

ControlDesk

# Layouting

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:	<a href="mailto:info@dspace.de">info@dspace.de</a>
Web:	<a href="http://www.dspace.com">http://www.dspace.com</a>

## How to Contact dSPACE Support

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

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

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

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

## Software Updates and Patches

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

## Important Notice

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

© 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	7
Basics and Instructions	9
Introduction to Layout Handling	10
Basics of Handling Layouts	10
How to Create a New Layout	12
How to Configure the Properties of a Layout	12
How to Export/Import Layouts	15
How to Create a Layout Pool	17
How to Export/Import Variable Connections	21
Placing Variables on a Layout	24
Basics of Placing Variables on a Layout	24
How to Place Variables on a Layout via the Variables Controlbar	29
How to Place Variables on a Layout via the Instrument Selector	32
How to Customize the Connection Assignment of Variables to Instruments	36
How to Copy Variables Between Instruments	41
Arranging Instruments on a Layout	44
Selecting and Positioning Instruments on a Layout	44
How to Optimize the Instrument Arrangement	46
Reference Information	49
Activate (Layout)	53
Align Bottom	53
Align Horizontal Center	54
Align Left	55
Align Right	55
Align Top	56
Align Vertical Center	57
Assign Platforms	57
Autoselect Variable	59
Bring to Front	60
Center Horizontal	61
Center Vertical	61
Close (Layout)	62

Collapse All.....	62
Create Copy.....	63
Create Python Reference.....	64
Delete Invalid.....	65
Expand All.....	65
Export Active Layout Connections.....	66
Export Connections.....	66
Export Layout / Export.....	67
Go to Instrument.....	68
Group.....	69
Import Connections.....	70
Import Layout/Import.....	70
Insert Layout.....	72
Instrument Navigator.....	73
Layout Navigator.....	76
Layout Properties.....	80
Locked Mode.....	82
Make Same Height.....	83
Make Same Size.....	84
Make Same Width.....	84
Move Down.....	85
Move Up.....	86
Open (Layout).....	86
Paste / Paste Instruments with Connections.....	87
Paste Instruments Without Connections.....	88
Remove (Layout).....	88
Rename (Layout).....	89
Scroll to Instrument.....	90
Send to Back.....	90
Show Disconnected.....	91
Space Across.....	92
Space Down.....	92
Switch Data Source.....	93
Synchronize (Layout).....	94
Ungroup.....	95
Visualization Page.....	95
Visualize Variables.....	97

Automation	99
Programming ControlDesk Automation.....	100
Automating Visualization of Variables on Layouts.....	100
Automating Parameter Calibration.....	107
Layout and Instrument Handling.....	110
Layout- and Instrument-Related Interfaces.....	110
Limitations	113
Limitations for Layouting.....	113
Glossary	115
Index	153





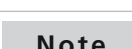





# About This Document

**Content** This document introduces you to ControlDesk's layouting.

## 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.

---

## 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](http://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

<a href="#">Introduction to Layout Handling.....</a>	<a href="#">10</a>
In ControlDesk, variables (parameters and measurement variables) are visualized in instruments that are placed on layouts.	
<a href="#">Placing Variables on a Layout.....</a>	<a href="#">24</a>
Variables can be assigned to instruments to calibrate parameters, or to measure and record data.	
<a href="#">Arranging Instruments on a Layout.....</a>	<a href="#">44</a>
You can arrange instruments on a layout according to your requirements.	

# Introduction to Layout Handling

## Where to go from here

## Information in this section

<a href="#">Basics of Handling Layouts.....</a>	<a href="#">10</a>
Layouts are used for visualizing instruments. You can use the Layout Navigator and the Instrument Navigator to get an overview of the layouts, the instruments and the connected variables.	
<a href="#">How to Create a New Layout.....</a>	<a href="#">12</a>
You can calibrate and measure variables in a layout.	
<a href="#">How to Configure the Properties of a Layout.....</a>	<a href="#">12</a>
You can change, for example, the name, background color and grid properties of a layout.	
<a href="#">How to Export/Import Layouts.....</a>	<a href="#">15</a>
You can export a layout with all its instruments and variable connections, and import it in another experiment.	
<a href="#">How to Create a Layout Pool.....</a>	<a href="#">17</a>
You can store layout files in an external folder and synchronize their settings with the local layout files in your experiment.	
<a href="#">How to Export/Import Variable Connections.....</a>	<a href="#">21</a>
You can export all the variable connections of one or more layouts to a CONX file, and import them to instruments on other layouts.	

## Basics of Handling Layouts

### Introduction

Layouts are used for visualizing instruments. You can use the Layout Navigator and the Instrument Navigator to get an overview of the layouts, the instruments and the connected variables.

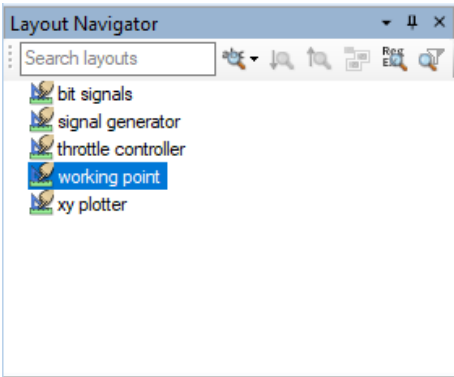
### Layouts and instruments

In a layout, variables are calibrated and measured graphically with instruments. Instruments are used for calibration, measurement, and ECU diagnostics tasks. ControlDesk provides different instruments, such as the Time Plotter for displaying measurement variables and the Table Editor for changing multidimensional parameters.

### Layout Navigator

A controlbar that gives access to all the layouts of the currently active experiment. The Layout Navigator gives you quick access to the context menu commands of each layout. The controlbar can be used for switching between layouts.

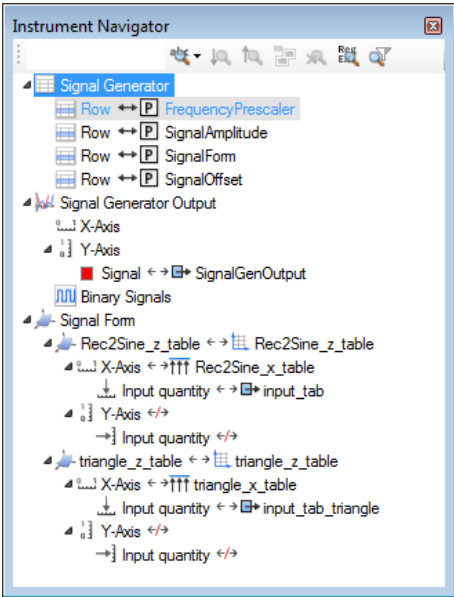
The Layout Navigator looks like this:



Instrument Navigator

A controlbar that gives access to all the instruments and variables that are placed on the currently active layout. The Instrument Navigator's main function is easy selection of instruments in complex layouts.

The Instrument Navigator looks like this:



Extending the functionality of layouts and instruments

You can extend the functionality of a layout or an instrument by adding Python code to it. Refer to [Script Property \(ControlDesk Instrument Handling\)](#).

Related topics

References

<a href="#">Instrument Navigator</a>	73
<a href="#">Layout Navigator</a>	76

## How to Create a New Layout

**Objective** You can calibrate and measure variables in a layout.

**Method** **To create a new empty layout**  
**1** On the Layouting ribbon, click Layouts – Insert Layout.  
 A new empty layout opens.

**Result** You created a new empty layout.

### Tip

If you want to save a single layout without saving the whole project/experiment, select the layout and press **Ctrl+S**.

**Next steps**

- You can configure the layout properties. Refer to [How to Configure the Properties of a Layout](#) on page 12.
- You can place variables or instruments on layouts. Refer to [How to Place Variables on a Layout via the Variables Controlbar](#) on page 29 and [How to Place Variables on a Layout via the Instrument Selector](#) on page 32.

### Related topics

#### HowTos

[How to Export/Import Layouts.....](#) 15

#### References

[Insert Layout.....](#) 72

## How to Configure the Properties of a Layout

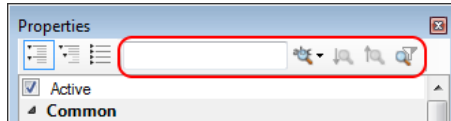
**Objective** You can change, for example, the name, background color and grid properties of a layout.

