SYNECT

- ECU test management and automation
- Efficient test processes and data management
- Support of agile software development
SYNECT
Data management and collaboration software for automated ECU testing

Highlights
- Close collaboration in globally distributed test teams
- Full traceability of test data
- Integrated variant management
- Optimized Hardware-in-the-loop (HIL) utilization through automation
- Direct connection to application lifecycle management (ALM) and test tools

Application Areas
SYNECT is a data management and collaboration tool with a focus on the efficient and automated verification and validation of ECU (electronic control unit) software. SYNECT helps you manage and automate testing in local as well as globally distributed teams and connects management processes, actual test work, and tools. SYNECT provides full traceability from requirements to test results. Powerful monitoring options, such as reports or dashboards, visualize the test status and progress. With its integrated variant management, SYNECT provides a systematic and transparent approach for testing ECU variants and versions. SYNECT furthermore automates test system setup, including HIL preparation and build processes, and enables agile development by supporting continuous testing. Using SYNECT will help you set up 24/7 testing and optimize HIL utilization.

Key Benefits
SYNECT enables collaboration and automation for ECU testing.
- Automate HIL testing and achieve more efficient HIL utilization
- Introduce model-in-the-loop (MIL) and software-in-the-loop (SIL) testing using the same tools and workflows as for HIL testing
- Establish full traceability from requirements to test results
- Integrate with ALM tools and processes to bridge the gap to actual test execution
- Monitor the test progress with customizable dashboards and comprehensive reports
Module Overview

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Base</td>
<td>￭ SYNECT module for working with multiple clients on a common server, including user and rights management</td>
</tr>
<tr>
<td></td>
<td>￭ Support for requirements management, signal and parameter management, and for connection/integration to ALM systems</td>
</tr>
<tr>
<td>SYNECT Test Management – Planning &amp; Execution (p. 13)</td>
<td>￭ SYNECT module for test planning, scheduling, and execution</td>
</tr>
<tr>
<td>SYNECT Test Management – Test Development (p. 12)</td>
<td>￭ SYNECT module for test case development and parameterization</td>
</tr>
<tr>
<td>SYNECT Model Management (p. 19)</td>
<td>￭ SYNECT module for managing models, their interfaces, parameters, and related files</td>
</tr>
<tr>
<td>SYNECT Variant Management (p. 18)</td>
<td>￭ SYNECT module for defining and managing variants as well as their impact on design and development data</td>
</tr>
<tr>
<td>SYNECT Workflow Management (p. 20)</td>
<td>￭ SYNECT module for automatic test system setup, i.e., for HIL simulation and modeling, by means of workflow creation</td>
</tr>
<tr>
<td></td>
<td>￭ Tool for continuous integration, from autocode generation to SIL and HIL simulation.</td>
</tr>
<tr>
<td></td>
<td>￭ Support for variant and parameter handling in highly complex build and test environments</td>
</tr>
<tr>
<td>Workflow Runner Package</td>
<td>Required for SYNECT Workflow Management</td>
</tr>
<tr>
<td></td>
<td>￭ Starter/Runner package manages the execution of workflows via the Starter user interface, automated jobs, MATLAB® or Python API</td>
</tr>
<tr>
<td></td>
<td>￭ Lets you start a custom workflow, e.g., to start dSPACE ControlDesk with the experiment associated with the selected variant configuration</td>
</tr>
<tr>
<td>Workflow Management Connector Package</td>
<td>Required for SYNECT Workflow Management</td>
</tr>
<tr>
<td></td>
<td>￭ Manages the connection between SYNECT and other dSPACE or third-party tools required for automatic test system preparation</td>
</tr>
<tr>
<td></td>
<td>￭ Handles connections to ConfigurationDesk, AutomationDesk, ControlDesk, ModelDesk, and MATLAB®/Simulink®</td>
</tr>
<tr>
<td>SYNECT AUD Test Authoring Framework (TAF) (p. 12)</td>
<td>￭ SYNECT-based test authoring and automated test execution with AutomationDesk (please see the AutomationDesk product information)</td>
</tr>
</tbody>
</table>

