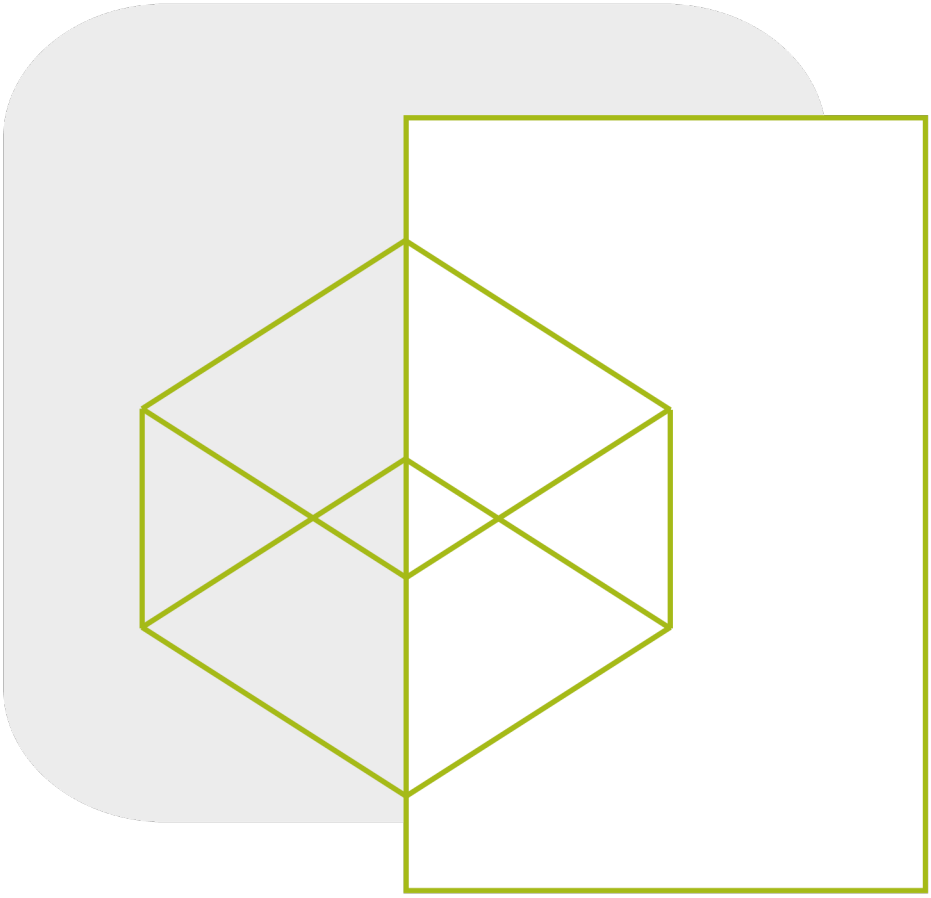
PIKETEC
software engineering

Release Notes

Version 12U2



TPT
Time Partition Testing

PIKETEC software engineering

Table of Contents

Release Notes TPT 12

TPT 12U2

Bug fix

- Severe issue: TPT could potentially corrupt the value of some channels or parameters during runtime, leading to unexpected corrupted data without any warning. More detailed information is available in the TPT Severe Issues document. Please visit the PikeTec website, Download section.

TPT 12U1

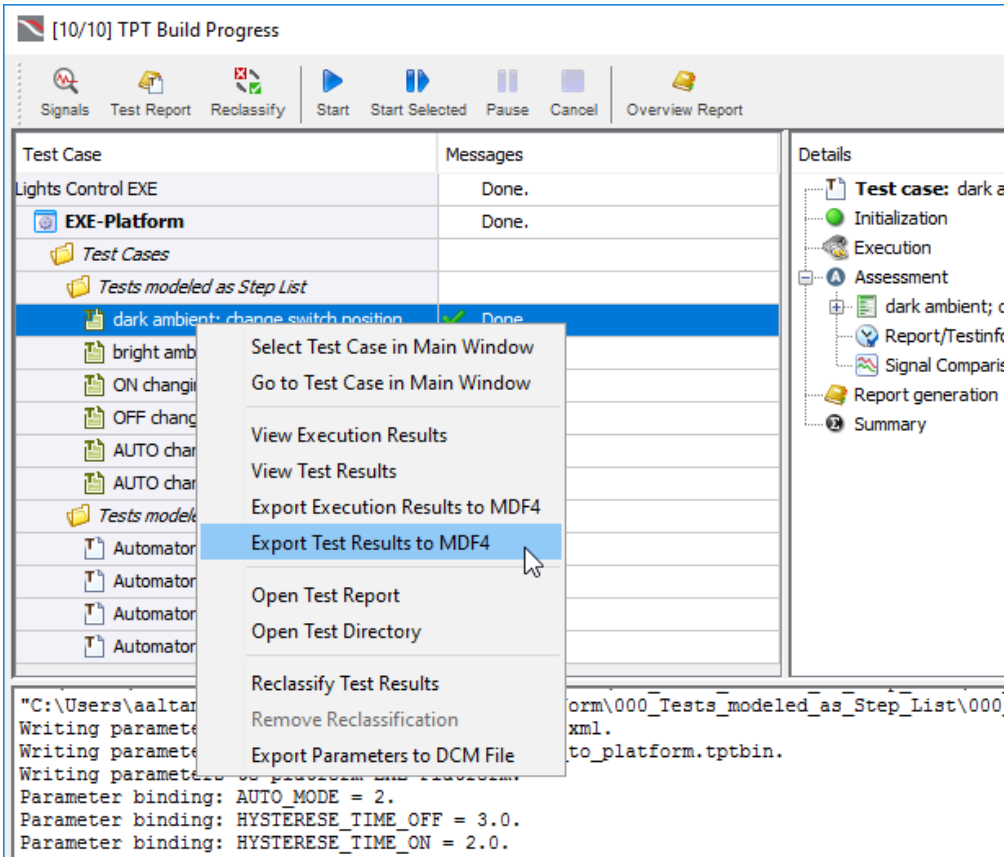
Bug fix

- Deadlock fixed in Build Progress dialog.
- Bug fix when importing map values with referenced axes.
- Bug fix in the Automatic Include feature for variant groups.
- Bug fix during calculations with boolean types in constant channels.
- Improved performance for HTML report generation.
- Bug fix with structs in mappings used with Testlet Libraries.
- Deadlock fixed in Test Set Definition dialog.

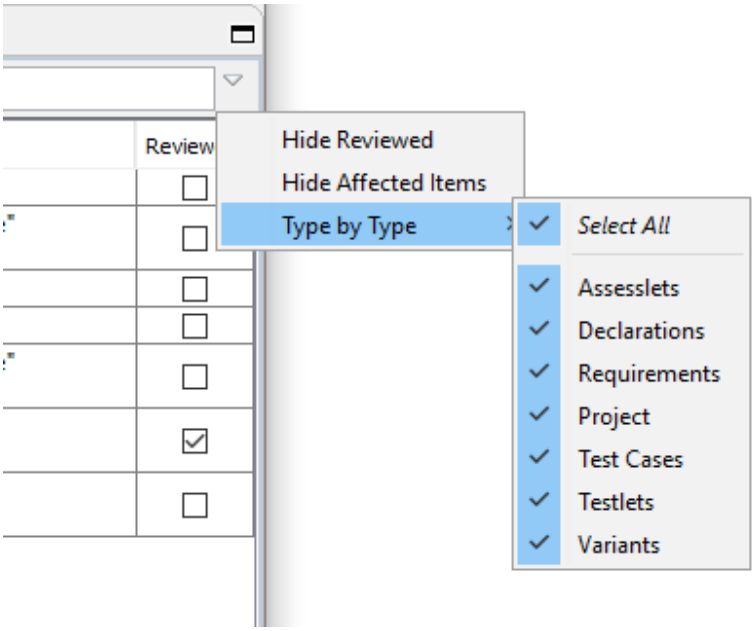
General

New

- Step list: "No Button"option in the Message box test step to close the message box only when the termination condition becomes true.
- Build progress: It's possible to reclassify test results of several test cases at once.
- Build Progress: Test results and execution results can be exported to an MDF4 file directly from the Build Progress.

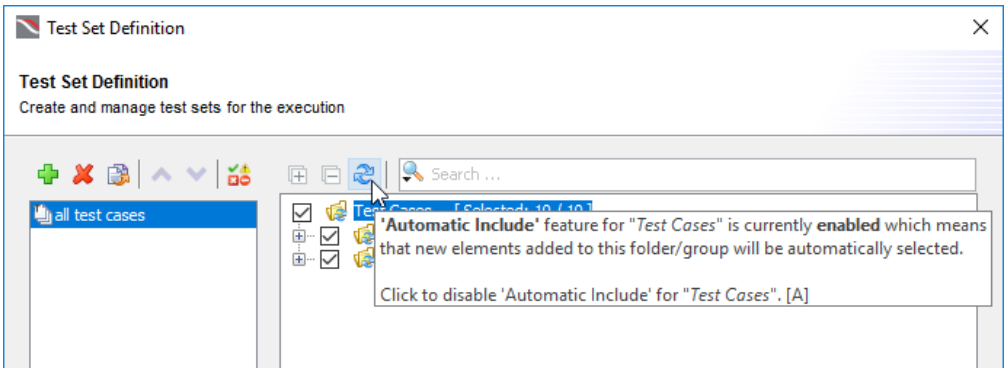


- Modifications view: New Hide Affected Items filter option in the modifications view to see a simple list of modified items.

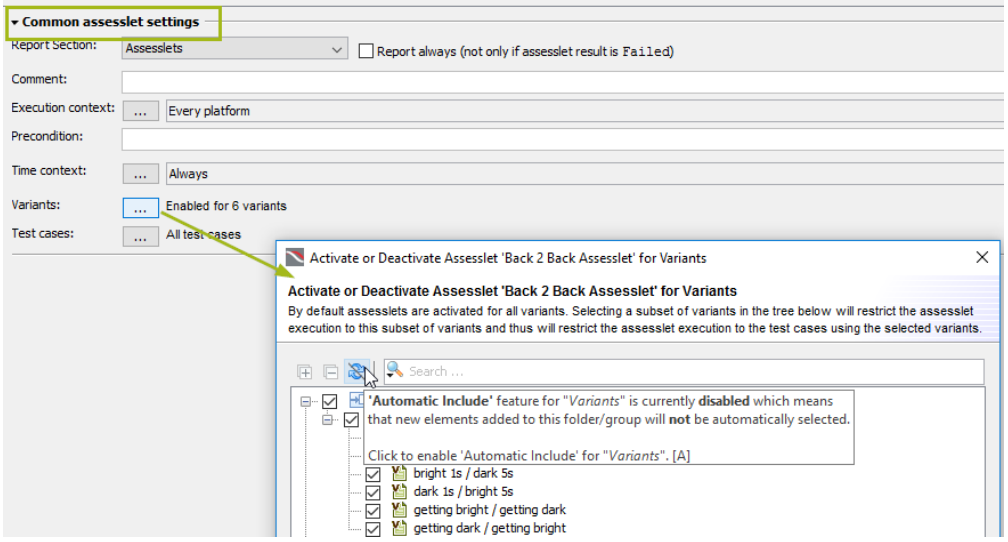


- Test Set: New button to activate or deactivate the automatic include of new test cases to Test Sets. This button is available in:

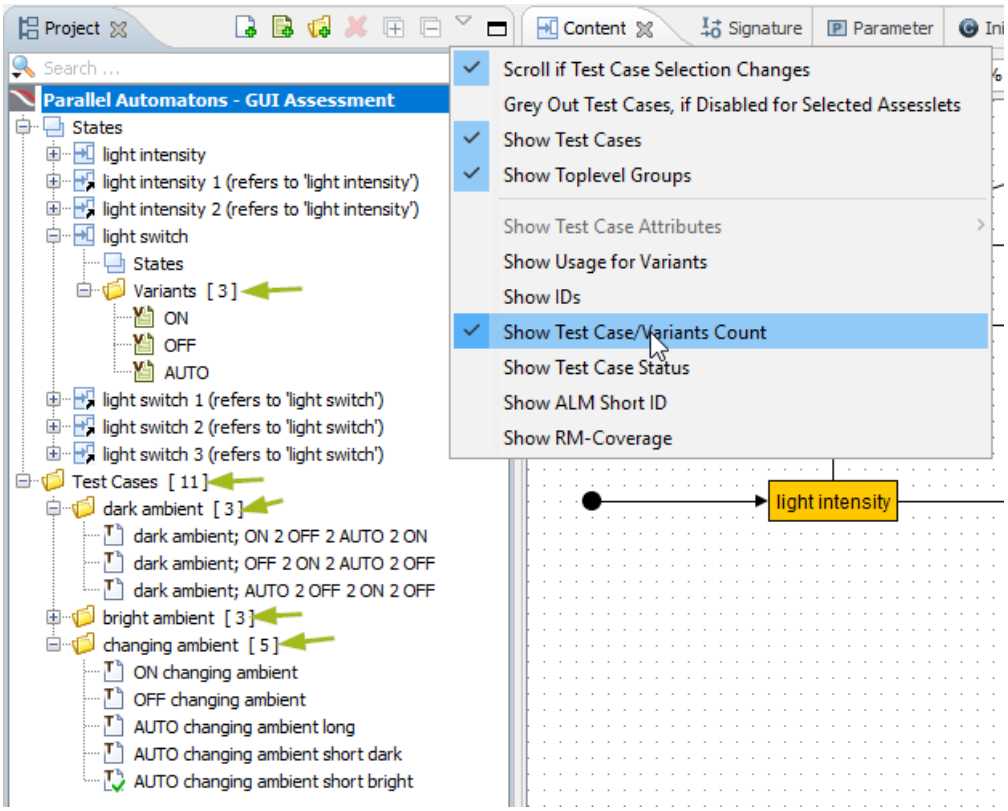
- The Test Set Definition dialog



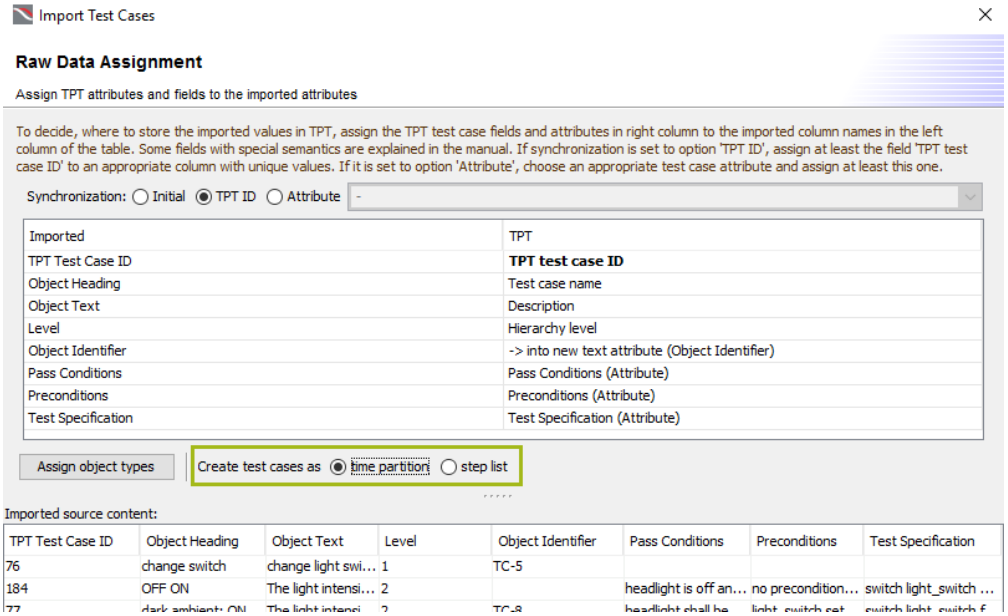
- In the common assesslets settings section, activate or deactivate assesslets, both for test cases and variants



- Copy / Paste is now available for Test Case Attributes in the Test Case Details View.
- Project view: New option to show test case count and variant count for each folder in the project view.



