SYNECT®

- Data management software for model-based development and ECU testing
- Modules for test management, signal & parameter management, model management, and variant management
SYNECT®

Data management software for model-based development and ECU testing

Highlights

- For model-based development and ECU testing
- Integrated variant management
- Direct connection to engineering tools
- Traceability between requirements, models, parameters, and test cases
- Open API for integration into existing IT infrastructure and workflows
- High scalability to variably-sized, distributed teams

Application Areas

SYNECT is a data management and collaboration software tool with a special focus on model-based development and ECU testing. The software is designed to help you manage data throughout the entire development process. This data can include models, signals, parameters, tests, test results, and more. SYNECT also handles data dependencies, versions and variants, as well as links to the underlying requirements. One key aspect of SYNECT is direct connection to engineering tools, e.g., MATLAB®, Simulink®, TargetLink®, or AutomationDesk, and application/product lifecycle management systems (ALM/PLM) so that you can work with your preferred tools and seamlessly exchange data. SYNECT is ideal for automotives, aerospace, industrial automation and medical engineering – and wherever embedded systems are developed through model-based design.

Key Benefits

SYNECT integrates all your development data into one single environment.

- Directly store and reuse the data objects found in your model-based development and ECU testing processes.
- Work with your preferred tools in every development phase and benefit from easy data flow, because SYNECT integrates with engineering tools and supports standards commonly used in model-based development.
- Stay on top of the escalating number of variants with SYNECT’s built-in variant management.
- Benefit from fast deployment due to the modular and scalable product architecture.
- Customize SYNECT to your needs and integrate it into your IT architecture thanks to SYNECT’s open structure.
- Benefit from SYNECT’s open architecture to implement company-wide processes via connections to ALM/PLM tools

Modular, Scalable, and Secure

SYNECT focuses on the efficient, integrated management of all your design and development data throughout the model-based development process, yet it can also be scaled to fit specific use cases. SYNECT has modules for individual areas such as test management, model management, signal and parameter management, and variant management. This means that you can introduce SYNECT step-by-step and in specific teams such as testing groups or function development teams, and scale your system to the team size. From local to global, SYNECT grows with your needs. The platform supports access control and easy sign on, for example via Windows® authentication, and offers secure communication (e.g., automatic transaction protection for server scripts via SYNECT API), version control, and automation. Further secure mechanisms come with the easy integration in Windows Server Systems (Internet Information Services, IIS).
Module Overview

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Test Management (p. 4)</td>
<td>SYNECT module for the intuitive management, planning, monitoring and evaluation of all test activities</td>
</tr>
<tr>
<td>SYNECT Signal &amp; Parameter Management (p. 9)</td>
<td>SYNECT module for managing signals, parameters and parameter sets for controller functions, ECU software and also plant models throughout the entire development process</td>
</tr>
<tr>
<td>SYNECT Model Management (p. 12)</td>
<td>SYNECT module for managing models together with their interfaces, parameters and related files</td>
</tr>
<tr>
<td>SYNECT Variant Management (p. 16)</td>
<td>SYNECT module for defining and managing variants and their impact on design and development data</td>
</tr>
<tr>
<td>SYNECT Base</td>
<td>SYNECT module for working with multiple clients on a common server, for user and rights management, and for fine-grained version and configuration management</td>
</tr>
</tbody>
</table>

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNECT Test Management</td>
<td>SYNECT_TEST_MGMT</td>
</tr>
<tr>
<td>SYNECT Signal &amp; Parameter Management</td>
<td>SYNECT_PAR_MGMT</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 Base</td>
<td>SYNECT_BASE</td>
</tr>
</tbody>
</table>

Additionally, dSPACE offers the SYNECT-based Workflow Management solution, which lets you handle complex, variant-based workflows, especially for HIL and modeling use cases.

SYNECT has a modular and scalable structure, providing several functional modules.
SYNECT Test Management

**Highlights**
- Manage MIL/SIL/PIL/HIL tests and test data
- Off-the-shelf integration with AutomationDesk and BTC EmbeddedTester®
- Connection to MATLAB® and Python
- Monitor, analyze and visualize test results
- Interface to requirements management
- Test case parameterization
- Integrated variant support

**Application Areas**
SYNECT Test Management supports the intuitive management, planning, monitoring and evaluation of all test activities in ECU development projects. The software provides special support for model-in-the-loop (MIL), software-in-the-loop (SIL), processor-in-the-loop (PIL) and hardware-in-the-loop (HIL) testing by directly connecting to test automation tools. Your whole team’s tests, test data and test results are stored centrally and made available for analysis and status monitoring independently of the test system and tool that you use.