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td>SYNECT Test Management – Planning &amp; Execution</td>
<td>SYNECT_TM_PLAN_EXEC</td>
</tr>
<tr>
<td>SYNECT Test Management – Test Development</td>
<td>SYNECT_TM_DEV</td>
</tr>
<tr>
<td>SYNECT Model Management</td>
<td>SYNECT_MODEL_MGMT</td>
</tr>
<tr>
<td>SYNECT Variant Management</td>
<td>SYNECT_VARIANT_MGMT</td>
</tr>
<tr>
<td>SYNECT Workflow Management 1)</td>
<td>SYNECT_WORKFLOW_MGMT</td>
</tr>
<tr>
<td>SYNECT AUD Test Authoring Framework</td>
<td>SYNECT_AUD_TAF</td>
</tr>
</tbody>
</table>

1) Customers using SYNECT Workflow Management require at least one license of SYNECT Variant Management

SYNECT has a modular and scalable structure.
Relevant Software for SYNECT Workflow Management

<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td></td>
<td>SYNECT Variant Management</td>
<td>SYNECT_VARIANT_MGMT</td>
</tr>
<tr>
<td></td>
<td>Workflow Management Runner Package</td>
<td>WFM_WF_RUNNER</td>
</tr>
<tr>
<td></td>
<td>Workflow Management Connector Package</td>
<td>WFM_CONNECTORS</td>
</tr>
</tbody>
</table>

1) SYNECT Base or SYNECT Variant Management are not required on every workstation. However, they must be installed on at least one workstation for configuration in a multi-user scenario.

Relevant Software for SYNECT AUD Test Authoring Framework (TAF)

<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td></td>
<td>AutomationDesk</td>
<td>AUD_BASIC</td>
</tr>
</tbody>
</table>

Engineering Services and Process Consulting – The Complete dSPACE Offer

To make sure that dSPACE platforms, including the software tool chains, enable you to reach your validation and verification goals, our dSPACE Engineering Services team assists you in defining and setting up your testing tool chain (please see www.dSPACE.com/go/engineering).

With years of experience in developing best practices for various industries and customer-specific solutions, the dSPACE Process Consulting team uses its expertise to optimize existing processes so you are ready for ISO 26262 or Automotive SPICE and CMMI, and can implement new processes to tackle new challenges (please see www.dSPACE.com/go/consulting).
Use Cases – Working with SYNECT

Requirement-Based Testing

The number of requirements in modern ECU development is continuously increasing, and end users want to know how many of these requirements have been successfully validated. Additionally, safety-related standards demand full traceability, from requirements to all relevant test data like test cases and test case results.

Challenges
- Establishing a traceable and efficient process for verifying the rising number of requirements
- Connecting the requirements with the test activities in the expert test tools
- Ensuring that all requirements are verified by tests and that the tests are passed
- Tracking the impact of requirement changes
- Generating standard compliance reports of the requirements test coverage without error-prone manual copying

Solution
SYNECT helps you bridge the gap between your requirements management system and the tools you are currently using for test execution and automation on HIL and SIL platforms. As a centralized database management system, SYNECT stores all the relevant data in a central location, which enables global teams to collaborate, access data from any location, and perform testing on a local workstation.
SYNECT integrates with various commonly used requirements management systems, enabling full traceability from requirements to test results to achieve requirement-based testing. Requirements coverage analysis visualizes the current testing progress.

Some widely used requirements management systems that are integrated with SYNECT are:
- IBM® Rational DOORS® or DOORS NG®
- PTC® Integrity®
- Microsoft® Team Foundation Server
- Microsoft® Excel®
- ReqIF™
- Siemens Polarion ALM
- HP ALM

SYNECT supports the following requirements-based actions:
- Deriving and implementing tests, and managing the test cases
- Planning and executing tests automatically on a local or remote machine
- Achieving full traceability from the requirements to the test case and the test results

SYNECT supports the following process iterations:
- Monitoring and analyzing the test coverage for requirements
- Identifying which test cases are affected when requirements change

