

TITAN 6.5.1 CHANGE LOG

2019.01.22

BUG 539514 - REAL-TIME TESTING IN TITAN



- › Continued from 6.5.0:
- › Added default values to generated port class member functions 'send', 'call', 'reply' and 'raise' for backward compatibility (so they can be called from external functions without the timestamp redirect parameter).
- › Regression test added, too.

BUG 538482 - TYPE DESCRIPTOR GENERATION PROBLEMS CAUSED BY OER FIX



- › A fix to OER problem
https://bugs.eclipse.org/bugs/show_bug.cgi?id=537888
(which has now been removed) caused type descriptors for certain ASN.1 types to be generated incorrectly, when using XER in ASN.1 (compiler option '-a').
- › The error occurred in test
regression_test/compileonly/openTypeXER.
- › The OER fix has been re-added, and the type descriptor fault has been fixed.

BUG 541748 - XSD2TTCN: NAME CLASHES AND REFERENCES NOT HANDLED WHEN USING SINGLE MODULE



- › The following problems occur when generating a single module, using the command line option '-o':
 - › - Name clashes: two types with the same name in different XSDs cause name clash errors in the generated TTCN-3 file (normally these would be generated into different modules and not cause errors).
 - › Solution: suffix the types (e.g. with _1, _2, etc.) and give them 'name as' attributes.
 - › - References to other modules: references to values from other namespaces (e.g. default values) are still prefixed with their module names.
 - › Solution: remove the prefixes, as they are in the same module.
- › For examples see attached XSD files.
- › Solutions implemented.
- › Regression test added under regression_test/XML/XmlWorkflow
- › Tests have passed.
- › Improved this fix to work for name clashes across 3 or more XML namespaces.
- › Regression test updated.

BUG 543155 - NEW ERROR BEHAVIOR OPTION FOR DATA REMAINING AFTER DECODING



- › Currently, if there is data left in the buffer after decoding succeeds, the decoder function always displays a warning about it.
- › A new option should be added in the 'errorbehavior' extension of decoder functions to change this warning to error or to ignore it.
- › Implemented error behavior 'EXTRA_DATA', which governs the mentioned warning
Possible backward incompatibility: setting 'errorbehavior' to 'ALL:ERROR' now also set this to error, which would cause code that previously only displayed a warning to now display a DTE.

BUG 542610 - TRANSLATION PORTS ACROSS MULTIPLE ECLIPSE PROJECTS



- › The generated code for provider ports (i.e. ports in the 'map to' clauses of translation ports) changes, depending on how many translation ports are mapped to them.
- › Because of this the provider port's module and all modules containing translation ports that are mapped to the provider port must be in the same compiler command. If the provider port is compiled first on its own (i.e. without the modules that contain translation ports it is mapped to), and the compiled C++ code is used in a different project, where the translation ports are (such as with central storage, and with project references in the Eclipse plug-ins), then the generated C++ class for the provider port won't contain all members and methods necessary for the port translation (causing C++ compilation errors in the second project).
- › This limitation has been documented in the reference guide.
- › A possible solution for this issue should be investigated.
- › Created attachment 276873 [details]
- › Example

BUG 542610 - TRANSLATION PORTS ACROSS MULTIPLE ECLIPSE PROJECTS



- › Proposed solution:

- › The classes generated for all ports with the 'provider' attribute would contain all necessary code to handle port translation (with any translation port).
- › This would include the following changes:
 - › - a port pointer array (PORT**) and its size as new members for the provider port's class, which would be used in the same way as the current port-type-specific pointer arrays (p_0, p_1, etc. and n_0, n_1, etc.);
 - › - the member functions 'add_port' and 'remove_port' would be generated for all provider port classes and would add or remove ports from the new pointer array (without dynamic casting, since the actual port types are unknown);
 - › - the member functions 'reset_port_variables' and the 'outgoing_public_send/call/reply/raise' functions for all message/signature types would also be generated (their contents would be unchanged from the ones generated for current provider ports);
 - › - the member functions 'incoming_message/call/reply/exception' would contain the 'for' cycle that goes through the mapped translation ports (from current provider port code), except now it would go through all ports, and call a new function in the port base class that calls the translation port's 'incoming_message/call/reply/exception' member function (since these are not part of the base port class).

- › The base port class (PORT) would have the following new function:
 - › virtual boolean incoming_message_handler(const void* message_ptr, const char* message_type, component sender_component, const FLOAT& timestamp);

 - › This would be implemented by all translation ports, and it would call the appropriate 'incoming_message' function for the type indicated by the parameter 'message_type'. There would either be four of these functions in total (3 more for calls, replies and exceptions), or it would have an extra parameter that indicates the operation type.

- › Performance:
 - › - in case of translation ports there would be minor performance losses (extra functions being called, more generated code, etc.);
 - › - if the provider port is used in normal mode, or in a dual-faced port, then there would be a considerable amount of extra generated code, and minor performance losses during runtime.

BUG 542610 - TRANSLATION PORTS ACROSS MULTIPLE ECLIPSE PROJECTS



- › Created attachment 276938 [details]
- › Modified C++ files with the proposed solution

- › These files contain the extra code that would be generated into A.cc, A.hh, B.cc and B.hh in the previously attached example (modifications are marked with lots of '/'-es).

- › This still needs the addition of the `incoming_message_handler` function in `PORT.hh` and its base implementation (which just returns `FALSE`) in `PORT.cc` in the runtime library.
- › Proposed solution implemented.

- › Regression tests added under `regression_test/portTranslationCentralStorage`.

BUG 539612 - UBUNTU 18.04, GCC 7, C++11 COMPATIBILITY



- › The ttcn3float.hh is not compatible with the c++11, gcc 7 used by Ubuntu 18.04
- › In c++11 the signbit() is not a macro any more, so the
- › `#ifndef signbit` is true and the ttcn3float.hh redeclares the signbit as a macro.
- › It doesn't cause any problem until something other than struct ttcn3float tries to use the signbit().
- › For example an `#include <complex>` after the `#include <TTCN3.hh>`
- › Either protect the signbit checking with
- › `#if __cplusplus < 201103L // c++11 -> __cplusplus == 201103L`
- › `#ifndef signbit`
- ›
- › `#endif // def signbit`
- › `#endif // __cplusplus < 201103L`
- › Or remove it.
- › Implemented the proposed extra C++ version check.

BUG 517843 - SUPPORT FOR MULTIPLE ENCODINGS



- › Continued
- › Updated the syntax analysis of '<port>.setencode' to allow subreferences (i.e. port array indexes). This form of 'setencode' is still not supported, but the compiler now displays a more appropriate error message for it.



Bug ID	Product	Component	Assignee	Status	Resolution	Summary	Changed
538482	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Type descriptor generation problems caused by OER fix	12/12/2018 5:57
541748	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	xsd2ttcn: Name clashes and references not handled when using single module	12/14/2018 5:45
543155	Titan	Core	titan-inbox@eclipse.org	RESOLVED	FIXED	New error behavior option for data remaining after decoding	1/4/2019 9:34
542610	Titan	Core	titan-inbox@eclipse.org	RESOLVED	FIXED	Translation ports across multiple Eclipse projects	1/14/2019 11:50
539612	Titan	Core	titan-inbox@eclipse.org	RESOLVED	FIXED	Ubuntu 18.04, gcc 7, c++11 compatibility	1/17/2019 5:59
517843	Titan	Core	titan-inbox@eclipse.org	CLOSED	FIXED	Support for multiple encodings	1/21/2019 8:36



ERICSSON