

# vTESTstudio

## Comfortable Development of Automated Test Sequences for Embedded Systems

### What is vTESTstudio?

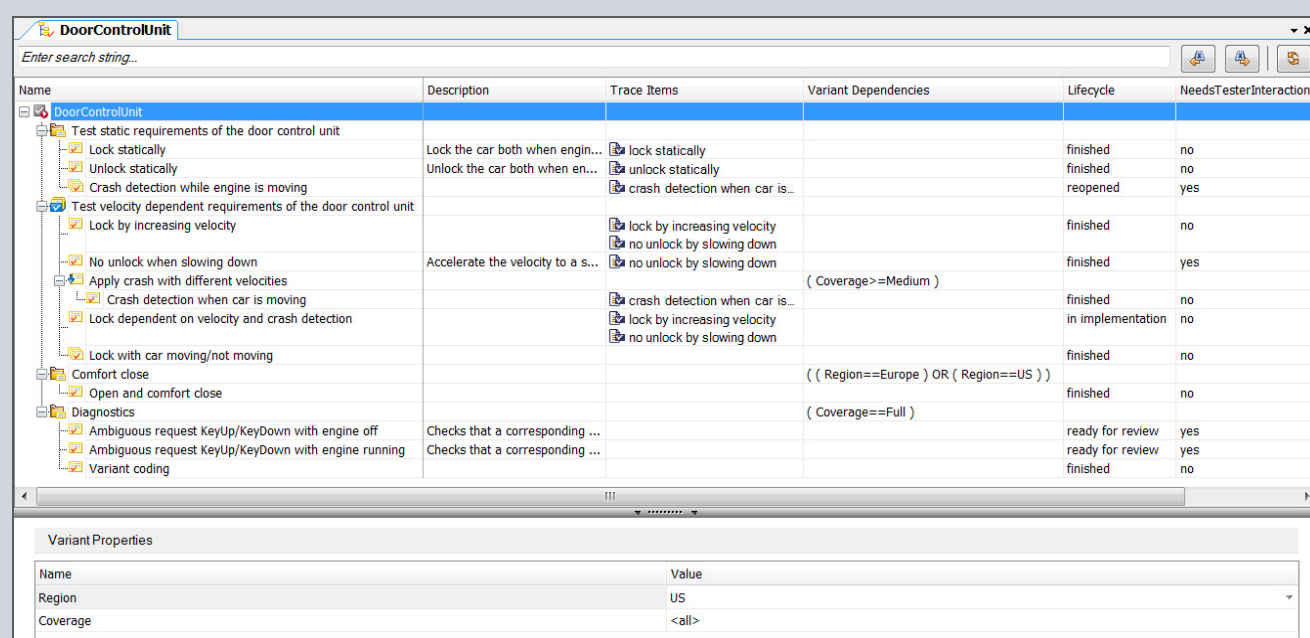
- > A high-performance development environment for creating test sequences
- > Can be used in all product development phases – from model tests, to tests that support development, to tests on the HIL test bench
- > Seamlessly integrates both proven and new types of test design methods and test notations
- > Generates test sequences which can be executed with the CANoe test sequencer in real-time and evaluated in detailed reports

### Highlights of Version 3.0

- > **Ribbon and tabs** sorted by application areas
- > **Test Case Overview** for a quick overview of all test cases including their attributes and trace items
- > **Navigation to test cases, test commands, etc.** from within other tools (e.g. CANoe, test management systems ...)
- > **Connection to MS Visual Studio** to create and maintain C# code for vTESTstudio projects
- > Full **multi-monitor support**

### Overview of Advantages

- > **Broad range of applications** through the use of various test design editors
- > **Flexible parameterization** of test sequences with scalar values, test vectors and stimulation curves which can be accessed from all test design languages
- > **Universal variant support** for structure, implementation and parameterization of tests
- > **Simple maintenance** of test projects on account of the modular structure and a variety of structuring options such as libraries and folders
- > **Direct access to XIL test environment symbols** in the test sequences (HIL, SIL, MIL, ...)
- > **Universal traceability** of externally defined requirements and test specifications in test implementation and test report
- > **High test coverage** without extensive programming thanks to the support of parameterized test case lists
- > **Open interfaces** for simple integration into existing tool landscapes



The screenshot shows the 'Test Case Overview' window for a test unit named 'DoorControlUnit'. It displays a table of test cases with columns for Name, Description, Trace Items, Variant Dependencies, Lifecycle, and NeedsTesterInteraction. A tree view on the left shows the hierarchical structure of the test cases. Below the table, the 'Variant Properties' section shows the current configuration: Region is set to 'US' and Coverage is set to '<all>'.