- Test Cases Import, Raw Data Assignment: new option to decide whether to create test cases as time partition or as step list.



- The TPT menu item Tools | Copy Outputs to Local Proxies now also works for step list test cases.
- TPT RMI API: equals() and hashCode() supported for all API objects.

- Batch script: New command "`--testSet "test_set_name"`" to use a specific test set via a batch script.

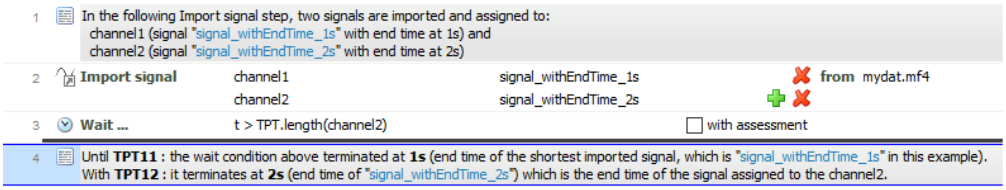
Changed

- The semantics of `TPT.length()` in "Wait" steps in conjunction with "Import signal" steps has changed, but only when the following conditions are given:
  - An "Import signal" step exists with at least two imported signals which have different end times.
  - A subsequent "Wait" step exists that contains the expression `TPT.length(X)`. Consider "X" to be a channel used in the "Import signal" step, which assigned signal has not the shortest end time (is not the shortest signal found in the "Import signal" step).

  The semantic change is as follows:

  - Until TPT11: `TPT.length(X)` had been resolved to the shortest overall end time (shortest signal) found in the "Import signal" step.
  - With TPT12: `TPT.length(X)` is now resolved as the end time of the signal assigned to the channel "X".

  See the following example to better understand this special case:



  This semantic change has been introduced to ensure a consistent and intuitive behavior of `TPT.length()` in all cases.

- Units cannot be set for structs, curves and maps. When loading old files that had units defined in structs, curves or maps, these units will be passed on just to those elements that are one level down, provided they are not structs, curves or maps and do not have their own unit (if any existing, it will be kept). If the type of a signal is set to struct, curve or map, an existing unit is deleted.

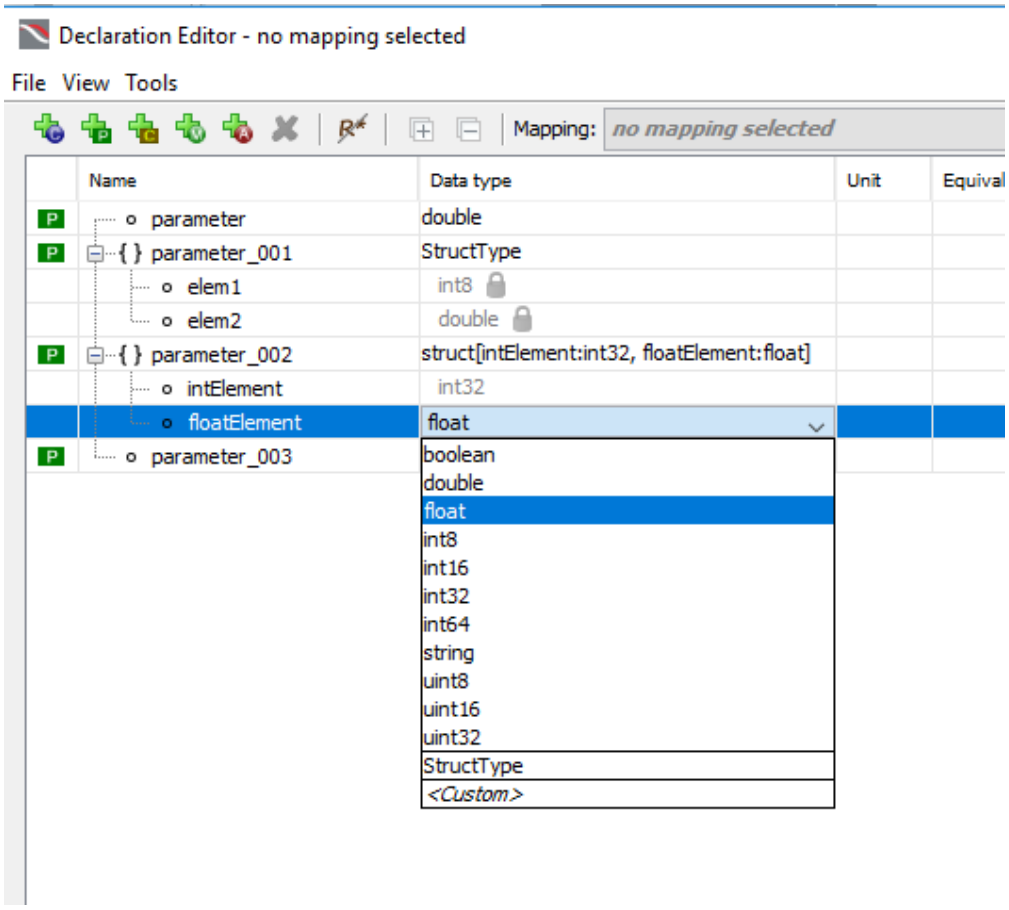
Bug fix

- Fixed axes of maps and curves are supported when importing the interface from MDF files.

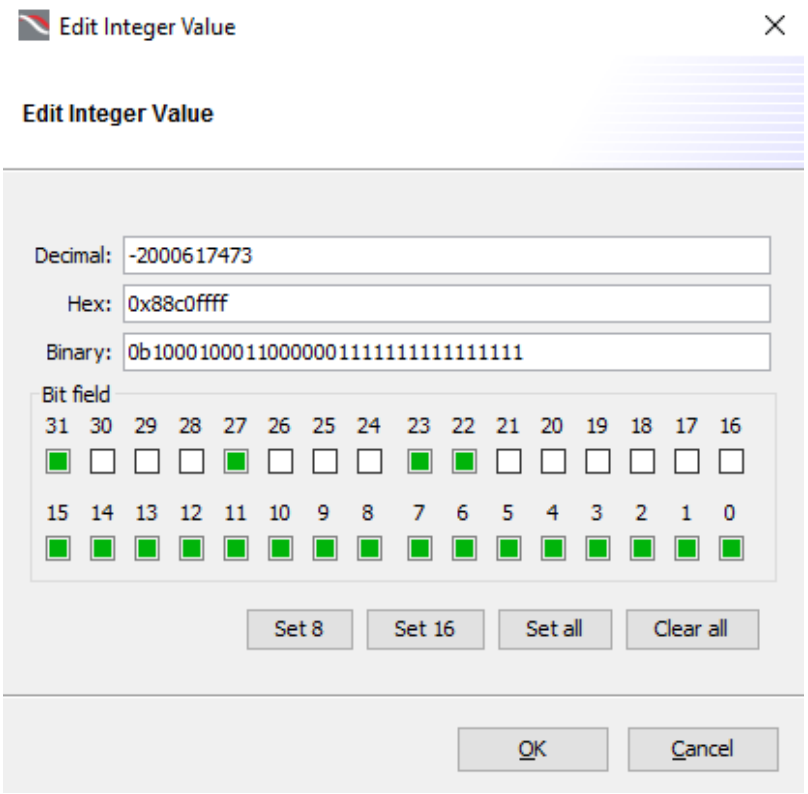
Declaration Editor

New

- The types of struct elements are now displayed in the Declaration Editor and can be changed for custom types.



- Interface import from FMI model description files (*.xml)
- Import/ Export: Importing interface from an XLSX file: you can decide how to order the columns to be imported. Columns with unreadable names are ignored.
- Type Editor: "Select unused" button to highlight those custom data types that are not used in any channel, parameter, constraint, assessment, measurement, or function, and that are not referred to by other types.
- You can now edit integer values bit by bit using the value editor.

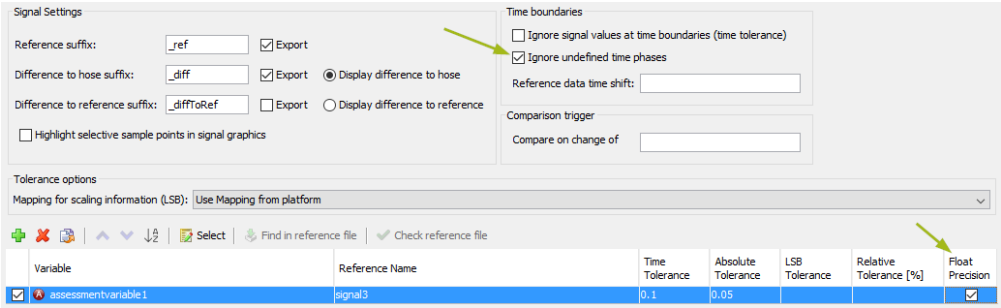


- The currently selected mapping is used during the interface import. In case no mapping is selected in the Declaration Editor, the last mapping used is automatically selected.
- New shortcut to open the Import Interface wizard (Ctrl + I). The Export Interface wizard can be opened from within the Declaration Editor by using Ctrl + O.
- For int8, int16, int32, and int64 data types, negative values can be represented in complementary notation in HEX- or BINARY-format in a step list.

Assesslets

New

- A comment can be added to every definition interval of a signal using the function `signal.setComment()`.
- Signal comparison assesslet and Back-to-back settings:
  - New option to "Ignore undefined time phases" in both reference and observed signals. This way, signals are not compared in those contexts where signals are not defined.
  - Float precision check box: select this check box to compare the observed and reference signals with float precision instead of double precision.



Changed

- Behavior of TPT.checkAlways changed: For `TPT.checkAlways (EXPR, "text1", "text2")`, if the `EXPR` is "undefined" for all samples in the current context interval, this is now considered to be an error and the result signal is accordingly set to "false" in the entire context interval.

Assessment

Changed

- For signals that do not start at t == 0s, the function `TPT.length()` now always returns the duration from 0s to the signal end.
- New assessment functions for Script assesslet:
  - **TPT.assertAlways**: checks if the time dependent expression `expr` of type `boolean` is `true` for all points in time of the current context interval and generates a report entry.
  - **TPT.assertExists**: checks if the time dependent expression `expr` of type `boolean` is `true` for at least one point in time of the current context interval and generates a report entry.
  - **TPT.assertTrue**: checks if the condition `cond` is true and generates a report entry.
  - **TPT.assertFalse**: checks if the condition `cond` is false and generates a report entry.
- New "Save" button to synchronize the condition tree inside the Signal Viewer with the already created condition tree assesslet.

Requirements

New

- Requirements marked in TPT as removed are displayed in the report with struck through ID and struck through object text.

1.2 Report Linked Requirements

State	ID	Linked	Details	Text
?	SPEC-9	indirectly via /Init/dark AUTO	Requirement was not checked by any assesslet.	lights control shall provide an applicable parameter MIN_LIGHT_ON (which represents the lower bound for the hysteresis).
✔	~~SPEC-17~~	directly	✔ headlight ON if switch AUTO and dark, headlight hold if switch AUTO hysteresis dark	~~During light_switch operates in mode AUTO and light_intensity is less than MIN_LIGHT_ON for at least HYSTERESIS_TIME_ON, headlight shall be ON.~~
✔	SPEC-18	directly	✔ headlight OFF if switch AUTO and bright, headlight hold if switch AUTO hysterese bright	During light_switch operates in mode AUTO and light_intensity is greater than MIN_LIGHT_OFF for at least HYSTERESIS_TIME_OFF, headlight shall be OFF.

Runtime Coverage of Requirements (only linked requirements)

- Auto review check box in Raw Data Assignment: Changes in requirement attributes during a requirements import can be automatically reviewed during import to prevent:
  - Your requirements from being marked as changed.
  - All linked elements from being marked as to be reviewed in TPT.

Raw Data Assignment

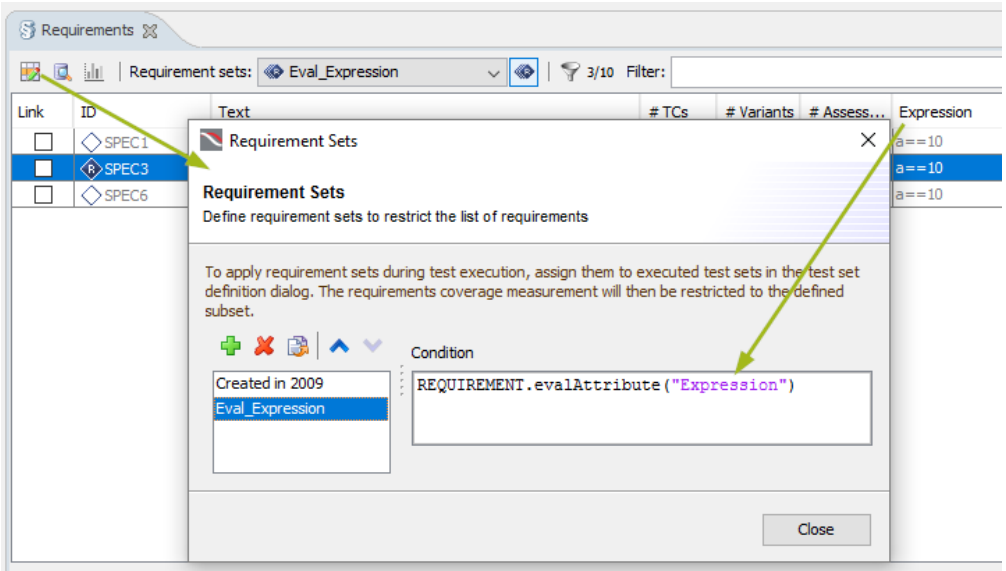
Assign TPT attributes and fields to the imported attributes

To decide, where to store the imported values in TPT, assign TPT requirement fields in the right to the imported column names in the left column of the table. Some fields with special semantics are explained in the manual. Assign at least the fields 'Requirement ID' and 'Requirement text' to an appropriate column.

