TITAN 6.6.1
CHANGE LOG

2019.11.29
This release is mainly a corrective release, it introduces no major new features. No backward incompatibilities are to be expected, with maybe 2 exceptions we see as a minor risk:

- The fix for 546326: Element "freeTextLinkerOptions" in TPD.xsd is not handled by makefilegen may induce under some unlikely circumstances an incompatible behavior during building with TPD -s.

- The fix for 553542: GCC aborts with "malloc.c:3546: munmap_chunk: Assertion `ret == 0' failed." shows that to compile very large modules users might need to use gcc versions above 4.3, or the –U option to split their code in several parts.
There are several new features in progress, that will enhance the user experience, but can not be made available for users in this release.

- 552011 Object-oriented features
- 553219 add support for JSON to the Java code generator
- 553220 Main Controller in pure Java
NEW FEATURES

› 547552  Enhance performance of TIMER::get_min_expiration
› 547876  Legitimize compiler option '-F'
› 547318  Colorize compiler error/warning messages
› 548963  change the internal representation of octetstring from char[] to byte[] for efficiency
› 548969  CSN.1 L/H in the RAW codec
› 550942  change in how xsd files with NoTargetNamespace should be mapped to TTCN-3
› 553217  add support for structured type compatibility to the Java code generator
› 553245  add support for module parameter handling to the Java side runtime
› 553271  call hierarchy in the designer
The features listed here are being developed.
Yet they are not yet complete enough to be made generally available.

› 552011 Object-oriented features
› 553219 add support for JSON to the Java code generator
› 553220 Main Controller in pure Java
BUGFIXES

- 547385 RAW: attribute 'PRESENCE' on embedded record causes compilation error
- 546800 Define function type which parameters with timer would get compiler error
- 546326 Element "freeTextLinkerOptions" in TPD.xsd is not handled by makefilegen
- 549218 Code generation fault for record of anytype with compiler option '-X'
- 549252 TITAN BER decoder fails to decode S1AP in BER
- 550526 Fatal error during semantic analysis of recursive ASN.1 type
- 550528 Incorrect float2str result
- 550623 issue with new way encoding
- 551631 Config file parsing error for float module parameter expressions
- 551516 Unnesseray warning: "Data remained at the end of the stream after successful decoding: '00'O"
- 551780 Segmentation fault caused by universal charstring 'replace'
- 552232 mandatory to specify optional fields in default values of record in function parameter
- 552116 Java 11 and the newer versions no longer contains the javax.xml.bind package.
- 552492 Invalid code generated for integer literal -2147483648
- 552529 Code generation fault when using optional field in value redirect
- 552983 Warnings when building time
- 553542 GCC aborts with "malloc.c:3546: munmap_chunk: Assertion \"ret == 0\" failed."
NEW FEATURE DETAILS
A minor change done to this function on the Java side was able to improve the performance.

Implemented on the C++ side, too.
Compiler option '-F' forces the generation of record of/set of types with basic element types, and disables their type compatibility. This is used for pre-generating the C++ classes for these types into the runtime library.

The option should be made public, since it also appears in the Designer plug-in.

Documented the option in the reference guide, compiler usage output and compiler manual.
Suggestion and initial implementation by Harald Welte.
See https://review.gerrithub.io/#/c/laf0rge/titan.core/+/454000/

Pushed the mentioned commit and a change that restricts the type of texts that are colorized.
Changing the internal representation of Octetstrings from char[] to byte[] would improve networking efficiency, as the current representation requires a slow copy before sending and after receiving data.

In a ping-pong scenario this change was able to increase the number of handled messages by 5% - 8%.

Closed fixed
Implemented new variant attribute "CSN.1 L/H" for the RAW codec.

From the reference guide:
- Attribute syntax: `CSN.1 L/H`
- Default value: unset
- Can be used with: all basic types, `records`/`sets`/`unions` (in which case the attribute is set for all fields of the `record`/`set`/`union`)
- Description: If set, the bits in the bitfield are treated as the relative values `L` and `H` from `CSN.1` instead of their absolute values (`0` is treated as `L` and `1` is treated as `H`). These values are encoded in terms of the default padding pattern `2B'O ('00101011'B), depending on their position in the bitstream.

