

ControlDesk

Message Handling

For ControlDesk 7.4

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:	info@dspace.de
Web:	http://www.dspace.com

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

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.

© 2010 - 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.





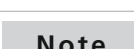



Contents

About This Document	5
Basics and Instructions	7
Introduction to Message Handling.....	8
Basics on Message Handling.....	8
Introduction to the Message Reader API.....	10
Reading dSPACE Log Messages via the Message Reader API.....	10
Supported dSPACE Products and Components.....	12
Example of Reading Messages with Python.....	12
Example of Reading Messages with C#.....	14
Reference Information	17
dSPACE Log.....	17
Filter Panel.....	19
Message Dialog.....	20
Messages.....	21
Automation	25
Log Handling.....	26
Log-Related Interfaces.....	26
dSPACE.Common.MessageHandler.Logging Reference.....	27
ILogMessage Interface.....	27
ILogSession Interface.....	28
MessageReader Class.....	30
MessageReaderSettings Class.....	31
Severity Enumeration.....	33
Glossary	35
Index	73

About This Document

Content This document introduces you to ControlDesk's message handling.

Symbols dSPACE user documentation uses the following symbols:

Symbol	Description
	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
	Indicates a hazard that, if not avoided, could result in property damage.
	Indicates important information that you should take into account to avoid malfunctions.
	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>

or

%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>

Documents folder A standard folder for user-specific documents.

%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>

Accessing dSPACE Help and PDF Files


After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the  icon in dSPACE Help. The PDF opens on the first page.

Basics and Instructions

Where to go from here

Information in this section

Introduction to Message Handling.....	8
Introduction to the Message Reader API.....	10

Introduction to Message Handling

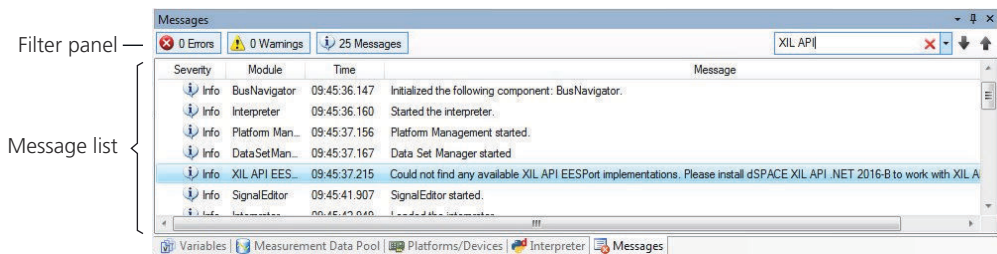
Basics on Message Handling

Introduction

ControlDesk lets you handle messages.

Messages controlbar

The Messages controlbar provides a history of all the info, advice, error and warning messages, and all the questions that occur when you work with the product. This helps you check the system state.

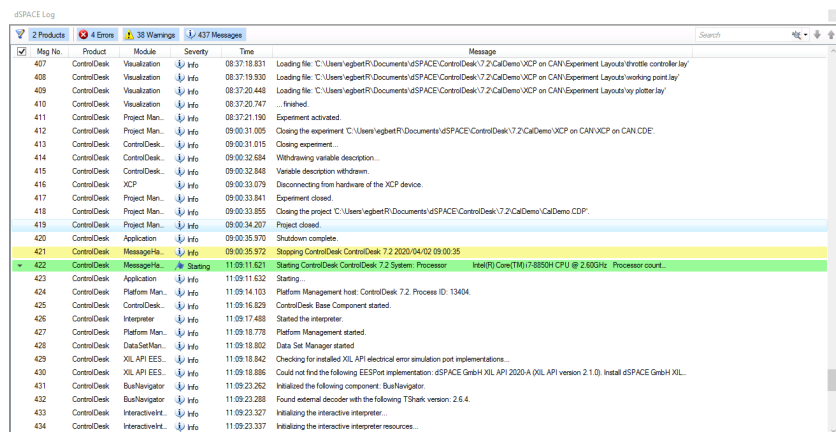


For more information, refer to [Messages](#) on page 21.

dSPACE Log

The dSPACE Log is a collection of errors, warnings, information, questions, and advice issued by all dSPACE products and connected systems over more than one session.

The following illustration shows an example:



For more information, refer to [dSPACE Log](#) on page 17.

Automation interface The ControlDesk automation interface lets you write messages to the [dSPACE Log](#). You can also clear messages from the Messages controlbar.

For more information, refer to [Log-Related Interfaces](#) on page 26.

Message Reader API ControlDesk supports the *Message Reader API*. You can read out log messages of the [dSPACE Log](#), for example, in C# applications and Python scripts by using the API. The API lets you filter log messages by the dSPACE product and by time, for example.

For more information, refer to [Reading dSPACE Log Messages via the Message Reader API](#) on page 10.

Related topics	Basics
	Reading dSPACE Log Messages via the Message Reader API..... 10

Introduction to the Message Reader API

Where to go from here

Information in this section

Reading dSPACE Log Messages via the Message Reader API.....	10
You can read log messages of the dSPACE Log via the Message Reader API.	
Supported dSPACE Products and Components.....	12
Provides an overview of all dSPACE products and components whose messages you can access via the Message Reader API.	
Example of Reading Messages with Python.....	12
You can read the log messages via Python. You can combine multiple filters to display only messages according to your specifications.	
Example of Reading Messages with C#.....	14
You can read the log messages via C#. You can combine multiple filters to display only messages according to your specifications.	

Reading dSPACE Log Messages via the Message Reader API

Introduction

You can read log messages of the dSPACE Log via the Message Reader API.

dSPACE Log

The dSPACE Log is a collection of errors, warnings, information, questions, and advice issued by all dSPACE products and connected systems over more than one session.

The dSPACE Log is saved as a collection of binary message log files. These files are created when a dSPACE product is running. A single run of a dSPACE product is called a *log session*.

Note

If the maximum file size for the binary message log file is reached, messages at the beginning of the dSPACE Log might get deleted. Contact dSPACE Support to solve this.

Message Reader API

You can use the Message Reader API to access all binary message log files of the dSPACE Log. You can combine multiple filters to display only log messages according to your specifications. For example, you can configure the Message Reader API to display only log messages from a specific dSPACE product.

The Message Reader API is available as of dSPACE Release 2020-A. For information on the dSPACE products and components that support the Message Reader API, refer to [Supported dSPACE Products and Components](#) on page 12.

dSPACE.Common.MessageReader.dll The Message Reader API is implemented by the `dSPACE.Common.MessageReader.dll` file. It is located in the `bin` subfolder of the installation folder of each dSPACE product that supports the Message Reader API.

Supported dSPACE Releases

The Message Reader API lets you access log messages written by dSPACE products since dSPACE Release 2016-B.

Message Reader API change in dSPACE Release 2021-A

There is a migration issue specific to the Message Reader API. The issue occurs if you use the API with Python. The issue was caused by the migration to Python 3.9/pythonnet 2.5.3 with dSPACE Release 2021-A.

There is no migration issue to consider if you use the API with C#.

Specifying a product filter As of dSPACE Release 2021-A, the `Products` property of the `MessageReaderSettings` class can no longer be used to set the list of products for which to filter in the log sessions. The Message Reader API provides the `SetProducts` method for this purpose.

The following table shows how to specify a product filter before and after migration:

Using Message Reader API of ...	
... dSPACE Release 2020-B and Earlier (Python 3.6)	... dSPACE Release 2021-A and Later (Python 3.9)
<pre># Specify products whose messages to read: Settings = MessageReaderSettings() Settings.Products.Add('ControlDesk') Settings.Products.Add('AutomationDesk')</pre>	<pre># Specify products whose messages to read: Settings = MessageReaderSettings() Settings.SetProducts(['ControlDesk', 'AutomationDesk'])</pre>

Related topics

Basics

[Supported dSPACE Products and Components](#)..... 12

Examples

[Example of Reading Messages with C#](#)..... 14
[Example of Reading Messages with Python](#)..... 12

References

[MessageReaderSettings Class](#)..... 31

Supported dSPACE Products and Components

Supported dSPACE products and components

You can use the Message Reader API to access messages from the following dSPACE products and components:

- ASM KnC
- AutomationDesk
- Bus Manager (stand-alone)
- cmdloader
- ConfigurationDesk
- Container Management
- ControlDesk
- dSPACE AUTOSAR Compare
- dSPACE XIL API .NET Implementation
- Firmware Manager
- ModelDesk
- MotionDesk
- Real-Time Testing
- RTI Bypass Blockset
- SYNECT client
- SystemDesk
- TargetLink Property Manager
- VEOS

Related topics

Basics

[Reading dSPACE Log Messages via the Message Reader API..... 10](#)

Example of Reading Messages with Python

Introduction

You can read the log messages via Python by using the `c1r` module. You can combine multiple filters to display only messages according to your specifications.

Referencing a message reader assembly

You have to reference a `dSPACE.Common.MessageReader.dll` assembly. For information on the location of the assembly, refer to [dSPACE.Common.MessageReader.dll](#) on page 11.

In the following examples it is assumed that the dSPACE Installation Manager is installed and that the message reader assembly is installed in `C:\Program Files\Common Files\dSPACE\InstallationManager\bin`.

The following code references and imports the message reader assembly.

```
# Insert path of message log file access assembly:
import sys
AssemblyPath = r'C:\Program Files\Common Files\dSPACE\InstallationManager\bin'
if not sys.path.count(AssemblyPath):
    sys.path.insert(1, AssemblyPath)

# Add reference to assembly and import it:
import clr
clr.AddReference('dSPACE.Common.MessageReader')
from dSPACE.Common.MessageHandler.Logging import *
```

Reading all messages

The following example reads all existing message log files and prints all messages via Python. It is assumed that the message reader assembly is referenced and imported. Refer to [Referencing a message reader assembly](#) on page 12.

```
# Create message reader and print text of each message:
Reader = MessageReader(None)
for Message in Reader.ReadMessages():
    print(Message.MessageText)
Reader.Dispose()
```

Filtering messages by severity, product, and session

The following example reads and prints messages with a severity of `Error`, `SevereError`, or `SystemError`. Also, only messages of the last sessions of `ControlDesk` and `AutomationDesk` are read and printed. It is assumed that the message reader assembly is referenced and imported. Refer to [Referencing a message reader assembly](#) on page 12.

```
# Define error severities:
SEVERITY_ERROR = 3
SEVERITY_SEVERE_ERROR = 4
SEVERITY_SYSTEM_ERROR = 5

# Configure products and sessions whose messages to read:
Settings = MessageReaderSettings()
Settings.MaximalSessionCount = 1
Settings.SetProducts(['ControlDesk', 'AutomationDesk'])

# Create message reader and print text of each error message:
Reader = MessageReader(Settings)
for Message in Reader.ReadMessages():
    # Print error messages only:
    if Message.Severity == SEVERITY_ERROR or \
        Message.Severity == SEVERITY_SEVERE_ERROR or \
        Message.Severity == SEVERITY_SYSTEM_ERROR:
        print('%s: %s' % (Message.Session.ProductName, Message.MessageText))
Reader.Dispose()
```

Note

The ReadMessages method returns an enumerator which must either read all messages or must be disposed when no longer used. It is not possible to use two enumerators interleaved, only one enumerator may read messages at a time. Refer to [MessageReader Class](#) on page 30.

Filtering messages by time

Times are given by .NET DateTime objects. Times are given as UTC times (Coordinated Universal Time). You can obtain the current UTC time by `System.DateTime.UtcNow`.

The following example reads all messages after a certain start time. It is assumed that the message reader assembly is referenced and imported. Refer to [Referencing a message reader assembly](#) on page 12.

```
import System
Settings = MessageReaderSettings()
Settings.MessageTimeAfter = System.DateTime.UtcNow # Read messages after now

# Create message reader and print time and text of each message:
Reader = MessageReader(Settings)
for Message in Reader.ReadMessages():
    print('%s: %s' % (Message.UtcTimeStamp, Message.MessageText))
Reader.Dispose()
```

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API.....	10
Supported dSPACE Products and Components.....	12

References

MessageReaderSettings Class.....	31
--	--------------------

Example of Reading Messages with C#

Introduction

You can read the log messages via C#. You can combine multiple filters to display only messages according to your specifications.