**Layout properties** **Properties controlbar** You can configure the properties of layouts via the Properties controlbar, which displays the properties of the currently selected layout hierarchically.

**Collapsing and expanding the properties tree** The Properties controlbar provides buttons to expand or collapse categories and properties.



**Searching and filtering** The Properties controlbar provides buttons to search and filter the content. The results are highlighted and you can jump to the previous or next occurrence of the search string.



**Extending the functionality of a layout** You can extend the functionality of a layout by adding Python code to it. Refer to [Layout Properties](#) on page 80.

---

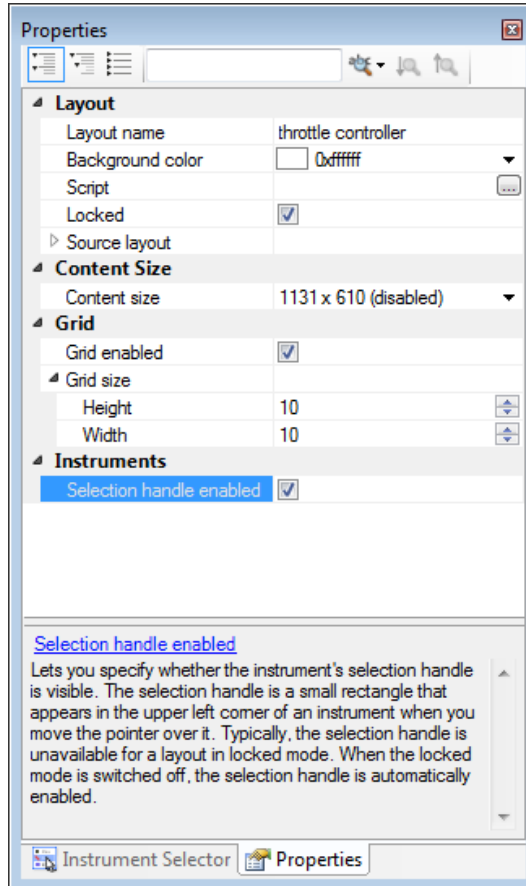
**Preconditions**

At least one layout is opened.

## Method

### To configure the properties of a layout

- 1 From the context menu of a layout, select Properties.  
The Properties controlbar opens.



- 2 Configure the properties of the layout.

## Result

You configured the layout's properties.

## Related topics

### References

Layout Properties..... 80  
Properties (Controlbar) (ControlDesk User Interface Handling )

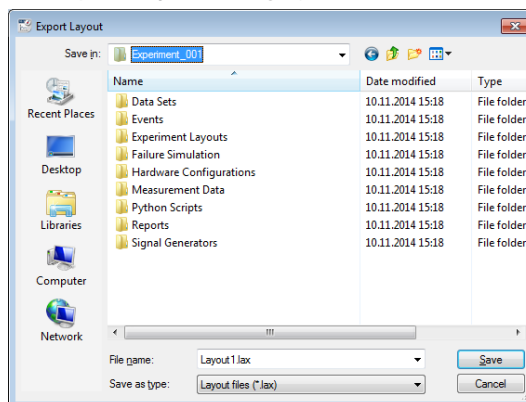
## How to Export/Import Layouts

<b>Objective</b>	You can export a layout with all its instruments and variable connections, and import it in another experiment.
<b>Layout file formats</b>	<p><b>LAY file format</b> Layouts in an experiment have the LAY file format. You cannot use LAY files in other experiments since they do not contain variable connection information.</p> <p><b>LAX file format</b> To use a layout (including variable connection information) in another experiment, you can export the layout to the LAX file format.</p>
<b>Referenced layout files</b>	<p>You can import a layout as a local working copy with a reference to the external layout file (LAX file). The settings of the imported layout can be synchronized with the settings of the external LAX file.</p> <p>This allows you to create external layout pools with layout files that can be referenced from various projects and experiments. Refer to <a href="#">How to Create a Layout Pool</a> on page 17.</p>
<b>Restriction</b>	Exported LAX files can be imported to the same or to a newer ControlDesk version only.

### Part 1

#### To export a layout

- 1 On the Layouting ribbon, click Layouts – Export Layout.  
The Export Layout dialog opens.



- 2 Navigate to the destination folder.
- 3 Enter a new file name or select an existing file from the folder.  
If you select an existing layout file, ControlDesk overwrites it.
- 4 Click Save to close the dialog.

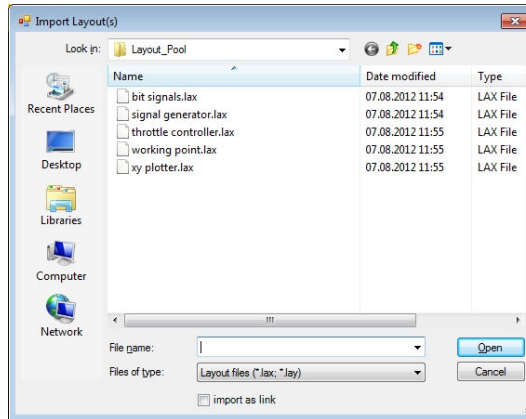
## Interim result

You exported a layout. Continue with [Part 2](#) on page 16.

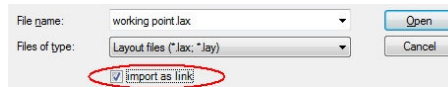
## Part 2

### To import a layout

- 1 Open the experiment to which you want to import the layout.
- 2 On the Layouting ribbon, select Layouts – Import Layout.  
The Import Layout dialog opens.



- 3 Navigate to the layout file which you exported previously and select it.
- 4 If you want to create a layout that references the selected layout file and can be synchronized with it, select Import as link.



- 5 Click Open.  
The imported layout opens.

## Result

You exported a layout from an experiment and imported it to another experiment.

ControlDesk copies the selected layout and establishes all the possible variable connections.

### Note

ControlDesk tries to establish all the variable connections of the imported layout. Variables that cannot be connected in the new experiment are marked as unavailable in the layout.



Tip

You can reuse layouts with instruments that have connections to variables of an SDF file (platform) in an ECU calibration project (calibration device with an A2L file) and vice versa. The precondition is that the variable names used in the SDF file are the same as those used in the A2L file. In addition, the SDF file parameters must be located in the Tunable Parameters node and the SDF file measurement variables must be located in the Labels node.

Related topics

References

Export Layout / Export.....	67
Import Layout/Import.....	70

# How to Create a Layout Pool

**Objective** You can store layout files in an external folder and synchronize their settings with the local layout files in your experiment.

**Referenced layout files** You can import a layout as a local working copy with a reference to an external layout (LAX file). The settings of the imported layout can be synchronized with the settings of the external layout.

This allows you to create external layout pools. You can reference the layouts in the layout pools from various projects and experiments.

**Method** **To create a layout pool**

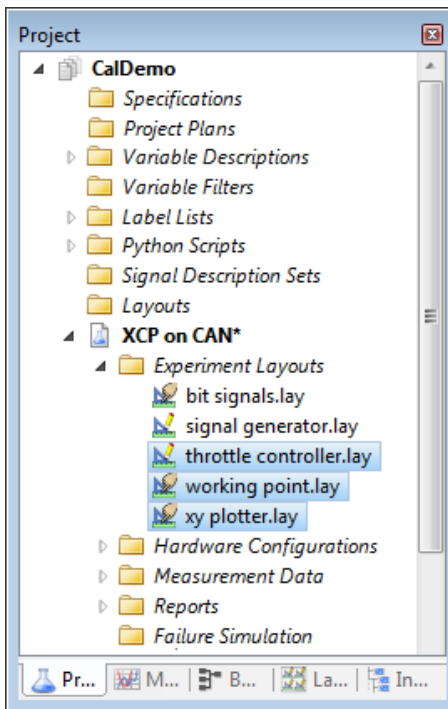
1 In the file system, select or create a folder that you want to use as layout pool folder and name it **LayoutPool1**, for example.