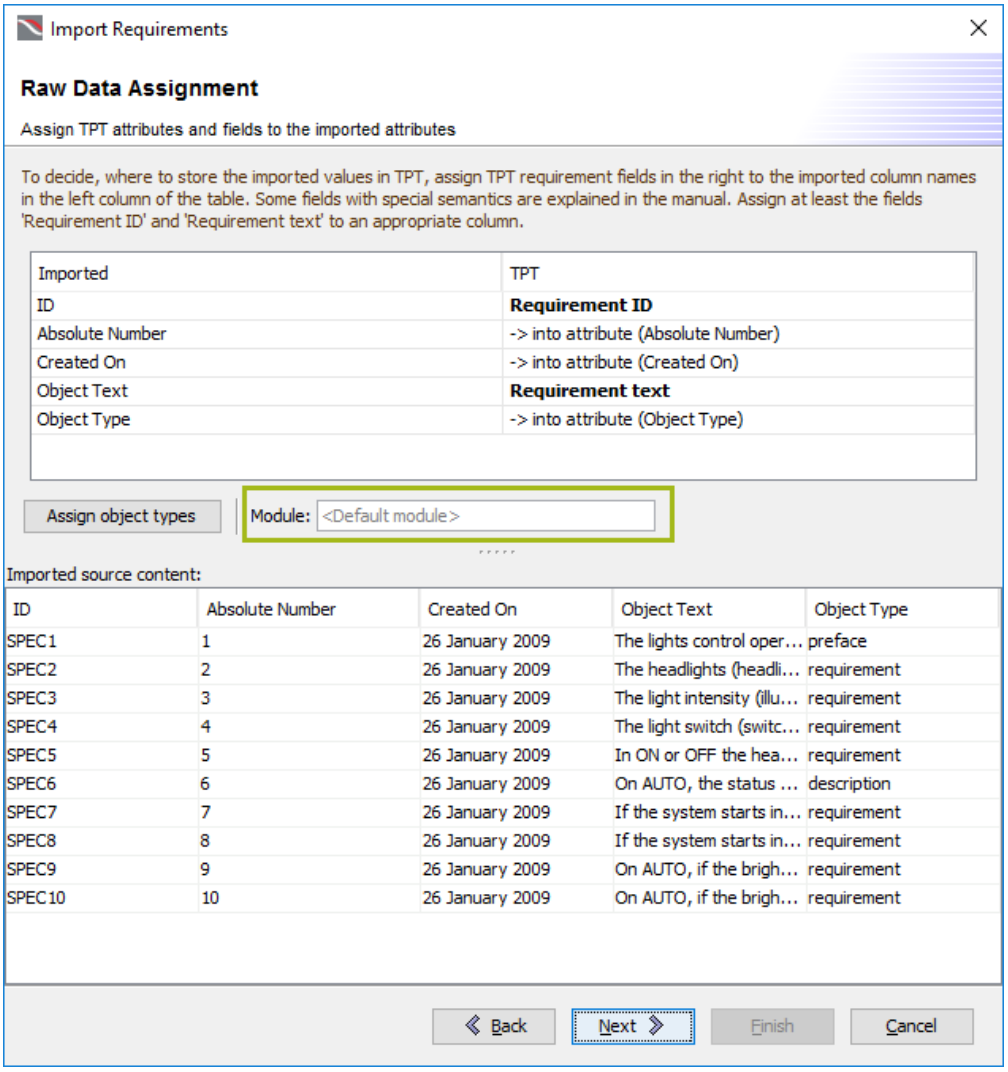
Imported	TPT	Auto review
ID	**Requirement ID**	☐
Absolute Number	-> into attribute (Absolute Number)	☑
Created On	-> into attribute (Created On)	☑
Object Text	**Requirement text**	☐
Object Type	-> into attribute (Object Type)	☐

Assign object types | Module: <Default module>

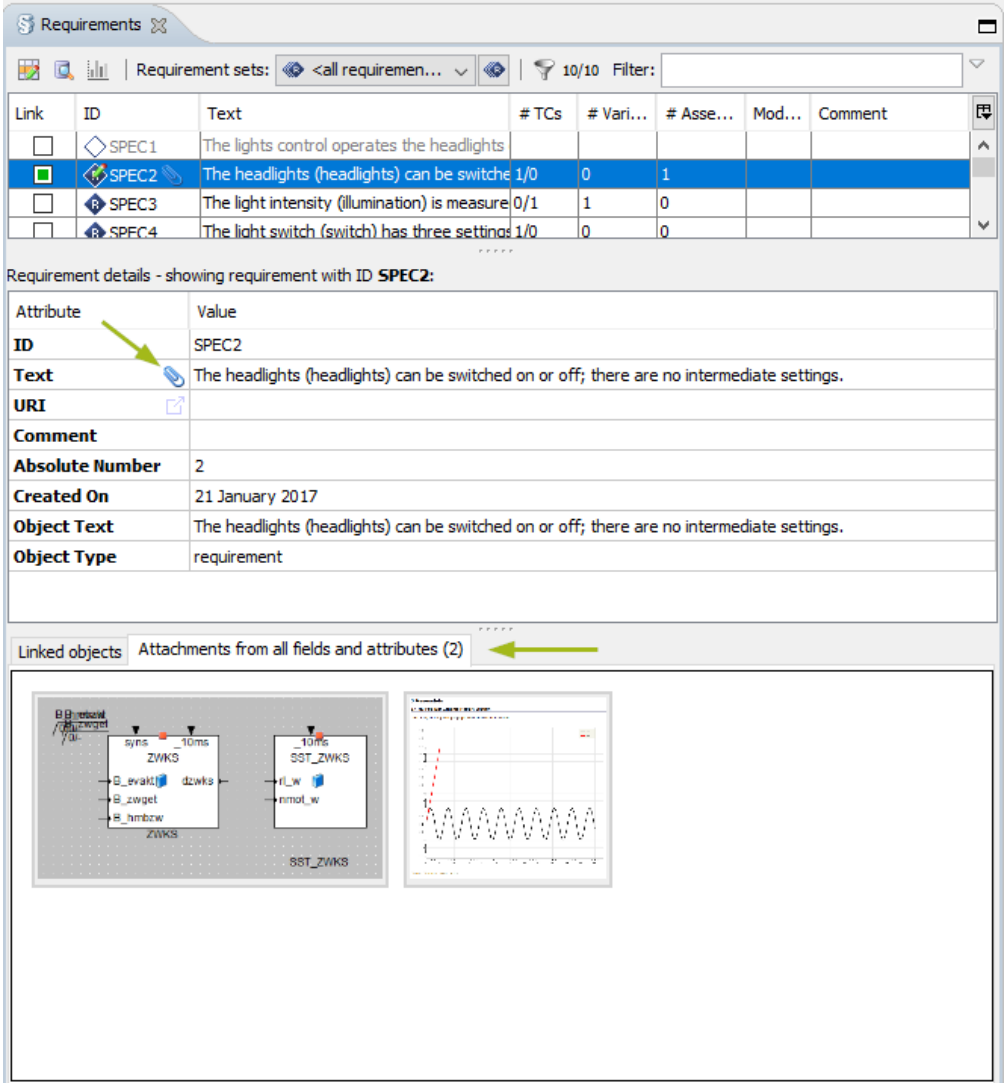
- Comments added to requirements in TPT can be displayed in in the report.
- Requirement sets can now be created also based on the content of the TPT internal comment field.
- Requirement sets can now be dynamically generated based on conditions given in requirements attributes, using the new `REQUIREMENT.evalAttribute(String attributeName)` function.



- Requirements Import, Raw Data Assignment: In case no module information is found, you can manually assign a module name using the Module field.



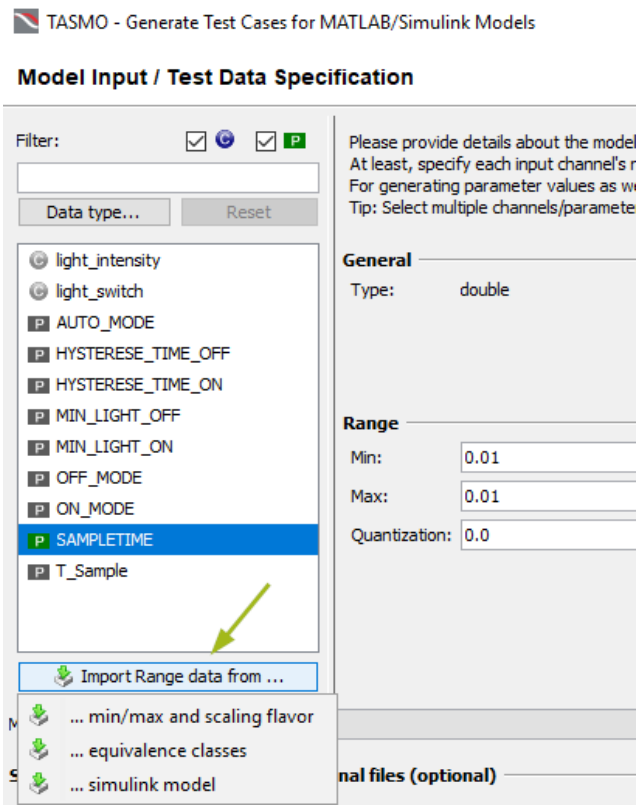
- You can import OLE objects from DOORS as attachments (images) to TPT.



TASMO

New

- Instead of entering the range information for signals manually, you can now import the range data from an equivalence class, a min/max and scaling flavor, and in case of TASMO for Simulink also from a Simulink model.



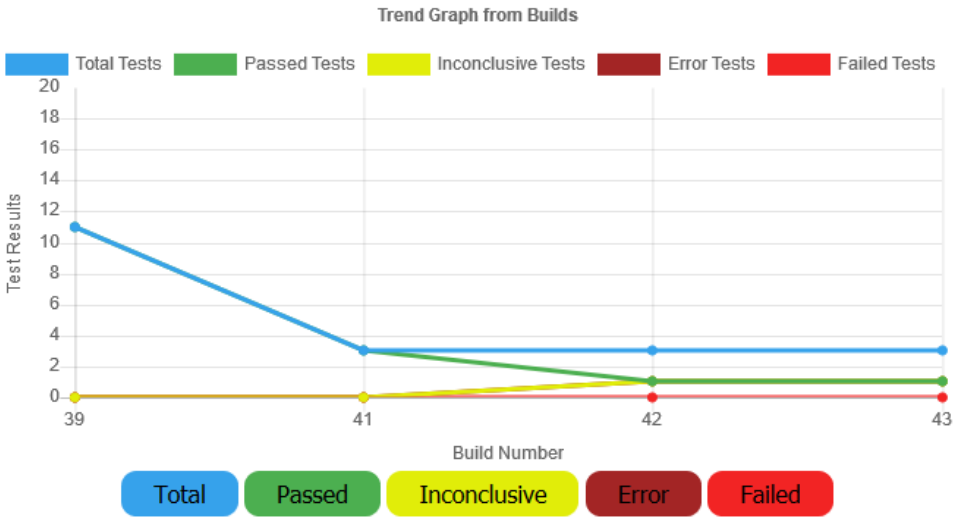
- Automatic import of min/max values from TargetLink Data Dictionary as default values.
- Support for matrix-signals as model inputs.
- Model reference blocks are now supported by TASMO.
- Regarding the coverage analysis by TASMO for C, all data types used inside a condition are supported.

Bug fix

- In some cases TASMO did generate step lists that did not match the reported coverage. This issue is fixed.

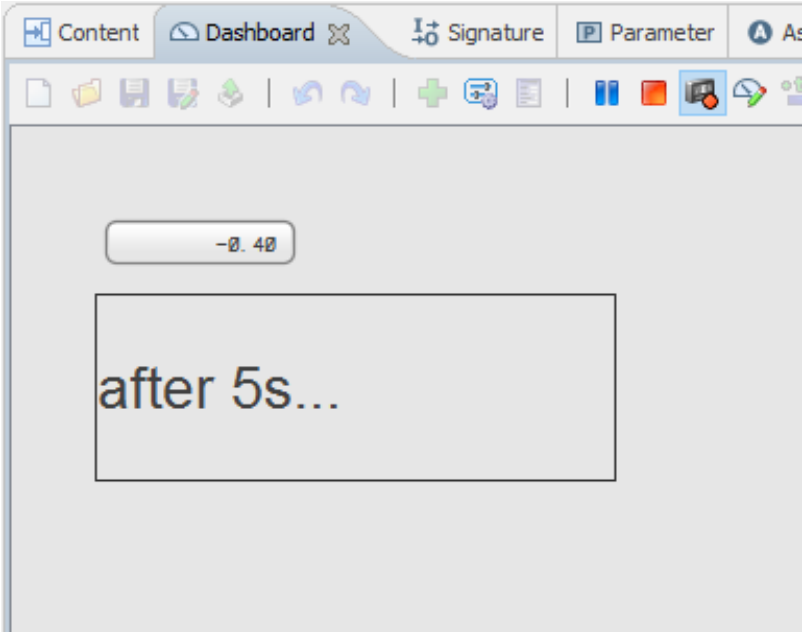
Jenkins

- Improved Jenkins plug-in:
  - You can now select test sets for the execution.
  - The TPT HTML report can be opened in Jenkins.
  - A TPT test results trend graph is shown in Jenkins.