Referencing a message reader assembly

You have to reference a `dSPACE.Common.MessageReader.dll` assembly. For information on the location of the assembly, refer to [dSPACE.Common.MessageReader.dll](#) on page 11.

Reading all messages

The following example reads all existing message log files and prints the messages:

```
using dSPACE.Common.MessageHandler.Logging;
...

// Create message reader and print text of each message:
using (MessageReader reader = new MessageReader(null))
{
    foreach (message in reader.ReadMessages())
    {
        Console.WriteLine(message.MessageText);
    }
}
```

Filtering messages by severity, product, and session

The following example reads and prints messages with a severity of **Error**, **SevereError**, or **SystemError**. Also, only messages of the last sessions of **ControlDesk** and **AutomationDesk** are read and printed.

```
using dSPACE.Common.MessageHandler.Logging;
...

// Read the Last Log sessions of ControlDesk and AutomationDesk only:
MessageReaderSettings settings = new MessageReaderSettings();
settings.MaximalSessionCount = 1;
settings.Products.Add("ControlDesk");
settings.Products.Add("AutomationDesk");

using (MessageReader reader = new MessageReader(settings))
{
    foreach (ILogMessage message in reader.ReadMessages())
    {
        // Print error messages only:
        if (message.Severity == Severity.Error
            || message.Severity == Severity.SevereError
            || message.Severity == Severity.SystemError)
        {
            Console.WriteLine(message.Session.ProductName + ": " + message.MessageText);
        }
    }
}
```

Note

The `ReadMessages` method returns an enumerator which must either read all messages or must be disposed when no longer used. It is not possible to use two enumerators interleaved, only one enumerator may read messages at a time. Refer to [MessageReader Class](#) on page 30.

Related topics**Basics**

[Reading dSPACE Log Messages via the Message Reader API..... 10](#)

Supported dSPACE Products and Components.....	12
References	
MessageReaderSettings Class.....	31

Reference Information

Where to go from here

Information in this section

dSPACE Log.....	17
To open the dSPACE Log.	
Filter Panel.....	19
To filter the message list and search for text in the message list.	
Message Dialog.....	20
To display a message of ControlDesk.	
Messages.....	21
To show or hide the Messages controlbar.	

dSPACE Log

Access

You can access this command via:

Ribbon	View – Show
Context menu of	None
Shortcut key	None
Icon	

Purpose

To open the dSPACE Log.

Result

The dSPACE Log is opened in ControlDesk's working area.

Description The dSPACE Log is a collection of errors, warnings, information, questions, and advice issued by all dSPACE products and connected systems over more than one session.

Filter panel The filter panel lets you filter the message list and search for text in the message list.

Refer to [Filter Panel](#) on page 19.

Message list The message list provides the following information for each message:

Information	Description
Date ¹⁾	The date the message was issued.
Main Module Number ¹⁾	The main module that issued the message.
Message	The content of the message.
Message Code ¹⁾	The code of the message.
Message Number ¹⁾	The number of the message.
Module ¹⁾	The module that issued the message.
Process ID ¹⁾	The ID of the process in which the message was issued.
Product ¹⁾	The name of the product that issued the message.
Severity ¹⁾	The severity level of the message, indicated by one of the following symbols: <ul style="list-style-type: none"> ✖ errors ⚠ warnings ℹ other messages, i.e., infos, advice, and questions
Submodule Number ¹⁾	The submodule that issued the message.
Thread ID ¹⁾	The ID of the thread in which the message was issued.
Time ¹⁾	The time the message was issued.

¹⁾ You can specify whether this information is displayed via the Show Columns command.

Show/Hide Messages of Log Session (Available in the column header) Lets you expand/collapse the messages in the message list according to log sessions. A log session starts when a dSPACE product is started.

Fit Column Width (Available from the context menu of column headers) Lets you optimize the width of the selected column.

Fit All Columns (Available from the context menu of column headers) Lets you optimize the widths of all columns to fit the width of the list.

Show Columns (Available from the context menu of column headers) Lets you add/remove a column to/from the message list.

Reset Columns Lets you reset the display of message list columns to the default.

Copy (Available from the context menu of messages) Lets you copy the selected messages to the Clipboard.

Show Message (Available from the context menu of messages) Lets you display the selected message in a separate dialog.

Show Filter Panel (Available from the context menu of messages) Lets you show/hide the filter panel.

Refer to [Filter Panel](#) on page 19.

Show Details (Available from the context menu of messages) Lets you display/hide all the details of the currently selected message.

Lock Scrolling (Available from the context menu of messages) Lets you disable the automatic horizontal scrolling mechanism in the viewer.

Related topics

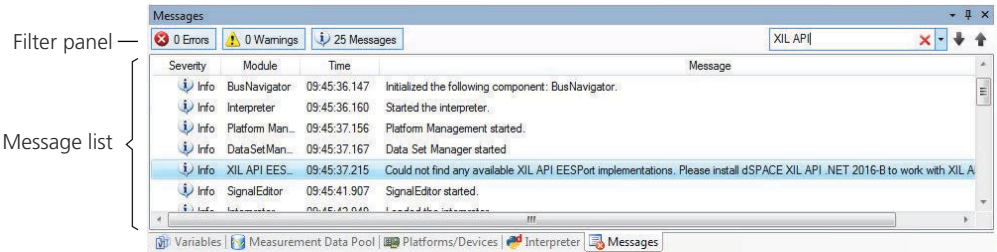
References

Messages.....	21
---------------	----

Filter Panel

Access

- The filter panel is part of the [Messages](#) controlbar. Refer to [Messages](#) on page 21.
The following illustration shows the Messages controlbar with the filter panel as an example:



- The filter panel is displayed when the dSPACE Log is opened in ControlDesk's working area. Refer to [dSPACE Log](#) on page 17.

Purpose

To filter the message list and search for text in the message list.

Filter panel	<p>The filter panel lets you filter the message list and search for text in the message list.</p> <p>Edit Product Filter (🔍 button - only in dSPACE Log) Lets you specify a list of products for filtering the message list.</p> <p>To apply the product filter, click the Enable/Disable Product Filter button next to the 🔍 button.</p> <p>Enable/Disable Product Filter (🔍 Products button - only in dSPACE Log) Lets you enable/disable the product filter.</p> <p>Show/Hide Errors (🚫 Errors button) Lets you display or hide errors.</p> <p>Show/Hide Warnings (⚠️ Warnings button) Lets you display or hide warnings.</p> <p>Show/Hide other Messages (📄 Messages button) Lets you display or hide other messages, i.e., infos, advice, and questions.</p> <p>Search Lets you enter a text string for searching the message list.</p> <p>You can use the following wildcards in the text string:</p> <ul style="list-style-type: none">▪ ? (wildcard for one character)▪ * (wildcard for any number of characters) <p>To mask a wildcard, enter the \ character before the wildcard.</p> <p>To select the next occurrence of the search string, click ⏴ next to the edit field, or press the Enter key while the search field has the focus.</p> <p>To select the previous occurrence of the search string, click ⏵ next to the edit field.</p>
---------------------	--

Related topics	<p>References</p> <div><p>dSPACE Log..... 17</p><p>Messages..... 21</p></div>
-----------------------	---


Message Dialog

Purpose	To display a message of ControlDesk.
Description	<p>Copy (Available from the context menu of a message dialog) Lets you copy the selected text from the message to the Clipboard.</p> <p>Select all (Available from the context menu of a message dialog) Lets you select the entire message text.</p> <p>Copy Message (Available from the context menu of a message dialog) Lets you copy the entire message and its attributes to the Clipboard.</p>

Messages

Access

You can access this command via:

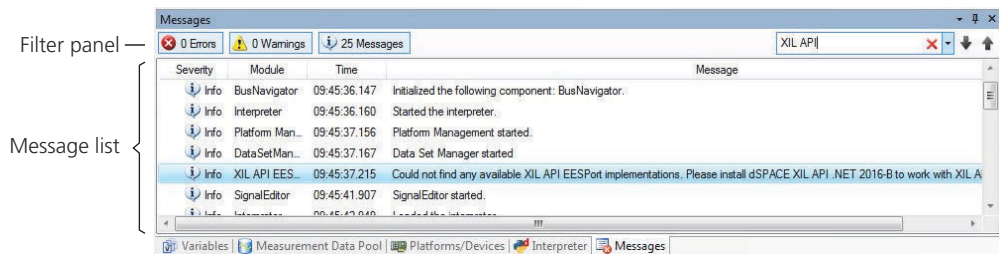
Ribbon	View – Controlbar – Switch Controlbars
Context menu of	None
Shortcut key	Alt+Shift+1
Icon	

Purpose

To show or hide the Messages controlbar.

Description

The Messages controlbar provides a history of all the info, advice, error and warning messages, and all the questions that occur when you work with the product. This helps you check the system state.




Filter panel

The filter panel lets you filter the message list and search for text in the message list.

Refer to [Filter Panel](#) on page 19.

Message list

The message list provides the following information for each message:

Information	Description
Date ¹⁾	The date the message was issued.
Main Module Number ¹⁾ (Main Module)	The main module that issued the message.
Message	The content of the message.
Message Code ¹⁾	The code of the message.
Module ¹⁾	The module that issued the message.
Severity	The severity level of the message, indicated by one of the following symbols: <ul style="list-style-type: none">  Errors

Information	Description
	<ul style="list-style-type: none"> ⚠ Warnings ℹ Other messages, i.e., infos, advice, and questions
Submodule Number ¹⁾ (Submodule)	The submodule that issued the message.
Time ¹⁾	The time the message was issued.

¹⁾ You can specify whether this information is displayed via the Show Columns command.

Fit Column Width (Available from the context menu of column headers) Lets you optimize the width of the selected column.

Fit All Columns (Available from the context menu of column headers) Lets you optimize the widths of all columns to fit the width of the list.

Show Columns (Available from the context menu of column headers) Lets you add/remove a column to/from the message list.

Reset Columns Lets you reset the display of message list columns to the default.

Copy (Available from the context menu of messages) Lets you copy the selected messages to the Clipboard.

Help (Available from the context menu of messages) Lets you display message-specific help.

Expand (Available from the context menu of messages; available only if the tree view of messages is active) Lets you expand all the collapsed subelements of the selected element.

The hidden subelements of the selected element are displayed.

Collapse (Available from the context menu of messages; available only if the tree view of messages is active) Lets you collapse all the subelements of the selected element.

The subelements of the selected element are hidden.

Show Message (Available from the context menu of messages) Lets you display the selected message in a separate dialog.

Tree View (Available from the context menu of messages) Lets you toggle between flat and tree views of the messages.

Show Filter Panel (Available from the context menu of messages) Lets you show/hide the filter panel.

Refer to [Filter Panel](#) on page 19.

Clear Messages (Available from the context menu of messages) Lets you clear all the messages in the list.

Lock Scrolling (Available from the context menu of messages) Lets you disable the automatic horizontal scrolling mechanism in the viewer.

Related topics

References

[dSPACE Log.....](#) 17

Automation

Where to go from here

Information in this section

Log Handling.....	26
dSPACE.Common.MessageHandler.Logging Reference.....	27

Log Handling

Log-Related Interfaces



Introduction

You can write messages to the dSPACE log file. You can also clear the messages in the Log Viewer.

Description

The Log property of the Application object implements the *ILoLog* interface. You can use it to access the log functions.

Related interfaces

Interface	Description
ILoLog (refer to Log / ILoLog <<Interface>> (ControlDesk Automation ))	Interface to access the message log.
IXaApplication (refer to Application / IXaApplication <<Interface>> (ControlDesk Automation ))	This interface is used to access the ControlDesk application object.

dSPACE.Common.MessageHandler.Logging Reference

Where to go from here

Information in this section

ILogMessage Interface	27
To access information about a message as written to a log file.	
ILogSession Interface	28
To access information about a message log session.	
MessageReader Class	30
To read serialized messages written by dSPACE products.	
MessageReaderSettings Class	31
To define the settings of a message reader.	
Severity Enumeration	33
To specify the severity of a message.	