**Benefits**
- Full traceability from requirements to test cases, test parameters, and test results – directly supporting ISO 26262 compliance
- Integration of a wide range of requirement management and test tools, and automatic control of test execution with SYNECT
- Collaboration between globally distributed test teams

<table>
<thead>
<tr>
<th>Required Products</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td>SYNECT Test Management – Planning &amp; Execution</td>
<td>SYNECT_TM_PLAN_EXEC</td>
</tr>
<tr>
<td>SYNECT Test Management – Test Development</td>
<td>SYNECT_TM_DEV</td>
</tr>
</tbody>
</table>
24/7 Testing on Different Platforms

With the trend towards shorter development cycles and higher availability of software builds, the number of tests is continuously rising. This calls for an optimal utilization, i.e., 24/7 usage, of the HIL system to achieve the required test throughput. Furthermore, it is important to reduce the test load of the HIL systems by performing functional tests on SIL test platforms.

**Challenges**

- Managing a large number of test executions on several test platforms and systems
- Planning the availability of the HIL test systems
- Using tests seamlessly on HIL and SIL platforms

**Solution**

SYNECT supports the planning and scheduling of test executions worldwide, allowing to distribute test executions globally. The teams stay informed about the test status and test system status with SYNECT’s test system booking and resource management. Test systems automatically start test executions at the scheduled time or as soon as they have the required capacity. Tests that can be reused on HIL and SIL platforms are generated in SYNECT by using the standardized XIL API and proper variable mapping (XIL mapping). This reduces HIL preparation and configuration time to a minimum with fully automated, reproducible, and reliable workflows managed in SYNECT (p. 20).
Benefits

- Maximized HIL utilization by enabling 24/7, overnight and weekend testing
- Configuration and control of HIL and SIL test farms
- Reduction of HIL downtime by introducing automated HIL preparation

SYNECT optimizes the use of the available HIL systems.

<table>
<thead>
<tr>
<th>Required Products</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td>SYNECT Test Management – Planning &amp; Execution</td>
<td>SYNECT_TM_PLAN_EXEC</td>
</tr>
<tr>
<td>SYNECT Variant Management</td>
<td>SYNECT_VARIANT_MGMT</td>
</tr>
<tr>
<td>SYNECT Workflow Management</td>
<td>SYNECT_WORKFLOW_MGMT</td>
</tr>
<tr>
<td>Workflow Management Runner Package</td>
<td>WFM_WF_RUNNER</td>
</tr>
<tr>
<td>Workflow Management Connector Package</td>
<td>WFM_CONNECTORS</td>
</tr>
</tbody>
</table>
Variant-Based Automation and Configuration

ECU software is often built to support multiple variants, e.g., different vehicle platforms, engine or gearbox variants. The number of different vehicle versions has been steadily increasing for many years, and with it the number of vehicle and ECU combinations to be tested. Systematic support for variant handling in HIL testing and an established procedure for the development and test of electronic control units (ECUs) is therefore indispensable.

Challenges:

- Handling the rising number of variants
- Automating complex and repetitive HIL and software build preparation workflows
- Managing the flow of data from different development and test tools, and introducing a standardized process

Solution

The SYNECT API and the software’s versatility allow for the integration of all engineering tools involved in the HIL process, e.g., tools for ECU calibration, ECU flashing, or downloading and parameterizing the real-time application. The integrated SYNECT variant management provides a systematic approach for defining and handling variants and makes it easy to obtain the correct data, e.g., parameters or tests, for a specific software variant. Fully automated workflows in SYNECT perform all the preparation and configuration steps for the specific variant. SYNECT Workflow management is well established at major OEMs and a range of companies performing ECU tests.

Benefits

Use of standard off-the-shelf workflows for:
- dSPACE Real-Time Interface, MATLAB®/Simulink® variants
- Build process management for HIL systems and HIL farms
- Online variant management in ModelDesk

Project-specific workflows for:
- SCALEXIO/ConfigurationDesk variants
- HIL control with ControlDesk
- Providing variant-dependent data to AutomationDesk
Reducing downtime with automated workflows.