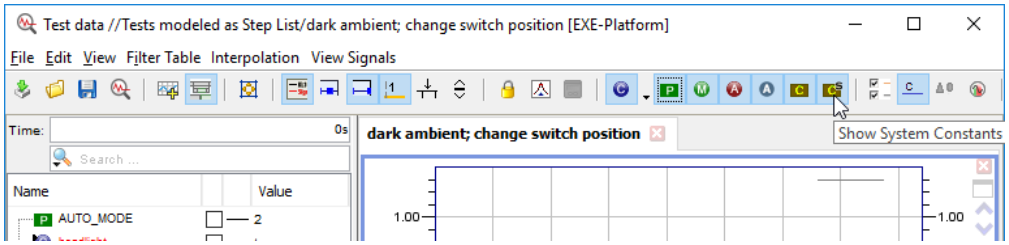
Dashboard

- New Dashboard String Display widget to display the value of string channels.



Signal Viewer

- Signal Viewer preferences can be loaded together with test data by using the command line.
- Constants and System Constants are visible in the Signal Viewer.

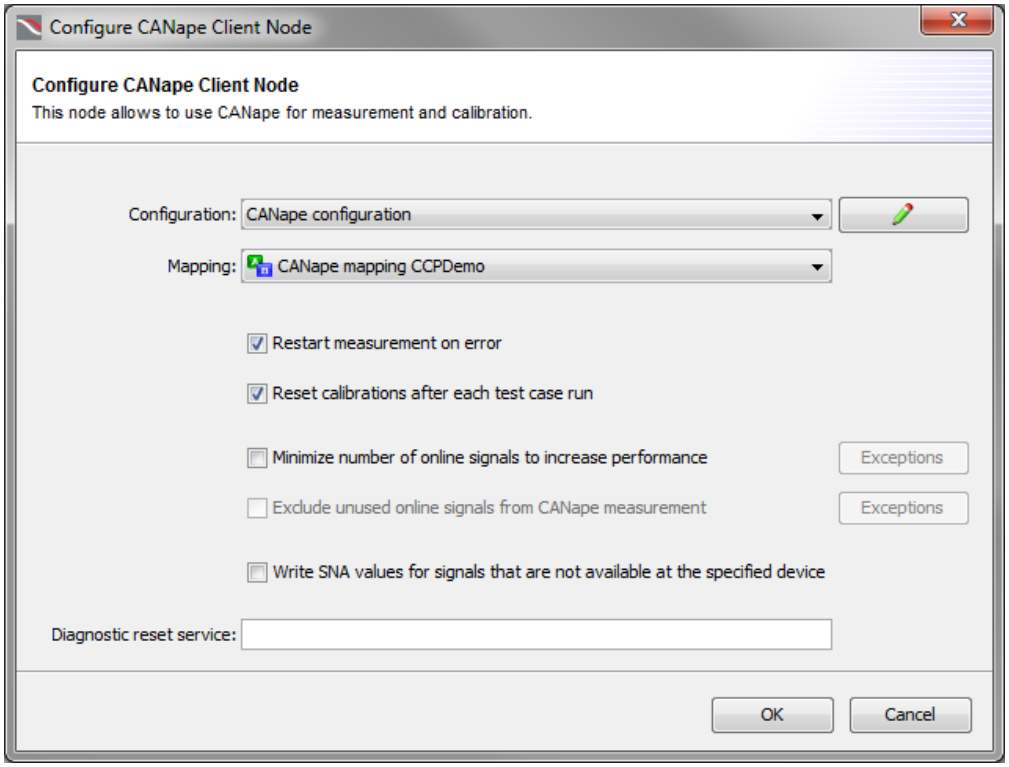


PLATFORMS

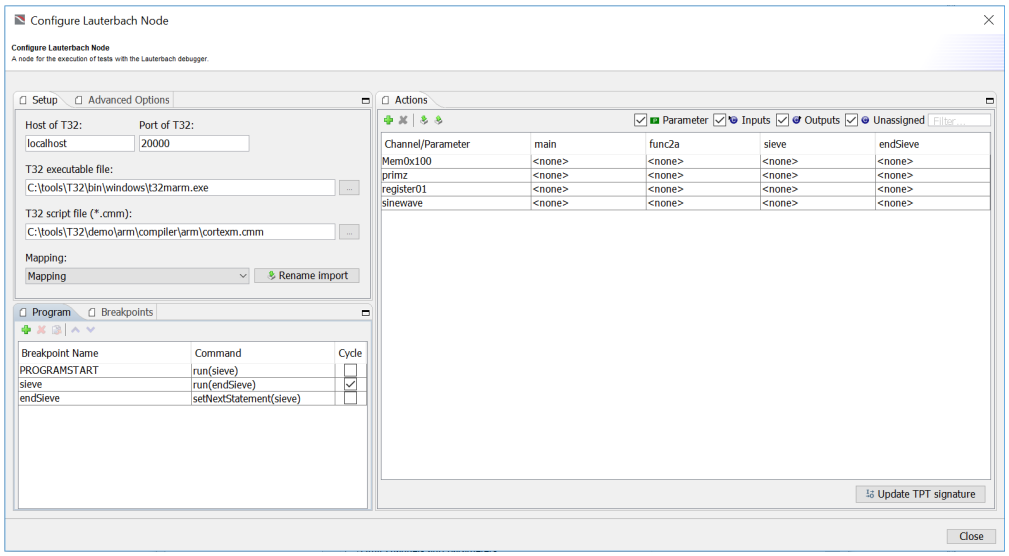
FUSION Platform

New

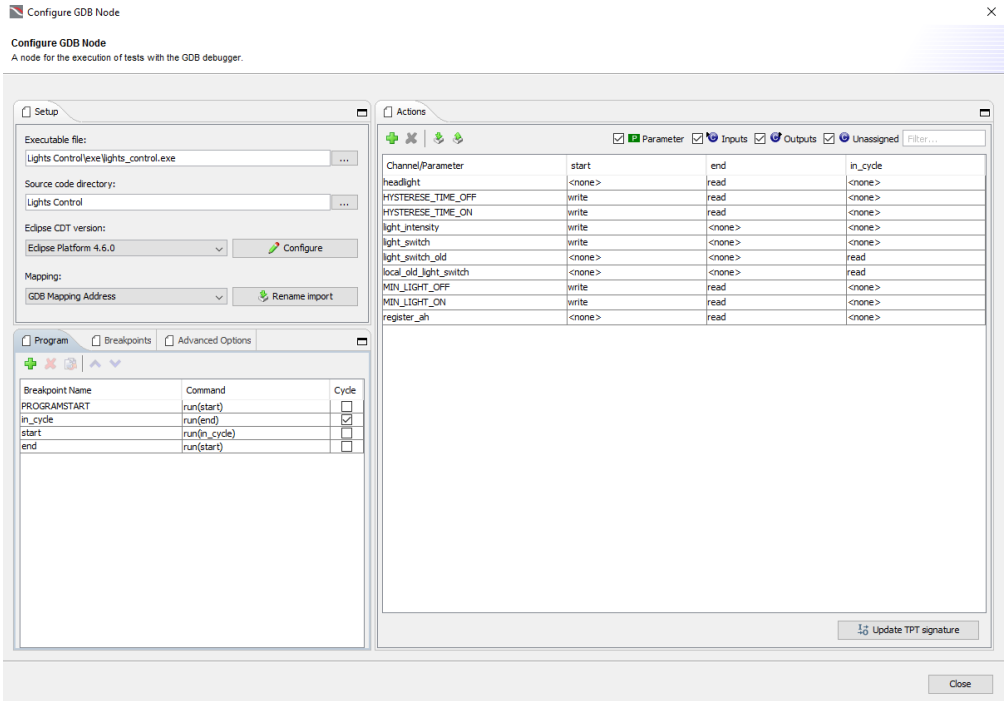
- New CANape connection with new features:
  - Measurement
  - Calibration
  - Diagnostic request, read / write requests
  - Restart measurement on error
  - Reset calibrations after test run
  - Minimize number of online signals
  - Exclude unused online signals
  - Write SNA values for signals that are not available at the specified device



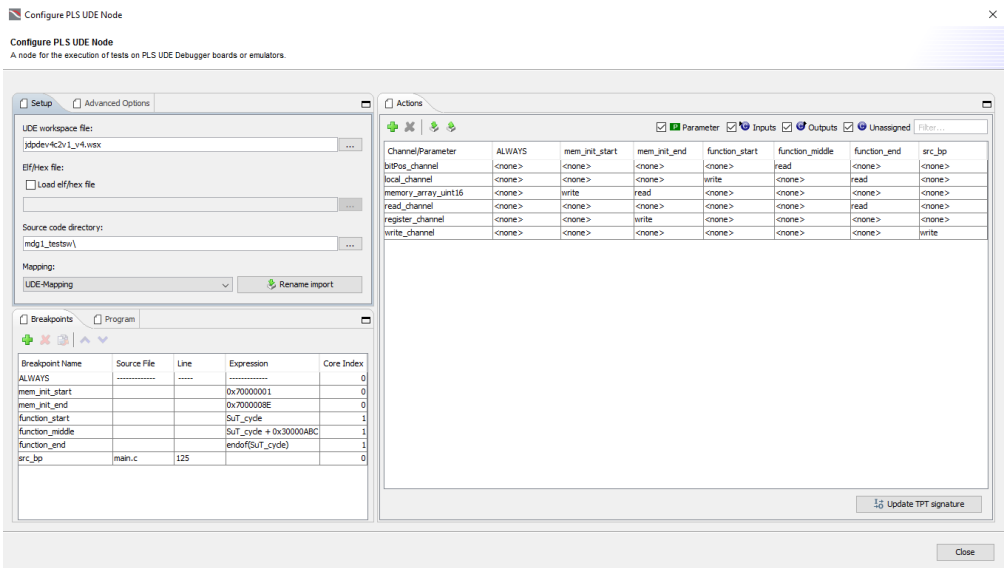
- New FUSION node to connect TPT with Lauterbach Trace32 software.
  - Support of Lauterbach commands. You can also configure a number of breakpoints, without being listed in the program for each program step, in the advanced options tab.
  - Support of PiL testing via Lauterbach.



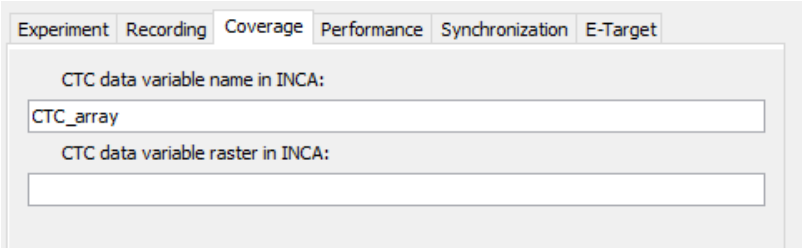
- GNU Debugger node is not an incubation feature anymore and it is now available for all users.



- PLS UDE node:
  - New dockable GUI.
  - Specifying breakpoints using long description of location is now possible.
  - Search field now available.
  - Breakpoints can be ordered in the breakpoints table.
  - Changes made through the GUI take effect immediately (OK and Cancel buttons were removed).
  - New option to display a message during the test execution whenever UDE stops at a breakpoint that is not specified in TPT.



- FMUs for co-simulation are supported as FUSION nodes.
- "New Test Execution - Co-Simulation with FUSION" example available in TPT.
- New coverage tab in the INCA Client node to define the INCA variable and its raster.



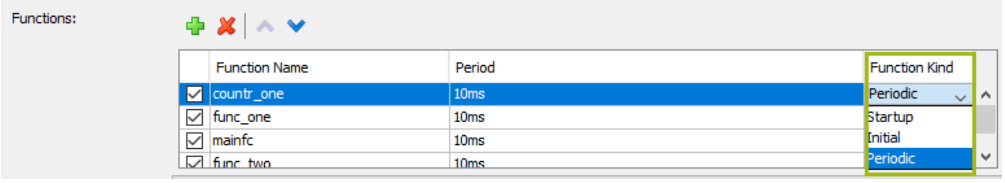
Bug fix

- String parameters from INCA are now read correctly by TPT and can be manipulated in TPT.

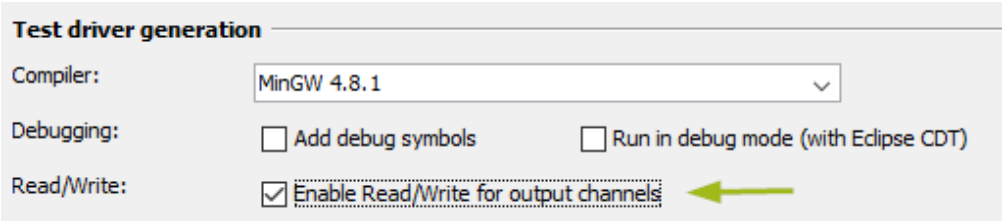
C Platform

New

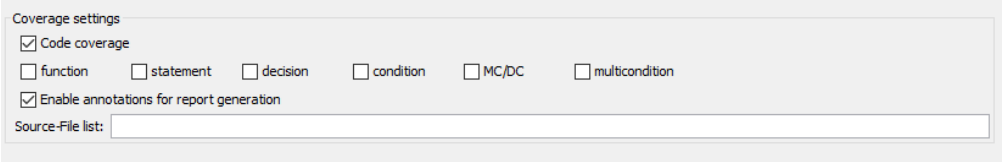
- Enums are supported.
- You can now start a debug session in Eclipse CDT from within TPT during the test execution. TPT can automatically create an Eclipse project for the C platform.
- Apart from the data type Boolean, TPT can now also handle variables of the data type _Bool.
- The extracted interface is automatically bound to the SUT by TPT.
- Function Kind: You can select if a function should be executed directly after starting the executable (Startup), at the beginning of the test execution (Initial), or repeatedly during the test execution (Periodic).



- "Enable Read / Write for output channels" option enables read / write functionality for TPT output channels when you generate and compile the test driver.



- C-Platform supports CTC++ code coverage. Coverage can be enabled, and metrics can be selected.



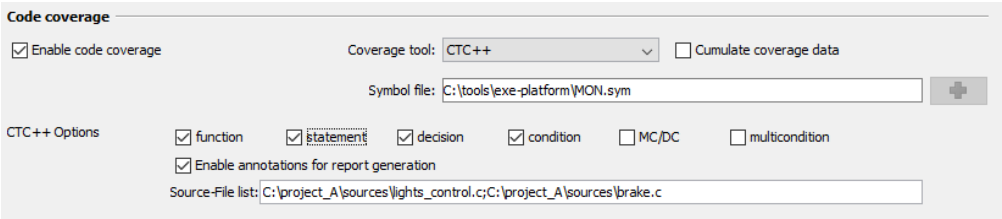
Bug fix

- Scalar pointer variables like int *p are now properly connected.