ILogMessage Interface

Namespace dSPACE.Common.MessageHandler.Logging

Description To access information about a message as written to a log file.

Properties The element has the following properties:

Name	Description	Get/Set	Type
IsStartMessage	Gets a value indicating whether the message is a session start message.	Get	Boolean
IsStopMessage	Gets a value indicating whether the message is a session stop message.	Get	Boolean
MainModuleNumber	Gets the main module number of the message.	Get	Integer
MessageCode	Gets the error code of the message.	Get	Integer
MessageText	Gets the text of the message.	Get	String
ModuleName	Gets the module name of the message.	Get	String
Session	Gets the log session which issued the message.	Get	ILogSession (refer to ILogSession Interface on page 28)
Severity	Gets the severity of the message.	Get	Severity (refer to Severity Enumeration on page 33)

Name	Description	Get/Set	Type
SubmoduleNumber	Gets the submodule number of the message.	Get	<i>Integer</i>
ThreadId	Gets the thread ID of the submitting thread.	Get	<i>Integer</i>
TimeStamp	Gets the time when the message was submitted. Given as local time in the time zone of the session.	Get	<i>DateTime</i>
UtcTimeStamp	Gets the time when the message was submitted in UTC time.	Get	<i>DateTime</i>

Methods The element has no methods.

Related topics

Basics

[Reading dSPACE Log Messages via the Message Reader API.....](#) 10

Examples

[Example of Reading Messages with C#.....](#) 14
[Example of Reading Messages with Python.....](#) 12

References

[ILogSession Interface.....](#) 28
[Severity Enumeration.....](#) 33

ILogSession Interface

Namespace `dSPACE.Common.MessageHandler.Logging`

Description To access information about a message log session.

Properties The element has the following properties:

Name	Description	Get/Set	Type
CloseTime	Gets the time when the session was closed. Returns an undefined time (0, <code>DateTimeKind.Unspecified</code>) if the session is still open or was not closed successfully. Given as local time in the time zone of the session.	Get	<i>DateTime</i>

Name	Description	Get/Set	Type
IsOpen	Gets a value indicating whether the session is still open. If true, the session is still open and new messages can be written.	Get	<i>Boolean</i>
IsValid	Gets a value indicating whether the session is valid. A session can become invalid if its log files are corrupted.	Get	<i>Boolean</i>
MetaData	Gets the products metadata as read from log file session info.	Get	<i>Dictionary< String, String ></i>
ProcessId	Gets the process ID of the log session.	Get	<i>Integer</i>
ProductName	Gets the product name of the log session.	Get	<i>String</i>
SessionId	Gets the ID of the log session. This ID is unique in the context of its session reader.	Get	<i>Integer</i>
StartTime	Gets the sessions start time. Given as local time in the time zone of the session.	Get	<i>DateTime</i>
TimezoneName	Gets the standard time zone name of the session.	Get	<i>String</i>
TimezoneOffset	Gets the time zone offset of the session relative to UTC.	Get	<i>TimeSpan</i>
UtcCloseTime	Gets the time when the session was closed as UTC time. Returns an undefined time (0, <i>DateTimeKind.Unspecified</i>) if the session is still open or was not closed successfully.	Get	<i>DateTime</i>
UtcStartTime	Gets the start time of the log session as UTC time.	Get	<i>DateTime</i>

Methods

The element has the following methods:

Name	Description	Parameter ¹⁾	Returns
ToSessionTime	Converts UTC time to time zone used when the session was written.	<ul style="list-style-type: none"> <DateTime> utcTime: Specifies the UTC time to convert. 	Time in the time zone of the logging session. <ul style="list-style-type: none"> DateTime

¹⁾ <Type> Name: Description

Related topics

Basics

[Reading dSPACE Log Messages via the Message Reader API..... 10](#)

Examples

[Example of Reading Messages with C#..... 14](#)

[Example of Reading Messages with Python..... 12](#)

MessageReader Class

Description To read serialized messages written by dSPACE products.

Constructor The element has the following constructor:

Name	Description	Parameter ¹⁾	Returns
MessageReader	Initializes a new instance of the MessageReader class.	<ul style="list-style-type: none">▪ <MessageReaderSettings>²⁾ settings: Settings which allow to specify which sessions and messages are read. Can be null, causing all existing log files to be read.	None

¹⁾ <Type> Name: Description

²⁾ Refer to [MessageReaderSettings Class](#) on page 31

Properties The element has no properties.

Methods The element has the following methods:

Name	Description	Parameter ¹⁾	Returns
Dispose	Performs application-specific tasks associated with freeing, releasing, or resetting unmanaged resources.	None	None

Name	Description	Parameter ¹⁾	Returns
ReadMessages	<p>Reads the messages written to the log files of the sessions up to now.</p> <p>The messages are returned in chronological order according to their time stamps.</p> <div>Note<p>The ReadMessages method returns an enumerator which must either read all messages or must be disposed when no longer used. It is not possible to use two enumerators interleaved, only one enumerator may read messages at a time.</p></div>	None	<p>Messages read from log file.</p> <ul style="list-style-type: none">▪ IEnumerable< ILogMessage (refer to ILogMessage Interface on page 27) >

¹⁾ <Type> Name: Description

Related topics

Basics
[Reading dSPACE Log Messages via the Message Reader API.....](#) 10

Examples
[Example of Reading Messages with C#.....](#) 14
[Example of Reading Messages with Python.....](#) 12

References
[MessageReaderSettings Class.....](#) 31

MessageReaderSettings Class

Description	<p>To define the settings of a message reader.</p> <p>Used to filter the log sessions and messages read.</p>
--------------------	--

Constructor

The element has the following constructor:

Name	Description	Parameter ¹⁾	Returns
MessageReaderSettings	Initializes a new instance of the MessageReaderSettings class.	None	None

¹⁾ <Type> Name: Description**Properties**

The element has the following properties:

Name	Description	Get/Set	Type
DirectoryNames	Gets a list of specific directory names from which to read log files. If the list is empty, all standard directories are searched for log files.	Get	<i>List< String ></i>
MaximalSessionCount	Gets or sets the maximal number of log sessions read for each product. If the count is a positive number n, only the last n sessions are read. If the count is not positive, an unlimited number of sessions is read. The default value is zero, i.e., unlimited.	Get/Set	<i>Integer</i>
MessageTimeAfter	Gets or sets the minimal time for which messages are read, given as UTC time. Only messages submitted after the message time are read. The message time may be in the past. The message time must be given as valid UTC time. The default time is undefined, i.e., each message time is allowed.	Get/Set	<i>DateTime</i>
Products	Gets the list of product names for which to read log sessions. If the list is empty sessions of all products are read.	Get	<i>List< String ></i>
StartTimeAfter	Gets or sets the minimal start time for which sessions are read, given as UTC time. Only sessions which started after the start time are read. The start time may be in the past. The start time must be given as valid UTC time. The default time is undefined, i.e., each start time is allowed.	Get/Set	<i>DateTime</i>

Methods

The element has the following methods:

Name	Description	Parameter ¹⁾	Returns
SetDirectoryNames	Sets the list of specific directory names from which to read log files. You do not have to specify a list. If the list is empty, all standard directories are searched for log files.	<code><string[]> names</code> : Array of directory names.	None
SetProducts	Sets the list of product names for which to read log sessions.	<code><string[]> products</code> : Array of product names.	None

¹⁾ <Type> Name: Description**Related topics****Basics**

[Migrating from ControlDesk 7.3 to 7.4 \(ControlDesk Introduction and Overview !\[\]\(003082e50e3009141f59bd5df831749f_img.jpg\)](#))

[Reading dSPACE Log Messages via the Message Reader API..... 10](#)

Examples

[Example of Reading Messages with C#..... 14](#)

[Example of Reading Messages with Python..... 12](#)

Severity Enumeration

Description

To specify the severity of a message.

Enumeration values

The enumeration has the following values:

Value	Name	Description
0	Trace	A trace message. Trace messages are usually not created. It depends on the host application if it is possible to configure the message handler to create trace messages.
1	Info	An information message.
2	Warning	A warning message.
3	Error	An error message.
4	SevereError	A severe error message.
5	SystemError	A system error message.

Value	Name	Description
6	Question	A question message.
7	Advice	An advice message.

Related topics**Basics**

[Reading dSPACE Log Messages via the Message Reader API.....](#) 10

Examples

[Example of Reading Messages with C#.....](#) 14
[Example of Reading Messages with Python.....](#) 12

Glossary

Introduction	Briefly explains the most important expressions and naming conventions used in the ControlDesk documentation.
--------------	---

Where to go from here

Information in this section

Numerics.....	36
A.....	36
B.....	37
C.....	38
D.....	42
E.....	46
F.....	49
G.....	50
H.....	50
I.....	51
K.....	53
L.....	53
M.....	54
N.....	57
O.....	57
P.....	59
Q.....	61
R.....	62

S.....	63
T.....	66
U.....	67
V.....	68
W.....	70
X.....	71

Numerics

3-D Viewer An instrument for displaying items in a 3-D environment.

A

A2L file A file that contains all the relevant information on measurement and calibration variables in an [ECU application](#) and the ECU's communication interface(s). This includes information on the variables' memory addresses and conversion methods, the memory layout and data structures in the ECU as well as [interface description data \(IF_DATA\)](#).

Acquisition An object in the [Measurement Configuration](#) controlbar that specifies the variables to be measured and their measurement configuration.

Active variable description The variable description that is currently active for a platform/device. Multiple variable descriptions can be assigned to one platform/device, but only one of them can be active at a time.

Additional write variable A scalar parameter or writable measurement variable that can be connected to an instrument in addition to the [main variable](#). When the value of the main variable changes, the changed value is also applied to all the additional write variables connected to the instrument.

Airspeed Indicator An instrument for displaying the airspeed of a simulated aircraft.

Altimeter An instrument for displaying the altitude of a simulated aircraft.

Animated Needle An instrument for displaying the value of a connected variable by a needle deflection.

Application image An image file that contains all the files that are created when the user builds a real-time application. It particularly includes the variable

description (SDF) file. To extend a real-time application, ControlDesk lets the user create an updated application image from a data set. The updated application image then contains a real-time application with an additional set of parameter values.

Artificial Horizon An instrument displaying the rotation on both the lateral and the longitudinal axis to indicate the angle of pitch and roll of a simulated aircraft. The Artificial Horizon has a pitch scale and a roll scale.

Automatic Reconnect Feature for automatically reconnecting to platform/device hardware, for example, when the ignition is turned off and on, or when the physical connection between the ControlDesk PC and the ECU is temporarily interrupted.

If the feature is enabled for a platform/device and if the platform/device is in the 'unplugged' [🔗](#) state, ControlDesk tries to re-establish the logical connection to the platform/device hardware. After the logical connection is re-established, the platform/device has the same state as before the unplugged state was detected. A measurement started before the unplugged state was detected is resumed.

Automation A communication mechanism that can be used by various programming languages. A client can use it to control a server by calling methods and properties of the server's automation interface.

Automation script A script that uses automation to control an automation server.

Axis point object [Common axis](#) [🔗](#)

B

Bar An instrument (or a value cell type of the [Variable Array](#) [🔗](#)) for displaying a numerical value as a bar deflection on a horizontal or vertical scale.

Bitfield A value cell type of the [Variable Array](#) [🔗](#) for displaying and editing the source value of a parameter as a bit string.

Bookmark A marker for a certain event during a measurement or recording.

Browser An instrument for displaying HTML and TXT files. It also supports Microsoft Internet Explorer® plug-ins that are installed on your system.

Bus communication replay A feature of the [Bus Navigator](#) [🔗](#) that lets you replay logged bus communication data from a log file. You can add replay nodes

to the Bus Navigator tree for this purpose. You can specify filters to replay selected parts of the [logged bus communication](#).

