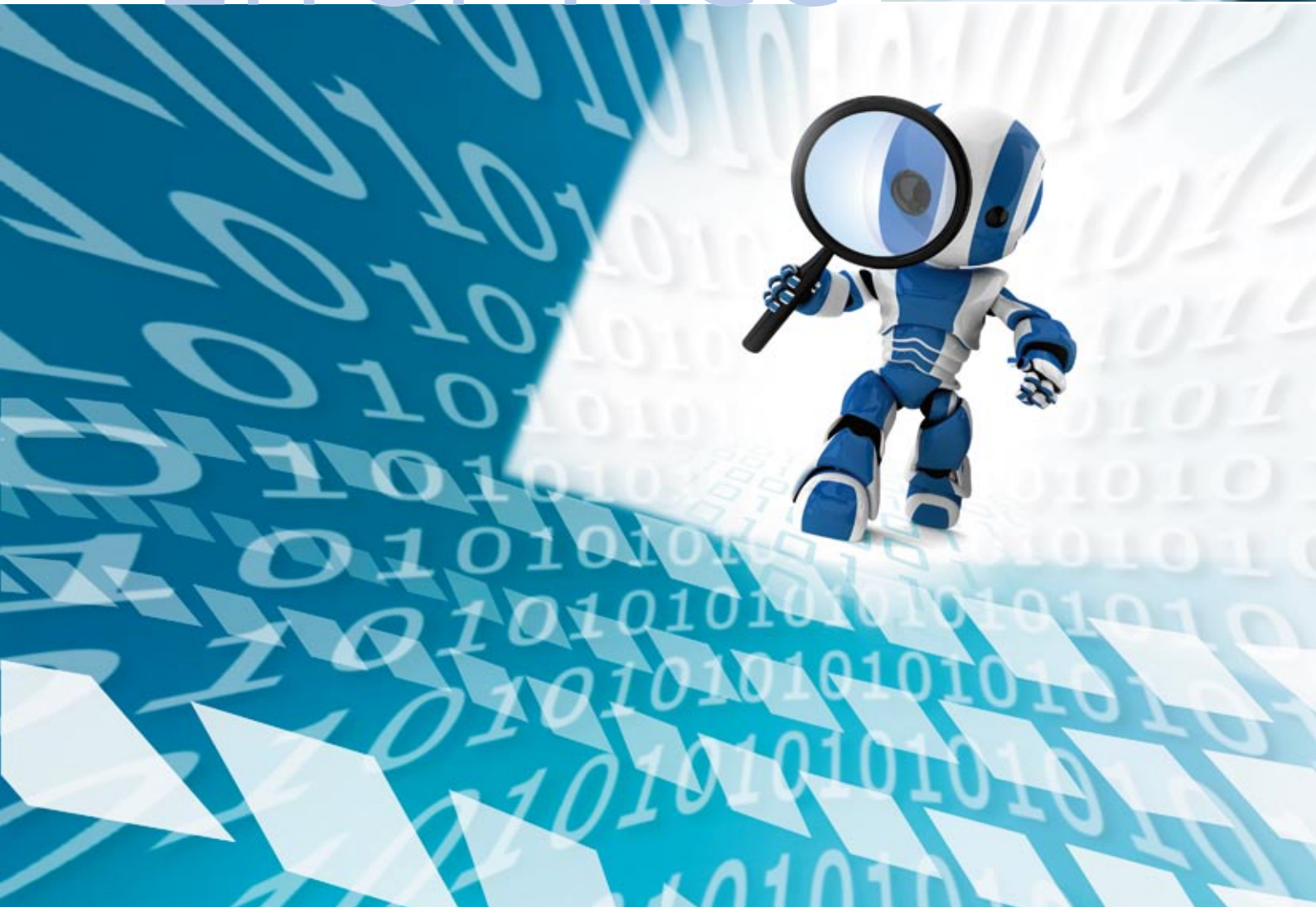


Completely Error-Free



Finding errors in complex test cases is like looking for a needle in a haystack. To make this difficult and intricate job easier, AutomationDesk 2.2 has a brand-new feature: the Debugger. This feature lets users execute and check their test sequences step by step, locating error sources and eliminating them.

Test automation is a standard part of hardware-in-the-loop (HIL) simulation, and as ECU functions become more complex, so do test cases.

AutomationDesk Version 2.2, dSPACE's test automation software, facilitates the checking of a large volume of test cases. The Find Inconsistencies function performs static analyses before test execution begins. And with the new Debugger, test sequences can be checked and corrected step by step while they are still being implemented.