EXE Platform

New

- New CTC++ options, metrics can be selected.



MATLAB Platform

New

- Support for writing string array parameters.
- Import Code coverage with CTC++ and Targetlink Version 4.3
- Code coverage with MATLAB 2017b supported through "Simulink Coverage" MATLAB toolbox.

Bug fix

- The "Prepare model for fusion" button now makes MATLAB to use a matching ert target if the original model also uses an ert target (for example: ert.tlc).
- Generation of a custom FUSION dll node now suports MATLAB from version 2016b onwards.

VeriStand Platform

New

- Parameters can be imported from a SDF file and manipulated in TPT.

dSPACE HIL Platform

New

- Import interfaces via the XiL-API.
- Running tests in real-time via the XiL-API.

dSPACE HIL@FUSION Platform

New

- The dSPACE HiL@FUSION platform can now run test cases with both dSPACE 32 and 64 bit installations.

Silver Platform

New

- New example for the Siver platform to run the lights control model.

ASCET@FUSION Platform

New

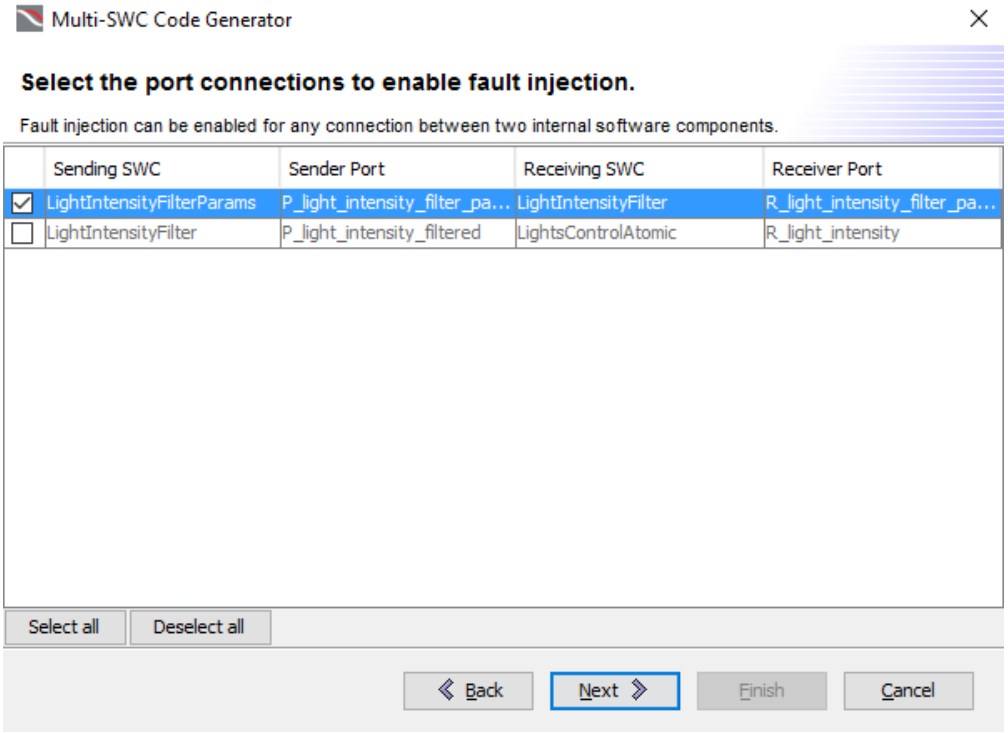
- You can now start a debug session in Eclipse CDT from within TPT during the test execution.

AUTOSAR platform

New

- Support of Rte_feedback functions for sender ports with acknowledgement request.
- New option to add extra "include" files outside an AUTOSAR project using the AUTOSAR code generator wizard.
- In case of S/R ports: channels are only created if any access to the corresponding AUTOSAR element is declared.
- New CTC++ options, metrics can be selected.
- Support of per instance parameters.
- Multiple instantiation of Atomics are now supported.

- Ports that are sender and receiver at the same time are now supported.
- Fault injection feature to inject faults (manipulate data) between two ports.



- Fault injection also with S/R ports using queued communication.
- TPT supports now explicit reading via return value (Rte_DRead). Previously, only explicit reading by argument (Rte_Read) was supported.
- If the generation of the CDS component has been suppressed during the generation of the C-code, TPT no longer uses Rte_DE and Rte_DES types.

Changed

- You can now choose a Visual Studio compiler from a list of available compilers when specifying the settings for generated SWC C code wrappers.

Bug fix

- Implicit communication is supported with PR ports.
- PIMs of a declared type Boolean with constants are now imported correctly.
- Support of shared parameter for software components that support multi-instantiation.

PREVIOUS RELEASES

TPT 11

General

- A transition can now refer to a another transition.
- A new view has been added, called Modifications: this view is used to show changes and the items affected by these changes in the project and also the differences between two TPT files.
- The default or custom perspective can now be selected from the toolbar. The last two used perspectives are displayed as buttons beside the default perspective icon.
- All open TPT projects can now be saved by clicking a "Save all" button.
- Auto-cast feature. TPT will now automatically cast the expression of type X to type Y if type Y is a superset of the value range of X and is defined for primitive types only.
- New functions to convert a int64 value to the corresponding IEEE754 64bit floating point equivalent (ieee754double), and an int32 value to the corresponding IEEE754 32bit floating point equivalent (ieee754float).
- New cast operator (string)num to convert numbers into strings.
- Information about the transition usage has been added to the Specification of Transition to see in how many test cases a transition variant is at least used once.
- The parameter consistency check inside a TPT model is now more strictly. Detected inconsistencies are immediately shown in the TPT project browser as well as in the test step list.
- Windows XP is no longer supported.
- The Step List testlet and the Time Partition testlet have been merged into the Local Content testlet.
- A transition can now refer to a another transition.
- A new view has been added, called Modifications: this view is used to show changes and the items affected by these changes in the project and also the differences between two TPT files.
- The default or custom perspective can now be selected from the toolbar. The last two used perspectives are displayed as buttons beside the default perspective icon.
- All open TPT projects can now be saved by clicking a "Save all" button.
- Auto-cast feature. TPT will now automatically cast the expression of type X to type Y if type Y is a superset of the value range of X and is defined for primitive types only.
- New functions to convert a int64 value to the corresponding IEEE754 64bit floating point equivalent (ieee754double), and an int32 value to the corresponding IEEE754 32bit floating point equivalent (ieee754float).
- New cast operator (string)num to convert numbers into strings.

- Information about the transition usage has been added to the Specification of Transition to see in how many test cases a transition variant is at least used once.
- The parameter consistency check inside a TPT model is now more strictly. Detected inconsistencies are immediately shown in the TPT project browser as well as in the test step list.
- Windows XP is no longer supported.
- The Step List testlet and the Time Partition testlet have been merged into the Local Content testlet.

Declaration Editor

- New refactoring feature in the Declaration Editor to rename signals in the TPT model.
- Measurement variables can be now also be of the data type "struct".

Import and Export

- New wizard to import and export test cases and requirements.
- Settings concerning the requirements import and and the test case import / export are now saved within the TPT project file.
- The two pie charts of the requirements section in the report have been merged into a single pie chart.
- New option to import only undefined parameters in the Parameter view.
- When importing test cases with URIs from Excel files, the URIs must be written in plain text. TPT does not read Excel links or hyperlink formulas anymore.

Test Step List

- With the Import Signal step, signals of the data type "struct " can be imported.
- With the Ramp step, signals of the data type "struct" can be ramped.
- Python's math functions can be used in the Table step check columns.
- MATLAB functions can be called by the Call Function step.

Automatic Test Data Generation

- New option to generate individual variants (single probes) for each permutation while generating test cases from value ranges.
- Test cases from value ranges can be inserted as Channel step or as Embedded step.

Platforms

New execution platform named C platform.

AUTOSAR platform

- Queued receiver communication is supported.

dSPACE HiL platform

- Configuration of failures in the Call Function step of TPT's test step list via a graphical interface.

C Platform

- Option to select individual source files for the interface analysis and instrumentation.
- The compile commands are written to a batch file that can be manually adapted.
- Each function can be individually set to run only once at the start of a test case or periodically.

EXE platform

- Option in the Platform Configuration to include the I/O consistency check into test driver.
- New function tpt_vmapi_bindSignalFinalize().

FUSION platform

- New option in the INCA Client Node to define whether TPT should try to re-connect to hardware or not after a failed initial connection.

ASCET@FUSION platform

- New option to compile the DLL with debug information for the GNU debugger.
- ASCET processes can now be called directly from TPT test cases, that is independently from the scheduler.

MATLAB platform

- MATLAB functions can be called by the Call Function step in TPT's test step list.
- Simulink bus signals are now supported by TPT internal signals and can be imported as measurements.
- New option in the Platform Configuration to use the actual names that are available to the bus selector instead of the names provided by the Simulink.Bus object during the test frame generation.
- The TPT S-function supports the usage of symbolic dimensions.
- The test frame can be manipulated after its generation by using a custom script that is set up in the Platform Configuration.

CANoe Platform

- CANoe platform is not an incubation feature anymore but a full licensed feature.
- COM-interface importer improvements. Stability and performance improvements

PLS UDE node

- Core index of the ALWAYS breakpoint is editable.
- A source code directory can be chosen.
- New option to write only those variables (parameters/channels) that are used within the current test and assessments via Fusion to UDE.
- In breakless mode, the debugger will be started once after the target is reset.

TASMO for C

- TASMO for C (acronym for "Testing via Automated Search for Models for C-code") generates test data for the testing of C-code. It uses the C-platform.

TASMO

- A set of signals can be assigned to certain TPT channels or parameters
- TASMO supports the data port order "Specify indices" for multiport-switch blocks.
- The block paths in TASMO can be filtered with or without using regular expressions.
- Multiple selection of coverage goals is supported.
- Coverage goals for individual subsystems as well as single expressions of a subsystem can be manually canceled before running the test data generation.
- Export of TASMO input and/or the coverage overview to CSV.

Requirements

- New requirement set definitions to define requirement subsets.
- Requirements can be linked with test case groups and assesslet groups by a drag-and-drop operation.
- Instead of the linked object's name, its path and/or ID can be displayed.
- The status of unlinked requirements can be manually set to "New".
- Directly linked requirements of a test case can be shown in the Test Case Details view.
- Delete or create requirement links to selected test cases.
- TPT internal comments can be added to individual requirements.
- When using requirements in Script assesslets with the REQUIREMENTS.checked() function, autocompletion is supported.
- The order of requirement attributes can be manually changed.
- The requirements coverage statistics shows also the assesslet coverage.
- When importing requirements with URIs from Excel files, the URIs must be written in plain text. TPT does not read Excel links or HYPERLINK functions anymore.
- An optional table called "Requirements Assesslet Results" in the report shows the result of the requirements checked by assesslets for each execution platform.
- Modifications concerning requirements are now shown in the Modifications view.
- The "Needs to be reviewed" flag is replaced by the "modified" flag. The flag is controlled in the Modifications view.

Assesslets

- Python math module is automatically loaded for assessments so its functions can be used to assess tests.
- It is now allowed to call the function TPT.detectTimeShift() in time contexts. A cache is used to avoid redundant calculation for each time step.
- Compelete arrays can be compared in assesslets element by element.

- The method TPT.readMeasurementRecord() has again only two arguments.
- More information is added to the report when signals are compared

Min/Max Comparison assesslet

- Result channels (range_check, min_value, max_ value) can be exported individually.

Report Signal Graphic assesslet

- The Report Signal Graphic assesslet can also handle arrays and structs.

Report Signal Table assesslet

- When selecting the check box "Show as trace table" in the Report Signal Table assesslet, the generated table shows 32 entries also when the amount of entries exceeds 32

Script assesslet

- Script assesslets can be used as libraries to insert functions, classes, and objects defined in this assesslet via autocompletion in the following / next assesslets.

Signal Comparison assesslet

- Signal comparison assesslet: new "Find in reference file" option to specify a regular expression pattern to match TPT channels to signals in the reference file.
- When using the Signal Comparison assesslet, also the relative tolerance is now graphically shown in the report.

Trigger Rule assesslet

- Multiple THEN - ELSE conditions can be specified in the Trigger Rule assesslet. All conditions have to be fulfilled to pass the assesslet.

Execution Configuration

- The test execution can now be terminated after a given number of failed tests.
- You can switch from simple to advanced mode to specify the execution items.

Back-to-Back Test

- Back-to-Back tests can be configured in the Execution Configuration.
- Parameter sets can be selected for Back-to-Back testing.
- The compare mode in the Back-to-Back settings lets you specify if the signals must be of the same length (strict) or if differences in signal length are tolerable (normal).

Build Progress

- The toolbar of the Build Progress is undockable.

Signal Viewer

- The Signal Viewer is now as a standalone application available with its own documentation.
- Up to five quick preferences can be saved for a single project and applied by shortcut or button to the currently presented signal data.
- Test case data or assesslet data can be placed in the Signal Viewer by a drag-and-drop operation.
- It is now possible to search for signals in plain text or by using regular expressions.
- The graphical representation of a signal in the Signal Viewer is highlighted when the signal is selected in the list of signals. New shortcut keys for this.
- Double-click on a tab name to open the respective test case in the TPT Project Browser.

Report

- It is now possible to generate a separate report that contains an overview of all assessment variables/assesslets and their usage in test cases.
- Test Case Attributes are no longer chosen in the Advanced Report Settings of the Execution Configuration but in the Meta Info Report Table assesslet.

Dashboard

- New methods to change panels by condition and track panel changes by using the Dashboard script.

Remote API

- New classes and methods have been added to the Remote-API, for example, to create assessment variables and to add mappings and mapping falvors.

Search and Replace

- Parameters whose values have been defined in the Parameter view are now found by the local and global search.
- Local search is now also supported in the Debug view and the Functions view.

Documentation

- In the HTML5 help, a client feedback button has been added so you can easily submit a feedback to the documentation team.
- Documentation updated describing all TPT-VM-API functions for the EXE platform.
- New Equivalence Classes example and documentation.
- New "Test Modeling - Automatons and Step Lists" example and documentation.