Bus configuration A configuration of all the controllers, communication matrices, and messages/frames/PDUs of a specific communication bus such as CAN. ControlDesk lets you display and experiment with bus configurations in the [Bus Navigator](#).

Bus connection A mode for connecting dSPACE real-time hardware to the host PC via bus. The list below shows the possible bus connections:

- dSPACE real-time hardware installed directly in the host PC
- dSPACE real-time hardware installed in an expansion box connected to the host PC via dSPACE link board

Bus instrument An instrument available for the [Bus Navigator](#). It can be configured for different purposes, for example, to display information on received messages (RX messages) or to manipulate and transmit messages (TX messages). The instrument is tailor-made and displays only the message- and signal-specific settings which are enabled for display and/or manipulation by ControlDesk during run time.

Bus logging A feature of the [Bus Navigator](#) that lets you log raw bus communication data. You can add logger nodes on different hierarchy levels of the Bus Navigator tree for this purpose. You can specify filters to log filtered bus communication. The logged bus communication can be [replayed](#).

Bus monitoring A feature of the [Bus Navigator](#) that lets you observe bus communication. You can open monitoring lists and add monitor nodes on different hierarchy levels of the Bus Navigator tree for this purpose. You can specify filters to monitor filtered bus communication.

Bus Navigator A [controlbar](#) for handling bus messages, such as CAN messages, LIN frames, and Ethernet packets.

Bus statistics A feature of the [Bus Navigator](#) that lets you display and log statistical information on the bus load during [bus monitoring](#).

Bypassing A method for replacing an existing ECU function by running a new function.

C

Calculated variable A scalar variable that can be measured and recorded, and that is derived from one or more *input variables*.

The following input variable types are supported:

- [Measurement variables](#)
- Single elements of [measurement arrays](#) or [value blocks](#)
- Scalar [parameters](#), or existing calculated variables

The value of a calculated variable is calculated via a user-defined *computation formula* that uses one or more input variables.

Calculated variables are represented by the  symbol.

CalDemo ECU A demo program that runs on the same PC as ControlDesk. It simulates an ECU on which the Universal Measurement and Calibration (XCP [🔗](#)) protocol and the Unified Diagnostic Services (UDS) protocol are implemented.

The CalDemo ECU allows you to perform parameter calibration, variable measurement, and ECU diagnostics with ControlDesk under realistic conditions, but without having to have a real ECU connected to the PC. Communication between the CalDemo ECU and ControlDesk can be established via XCP on CAN or XCP on Ethernet, and UDS on CAN.

Tip

If communication is established via XCP on Ethernet, the CalDemo ECU can also run on a PC different from the PC on which ControlDesk is running.

The memory of the CalDemo ECU consists of two areas called [memory page](#) [🔗](#). Each page contains a complete set of parameters, but only one page is accessible by the CalDemo ECU at a time. You can easily switch the memory pages of the CalDemo ECU to change from one [parameter](#) [🔗](#) to another in a single step.

Two ECU tasks run on the CalDemo ECU:

- ECU task #1 runs at a fixed sample time of 5 ms. In ControlDesk's Measurement Configuration, ECU task #1 is related to the time-based 5 ms, 10 ms, 50 ms and 100 ms measurement rasters of the CalDemo ECU.
 - ECU task #2 has a variable sample time. Whenever the CalDemo ECU program is started, the initial sample time is 5 ms. This can then be increased or decreased by using the dSPACE CalDemo dialog.
- ECU task #2 is related to the extEvent measurement raster of the CalDemo ECU.

The CalDemo ECU can also be used to execute diagnostic services and jobs, handle DTCs and perform measurement and calibration via ECU diagnostics.

The CalDemo ECU program is run by invoking **CalDemo.exe**. The file is located in the `. \Demos\CalDemo` folder of the ControlDesk installation.

Calibration Changing the [parameter](#) [🔗](#) values of [real-time application](#) [🔗](#)s or [ECU application](#) [🔗](#)s.

Calibration memory segment Part of the memory of an ECU containing the calibratable parameters. Memory segments can be defined as `MEMORY_SEGMENT` in the A2L file. ControlDesk can use the segments to evaluate the memory pages of the ECU.

ControlDesk lets you perform the calibration of:

- Parameters inside memory segments
- Parameters outside memory segments
- Parameters even if no memory segments are defined in the A2L file.

CAN Bus Monitoring device A device that monitors the data stream on a CAN bus connected to the ControlDesk PC.

The CAN Bus Monitoring device works, for example, with PC-based CAN interfaces such as the DCI-CAN2 or the DCI-CAN/LIN1.

The device supports the following variable description file types:

- DBC
- FIBEX
- AUTOSAR system description (ARXML)

CANGenerator A demo program that simulates a CAN system, that is, it generates signals that can be measured and recorded with ControlDesk. The program runs on the same PC as ControlDesk.

The CANGenerator allows you to use the [CAN Bus Monitoring device](#) under realistic conditions, but without having to have any device hardware connected to the PC.

The CAN (Controller Area Network) protocol is used for communication between the CANGenerator and ControlDesk. However, since the CANGenerator runs on the same PC as ControlDesk, ControlDesk does not communicate with the device via a real CAN channel, but via a *virtual CAN channel* implemented on the host PC.

You can start the CAN generator program by running **CANGenerator.exe**. The file is located in the `. \Demos\CANGenerator` folder of the ControlDesk installation.

Capture A data packet of all the measurement variables assigned to a [measurement raster](#). The packet comprises the data that results from a single triggering of the raster.

CCP Abbreviation of CAN Calibration Protocol. This protocol can be implemented on electronic control units (ECUs) and allows users to access ECUs with measurement and calibration systems (MCS) such as ControlDesk.

The basic features of CCP are:

- Read and write access to the ECU memory, i.e., providing access for calibration
- Synchronous data acquisition
- Flash programming for ECU development purposes

The CCP protocol was developed by ASAM e.V. (Association for Standardization of Automation and Measuring Systems e.V.). For the protocol specification, refer to <http://www.asam.net>.

The following device supports ECUs with an integrated CCP service:

- [CCP device](#)

CCP device A device that provides access to an ECU with CCP connected to the ControlDesk PC via CAN, for example, for measurement and calibration purposes via [CCP \(CAN Calibration Protocol\)](#).

Check Button An instrument (or a cell type of the [Variable Array](#)) for displaying whether the value of a connected variable matches predefined values or for writing a predefined value to a connected variable.

cmdloader A command line tool for handling applications without using the user interface of an experiment software.

Common axis A [parameter](#) that consists of a 1-dimensional array containing axis points. A common axis can be referenced by one or more [curves](#) and/or [map](#)s. Calibrating the data points of a common axis affects all the curves and/or maps referencing the axis.

Common axes are represented by the  symbol.

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`


or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Computation method A formula or a table that defines the transformation of a source value into a converted value (and vice versa). In addition to the computation methods defined in the variable description file, ControlDesk provides the `__Identity` computation method which means the converted and the source value are equal.

Connected A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- A platform/device must be in the 'connected' state before it can change to the 'measuring/recording' or 'online calibration started' state.
- Online calibration is impossible. ControlDesk did not yet adjust the memory segments containing calibration data in the platform/device and on the corresponding hardware. Offline calibration is possible.
- Platform/device configuration is not possible. However, you can invoke platform/device configuration for a platform/device that is in the connected state. ControlDesk temporarily sets the platform/device to the disconnected state.

The 'connected' platform/device state is indicated by the  icon.

Connection mode dSPACE real-time systems can be installed within the host PC or connected to the host via a bus interface and/or via Ethernet. When the Ethernet is being used, different network clients might exist. The connection type being used and, in the case of Ethernet, the network client being used, determine the dSPACE systems that can be accessed.

Control primitive A special diagnostic communication object for changing communication states or protocol parameters, or for identifying (ECU) variants.

Controlbar A window or pane outside the working area. Can be docked to an edge of the main window or float in front of it. A controlbar can contain a

document, such as a layout, or a tool, such as the **Bus Navigator**. It can be grouped with other controlbars in a window with tabbed pages.

ControlDesk The main version of ControlDesk for creating and running experiments, and for accessing dSPACE real-time hardware and VEOS. The functionality can be extended by optional software modules.

ControlDesk - Operator Version A version of ControlDesk that provides only a subset of functionality for running existing experiments. The functionality can be extended by optional software modules.

ControlDesk Bus Navigator Module An optional software module for ControlDesk for handling bus messages, such as CAN, LIN, and FlexRay messages, frames, and PDUs and Ethernet packets.

ControlDesk ECU Diagnostics Module An optional software module for ControlDesk that facilitates the calibration and validation of ECU diagnostic functions.

ControlDesk ECU Interface Module An optional software module for ControlDesk for calibration and measurement access to electronic control units (ECUs). The module is also required for calibration and measurement access to virtual ECUs (V-ECUs) used in SIL testing scenarios.

ControlDesk Signal Editor Module An optional software module for ControlDesk for the graphical definition and execution of signal generators for stimulating model variables of real-time/offline simulation applications.

Controller board Single-board hardware computing the real-time application. Contains a real-time processor for fast calculation of the model and I/O interfaces for carrying out the control developments.

Conversion table A table that specifies the [value conversion](#) of a source value into a converted value. In the case of [verbal conversion](#), the converted value is a string that represents one numerical value or a range of numerical values.

Conversion type The type of a [computation method](#), for example a linear function or a verbal computation method.

Curve A [parameter](#) that consists of

- A 1-dimensional array containing the axis points for the x-axis. This array can also be specified by a reference to a [common axis](#).
- Another 1-dimensional array containing data points. The curve assigns one data point to each axis point.

Curves are represented by the  symbol.

D

DAQ module A hardware module for the acquisition of physical quantities

Data Cursor One or two cursors that are used to display the values of selected chart positions in a [Time Plotter](#) or an [Index Plotter](#).

Data logger An object in the [Measurement Configuration](#) controlbar that lets you configure a [data logging](#).

Data logger signal list A list that contains the variables to be included in subsequent [data loggings](#) on real-time hardware.

Data logging The recording of data on dSPACE real-time hardware that does not require a physical connection between the host PC and the real-time hardware. In contrast to [flight recording](#), data logging is configured in ControlDesk.

Data set A set of the parameters and their values of a platform/device derived from the variable description of the platform/device. There are different types of data sets:

- [Reference data set](#)
- [Sub data set](#)
- [Unassigned data set](#)
- [Working data set](#)

DCI-CAN/LIN1 A dSPACE-specific interface between the host PC and the CAN/CAN FD bus and/or LIN bus. The DCI-CAN/LIN1 transfers messages between the CAN-/LIN-based devices and the host PC via the universal serial bus (USB).

DCI-CAN2 A dSPACE-specific interface between the host PC and the CAN bus. The DCI-CAN2 transfers CAN and CAN FD messages between the CAN-based devices and the host PC via the universal serial bus (USB).

DCI-GSI2 Abbreviation of *dSPACE Communication Interface - Generic Serial Interface 2*. A dSPACE-specific interface for ECU calibration, measurement and ECU interfacing.

DCI-GSI2 device A device that provides access to an ECU with DCI-GSI2 connected to the ControlDesk PC for measurement, calibration, and bypassing purposes via the ECU's debug interface.

DCI-KLine1 Abbreviation of *dSPACE Communication Interface - K-Line Interface*. A dSPACE-specific interface between the host PC and the diagnostics bus via K-Line.

Debug interface An ECU interface for diagnostics tasks and flashing.

Default raster A platform-/device-specific [measurement raster](#) that is used when a variable of the platform/device is connected to a [plotter](#) or a [recorder](#), for example.

Deposition definition A definition specifying the sequence in which the axis point values of a curve or map are deposited in memory.

Device A software component for carrying out [calibration](#) and/or [measurement](#), [bypassing](#), [ECU flash programming](#), or [ECU diagnostics](#) tasks.