Note

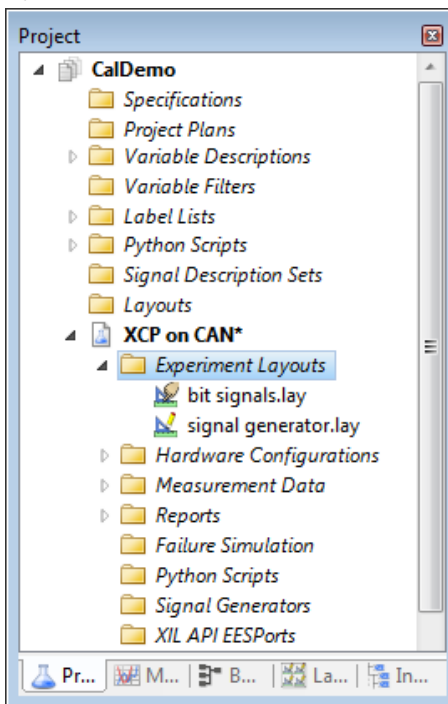
Make sure to select a local or a network folder that is available each time you want to synchronize the local and the referenced layout files.

2 Open an experiment that contains one or more layouts you want to store in the layout pool.

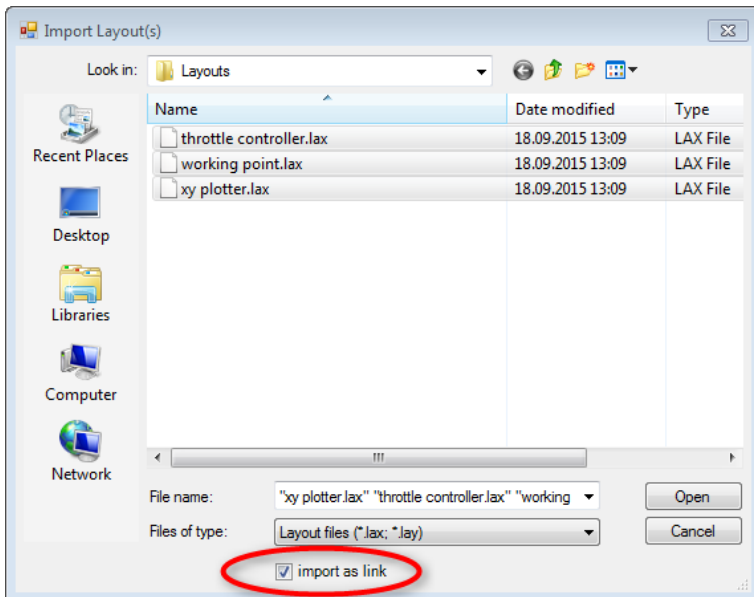
- 3 In the **Project** controlbar, select the layout(s) and export them to the layout pool folder via the context menu.



- 4 In the Project controlbar, remove the exported layouts from the experiment layouts.



- 5 In the context menu of the Experiment Layouts folder, select Import to open the Import Layout(s) dialog and reimport the exported layouts to the experiment.

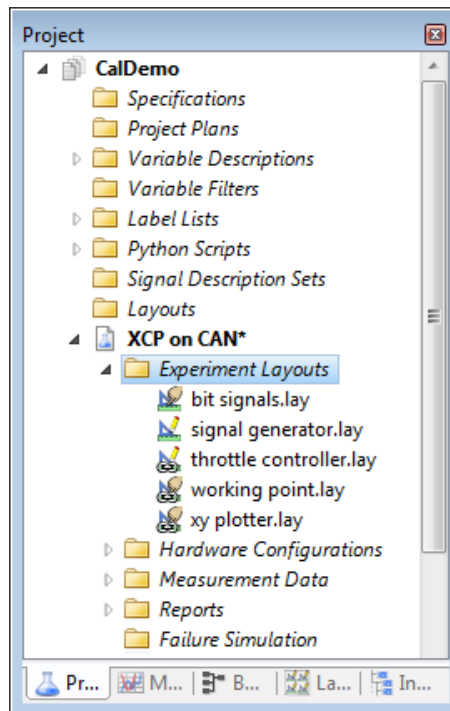
**Note**

Make sure to activate Import as link in the Import Layout(s) dialog.

- 6 Click Open.

**Result**

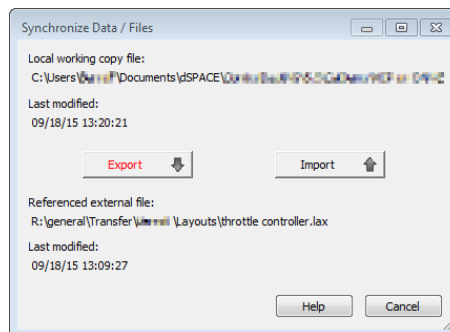
The reimported layouts are added to the experiment layouts as local working copies with references to the related layouts in the layout pool. This is visualized by an additional chain on the layout symbol.



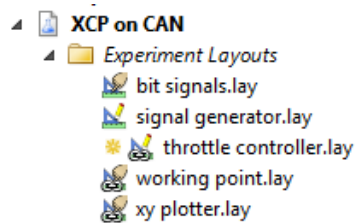
## Synchronizing

If a layout is referenced to a layout in the layout pool, its context menu offers a Synchronize command. After you have changed the layout, you can synchronize the layout settings with the settings of the referenced layout in the layout pool.

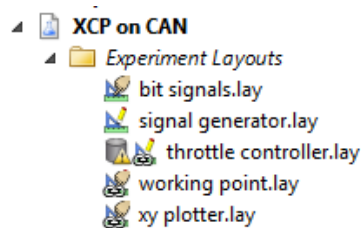
- Select Export to change the layout file in the layout pool.
- Select Import to discard the changes in the local layout and overwrite them with the settings of the reference in the layout pool.



If you store an experiment without performing this synchronization step, the unsynchronized layout is marked by a star.



If you open a new experiment and a local layout is not synchronized with the referenced layout, ControlDesk prompts you for synchronizing the layouts. If you reject synchronizing, the unsynchronized layout is marked by a data base symbol with an exclamation mark.



Related topics

HowTos	
How to Export/Import Layouts.....	15
References	
Export Layout / Export.....	67
Import Layout/Import.....	70
Synchronize (Layout).....	94

# How to Export/Import Variable Connections

**Objective** You can export all the variable connections of one or more layouts to a CONX file, and import them to instruments on other layouts.

**Connection files** ControlDesk can export the variable connections of the instruments on the active layout or on all the opened layouts to an XML connection file in CONX file format.

CONX files can be imported to an active experiment. If the experiment contains layouts and instruments that have the same types and names as those stored in

the connection file, the stored variable connections are reestablished. Instruments that are not affected by this keep their current connections.

To use connection files created with ControlDesk 3.x, you can also import connection files in CON file format.

## Export and import options

You can specify general options for the export and import of variable connections. Refer to [Visualization Page](#) on page 95.

## Preconditions

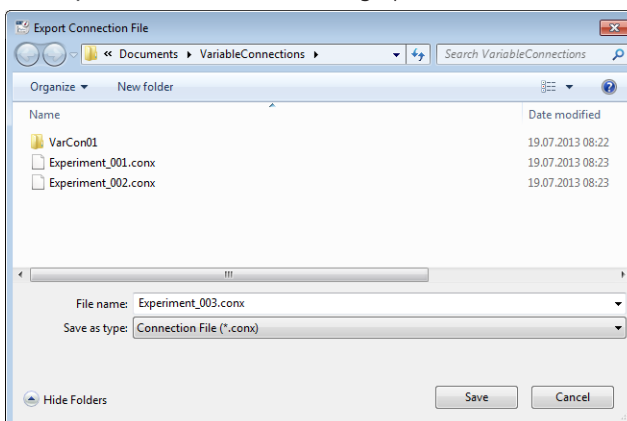
To import variable connections from a connection file, the target experiment must have an appropriate platform/device and an appropriate variable description for the exported connections.

## Part 1

### To export variable connections of all opened layouts

- 1 On the Layouting ribbon, click Connections – Export Connections – Export Connections.

The Export Connection File dialog opens.