Name	Description	Trace Items	Variant Dependencies	Lifecycle	NeedsTesterInteraction
DoorControlUnit					
Test static requirements of the door control unit					
Lock statically	Lock the car both when engin...	lock statically		finished	no
Unlock statically	Unlock the car both when en...	unlock statically		finished	no
Crash detection while engine is moving		crash detection when car is...		reopened	yes
Test velocity dependent requirements of the door control unit					
Lock by increasing velocity		lock by increasing velocity		finished	no
No unlock when slowing down	Accelerate the velocity to a s...	no unlock by slowing down		finished	yes
Apply crash with different velocities		crash detection when car is...	( Coverage>=Medium )	finished	no
Crash detection when car is moving		lock by increasing velocity		in implementation	no
Lock dependent on velocity and crash detection		no unlock by slowing down		finished	no
Lock with car moving/not moving			(( Region==Europe ) OR ( Region==US ))	finished	no
Comfort close					
Open and comfort close					
Diagnostics			( Coverage==Full )		
Ambiguous request KeyUp/KeyDown with engine off	Checks that a corresponding ...			ready for review	yes
Ambiguous request KeyUp/KeyDown with engine running	Checks that a corresponding ...			ready for review	yes
Variant coding				finished	no

Variant Properties	
Name	Value
Region	US
Coverage	<all>

Quick overview of all test cases in a test unit with the Test Case Overview of vTESTstudio

## Test Design Editors

Depending on the task and requirements involved, users can switch between the various complementary test design techniques integrated into vTESTstudio for the development of automation sequences.

> **Test tables:** Creation of test sequences from pre-defined and self-defined test commands in tabular form without the need for programming skills

> **Programming in CAPL and C#:** Definition of sequential or event-based test flows for asynchronous preprocessing of measurement values

> **Graphical test design notations (as separate option):**

Modeling of test sequences in graphical notation or state diagrams for automatic test case generation simplifies reviews of complex test scenarios

## Interfaces, Bus Systems and Protocols

### > Test sequencer:

CANoe is required for test execution. The range of functions that can be used in vTESTstudio depends on the CANoe version.

- Minimum: CANoe version 8.5 SP5 / 9.0 / 10.0
- Recommended: CANoe version 11.0

### > Test data management:

- IBM Rational DOORS from Version 8.1
- IBM Rational DOORS NG / RQM
- Siemens Polarion ALM
- PTC Integrity
- Open interfaces for integration into third party systems

> **Bus systems:** CAN, LIN, FlexRay, Ethernet, WLAN, ARINC 429

> **Protocols:** J1939, KWP2000, UDS, K-Line

> **Calibration:** XCP

More information: [www.vector.com/vTESTstudio](http://www.vector.com/vTESTstudio)

The screenshot displays the vTESTstudio interface with four main editors:

- Test Table Editor:** Shows a table with columns for Command and Caption. The table contains test steps like 'Preparation', 'ResetSignals', 'State Change', 'Set', 'Accelerate/DecelerateToTargetSpeed', 'Wait', 'Check', and 'Completion'.
- Test Diagram Editor:** Displays a flowchart with nodes for 'Engine-On', 'Engine-Off', 'Doors are locked/unlocked', and 'Crash'. It includes logic for 'Manual locked/unlocked' and 'Velocity < LockVelocity'.
- Programming Editor:** Shows C code for a test function. The code includes comments and logic for accelerating and decelerating the vehicle speed based on target speed and current speed.
- State Diagram Editor:** Displays a state machine diagram with states like 'Unlocked', 'Locked', and 'Finalize'. Transitions are labeled with events like 'LockRequest-RequestLock' and 'LockRequest-RequestUnlock'.

Programming Editor

State Diagram Editor

Test Table Editor

Test Diagram Editor

Selected editors in vTESTstudio for table-based, graphical and programming-based creation of test sequences