ControlDesk provides the following devices:

- Bus devices:
 - [CAN Bus Monitoring device](#)
 - [Ethernet Bus Monitoring device](#)
 - [LIN Bus Monitoring device](#)
- [ECU Diagnostics device](#)
- [GNSS device](#)
- Measurement and calibration devices:
 - [CCP device](#)
 - [DCI-GSI2 device](#)
 - [XCP on CAN device](#)
 - [XCP on Ethernet device](#)

Each device usually has a [variable description](#) that specifies the device's variables to be calibrated and measured.

Diagnostic interface Interface for accessing the [fault memory](#) of an ECU.

Diagnostic job (often called Java job) Programmed sequence that is usually built from a sequence of the [diagnostic service](#). A diagnostic job is either a single-ECU job or a multiple-ECU job, depending on whether it communicates with one ECU or multiple ECUs.

Diagnostic protocol A protocol that defines how an ECU communicates with a connected diagnostic tester. The protocol must be implemented on the ECU and on the tester. The [diagnostics database](#) specifies the diagnostic protocol(s) supported by a specific ECU.

ControlDesk's ECU Diagnostics device supports CAN and K-Line as the physical layers for communication with an ECU connected to the ControlDesk PC. For information on the supported diagnostic protocols with CAN and K-Line, refer to [Basics of ECU Diagnostics with ControlDesk \(ControlDesk ECU Diagnostics\)](#).

Diagnostic service A service implemented on the ECU as a basic diagnostic communication element. Communication is performed by selecting a service, configuring its parameters, executing it, and receiving the ECU results. When a service is executed, a defined request is sent to the ECU and the ECU answers with a specific response.

Diagnostic trouble code (DTC) A hexadecimal index for the identification of vehicle malfunctions. DTCs are stored in the [fault memory](#) of ECUs and can be read by diagnostic testers.

Diagnostics database A database that completely describes one or more ECUs with respect to diagnostics communication. ControlDesk supports the ASAM MCD-2 D [ODX database](#) format, which was standardized by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.). For the format specification, refer to <http://www.asam.net>.

Proprietary diagnostics database formats are not supported by ControlDesk.

Diagnostics Instrument An instrument for communicating with an ECU via the diagnostic protocol using [diagnostic services](#), [diagnostic jobs](#), and [control primitives](#).

Disabled A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware.
- When a platform/device is disabled, ControlDesk does not try to establish the logical connection for that platform/device. Any communication between the platform/device hardware and ControlDesk is rejected.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disabled' platform/device state is indicated by the  icon.

Disconnected A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware.
- When a platform/device is in the disconnected state, ControlDesk does not try to re-establish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disconnected' platform/device state is indicated by the  icon.

Display An instrument (or a value cell type of the [Variable Array](#)) for displaying the value of a scalar variable or the text content of an ASCII variable.

Documents folder A standard folder for user-specific documents.

%USERPROFILE%\Documents\dSPACE\
<ProductName>\<VersionNumber>

DS1006 Processor Board platform A platform that provides access to a DS1006 Processor Board connected to the host PC for HIL simulation and function prototyping purposes.

DS1007 PPC Processor Board platform A platform that provides access to a single multicore DS1007 PPC Processor Board or a DS1007 multiprocessor system consisting of two or more DS1007 PPC Processor Boards, connected to the host PC for HIL simulation and function prototyping purposes.

DS1104 R&D Controller Board platform A platform that provides access to a DS1104 R&D Controller Board installed in the host PC for function prototyping purposes.

DS1202 MicroLabBox platform A platform that provides access to a MicroLabBox connected to the host PC for function prototyping purposes.

DsDAQ service A service in a [real-time application](#) or [offline simulation application \(OSA\)](#) that provides measurement data from the application to the

host PC. Unlike the [host service](#), the DsDAQ service lets you perform, for example, triggered measurements with complex trigger conditions.

The following platforms support applications that contain the DsDAQ service:

- [DS1007 PPC Processor Board platform](#)
- [DS1202 MicroLabBox platform](#)
- [MicroAutoBox III platform](#)
- [SCALEXIO platform](#)
- [VEOS platform](#)
- [XIL API MAPort platform](#)

dSPACE Calibration and Bypassing Service An ECU service for measurement, calibration, bypassing, and ECU flash programming. The dSPACE Calibration and Bypassing Service can be integrated on the ECU. It provides access to the ECU application and the ECU resources and is used to control communication between an ECU and a calibration and/or bypassing tool.

With the dSPACE Calibration and Bypassing Service, users can run measurement, calibration, bypassing, and flash programming tasks on an ECU via the DCI-GSI2. The service is also designed for bypassing ECU functions using dSPACE prototyping hardware by means of the RTI Bypass Blockset in connection with DPMEM PODs. The dSPACE Calibration and Bypassing Service allows measurement, calibration, and bypassing tasks to be performed in parallel.

dSPACE Internal Bypassing Service An ECU service for on-target prototyping. The dSPACE Internal Bypassing Service can be integrated in the ECU application. It lets you add additional functions to be executed in the context of the ECU application without the need for recompiling the ECU application.

dSPACE Log A collection of errors, warnings, information, questions, and advice issued by all dSPACE products and connected systems over more than one session.

dSPACE system A hardware system such as a MicroAutoBox III or SCALEXIO system on which the [real-time application](#) runs.

Duration trigger A [trigger](#) that defines a duration. Using a duration trigger, you can, for example, specify the duration of data acquisition for a [measurement raster](#). A duration trigger can be used as a [stop trigger](#).

E

ECU Abbreviation of *electronic control unit*.

ECU application A sequence of operations executed by an ECU. An ECU application is mostly represented by a group of files such as [ECU Image files](#), MAP files, [A2L files](#) and/or software module description files.

ECU calibration interface Interface for accessing an ECU by either emulating the ECU's memory or using a communication protocol (for example, XCP on CAN).

ECU diagnostics Functions such as:

- Handling the ECU fault memory: Entries in the ECU's fault memory can be read, cleared, and saved.
- Executing diagnostic services and jobs: Users can communicate with an ECU via a diagnostic protocol using diagnostic services, diagnostic jobs, and control primitives.

ControlDesk provides the [ECU Diagnostics device](#) device to access ECUs for diagnostic tasks. Communication is via [diagnostic protocol](#)s implemented on the ECUs.

ECU diagnostics with ControlDesk are completely based on Open Diagnostic Data Exchange (ODX), the ASAM MCD-2 D diagnostics standard.

ControlDesk provides the [Fault Memory Instrument](#) and the [Diagnostics Instrument](#) for ECU diagnostics tasks.

ECU Diagnostics device A device that provides access to ECUs connected to the ControlDesk PC via CAN or K-Line for diagnostics or flash programming purposes.

ControlDesk provides the *ECU Diagnostics v2.0.2* device, which supports the ASAM MCD-3 D V2.0.2 standard.

ControlDesk supports the following ODX database standards:

- ASAM MCD-2 D V2.0.1
- ASAM MCD-2 D V2.2.0 (ISO 22901-1)

ECU flash programming A method by which new code or data is stored in ECU flash memory.

ECU Image file A binary file that is part of the [ECU application](#). It usually contains the code of an ECU application and the data of the parameters within the application. It can be stored as an Intel Hex (HEX) or Motorola S-Record (MOT or S19) file.

EESPort Configurations controlbar A [controlbar](#) for configuring [error configuration](#)s.

Electrical error simulation Simulating electrical errors such as loose contacts, broken cables, and short-circuits, in the wiring of an ECU. Electrical error simulation is performed by the failure simulation hardware of an HIL simulator.

Electrical Error Simulation port (EESPort) An *Electrical Error Simulation port* (EESPort) provides access to a failure simulation hardware for simulating electrical errors in an ECU wiring according to the ASAM AE XIL API standard.

The configuration of the EESPort is described by a hardware-dependent *port configuration* and one or more *error configurations*.

Environment model A model that represents a part or all of the ECU's environment in a simulation scenario.

The environment model is a part of the [simulation system](#).

Environment VPU The executable of an [environment model](#) built for the VEOS platform. An environment VPU is part of an offline simulation application (OSA).

Error An electrical error that is specified by:

- An error category
- An error type
- A load type

Error category The error category defines how a signal is disturbed. Which errors you can create for a signal depends on the connected failure simulation hardware.

Error configuration An XML file that describes a sequence of errors you want to switch during electrical error simulation. Each error configuration comprises error sets with one or more errors.

Error set An error set is used to group errors (pin failures).

Error type The error type specifies the way an error category – i.e., an interruption or short circuit of signals – is provided. The error type defines the disturbance itself.

Ethernet Bus Monitoring device A device that monitors the data stream on an Ethernet network connected to the ControlDesk PC.

The device supports the following variable description file type:

- AUTOSAR system description (ARXML)

Ethernet connection A mode for connecting dSPACE real-time hardware to the host PC via Ethernet. The list below shows the possible Ethernet connections:

- dSPACE real-time hardware installed in an expansion box connected to the host PC via Ethernet.
- MicroAutoBox II/III and MicroLabBox connected via Ethernet.

Ethernet decoding A feature of the [Bus Navigator](#) that lets you view protocol data and raw data of an Ethernet frame.

Event An event that is triggered by an action performed in ControlDesk.

Event context The scope of validity of [event source](#)s and [event](#)s. There is one [event handler](#) code area for each event context.

Event handler Code that is executed when the related [event](#) occurs.

Event management Functionality for executing custom code according to actions triggered by ControlDesk.

Event source An object providing and triggering [event](#)s. *LayoutManagement* is an example of an event source.

Event state State of an [event](#). ControlDesk provides the following event states:

- No [event handler](#) is defined
- Event handler is defined and enabled
- Event handler is defined and disabled
- Event handler is defined, but no Python code is available
- Event handler is deactivated because a run-time error occurred during the execution of the Python code

Expansion box A box that hosts dSPACE boards. It can be connected to the host PC via bus connection or via network.

Experiment A container for collecting and managing information and files required for a parameter calibration and/or measurement task. A number of experiments can be collected in a project but only one of them can be active.

Extension script A Python script (PY or PYC file) that is executed each time ControlDesk starts up. An extension script can be executed for all users or user-specifically.

F

Failure insertion unit Hardware unit used with dSPACE simulators to simulate failures in the wiring of an ECU, such as broken wire and short circuit to ground.


Fault memory Part of the ECU memory that stores diagnostic trouble code (DTC) entries with status and environment information.

Fault Memory Instrument An instrument for reading, clearing, and saving the content of the ECU's [fault memory](#).

Firmware update An update for the firmware installed in the board's flash memory. Firmware should be updated if it is older than required by the real-time application to be downloaded.

Fixed axis An axis with data points that are not deposited in the ECU memory. Unlike a [common axis](#), a fixed axis is specified within a [curve](#) or [map](#). The parameters of a fixed axis cannot be calibrated.

Fixed parameter A [parameter](#) that has a fixed value during a running simulation. Changing the value of a fixed parameter does not immediately affect the simulation results. The affect occurs only after you stop the simulation and

start it again. A fixed parameter is represented by an added pin in its symbol, for example: .

Flash job A specific diagnostic job for flashing the ECU memory. A flash job implements the process control for flashing the ECU memory, such as initialization, security access, writing data blocks, etc.

Flight recording The recording of data on dSPACE real-time hardware that does not require a physical connection between the host PC and the real-time hardware. In contrast to [data logging](#), flight recording is not configured in ControlDesk but via RTI and RTLib.

Frame An instrument for adding a background frame to a layout, for example, to visualize an instrument group.

G

Gauge An instrument for displaying the value of the connected variable by a needle deflection on a circular scale.

Gigalink module A dSPACE board for connecting several processor boards in a multiprocessor system. The board allows high-speed serial data transmission via fiber-optic cable.

GNSS data Positioning and timing data that is transmitted by a Global Navigation Satellite System (GNSS), such as GPS, GLONASS, or Galileo. GNSS receivers use this data to determine their location.

GNSS device A device that provides positioning data from a GNSS receiver (e.g., a serial GPS mouse) in ControlDesk.

ControlDesk provides the *GNSS (GPS, GLONASS, Galileo, ...)* device that supports various global navigation satellite systems.

