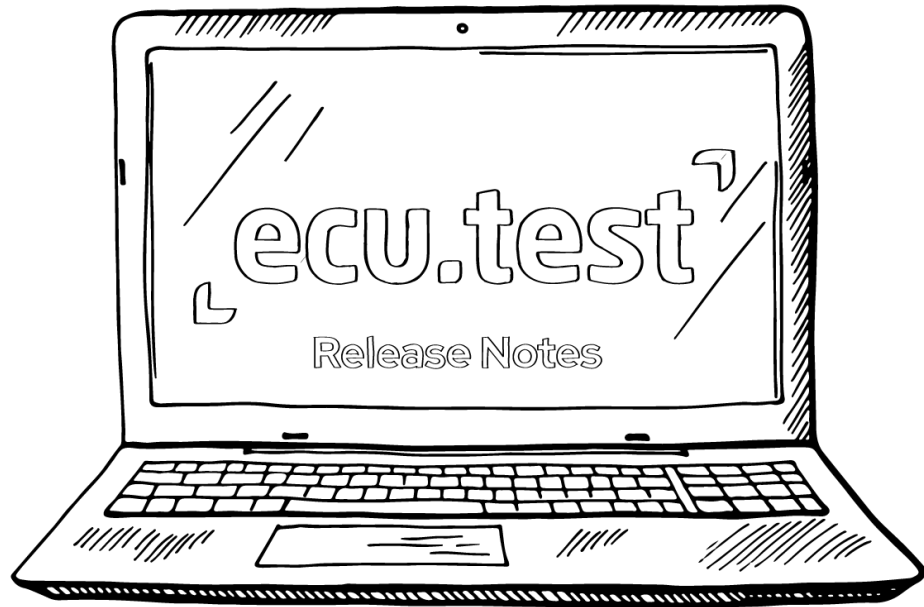




Release Notes

ecu.test 2026.3

trace.check 2026.3

Datum: 29.09.2026

© 2026 tracetronic GmbH

tracetronic GmbH
Stuttgarter Str. 3
01189 Dresden
www.tracetronic.de

Inhalt

Überblick	1
1 Highlights in ecu.test 2026.3	2
2 ecu.test extras	11
2.1 ecu.test <i>calibration</i>	11
2.2 ecu.test <i>drive</i>	12
2.3 ecu.test <i>lab</i>	13
2.4 trace.xplorer	16
3 Usability	19
4 Testaspekte	23
4.1 Kommunikation	23
4.2 MiL	24
4.3 Multimedia	25
4.4 Testmanagement	27
4.5 Traceanalyse	28
4.6 Weitere Testaspekte	29
5 Versionen und Schnittstellen	31
5.1 Neue Tools und Versionen	31
5.2 APIs	31
5.2.1 Command Line Interface	31
5.2.2 Custom Checks	32
5.2.3 Object API	32
5.2.4 REST API	33
5.2.5 Test Management API	33
6 Abkündigungen	34
6.1 Abkündigungen und Inkompatibilitäten in dieser Version	34
6.2 Abkündigungen in zukünftigen Versionen	37

Überblick

ecu.test 2026.3 nimmt einige der zeitintensivsten Aufgaben im Testalltag ins Visier: repetitive Test- und Projektarbeit, aufwändiges Zusammenstellen von Multimedia-Testabläufen, Versionskonflikte in geteilten Workspaces.

Der **ecu.test agent** unterstützt jetzt durchgängig bei der Erstellung und Anpassung von Testfällen sowie Traceanalysen und beantwortet zusätzlich Fragen zur Bedienung von **ecu.test** – all das einfach per Chat. Dabei lernt er aus bestehenden Packages und berücksichtigt auch kundenspezifische Regeln.

Vier weitere Highlights ergänzen das Release:

- **Agentischer Multimedia-Testschritt:** plant und führt komplexe Multimedia-Testabläufe per frei formulierten Prompts aus.
- **Lokal entwickeln, remote ausführen:** Testfälle auf dem Rechner erstellen und mit nur wenigen Klicks auf dem Ziel-Prüfstand ausführen.
- **Workspace-Kompatibilität:** ermöglicht parallelen Betrieb verschiedener **ecu.test**-Versionen auf demselben Workspace.
- **Neue Python-Bibliothek:** erleichtert die Entwicklung eigener Erweiterungen in externen IDEs, wie VS Code, PyCharm & Co.

ecu.test extras

- **calibration:** registriert Messgrößen testfallübergreifend, unterstützt XAM-Messlisten.
- **drive:** zeigt jetzt auch Auswahllisten, Eingabedialoge und Restwartezeiten an.
- **lab:** direkter Zugriff auf Signale und Parameter, plus Fehleranzeige direkt am Widget.
- **trace.xplorer:** mit überarbeiteter Diagrammansicht und Bildexport.

Performance im Alltag

- **EasyInsert:** bessere Übersicht und Sortierung für Bibliothekspaces
- **Selektive Projektausführung:** für jede Ausführungsform verfügbar
- **Neues Changelog-Format:** modern, durchsuchbar und filterbar

Außerdem gibt es Neuerungen u. a. zu Kommunikation, MiL oder Testmanagement. Details folgen auf den nächsten Seiten. Viel Spaß beim Weiterlesen!

Hinweis: Die Icons zeigen, für welches Produkt ein Thema relevant ist:  **ecu.test**  **trace.check**

1 Highlights in ecu.test 2026.3

ecu.test agent chat



Beim Erstellen von Testschritten steckt viel Kontextwissen implizit im Projekt und muss bei jedem Schritt aufs Neue manuell berücksichtigt werden. Das kostet Zeit, erzeugt Inkonsistenzen und ist fehleranfällig.

Mit dem **ecu.test agent Chat** zieht KI-Unterstützung direkt in **ecu.test** ein und setzt genau da an. Er erledigt repetitive Arbeit beim Erstellen von Testschritten, sorgt für konsistente Ergebnisse und beantwortet Fragen rund um **ecu.test**. So bleibt mehr Zeit für das Wesentliche: die eigentliche Testlogik.

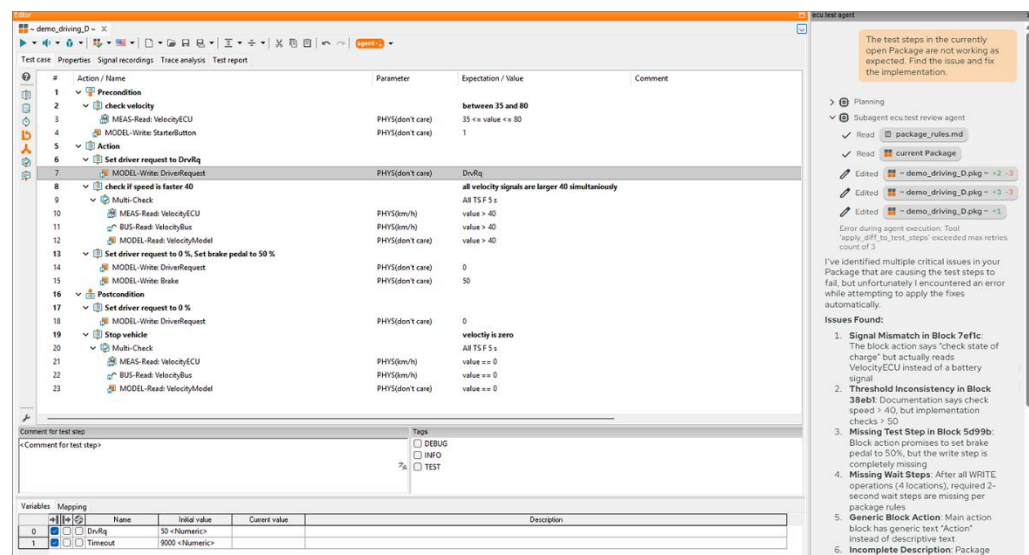


Abbildung 1: ecu.test agent chat

KI-gestützte Testschrittgenerierung

Der Agent unterstützt bei der Erstellung von Testfallschritten und bei der Generierung von Traceanalyse-Schritten. In beiden Fällen analysiert er den vorhandenen Projektkontext automatisch und berücksichtigt dabei:

- Signal-Mappings, Tool- und Port-Jobs sowie Package-Mappings
- Package-Variablen und globale Konstanten
- Die Block-Beschreibung als Grundlage für passende Testschritte

Da der Agent direkt auf das gesamte Projekt-Setup zugreift, passen die generierten Schritte zur bestehenden Testumgebung und sind in der Regel direkt einsatzbereit. Manuelle Fehler bei der Auswahl von Signalen, Mappings oder Package-Aufrufen werden so erheblich reduziert.

Hinweis: Die Unterstützung der Traceanalyse befindet sich in aktiver Weiterentwicklung und ist derzeit als Beta-Funktion verfügbar. Der Funktionsumfang wird in kommenden Versionen kontinuierlich erweitert.

Konsistenz durch Referenzimplementierungen

Der Agent lernt aus ähnlichen Blöcken in bestehenden Packages. Bewährte Testmuster aus dem eigenen Projekt fließen in die Generierung ein und werden auf neue Testfälle übertragen. So entstehen konsistente und standardisiert wiederkehrende Lösungen, ohne dass Teams aktiv eingreifen müssen.

Anpassbare Generierung

Die Generierung lässt sich flexibel auf projektspezifische Anforderungen einstellen:

- Konfigurierbare Filter steuern, wie viele Elemente aus den einzelnen Kategorien einfließen.
- Custom Instructions ermöglichen die Anpassung an projektspezifische Namenskonventionen, Stilregeln und individuelle Präferenzen.

ecu.test-Hilfe direkt im Chat

Unsicher, wie ein bestimmtes Feature funktioniert oder wie ein Problem zu lösen ist? Einfach im Chat fragen – der Agent hilft weiter, ohne dass die Umgebung verlassen werden muss.

Agentischer Multimedia Testschritt



Mit dem neuen Testschritt **Agentic Multimedia** lassen sich komplexe Multimedia-Testabläufe anhand eines frei formulierten Prompts selbstständig planen, ausführen und bewerten.

Der Testschritt steht dazu mit allen in **ecu.test** vorhandenen Multimedia-Tools sowie selbst erstellten Usertools zur Verfügung. Das für die Multimedia-Verarbeitung verwendete Modell wird zentral in den **Workspace-Einstellungen** über die **Multimedia-Einstellungen** gewählt.

Zur Laufzeit analysiert der Testschritt den Prompt, leitet daraus geeignete Aktionsfolgen ab und führt diese auf dem verknüpften Multimedia-Tool aus (siehe Beispiel).

Optional können bereits ermittelte Prompt-Analysen, Ausführungsschritte und Kontextinformationen über einen Cache wiederverwendet werden.

Dadurch lassen sich wiederholte Ausführungen beschleunigen und unnötige Modellanfragen reduzieren. Über die Testschrittparameter kann gesteuert werden,

- ob der Cache verwendet wird,
- wie viele Modellanfragen maximal zulässig sind und
- wie lange nach Aktionen gewartet wird.

Mit aktiviertem Self-Healing können fehlerhafte Cache-Einträge bei Bedarf verworfen und der Ablauf erneut ohne Wiederverwendung ausgeführt werden.

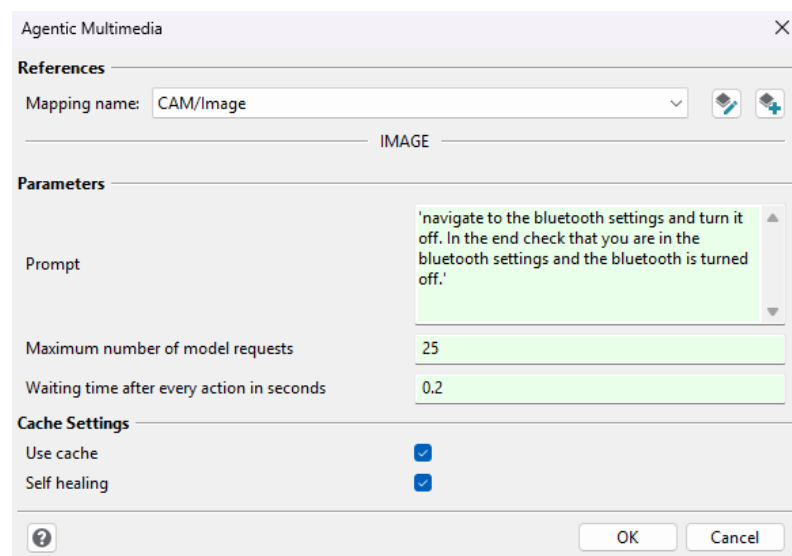


Abbildung 2: Testschritt: Agentic Multimedia mit dem Prompt für das Beispiel

Beispiel:

Der Testschritt (siehe Abbildung 9) enthält den Prompt: „Geh in die Bluetooth-Einstellungen und schalte sie aus. Stelle sicher, dass du dich in den Bluetooth-Einstellungen befindest und Bluetooth ausgeschaltet ist.“

(„navigate to the bluetooth settings and turn it off. In the end, check that you are in the bluetooth settings and the bluetooth is turned off.“)

Der aktuelle Bildschirminhalt des verbundenen System-under-Test (SUT) wird ausgelesen und analysiert. Der Agent ermittelt die nächste Aktion und führt sie aus – Schritt für Schritt, bis das Ziel erreicht ist. In diesem Fall navigiert er vollständig selbstständig zum Menü **Einstellungen** (Schritt 1), wechselt in das Menü **Verbundene Geräte** (Schritt 2) und deaktiviert **Bluetooth** (Schritt 3), ohne fest codierte Koordinaten oder manuelle Testschritte.