Test the Test

Test sequences are described graphically in AutomationDesk. They consist of a large number of single test steps called blocks. Before the tests are even executed, users can run the Find Inconsistencies function to find and correct errors. The function analyzes the test structure and all the defined data objects, references, and code

The Debugger makes trouble-shooting fast and efficient, allowing the targeted observation of potential error sources.

fragments so that problems such as open references or syntax errors are detected automatically.

The next step is to run the test in offline mode to find errors in the test sequence without having to have an entire test environment available, such as an HIL simulator or calibration tools. This is done by defining default values to be returned when unavailable hardware or software is accessed. This procedure finds errors in the sequence or control structure such as erroneous conditions for If-Else branches or errors in loops, incorrect parameterizations, and run-time errors. When all these have been corrected, the test is executed on the HIL simulator. The offline mode means the HIL simulator is available for

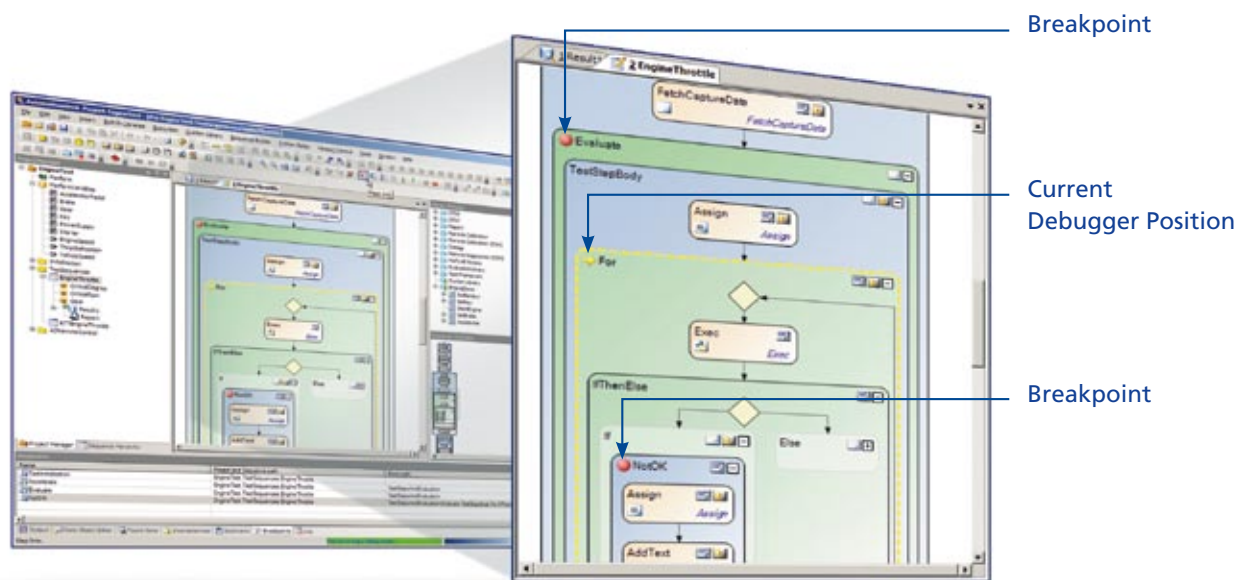
executing other tests during test development.

Debugging on the PC

When unexpected test results or errors occur during test execution – offline on a PC or online on a HIL simulator – the Debugger's step-wise procedure helps pinpoint the cause.

Developers can use the Debugger to insert break-points at any desired positions in the test sequence. When the sequence is run in debug mode, execution automatically stops when it reaches a breakpoint. The Debugger's toolbar then provides the following options:

- Continue running the test sequence up to the next breakpoint.



The Debugger stops the test execution at each breakpoint.



The Debugger's toolbar lets users check their test sequences step by step.

- Execute the test sequence step-wise.
- Either go down one level in complex, hierarchical test blocks to execute that level, or run the whole current hierarchy level.

The current variable values can be viewed throughout the debugging process. If they deviate, they can be changed to allow sequence testing to continue with correct values. Thus, users can check a test step by step and correct it if necessary as they go along.

Efficient Processes

When debugging has finished, users return to the normal execution mode to restart the test sequence. Step-by-step execution supports fast, efficient error-finding by allowing specific potential error sources to be investigated. Using the Find Inconsistencies function, the offline mode and the Debugger speeds up test development, enhances the quality of tests, and improves the overall test process.