**Key Benefits**
SYNECT Test Management handles all the information on your tests and supports the entire test process.

**Master the complexity:**
- Centrally manage all test activities across different test levels and tools in your development project.
- Trace your requirements up to test results by interfacing to requirements management tools.
- Monitor your test results, archive your test result history, and generate reports for a comprehensive overview.
- Let all users work on the same, consistent set of tests, test data, and test results with multi-user support.

**Test efficiently:**
- Systematically plan and control test execution.
- Check the requirement coverage and assess the maturity of your system under test at any time.
- Use existing test authoring and execution tools such as AutomationDesk, BTC EmbeddedTester®, Python scripting or MATLAB® scripting as a basis for test cases in SYNECT Test Management.
- Manage test execution on different platforms: dSPACE HIL systems, Simulink®, TargetLink®, VEOS®, MicroAutoBox II, etc.
- Adapt the system to your small or even large, globally distributed teams with advanced, scalable client-server architecture.
**Working with SYNECT Test Management**

Manage all test activities with SYNECT Test Management.

**Supported Tools and Interfaces for SYNECT Test Management**

<table>
<thead>
<tr>
<th>SYNECT Test Management</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Management</strong></td>
</tr>
<tr>
<td>Tool / Interface:</td>
</tr>
<tr>
<td>- IBM® Rational® DOORS</td>
</tr>
<tr>
<td>- PTC® Integrity</td>
</tr>
<tr>
<td>- Microsoft® Excel®</td>
</tr>
<tr>
<td>- Microsoft Team Foundation Server</td>
</tr>
<tr>
<td>- ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Authoring and Execution Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool / Interface:</td>
</tr>
<tr>
<td>- AutomationDesk</td>
</tr>
<tr>
<td>- BTC EmbeddedTester®</td>
</tr>
<tr>
<td>- Python scripting</td>
</tr>
<tr>
<td>- MATLAB® scripting</td>
</tr>
<tr>
<td>- Microsoft® Excel®</td>
</tr>
<tr>
<td>- Vector CANoe</td>
</tr>
<tr>
<td>- Vector vTESTstudio</td>
</tr>
<tr>
<td>- ...</td>
</tr>
</tbody>
</table>
Importing Test Cases
When you define your tests, you can reuse existing tests by importing them to SYNECT Test Management. You can source data from several tools: test lists from Microsoft® Excel®, and test implementations from AutomationDesk, BTC EmbeddedTester®, MATLAB®, or Python. Tests in SYNECT come with a standard set of attributes (e.g., test priority, test platform, test manager, date last executed), but you can also define your own attributes, of course. And you can define your own test adapters, so you can use virtually any of your testing tools or simulation platforms.

Traceability to Requirements and Test Specifications
Thanks to the requirements handling support (p. 21), test cases can be linked to requirements. You can use this to conduct traceability analyses, check the completeness of a test plan, etc. SYNECT Test Management takes advantage of SYNECT’s automation features by automatically resynchronizing requirements or automatically creating test cases from formally written test specifications. After updating requirements, with the help of SYNECT you can analyze if test cases linked to requirements also need to be changed.

Versioning and Variant Support
Each test case can be placed under version management. For example, test cases can be released so that they can no longer be modified. This lets you reuse existing test cases in new test plans (see Versioning on p. 20). To manage the test data for different variants, you can specify variant dependencies and select which variant configurations will be supported. You can work with variants by using this module in combination with SYNECT Variant Management (p. 16).

You can use filters and conveniently group or sort your tests for an optimal overview.
Test Planning and Execution
SYNECT Test Management easily integrates with test automation tools such as dSPACE AutomationDesk and BTC EmbeddedTester. It does this via adapters that control the test automation tool.

To carry out tests, you set up execution plans and add previously defined tests to them. An execution plan is a collection of tests that have to be executed, and it can be based on various criteria: for example, it can include all your tests for one variant or for one ECU function. Execution plans can also be defined using queries that will select test cases which fulfill a defined criterion. Your test tools use the execution plan to run the tests and return the results to SYNECT Test Management, where you can view the overall tests results and assess the progress of the test.