Im Testreport wird jeder Zwischenschritt mit dem Reasoning des Modells und einem Abbild des SUT dokumentiert.

Agentic Multimedia		navigate to the bluetooth settings and turn it off. In the end check that you are in the bluetooth settings and the bluetooth is turned off' SUCCESS
1	Image-Read: CAM/Image Comment Image-Touch input: CAM/Image Image-Read: CAM/Image	Image(Result image): 720x960 I'll help you navigate to the Bluetooth settings and turn it off. Let me start by opening the Settings app. Tap: (409px, 597px) Image(Result image): 720x960
2	Image-Read: CAM/Image Image-Touch input: CAM/Image Image-Read: CAM/Image	Good! The Settings app has opened. I can see the "Connected devices" option which mentions "Bluetooth, pairing". Let me tap o... Tap: (360px, 271px) Image(Result image): 720x960
3	Image-Read: CAM/Image Image-Touch input: CAM/Image Image-Read: CAM/Image Comment	Perfect! I'm now in the "Connected devices" screen and I can see the Bluetooth toggle switch at the top. The toggle is currently O... Tap: (505px, 221px) Image(Result image): 720x960 Excellent! I have successfully completed the goal. The Bluetooth has been turned off...

Abbildung 3: Testreport mit agentisch dokumentierten Zwischenschritten während der Testausführung

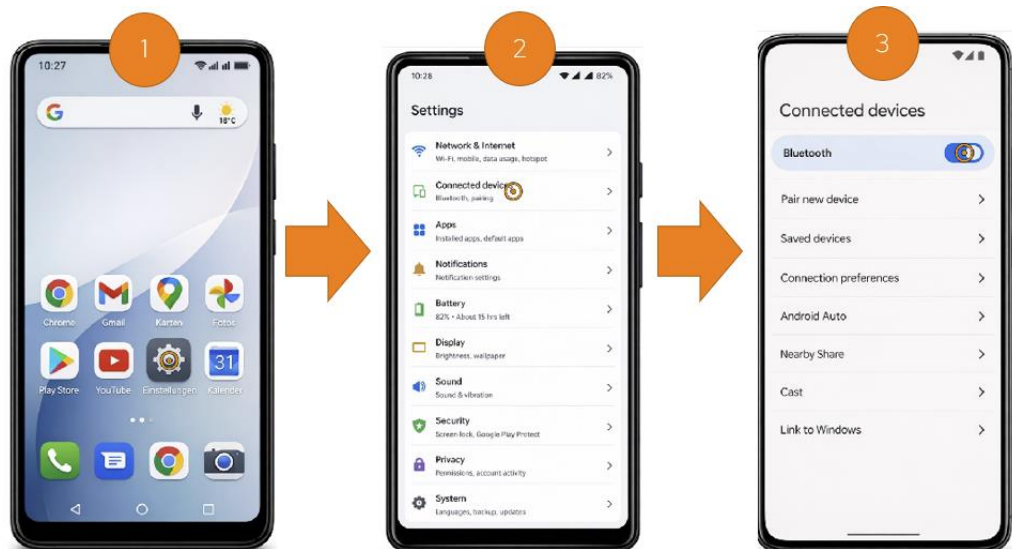


Abbildung 4: Abbild des SUT im Testreport während der Ausführung des Testschritts

Am Ende bewertet der Testschritt das Ergebnis mit SUCCESS oder FAILED und gibt eine Zusammenfassung aus, die das Ergebnis und die Begründung des Modells enthält.

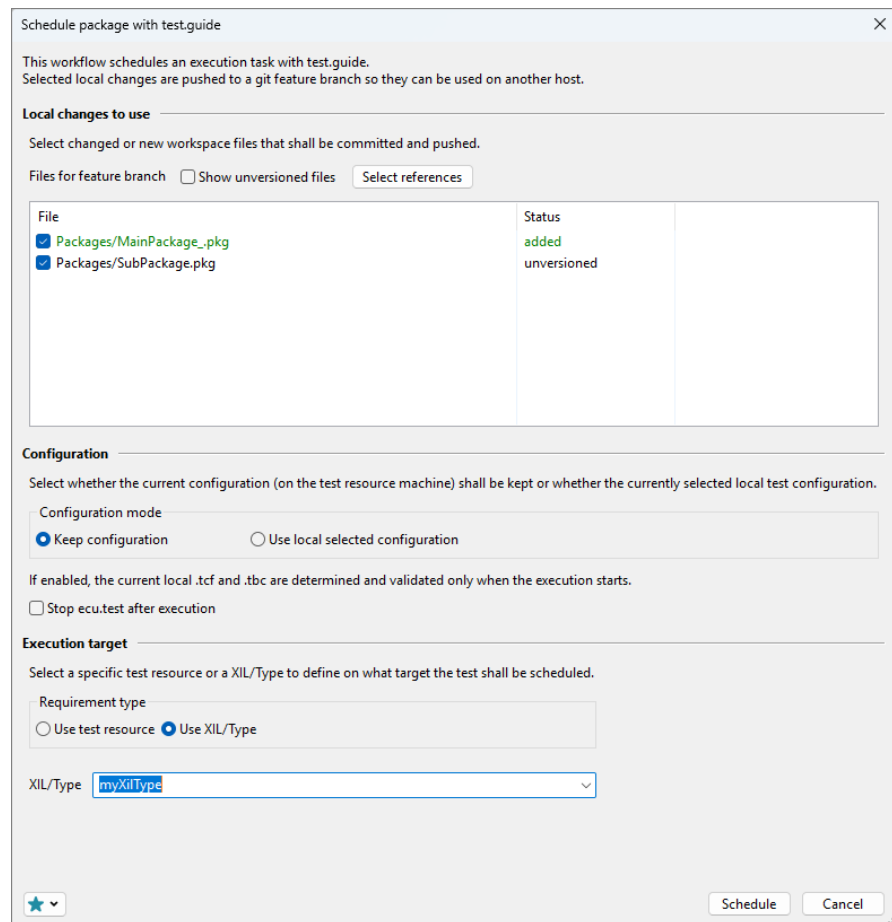
Testfälle lokal entwickeln und remote über one:cx/test.guide ausführen



Prüfplätze sind ein knappes Gut und entfalten ihren größten Mehrwert, wenn sie nahezu durchgehend Tests ausführen. Die manuelle Entwicklung und Anpassung neuer Testfälle blockiert diese Ressourcen jedoch häufig und reduziert dadurch die Gesamtauslastung.

Mit dem Feature Ausführung via test.guide werden Entwicklung und Ausführung zu einem durchgängigen Workflow verbunden.

- **Lokal entwickeln, remote ausführen:** Testfälle werden am Arbeitsplatz erstellt und mit wenigen Klicks auf dem Zielprüfplatz ausgeführt.
- **Intelligente Auswahl:** Im Ausführungsdialog werden Package-Änderungen, Zielkonfiguration und passendes Zielsystem gezielt ausgewählt.



Schedule package with test.guide

This workflow schedules an execution task with test.guide.
Selected local changes are pushed to a git feature branch so they can be used on another host.

Local changes to use

Select changed or new workspace files that shall be committed and pushed.

Files for feature branch ☐ Show unversioned files

File	Status
☒ Packages/MainPackage_pkg	added
☒ Packages/SubPackage.pkg	unversioned

Configuration

Select whether the current configuration (on the test resource machine) shall be kept or whether the currently selected local test configuration.

Configuration mode
☒ Keep configuration ☐ Use local selected configuration

If enabled, the current local .tcf and .tbc are determined and validated only when the execution starts.
☐ Stop ecu.test after execution

Execution target

Select a specific test resource or a XIL/Type to define on what target the test shall be scheduled.

Requirement type
☐ Use test resource ☒ Use XIL/Type

XIL/Type

Abbildung 5: Dialog zur Planung der Ausführung via one:cx/test.guide

- **Automatisierte Queue:** Änderungen werden automatisch per Git übertragen und der Test in die Prüfplatz-Warteschlange eingereiht.

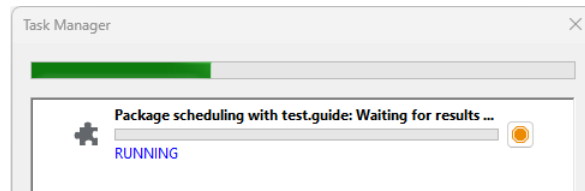


Abbildung 6: Fortschrittsanzeige der Ausführung

- **Maximale Auslastung:** Der Prüfplatz wird während der Entwicklung nicht belegt und wird ausschließlich für die eigentliche Testausführung belegt – optimal für die parallele Nutzung und CI-Automatisierung.

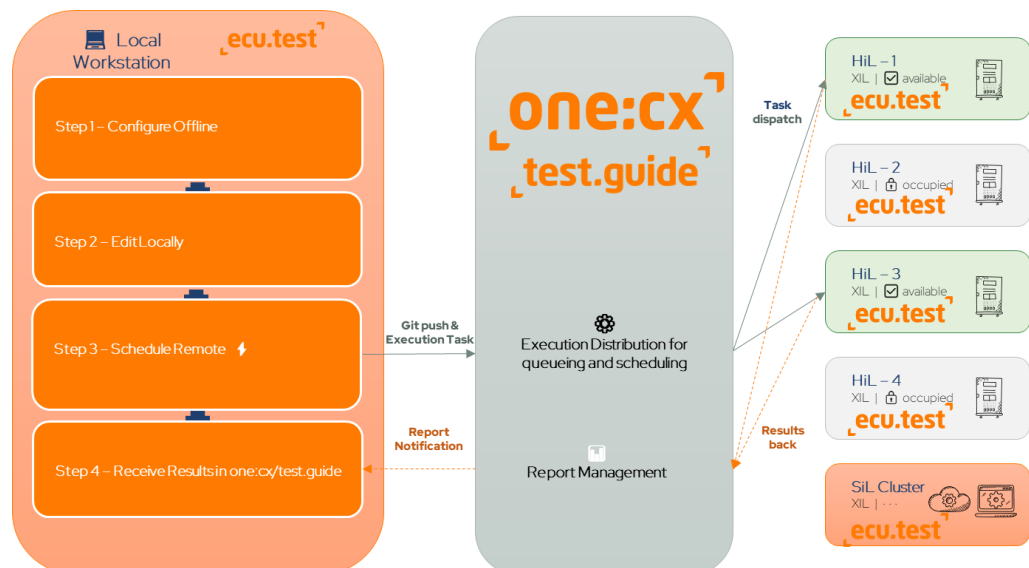


Abbildung 7: Schematische Darstellung des gesamten Workflows

Workspace-Kompatibilität zwischen verschiedenen ecu.test-Versionen



Ab **ecu.test** 2026.3 läuft der parallele Betrieb verschiedener **ecu.test**-Versionen auf demselben Workspace deutlich reibungsloser.

Das hilft besonders in Umgebungen, in denen mehrere Teams einen gemeinsamen Workspace nutzen, etwa Bibliotheksworkspaces, und bisher gezwungen waren, Updates gleichzeitig und organisationsübergreifend auszurollen.

In den **Workspace-Einstellungen** lässt sich nun eine **ecu.test**-Version festlegen, mit der die Artefakte im Workspace kompatibel bleiben sollen. Beim Speichern prüft **ecu.test** die Artefakte gegen diese Zielversion und verhindert so, dass sie versehentlich für ältere Versionen inkompatibel werden.

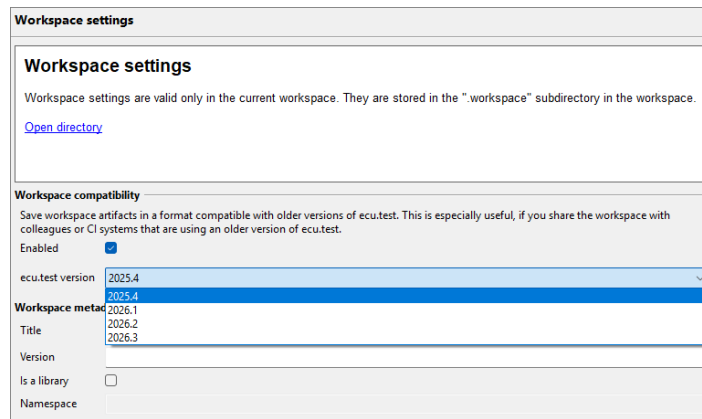


Abbildung 8: Kompatibilitätsmodus in den Einstellungen konfigurieren

Ein Dialog weist darauf hin, falls sich eine Inkompatibilität nicht vermeiden lassen würde.