- 2 Navigate to the destination folder.
- 3 Enter a new file name or select an existing file from the folder.
- 4 Click Save to close the dialog.

## Interim result

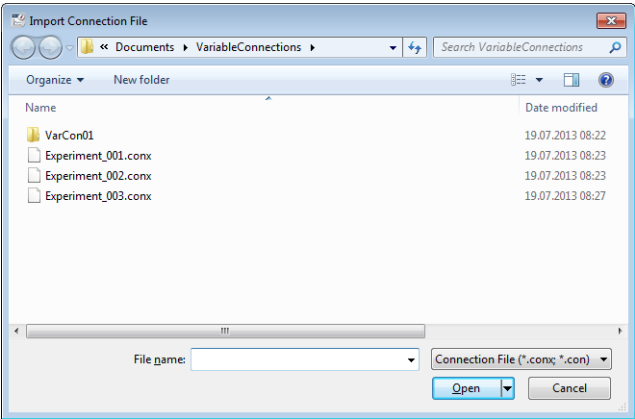
You exported the connections of all the opened layouts. Continue with [Part 2](#) on page 22.

## Part 2

### To import variable connections

- 1 Open the experiment to which you want to import the variable connections.
- 2 Open all the layouts you want to import connections to.
- 3 On the Layouting ribbon, click Connections – Import Connections.

The Import Connection File dialog opens.



- 4 Navigate to the connection file which you exported previously and select it.
- 5 Click Open.  
The imported connections are reestablished for all matching layouts and instruments.

Result

You exported the connections of all the opened layouts of one experiment and imported them to the layouts of another experiment.  
ControlDesk establishes all the possible variable connections.

Related topics

References

Export Active Layout Connections.....

Export Connections.....

Import Connections.....

Visualization Page.....

66

66

70

95

# Placing Variables on a Layout

## Where to go from here

## Information in this section

### [Basics of Placing Variables on a Layout.....24](#)

Variables can be assigned to instruments to calibrate parameters, or to measure and record data.

### [How to Place Variables on a Layout via the Variables Controlbar.....29](#)

You can use the Variables controlbar to place variables in instruments on a layout.

### [How to Place Variables on a Layout via the Instrument Selector.....32](#)

You can use the Instrument Selector to place instruments on a layout and connect variables to them.

### [How to Customize the Connection Assignment of Variables to Instruments.....36](#)

Customizing the connection assignment allows you to create special variable-instrument connections, such as connecting a group of variables to multiple instruments in one step, for example.

### [How to Copy Variables Between Instruments.....41](#)

You can copy variables between instruments or within an instrument which can display multiple variables.

## Basics of Placing Variables on a Layout

### Introduction

Variables can be assigned to instruments to calibrate parameters, or to measure and record data.

### Layouts and instruments

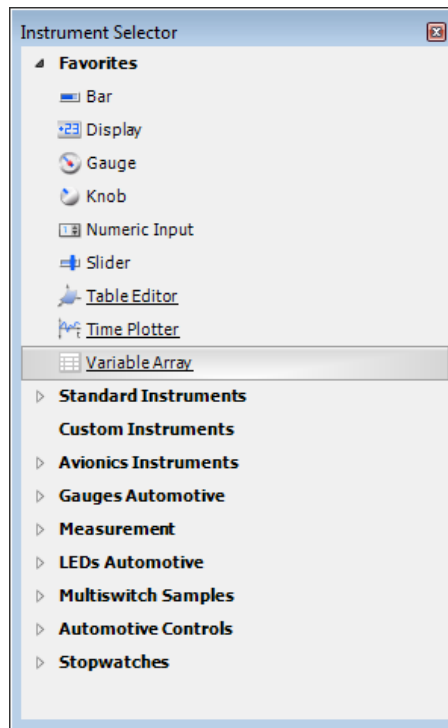
Layouts are used for visualizing instruments. Variables can be assigned to instruments to calibrate parameters, or to measure and record data.

In the layout, variables are calibrated and measured graphically with instruments. Instruments are used for calibration, measurement, and ECU diagnostics tasks. ControlDesk provides different instruments, such as the Time Plotter for displaying measurement variables and the Table Editor for changing multidimensional parameters.

### Instrument Selector












The Instrument Selector is a controlbar that provides access to ControlDesk's instruments. You can select instruments from the Instrument Selector and place them on layouts.
















### Instruments and the variable types they can visualize

The table below lists all the ControlDesk instruments, and shows which variable type an instrument can visualize.

Instrument	Variable Types										
	 1)	 2)	 3)	 4)	 5)	 6)	 7)	 8)	 9)	 10)	 11)
3-D Viewer <sup>12)</sup>	✓	✓	✓	–	–	–	✓	–	✓	–	–
Airspeed Indicator	✓	–	✓	–	–	–	–	–	✓	–	–
Altimeter	✓	–	✓	–	–	–	–	–	✓	–	–
Animated Needle	✓	–	✓	–	–	–	–	–	✓	–	–
Artificial Horizon	✓	–	✓	–	–	–	–	–	✓	–	–
Bar	✓	–	✓	–	–	–	–	–	✓	–	–
Check Button	✓	–	✓	–	–	–	–	–	✓	–	–
Diagnostics Instrument	–	–	–	–	–	–	–	–	–	–	–
Display	✓	–	✓	–	–	–	–	✓	✓	–	–
Fault Memory Instrument	–	–	–	–	–	–	–	–	–	–	–
Frame	✓	–	✓	–	–	–	–	✓	✓	–	–
Gauge	✓	–	✓	–	–	–	–	–	✓	–	–
Heading Indicator	✓	–	✓	–	–	–	–	–	✓	–	–
Index Plotter	✓	–	✓	–	–	–	–	–	✓	–	–

Instrument	Variable Types										
	 1)	 2)	 3)	 4)	 5)	 6)	 7)	 8)	 9)	 10)	 11)
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 (single connection)	–	✓	–	✓	✓	✓	✓	–	–	–	–
Table Editor (multiple connections)	✓	✓	✓	✓	–	–	✓	–	✓	–	–
Time Plotter	✓	–	✓	–	–	–	–	–	✓	–	–
Variable Array	✓	✓	✓	–	–	–	✓	✓	✓	(✓) <sup>13)</sup>	(✓) <sup>13)</sup>
XY Plotter	✓	✓ <sup>14)</sup>	✓	–	–	–	–	–	✓	–	–

1) [Measurement](#) ⓘ

2) [Measurement array](#) ⓘ

3) [Value \(Float, Int\)](#) ⓘ

4) [Common axis](#) ⓘ

5) [Curve](#) ⓘ

6) [Map](#) ⓘ

7) [Value block](#) ⓘ

8) [String](#) ⓘ

9) [Calculated variable](#) ⓘ

10) [Struct](#) ⓘ

11) [Struct array](#) ⓘ

12) You can connect variables to properties of the 3-D Viewer as well as properties of items displayed in the 3-D Viewer. For more information, refer to [Overview of Possible Variable Connections \(ControlDesk Instrument Handling\)](#).

13) You can connect the contained variables to different instruments in one step by customizing the connection assignment. Refer to [How to Customize the Connection Assignment of Variables to Instruments](#) on page 36.

14) You can connect one-dimensional measurement arrays to the x- and y-axis if they have the same length.

## Handling conversion tables and multiscalings

Some instruments support variables with conversion tables and multiscalings. The following table shows the most suitable instruments:

Instrument	Conversion Table	Multiscaling
<a href="#">Display</a> ⓘ	✓	✓
<a href="#">Numeric Input</a> ⓘ	✓	✓
<a href="#">Push Button</a> ⓘ	✓	—
<a href="#">Radio Button</a> ⓘ	✓	—
<a href="#">Selection Box</a> ⓘ	✓	—
<a href="#">Variable Array</a> ⓘ	✓	✓

For details, refer to [Basics on Variables Using Conversion Tables \(ControlDesk Variable Management ⓘ\)](#).

## Connecting variables to instruments


You can use various ways to connect variable to instruments.