<table>
<thead>
<tr>
<th>Required Products</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td>SYNECT Test Management – Planning &amp; Execution(^1)</td>
<td>SYNECT_TM_PLAN_EXEC</td>
</tr>
<tr>
<td>SYNECT Variant Management</td>
<td>SYNECT_VARIANT_MGMT</td>
</tr>
<tr>
<td>SYNECT Workflow Management</td>
<td>SYNECT_WORKFLOW_MGMT</td>
</tr>
<tr>
<td>Workflow Management Runner Package</td>
<td>WFM_WF_RUNNER</td>
</tr>
<tr>
<td>Workflow Management Connector Package</td>
<td>WFM_CONNECTORS</td>
</tr>
</tbody>
</table>

\(^1\) Required only for automated testing with SYNECT

Continuous Integration and Testing for SIL and HIL

More complex ECU functionality and the growing number of ECU variants increase the amount and complexity of testing for ECU software validation. To implement an efficient test process that masters this challenge, software developers and testers have to cooperate closely. For this purpose, processes already established in general software development, such as agile development with continuous integration and continuous testing, are making their way into ECU software development. This calls for test management for ECU testing and an intelligent process automation tool, such as SYNECT.

**Challenges**
- Meeting tight and accelerating release schedules
- Ensuring that the correct models, including variant-specific parts and parameterizations, are used for the automatic build
- Ensuring that new or modified models are automatically included in nightly builds
- Having the right ECU software build, real-time application, and parameterization on the test system and trigger the relevant tests
SYNECT ensures that the right chain of actions is processed when individual artifacts are modified. It also evaluates the relevant variant dependencies. Actions include the rebuild of model containers after a model is checked in, or the use of these containers during a recurring nightly build of the real-time application. It is also possible that a new version of ECU software, provided only via a configuration management system, must trigger a set of regression tests using the latest build of the real-time application on the HIL system and the proper parameterization corresponding to the ECU software variant. The high automation SYNECT offers in this regard speeds up the development and test cycle, reduces error-prone manual work, and automatically provides full traceability between all the artifacts used in the process.

**Benefits**

- Introduce continuous integration and testing as a company-wide process for ECU software verification and validation
- Speed up continuous integration by using variant-dependent data management that verifies only modified variants
- Schedule test system preparation as well as overnight and weekend tests
- Reduce the testing complexity caused by software variants and versions by providing pre-configured and easy-to-use workflows for all testers

<table>
<thead>
<tr>
<th>Required Products</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Base</td>
<td>SYNECT_BASE</td>
</tr>
<tr>
<td>SYNECT Test Management – Planning &amp; Execution</td>
<td>SYNECT_TM_PLAN_EXEC</td>
</tr>
<tr>
<td>SYNECT Variant Management</td>
<td>SYNECT_VARIANT_MGMT</td>
</tr>
<tr>
<td>SYNECT Workflow Management</td>
<td>SYNECT_WORKFLOW_MGMT</td>
</tr>
<tr>
<td>Workflow Management Runner Package</td>
<td>WFM_WF_RUNNER</td>
</tr>
<tr>
<td>Workflow Management Connector Package</td>
<td>WFM_CONNECTORS</td>
</tr>
</tbody>
</table>
SYNECT Key Features

Test Management for Test Developers

The SYNECT Test Management – Test Development module gives test developers the means to develop tests efficiently and traceably. Tests can be easily registered in SYNECT, since SYNECT provides direct integration with common test tools (Test Tool Integration, p. 14). Test cases in SYNECT can be directly associated with system or test requirements, providing a direct overview of requirements coverage. If formalized test requirements are used, test cases can also be automatically derived from the textual requirements. Different sets of parameter values for different test variants can be easily stored and referenced for future testing. After the test implementation is finished, test developers can simply update the implementation status and release the tests for testing. Throughout the test project, the test developers have a clear overview of the test performance in different test stages and test environments.