Practically the bits in the bitfield are XOR-ed with the pattern `2B'O before being inserted into the stream.

Closed.
The current running STF had to introduce a change in how XSD modules with NoTargetNamespace should be mapped to TTCN-3: http://oldforge.etsi.org/mantis/view.php?id=7848

The original behaviour needs to be supported as it is already used in several standards and by several users.

A new flag should be added to the XSD converter, to generate the files according to this new method.
Please note, that this will not only the names of the generated files, but also the number of generated files (which might have other effects too)

Closed.
Implemented command line option `-N` for xsd2ttcn, which generates separate modules for each XSD with no target namespace.
Extend the Java Code generator to offer support for structured type compatibility, similar to that of the C side
The Java side of the Titan toolset is not yet able to process module parameters.
Adding this will enable user to use the same configuration files to configure the runtime execution of their tests.

implemented
It would be nice if the call hierarchy view would also function for TTCN-3 files.

Partly implemented.
Now it is supported to populate the call hierarchy view with the calling sites of a function.

Please note, that there is limited functionality in the sense, that the control part is not listed if it is the calling site.
DETAILS OF FEATURES IN PROGRESS

› The features listed here are being developed.
› Yet they are not yet complete enough to be made generally available.
Implement object-oriented features according to chapter 5 of the standard extension:
https://www.etsi.org/deliver/etsi_es/203700_203799/203790/01.01.01_60/es_203790v010101p.pdf

Work ongoing.
Current implementation can parse OO code and do some codegeneration.
But does not offer strong enough semantic checks and has runtime issues … for which can not be supported as release feature yet.
Add support for JSON encoding/decoding to the Java code generator, similarly to what is already present on the C side.

Work ongoing
- stage 1 ongoing
- The JSON related variant attributes are parsed correctly, and semantic checking also works.

- Work on the code generation and runtime structures is beginning.
- The encoding/decoding will not be available during execution in this release for the users.
Create a version of the Main Controller that works purely in Java/Eclipse.

This has several benefits:
- platform independence allows users to work on Windows.
- (also reduces the cost of supporting several platforms)
- Can simplify the code of launch configurations used in eclipse as there is no need to communicate with external tools via command line.
- Can reduce the number and complexity of launch configurations for users as Java native MC can be controlled without command line interface.
- Much better GUI is possible, as the Java native MC can have better connection interface.
- Enhanced information extraction, as the Java native MC could provide the user with additional dynamic information previously unavailable.
  for example: visualizing PTC connection as the tests are running.

Work ongoing
- Batch mode already works in a Java native way.
- This proves the concept on a functional level, but is not yet good enough solution to be made available/supported in its current state, as a product.
- The Java native Main Controller will not be available in this release for the users.
module test {
    type record R1{
        integer f1,
        record {integer v2} f2 optional
    } with {
        variant (f2) "PRESENCE(f1 = 0)"
    }
} with { encode "RAW"}

Results in: test.ttcn:4.3-7.3: In type definition `R1':
test.ttcn:6.5-23: error: Invalid fieldname in RAW parameter PRESENCE for the record @test.R1.f2: f1

Fixed.

'PRESENCE' attributes to embedded types in records are now only interpreted as an attribute for the field in that record (i.e. as the first case in the above list), and not as an attribute for the embedded type itself.
I defined a function type with a timer parameters, it would failed when I compiled the TTCN file contains it.

// TTCN file 'test.ttcn'
module test {
  type function test_func(timer T); // remove this line could pass compile
  function f_test(timer T) {
    log("just for test.");
  }
}

Fixed.
The compiler should no longer crash when it encounters a timer parameter in a function type.
The element "freeTextLinkerOptions" of TPD.xsd is not handled by makefilegen.

The built-in makefile generator of titan.EclipsePlug-ins handles this element and generates the Makefile.