- Via drag and drop:
  - Drag variables from the Variables controlbar to an empty place on a layout. Refer to [How to Place Variables on a Layout via the Variables Controlbar](#) on page 29.
  - Drag variables from the Variables controlbar to an instrument on a layout. Refer to [How to Place Variables on a Layout via the Instrument Selector](#) on page 32.
  - Drag variables from an instrument to an empty place on a layout. Refer to [How to Copy Variables Between Instruments](#) on page 41.
  - Drag variables from an instrument to another instrument. Refer to [How to Copy Variables Between Instruments](#) on page 41.
- Via the context menu and dialogs:
  - Select variables in the Variables controlbar and select **Visualize Variables** from the context menu. Refer to [How to Place Variables on a Layout via the Variables Controlbar](#) on page 29.
  - Select instruments on a layout and select **Variables – Add** from the context menu. Refer to [Variable\(s\) - Add \(ControlDesk Instrument Handling ⓘ\)](#).
  - Select instruments on a layout and select **Variables – Copy to instrument** from the context menu. Refer to [How to Place Variables on a Layout via the Instrument Selector](#) on page 32.
  - Select instrument or connection nodes in the Instrument Navigator and select **Variable Add** from the context menu. Refer to [Variable\(s\) - Add \(ControlDesk Instrument Handling ⓘ\)](#).


## Preserving variable connections

If the data source changes and the variable of a variable connection is not available any longer, the instrument preserves the variable name.

The following example shows this case after the data source of the layout has been switched to a different measurement data file:


Variable Array_48267: SignalGenOutput_2 [mV]			
-256..255.9921875 Converted Incr. +0.0078125 / 0.078125			
Output of signal generator			
	Variable	Value	Unit
<input type="checkbox"/>	currentTime	17237696.126709	s
<input checked="" type="checkbox"/>	SignalGenOutput_2		mV

Variable Array_48267: rec1_001.idf : XCP/SignalGenOutput_2 [mV]			
-256..255.9921875 Converted Incr. +0.0078125 / 0.078125			
Output of signal generator			
	Variable	Value	Unit
<input type="checkbox"/>	currentTime		
<input checked="" type="checkbox"/>	XCP/SignalGenOutput_2		mV

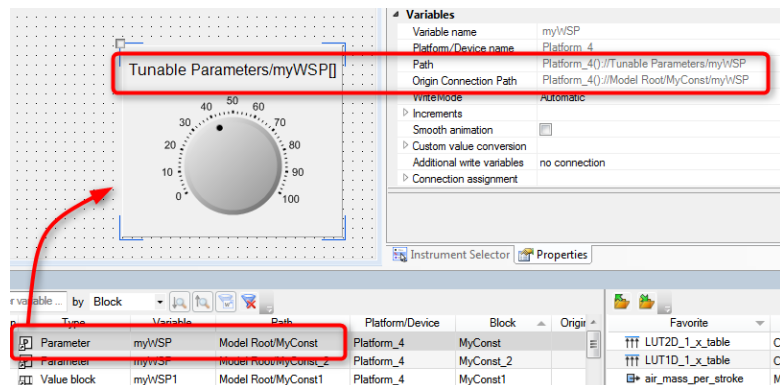
ControlDesk reactivates the variable connection when the data source offers an appropriate variable with the same name and data type again.

## References and origin connection paths

The icons of variables displayed under a subnode in the variable list can have an additional  symbol. The arrow indicates that this variable is a reference to a variable residing in another node of the variable description. For example, variables in an A2L file always reside under the root node.

If you drag a reference to an instrument, the referenced variable is connected, not the reference itself.

The path of the connected variable and the *origin connection path* of the dragged variable are displayed in the instrument properties.



The screenshot shows the ControlDesk interface. A red box highlights a variable reference in the variable list: **Tunable Parameters/myWSP**. A red arrow points from this reference to the **Properties** panel on the right. The **Properties** panel shows the following information:

- Variable name:** myWSP
- Platform/Device name:** Platform\_4
- Path:** Platform\_4/Tunable Parameters/myWSP
- Origin Connection Path:** Platform\_4/Model Root/MyConst/myWSP
- WriteMode:** Automatic
- Increments:** ☐
- Smooth animation:** ☐
- Custom value conversion:** ☐
- Additional write variables:** no connection
- Connection assignment:** ☐

Below the properties panel, the **Instrument Selector** and **Properties** tabs are visible. At the bottom, a table lists the variables in the variable list:

Type	Variable	Path	Platform/Device	Block	Origin
Parameter	myWSP	Model Root/MyConst	Platform_4	MyConst	
Parameter	myWSP	Model Root/MyConst_2	Platform_4	MyConst_2	
Value block	myWSP1	Model Root/MyConst1	Platform_4	MyConst1	

In the Variables controlbar, you can get information on the referenced variable in the following ways:

- You can display the path to the referenced variable by adding the Referenced Variable column to the variable list.

- You can select and highlight the referenced variable via **Navigate to Referenced Variable** in the context menu.

## Related topics

### Basics

[Basics on Variable Types \(ControlDesk Variable Management !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)\)](#)

### HowTos

[How to Place Variables on a Layout via the Variables Controlbar..... 29](#)

### References

[Navigate to Referenced Variable \(ControlDesk Variable Management !\[\]\(0b5e7e25e8775f7e7e80906ada4f0021\_img.jpg\)\)](#)

## How to Place Variables on a Layout via the Variables Controlbar

**Objective** You can use the Variables controlbar to place variables in instruments on a layout.

**Preconditions** Layout editing must be unlocked. Refer to [Locked Mode](#) on page 82.

**Possible methods** You can place variables on a layout via the **Visualize Variables** command or via drag and drop.

- The **Visualize Variables** command allows you to assign variables to their default instruments and place them on a layout in one step. Refer to [Method 1](#) on page 29.
- You can select variables from the Variables controlbar and place them on a layout by dragging them directly. Refer to [Method 2](#) on page 30.

### Method 1

#### To place variables on a layout (using the Visualize Variables command)

- 1 Press **Alt+Shift+3** or click the Variables tab to switch to the Variables controlbar.
- 2 Scroll to the variables using the mouse or the arrow keys.
- 3 Select variables by highlighting them or by activating their checkboxes. If you use checkboxes, the variables you select appear in the **Checked variables** list on the right.

**Tip**

Multiple selection is possible by pressing **Ctrl** or **Shift** when clicking a variable. Press **Ctrl** to highlight a number of single variables with the mouse or **Shift** to highlight a block of variables. Press the **Spacebar** to activate the checkboxes of all highlighted variables.

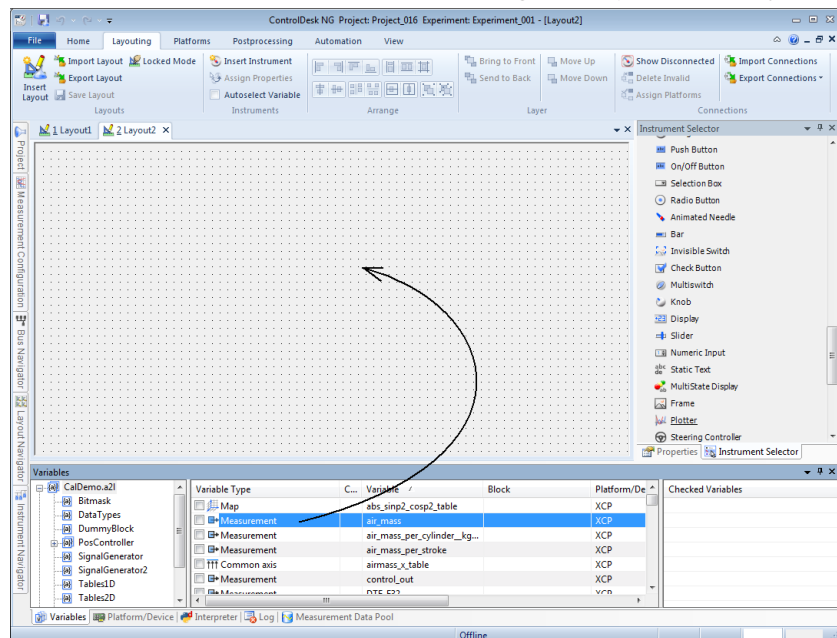
Using the keyboard: Press **Shift** to highlight blocks of variables. Press the **Spacebar** to activate the checkboxes of all highlighted variables

- 4 Select Visualize Variables in the context menu of the variable list, or press **Enter**.