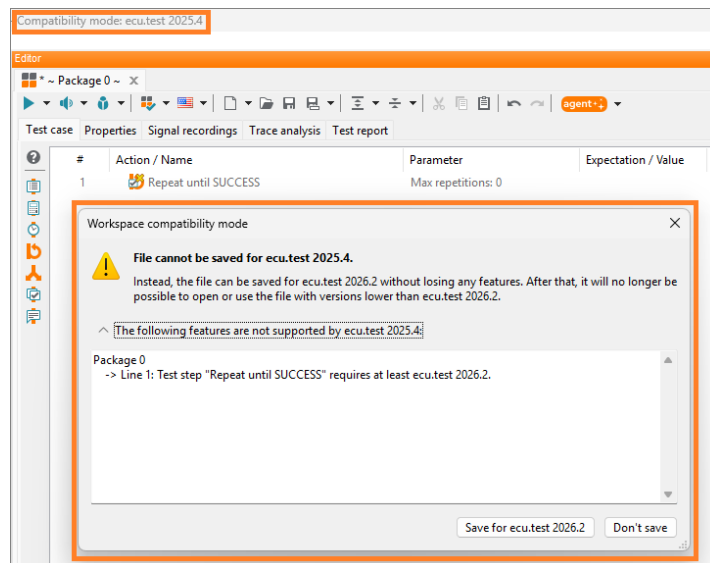


Abbildung 9: Prüfung im Kompatibilitätsmodus, ob der Testfall in älteren ecu.test-Versionen kompatibel bleibt

Zusätzlich zeigt der Workspace Explorer direkt an, für welche **ecu.test**-Version das Artefakt oder die Bibliothek kompatibel sind.

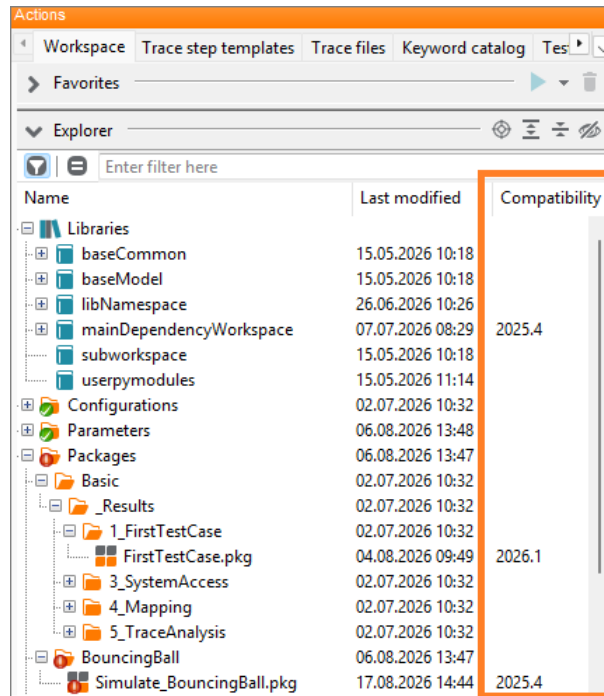


Abbildung 10: Anzeige der ecu.test-Version im Workspace-Explorer, mit der ein Artefakt kompatibel ist

Weitere Ergänzungen:

- Eine neue Migrationshilfe (siehe Hilfe – Migrationshilfe) aktualisiert einen kompletten Workspace per Wizard auf eine neue **ecu.test**-Version.

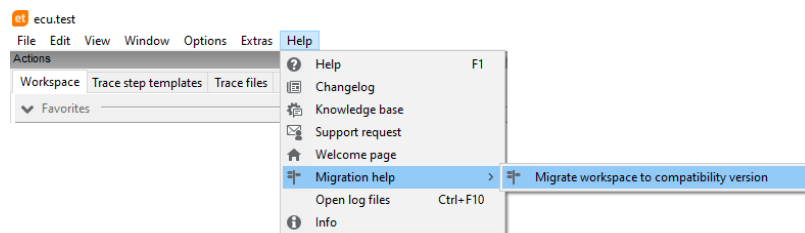


Abbildung 11: Öffnen der Migrationshilfe

Das manuelle Öffnen und Speichern jedes einzelnen Artefakts entfallen damit.

- Die Object API wurde passend ergänzt, sodass auch das Speichern von Packages über die API die Kompatibilitätseinstellung berücksichtigt und das Feature durchgängig in automatisierten Abläufen nutzbar ist.
- In der [Anwenderhilfe](#) sind die Einrichtung, empfohlene Abläufe für den Versionswechsel, den Umgang mit gemeinsam genutzten Workspaces sowie Einschränkungen dokumentiert.

ecu.test Interface Python Bibliothek



Die neue **ecu.test** Interface Python Bibliothek stellt die öffentlichen Python-Schnittstellen für die Entwicklung eigener **ecu.test**-Erweiterungen bereit. Dadurch wird es deutlich einfacher, diese Erweiterungen in einer externen IDE wie VS Code oder PyCharm zu entwickeln.

Die Bibliothek wird als Python-Wheel bereitgestellt und enthält neben der API-Dokumentation auch Beispiele für die Verwendung.

Moderne IDEs ermöglichen so Autovervollständigung einschließlich Inline-Hilfe. Außerdem können dadurch AI-Coding-Assistenten wie GitHub Copilot oder Claude Code bei der Erstellung, Änderung oder Absicherung von ecu.test-Erweiterungen besser unterstützen.

Die Verwendung wird ausführlich in der [ecu.test-Hilfe](#) beschrieben und umfasst derzeit folgende API-Bestandteile:

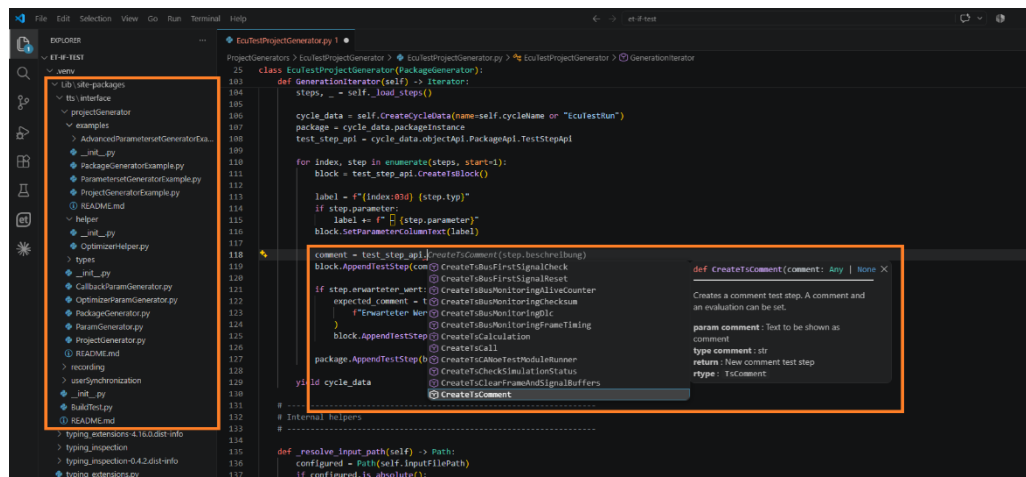


Abbildung 12: Verwendung der ecu.test-Interfaces in VS Code

2 ecu.test extras

2.1 ecu.test calibration

XAM-Messlisten und Registrierung von Messgrößen



Das Registrieren einer großen Anzahl von Messgrößen kann unter Umständen einige Zeit dauern. Werden dieselben Messgrößen in mehreren aufeinanderfolgenden Testfällen verwendet, lassen sie sich jetzt nach Testfallende registriert halten – eine erneute Registrierung zu Beginn des nächsten Testfalls entfällt damit.

Dieses Vorgehen bietet sich insbesondere an, wenn bei **Connection setup** "At start of configuration" gewählt wird. Die Vorbereitungszeit bei der Testfallausführung reduziert sich dadurch deutlich.

Zur einfacheren Verwaltung großer Messlisten, kann die Messliste, die für die Aufnahme verwendet werden soll, als globale Mappingdatei (XAM) definiert werden.

Diese Datei lässt sich beispielsweise aus einer A2L-Datei über die Object API generieren.

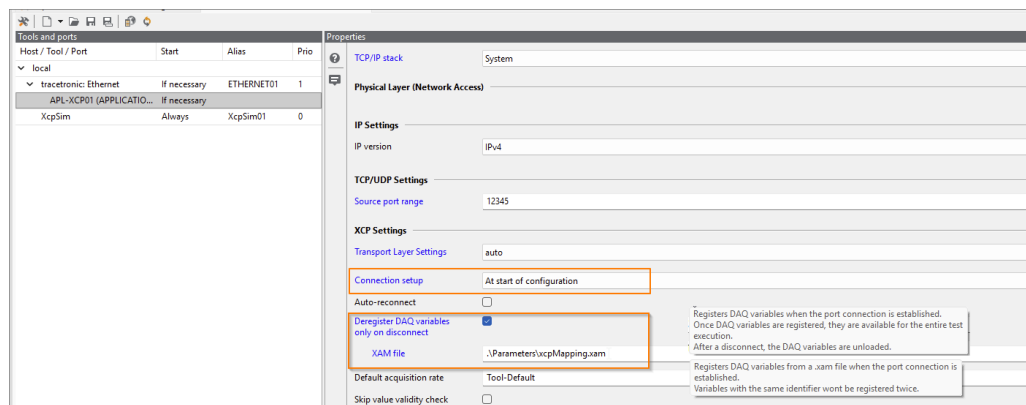


Abbildung 13: XCP-Eigenschaften festlegen

2.2 ecu.test drive

Unterstützung weiterer Testschritte



ecu.test drive visualisiert nun zusätzlich die folgenden Testschritte:

- **Auswahlliste:** Auswahl einer von mehreren vordefinierten Optionen
- **Eingabedialog:** Einzeilige Texteingabe
- **Wait:** Zeigt die Restwartezeit

Wenn mit Erwartung und Zeitoption versehen auch:

- **Job-Call**
- **Variable Referenz lesen**
- **Package-Call**
- **Service ausführen**

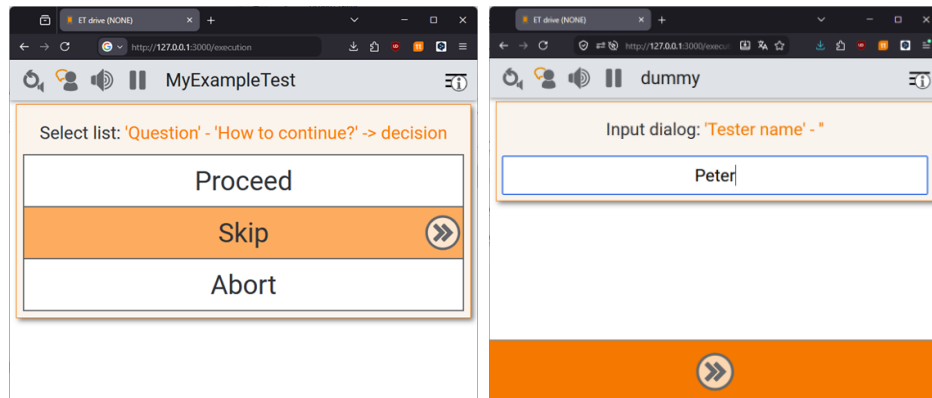


Abbildung 14: Testschritte in ecu.test drive

Kleine UI- und Workflowverbesserungen

- Der Name des Testfalls wird nun auch während der Testausführung in der Titelleiste angezeigt.
- Vorlesen der Prefixe "Aktion" und "Erwartung" ist nun in *drive* konfigurierbar.

- Um zu verhindern, dass eine Projektkomponente ungewollt nochmals ausgeführt wird, wurde die Vorauswahl bei der Anzeige eines bereits gestarteten Projekts angepasst. Außerdem wurden ein Warnhinweis und eine Rückfrage hinzugefügt.