Efficient and traceable test development

- Use expert test authoring tools, such as AutomationDesk, NI TestStand, Vector CANoe/TESTStudio, Python or MATLAB® scripting and publish the tests in SYNECT Test Management
- Store and manage test case parameters
- Document variant dependencies and the applicability of the developed tests for automatic consistency checks during test planning
- Investigate the results of assigned tests
- Develop requirements-based tests, including the automatic derivation of test definitions from formalized requirements and an automatic requirements coverage overview

**dSPACE Test Authoring Framework (TAF)**

dSPACE has developed a solution that simplifies the way you use AutomationDesk and lets you author test cases and test steps faster and easier. With the dSPACE Test Authoring Framework (TAF), test developers can intuitively drag & drop test steps from libraries and create test cases.

With a step-based test authoring approach, the TAF introduces a new tabular user interface.
Even testers with little AutomationDesk experience are able to quickly start authoring basic tests thanks to a new tabular user interface format. The step-based approach is easy to use and lets users quickly start writing basic tests. Some immediate benefits provided by the new TAF:

- Introduction of a sophisticated, tabular user interface for authoring test steps
- Increased flexibility to integrate custom-developed tests/actions
- Reusability of the same framework for multiple projects, resulting in less maintenance
- Optimized approach to organizing custom libraries and test cases
- A platform-independent framework that works with different HIL models for the same test cases

Test Planning and Execution

The SYNECT Test Management - Planning & Execution module helps you manage and automate test execution. Plan tests for different test environments or different versions of the system under test, and SYNECT checks that the correct tests are executed with the correct parameterization. Execute tests on specific test systems or queue tests to be automatically executed by available test systems, whether the test systems are next door or in another country. Execute tests interactively or plan and schedule automatic and recurring test executions. The built-in execution queue displays the status of all executions on the different test systems.

**Powerful test planning and scheduling**

- Centrally manage all test activities across different test levels and tools in a development project
- Systematically plan and control test execution
- Track and evaluate the test progress in its entirety
- Monitor test results, archive the test result history, and generate reports for a comprehensive overview
- Trace requirements up to test results by connecting to requirements management tools
- Check the requirement coverage and assess the maturity of the system under test at any time
- Manage test execution on different platforms: dSPACE HIL systems, Simulink®, VEOS, MicroAutoBox, etc.
One main goal of our customers is to ensure maximum efficiency of the available test systems. When you execute tests with SYNECT, the test execution durations are automatically measured and stored for each test. This information can be used for future test planning to facilitate decisions on how many tests can be run within the next two hours, overnight, or over the weekend.

The full traceability from requirements to test case results helps you analyze which tests have failed and which requirements are affected.

Using SYNECT Test Management in combination with the integrated SYNECT Workflow Management module (p.20), you can fully automate all steps required for HIL system setup test preparation, execution, and evaluation, making testing much more efficient.

Test Tool Integration

Our customers’ verification and validation strategies comprise different test stages on different test platforms, ranging from MIL and SIL platforms to highly customized HIL test benches to the testing of complete vehicles. This calls for a common solution to manage all test results from multiple test platforms and test automation tools. SYNECT is the one tool to manage all tests and test results throughout the different test stages. SYNECT provides direct tool integration for various test authoring and test execution tools, such as dSPACE AutomationDesk, NI TestStand, Vector vTESTstudio/CANoe, and Python. SYNECT automatically collects the test results immediately after the test execution, no manual report or verdict copying is required.

SYNECT also provides open interfaces to seamlessly integrate further test authoring tools. If you cannot find a tool in the list of supported tools, do not hesitate to contact us. SYNECT has an open API and we can integrate your tool of choice into the tool chain.

With its powerful analysis and reporting options, SYNECT provides an overview of the entire test process across test tools and test platforms (Dashboards and Reporting, p. 16).
Application Lifecycle Management Integration

Application lifecycle management (ALM) tools are often predetermined company-wide. ALM tools like IBM® Collaborative Lifecycle Management (CLM), PTC® Integrity® Lifecycle Manager (ILM), and Siemens Polarion ALM are used to manage functional requirements, test specifications, software configurations, and defects. These tools have limitations regarding test tool integration and test automation, e.g., they do not have access to HIL test results. These limitations cause extra manual effort to achieve traceability.

SYNECT bridges the gap between widely used ALM solutions and testing, regardless of which test tool or platform is being used. In combination with the extensive test tool integration, SYNECT and the ALM tool provide a comprehensive tool chain for your ECU testing projects.