ControlDesk assigns the default instrument(s) to the selected variable(s) and puts the instrument(s) on the active layout.

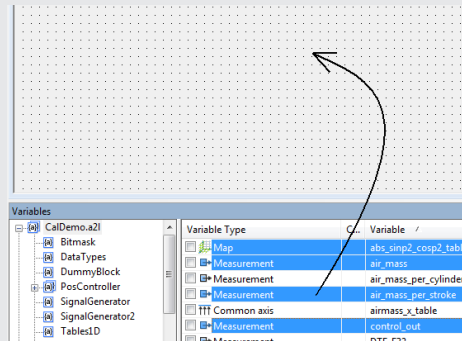
**Method 2****To place variables on a layout (using drag and drop)**

- 1 Press **Alt+Shift+3** or click the Variables tab to switch to the Variables controlbar.
- 2 From the variable list, click a variable, then drag and drop it onto the layout.

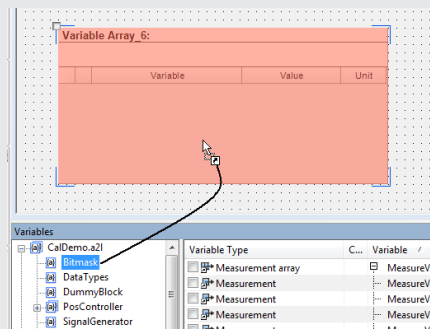


**Tip**

- Multiple selection is possible by pressing **Ctrl** or **Shift** when clicking a variable. Press **Ctrl** to select a number of single variables or **Shift** to select a block of variables.



- If an instrument is able to display all the variables of a subgroup, e.g., all the variables of a block or sub system, you can drag the group node from the variable tree to the instrument.



The result depends on which mouse button you use and where you drop the variable:

- If you drag the variable onto an existing instrument on a layout, the variable is displayed in the instrument.
- If you drag the variable onto an empty space of the layout using the left mouse button, and "Show Drop-List always" is available, a list opens for you to select an instrument to be assigned as above.  
If "Show Drop-List always" is unavailable, ControlDesk adds the default instrument for the connected variable(s) to the layout automatically.
- If you drag the variable onto an empty space on the layout using the right mouse button, a list opens for you to select an instrument to be assigned. Select an instrument from the list.

ControlDesk adds the selected instrument to the layout with the connected variable(s).

#### Note

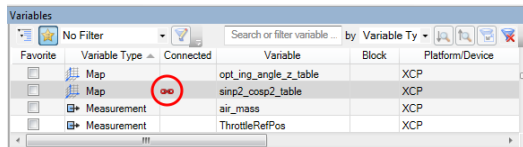
You can place new variables on a layout even during a running measurement, but they will not be measured until the measurement is restarted. ControlDesk therefore asks you if you want to continue the running measurement.

- If you let ControlDesk continue the running measurement, the new variable is not measured (until the next restart of the measurement or recording).
- If you let ControlDesk stop and restart the measurement, the new variable is measured. The data in the measurement data buffer is lost.

## Result

The selected variables are visualized in instruments and placed on one or more layouts. By default, a new instrument is named 'Instrument\_nnn', where 'nnn' stands for a number that will be increased to get a unique instrument name for automatic purposes.

In the Variables controlbar, a red chain symbol in the Connected column indicates that a variable is visualized on a layout.



## Related topics

## References

[Visualize Variables.....](#) 97

# How to Place Variables on a Layout via the Instrument Selector

## Objective

You can use the Instrument Selector to place instruments on a layout and connect variables to them.

## Preconditions

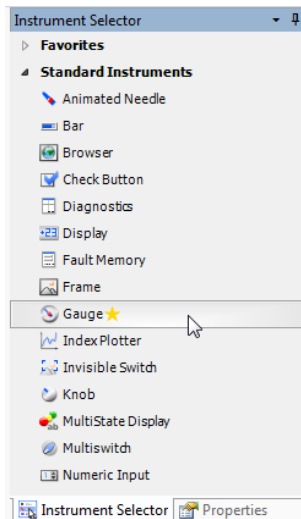
- The Instrument Selector must be opened.
- Layout editing must be unlocked. Refer to [Locked Mode](#) on page 82.



## Method

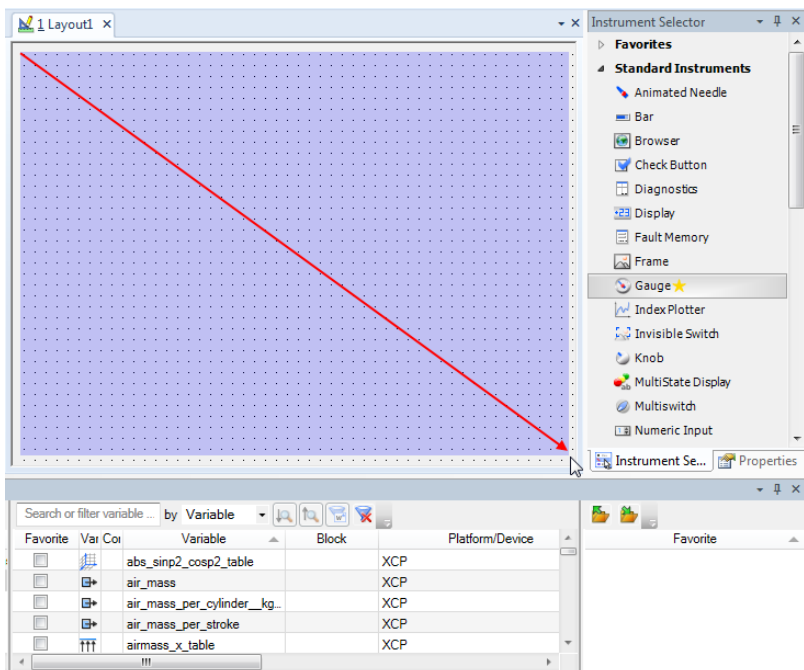
### To place variables on a layout via the Instrument Selector

- 1 Click an instrument in the Instrument Selector.

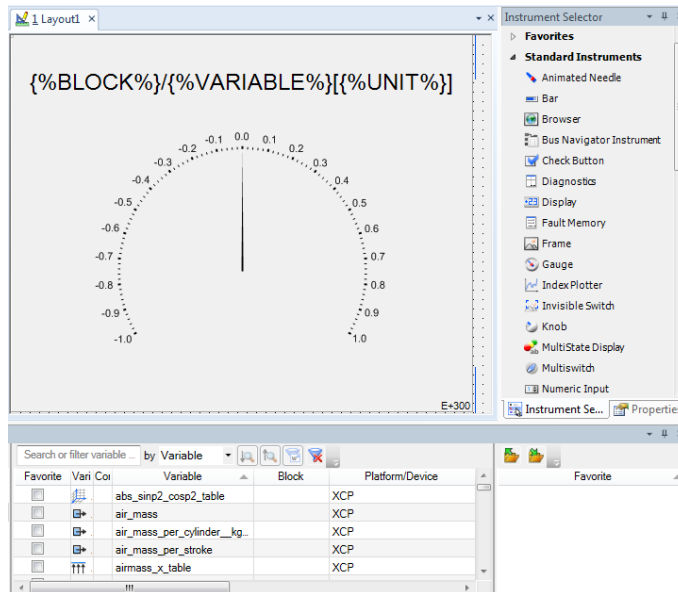


ControlDesk highlights the selected instrument.

- 2 Move the cursor onto the layout.  
On the layout, the cursor turns into a crosshair cursor.
- 3 Click the start position of the instrument on the layout, keep the left mouse button pressed and drag the crosshair cursor to the desired instrument size, then release the left mouse button.