This setting

<freeTextLinkerOptions>bar</freeTextLinkerOptions>

results this in the Makefile:

# Flags for the linker:
LDFLAGS = bar

instead of the default

# Flags for the linker:
LDFLAGS =

Fixed.
module test {
  type record of anytype Anytypes;
}

Compiled with option '-X' causes the following C++ error in the generated code:

```
  test.cc:22:104: error: 'Anytypes_0_descr_' was not declared in this scope
  const TTCN_Typedescriptor_t Anytypes_descr_ = { "@test.Anytypes", NULL, NULL, NULL, NULL, NULL, NULL, &Anytypes_0_descr_, TTCN_Typedescriptor_t::DONTCARE };  
  const TTCN_Typedescriptor_t Anytypes_descr_ = { "@test.Anytypes", NULL, NULL, NULL, NULL, NULL, NULL, &Anytypes_0_descr_, TTCN_Typedescriptor_t::DONTCARE };  
```

Fixed.
BUG 549252 - TITAN BER DECODER FAILS TO DECODE S1AP IN BER

- Let's assume I have the following BER-encoded S1AP PDU:

- …

- TITANs built-in BER decoder will fail to decode it:

  TC_s1ap_attach-S1AP0(3)@nataraja: Dynamic test case error: While BER-decoding type '@S1AP-PDU-Descriptions.S1AP-PDU': While decoding '@S1AP-PDU-Descriptions.S1AP-PDU' type: Alternative 'initiatingMessage': While decoding '@S1AP-PDU-Descriptions.InitiatingMessage' type: While decoding open types: Component 'value_': While decoding open type '@S1AP-PDU-Descriptions.InitiatingMessage.value.type': Alternative 'initialContextSetupRequest': While decoding '@S1AP-PDU-Contents.InitialContextSetupRequest' type: Component 'protocolIEs': While decoding '@S1AP-PDU-Contents.InitialContextSetupRequest.protocolIEs' type: Component #3: While decoding '@S1AP-Containers.ProtocolIE-Field.S1AP-PDU-Contents.inst26' type: While decoding open types: Component 'value_': While decoding open type '@S1AP-Containers.ProtocolIE-Field.S1AP-PDU-Contents.inst26.value.type': Alternative 'e-RABToBeSetupListCtxtSUReq': While decoding '@S1AP-PDU-Contents.E-RABToBeSetupListCtxtSUReq' type: Component #0: While decoding '@S1AP-PDU-Contents.E-RABToBeSetupItemCtxtSUReq' type: Component 'e_RABlevelQoSParameters': While decoding '@S1AP-IEs.E-RABLevelQoSParameters' type: Invalid 'constructed' flag (must be set).

- Closed.

- The difference between them (or the part that is relevant to this decoding) is in S1AP_PDU_Contents.asn.

- In the version you gave me, type 'E-RABToBeSetupListCtxtSUReq' is merely a 'record of' with 'E-RABToBeSetupItemCtxtSUReq' as its element type.

  E-RABToBeSetupListCtxtSUReq ::= SEQUENCE (SIZE(1..maxnoofE-RABs)) OF E-RABToBeSetupItemCtxtSUReq

- In the version I found, its element type is a parameterized type (open type?) with 'E-RABToBeSetupItemCtxtSUReq' as its parameter.

  E-RABToBeSetupListCtxtSUReq ::= SEQUENCE (SIZE(1..maxNrOfE-RABs)) OF ProtocolIE-SingleContainer { {E-RABToBeSetupItemCtxtSUReqIEs} }
BUG 550526 - FATAL ERROR DURING SEMANTIC ANALYSIS OF RECURSIVE ASN.1 TYPE

Test
DEFINITIONS AUTOMATIC TAGS ::= BEGIN
MultiplexElement ::= SEQUENCE {
typefield CHOICE {
    logicalChannelNumber INTEGER(0..65535),
    subElementList SEQUENCE SIZE (2..255) OF MultiplexElement
},
repeatCount CHOICE {
    finite INTEGER (1..65535), -- repeats of type
    untilClosingFlag NULL -- used for last element
}
}
END

Result:
FATAL ERROR: compiler: In line 440 of Setting.cc: Setting::get_genname_own(): genname is not set in @Test.MultiplexElement.repeatCount

Fixed.
The float2str is local dependent, but according to the TTCN-3 standard explicitly defines the format to be used. See chapter 6.1.0 a) & b)

On Swedish local the str2float(float2str(0.0)) generates DTE, because the decimal separator is ,

... Fixed.

Float2str and the logging of floats are no longer affected by locale settings.
There is an issue with the new encode handling as shown in the below code.