- In the HTML5 help, a client feedback button has been added so you can easily submit a feedback to the documentation team.
- Documentation updated describing all TPT-VM-API functions for the EXE platform.
- New Equivalence Classes example and documentation.
- New "Test Modeling - Automatons and Step Lists" example and documentation.

TPT 10

Declaration Editor

- The Declaration Editor has been completely redesigned.
- Direct display and edit of mapping flavors (scaling, rename, ...).
- Direct manage and edit of equivalence classes.
- Set interface roles directly (IN/OUT/LOCAL).
- Multi-editing supported.
- You can switch between the decimal, binary, and hexadecimal representation of an integer.
- Global undo (CTRL+Z) and redo (CTRL+Y) affects also actions made in the Declaration Editor.
- You can show and hide columns in the Declaration Editor.
- New shortcuts to create channels (CTRL+1), parameters (CTRL+2), constants (CTRL+3), measurements (CTRL+4), and assessment variables (CTRL+5).
- The single elements of a struct are now displayed as an expandable list in the Declaration Editor.
- Display of a struct in the Declaration Editor
- The values of string channels are now displayed.
- Min and max values for a struct or an array can also be scalar values that apply to all elements of the struct or array.
- Initial values can be set for channels by using an Init Values mapping flavor and individual test cases by using the Initial Values view.

Equivalence Classes

- New Equivalence Classes Editor.
- Direct editing of all equivalence class sets in the Declaration Editor.
- Equivalence classes can be accessed in assessment scripts.
- Equivalence classes set in an assesslet with auto-completion.
- You can automatically generate test cases from subsets of Equivalence Classes Sets.
- Equivalence classes can be defined as mandatory or forbidden in the Equivalence Classes assesslets.

Mapping Editor

- The Mapping Editor has been redesigned.

Import Interface

- Instead of remove, you can hide a signal during the interface import if this signal is declared in TPT but cannot be found in the external interface.

Import and Export

- TPT can import array elements from MDF files as array. The import can be configured in the Preferences dialog, TPT Model Behavior section.
- Comment information from MDF3 is imported as string.
- MDF4 event blocks are imported as string variables. These string variables are also displayed in the Test Data Viewer.

- TPT can import array elements from MDF files as array. The import can be configured in the Preferences dialog, TPT Model Behavior section.
- Comment information from MDF3 is imported as string.
- MDF4 event blocks are imported as string variables. These string variables are also displayed in the Test Data Viewer.

Test step list

- New Table step.
- Each step in the step list has its own documentation field.
- The Ramp channel step can now ramp parameters.
- Inside the step list, the set If, Else, While, and the Parallel steps can now be expanded and collapsed.
- Test cases can be generated from value ranges.
- You can use mapping information of test platforms by typing the channel name and flavor and set a ->. For example: light_intensity > light_intensity->Min.

Platforms

- You can now add custom Python scripts to the platform configuration. The scripts run either before or after the execution of the platform / test cases.

ASCET

- You can now extract the SUT interface, generate the test environment, and configure ASCET via the RMI API.

ASCET@FUSION

- New option "Limit channels and parameters" in the platform configuration lets you limit the physical values of channels and parameters to the bounds of a Min/Max flavor.
- Parameter are read back from the SUT. That is, in case parameter values are limited or changed in the SUT, TPT reads the limited value.
- Support of quantized physical experiment.
- Support of different implementations.
- You can set individual task–counter names.
- The order of tasks/processes is no longer limited.

- You can specify "round" or "cut" for calculations of internal values (scaling).
- TPT 64bit can now communicate with ASCET.

VERISTAND

- TPT checks whether the signals from a TPT project file are present on the HiL. The test cases are executed only when all signals given in the TPT project file can be found on the HiL.

FUSION

- New section in the FUSION platform configuration, named "Advanced Options", to specify the runtime and the debug options.
- Parameters in FUSION have now read-write semantic, so the values of parameters can be read back from the SUT via FUSION and become a part of the test data.
- 64bit DLLs can be used.
- Rename mapping flavor supported.
- The name of the MS.NET Assembly Node has been changed to Custom Node.NET DLL.

MCD3 client node

- You can now specify how many times TPT should try to connect to the controller if the initial connection breaks down, and how many seconds take between each attempt to reconnect.

INCA node

- The initialization process for the INCA Node has been optimized.
- You can suppress the initialization of INCA if INCA has already been started and configured and a test has to be executed several times.
- The INCA node supports array signals and measurements.
- INCA array measurements are now also array channels at TPT.

PLS UDE node

- You can now read from bit fields or write to bit fields.
- You can specify to open a given Executable and Linking Format file (ELF) automatically when the UDE is opened.
- If the ELF code is generated with CTC++, also code coverage is possible.
- The code coverage done with CTC++ on the UDE debugger target can be integrated in the HTML report of TPT.
- You can set the maximal timeout for waiting that UDE stops at a specified breakpoint by setting a breakpoint timeout in the node configuration.

LABCAR

- A new platform has been implemented, called "LABCAR Platform". It runs TPT test cases via the FUSION and exchanges the signal data with the LABCAR OPERATOR (LCO). The measurement and calibration is done via INCA.

dSPACE HiL@FUSION

- A new platform has been implemented, called "dSPACE HiL@FUSION". It runs TPT test cases via the FUSION in non-realtime on a dSPACE HiL environment. Communication is performed using the ASAM XiL-API.

MATLAB

- Better support of TargetLink Data Dictionaries.
- TargetLink subsystems built from a library are supported by TPT for automatic test frame generation.
- TPT environment variables are now written to the MATLAB workspace.
- In conjunction with the MATLAB platform, the MinGW 64 bit is supported in TPT.
- TPT can handle the Simulink.LookupTable that holds explicit values and the Simulink.Breakpoint that represents an axis of a Simulink.LookupTable.
- TPT can connect to resettable subsystems in MATLAB to analyze the subsystem's interface and to generate the test frame.
- Instead of memory blocks in the test frame, unit delay blocks (1 / z) are generated.
- Parameters in referenced Simulink models are taken into account by TPT during the import.
- Coverage with V&V is always cumulative.
- Coverage measurement is possible for referenced models.

TASMO

- TASMO is more robust and faster, and supports more Simulink blocks for coverage goals.
- TASMO supports decision coverage for the following blocks: Sign (Simulink, TargetLink), Relay (Simulink, TargetLink), and Dead Zone (Simulink).
- TASMO supports decision and condition coverage for Stateflow Truth Tables.
- You can filter subsystems by block type and coverage criteria in the "Select Coverage Criteria / Goals" dialog section.
- New signal characteristic named "Constant" has been added.
- TASMO performs a static analysis of the test model before the test data generation to detect non-reachable coverage goals.
- TASMO can analyze existing test cases and achieved coverage during the normal test execution to avoid redundant test data and to increase the coverage goals.
- The coverage information of the test cases generated by TASMO can be exported from TASMO into a CSV file.

- TASMO supports parameter changes in the test data generation.
- TASMO can generate test data as linear step lists or as parallel step lists.

Requirements

- TPT10 supports DOORS 9.6.
- You can deactivate in the Advanced Report option in TPT, that a requirement which is not linked to any assesslet, derives its result from a test case. When this option is deactivated, the requirement is set to "not covered" instead of "inconclusive".
- TPT can handle the URI Object attribute from DOORS. URI Objects are links.
- For better visibility, the position of the number located in variant and test case icons that shows how many requirements are linked to that specific test case or variant has moved upward.
- The requirements result of several execution platforms are displayed in a single report table; the pie charts show the results of the different execution platforms combined.
- The "Report Linked Requirement" displays more information, for example about the kind of link (direct, inherited).
- Report shows if the link to a requirement is direct or inherited.

- TPT10 supports DOORS 9.6.
- You can deactivate in the Advanced Report option in TPT, that a requirement which is not linked to any assesslet, derives its result from a test case. When this option is deactivated, the requirement is set to "not covered" instead of "inconclusive".
- TPT can handle the URI Object attribute from DOORS. URI Objects are links.
- For better visibility, the position of the number located in variant and test case icons that shows how many requirements are linked to that specific test case or variant has moved upward.
- The requirements result of several execution platforms are displayed in a single report table; the pie charts show the results of the different execution platforms combined.
- The "Report Linked Requirement" displays more information, for example about the kind of link (direct, inherited).
- Report shows if the link to a requirement is direct or inherited.

Assessment

- The Jython version has been migrated to Jython 2.7, thus TPT can now load DLLs at assessment time.
- The Assessment Library is now a regular feature and has been moved to the Preferences dialog.
- The Assessment Library has file extension *.tptpy.
- Specified functions can be used in assessments.
- New dialog to select variables in assesslets (Min/Max, Signal Comparison, Equivalence Classes).
- You can increase the context interval by positive arguments or reduce them by negative arguments with the new assessment function TPT.extendContextRel (before, after).
- In STRICT mode, undefined inputs lead to undefined results.

- The new option in assesslets tree "Copy the Structure of the Selected Test Cases" creates assesslet groups that are structured with regard to the test case groups. The corresponding linked assesslets are sorted accordingly.
- New operator "=~=" that compares two signals with respect to a tolerance.
- The semantics of the getTolerance() function also considers that the ScalingMode can be OFF.

Condition Tree assesslet

- You can disable and enable single checks by selecting the "Enabled" check box.

Trigger Rule assesslet

- The trigger condition can trigger an "Else check" for the intervals in which "Then check" is not true.

Min/Max Comparison assesslet

- You can now select a mapping with Min/Max flavor in the section "Mapping for bound information".

Script assesslet

- Locally declared functions and variables can be entered via auto-completion.
- Already declared elements are highlighted when you hover with the mouse over the script and at the same time press the CTRL key. When you click on a highlighted element, the Declaration Editor opens and directly shows the selected element, so you can immediately edit it .

Import Measurements assesslet

- The check box "Use mapping from platform" has been removed. Go to the "Import" section, click and select "Use mapping from platform" from the mapping list.

Report Signal graphic assesslet

- You can now filter signals that differ from default value to include them in the Report Signal graphic.

Build Progress dialog

- The Build Progress dialog has been redesigned.
- The execution details tree shows also which assesslet in what time context has been erroneous and at which Compare step something went wrong.
- You can directly jump to the assesslet in the Assesslet view or in the Assesslet Content view by clicking on it in the Build Progress.
- New toolbar icons added to cancel the current execution, to create and to open the overview report, to automatically generate an overview report after every test case is run.

Test Data Viewer

- You can load several test or measurement data into an independent Test Data Viewer tab or window.

Report

- The section "Test Case Status Summary" in the report table shows now the test case status that you set.
- Report shows information, which ASCET@FUSION project has been used for the generation of the DLL.
- PDF reports are now generated in landscape format for better table display.
- New variable ${tpt.scenario.inheritedcomment} to add the descriptions that were set to test case groups or test cases to the report.
- You can specify for which files the coverage report should be generated using CTC++.
- New Section "Code Coverage" in the report.

Dashboard

- You can now choose to set the mode of the button widget and check box widget either to push button or to toggle button.

Dashboard Player

- Simplified generation of the executable Dashboard Player file in a new format called DBPLAY.
- The DBPLAY files are automatically associated with the Dashboard Player.
- New command line options to start the a DBPLAY file automatically (--autostart) and in fullscreen.
- New button to select a TVM file in case several TVM files exist.
- Choose a TVM file in the Dashboard Player

Miscellaneous

- New command line option --noconsole to suppress any output in the command prompt.
- The variables that should be used in all projects of a TPT installation can be now defined in the "General" section of the Preferences dialog. Variables that should only apply to a specific TPT project are defined in a project-specific section in the Preferences dialog.
- Select "Enable auto line wrap" menu option in the Description view to automatically break lines.
- To open the most recently closed TPT file, select the new menu item "Open Most Recent".
- Multiple selection is now available in the project tree and the test cases tree.
- The names of project tree elements (testlet, variant, test case) that are created, copied or duplicated and have the same name as another project tree element, are automatically extended by an underscore and a number to ensure the uniqueness of the name.
- New search dialog: The local and global search are now in the same search dialog (CTRL+F).
- Text in the Description view and Documentation steps and fields can be formatted (font size, color, style).
- You can assign initial channel values for every single test case, variant, and group.
- New documentation available and searchable in TPT. New help view.
- Improved context sensitive help in TPT (F1).

- New Preferences dialog.
- The image resolution settings are directly available in the "Save image as" dialog.

TPT 9

Test Step List

- New Parallel step.
- New options "first" and "last" for the Compare step.
- Nested If-steps.
- Improved Signal Import Wizard in Step List.

Assessment

- Assessment variables can be declared as arrays.
- Comparison of array, matrix and structured signals supported in the Signal Comparison assesslet and Min/Max Comparison assesslet.
- New assessment function resampleOnChange().
- New report table of the Signal Comparison assesslet.

Incubation Features

- New Condition Tree Assesslet.
- Linked Declarations: share declarations between different files .

TASMO

- TASMO is now a regular feature and no longer incubation feature.
- Support of Stateflow models with specific coverage goals.
- Support of a static analysis of model input dependencies for more efficient test data search.
- Support of more blocks like "Compare to Zero " and "Compare to Constant".
- Matrix and vector signals are supported.
- TASMO will generate test data even if the model contains elements that are not specifically supported.

Platforms

- New Silver platform.
- Multi-core test execution possible for EXE and FUSION platforms.
- Support of internal signals in MATLAB/Simulink improved.
- Interface import and parameter exchange from Simulink Data Dictionaries are supported.
- Dataset format is now supported for Simulink signal logging.
- Signal logging in referenced models is supported.
- Support of AUTOSAR signals.

- Test of inner TargetLink subsystems possible .
- Support of Simulink Fast Restart for MATLAB 2015b or newer.

PLS UDE

- The PLS UDE FUSION node can handle all data types available in , including structs and user defined custom data types.
- Several instances of TPT and UDE can be executed in parallel.
- Progress logging of the PLS UDE FUSION node can be turned off to increase the performance.
- The PLS UDE FUSION node can read and write axis of maps and curves.

FUSION

- Use of two .NET dlls is now possible.
- The initial signal values may optionally be taken from the custom nodes dll instead of the default values from the Declaration Editor.

Equivalence classes

- Sets of equivalence classes can be created and managed in the new Equivalence Classes Editor.
- Assign equivalence classes to channels in a Test Step List.