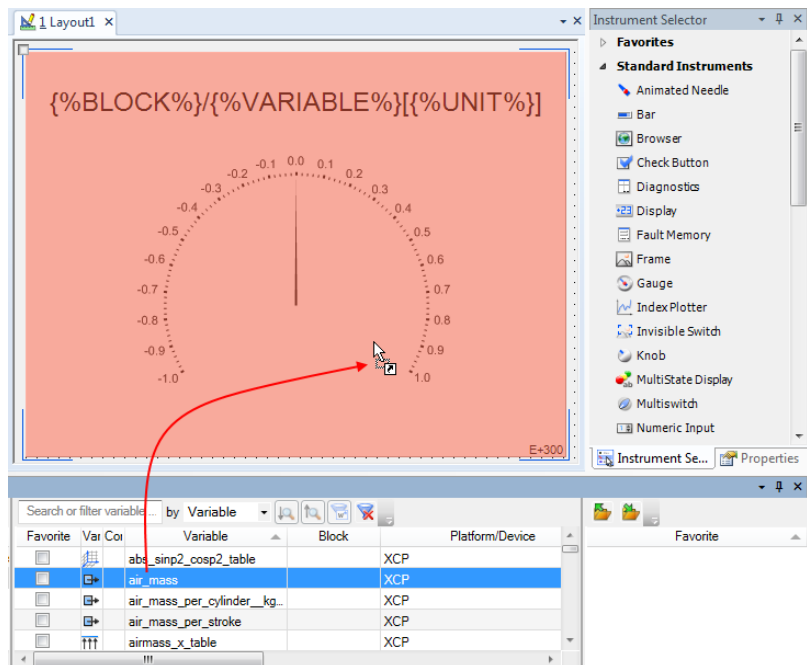
ControlDesk adds the selected instrument to the layout.



### Tip

Alternatively, you can add an instrument with default size by double-clicking the selected instrument in the Instrument Selector or by dragging and dropping it from the Instrument Selector to the layout.

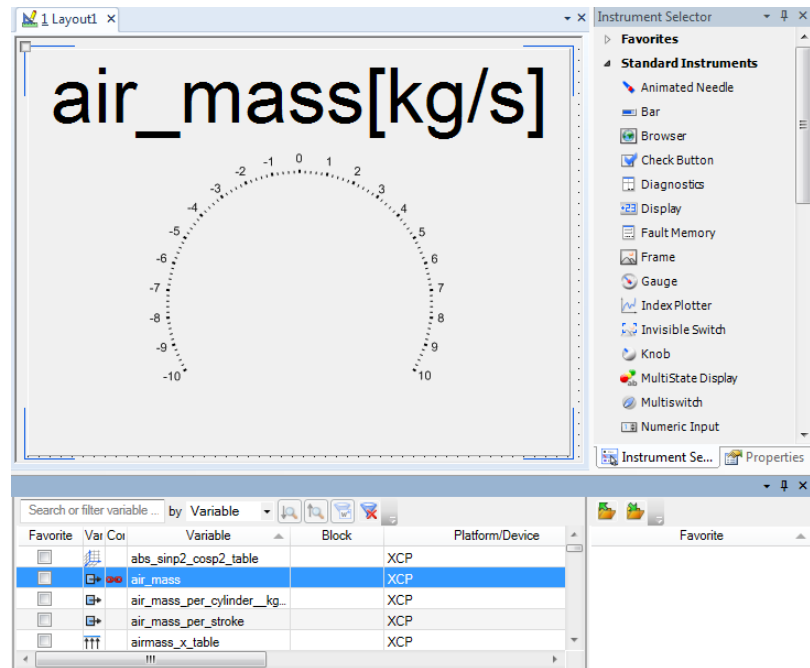
- 4 Click the Variables tab to switch to the Variables controlbar.
- 5 Drag and drop a variable onto the instrument.



**Tip**

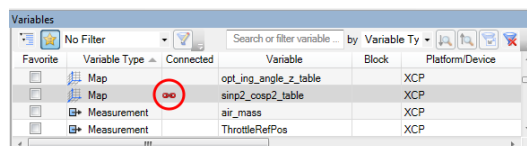
Alternatively, you can open a dialog to select the variables to be connected. Select one or more instruments on the layout or in the Instrument Navigator, open the context menu and select Variables – Add. Refer to [Variable\(s\) - Add \(ControlDesk Instrument Handling\)](#).

The selected variable is displayed in the instrument.

**Result**

You added an instrument and assigned a variable to it. By default, a new instrument is named 'Instrument\_nnn', where 'nnn' stands for a number that will be increased to get a unique instrument name for automatic purposes.

In the Variables controlbar, a red chain symbol in the Connected column indicates that a variable is visualized on a layout.

**Tip**

If you want to replace an existing variable, press the right mouse button when you drag and drop the variable. This opens a context menu that allows you to specify the insert behavior.

## Related topics

### HowTos

[How to Configure the Instrument Selector \(ControlDesk Instrument Handling !\[\]\(dfbd6b3763a6d1d9afaa974f64e2e4b5\_img.jpg\)\)](#)

### References

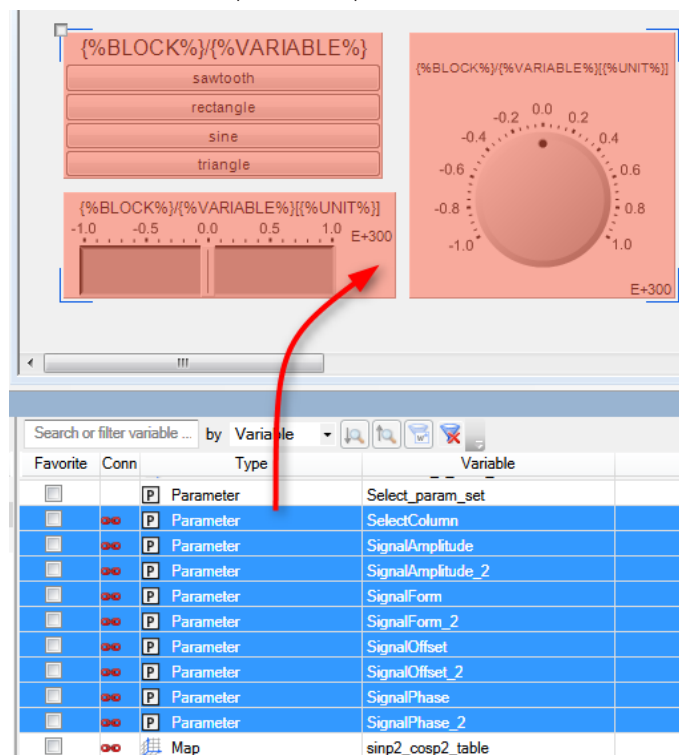
[Instrument Selector \(ControlDesk Instrument Handling !\[\]\(23d9fc146e83b5c3013cfa32c784f8d5\_img.jpg\)\)](#)

Visualize Variables.....97

# How to Customize the Connection Assignment of Variables to Instruments

## Objective

Customizing the connection assignment allows you to create special variable-instrument connections, such as connecting a group of variables to multiple instruments in one step, for example.

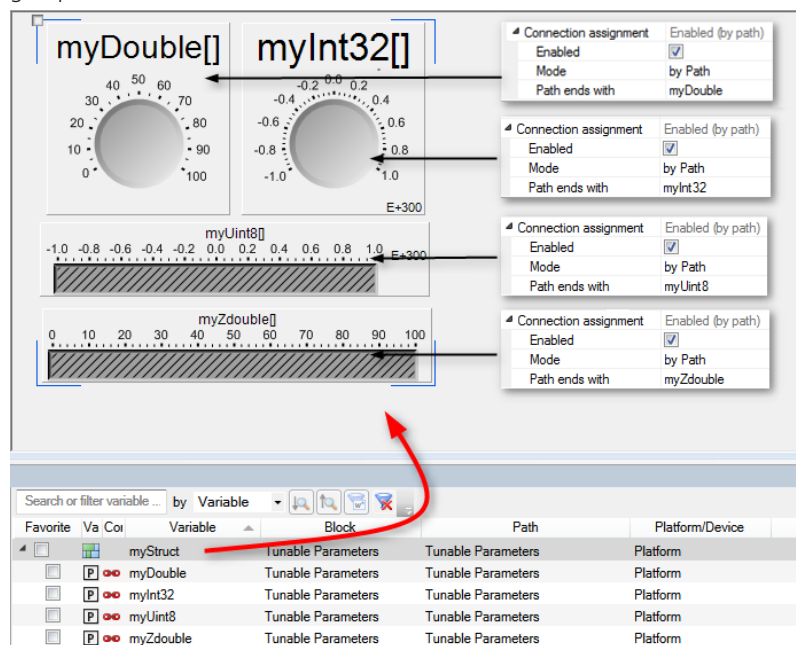


## Basics

In some cases, you might have to create the same variable-instrument connections repeatedly: for example, if you have to connect the variables of a struct to appropriate instruments. In these cases, you can create a specific instrument group that you can connect a variable or a group of variables to in one step. See the following table.