GPX file An XML file that contains geodata, such as waypoints, routes, or tracks. In ControlDesk, you can import GPX files to visualize GNSS positioning data in a Map instrument.

Group A collection of variables that are grouped according to a certain criterion.

H

Heading Indicator An instrument displaying the heading direction of a simulated aircraft on a circular scale.

Host service A service in a [real-time application](#) that provides measurement data from the application to the host PC.

The following platforms support applications that contain the host service:

- [DS1006 Processor Board platform](#)
- [DS1104 R&D Controller Board platform](#)
- [MicroAutoBox platform](#)
- [Multiprocessor System platform](#)

Index Plotter A [plotter instrument](#) for displaying signals that are measured in an event-based raster (index plots).

Input quantity A measurement variable that is referenced by a common axis and that provides the input value of that axis.

Instrument An on-screen representation that is designed to monitor and/or control simulator variables interactively and to display data captures. Instruments can be arranged freely on [layout](#)s.

The following instruments can be used in ControlDesk:

- [3-D Viewer](#)
- [Airspeed Indicator](#)
- [Altimeter](#)
- [Animated Needle](#)
- [Artificial Horizon](#)
- [Bar](#)
- [Browser](#)
- [Bus Instrument](#)
- [Check Button](#)
- [Diagnostics Instrument](#)
- [Display](#)
- [Fault Memory Instrument](#)
- [Frame](#)
- [Gauge](#)
- [Heading Indicator](#)
- [Index Plotter](#)
- [Invisible Switch](#)
- [Knob](#)
- [Multistate Display](#)
- [Multiswitch](#)
- [Numeric Input](#)
- [On/Off Button](#)

- [Push Button](#)
- [Radio Button](#)
- [Selection Box](#)
- [Slider](#)
- [Sound Controller](#)
- [Static Text](#)
- [Steering Controller](#)
- [Table Editor](#)
- [Time Plotter](#)
- [Variable Array](#)
- [XY Plotter](#)

Instrument Navigator A [controlbar](#) that displays a tree with all the [instrument](#)s of the active [layout](#) and all the variables that are connected to them. The Instrument Navigator's main function is easy selection of instruments in complex layouts.

Instrument script A Python script used to extend the functionality of an [instrument](#).

Instrument Selector A [controlbar](#) that provides access to ControlDesk's [instrument](#)s. The instruments can be placed on a [layout](#) via double-click or drag & drop.

Interface description data (IF_DATA) An information structure, mostly provided by an [A2L file](#), describing the type, features and configuration of an implemented ECU interface.

Internal Interpreter ControlDesk's built-in programming interface for editing, running and importing Python scripts. It contains an [Interpreter controlbar](#) where the user can enter Python commands interactively and which displays output and error messages of Python commands.

Interpreter controlbar A [controlbar](#) that can be used to execute line-based commands. It is used by the [Internal Interpreter](#) to print out Python standard error messages and standard output during the execution or import of Python scripts.

Invisible Switch An instrument for defining an area that is sensitive to mouse operations.

IOCNET IOCNET (I/O carrier network) is a dSPACE-specific high-speed serial communication bus that connects all the real-time hardware in a SCALEXIO system. IOCNET can also be used to build a multiprocessor system that consists of multiple SCALEXIO processor hardware components.

K

Knob An instrument for displaying and setting the value of the connected variable by means of a knob on a circular scale.

L

Label list A list of user-defined variables that can be used for saving connected variables, etc.

Layout A window with [instrument](#) ^Us connected to variables of one or more simulation models.

Layout Navigator A [controlbar](#) ^U that displays all opened [layout](#) ^Us. It can be used for switching between layouts.

Layout script A Python script used to extend the functionality of a [layout](#) ^U.

Leading raster The [measurement raster](#) ^U that specifies the [trigger](#) ^U settings for the [Time Plotter](#) ^U display. The leading raster determines the time range that is visible in the plotter if a start and stop trigger is used for displaying the signals.

LIN Bus Monitoring device A device that monitors the data stream on a LIN bus connected to the ControlDesk PC.

The LIN Bus Monitoring device works, for example, with PC-based LIN interfaces. The device supports the following variable description file types:

- LDF
- FIBEX
- AUTOSAR system description (ARXML)

Load type The load type specifies the option to disturb a signal with or without load rejection.

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>

Logical link A representation of an ECU specified in the diagnostics database. A logical link contains information on the ECU itself, and all the information required for accessing it, such as the [diagnostic protocol](#) ^U used for

communication between the ECU and ControlDesk. Each logical link is represented by a unique short name in the [ODX database](#).

Look-up table A look-up table maps one or more input values to one output value. You have to differentiate between the following look-up table types:

- A 1-D look-up table maps one input value to one output value.
- A 2-D look-up table maps two input values to one output value.
- An n-D look-up table maps multidimensional table data with 3 or more input values to one output value.

Look-up table is a generic term for [curves](#) and [maps](#).

M

Main variable A scalar variable that is visualized in an instrument that can be used to change parameter values. In addition to the main variable, [additional write variable](#)s can also be connected to (but not visualized in) the same instrument. When you change the value of the main variable in an instrument, the changed value is also applied to all the additional write variables connected to that instrument.

Map A [parameter](#) that consists of

- A 1-dimensional array containing the axis points for the x-axis. This array can also be specified by a reference to a [common axis](#).
- A 1-dimensional array containing the axis points for the y-axis. This array can also be specified by a reference to a [common axis](#).
- A 2-dimensional array containing data points. The map assigns one data point of the array to each pair of x-axis and y-axis points.

Maps are represented by the  symbol.

Map file A file that contains symbols (symbolic names) and their physical addresses. It is generated during a build process of an ECU application.

Map instrument A customized [Browser](#) instrument. It uses an instrument script to open a web map and connect positioning data to the map. The Map instrument offers prepared connection nodes to connect variables with [GNSS data](#).

Measurement Viewing and analyzing the time traces of [variables](#), for example, to observe the effects of ECU parameter changes.

ControlDesk provides various [instruments](#) for measuring variables.

Measurement (variable type) A scalar variable that can be measured, including individual elements of a measurement array.

Measurement variables are represented by the  symbol.

Measurement array A 1-, 2-, or 3-dimensional array of measurement variables. In variable lists, ControlDesk displays entries for the measurement array itself and for each array element.

Measurement arrays are represented by the  symbol.

Measurement buffer A ring buffer that buffers measurement data at the start of a [measurement](#). The measurement buffer size determines the amount of data that can be buffered. Earlier values are overwritten by later values when the buffer capacity is exceeded (buffer overflow).

Measurement Configuration A [controlbar](#) that allows you to configure [measurement](#), [recording](#) and [data logging](#).

Measurement Data API Application programming interface for accessing measurement data. The API lets the user access measurement data without having to use ControlDesk.

Measurement Data Pool A [controlbar](#) that provides access to measurement data recorded in measurement data files.

Measurement raster Specification of how often a value of a [variable](#) is updated during a [measurement](#). A measurement raster can be derived from a [measurement service](#).

Measurement service The generic term for the following services:

- [CCP](#) service
- [DsDAQ service](#)
- [Host service](#)
- [XCP](#) service

Measurement signal list A list containing the variables to be included in subsequent measurements and recording. The list is global for all platforms/devices of the current experiment. The measurement signal list is available in the configuration area of the [Measurement Configuration](#) controlbar.

Measurement variable Any variable type that can be measured but not calibrated.

Measuring/recording A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- Online calibration is possible. Parameter values can be changed directly on the platform/device hardware.
- A measurement (or recording) is running.
- Platform/device configuration is not possible.

The 'measuring' / 'recording' platform/device state is indicated by the  icon.

Memory page An area of a calibration memory. Each page contains a complete set of parameters of the platform/device hardware, but only one of the pages is "visible" to the microcontroller of the ECU or the real-time processor (RTP) of the platform hardware at a time.

ControlDesk supports platform/device hardware with up to two memory pages. These are usually the [working page](#) and the [reference page](#). The parameter values on the two memory pages usually are different. ControlDesk lets you switch from one page to the other, so that when parameters are changed on one page, the changes can be made available to the ECU or prototyping hardware via a single page switch.

Messages controlbar A [controlbar](#) displaying a history of all error and warning messages that occur during work with ControlDesk.

MicroAutoBox III platform A platform that provides access to a MicroAutoBox III connected to the host PC for function prototyping purposes such as [Bypassing](#).

MicroAutoBox platform A platform that provides access to a MicroAutoBox II connected to the host PC for function prototyping purposes such as bypassing.

Mirrored memory A memory area created by ControlDesk on the host PC that mirrors the contents of the available memory pages of calibration and prototyping hardware. For hardware with two memory pages, the mirrored memory is divided into a reference and a working page, each of them containing a complete set of parameters. When a calibration or prototyping platform/device is added to an experiment, ControlDesk initially fills the available memory pages of the mirrored memory with the contents of the [ECU Image file](#) (initial filling for calibration devices) or with the contents of the SDF file (initial filling for platforms).

- **Mirrored memory for offline calibration**
Parameter values can even be changed [offline](#). Changes to parameter values that are made offline affect only the mirrored memory.
- **Offline-to-online transition for online calibration**
For online calibration, an offline-to-online transition must be performed. During the transition, ControlDesk compares the [memory page](#)s of the hardware of each platform/device with the corresponding pages of the mirrored memory. If the pages differ, the user has to equalize them by uploading them from the hardware to the host PC, or downloading them from the host PC to the hardware.
- **Mirrored memory for online calibration**
When ControlDesk is in the online mode, parameter value changes become effective synchronously on the memory pages of the hardware and in the mirrored memory. In other words, parameter values on the hardware and on the host PC are always the same while you are performing online calibration.

Modular system A dSPACE processor board and one or more I/O boards connected to it.

Multi-capture history The storage of all the [capture](#)s acquired during a [triggered measurement](#). The amount of stored data depends on the measurement buffer.

Multi-pin error A feature of the SCALEXIO concept for electrical error simulation that lets you simulate a short circuit between three or more signal

channels and/or bus channels. The channels can be located on the same or different boards or I/O units. You can simulate a short circuit between:

- Channels of the same signal category (e.g., four signal generation channels)
- Channels of different signal categories (e.g., three signal generation channels and two signal measurement channels)
- Signal channels and bus channels (e.g., two signal generation channels, one signal measurement channel, and one bus channel)

Multiple electrical errors A feature of the SCALEXIO concept for electrical error simulation that lets you switch electrical errors at the same time or in succession. For example, you can simulate an open circuit for one channel and a short circuit for another channel at the same time, without deactivating the first error.

Multiprocessor System platform A platform that provides access to:

- A multicore application running on a multicore DS1006 board
- A multiprocessor application on a multiprocessor system consisting of two or more DS1006 processor boards interconnected via Gigalink.

ControlDesk handles a multiprocessor/multicore system as a unit and uses one system description file (SDF file) to load the applications to all the processor boards/cores in the system.

Multistate Display An instrument for displaying the value of a variable as an LED state and/or as a message text.

Multistate LED A value cell type of the [Variable Array](#) for displaying the value of a variable as an LED state.

Multiswitch An instrument for changing variable values by clicking sensitive areas in the instrument and for visualizing different states depending on the current value of the connected variable.

N

Numeric Input An instrument (or a value cell type of the [Variable Array](#)) for displaying and setting the value of the connected variable numerically.

O

Observing variables Reading variable values cyclically from the dSPACE real-time hardware and displaying their current values in ControlDesk, even if no [measurement](#) is running. Variable observation is performed without using a measurement buffer, and no value history is kept.

For platforms that support variable observation, variable observation is available for [parameters](#) and [measurement variables](#) that are visualized in [single-shot instruments](#) (all instruments except for a [plotter](#)). If you visualize a variable in a single-shot instrument, the variable is not added to the [measurement signal list](#). Visualizing a parameter or measurement variable in a plotter automatically adds the variable to the measurement signal list.

