

Real-Time Testing

# Glossary

Release 2021-A – May 2021

## How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	<a href="mailto:info@dspace.de">info@dspace.de</a>
Web:	<a href="http://www.dspace.com">http://www.dspace.com</a>

## How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: <http://www.dspace.com/go/locations>
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.  
Tel.: +49 5251 1638-941 or e-mail: [support@dspace.de](mailto:support@dspace.de)

You can also use the support request form: <http://www.dspace.com/go/supportrequest>. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

## Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit <http://www.dspace.com/go/patches> for software updates and patches.

## Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2016 - 2021 by:  
dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

This publication and the contents hereof are subject to change without notice.

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

# Real-Time Testing Glossary

Introduction	The glossary briefly explains the most important expressions and naming conventions used in the Real-Time Testing documentation.
--------------	--

Where to go from here	Information in this section
	<div>D.....3</div> <div>G.....4</div> <div>H.....4</div> <div>M.....4</div> <div>O.....4</div> <div>P.....5</div> <div>R.....5</div> <div>S.....6</div> <div>T.....6</div> <div>V.....6</div> <div>Y.....6</div>

## D

**Documents folder**    A standard folder for user-specific documents.  
%USERPROFILE%\Documents\dSPACE\<ProductName>\  
<VersionNumber>

## G

---

**Generator function** A generator function is a Python function which is executed in several steps. Each step is separated by a `yield` statement. For detailed information, refer to [Basics on Generator Functions \(Real-Time Testing Guide !\[\]\(2bdfe261b986065ee0ac76460d6528c9\_img.jpg\)](#)).

## H

---

**Host PC** A standard PC that the dSPACE test and experiment software is installed on. In the host PC, you can configure the dSPACE hardware, download a simulation application to the platform and control the simulation.

## M

---

**Multiprocessor system** A multiprocessor system consists of several processor boards (DS1006 or DS1007) which are connected via Gigalinks or SCALEXIO processing hardware (SCALEXIO Processing Unit or DS6001) which are connected via IOCNET.

## O

---

**Offline simulation application (OSA)** An offline simulation application (OSA) file is an executable file for VEOS. After the build process with a tool such as VEOS Player, the OSA file can be downloaded to VEOS. An OSA can contain multiple VPUs, each of which contains either a V-ECU or environment VPU. Each VPU runs in a separate process of the host PC's operating system. To perform real-time testing, a service is integrated into the offline simulation application during the build process, refer to [Enabling Real-Time Testing for dSPACE Platforms \(Real-Time Testing Guide !\[\]\(05be7c7a8995decd503647c99211f7c2\_img.jpg\)](#)).

## P

**Parallel() / ParallelRace()** Generator functions can be executed quasi in parallel. They must be embedded in a Parallel() or ParallelRace generator function by the scheduler module for this. The embedded generator functions are executed consecutively inside an RTT sequence, but all functions are executed once in a sampling step. To the simulation model, the Parallel() or ParallelRace() generator functions seem to be executed in parallel. A Parallel() generator function finishes when the last embedded generator function has finished. A ParallelRace() generator function finishes when the first embedded generator function has finished. For details, refer to [Using the Parallel\(\) Generator Function \(Real-Time Testing Guide !\[\]\(2e897e890e69d81eae4503a8342c36b0\_img.jpg\)](#)) and [Using the ParallelRace\(\) Generator Function \(Real-Time Testing Guide !\[\]\(ce4e2504c7100a62a9a9496b2e01b6e4\_img.jpg\)](#)).

## R

**Real-time application** An application running in real time on a dSPACE platform. A real-time application can be built, for example, from a Simulink model containing RTI blocks. A real-time application for a SCALEXIO system is built from a real-time model in ConfigurationDesk. To perform real-time testing, a service must be integrated into the real-time application during the build process for some platforms, refer to [Enabling Real-Time Testing for dSPACE Platforms \(Real-Time Testing Guide !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)](#)).

**Real-time model** A real-time model is a model designed in MATLAB/Simulink for real-time simulation. A model contains the control algorithm for a controller (rapid control prototyping) or the algorithm of the controlled system (hardware-in-the-loop). The source code of the real-time application is automatically generated from the real-time model.

**Real-Time Test Manager** The Real-Time Test Manager is a graphical user interface for managing RTT sequences on the real-time platform. You can use the Real-Time Test Manager to download the RTT sequences and control their execution. It has a Log Viewer for the outputs of the RTT sequences. Refer to [Managing RTT Sequences Using the Real-Time Test Manager \(Real-Time Testing Guide !\[\]\(0aff635c4179ba9e710b00f4b01d3b20\_img.jpg\)](#)).

**Real-Time Test Manager Server interface** The Real-Time Test Manager Server interface is a Python module (rttmanagerlib). The module is used in host scripts which manage the RTT sequences. It has all the required functions to download, start, pause, continue, or stop RTT sequences. Additionally, it can be used to implement event handling on the host. For details, refer to [Managing RTT Sequences in Python Scripts \(Real-Time Testing Guide !\[\]\(830769b31eeeaca920791081939ff8ba\_img.jpg\)](#)).

**RTT sequence** RTT sequences are the tests implemented for real-time testing. They are programmed in Python. The RTT sequences are downloaded to the real-time hardware and executed in parallel to the real-time application.

## S

---

**Simulation application** The generic term for real-time application and offline simulation application. The application runs on a simulation platform.

**Simulation model** A model that is designed in MATLAB/Simulink for simulating control algorithms or a controlled system. It is a generic term for real-time models or models that are used for offline simulations.

**Simulation platform** The generic term for real-time systems that calculates the real-time application in real time and simulators that are not connected to a physical system and therefore independent of the real time.

## T

---

**TRC variable** A variable of a simulation application that is available in the variable description file (TRC file).

## V

---

**V-ECU** Abbreviation of *virtual ECU*.

An executable element in a simulation system that emulates a real ECU in a simulation scenario.

**VEOS** A simulator which is part of the PC and allows the user to run an offline simulation application (OSA) without relation to real time.

VEOS Player is the graphical user interface for VEOS.

**VPU** Abbreviation of *virtual processing unit*.

VPU is a generic term for a part of a simulation system that can be run in an offline simulation by VEOS.

## Y

---

**yield** Real-Time Testing uses the Python **yield** statement as a way to suspend and resume the execution of an RTT sequence. The Python interpreter executes the code between the current and the next **yield** statement in each simulation step.