Dialog for defining test execution plans.
Test Result, Test Report and Requirements Coverage Views

Monitoring testing activities and test progress can be a challenge, especially when many people are involved. SYNECT Test Management helps you keep track of all tests, whether planned or already executed. An interactive evaluation functionality and the ability to generate and export reports lets you evaluate results systematically. For example, you can display selected test attributes and use them for actions such as creating an error burndown chart for your project. A requirements coverage view is available at a mouse click, so you can see which requirements were already tested and what the associated verdicts were. Requirements coverage results can also be exported in a customized report.

Example of a result evaluation, showing test progress over several calendar weeks.

Requirements coverage view.
SYNECT Signal & Parameter Management

Highlights
- Manage signals, parameters, and parameter sets
- For controller software development and plant models
- Interface to Simulink® and TargetLink®
- Support of standards and standard file formats
- Interface to calibration and experiment tools
- Integrated variant support

Application Areas
SYNECT Signal & Parameter Management is used for the central management of signals, parameters and parameter sets throughout the entire model-based development process. It can handle the signals and parameters of controller functions, ECU software and also plant models. It supports commonly used file formats and standards and provides direct connections to engineering tools, so you can easily store and retrieve your signals and parameters.

Data stored in SYNECT can be shared by team members and also reused in subsequent development steps and across projects.

Moreover, SYNECT Signal & Parameter Management extends SYNECT Model Management for interface specification and parameter definition, and complements SYNECT Test Management for test case parameter management.

Key Benefits
- Use the same tool for parameters in all development phases – from early M files to ECU calibration
- Work in collaboration on large parameter sets
- Keep track of the history of signals and parameter values
- Master variant complexity by automatic synchronization of variant-dependent parameters across all variants
Working with SYNECT Signal & Parameter Management

So that you can work with your signals, parameters and parameter sets efficiently and communicate with the “outside world”, there are several import/export filters that let you exchange and use data during the development process. Simulink® users benefit from the import and export of MATLAB® M files to flexibly exchange parameter values and signal and parameter definitions between the modeling environment and SYNECT. TargetLink users benefit from the option to exchange variables, type definitions and scalings between the file-based TargetLink Data Dictionary and SYNECT, the central database with multi-user support. To manage parameter sets for different variants, you can specify variant dependencies and select which variant configurations will be supported. You can work with variants by using this module in combination with SYNECT Variant Management (p. 16).

Exchange variables, type definitions and scalings easily between TargetLink Data Dictionary and SYNECT.
SYNECT supports the import and export of several commonly used file formats, and is open to realize customer-specific import/export filters via an open plug-in concept.

File-based parameter management generates a large number of files. SYNECT centrally manages parameter sets for different variants.
SYNECT Model Management

Highlights

- Manage models together with their interfaces, parameters and related files
- For distributed development and for sharing and reusing models
- For controller models and simulation models from various modeling environments

- Supports Simulink® models, TargetLink models and FMUs
- Provides open interfaces to support further modeling environments
- Comprehensive model categorization through customizable metadata
- Integrated variant support

Application Areas

SYNECT Model Management lets you manage your prime asset in a model-based development process – your models. It combines model file management, model metadata management and model interface management – all in one tool, all accessible from one GUI. You can organize your models in libraries with dedicated user and rights permissions and according to their variant dependencies. The modeling of hierarchical dependencies between models and submodels allows building and configuring model compositions from different libraries or work projects. Defining and inspecting the model architecture directly in SYNECT not only increases the transparency of the models directly in the data management solution, it also supports implementing top-down development processes for new models.

Key Benefits

- Share and reuse models within an organization by managing them in one central location
- Make it easy to find the right models by organizing them in libraries, projects, together with alternative implementation variants, and in relation to other models
- Increase model transparency with comprehensive, customizable metadata
- Keep track of model evolution with fine-grained versioning of models, interfaces, parameters and model files
- Implement top-down model development processes by defining the model architecture and interfaces before implementation

1) Requires SYNECT Signal & Parameter Management.
Models
SYNECT Model Management manages both the model structure and the actual model implementations by using model files of the supported modeling environments. Model structure information can be used as a purely descriptive feature, increasing transparency and model understanding directly in the data management solution, without accessing the actual modeling environment. You will get the most out of SYNECT Model Management when you start to actively manage model structure and interfaces directly in SYNECT. By linking signals and parameters from SYNECT Signal & Parameter Management data pools to your models, you can easily implement central signal and parameter management for your models. This enables top-down model development and makes it easier to keep model interface consistency for smoother model integration.