SYNECT provides support for the following ALM systems:
- IBM® Collaborative Lifecycle Management (CLM), including IBM® DOORS® and IBM® DOORS NG®
- PTC® Integrity® Lifecycle Manager
- Siemens Polarion ALM
- HP ALM

For customers who do not have a full ALM suite, SYNECT integrates into other tools:
- IBM® Rational DOORS® or DOORS NG®
- Microsoft® Excel®
- Microsoft® Team Foundation Server (TFS)
- Atlassian Jira

If you cannot find a tool in the list of supported tools, do not hesitate to contact us. SYNECT has an open API and we can integrate your tool of choice into the tool chain.
Dashboards and Reporting

Monitoring testing activities and test progress can be a challenge, especially when many people across different teams, departments, and countries are involved. To make monitoring easier, SYNECT offers dashboards to visualize the test project status. Dashboards can be highly customized in order to display all relevant project metrics. Examples of typical project metrics include:

- Total number of system requirements and number of requirements linked to tests over time
- Total number of tests and number of tests per implementation status over time
- Total number of tests and number of tests per last verdict over time
- Number of failed tests per test criticality
- Number of failed and not reviewed tests
- Test status per test environment variant
- Test status per used test tool

The dashboards are accessible without a SYNECT client installation, only a compatible Internet browser is required. Supported browsers are Microsoft Edge, Google Chrome, and Mozilla Firefox.

To get the final approval for a software release, the test status has to be documented – in compliance with the process and ISO standard. SYNECT’s reporting feature helps you document and share the test status, including an overview of planned and completed tests. The requirement test coverage and requirement test status are available at the click of a button. Requirement coverage results can also be exported to a customizable report.
Coverage Summary

<table>
<thead>
<tr>
<th>Status</th>
<th>Amount</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>Not Linked</td>
<td>0</td>
<td>0.00 %</td>
</tr>
<tr>
<td>Total</td>
<td>4</td>
<td>100.00 %</td>
</tr>
</tbody>
</table>

Verdict Summary

<table>
<thead>
<tr>
<th>Verdict</th>
<th>Amount</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>3</td>
<td>75.00 %</td>
</tr>
<tr>
<td>Blocked</td>
<td>0</td>
<td>0.00 %</td>
</tr>
<tr>
<td>Skipped</td>
<td>0</td>
<td>0.00 %</td>
</tr>
<tr>
<td>Undefined</td>
<td>0</td>
<td>0.00 %</td>
</tr>
<tr>
<td>NoResult</td>
<td>0</td>
<td>0.00 %</td>
</tr>
<tr>
<td>Failed</td>
<td>1</td>
<td>25.00 %</td>
</tr>
<tr>
<td>Exception</td>
<td>0</td>
<td>0.00 %</td>
</tr>
<tr>
<td>Crashed</td>
<td>0</td>
<td>0.00 %</td>
</tr>
</tbody>
</table>

Requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Verdict</th>
<th>Test Case: Name</th>
<th>Test Case: Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open command safety-unlocks car (1)</td>
<td>✓</td>
<td>Open command safety-unlocks car (1)</td>
<td>Preparation: 1.) Car is in safe mode. 2.) No incoming lock or unlock commands. Test Steps: [A] Issue remote unlock command to central locking controller. [R] Central controller switches from safety mode to lock mode. [R] Central controller switches from lock mode to unlock mode. Final State: Car is unlocked.</td>
</tr>
<tr>
<td>Close command safety-locks car (1)</td>
<td>✓</td>
<td>Close command safety-locks car (1)</td>
<td>Preparation: 1.) Car is unlocked. 2.) No incoming lock or unlock commands. Test Steps: [A] Issue remote locking command to central locking controller. [R] The controller first locks all doors and afterwards safety-locks them. Final State: Car is safety-locked.</td>
</tr>
<tr>
<td>Car safety-locks on unintentional unlock (1)</td>
<td>✓</td>
<td>Car safety-locks on unintentional unlock (1)</td>
<td>Preparation: 1.) Car is locked. 2.) No incoming lock or unlock commands. Test Steps: [A] Issue remote unlock command to the central locking controller. [R] The central locking controller unlocks all doors. [R] After 20 seconds, the central locking controller first locks and then safety-locks the car. Final State: Car is locked.</td>
</tr>
<tr>
<td>Car unlocks on crash detection (1)</td>
<td>✗</td>
<td>Car unlocks on crash detection (1)</td>
<td>Preparation: 1.) Car is locked. Test Steps: [A] Issue crash signal to central locking controller. [R] The controller unlocks all doors immediately. Final State: Car is unlocked.</td>
</tr>
</tbody>
</table>