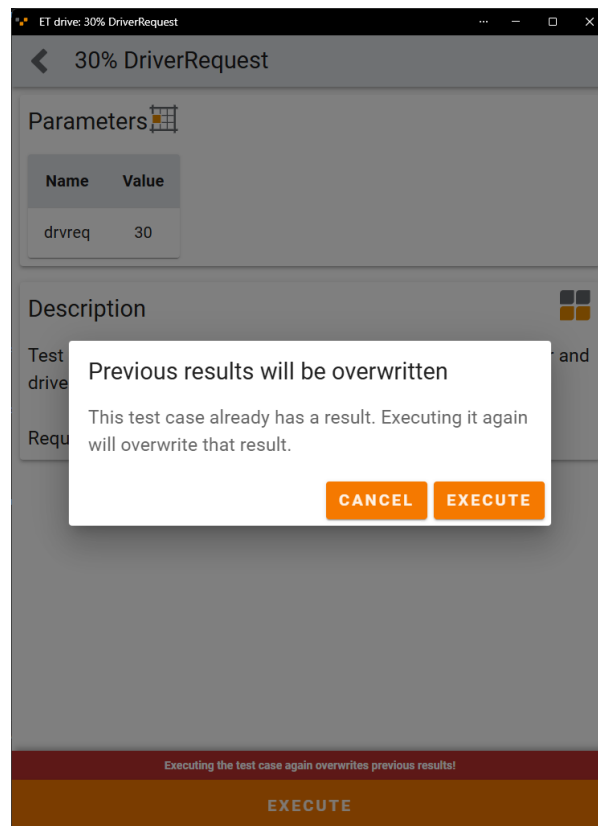


Abbildung 15: Warnhinweis und Rückfrage

2.3 ecu.test lab

Mehrfachauswahl von Widgets

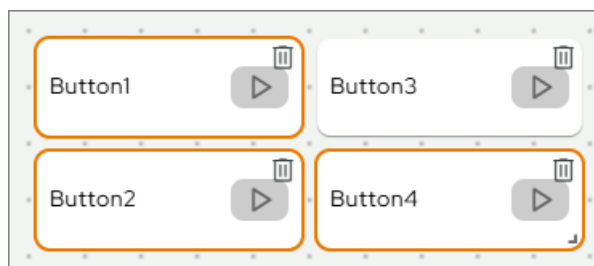


Abbildung 16: Mehrfachauswahl von Widgets

Mit gedrückter **Strg**-Taste können nun mehrere Widgets gleichzeitig selektiert werden, um diese gleichzeitig kopieren, ausschneiden oder löschen zu können.

Zwei Seiten nebeneinander darstellen



Es ist nun möglich, zwei Seiten in einem beliebig veränderbaren Seitenverhältnis nebeneinander dazustellen. Ergänzt um die Funktion, direkte Seitenreferenzen einzufügen, verschafft dies deutlich mehr Möglichkeiten in der Gestaltung und vermeidet die doppelte Erstellung von Seiten.

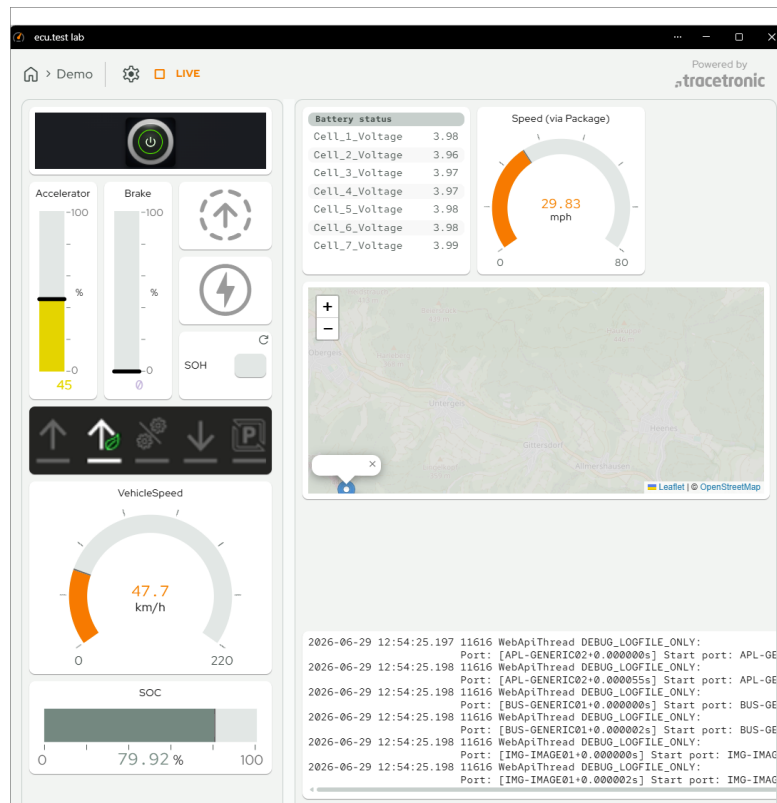


Abbildung 17: Zwei-Seiten-Darstellung in ecu.test lab

Expressions mit Abtastrate in fast allen Widgets verfügbar



Die Zugriffsart **Expression** steht nun in deutlich mehr Widgets zur Verfügung und kann ebenfalls mit einer zyklischen Aktualisierung konfiguriert werden.

Read configuration

Expression ▼

Read expression *

user.myModule.GetCurrentCpuLoad()

☒ Update automatically

Acquisition rate in ms (min 50ms) *

100 123

Abbildung 18: Expression mit Abtastrate konfigurieren

Direkter Zugriff auf Signale und Parameter



Bisher konnte auf Signale und Parameter der Testumgebung ausschließlich über vorab erstellte Mappings oder Packages zugegriffen werden.

Von nun an ist es möglich, direkt auf Größen von Bus, Modell, Messung und Kalibrierung zuzugreifen.

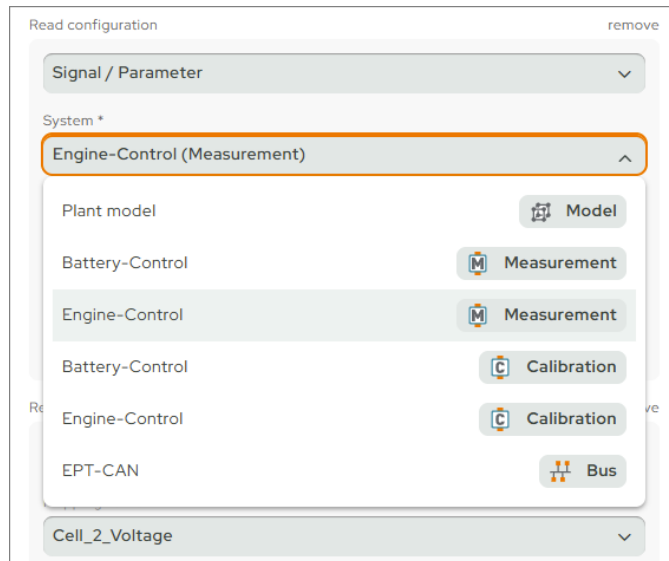


Abbildung 19: System für den Zugriff wählen

Hinweis: Es gelten Zugriffsbeschränkungen. Der direkte Zugriff muss vorab in **ecu.test** aktiviert werden.

Fehlermeldungen direkt am Widget



Bisher wurden Fehlermeldungen zusammengefasst als Meldungen am oberen Fensterrand dargestellt. Dies ermöglichte keine Zuordnung zu einem bestimmten Widget und erschwerte die Behebung der Fehler.

Von nun an werden Konfigurationsfehler bei gestarteter Konfiguration direkt am betroffenen Widget angezeigt. Das vereinfacht die Fehlersuche und verschafft einen schnellen Überblick über das Ausmaß der Fehler.

Laufzeitfehler im LIVE-Modus werden ebenfalls am Widget gemeldet und unterbrechen nun die zyklische Abfrage von Werten. Durch einen Klick auf den Fehler-Indikator kann die zyklische Abfrage wieder aufgenommen werden.

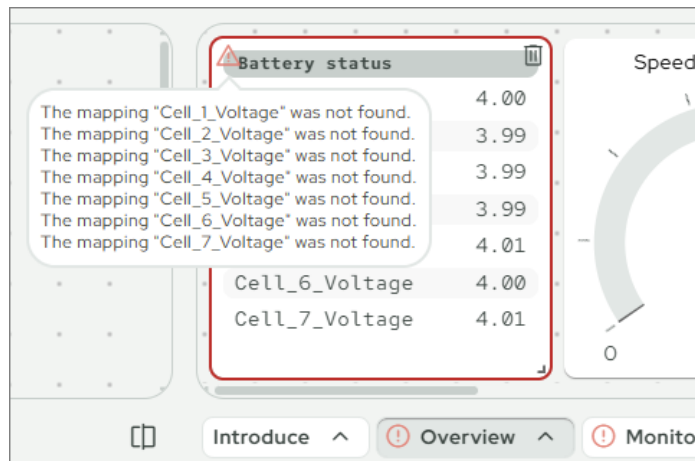


Abbildung 20: Anzeige von Fehlermeldungen direkt am Widget

2.4 trace.xplorer

Diagrammansicht vollständig überarbeitet



Die Implementierung der Diagrammansicht wurde vollständig überarbeitet. Die Änderungen wirken sich überwiegend intern aus. Neben optischen Detailverbesserungen sind bereits jetzt Performancesteigerungen bei der Darstellung großer Traces spürbar.

Das Ändern der Formatierung, der zeitlichen Verschiebung oder der Darstellungshöhe einzelner Signale erfordert nicht mehr das Neuzeichnen der gesamten Ansicht. Gleiches gilt für das Umsortieren von Signalen per Drag'n'Drop. Das Ergebnis ist eine deutlich flüssigere und reaktionsschnellere Bedienung.

Die Tabellenansicht ist von diesen Änderungen nicht betroffen.

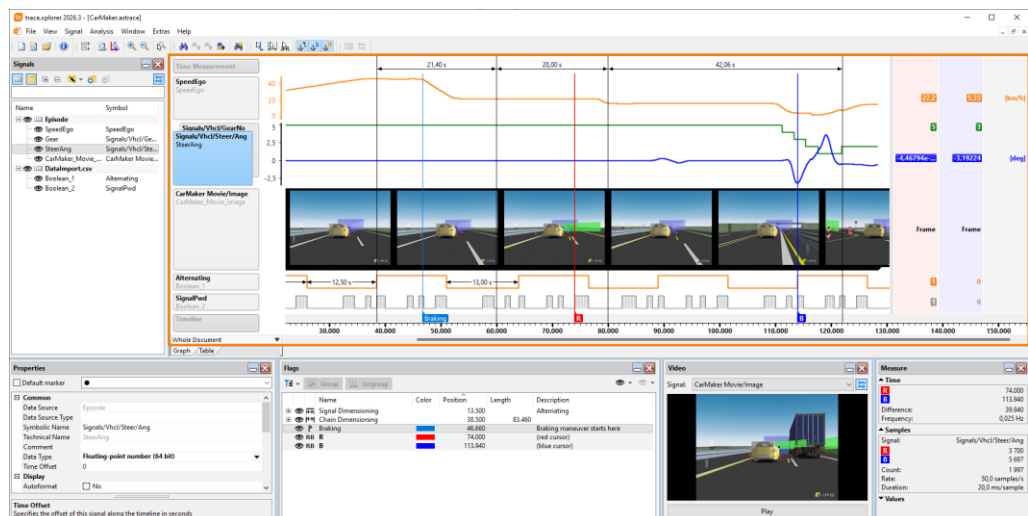


Abbildung 21: Diagrammansicht des trace.xplorer

Bilder von der Diagrammansicht erzeugen



Zur Dokumentation oder Präsentation von Testergebnissen kann es notwendig sein, spezifische Signalverläufe in Bildform erzeugen zu können. Ein bloßer Screenshot des Anwendungsfensters reicht jedoch nicht aus, sobald der umfasste Zeitbereich oder die Anzahl der Signale zu groß werden.

Dafür steht ab sofort die neue Funktion **Datei > Bild erzeugen ...** bereit.

Diese Funktion ist in der Lage, ein Bild der aktiven Diagrammansicht wahlweise in die Zwischenablage oder eine Bilddatei (BMP, JPG oder PNG) zu schreiben. Alle sichtbaren Signale werden in ihrer Anordnung (Höhe, Reihenfolge, Stapel) und Formatierung genau wie in der Diagrammansicht dargestellt. Flags und Be-
maßungen lassen sich bei Bedarf deaktivieren. Der auszugebende Zeitbereich kann individuell bestimmt werden, die maximale Größe des erzeugten Bildes ist begrenzt. Ein Vorschaufenster gestattet das Prüfen des Bildes vor der Ausgabe.