This code compiles without errors to C/C++ code:

```
type record templateHexstr_rec {
  hexstring x1,
  hexstring x2,
  hexstring x3 optional
}
with {
  encode "RAW";
  // encode (x1) "RAW";
}
```

```
template templateHexstr_rec templateHexstr_tDecmatchSelfRef := {
  // decoded content match with self-reference
  x1 := '01A'H,
  x2 := decmatch templateHexstr_tDecmatchSelfRef.x1,
  x3 := decmatch templateHexstr_rec: { x1 := templateHexstr_tDecmatchSelfRef.x1, x2 := ?, x3 := * }
};
```

but that code is incorrect:

```
if (_decoder(os, *dec_val, templateHexstr__rec_x1_default_coding) != 0) {
  
  Fixed.
  The coder functions for base types, that inherit 'encode' attributes from a higher scope (e.g. the parent record), weren't generated, and their usages were generated incorrectly.
```
The config file parser incorrectly interprets the 2nd part of a float expression as another float value (in the module parameters section), if there are no spaces in the expression.

Example:
TTCN-3:
module test2 {
  type component CT{}
  modulepar float mp_f;
  control {
    action(mp_f);
  }
}

Config file:
[MODULE_PARAMETERS]
mp_f := 1.0-2.1

In this case '-2.1' is treated as a float value, instead of part of an expression, which causes an error. (The parser thinks that 'mp_f := 1.0' is the first parameter assignment, and '-2.1' is the beginning of the 2nd.)

Fixed.
Unnecessary warning issued by the built-in text decoder:

Warning: f_SDPAM_Dec(): Data remained at the end of the stream after successful decoding: '00'O

module proba {
  external function f_SDPAM_Enc(in SDPAM_Packet pdu) return charstring with { extension "prototype(convert) encode(TEXT)" };  
  external function f_SDPAM_Dec(in charstring msg) return SDPAM_Packet with { extension "prototype(convert) decode(TEXT)" };  
  type record SDPAM_Packet {
    charstring f1
  } with {
    variant "BEGIN('-')"
    variant "END('-')"
  }
  Control {
    var SDPAM_Packet s:="salala"
    var charstring c:=f_SDPAM_Enc(s)
    log(c)
    s:=f_SDPAM_Dec(c)
    log(s)
  }
} with {
  encode "TEXT"
}

Fixed.
The following code causes a segmentation fault:

```java
module UcharBug {
    type component CT{}
    // remove 2 chars from the beginning, insert 1 char runtime
testcase tc_predef_replace_ucharstring_4() runs on CT {
    var universal charstring vl_ucs0 := "0000 1111"
    var integer idx := 0, len := 2;
    var universal charstring vl_ucs1 := "1";
    var universal charstring vl_ucs := replace(vl_ucs0, idx, len, vl_ucs1);
    if(vl_ucs != "100 1111"){
        setverdict(fail, "expected \"100 1111\" got:\", vl_ucs);
    } else {
        setverdict(pass);
    }
}    
control {
    execute(tc_predef_replace_ucharstring_4());
}
}

Fixed.
```

**BUG 551780 - SEGMENTATION FAULT CAUSED BY UNIVERSAL CHARSTRING 'REPLACE'**
Hi,

Consider the following scenario:

... 

Although the definition in lines 21-24 is allowed (vl_m.a becomes 2 and vl_m.b becomes unbound), the definition in lines 9-12 does not work if it is changed to:

```plaintext
9        in myRec  pl_rec :=
10       {
11              2
12       }
```

Is this an expected behavior? It would make more sense if these two definitions worked in the same way.

Fixed.

According to the TTCN-3 standard, function parameters (or their default values) are allowed to be incomplete (i.e. they shouldn't cause errors if they are not fully initialized).

I've implemented a fix that changes these errors into warnings.
The Java 11 and newer versions no longer contains the javax.xml.bind package, but the org.eclipse.titanium.sonar.metrics and the org.eclipse.titanium.utils packages use it. 

The project cannot be compiled with JDK 11 or newer versions.

Fixed in commit [https://github.com/eclipse/titan.EclipsePlug-ins/commit/49a16c3c0d013dd2cfe8d3b83dfe2685bcebd0](https://github.com/eclipse/titan.EclipsePlug-ins/commit/49a16c3c0d013dd2cfe8d3b83dfe2685bcebd0)
The code generated for integer literal -2147483648 in certain operations does not compile on some systems.