Dashboard

- Dashboard script API extended.
- New Dashboard Player Guide.
- Undo / Redo for text fields in the Dashboard Configuration is now supported.
- Signals in widgets can be set via drag and drop operation from the Declaration Editor.
- Creation of new widgets by dragging signals onto an empty space in the dashboard.
- Dashboard file (* .dashboard) can be opened by a drag and drop operation into TPT.

Import/Export

- TPT can read signal data from .xls and .xlsx files .
- Import interface :
    - Check boxes to ignore differences of specific properties.
    - It is possible to import the rename information and the default values into a mapping instead of changing the values in the Declaration Editor.

Requirements

- In case a requirement is linked to test cases, the result of the test cases will influence the result of the requirement. This way inconclusive results are prevented if requirements are not directly linked to assesslets but to test cases.
- Re-import of test cases keeps the folder structure of tests.

- Requirements coverage information in the report extended.
- Improved display of the changes in the modifications table of the requirements.
- The modifications table in the Requirements view is accessible via a shortcut.

Miscellaneous

- Revised Type Editor.
- Revised FUSION documentation.
- New Test Modeling Quick Reference.
- New Jenkins plug-in documentation.
- Transitions can be prioritized (primary, secondary).
- Coverage information table is also shown in the Reclassification overview report.
- Test sets can be defined using a conditional expression.
- Integration with Test Rail (Gurock) for the exchange of test cases and test results.
- Local Search mechanism extended to more views and dialogs.
- Auto-completion in the Parameter tab.
- New examples
- Mil/Sil example for Simulink
- MATLAB platform for Stateflow
- Simulink Datastores
- Revised Type Editor.
- Revised FUSION documentation.
- New Test Modeling Quick Reference.
- New Jenkins plug-in documentation.
- Transitions can be prioritized (primary, secondary).
- Coverage information table is also shown in the Reclassification overview report.
- Test sets can be defined using a conditional expression.
- Integration with Test Rail (Gurock) for the exchange of test cases and test results.
- Local Search mechanism extended to more views and dialogs.
- Auto-completion in the Parameter tab.
- New examples
- Mil/Sil example for Simulink
- MATLAB platform for Stateflow
- Simulink Datastores

TPT 8

Test Step List

- Improvement of the performance when loading measurement data using the Import signal step.
- Import several signal from the same file using only one Import signal step.

Execution Configuration

- Assessments can be deactivated for parts of the Execution Configuration.

Assessment

- New completely revised browser based Assessment API for script assessment function. This API is accessible through the context sensitive help system (Ctrl+Spacebar while writing a script).
- The header area for assesslet in the "Assesslet Content" view has been redesigned and can be folded/unfolded on demand.
- New PT1 filter function.
- The report option "Report always (not only on error)" is by default deactivated for each assesslet.
- The functions TPT.always, TPT.exists, and TPT.never return FALSE if the expression is undefined for all points in time in the current context interval.
- The State Sequence Viewer known from the Test Data Viewer is now also available in the report.
- Previous assessment results were imported whithin signals when using import measurements, signal comparison or a script assesslets. Now, no result information is imported within signals, so imported signals cannot influence the actual test result.

- New completely revised browser based Assessment API for script assessment function. This API is accessible through the context sensitive help system (Ctrl+Spacebar while writing a script).
- The header area for assesslet in the "Assesslet Content" view has been redesigned and can be folded/unfolded on demand.
- New PT1 filter function.
- The report option "Report always (not only on error)" is by default deactivated for each assesslet.
- The functions TPT.always, TPT.exists, and TPT.never return FALSE if the expression is undefined for all points in time in the current context interval.
- The State Sequence Viewer known from the Test Data Viewer is now also available in the report.
- Previous assessment results were imported whithin signals when using import measurements, signal comparison or a script assesslets. Now, no result information is imported within signals, so imported signals cannot influence the actual test result.

Parameter

- The Parameter tab has one new column showing whether values have been defined in variants or subgroups.
- MATLAB: the parameter exchange of curves, maps and structs parameters is now supported. You have to use a custom script/function. See User Guide for detailed information.
- TASMO: Structural Test Case Generation for Simulink Models

- With the help of such automatically generated test data, TASMO supports you to improve your MATLAB/Simulink model test coverage in an early state of the development process.
- TASMO finds the minimum number of test cases to achieve a maximum of the structural coverage.
- You can choose among single criteria, decision coverage, condition coverage, and all coverage types together.
- The generated test data can also be used for back-to-back tests to check whether the model equals the compiled code.

Dashboard

- Automatic build of the executable for the Dashboard Player from the EXE platform or the MATLAB platform.
- Dashboard projects can be exported to a separate folder containing all necessary data to be run in the Dashboard Player.
- You can now record user actions in the Dashboard during the test execution. Afterwards, you can use this recording to generate a Test Step List.
- New widgets and updates.

Platforms

- New Assessment platform available for the assessment of measurement data without any test execution.
- New ASCET@FUSION platform.
- The step size for the FUSION can be set independently from the step size of the virtual machine.
- ASCET platforms can send and receive messages.

Import/Export

- MDF 4 supported for export and import.
- For CSV imports arrays, maps and curves are supported.
- Automatic Test Case Generation from Equivalence Classes.
- Automatic test case and variant generation from equivalence classes
- Light switch: OFF = [0,0], ON = [1,1], AUTO = [2,2]
- Before generating the test cases/variants, you can choose from different combinatorics, like for example, merging all generated data into one single variant.

Miscellaneous

- The new First Use Wizard can build a new fully configured MATLAB/Simulink or ASCET project for you in just a few steps.
- The new Test Case Status view gives you an overview over the current status of your test cases, as well as the revisions of a test case.

- New Test Case Details view with editable test case attributes.
- Signal preview can be switched to different platforms with different mappings in the step list view.
- Installer allows combined installation of a 32 bit and 64 bit process.
- Installation process possible without GUI setup wizard: silent mode.
- Improved display of structured signal names in the test data viewer.
- New context menu for test step list
- New local search dialog box available at certain views.
- Scaling removed from the Declaration Editor. Scaling only applies to the scaling flavor.
- Execution information in the project browser: test report, test data, and assessment data is now directly available in a dropdown menu for each test case.

TPT 7

Step list

- Direct definition step is now integrated in the channel step. You can choose to assign a value to the channel once or to assign a direct definition to the channel always.
- New while loop step.
- Compare steps are now shown in the report.
- Embedded signal step can now handle elements from structs, arrays and matrices.
- Ramp channel supports matrices and arrays.

Assessments

- Automatic testing for equivalence classes using an equivalence classes mapping.
- New tpt.hose options

Requirements

- Improved multiselection for requirements.
- Remove all "Needs to be reviewed" marks at once.

Dashboard

- Use a dashboard player to run dashboard files without a TPT license.
- Create a dashboard in a more detailed way using a script environment instead of a GUI.
- New widgets: gauge, multi state and selector.
- Graph widget can display several signals.
- Dashboard browser.
- Image widget can change rotation, brightness and transparency according to a signal.
- You can use internal signals to influence widgets without having to declare them in the declaration editor.
- Widgets can be now grouped.

- Quickly switch between panels.
- Change the visibility of widgets or groups.
- All widgets have notes / usage instructions that can be displayed during runtime.

Documentation

- New examples description tab. From there you can directly open the examples files.

Platforms

- dSPACE HiL supports MLIB/MTRACE API.
- Integration with CANoe.
- Array-/Matrix signals can be now exported to the MATLAB workspace using the M-Script assesslet.
- You can configure different ASCET versions (or edit/rename/delete them) in the ASCET plaftorm preferences dialog without changing any configuration.
- Integration with Concurrent HiL and VeriStand (National Instruments) platforms.

Report

- New advanced report options through the execution configuration.
- Pie charts in all reports.
- Scatter plots with scatter_plot = TPTReport.ScatterPlot function.

Miscellaneous

- New incubation features
- Version control using Subversion.
- Remote API to run the main features.
- New functions and controls for the test data viewer.
- Attachments can be added in the description window.
- 32bits/64bits versions in just one installer.
- System constants support.
- New command line option "--prefvar foo=bar" to set environment variables.
- New copy/paste features.
- Maps and curves can be more easily edited with a new wizard.
- Declaration editor: It is possible to enter the value in the value field directly in a binary format (only with integer data types).
- SMF 4.0 support.

TPT 6.1

Step List

- New step list shortcuts system.
- Step list groups cannot have their own test steps. Test steps are separated for every version. A multiple edit feature is planned for future.
- Multiple rows can be used using Alt+Enter in:
- Define channel/Direct Definition step
- Trigger Rule assesslet
- Transition conditions
- Ctrl+left click on a testlet step variant field navigates to the selected variant.
- Use TAB and Shift-TAB to navigate and F2 for editing a text field.
- Ctrl key hold down+left click to select several steps
- Shift+left click to select bigger blocks of steps
- To select all steps: Ctrl+A
- Preview of signals in Step List will be displayed when the step is selected. Select several steps to view their signal in the signal preview. The length of the preview can be adjusted using the "Duration" field.
- When the check box "wait until testlet terminates" is selected, the test execution remains in the Testlet until it terminates.
- Wait step: real time check box eliminated (VM works now without this option)
- New step list shortcuts system.
- Step list groups cannot have their own test steps. Test steps are separated for every version. A multiple edit feature is planned for future.
- Multiple rows can be used using Alt+Enter in:
- Define channel/Direct Definition step
- Trigger Rule assesslet
- Transition conditions
- Ctrl+left click on a testlet step variant field navigates to the selected variant.
- Use TAB and Shift-TAB to navigate and F2 for editing a text field.
- Ctrl key hold down+left click to select several steps
- Shift+left click to select bigger blocks of steps
- To select all steps: Ctrl+A
- Preview of signals in Step List will be displayed when the step is selected. Select several steps to view their signal in the signal preview. The length of the preview can be adjusted using the "Duration" field.
- When the check box "wait until testlet terminates" is selected, the test execution remains in the Testlet until it terminates.
- Wait step: real time check box eliminated (VM works now without this option)

Assessment

- The "Show state information" checkbox activates another column in the trigger rule report information, showing which states were active or used during the time interval checked by the trigger rule.

- New Assessment function TPT.hoseRelative(Signal(t),Signal_ref(t), 0, 0)
- Import measurement Assesslet
- External test case id can be accessed in Assessment
- Use the variable ${tpt.scenario.externalid} to address an external test case ID in an assessment script.
- Parameter filter in Signal comparison Assesslet: New parameter filter checkbox in the "Select Channels" dialog box.
- Signal Comparison tolerance options: You can now work with absolute, relative or lsb (least significant bit) tolerances.
- Signal Comparison Assesslet/Check reference file: Unused items for signal comparison can be removed for Signal Comparison when items do not appear in the reference measurement file.
- Use mapping from platform check box.

Requirements

- Links between requirements and variants of substates in step list are now covered and included in test case calculation.
- Each script Assesslet can rate every requirement using REQUIREMENTS.checked ("SPEC", result).
- The requirements are categorized according to their rating (success, failed, don't know)
- You can have a requirement which is linked but not called with REQUIREMENTS.checked ("SPEC", result): this link will show a "?" icon in the report, in the "not covered" category.

Incubation features

- Decide on the number of cores to be used for test execution, depending on your system architecture.
- To work with Testlink open source test management software is now possible.

Dashboard

- Dashboard is now accesible via the execution configuration dialog.
- Dashboard widget properties can be edited while in pause mode.
- New Dashboard script widget for customization. Multiple signal capability.
- New Dashboard Player as a stand alone application. License free but no edit functionality.

Documentation

- New "TPT Examples Description" documentation with new TPT examples (located in the examples folder).

Platforms

- Test cases can be now executed in ETAS LABCAR-AUTOMATION platform.
- ASCET platform command line option to generate test environment
- Generates the testframe for the ASCET project specified in the ASCET configuration.
- ASCET models can run at the FUSION platform.

- Concurrent Computer Corp. provides HiL simulators. test cases can be executed on SIMulation Workbench.

Miscellaneous

- New preferences Dialog Box.
- Saved perspectives under the "Manage Perspectives" menu.
- Equivalence classes can be assigned to declared channels using equivalence class flavor in the mapping editor.
- An equivalence class assesslet is available to check if all equivalence classes have been reached.
- Selected testlets in the graphical automaton view are highlighted in the project tree.
- You can select which parts of the model to analyze: if only the test model or the test model and the assessment will be analyzed.
- Execute only test cases that have been identified in previous test runs as unsuccessful.
- New button in the Signature tab to show all output signals (unused signals are shown too).
- A new test set can be defined based on the test results of the last test execution.
- Hovering over the progress bar in the build progress dialog gives detailed information about the overall status.
- Copy and Past has been significantly improved for automatons, variants and others. Also between different models.
- DCM files are specified as parameters set file in the execution configuration. These parameters will be taken for the respective test execution.
- Parameter can be imported to be used in the parameter tab using the respective button.
- Junctions and final nodes can have a name.
- Simple code coverage configuration for MATLAB Simulink platform.
- All graphical elements must have a unique name.

- New preferences Dialog Box.
- Saved perspectives under the "Manage Perspectives" menu.
- Equivalence classes can be assigned to declared channels using equivalence class flavor in the mapping editor.
- An equivalence class assesslet is available to check if all equivalence classes have been reached.
- Selected testlets in the graphical automaton view are highlighted in the project tree.
- You can select which parts of the model to analyze: if only the test model or the test model and the assessment will be analyzed.
- Execute only test cases that have been identified in previous test runs as unsuccessful.
- New button in the Signature tab to show all output signals (unused signals are shown too).
- A new test set can be defined based on the test results of the last test execution.

- Hovering over the progress bar in the build progress dialog gives detailed information about the overall status.
- Copy and Past has been significantly improved for automatons, variants and others. Also between different models.
- DCM files are specified as parameters set file in the execution configuration. These parameters will be taken for the respective test execution.
- Parameter can be imported to be used in the parameter tab using the respective button.
- Junctions and final nodes can have a name.
- Simple code coverage configuration for MATLAB Simulink platform.
- All graphical elements must have a unique name.

TPT 6.0

Test step lists

- The Testlet "Direct Definition" has been merged with "Step List" Testlet. Direct Definition doesn't exist any longer.
- Test Step Lists can be hierarchical.