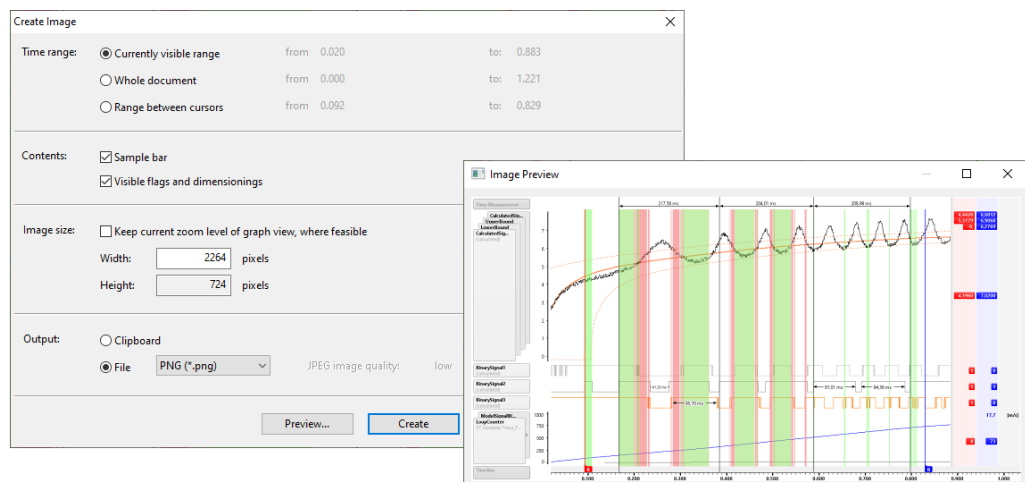


Abbildung 22: Neue Funktion zum Erzeugen von Bildern der Diagrammansicht inklusive Vorschaufenster

Hinweis: Die Nutzung dieses Features erfordert eine **trace.xplorer**-Lizenz.

Programmeinstellungen besser handhaben



Der Dialog zum Bearbeiten der Anwendungseinstellungen wurde modernisiert. Eine Suchfunktion gestattet das schnelle Finden bestimmter Optionen auf den Einstellungsseiten, die nun in Listenform angeboten werden.

Neue Schaltflächen unterhalb der Liste ermöglichen das Zurücksetzen der aktuellen Einstellungsseite oder aller Seiten auf ihre Standardwerte. Weiterhin können Einstellungen als Datei exportiert oder wieder importiert werden, um sie zu sichern oder mit geringem Aufwand auf andere Benutzerprofile zu übertragen.

Der Dialog ist über das Menü **Datei > Programmeinstellungen...** erreichbar.

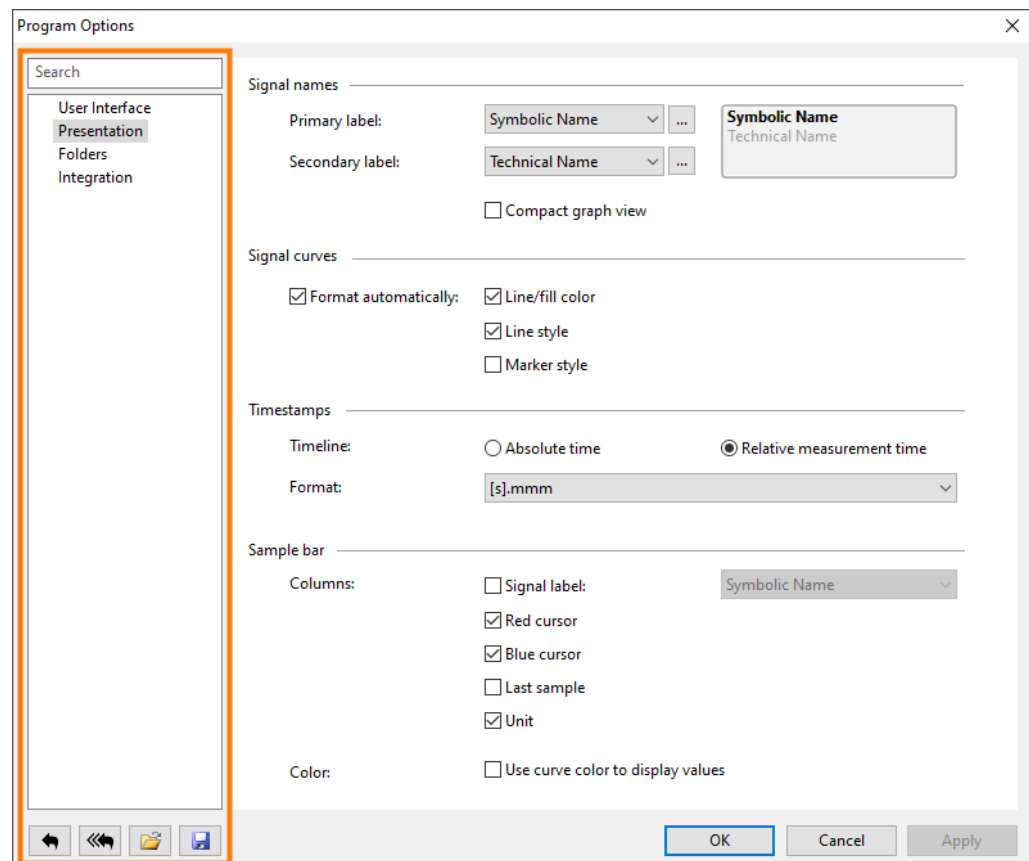


Abbildung 23: Dialog zum Bearbeiten der trace.xplorer-Anwendungseinstellungen

3 Usability

EasyInsert: Unterstützung von Bibliotheksworkspaces



In Bibliotheken enthaltene Artefakte, wie Packages, Projekte und Analyse-Packages, werden über EasyInsert nun mit demselben petrolfarbenen Icon wie im Workspace-Explorer angezeigt.

Der Namespace der Bibliothek erscheint in spitzen Klammern zusammen mit dem relativen Pfad des Artefakts. Bibliothekselemente werden als beste Treffer nach oben sortiert.

Das Einfügen von Bibliotheksinhalten im Editor wird damit deutlich schneller und übersichtlicher.

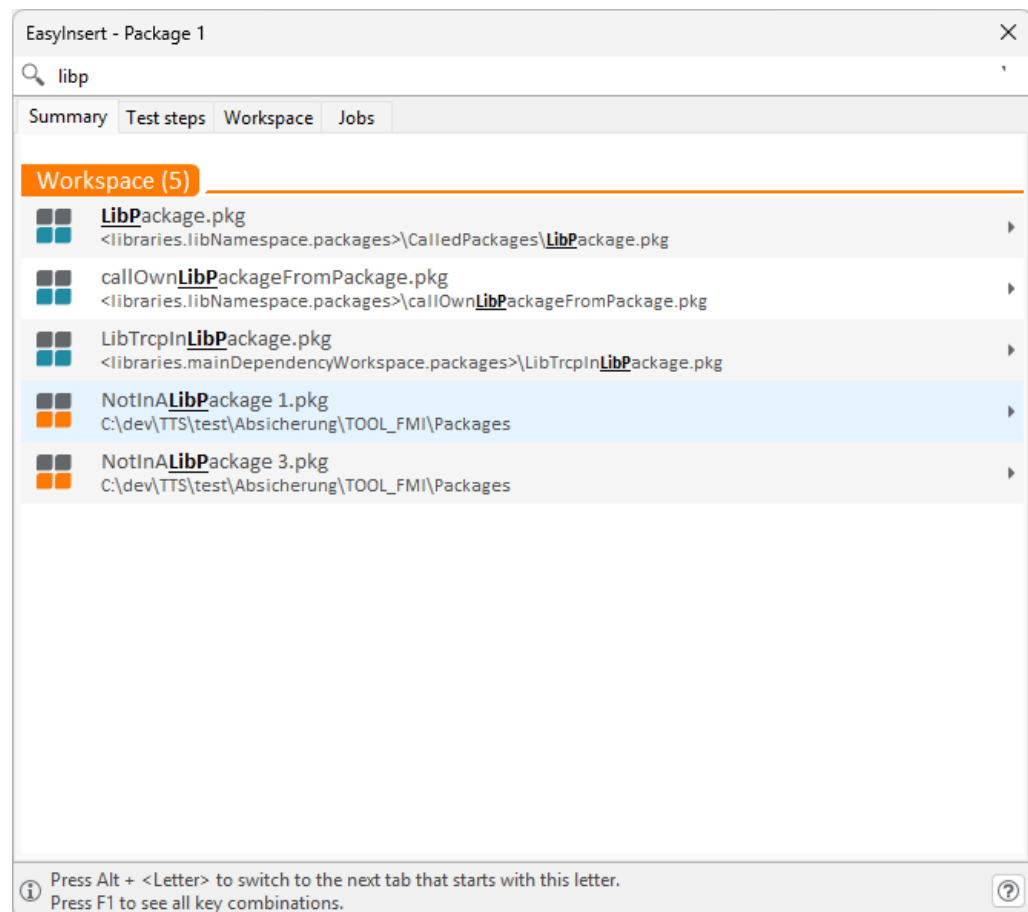


Abbildung 24: EasyInsert unterstützt Bibliotheksworkspaces

Generischer Testschritt PDU



Der neue generische Testschritt **Change PDU Timing** ermöglicht es, generische Testfälle mit PDUs zu beschreiben, die zyklisch versendet werden sollen.

Die konkrete Zuweisung einer PDU erfolgt bei der Parametrierung des Mappings im Testfall, z. B. in Verbindung mit dem [Mapping-Parametrierung im Projekt](#).

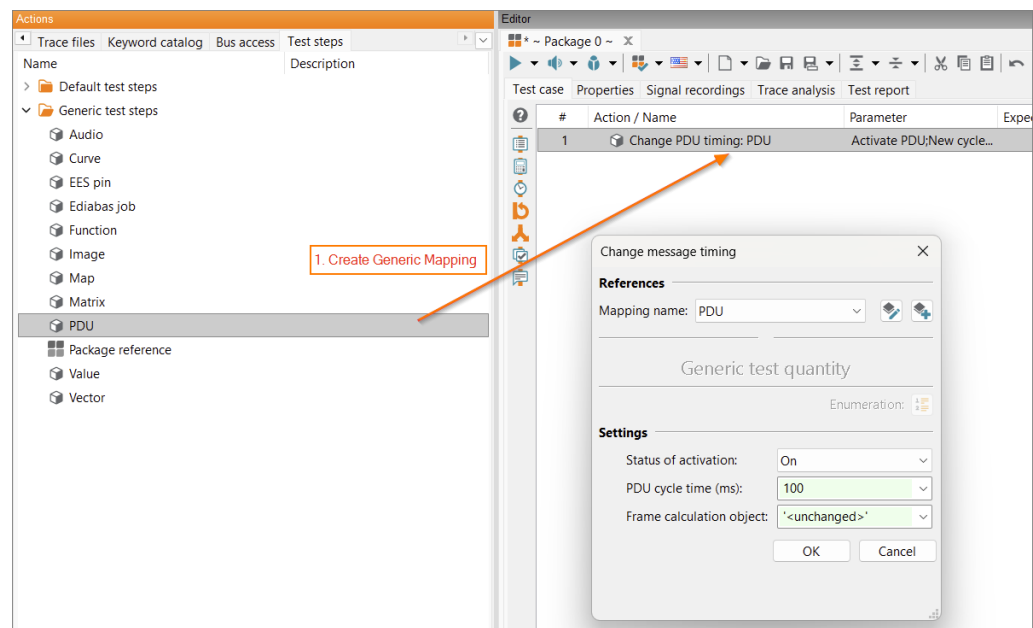


Abbildung 25: Generisches Mapping erstellen

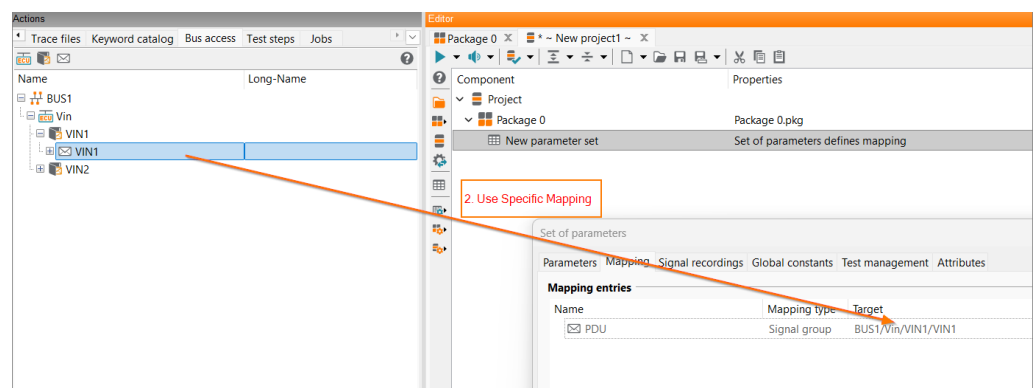


Abbildung 26: Spezifisches Mapping verwenden

Selektive Projektausführung



Die selektive Projektausführung war bisher auf Interaktives Testen beschränkt. Nun ermöglichen wir diesen Ausführungsmodi für jede Form der Projektausführung.

Im Ausführungsdialog kann gewählt werden, welche Komponenten des Projekts ausgeführt werden sollen, unabhängig von ihrer vorgegebenen Reihenfolge.