Model Implementations
Model implementations contain the actual model files – and all files that come with a model, such as initialization scripts, parameter files, documentation files, etc. For easy access to the files, these can be managed directly from the SYNECT GUI, in the context of the model and its structural information. SYNECT keeps the model files in an established file-based configuration management system, such as PTC® Integrity or Apache™ Subversion®. If you already manage your models in such a system, migration to SYNECT is easy: Keep the model files in your file-based configuration management systems, link them with your models in SYNECT, and manage them directly from SYNECT.
Model Variants:
One Model, Multiple Implementations
SYNECT lets you manage multiple implementations with the same model and its structural properties. As an organizational feature it helps you group alternative implementations, called implementation variants. You can declare the variant dependencies of the individual implementations according to the variant models defined in SYNECT (see SYNECT Variant Management module, p. 16).

Model Metadata
The SYNECT metadata model contains standard metadata and can also be extended with user-specific metadata attributes. Metadata is handled both on the model level as well as on the model implementation level. Properties that are valid for all implementations of a model can be managed at model level, such as the categorization of models as controller models, plant models, etc.. Implementation-specific information such as the modeling environment, fidelity level for simulation models, or implementation detail level for controller models can be handled at the implementation level. A well thought-out set of metadata will make it easy to find, sort and categorize the models you are managing in SYNECT.

Requirements
Requirements that are imported to SYNECT and maintained in a SYNECT requirements project can be linked to models bidirectionally. This information can be further processed and aggregated in order to implement traceability or coverage reports.
Workflows

**Bottom-Up**
With this workflow, existing models are imported into SYNECT Model Management via tool adapters that are specific to the modeling environment. During import, the essential structural information of the model is collected, such as its inputs, outputs and parameters. After the import these items can be managed individually with the SYNECT Signal & Parameter Management functionality. Import adapters are currently available for Simulink models, TargetLink models, and for FMUs (Functional Mock-up Units). The XML import adapter offers a generic way to import model structure information into SYNECT. The XML file can be provided directly by a modeling tool or a custom tool adapter.

**Top-Down**
In this workflow, new models can be defined by structure in SYNECT and then exported as structural frame models to supported modeling environments for the subsequent implementation of behavior. The same procedure can be applied if structural changes (such as interface changes) will be made in existing models. Export adapters are currently available for Simulink models and for TargetLink models. The existing adapters are open implementations that can be adapted to customer-specific conditions. The XML export adapter can be used to implement completely new tool adapters for additional modeling environments.

**Model Configuration**
You can define new composite models by combining existing models as submodels of the new composite model. To configure the composite model, just select the specific implementation variants and versions to be used for each of its submodels. By default, SYNECT provides the file system with the respective model files of the specified submodels in the correct version and variant for further use. Custom export adapters can be used to create or configure integration frame models according to your own integration mechanisms.
SYNECT Variant Management

Highlights

- Define variant models containing variants, variation points and constraints
- Centrally model and manage variant configurations
- Define relationships between variants and development data
- Import and export variant models

Application Areas

SYNECT Variant Management lets you create variant models, specify constraints between variants, and define variant configurations that need to be supported in subsequent development phases. When working with other SYNECT modules, e.g., SYNECT Test Management, SYNECT Signal & Parameter Management or SYNECT Model Management, you can specify relationships between your data and variants, and manage different variants of your data.

Working with SYNECT Variant Management

To manage your variants systematically, you explicitly define a variant model which you can use throughout the entire development process. This variant model consists of variation points, variants, and constraints between variants. The variant model is the basis for defining variant configurations, in other words concrete combinations of variants that need to be supported in your development project.

Example of a SYNECT variant model consisting of variation points, variants, and constraints between variants.
User-defined list of variant configurations to be supported in the current project.

## Dependencies Between Data and Variants

When working with other SYNECT modules, such as SYNECT Test Management or SYNECT Signal & Parameter Management, you can specify dependencies between your data and variants. For example, you can specify that a test case is valid for a certain variant only, that a parameter exists for a certain variant only, that a parameter has different values for selected variants, or that a model has different implementations for different variants.