ControlDesk starts observing variables if one of the following conditions is true:

- [Online Calibration is started](#) for the platform.
All the parameters and measurement variables that are visualized in single-shot instruments are observed.
- [Measurement is started](#) for the platform.
All the visualized parameters and measurement variables that are not activated for measurement in the measurement signal list are observed. Data of the activated parameters and measurement variables is acquired using measurement rasters.

ODX database Abbreviation of Open Diagnostic Data Exchange, a [diagnostics database](#) that is the central ECU description for working with an [ECU Diagnostics device](#) in ControlDesk. The ODX database contains all the information required to perform diagnostic communication between ControlDesk and a specific ECU or set of ECUs in a vehicle network. ControlDesk expects the database to be compliant with ASAM MCD-2 D (ODX).

Offline State in which the parameter values of platform/device hardware in the current experiment cannot be changed. This applies regardless of whether or not the host PC is physically connected to the hardware.

The [mirrored memory](#) allows parameter values to be changed even offline.

Offline simulation A PC-based simulation in which the simulator is not connected to a physical system and is thus independent of the real time.

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 the VEOS Player, the OSA file can be downloaded to VEOS.

An OSA contains one or more [VPUs](#), such as V-ECUs and/or environment VPUs.


On/Off Button An instrument (or a value cell type of the [Variable Array](#)) for setting the value of the connected parameter to a predefined value when the button is pressed (On value) and released (Off value).

Online calibration started A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- Online calibration is possible. Parameter values can be changed directly on the platform/device hardware.
- Platform/device configuration is not possible.

Before starting online calibration, ControlDesk lets you compare the [memory page](#)s on the platform/device hardware with the corresponding pages of the [mirrored memory](#). If the parameter values on the pages differ, they must be

equalized by uploading the values from the hardware to ControlDesk, or downloading the values from ControlDesk to the hardware. However, a page cannot be downloaded if it is read-only.

The 'online calibration started' platform/device state is indicated by the  symbol.

Operation signal A [signal](#) which represents the result of an arithmetical operation (such as addition or multiplication) between two other signals.

Operator mode A working mode of ControlDesk in which only a subset of the ControlDesk functionality is provided. You can work with existing experiments but not modify them, which protects them from unintentional changes.

Output parameter A [parameter](#) or [writable measurement](#) whose memory address is used to write the computed value of a [calculated variable](#) to.

P

Parameter Any variable type that can be calibrated.

Parameter (variable type) A scalar [parameter](#), as well as the individual elements of a [value block](#).

Scalar parameters are represented by the  symbol.

Parameter limits Limits within which parameters can be changed. Parameters have hard and weak limits.

- Hard limits

Hard limits designate the value range of a parameter that you *cannot* cross during calibration.

The hard limits of a parameter originate from the corresponding [variable description](#) and cannot be edited in ControlDesk.

- Weak limits

Weak limits designate the value range of a parameter that you *should not* cross during calibration. When you cross the value range defined by the weak limits, ControlDesk warns you.

In ControlDesk, you can edit the weak limits of a parameter within the value range given by the parameter's hard limits.

PHS (Peripheral High Speed) bus A dSPACE-specific bus for communication between a processor board and the I/O boards in a modular system. It allows direct I/O operations between the processor board (bus master) and I/O boards (bus slaves).

PHS-bus-based system A modular dSPACE system consisting of a processor board such as the DS1006 Processor Board and I/O boards. They communicate with each other via the [PHS \(Peripheral High Speed\) bus](#).

Pitch variable A variable connected to the pitch scale of an [Artificial Horizon](#).

Platform A software component representing a simulator where a simulation application is computed in real-time (on dSPACE real-time hardware) or in non-real-time (on VEOS).

ControlDesk provides the following platforms:

- [DS1006 Processor Board platform](#)
- [DS1007 PPC Processor Board platform](#)
- [DS1104 R&D Controller Board platform](#)
- [DS1202 MicroLabBox platform](#)
- [MicroAutoBox platform](#)
- [MicroAutoBox III platform](#)
- [Multiprocessor System platform](#)
- [SCALEXIO platform](#)
- [VEOS platform](#)
- [XIL API MAPort platform](#)

Each platform usually has a [variable description](#) that specifies its variables.

Platform trigger A [trigger](#) that is available for a [platform](#) and that is evaluated on the related dSPACE real-time hardware or VEOS.

Platforms/Devices controlbar A [controlbar](#) that provides functions to handle [devices](#), [platforms](#), and the [applications](#) assigned to the platforms.

Plotter instrument ControlDesk offers three plotter instruments with different main purposes:

- The [Index Plotter](#) displays signals in relation to events.
- The [Time Plotter](#) displays signals in relation to measurement time.
- The [XY Plotter](#) displays signals in relation to other signals.

Port configuration To interface the failure simulation hardware, an EESPort needs the hardware-dependent *port configuration file* (PORTCONFIG file). The file's contents must fit the connected HIL simulator architecture and its failure simulation hardware.

Postprocessing The handling of measured and recorded data by the following actions:

- Displaying measured or recorded data
- Zooming into measured or recorded signals with a [plotter](#)
- Displaying the values of measurement variables and parameters as they were at any specific point in time

Processor board A board that computes real-time applications. It has an operating system that controls all calculations and communication to other boards.

Project A container for collecting and managing the information and files required for experiment/calibration/modification tasks in a number of [experiments](#). A project collects the experiments and manages their common data.

Project controlbar A [controlbar](#) that provides access to projects and experiments and all the files they contain.

Project root directory The directory on your file system to which ControlDesk saves all the experiments and documents of a [project](#). Every project is associated with a project root directory, and several projects can use the same project root directory. The user can group projects by specifying several project root directories.

ControlDesk uses the [Documents folder](#) as the default project root directory unless a different one is specified.

Properties controlbar A [controlbar](#) providing access to the properties of, for example, platforms/devices, layouts/instruments, and measurement/recording configurations.

Proposed calibration A calibration mode in which the parameter value changes that the user makes do not become effective on the hardware until they are applied. This allows several parameter changes to be written to the hardware together. Being in proposed calibration mode is like being in the offline calibration mode temporarily.

Push Button An instrument (or a value cell type of the [Variable Array](#)) for setting the value of the connected parameter by push buttons.

Python Editor An editor for opening and editing PY files.

Q

Quick start measurement A type of measurement in which all the ECU variables configured for measurement are measured and recorded, starting with the first execution of an ECU task. ControlDesk supports quick start measurements on ECUs with DCI-GSI2, CCP, and XCP (except for XCP on Ethernet with the TCP transmission protocol).

Quick start measurement can be used to perform cold start measurements. Cold start means that the vehicle and/or the engine are cooled down to the temperature of the environment and then started. One reason for performing cold start measurements is to observe the behavior of an engine during the warm-up phase.

R

Radio Button An instrument for displaying and setting the value of the connected parameter by radio buttons.

Real-time application An application that can be executed in real time on dSPACE real-time hardware. A real-time application can be built from a Simulink model containing RTI blocks, for example.

Record layout A record layout is used to specify a data type and define the order of the data in the memory of the target system (ECU, for example). For scalar data types, a record layout allows you to add an address mode (direct or indirect). For structured (aggregated) data types, the record layout specifies all the structure elements and the order they appear in.

The **RECORD_LAYOUT** keyword in an A2L file is used to specify the various record layouts of the data types in the memory. The structural setup of the various data types must be described in such a way that a standard application system will be able to process all data types (reading, writing, operating point display etc.).

Record layout component A component of a record layout. A structured record layout consists of several components according to the ASAP2 specification. For example, the **AXIS_PTS_X** component specifies the x-axis points, and the **FNC_VALUES** component describes the function values of a map or a curve.

Recorder An object in the [Measurement Configuration](#) controlbar that specifies and executes the [recording](#) of variables according to a specific measurement configuration.

Recorder signal list A list that contains the variables to be included in subsequent [recordings](#).

Recording Saving the time traces of variables to a file. Both measurement variables and parameters can be recorded. Recorded data can be [postprocessed](#) directly in ControlDesk.

A recording can be started and stopped immediately or via a trigger:

- Immediate recording
The recording is started and stopped without delay, without having to meet a trigger condition.
- Triggered recording
The recording is not started or stopped until certain trigger conditions are met. These conditions can be defined and edited in ControlDesk.

Reduction data Additional content in an MF4 file that allows for visualizing the MF4 file data depending on the visualization resolution. Reduction data therefore improves the performance of the visualization and postprocessing of measurement data.

Reference data set A read-only data set assigned to the reference page of a device that has two [memory page](#)s. There can be only one reference data set for each device. The reference data set is read-only.

Reference page Memory area containing the parameters of an ECU. The reference page contains the read-only [reference data set](#).

Note

Some platforms/devices provide only a [working page](#). You cannot switch to a reference page in this case.

Resynchronization Mechanism to periodically synchronize the drifting timers of the platform/device hardware ControlDesk is connected to. Resynchronization means adjustment to a common time base.

Roll variable A variable connected to the roll scale of an [Artificial Horizon](#).

S

Sample count trigger A [trigger](#) that specifies the number of samples in a data capture.

A sample count trigger can be used as a [stop trigger](#).

SCALEXIO platform A platform that provides access to a single-core, multicore or multiprocessor [SCALEXIO system](#) connected to the host PC for HIL simulation and function prototyping purposes.

SCALEXIO system A dSPACE hardware-in-the-loop (HIL) system consisting of at least one processing hardware component, I/O boards, and I/O units. They communicate with each other via the [IOCNET](#). In a SCALEXIO system, two types of processing hardware can be used, a DS6001 Processor Board or a real-time industry PC as the SCALEXIO Processing Unit. The SCALEXIO system simulates the environment to test an ECU. It provides the sensor signals for the ECU, measures the signals of the ECU, and provides the power (battery voltage) for the ECU and a bus interface for restbus simulation.

SDF file The system description file that describes the files to be loaded to the individual processing units of a simulation platform. It also contains the variable description of the relevant [simulation application](#).

The SDF file is generated automatically when the [TRC file](#) is built.

Segment The minimum part a [segment signal](#) can consist of.

There are different kinds of segments to be used in segment signals:

- Segments to form synthetic signal shapes (sine, sawtooth, ramp, etc.)
- Segments to perform arithmetical operations (addition, multiplication) with other segments
- Segments to represent numerical signal data (measured data)

Segment signal A [signal](#) consisting of one or more [segment](#)s.

Selection Box An instrument for selecting a text-value entry and setting the respective numerical value for the connected variable.

Signal

- Representation of a [variable](#) measured in a specific [measurement raster](#).
- Generic term for [segment signal](#)s and [operation signal](#)s.

A signal is part of a [signal description set](#) which can be displayed and edited in the working area.

Signal description set A group of one or more [signals](#).

A signal description set and its signals can be edited in the working area by means of the [Signal Editor](#). Each signal description set is stored as an [STZ file](#) either in the Signal Description Sets folder or in the Signal Generators folder.

Signal Editor A software component to create, configure, display, and manage [signals](#) in [signal description sets](#).

Signal file A file that contains the wiring information of a simulator and that is part of the standard dSPACE documentation of dSPACE Simulator Full-Size. Normally, dSPACE generates this file when designing the simulator. Before using a failure simulation system, users can adapt the signal file to their needs.

Signal generator An STZ file containing a [signal description set](#) and optional information about the [signal mapping](#), the description of variables, and the real-time platform.

The file is located in the Signal Generators folder and used to generate, download, and control Real-Time Testing sequences, which are executed on the real-time platform to [stimulate](#) model variables in real time.

Signal Mapping A [controlbar](#) of the [Signal Editor](#) to map model variables to [signals](#) and [variable aliases](#) of a [signal generator](#).

Signal Selector A [controlbar](#) of the [Signal Editor](#). The Signal Selector provides [signals](#) and [segments](#) for arranging and configuring [signal description sets](#) in the working area.

SIL testing Abbreviation of *software-in-the-loop testing*.

Simulation and testing of individual software functions, complete virtual ECUs ([V-ECUs](#)), or even V-ECU networks on a local PC or highly parallel in the cloud independently of real-time constraints and real hardware.