Nach der Ausführung einer oder mehrerer Komponenten, kann die Ausführung beliebig lange unterbrochen werden und später mit weiteren oder bereits ausgeführten Komponenten fortgesetzt werden.

Zuletzt wird die Ausführung als abgeschlossen markiert und ist damit beendet. Dies kann ganz unabhängig davon geschehen, ob das Projekt vollständig oder nur teilweise ausgeführt wurde.

Hinweis: Die selektive Projektausführung unterstützt aktuell nur Packageaufrufe, Ordner und Parametersätze.

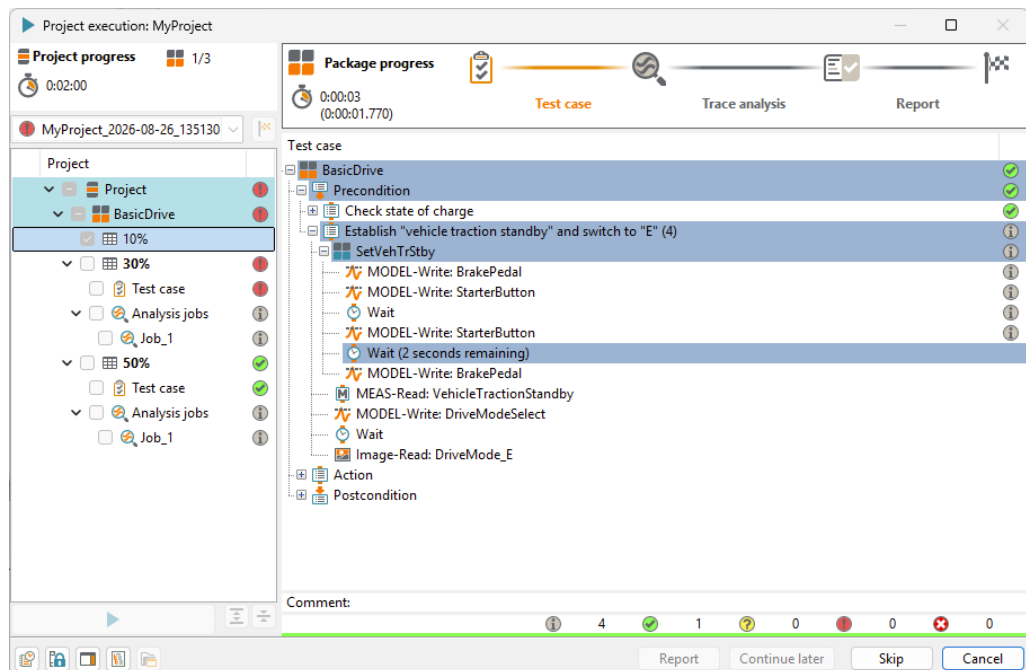


Abbildung 27: Auswahl der Komponenten des Projekts, das ausgeführt werden soll

Neues Changelog-Format



Changelogs – im Installationsverzeichnis und online verfügbar – liegen jetzt in einem Format vor, das zur **ecu.test**-Anwenderdokumentation passt.

Neben einem modernisierten Layout ermöglicht das neue Format unter anderem die Suche und Filterung nach bestimmten Einträgen.

Zum Lesen und Ausprobieren [hier klicken](#).

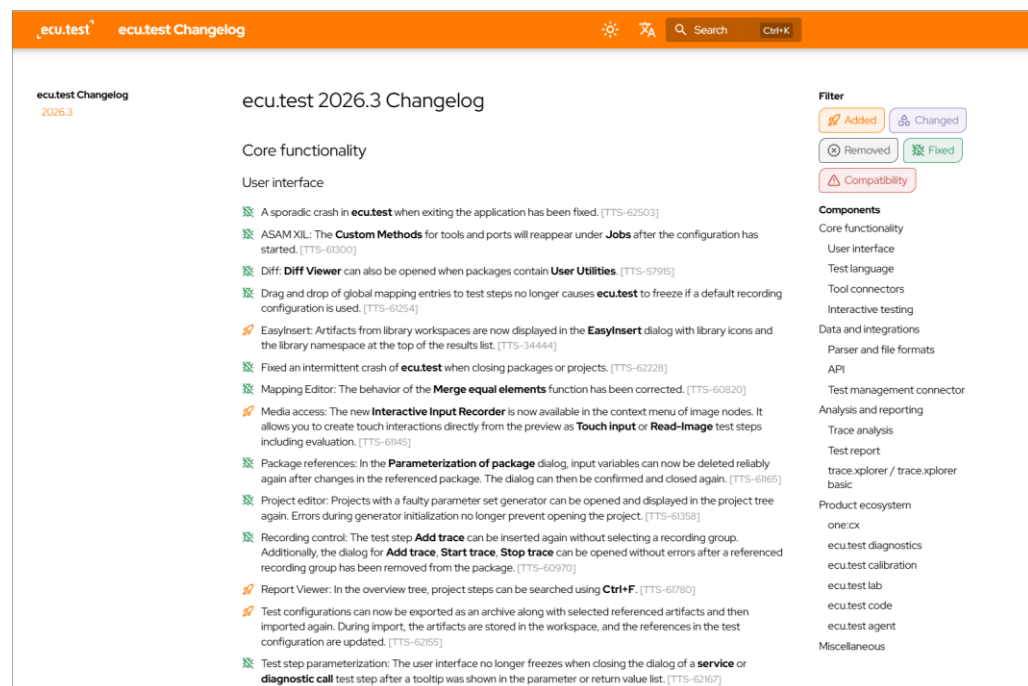


Abbildung 28: Neues Changelog-Layout

4 Testaspekte

4.1 Kommunikation

Ethernet-Logging jetzt auch im Gigabit-Bereich verlustfrei



SOME/IP und Socket-Adaptor-Kommunikation, einschließlich MACsec-geschützter Daten, können in **ecu.test** live aus den Ethernet-Loggerdatenströmen ASAM CMP, PLP und TECMP verarbeitet werden. Die Optimierung für DLT wurde bereits mit **ecu.test** 2026.2 eingeführt.

In diesem Release wurde die Verarbeitungsleistung auf alle weiteren unterstützten Protokolle deutlich erhöht. Dadurch lassen sich in **ecu.test** auch bei Datenraten im Gigabit-Bereich Traces verlustfrei aufzeichnen und Signale lesen.

Hinweis: Die Verbesserung ist automatisch aktiv und wird für alle Ethernet-Bus-Capture-Protokolle verwendet – eine manuelle Aktivierung ist nicht erforderlich.

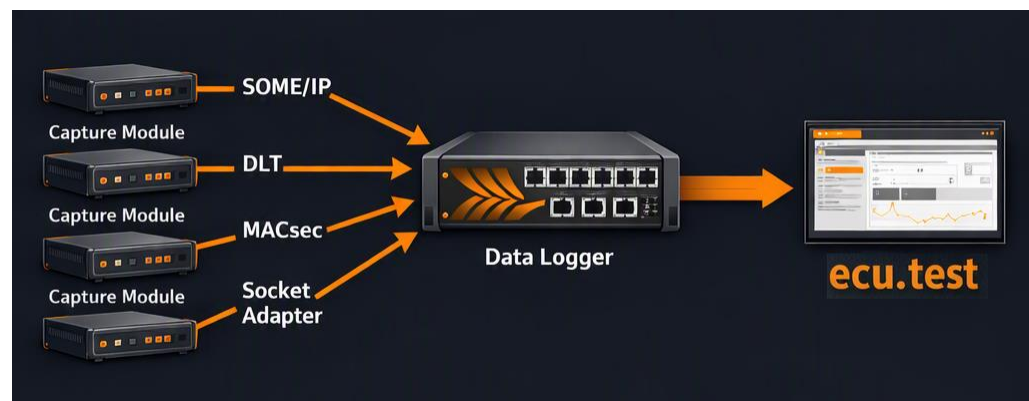


Abbildung 29: Schema zur Verarbeitung von Ethernet-Loggerdatenströmen in ecu.test

IEEE1722: IPv6 Unterstützung



CAN-Bus und J1939-Kommunikation, die via Ethernet über IEEE1722 übertragen wird, kann jetzt auch IPv6 als Netzwerkschicht nutzen.

TLS-Unterstützung für TCP-Client



Um nutzerdefinierte Kommunikation, die auf TCP aufsetzt und TLS 1.2 Authentifizierungs- und Verschlüsselungsgeschützt erfolgt, wurde der TCP-Client-Port erweitert.

Die Jobs zum Senden und Empfangen ver- und entschlüsseln die Übertragung dabei automatisch.

PTP Unterstützung mit dem Tool Ethernet für Windows und Linux



Die folgenden Jobs stehen jetzt auch für Linux auf dem Ethernet-RAW Port zur Verfügung:

- StartPtpClient
- StartPtpMaster

Sie ermöglichen die Simulation eines Timing-Masters bzw. -Clients.

Die Funktionen waren bislang der Ethernet-Unterstützung dem Tool Vector XL vorbehalten.

4.2 MiL

Nutzerdefinierte Restrukturierung von Modelgrößen



Große Modelle lassen sich jetzt deutlich einfacher beherrschen: Modellgrößen können gezielt gefiltert, umbenannt und in einer eigenen Struktur dargestellt werden. So entsteht aus komplexen technischen Modellen eine übersichtliche, projektspezifische Sicht, in der relevante Signale schneller gefunden und genutzt werden können.

Das erleichtert die tägliche Arbeit spürbar, besonders dann, wenn die fachliche Sicht nicht zur ursprünglichen Modellstruktur passt. Dabei bleiben alle Verknüpfungen zu den eigentlichen Modellgrößen erhalten, sodass weiterhin sicher auf die echten Modellvariablen zugegriffen wird.

4.3 Multimedia

Zentrale Konfiguration des Multimedia-Agents



Der **Multimedia-Agent** lässt sich nun zentral in den **Workspace-Einstellungen** konfigurieren. Dabei können unter anderem folgende Einstellungen vorgenommen werden:

- Modellauswahl
- Temperatur
- Optionen zur Bildvorverarbeitung:
 - Größenanpassung
 - maximale Kantenlänge
- Nachverarbeitung von Koordinaten
- zusätzliche Modellanweisungen

Änderungen werden direkt validiert und die resultierende Konfiguration kann vor dem Speichern geprüft werden. Das erleichtert eine konsistente und nachvollziehbare Parametrisierung von Multimedia-Testschritten.

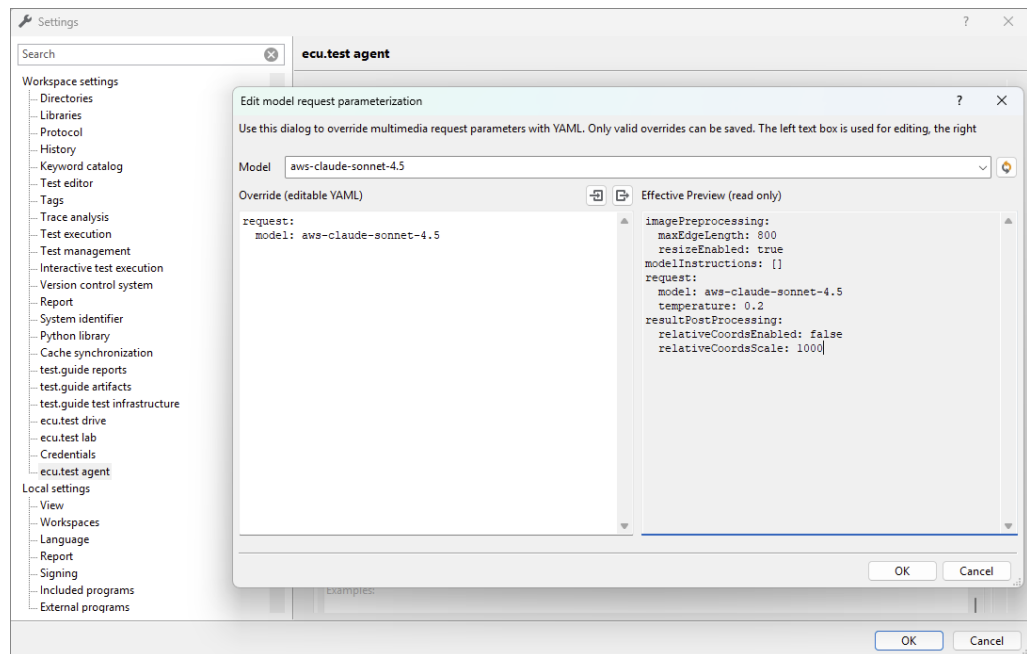


Abbildung 30: Zentrale Konfiguration des Multimedia-Agents

Interaktiver Input Recorder: Record & Replay



Das manuelle Erstellen von **TouchInput**- und **Bild-lesen**-Testschritten war bisher mühsam, da Koordinaten händisch als Parameter in die Testschritte übertragen werden mussten.

