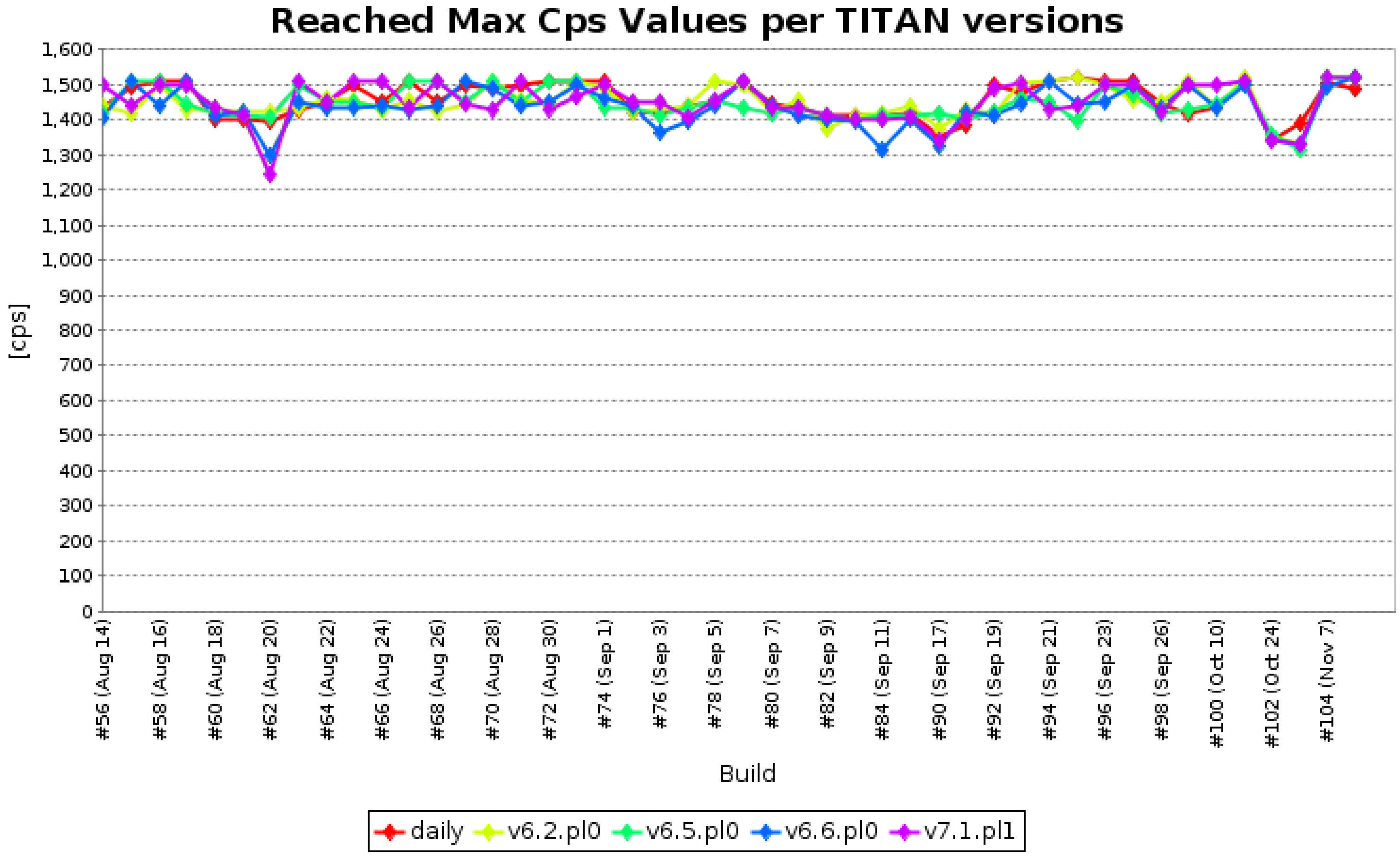
PERF. TEST COMPILE TIMES



PERF.TEST EXECUTION TIMES



PERF. TEST MAX CPS VALUES



ERICSSON