Requirements coverage report.
Managing Variant Information

In ECU testing, the number of variants and the necessity of testing more and more software releases in a short time requires a powerful variant management solution. SYNECT provides a systematic approach to variant handling and lets you define a central variant model, including constraints for individual variants. Furthermore, the individual data handled in SYNECT, e.g. test cases, parameters, and models, can be easily connected to the variants, thus enabling SYNECT to provide the right set of data in the appropriate step of the test process. For example, SYNECT automatically provides the relevant parameter set for ECU calibration, the required real-time application on the simulation platform, or the appropriate set of test cases for each test execution.

![Variant model defining variants, including subvariants.](image1)

![User-defined list of variant configurations for a project.](image2)

Signal and Parameter Management

SYNECT supports the central management of signals, parameters, and parameter sets throughout the entire ECU development process. It supports commonly used file formats and standards, and provides direct connections to engineering tools, so you can easily store and retrieve signals and parameters.

With the import and export adapters, you can swiftly exchange parameters with other tools. The following formats are supported by default: dSPACE TargetLink Data Dictionary (DD), DCM, A2L, HEX, MOT, CDFX, and PAR. Parameter sources based on Microsoft Excel can be integrated as well.

![Parameter import and export plug-ins.](image3)
Managing variant-specific parameters in SYNECT

You can specify variant or value dependencies for each parameter, which means that a parameter exists only for a certain variant or that a parameter has different values in different variants. This makes it easy to manage different parameter values for variant configurations. Parameter management can be used for test case parameterization in the same way.

Managing Models in SYNECT

SYNECT Model Management lets you manage models independently of the modeling tool. A model in SYNECT Model Management is described by the model’s metadata, including the model’s interfaces and references to the model files in a file system. The interface information includes the information on the model’s signal ports and bus access requests, down to the list of exchanged signals with their units, type definitions, and scalings. The metadata to identify the model is completely customizable. For example, it can be extended with required process information, such as the responsible model developer or the model’s maturity. To register the model information in SYNECT Model Management, the tool provides open and customizable interfaces. It supports the import of model information from Simulink models, XML or Excel files as well as from model containers, such as Functional Mock-up Units (FMU), Simulink implementation containers (SIC), V-ECUs, and bus implementation containers (BSC). Via the open interfaces, it is easy to connect other modeling tools.

SYNECT Model Management facilitates the exchange and reuse of models by integrating in existing processes and by supporting multiple file management systems. The actual model implementation files remain in the file management systems, but they are encapsulated as implementation objects in SYNECT Model Management. The model metadata then references these files and the specific versions. The references to these model implementation files in a configuration management system can be attributed with variant information from a central variant model.

SYNECT Model Management can be used for integrating models for simulation and testing. For example, a set of model containers can be imported as component models of an overall system model. In a dedicated graphical view, the system model can be visualized, and the model components can be connected either interactively or automatically. The automatic connection algorithm also supports matching based on regular expressions. A thereby defined overall system model can be exported and a convenience functionality to automatically build an Offline Simulation Application (OSA) for dSPACE VEOS is included.
SYNECT Model Management provides the functionality to validate the model metadata and the model interfaces both for a single model as well as for an overall system model. It is possible to check for process compliance, naming conventions, existence of information, and interface consistency. A set of sample validation rules is provided, but the framework lets you freely define custom validation rules.