Mit dem neuen interaktiven Input Recorder wird das System unter Test (SUT) direkt gestreamt und in einem Fenster dargestellt, etwa ein ADB-Device oder eine Webseite. In diesem Fenster können Gesten wie **Tap**, **Swipe** oder **Read** direkt ausgeführt werden.

Die Aktionen landen als korrekt parametrisierte Testschritte automatisch im Package.

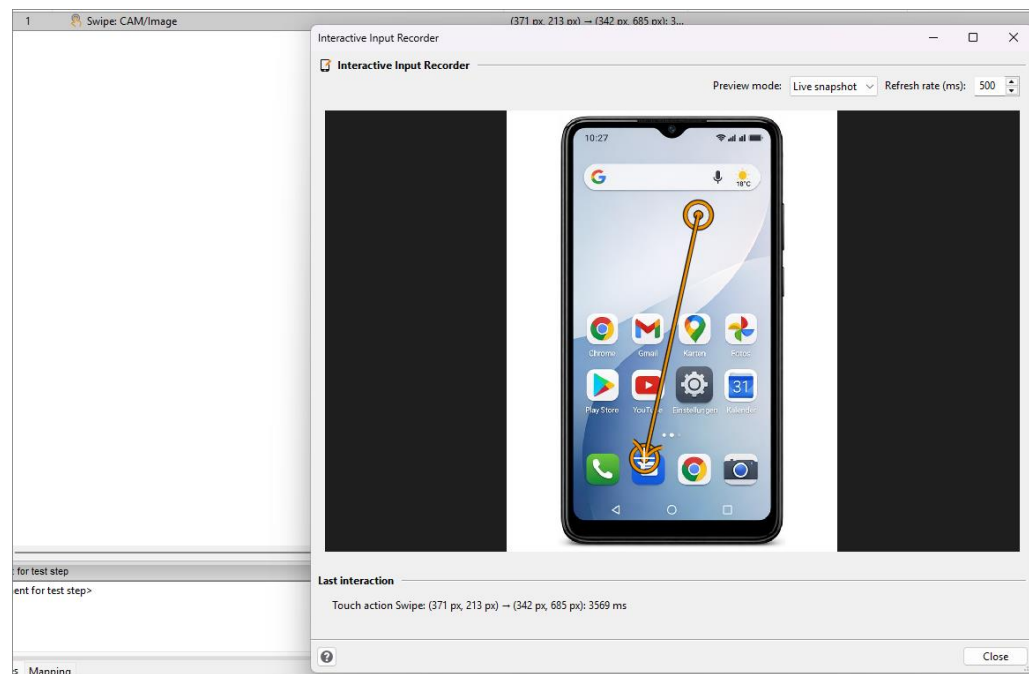


Abbildung 31: Interaktiver Input Recorder

Benchmark für Multimedia-Modelle



Zur Bewertung KI-gestützter Testschritte ist jetzt ein Benchmark-Workspace verfügbar.

ReadImage-, **Touch**- und **agentische** Testschritte werden dabei in einer simulierten Testumgebung automatisiert ausgeführt und anhand eines Scores bewertet.

So lassen sich verschiedene Modelle reproduzierbar vergleichen und Auswirkungen von Modell-, Einstellungs- oder Prompt-Änderungen frühzeitig erkennen.

Der Workspace kann über support@tracetronic.com angefragt werden.

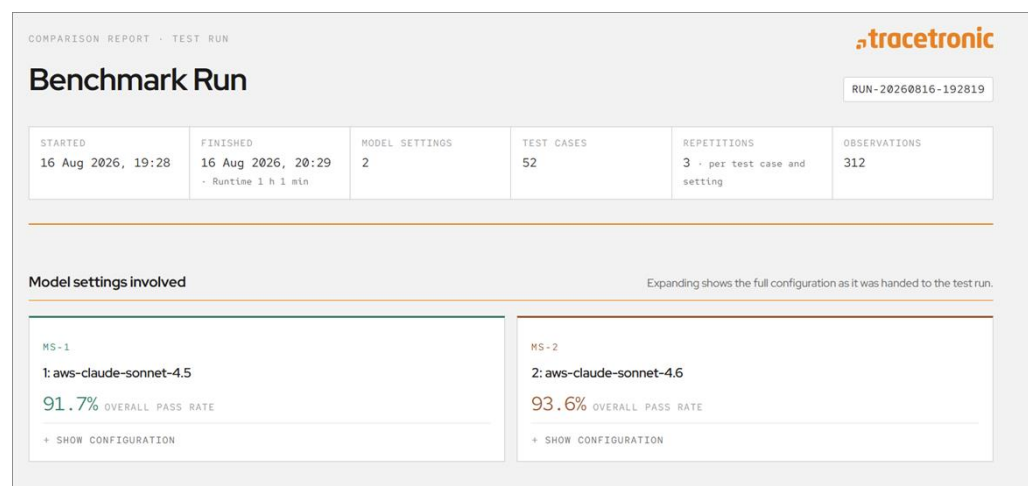


Abbildung 32: Bewertung KI-gestützter Testschritte ist einem Benchmark-Workspace

4.4 Testmanagement

Änderungen am Adapter für das integrierte Testmanagement



Mit **ecu.test 2026.1** wurde angekündigt, dass die integrierten Testmanagement Adapter für HP QC, PTC Integrity, ETM und Polarion **ab ecu.test 2027.1** entfernt werden.

Ab **ecu.test 2026.3** sind diese Adapter nur noch über ein Feature-Flag verfügbar – sie sind standardmäßig nicht mehr sichtbar, können aber durch Aktivieren des Flags weiterhin genutzt werden.

Als Nachfolger steht eine Python-basierte Lösung zur Verfügung, die mehr Flexibilität und eine bessere Wartbarkeit bietet.

Handlungsbedarf: Wer eine dieser Integrationen aktuell nutzt oder noch benötigt, sollte sich zeitnah an den Support wenden:

- Bereitstellung des Feature-Flags für die weitere Nutzung
- Gemeinsame Erstellung eines Übergangsplans

Unterstützung für Siemens Polarion 2606



Mit der Veröffentlichung von Siemens Polarion 2606 haben wir die Beispiel-Implementierungen sowie die tracetronic-Polarion-Bibliothek aktualisiert, um die vollständige Kompatibilität mit der neuen Version sicherzustellen. Dies gewährleistet die weitere Funktionsfähigkeit und ermöglicht die Nutzung der neuen Features von Polarion 2606.

4.5 Traceanalyse

Default-Werte für Traceschrittvorlagen



Bisher war das Festlegen von Standardwerten nur für arraybasierte NumPy-Traceschrittvorlagen möglich.

Mit **ecu.test** 2026.3 ist dies nun auch für ereignis- und flussbasierte Python-Traceschrittvorlagen möglich. Um die Rückverfolgbarkeit zu verbessern, werden die Standardwerte bei der Konfiguration von vorlagenbasierten Traceschritten angezeigt.

Trace step configuration (Template: 'ExampleTraceStepTemplates/MinimumMaximum/UpperBound.py')

Common

Name:

Description:

Signals

Name	Direction	Generic signal	Optional	Unit
Signal	IN	(- Please select -)	☐	don't care

Parameters

Name	Direction	Type	Value
Threshold	IN	PyObject	(100)

Add found spots to report

Maximum count: Minimum distance:

Trace step template

Path:

Description:

OK Cancel

Abbildung 33: Anzeige von Standardwerten bei der Konfiguration von vorlagenbasierten Traceschritten

4.6 Weitere Testaspekte

Default-Aufnahmekonfiguration: Namensvorlage für Aufnahmegruppen



Die Default-Aufnahmekonfiguration ermöglicht es, Aufnahmegruppen zentral über die Testkonfiguration zu verwalten. Ergänzend wird nun auch die Namensvorlage aus der passenden Aufnahmegruppe der Default-Aufnahmekonfiguration übernommen, wenn im Testpackage keine Namensvorlage definiert ist.

Verbesserungen im Report Viewer



Die Ansicht für Projektschritte enthält nun eine Spalte für die Dauer. Bisher war diese Information nur in den Details eines Schritts versteckt und kann nun für mehrere Schritte auf einen Blick erfasst werden.

#	Evaluation	Time (absolute)	Duration	Name
2	NONE	23.09.2026 16:00:18	0:00:00.58	PlotExamples
4	NONE	23.09.2026 16:00:19	0:00:00.46	ReferencingEpisodeExample
1	FAILED	23.09.2026 16:00:18	0:00:06.59	Projekt
6	FAILED	23.09.2026 16:00:19	0:00:04.83	CheckSignalInEnvelope

Abbildung 34: Projektschritte enthalten eine Spalte mit der Dauer

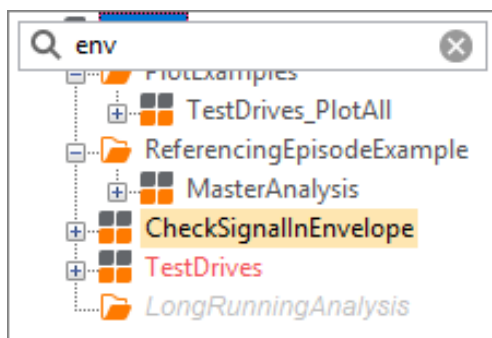


Abbildung 35: Strg+F zur Suche nach Projektschritten

Außerdem ist es nun möglich im Übersichtsbaum Projektschritte anhand des Namens zu suchen.

Die Suche kann bei Fokus im Baum per Tastatureingabe oder über **Strg+F** gestartet werden.

Report Downloader – einfaches Downloaden eines vollständigen Reportverzeichnis aus test.guide



Im Installationsverzeichnis steht die neue Anwendung **report_downloader** bereit.

Mithilfe dieser können Reports anhand der Report-ID aus **test.guide** heruntergeladen werden.

Die notwendigen Verbindungseinstellungen werden aus dem letzten Workspace genommen, können aber alternativ überschrieben werden. Über eine CLI können Downloads auch skriptgesteuert gestartet werden.

Name	Report ID	Status	Progress	Speed	ETA	Details
Demo_Regression_Run_01	581204	Finished	100%	-	-	
Demo_System_Check_02	581317	Finished	100%	-	-	
Demo_Connectivity_Test_03	581903	Error	43%	-	-	Connection interrupted: internet connection lost.
Demo_Road_Test_04	582041	Downloading	87%	2.1 MB/s	0:07:01	Downloading (19/39): A-CAN.asc.zip
Demo_Thermal_Run_05	582166	Downloading	65%	3.7 MB/s	0:00:02	Downloading (12/57): thermal_run_05.trf

Abbildung 36: Download eines vollständigen Reportverzeichnis aus test.guide

5 Versionen und Schnittstellen

5.1 Neue Tools und Versionen



	Provider	Web-seite	System	Produktname	Version
1	dSPACE	Release-Link	Sensorgestützte realistische Simulation für ADAS/AD in Echtzeit	AURELION	26.2
2	dSPACE	Release-Link	Software für die Entwicklung von ECUs	ModelDesk	R2025-B
3	Vector	Link	Open-Source-Bibliothek zur Anbindung von SIL-Umgebungen	SIL Kit	bis Version 5.0.6
4		Release-Link	Software-Bibliothek für Windows, mit dem Programme Netzwerkverkehr abfangen, überwachen und analysieren können	npcap	1.88

5.2 APIs

5.2.1 Command Line Interface

Flag für vollautomatisierte Nutzung



Mit dem neuen Flag **startupAutomated** wird sichergestellt, dass auch bei der Verwendung der **ecu.test**-GUI alle Popups, die normalerweise eine Nutzerinteraktion erfordern. Dadurch werden vollautomatisierte Workflows vereinfacht.

Python-Abhängigkeiten über CLI installieren



Bisher konnten die in einem Workspace definierten Python-Bibliotheken manuell über die Programmeinstellungen oder automatisch beim Programmstart installiert werden. Zusätzlich lassen sie sich nun über die Kommandozeile installieren:

- **ecu.test task install-python-libraries --workspace <path>**

Der Aufruf installiert die Bibliotheken für den angegebenen Workspace, ohne dass **ecu.test** dafür mit seiner Benutzeroberfläche gestartet werden muss. Das ist besonders hilfreich, wenn beispielsweise ein Docker-Image mit einem vollständig vorbereiteten Workspace erstellt werden soll.

5.2.2 Custom Checks

Ergebnisse um ID und Herkunft erweitert



Ergebnisse aus Package- und Projektprüfungen unterstützen nun die Felder **id** und **origin**. Mit **id** können eigene Implementierungen einen Befundtyp eindeutig und stabil kennzeichnen. **origin** wird von **ecu.test** gesetzt und gibt den Namespace des Workspaces an, aus dem der Check stammt, oder den Wert "internal" für interne Checks.

Fixes über API anwenden



Beim Prüfen von Packages über die Internal API oder REST API steuert ein neuer optionaler Parameter, ob eindeutige Fixes angewendet werden. Die Rückgabestruktur enthält zudem Informationen darüber, welche Probleme durch Fixes behoben wurden.