## Managing Variants in SYNECT Test Management

Variant dependencies in SYNECT Test Management can be used when defining test execution plans for a specific variant configuration. By default, only test cases that apply to the selected variant configuration are displayed. If you want to use an existing execution plan to test a specific variant configuration, it is easy to identify and exclude tests that are invalid or that do not apply.

You can specify variant dependencies for your test cases. For example, to define that a specific test case applies only to a specific variant.
Managing Variants in SYNECT Signal & Parameter Management

To manage parameters and parameter sets for different variants, in SYNECT you can define variant dependencies for parameters. You can specify that a parameter exists only for a certain variant or that a parameter has different values for different variants. You can then specify and manage different parameter values for your variant configurations, and also import and export values for selected variant configurations.

Parameter values for different variant configurations can be managed centrally. You can immediately see if a parameter depends on a variant and has different values.

Parameter values can be imported or exported for different variant configurations. You can select which variant to import or export values for.

Managing Variants in SYNECT Model Management

To manage model implementations for different variants, you can define variant dependencies for the different implementations of a model in SYNECT. Besides documenting the variant dependencies of implementations, these dependencies can be accessed via the tool automation API to facilitate variant-dependent configuration of composite models.

Workflow Management

dSPACE offers a SYNECT-based Workflow Management solution for integrating complex workflows with a large number of variants (such as in HIL and modeling scenarios) into the dSPACE tool chain and for automating them. The solution requires SYNECT Base, SYNECT Variant Management, and additional software from dSPACE.
General SYNECT Functionalities

**Highlights**
- Database support
- Data model customization
- User and rights management
- Version management
- Automation and openness
- Traceability to requirements
- Working with configuration management systems

**Database Support**
SYNECT supports SQL Server® and Oracle®. For a quick start, you can use the free SQL Server Express Edition.

**Data Model Customization**
SYNECT makes it easy to extend the data model by adding custom attributes (which you can also deactivate optionally, if required). For example, you can define additional attributes for your test cases: supplier or customer lists, test system IDs, etc. SYNECT also has convenient functions for importing and exporting data with this customized data model.

**User and Rights Management**
SYNECT provides user and rights management functionality. As an administrator, you can define who has what kind of access to a specific project or specific item types. Data access can be organized efficiently and securely. For example, you can restrict access to test project plans according to your supplier or customer, or provide read-only and export-only access to specific parameter sets. Authentication can be granted based on standard Windows® authentication.
**Fine-Grained Version Management**

SYNECT gives you the ability to manage different versions of items such as test cases, parameters or models. There is also a branching mechanism for creating a new development branch from an already released version of an item. The trunk, branches and versions are displayed graphically. A compare functionality enables you to compare different versions of the same item, or even of a complete project or workspace. Comments to element versions can be annotated.

Example of a version history.

Version compare window.
Automation and Openness
SYNECT provides an open COM API that enables you to automate actions that are usually manual, and also to define specific actions such as data imports and exports, report generation or consistency checks. These actions can also be accessed by extending the SYNECT user interface, and they can be triggered by client or server events. Interaction with external tools is also possible with this API, via connections created for different integration scenarios. The languages currently supported are Python, MATLAB®, Visual Basic for Applications, C++, C#, and Visual Basic.
To enable convenient use of customer-developed features, it is also possible to extend the SYNECT GUI. You can create new ribbons and controls, or extend the context menus of SYNECT items such as test cases, test plans, or models.

Traceability to Requirements and Specifications
With SYNECT you can link your data (e.g., test cases, parameters, or models) to requirements. SYNECT is flexible and provides various options for importing requirements. Excel® and the Requirements Interchange Format (RIF, ReqIF) enable you to import requirements e.g., for IBM Rational® DOORS, PTC® Integrity, or Microsoft Team Foundation Server (TFS). An open Python API lets you build customized importers for other requirements formats and tools.
SYNECT helps you analyze the impact of changes in requirements by providing an overview of linked objects that might be affected. To navigate between requirements and linked data, all you need is a mouse click. A special requirements coverage view is available to check if a requirement is covered by a test case and if this test case has been executed successfully.

Working with Configuration Management Systems
SYNECT supports file-based configuration management systems like Apache™ Subversion®, PTC® Integrity, or Microsoft TFS. You can browse file repositories from within SYNECT and compare two file versions located in the repository. You can also create links to a specific file version. These ‘versioned links’ can then be stored as an attribute of a SYNECT item, such as a test case or a model. An example of a use case is to ensure that the same version of a test case implementation is always used, in regression testing, for example.