Example:
(from regression_test/intOper/TintOper.ttcn)

var integer x2 := -2147483648;
if ( x2 == -2147483648 ) {setverdict(pass,x2)} else {setverdict(fail, x2)}

The generated C++ line (x2 == -2147483648) is ambiguous for some compilers.

Fixed.
type record myrec1 {
  integer f1 optional,
  integer f2
} with { variant "" encode "RAW" }
...
var myrec1 rec1 := { 1, 2 }
...
[] Q.receive(il4[0]) -> value rec1.f1 //<<<

Runtime2 only.

Fixed.
When building IMTAF with TCC, we saw lot of warnings (I listed in attach file) while building time. Could you check if the warning caused by Titan compiler? If it is so, could you to reduce or remove all of

Closed.

These merely warn you that some parts of the code might not work as you expect during runtime.
You can disable all warnings with the compiler option '-w' if they bother you.
on SLED 11.1 with gcc 4.3 the compilation fails with the following output:
...  
cc1plus: malloc.c:3546: munmap_chunk: Assertion `ret == 0' failed.
DIAMETER_Types.cc: In member function `void DIAMETER__Types::AVP__Data_template::copy_value(const DIAMETER__Types::AVP__Data&)':
DIAMETER_Types.cc:149299: internal compiler error: Aborted

Closed.

The problem could be reproduced. 
We believe the file to be compiled simply became too large for this gcc version (DIAMETER_Types.cc is ~30.4MB)

Compared to a previous Titan version the only additions include +1 parameter for the RAW descriptor and JSON descriptor, and 1 additional check during decoding (fixing a bug)