5.2.3 Object API

Erweiterung der API für pkgattr.spec und prjattr.spec



Bisher konnten Package- und Projekt-Attributspezifikationen nur durch manuelle Änderungen an den Dateien angepasst werden. Für benutzerdefinierte Testmanagement-Adapter ist jedoch eine automatische Ableitung und Änderung der Attribute aus dem ALM erforderlich.

Die neue Object API **AttributeSpecApi** ermöglicht das programmgesteuerte Lesen, Ändern und Speichern von Attributen. Dies erlaubt eine dynamische Synchronisation mit dem ALM, reduziert manuelle Aufwände und erhöht die Konsistenz der Test-Daten.

5.2.4 REST API

Abfrage allgemeiner **ecu.test**-Informationen



Mit dem neuen Endpoint-Info können allgemeine **ecu.test**-Informationen wie relevante Verzeichnisse und Sprache abgefragt werden. Dadurch können im Rahmen von vollautomatisierte Workflows komplexere Prozesse umgesetzt werden.

5.2.5 Test Management API

Änderungen



Die Test Management API wurde im Rahmen der neuen Schnittstellenbibliothek auf eine neue Schnittstelle migriert.

Davon profitieren die IDE-Integration und die KI-gestützte Entwicklung gleichermaßen – durch bessere Code-Vervollständigung, klarere Konventionen und einer verbesserten Auffindbarkeit der verfügbaren Funktionen.

Die Abwärtskompatibilität bleibt vollständig erhalten. Die bisherige Schnittstelle wird weiterhin unterstützt, sodass bestehende Implementierungen ohne Änderungen weiter funktionieren.

6 Abkündigungen

6.1 Abkündigungen und Inkompatibilitäten in dieser Version

Fibex-Unterstützung



Die Fibex-Unterstützung für Bus wurde entfernt. Die Fibex-Unterstützung für DLT bleibt weiterhin erhalten.

Basler: Pylon



Die Kameras der Firma Basler sind kompatibel zum GenICam Standard, was eine eigene Toolanbindung unnötig macht. Das Tool **Basler: Pylon** wurde entfernt.

Bei Verwendung einer Kamera von Basler kann alternativ die GenICam-Toolanbindung verwendet werden.

Bus-Monitoring-Testschritte



Die Bus-Monitoring-Testschritte zum Prüfen von **Timings** und **DLC** können deutlich besser mit den Mitteln der Traceanalyse und den mitgelieferten Analysebausteinen umgesetzt werden.

Mit **ecu.test 2026.3** wird das Einfügen neuer Testschritte nicht mehr unterstützt. Die Unterstützung der Ausführung wird frühestens in **ecu.test 2027.1** entfernt.

Import im Zusammenhang mit CustomChecks



Eigene Prüfungen verwenden unter Umständen ein Modul, das umgezogen ist. Daher müssen die Importe geprüft und ggf. angepasst werden. Die Abkündigung erfolgt mit **ecu.test 2026.3**.

- **alt:** `tts.core.common.check.Constants`
- **neu:** `tts.interface.customCheck.Constants`

Echtzeit-Packagereferenzen



dSPACE hat mit **ControlDesk 2024-B** ihre Schnittstelle zur Ansteuerung der Echtzeitumgebung geändert. Seit diesem dSPACE-Release ist die Ausführung von Echtzeit-Packagereferenzen mit **ecu.test** nicht mehr möglich.

Achtung: Die Abkündigung bezieht sich lediglich auf Echtzeit-Packagereferenzen. Weitere Features rund um das Thema Real Time Testing (RTT) sind nicht betroffen.

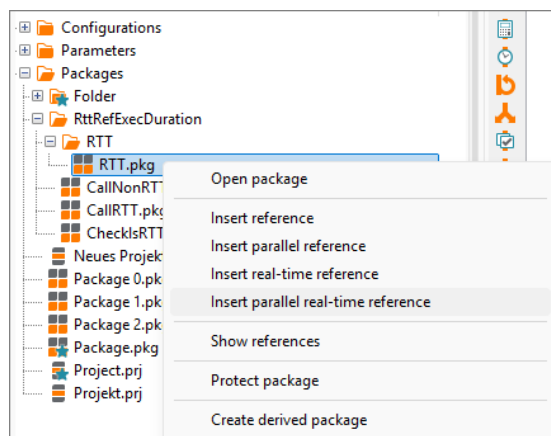


Abbildung 37: Insert parallel real-time package references

dSPACE MotionDesk



Die Anbindung dSPACE: MotionDesk wurde entfernt, da kein offizieller Support mehr seitens dSPACE besteht. Als Ersatz steht dSPACE AURELION zur Verfügung.

TRF-Dateien lassen sich für Projektberichte nicht mehr erzeugen



Im Zuge der Einführung des PRF-Formats für Projektberichte ist das TRF-Format obsolet. Die Fallback-Einstellung, das TRF-Format während der Migrationsphase über die **Workspace-Einstellungen** zu aktivieren, wurde entfernt.

Diese Abkündigung betrifft ausschließlich Projektberichte. Package-Ausführungen werden weiterhin im TRF-Format gespeichert.

Log-Verzeichnisse/Package-Unterordner mit Zeilennummer als Präfix



Log-Verzeichnisse und Package-Unterordner im Projektreport erhalten ab sofort die Zeilennummer der jeweiligen Projektkomponente als Präfix. Das verhindert, dass sich Ergebnisse von Projektkomponenten in Ausnahmefällen gegenseitig überschreiben.

Der korrekte Ordnername kann über die folgenden APIs abgefragt werden:

- **Internal API:** `api.TestEnvironment.ExecutionInfo.LogFolder`
- **Object API:** `Package.GetSubReportPath()`
- **Report API:** `Package.GetSubReportPath()`

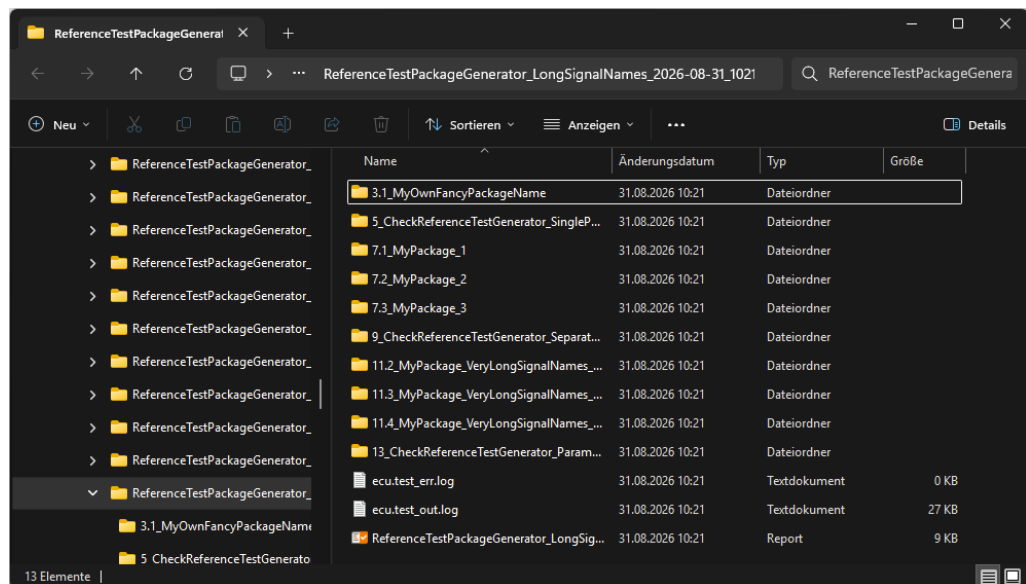


Abbildung 38: Log-Verzeichnisse/Package-Unterordner mit Zeilennummer als Präfix

Automatisches Workspace-Backup bei ecu.test-Updates entfällt



Das automatische Erstellen eines ZIP-Archivs der Workspace-Einstellungen bei **ecu.test**-Updates ist entfallen. Der Info-Dialog beim Start von **ecu.test**, der auf dieses Backup hingewiesen hat, wurde ebenfalls entfernt.

Als Ersatz empfiehlt sich der neue **Kompatibilitätsmodus** (siehe *Workspace-kompatibilität zwischen verschiedenen ecu.test-Versionen*). Alternativ können die Workspace-Einstellungen manuell als Backup gesichert oder ein SCM wie Git verwendet werden.

Erstellung von Ausführungsaufträgen in test.guide



Zwei Ausführungsmodi legen im Rahmen ihres Workflows Ausführungsaufträge in **test.guide** an: die nachgelagerte Traceanalyseausführung und die Ausführungsverteilung über **test.guide**.

Ab **ecu.test 2026.3** wird die Git-URL unverändert weitergegeben, statt auf HTTPS normiert zu werden.

test.guide ab Version 1.224.0 verarbeitet Git-URLs protokollunabhängig und findet ein konfigurierte SCM-Tool unabhängig vom verwendeten Protokoll.

Ist eine Aktualisierung von **test.guide** nicht zeitnah und wird HTTPS-Normierung weiterhin benötigt, kann das alte Verhalten über folgendes Feature-Flag erzwungen werden:

- **TG-EXECUTION-TASK-GIT-URL-NORMALIZE-HTTPS**

Hinweis: Dieses Feature-Flag ist nur übergangsweise verfügbar und wird mit **ecu.test 2027.1** wieder entfernt.

6.2 Abkündigungen in zukünftigen Versionen

Tektronix UTA 12



Das Tool **Tektronix UTA 12** ist nicht mehr auf dem Markt und wird zu **ecu.test 2026.4** abgekündigt.

QUANCOM



Der API-Schnittstelle **QLIB** für **Quancom** ist nicht mehr kompatibel zu aktuellen Geräten und wird zu **ecu.test 2026.4** abgekündigt.

Siemens Polarion



Der integrierte Test Management Adapter für **Siemens Polarion** wird mit **ecu.test 2027.1** entfernt.

Als Ersatz können benutzerdefinierte Test Management Adapter eingesetzt werden. Diese Adapter bieten deutlich mehr Flexibilität und ermöglichen eine optimale Anpassung an den jeweiligen Workflow.

Als Hilfestellung für die eigene Implementierung wird ein Muster-Workflow bereitgestellt.

IBM Rational Quality Manager (RQM)



Der integrierte Test Management Adapter für **IBM Rational Quality Manager (RQM)** wird mit **ecu.test 2027.1** entfernt.

Als Ersatz können benutzerdefinierte Test Management Adapter eingesetzt werden. Diese Adapter bieten deutlich mehr Flexibilität und ermöglichen eine optimale Anpassung an den jeweiligen Workflow.

Als Hilfestellung für die eigene Implementierung wird ein Muster-Workflow bereitgestellt.

HP Application Lifecycle Management (ALM)



Der integrierte Test Management Adapter **für HP Application Lifecycle Management (ALM)** wird mit **ecu.test 2027.1** entfernt.

Um nach dieser Version weiterhin mit HP Application Lifecycle Management (ALM) interagieren zu können, muss ein benutzerdefinierter Testmanagementadapter mithilfe der **ecu.test** Test Management API implementiert werden.

PTC Integrity



Der integrierte Test Management Adapter für **PTC Integrity** wird mit **ecu.test 2027.1** entfernt.

Um nach dieser Version weiterhin mit PTC Integrity interagieren zu können, muss ein benutzerdefinierter Testmanagementadapter mithilfe der **ecu.test** Test Management API implementiert werden.

Interaktive Testausführung in **ecu.test**



Die Anwendungsfälle der interaktiven Testausführung innerhalb von **ecu.test** und der Web-Applikation **ecu.test drive** überlappen sich größtenteils.

ecu.test drive ist die deutlich flexiblere Lösung aufgrund der

- intuitiveren Nutzerführung,
- besseren Anpassbarkeit an die Anwendungsgebiete im Fahrzeug,
- Verwendbarkeit mit **ecu.test** Runner und Remote durch Entkopplung von **ecu.test**-GUI

Daher konzentrieren wir uns in Zukunft auf diese Lösung und werde die interaktive Testausführung mit **ecu.test 2027.1** entfernen.

Erweitertes Interface der UserSynchronization



Die Methode **Synchronize()** wird zukünftig mit zwei zusätzlichen Parametern aufgerufen:

- **masterRecordingInfo**
- **recordingInfo**

Bestehende Implementierungen müssen zeitnah angepasst werden.
Mit **ecu.test 2027.1** wird das alte Interface nicht mehr unterstützt.

Bus Monitoring Testschritte in **ecu.test**



Die Bus-Monitoring-Testschritte zum Prüfen von Timings und DLC können deutlich besser mit den Mitteln der Traceanalyse und mitgelieferten Analysebausteinen umgesetzt werden. Die Unterstützung der Ausführung wird frühestens in **ecu.test 2027.1** entfernt.