Simulation application The generic term for [offline simulation application \(OSA\)](#) and [real-time application](#).

Simulation system A description of the composition of V-ECU models, environment models, real ECUs, and their interconnections required for simulating the behavior of a system. A simulation system is the basis for the generation of a [simulation application](#) for a given simulator platform.

Simulation time group Group of platforms/devices in an experiment whose simulation times are synchronized with each other. If [resynchronization](#) is enabled, ControlDesk synchronizes a simulation time group as a whole, not the single members of the group individually.

Simulator A system that imitates the characteristics or behaviors of a selected physical or abstract system.

Single-processor system A system that is based on one dSPACE processor or controller board.

Single-shot instrument An [instrument](#) that displays an instantaneous value of a connected variable without keeping a value history. In ControlDesk, all instruments except for a [plotter](#) are single-shot instruments. For [platforms](#) that support the [variable observer](#) functionality, you can use single-shot instruments to observe variables.

Slave application An application assigned to the [slave DSP](#) of a controller or I/O board. It is usually loaded and started together with the [real-time application](#) running on the corresponding main board.

Slave DSP A DSP subsystem installed on a controller or I/O board. Its [slave application](#) can be loaded together with the [real-time application](#) or separately.

Slider An instrument (or a value cell type of the [Variable Array](#)) for displaying and setting the value of the connected variable by means of a slide.

Sound Controller An instrument for generating sounds to be played.

Standard axis An axis with data points that are deposited in the ECU memory. Unlike a [common axis](#), a standard axis is specified within a [curve](#) or [map](#). The parameters of a standard axis can be calibrated, which affects only the related curve or map.

Start trigger A [trigger](#) that is used, for example, to start a [measurement raster](#). A [platform trigger](#) can be used as a start trigger.

Static Text An instrument for displaying explanations or inscriptions on the layout.

Steering Controller An instrument for changing variable values using a game controller device such as a joystick or a steering wheel.

Stimulation Writing signals to variables in real-time models during a simulation run.

Stop trigger A [trigger](#) that is used, for example, to stop a [measurement raster](#).

String A text variable in ASCII format.

Strings are represented by the  symbol.

Struct A variable with the struct data type. A struct contains a structured list of variables that can have various data types. In ControlDesk, a struct variable can contain either parameters and value blocks or measurement variables and measurement arrays. ControlDesk supports nested structs, i.e., structs that contain further structs and struct arrays as elements.

Structs are represented by the  symbol.

Struct array An array of homogeneous [struct](#)  variables.

Struct arrays are represented by the  symbol.

STZ file A ZIP file containing signal descriptions in the STI format. The STZ file can also contain additional MAT files to describe numerical signal data.

Sub data set A data set that does not contain the complete set of the parameters of a platform/device.



Symbol A symbolic name of a physical address in a MAP file. A symbol can be associated to a variable in the Variable Editor, for example, to support an address updates.

System variable A type of variable that represents internal variables of the device or platform hardware and that can be used as measurement signals in ControlDesk to give feedback on the status of the related device or platform hardware. For example, an ECU's power supply status or the simulation state of a dSPACE board can be visualized via system variables.

T

Table Editor An instrument for displaying and setting values of a connected curve, map, value block, or axis in a 2-D, 3-D, and grid view. The Table Editor can also display the values of a measurement array.

The Table Editor can be used for the following variable types:

- [Common axis](#) 
- [Curve](#) 
- [Map](#) 
- [Measurement array](#) 
- [Value block](#) 

Time cursor A cursor which is visible at the same time position in the following instruments:

- In all [Time Plotters](#)
- In all [XY Plotters](#)
- In all [bus monitoring lists](#)

You can use the time cursor to view signal values at a specific point in time. If you move the time cursor, all measured signals and the respective parameters are

updated. Instruments and bus monitoring lists display the values that are available at the selected time position.

Time Plotter A [plotter instrument](#) for displaying signals that are measured in a time-based raster (time plots).

Topology A description of the processor boards belonging to a multiprocessor system and their interconnections via Gigalinks. The topology also contains information on which Gigalink port of each processor board is connected to the Gigalink ports of other processor boards in the multiprocessor system.

Topology information is contained in the real-time application (PPC/x86/RTA) files of the multiprocessor system's processor boards.

TRC file A variable description file with information on the variables available in an [environment model](#) running on a dSPACE [platform](#).

Trigger A condition for executing an action such as starting and stopping a [measurement raster](#) or a [recorder](#).

The generic term for the following trigger types:

- [Duration trigger](#)
- [Platform trigger](#)
- [Sample count trigger](#)

Trigger condition A formula that specifies the condition of a [trigger](#) mathematically.

Triggered measurement The measurement of a [measurement raster](#) started by a [platform trigger](#). The data flow between the dSPACE real-time hardware or VEOS and the host PC is not continuous.

U

Unassigned data set A data set that is assigned neither to the working page nor to the reference page of a platform/device. An unassigned data set can be defined as the new working or reference data set. It then replaces the "old" working or reference data set and is written to the corresponding memory page, if one is available on the platform/device.


Unplugged A platform/device state defined by the following characteristics:

- The logical connection between ControlDesk and the hardware was interrupted, for example, because the ignition was turned off or the ControlDesk PC and the hardware were disconnected.
- Before the state of a platform/device changes to 'unplugged', the platform/device was in one of the following states:
 - 'Connected'
 - 'Online calibration started'
 - 'Measuring' / 'Recording'


Tip



A device for which the connection between ControlDesk and the device hardware currently is interrupted is also set to the 'unplugged' state when you start online calibration if both the following conditions are fulfilled:

- The device's **Start unplugged** property is enabled.
- The **Start online calibration** behavior property is set to 'Ignore differences'.

This is possible for CCP and XCP devices. For details on the two properties listed above, refer to [General Settings Properties \(ControlDesk Platform Management\)](#) .

- If the Automatic Reconnect feature is enabled for a platform/device and if the platform/device is in the 'unplugged' state, ControlDesk periodically tries to re-establish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.



The 'unplugged' platform/device state is indicated by the  icon.

Untriggered measurement The measurement of a [measurement raster](#)  not started by a [platform trigger](#) . The data flow between the dSPACE real-time hardware or VEOS and the host PC is continuous.

User function An external function or program that is added to the ControlDesk user interface for quick and easy access during work with ControlDesk.

User Functions Output A [controlbar](#)  that provides access to the output of external tools added to the Automation ribbon.





V


Value block A [parameter](#)  that consists of a 1- or 2-dimensional array of scalar [parameters](#) .

In variable lists, ControlDesk displays entries for the value block itself and for each array element.

Value blocks are represented by the  symbol.








Value conversion The conversion of the original *source values* of variables of an application running on an ECU or dSPACE real-time hardware into the corresponding scaled *converted values*.

Variable Any [parameter](#)  or [measurement variable](#)  defined in a [variable description](#) . ControlDesk provides various [instrument](#) s to visualize variables.

Variable alias An alias name that lets the user control the property of a [segment](#)  by a model parameter of a real-time application.

Variable Array An instrument for calibrating parameters and displaying measurement variable values.

The Variable Array can be used for the following variable types:

- [Measurement](#) 
- [Measurement array](#) 
- [String](#) 
- [Struct](#) 
- [Struct array](#) 
- [Value](#) 
- [Value block](#) 

Variable connection The connection of a [variable](#) to an [instrument](#). Via the variable connection, data is exchanged between a variable and the instrument used to measure or calibrate the variable. In other words, variable connections are required to visualize variables in instrument.

Variable description A file describing the variables in a simulation application, which are available for measurement, calibration, and stimulation.

Variable Editor A tool for viewing, editing, and creating variable descriptions in the ASAM MCD-2MC (A2L) file format. The Variable Editor allows you to create A2L files from scratch, or to import existing A2L files for modification.

Variable Filter A variable filter contains the filter configuration of a combined filter, which is used to filter the variable list in the Variables controlbar using a combination of filter conditions.

Variables controlbar A [controlbar](#) that provides access to the variables of the currently open experiment.

V-ECU Abbreviation of *virtual ECU*.

ECU software that can be executed in a [software-in-the-loop \(SIL\) testing](#) environment such as a local PC or highly parallel in the cloud independently of real-time constraints and real ECU hardware.

Vehicle information The [ODX database](#) can contain information for one or more vehicles. Vehicle information data is used for vehicle identification purposes and for access to vehicles. It references the access paths (logical links) to the ECUs.

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.

VEOS platform A platform that configures and controls the [offline simulation application \(OSA\)](#) running in [VEOS](#) and that also provides access to the application's [environment VPU](#).

VEOS Player An application running on the host PC for editing, configuring and controlling an [offline simulation application \(OSA\)](#) running in VEOS.

Verbal conversion A [conversion](#) in which a [conversion table](#) is used to specify the computation of numerical values into strings. The verbal conversion table is used when you switch the value representation from source to converted mode and vice versa.

Verbal conversion range A [conversion](#) in which a [conversion table](#) is used to specify the computation of a range of numerical values into strings. The verbal conversion range table is used when you switch the value representation from source to converted mode and vice versa.

View set A named configuration of the [controlbar](#)s of ControlDesk. A view set has a default state and a current state that can differ from the default state. The configuration includes the geometry, visibility, and docking or floating state of controlbars.

Visualization The representation of [variable](#)s in [instrument](#)s:

- [Measurement variable](#)s are visualized in instruments to view and analyze their time traces.
- [Calibration parameters](#) are visualized in instruments to change their values.

VPU Abbreviation of *virtual processing unit*. A VPU is part of an offline simulation application in VEOS. Each VPU runs in a separate process of the PC. VPU is also the generic term for:

- V-ECUs
- Environment VPUs
- Controller VPUs
- Bus VPUs

W

Working data set The data set currently residing in the memory of a platform/device hardware. There can be only one working data set for each calibration platform/device. The working data set is read/write.

Working page Memory area containing the parameters of an ECU or prototyping hardware ([memory page](#)). The working page contains the read/write working [data set](#).

If the platform/device also provides a [reference page](#), ControlDesk lets you switch between both pages.

Writable measurement A scalar variable that can be measured and calibrated.

XCP Abbreviation of *Universal Measurement and Calibration Protocol*. A protocol that is implemented on electronic control units (ECUs) and provides access to ECUs with measurement and calibration systems (MCS) such as ControlDesk.

XCP is based on the *master-slave principle*:

- The ECU is the slave.
- The measurement and calibration system is the master.

The “X” stands for the physical layers for communication between the ECU and the MCS, such as CAN (Controller Area Network) and Ethernet.

The basic features of XCP are:

- ECU parameter calibration (CAL)
- Synchronous data acquisition (DAQ)
- Synchronous data stimulation (STIM), i.e., for bypassing
- ECU flash programming (PGM)

The XCP protocol was developed by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.). For the protocol specification, refer to <http://www.asam.net>.

The following ControlDesk devices support ECUs with an integrated XCP service:

- [XCP on CAN device](#)
- [XCP on Ethernet device](#)

XCP on CAN device A device that provides access to an ECU with XCP connected to the ControlDesk PC via CAN. Using the XCP on CAN device, you can access the ECU for measurement and calibration purposes via XCP (*Universal Measurement and Calibration Protocol*).

XCP on Ethernet device A device that provides access to an ECU or [V-ECU](#) with XCP connected to the ControlDesk PC via Ethernet. The XCP on Ethernet device provides access to the ECU/V-ECU via XCP (*Universal Measurement and Calibration Protocol*) for measurement and calibration purposes.

XIL API EESPort [Electrical Error Simulation port \(EESPort\)](#)

XIL API MAPort platform A platform that provides access to a simulation platform via the ASAM XIL API implementation that is installed on your host PC.

XY Plotter A [plotter instrument](#) for displaying signals as functions of other signals.

C

Common Program Data folder 6, 41

D

Documents folder 6, 45

I

Instrument Selector 52

L

Local Program Data folder 6, 53

M

Measurement Data Pool 55

Messages controlbar 56

P

Platforms/Devices controlbar 60

Project controlbar 61

Project Manager 61

Properties controlbar 61