**Automating Complex Workflows Including Variants with SYNECT Workflow Management**

In order to achieve a high degree of test platform utilization (p. 7), automated test system configuration (p. 9), or continuous testing and continuous integration for SIL and HIL testing (p. 10), advanced process automation is required.
SYNECT Workflow Management lets you create workflows to automate recurring processes. Workflows consist of reusable steps. You can choose from a large set of predefined steps that support the dSPACE tool chain. Examples for workflow steps include starting a dSPACE tool, such as AutomationDesk, ControlDesk or ConfigurationDesk, loading applications and layouts, or building a real-time application. Workflow steps for common file operations are also available. You can also define your own workflow steps.

All workflow steps can retrieve and evaluate variant-dependent parameters. Therefore, the workflow can be executed for a specific variant and ensures that the correct layouts, build artifacts or working folders for the specific variant are selected, for example. With this mechanism, it is possible to automate highly complex test systems with a large number of variants and multiple software interactions. Once you configured a variant model and variant-dependent workflows, they are available company-wide in the central database.

Workflows can be triggered automatically, for example, for the automatic overnight preparation of large real-time testing applications. But workflows can also be used to assist less experienced users in successfully configuring and preparing an HIL test system. The users just have to start a simple user interface, the Workflow Starter, to interactively choose a predefined workflow and execute the workflow for the required variant.
Configuration Management Support

Automated processes typically rely on data in configuration management systems to make the workflow reproducible. Therefore, workflows in SYNECT also support configuration management file and folder operations. For example, a workflow can make sure that a sandbox is synchronized with a specific version from the configuration management system. Since this configuration management version information can be automatically documented, this level of process automation allows for the reproducibility required for tests of safety-critical systems, among other things. If required, a workflow can also automatically commit or check in the workflow results, such as reports, documentation, or build result artifacts, to a configuration management system. Any configuration management system can be integrated by means of SYNECT add-ons. For example, Git operations such as pull and push can be made available as dedicated workflow steps.

By using predefined workflows in combination with the open SYNECT API, it is possible to trigger workflows on the basis of actions or status changes in a configuration management system. For example, a build can be automatically started once a new model version has been checked in.

Due to the easy configuration of workflow steps in SYNECT Workflow Management, it is also possible to support multiple configuration management tools. For example, it is possible to reconfigure sandboxes for multiple configuration management systems, such as Apache™ Subversion®, Git, or PTC® Integrity® Source, to run a dedicated build. The final build result can also be stored in a different dedicated repository manager, such as JFrog Artifactory so that binary artifacts are not checked into an ASCII-specialized configuration management tool. If you cannot find a tool among the supported tools mentioned here, do not hesitate to contact us. SYNECT has an open API and we can integrate your tool of choice into the tool chain.
Issue Tracking Management Support

As for configuration management systems, it is possible to trigger workflows on the basis of actions or status changes in issue tracking systems like Atlassian JIRA, Redmine, Microsoft® TFS, or Microsoft® Sharepoint. For example, closing an issue can trigger a dependent build process or it can trigger a test execution.

Workflows from SYNECT Workflow Management can also trigger actions in the issue tracking system:

- A failed build workflow can create a new bug item that is assigned to the responsible modeling engineer
- A failed test execution can create a new bug item that is assigned to the responsible test manager
- A successful build can create a new test issue that is assigned to the test manager

APIs, Standards, and Openness

SYNECT integrates easily into existing tool chains and processes and even allows for automating process steps via powerful, open, and completely documented APIs. It provides a COM API that enables you to automate standard GUI actions, such as data imports and exports, report generation, and consistency checks. Furthermore, customer-specific actions and workflows can be integrated into SYNECT by extending its user interface with new ribbons, controls, and custom context menus. Actions can also be triggered automatically by client or server events. External tools can use the COM API for integration by using programming languages like Python, MATLAB®, C++, or C#.

SYNECT also provides an OData API for direct data access without a SYNECT client. This can be useful for web applications and external business intelligence tools, for example. This OData API enables full read and write access to SYNECT data for all authenticated users, provided they have the required permissions.

Finally, SYNECT supports the standard OSLC for seamless integration and data exchange with application lifecycle management tools, such as IBM® CLM.