This is gcc internal bug, in an old gcc version. 
One step forward could be to upgrade the gcc version.
If that is not an option we can also propose as a workaround for this issue to use the -U option, instructing Titan to generate more and smaller files from each module.
<table>
<thead>
<tr>
<th>ID</th>
<th>Product</th>
<th>Comp</th>
<th>Assignee</th>
<th>Status</th>
<th>Resolution</th>
<th>Summary</th>
<th>Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>547552</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Enhance performance of TIMER::get_min_expiration</td>
<td>19/05/22</td>
</tr>
<tr>
<td>547876</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Legitimize compiler option '-F'</td>
<td>19/06/03</td>
</tr>
<tr>
<td>547318</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Colorize compiler error/warning messages</td>
<td>19/06/06</td>
</tr>
<tr>
<td>ID</td>
<td>Product</td>
<td>Comp</td>
<td>Assignee</td>
<td>Status</td>
<td>Resolution</td>
<td>Summary</td>
<td>Changed</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>---------------</td>
<td>----------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>548963</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>change the internal representation of octetstring from char[] to byte[] for efficiency</td>
<td>19/07/04</td>
</tr>
<tr>
<td>549218</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Code generation fault for record of anytype with compiler option '-X'</td>
<td>19/07/16</td>
</tr>
<tr>
<td>549252</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>TITAN BER decoder fails to decode S1AP in BER</td>
<td>19/08/01</td>
</tr>
<tr>
<td>548969</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>CSN.1 L/H in the RAW codec</td>
<td>19/08/12</td>
</tr>
<tr>
<td>550526</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Fatal error during semantic analysis of recursive ASN.1 type</td>
<td>19/08/29</td>
</tr>
<tr>
<td>ID</td>
<td>Product</td>
<td>Comp</td>
<td>Assignee</td>
<td>Status</td>
<td>Resolution</td>
<td>Summary</td>
<td>Changed</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>------------</td>
<td>-----------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>550942</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>RESOLVED</td>
<td>FIXED</td>
<td>change in how xsd files with NoTargetNamespace should be mapped to TTCN-3</td>
<td>19/09/18</td>
</tr>
<tr>
<td>550528</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>RESOLVED</td>
<td>FIXED</td>
<td>Incorrect float2str result</td>
<td>19/09/25</td>
</tr>
<tr>
<td>550623</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>RESOLVED</td>
<td>FIXED</td>
<td>issue with new way encoding</td>
<td>19/09/26</td>
</tr>
<tr>
<td>ID</td>
<td>Product</td>
<td>Comp</td>
<td>Assignee</td>
<td>Status</td>
<td>Resolution</td>
<td>Summary</td>
<td>Changed</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>---------------</td>
<td>--------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>551631</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Config file parsing error for float module parameter expressions</td>
<td>19/10/01</td>
</tr>
<tr>
<td>551516</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Unnesseray warning: &quot;Data remained at the end of the stream after successful decoding: '00'O&quot;</td>
<td>19/10/01</td>
</tr>
<tr>
<td>551780</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Segmentation fault caused by universal charstring 'replace'</td>
<td>19/10/08</td>
</tr>
<tr>
<td>552232</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>mandatory to specify optional fields in default values of record in function parameters</td>
<td>19/10/21</td>
</tr>
<tr>
<td>552116</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Java 11 and the newer versions no longer contains the javax.xml.bind package.</td>
<td>19/10/29</td>
</tr>
<tr>
<td>552492</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Invalid code generated for integer literal -2147483648</td>
<td>19/10/29</td>
</tr>
<tr>
<td>552529</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>Code generation fault when using optional field in value redirect</td>
<td>19/10/29</td>
</tr>
<tr>
<td>ID</td>
<td>Product</td>
<td>Comp</td>
<td>Assignee</td>
<td>Status</td>
<td>Resolution</td>
<td>Summary</td>
<td>Changed</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>------</td>
<td>--------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>552011</td>
<td>Titan</td>
<td>Core</td>
<td>botond.baranyi</td>
<td>ASSIGNED</td>
<td>---</td>
<td>Object-oriented features</td>
<td>19/11/05</td>
</tr>
<tr>
<td>552983</td>
<td>Titan</td>
<td>Other</td>
<td>titan-inbox</td>
<td>UNCONFIRMED</td>
<td>---</td>
<td>Warnings when building time</td>
<td>19/11/13</td>
</tr>
<tr>
<td>553219</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>NEW</td>
<td>---</td>
<td>add support for JSON to the Java code generator</td>
<td>19/11/19</td>
</tr>
<tr>
<td>553220</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>NEW</td>
<td>---</td>
<td>Main Controller in pure Java</td>
<td>19/11/19</td>
</tr>
<tr>
<td>553217</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>add support for structured type compatibility to the Java code generator</td>
<td>19/11/20</td>
</tr>
<tr>
<td>553245</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>add support for module parameter handling to the Java side runtime</td>
<td>19/11/20</td>
</tr>
<tr>
<td>553271</td>
<td>Titan</td>
<td>Plug-ins</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>call hierarchy in the designer</td>
<td>19/11/20</td>
</tr>
<tr>
<td>553542</td>
<td>Titan</td>
<td>Core</td>
<td>titan-inbox</td>
<td>CLOSED</td>
<td>FIXED</td>
<td>GCC aborts with &quot;malloc.c:3546:munmap_chunk: Assertion `ret == 0' failed.&quot;</td>
<td>19/11/27</td>
</tr>
<tr>
<td>Document Code</td>
<td>Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRL 113 200/6 R6B</td>
<td>Release Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>109 21</td>
<td>CRL 113 200/6 4Uen Rev.B PRI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/1531</td>
<td>CRL 113 200/6 Uen Rev.G Install. Guide for....Test Executor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1551</td>
<td>CRL 113 200/6 Uen Rev.G Titanium Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/174 02</td>
<td>CRL 113 200/6 Uen Rev.G Statement of Compliance for Eclipse Titan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2/174 02</td>
<td>CRL 113 200/6 Uen Rev.G Statement of Compliance for use of XML schema in Eclipse Titan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/198 17</td>
<td>CRL 113 200/6 Uen Rev.G UG for Titan TTCN-3 Test Executor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/198 17</td>
<td>CRL 113 200/6 Uen Rev.G UG for the TITAN Designer for the Eclipse IDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/198 17</td>
<td>CRL 113 200/6 Uen Rev.G UG of the TITAN Executor for the Eclipse IDE plug-in</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/198 17</td>
<td>CRL 113 200/6 Uen Rev.G API Technical Reference for the TITAN executor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/1551</td>
<td>CRL 113 200/6 Uen Rev. G Titanium Refactoring Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>