Assessments and reporting

- The Trigger Rule Assesslet has been revised and simplified. The definition of intervals has been reduced to two.
- Multi-selection for the activation and deactivation of Assesslets is available.
- The Assessment editor comes with brace- and string highlighting.
- The REQUIREMENTS.checked() function has a new optional argument for the result of a check.
- The reporting of linked and checked requirements is more detailed.
- The Step-List report table has been extended with more information in case of failed checks.
- Assesslet-results can be seen in the data-viewer.
- Structured signals and arrays are displayed in the data-viewer.
- An XML overview report is available.

Platform

- PiL-testing and debugging is possible via UDE from PLS.
- A Simulink Realtime (XPC-Target) test execution platform is available.
- Simulink Alias-Types are supported.
- Multiple AUTOSAR components can be tested using EXE-platform. The test harness is generated automatically.
- The -VM-API has been changed for parameter exchange.

Miscellaneous

- The different editing modes of the model have been merged.
- Shared parts are of the test automaton are plotted bold in case of selecting several test cases.
- Single Testlets can be exported into an external file. Later it can be included as Testlet library.
- Structured signals are displayed in the data-viewer.
- Test execution and assessments can be executed on multiple CPU-cores.
- Tests can be executed in "Interactive mode" where tests are performed manually by the user.
- The different editing modes of the model have been merged.
- Shared parts are of the test automaton are plotted bold in case of selecting several test cases.
- Single Testlets can be exported into an external file. Later it can be included as Testlet library.
- Structured signals are displayed in the data-viewer.
- Test execution and assessments can be executed on multiple CPU-cores.
- Tests can be executed in "Interactive mode" where tests are performed manually by the user.

TPT 5.1

Test step lists

- New test step Define Channel / Direct definition test step behaves like Direct Definition testlet.
- New test step Embedded Signal test step is a graphical test data description or embedded measurement.
- New test step Import Signal links to signal data file.
- Variant support for test step lists: variants can inherit the test steps from a testlet. Modification of these inherited test steps is possible.

Installation

- TPT is available as 32-Bit as well as 64-Bit installer.

Miscellaneous

- TPT supports vector and matrix signals of Simulink Systems
- Import test data as variants: When test data is imported in order to create test cases, parameters are supported if the data contains parameter information in .mat files.
- MDF4 is supported.
- Signal mapping for structured data types is now supported.
- Signal declarations and mappings can be exported to Excel.
- The Dashboard is not an incubation feature any longer but can be licensed as product feature.
- The test result of the last test execution can be seen as an overlay icon in the project browser.
- A new search field is available in the project browser.

TPT 5.0

MATLAB/Simulink

- The MATLAB platform configuration has guided steps, making the set up more intuitive.
- Wizard to configure the standard use case.
- Import interface and parameter have been merged.
- Better support of modified interfaces. TPT analyses and supports changes in the interface of the SUT.
- Online and offline measurements are supported with less configuration effort.
- tpt_savelogs-Script combines the features of tpt_tl_savelogs and tpt_sl_savelogs.
- MATLAB enums are supported by TPT. Enums are declared in the new Type Editor.
- MATLAB test execution configuration "Check model I/O" is set to "off" by default.
- The MATLAB platform configuration has guided steps, making the set up more intuitive.
- Wizard to configure the standard use case.
- Import interface and parameter have been merged.
- Better support of modified interfaces. TPT analyses and supports changes in the interface of the SUT.
- Online and offline measurements are supported with less configuration effort.
- tpt_savelogs-Script combines the features of tpt_tl_savelogs and tpt_sl_savelogs.
- MATLAB enums are supported by TPT. Enums are declared in the new Type Editor.
- MATLAB test execution configuration "Check model I/O" is set to "off" by default.

Test modeling

- It is possible to access default values of channels in the first time step.
- It is possible to write a signal more than once in one time step, for example in parallel automatons. The last written wins.
- New sound emitter node. In step lists WAV-files can be executed for acoustic interaction.
- In step lists multi selection for copy and paste is possible.
- Function calls can be defined as server-functions and client-functions. Server functions are defined in TPT. This feature is not available for all platforms but can be used in exe-platform.
- In step lists, function calls can be embedded as individual test steps.

Assessments

- Manual reclassification after test execution is possible in the build progress dialog. Test results can be set to SUCCESS or FAILED manually.
- Completely restructured assessments with assesslet tree browser tab and assesslet content tab.
- Assesslets are no longer defined at the testlets and variants. Assesslets have a different activation mechanism.
- All assesslets come with a standardized header for activation, deactivation, assignment to testlets and variants and comments.
- The report is completely controlled by assesslets instead of report templates.
- New "Timeout" Assesslet.
- New "Check Log Entries" assesslet.

- Simplified regression testing. The "Signal Comparison" assesslet is extended by "Back2Back"-testing option where a reference platform can be specified.
- The "Trigger rule" assesslet is extended by an "Else"-branch.
- The "Trigger rule" assesslet check types have been extended by "Always true as long as if trigger condition is true".
- The "Import Measurement" assesslet declares assessment-signals automatically without manual Assessment-Signal declaration.
- The order of assesslet execution is independent of test cases. This may cause compatibility-issues regarding the execution order only in case of referenced testlets in conjunction with Assesslets at variants of the referenced testlet.
- The Assesslets can be placed into selected sections in the report.
- All assesslets results can be shown in the report.
- Manual reclassification after test execution is possible in the build progress dialog. Test results can be set to SUCCESS or FAILED manually.
- Completely restructured assessments with assesslet tree browser tab and assesslet content tab.
- Assesslets are no longer defined at the testlets and variants. Assesslets have a different activation mechanism.
- All assesslets come with a standardized header for activation, deactivation, assignment to testlets and variants and comments.
- The report is completely controlled by assesslets instead of report templates.
- New "Timeout" Assesslet.
- New "Check Log Entries" assesslet.
- Simplified regression testing. The "Signal Comparison" assesslet is extended by "Back2Back"-testing option where a reference platform can be specified.
- The "Trigger rule" assesslet is extended by an "Else"-branch.
- The "Trigger rule" assesslet check types have been extended by "Always true as long as if trigger condition is true".
- The "Import Measurement" assesslet declares assessment-signals automatically without manual Assessment-Signal declaration.
- The order of assesslet execution is independent of test cases. This may cause compatibility-issues regarding the execution order only in case of referenced testlets in conjunction with Assesslets at variants of the referenced testlet.
- The Assesslets can be placed into selected sections in the report.
- All assesslets results can be shown in the report.

Requirements

- Requirements can be linked to test cases and assessments.
- Requirements overview can be displayed in the report using the "Linked requirements" assesslet.

New incubation features

- Assesslet libraries can be managed and used in TPT.
- The Dashboard feature allows interactive observation and interaction using a graphical UI and user defined widgets.
- The Dashboard feature is available for all PC-based platforms such as FUSION, MATLAB, ASCET, EXE.
- The Distribute tests feature has been removed.

Report

- The report is completely controlled by assesslets instead of report templates.
- Attributes that are defined in the TPT Execution configuration dialog show up in the meta information section of the report.

Miscellaneous

- A second cursor is available in the data viewer for e.g. measuring differences.
- Testwell CTC++ coverage results and reports can be included into TPT reports.
- New import dialog for interface and parameter supports changes in the interface.
- At some platforms, TPT supports complex and structured data types (not available in MATLAB platform).
- The data types are defined in a new Type Editor.
- TPT files cannot be exported as previous version files any longer.
- For the Fusion platform, an MS .NET Assembly node is available.
- Windows environment variables can be easily accessed in TPT using ${VARIABLE}.

TPT 4.2

MATLAB/Simulink

- Vector indexing can be started with "0" or "1"
- New Simulink Measurement flavor
- New Simulink Object flavor
- New incubation features
- Slice Values feature for a more efficient storage of test data in memory

Assessments

- New TPT.readMeasurementRecord() function

Miscellaneous

- Prolog Assesslet can be activated/deactivated in Expert Mode
- Enumerations are supported in the report

- Minor improvements in GUI
- Prolog Assesslet can be activated/deactivated in Expert Mode
- Enumerations are supported in the report
- Minor improvements in GUI

TPT 4.1

MATLAB/Simulink

- New command line option "--closeMatlabAfterTestrun"
- New Enumeration data type supported
- New $-variable for the SUT-block in Simulink ${tpt.matlab.block.name}
- Rate transition blocks can be inserted automatically during test frame generation
- Workaround for MATLAB bug problems with "cvsim" in MATLAB 2010b
- Analyse Interface of model reference systems

New incubation features

- Test management feature

Assessments

- New Assessment summary view
- Function "PyRecord.includeUnits()" is deprecated

Requirements

- New user interface
- Requirements can be linked to variants of Testlets and Assesslets
- $-Variables ${tpt.scenario.precondition}, ${tpt.scenario.passcondition} and ${tpt.scenario.specification} can be used in both Assessment and Report

Miscellaneous

- Minor changes in default perspective
- ETAS Inca-Flavor in Mapping is renamed to Measurement-Flavor
- Batch test supports relative path
- Step Lists: Individual test steps can be activated/deactivated
- Test cases can be hidden in project browser
- Modularization of TPT-file allows distributed work
- Faster INCA coupling with other API "RCI2"
- Minor changes in default perspective
- ETAS Inca-Flavor in Mapping is renamed to Measurement-Flavor
- Batch test supports relative path

- Step Lists: Individual test steps can be activated/deactivated
- Test cases can be hidden in project browser
- Modularization of TPT-file allows distributed work
- Faster INCA coupling with other API "RCI2"

TPT 4.0

GUI

- TPT release 4.0 comes with a new user interface in a dockable framework.
- The UI has a new items menu structure.
- The testlet tree and the scenario tree are merged into a single browser.
- Scenarios are called variants or test cases respectively.
- Shortcut keys for navigation BACK, FORWARD, UP using Alt+LEFT, Alt+RIGHT, Alt+UP.
- Shortcut keys in the project browser for new variant/testcase and new group using SHIFT+INSERT and SHIFT+CTRL+INSERT.

MATLAB/Simulink

- In Simulink signals-data can be measured and accessed during test execution. Also TPT can react on these signals during test execution.
- For simulation performance on MATLAB platform local parameter changes during test execution will not be reported in test results.

New incubation features

- Tests can run in parallel on different computers (less time consuming and better resources management).
- Requirements and traceability can be displayed in the test report.
- "REQUIREMENTS.report()" in the assessment script reports covered requirements in the testcase-report.
- A TPT Library concept.

Miscellaneous

- Improvement in Array parameter display in the data viewer.
- Test step lists: Complex expressions like b+c >= d+e are supported in if-statements.
- Test step lists: Complex expressions are supported for wait-steps.
- Add possibility to assign custom links to signal graphs in TPT.
- Better support of the ASCET platform.
- Data type Strings are supported.
- The TPT support can now be accessed online via http://support.piketec.com or via Menu | Help | About TPT.

- Improvement in Array parameter display in the data viewer.
- Test step lists: Complex expressions like b+c >= d+e are supported in if-statements.
- Test step lists: Complex expressions are supported for wait-steps.
- Add possibility to assign custom links to signal graphs in TPT.
- Better support of the ASCET platform.
- Data type Strings are supported.
- The TPT support can now be accessed online via http://support.piketec.com or via Menu | Help | About TPT.

TPT 3.4.3

Undo/Redo

- TPT now supports undo and redo. Undo erases the last change done to the document reverting it to an older state.

Import measurements as scenarios

- Measurement or signal data files can be imported as scenarios in direct definitions. The 'Import measurements as scenarios' feature searches for a given pattern through the file system and generates a scenario and a test case for every file found.

Test case generation

- The test case generation is extended by the path combination through the automaton(s). Now combinations of path-variants, testlet-variants (state-variants) and transition-variants can be combined automatically.

Overview report without test execution and assessment

- If test cases have been executed and assessed before and the data is still available, the overview report can be generated individually. Thus individual tests can be executed separately but the report can be generated for all tests. The overview report generation is based on a test-case individual XML test result file.

MATLAB platform

- A new assesslet type where M-scripts can be used for the test assessment.
- MATLAB-Warnings from the MATLAB console can appear in the test report.
- The code coverage settings dialog in MATLAB can be opened and edited from TPT.
- If the SUT contains libraries the path to the libraries may be added to the testframe model initalisation script.

Miscellaneous

- Groups of parameter array, map and curve indices can now be accessed in a compact way. e.g array [1:3] = ...
- The import measurement assesslet supports chunks of measurement data if there is a gap between samples larger than a specified time.
- The message box in step lists has a termination condition.
- The size of an array, a map or a curve can be accessed. e.g. array:size

TPT 3.4.2

Removed Platform "Linked Config"

- The platform configuration type "Linked Config" has been removed. Each existing configuration of this type will automatically be converted to a simple platform configuration of the same type as the configuration that has been referenced by the corresponding "Linked Config".

Improvements to Extensions for CAN

- The extensions for configuring and running tests using the a CAN bus now provides:
  - More details about a message and its signals.
  - An easy way to define the send rate of each message.
  - Support for the definition and configuration of message counter signals.
  - Boosted Performance for Viewing Irregularly Sampled Signals
  - The performance for viewing irregularly sampled signals with the test data viewer has been improved massively.

TPT 3.4.1

- It is possible to mark one scenario as default in each testlet. This scenario will be selected as default when new scenarios higher in the hierarchy are selected.
- Instead of using the simulation time t, a system time can be accessed.

MATLAB platform

- MATLAB 64 Bit support.
- MATLAB 7.1 and higher is supported.
- Simulink Signal Object support with data types.
- Optionally parameter can be imported from MATLAB Workspace or from the system under test.
- MATLAB matrices can be imported as maps into TPT.
- The model I/O can be checked before test execution.

Test step list

- The system time can optionally been used in wait steps
- Wait for value and compare steps have a new column where comments can be added that result in report entries.
- Step lists may be shown in reports or not.

Dashboard

- A dashboard where the user can interactively view and change signals and parameter during test execution is available for the Fusion platform.

External signal data

- When import test data into direct definitions the user can optionally insert a termination condition if the end of the signal is reached or can set a value after the last sample that will be used for continuation after the measurement ended.

Assesslets

- Prolog: The Check timeout checks if the timeout happened in a test case.
- Import measurements: A mapping gan be given in order to map signal names of the measurement data file.
- Import measurements: Missing measurements can be generated.
- Import measurements: A suffix at a measurement signal can be ignored.
- Import measurements: The time synchronsiation methods have been extended.
- Bound check: The minimum and maximum values of the MIN/MAX assessment can be exported to the report.
- Signal comparison: The Signal comparison can now ignore times where the signals do not overlap.