Connection Assignment	Description	Example
Single variable to multiple instruments	To use the value of one variable simultaneously for all the instruments in an instrument group.	Visualizing an rpm value in an LED instrument group. Each LED displays a specific value range to show acceleration and deceleration.
Multiple variables to multiple instruments	To display the values of the variables contained in a group of variables in an instrument group.	Visualizing structs or block groups in an instrument group. Each variable in the struct or block group is connected to the appropriate instrument.

You customize the connection assignment via the Connection assignment properties of the single instruments in the instrument group. You can specify a text string that lets ControlDesk search for a matching variable in a variable group.



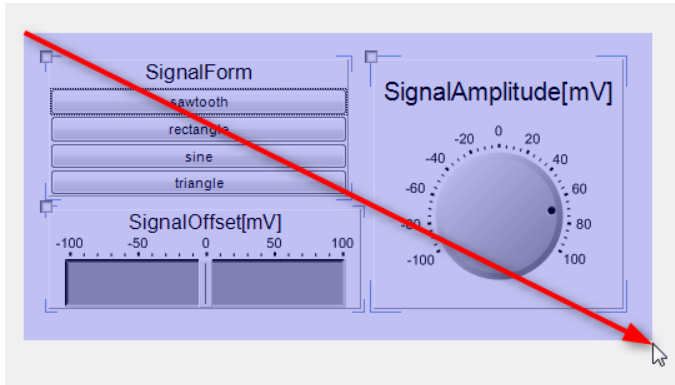
The search refers either to the end of the variable path or to a regular expression. This allows you to use relative paths for the connection of variables to instruments. For details, refer to [Connection Assignment Properties \(ControlDesk Instrument Handling\)](#).

## Method

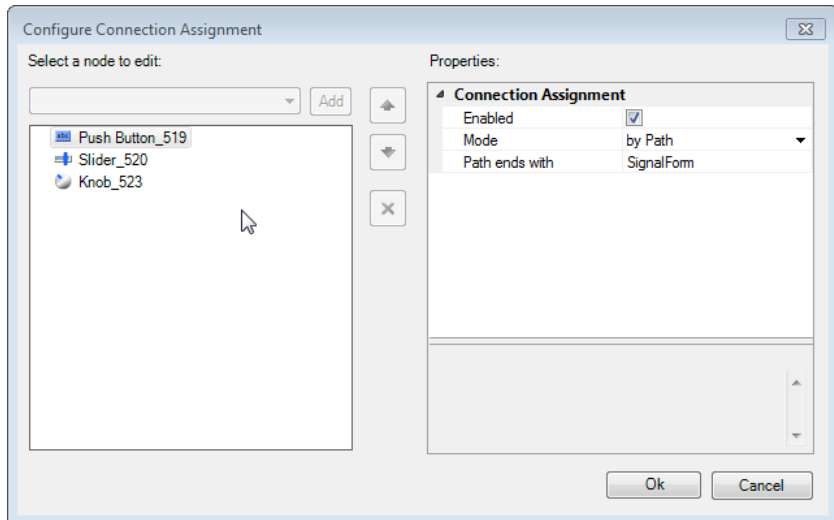
### To customize the connection assignment of variables and instruments

- 1 Place the instruments you want to use for the connections on a layout.
- 2 Drag the variables you want to connect to the appropriate instruments.

- 3 Drag a selection frame around the instruments to select them as a group.



- 4 Right-click the selected instruments and select Variables – Configure Connection Assignment from the context menu.  
The Configure Connection Assignment dialog opens, displaying the Connection Assignment properties of the selected instruments.



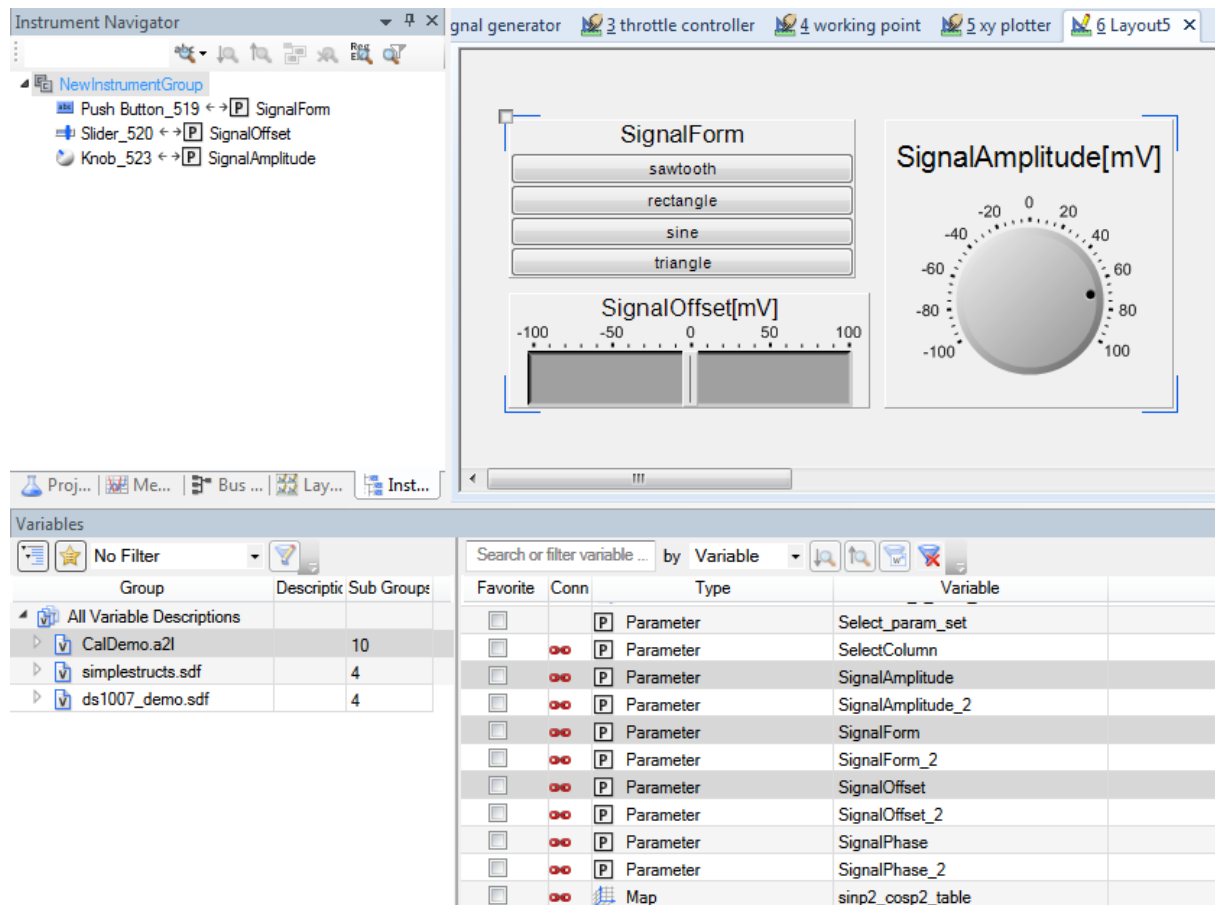
The connection assignment for each instrument is specified according to the current variable connections. Mode is set to by Path and Path ends with to the variable name. You can change the properties according to your needs.

- 5 Click Ok to close the dialog.

- 6 Drag the instrument group to a category in the Instrument Selector to make it a custom instrument that can be reused. ControlDesk opens a dialog to enter a name for the new custom instrument.

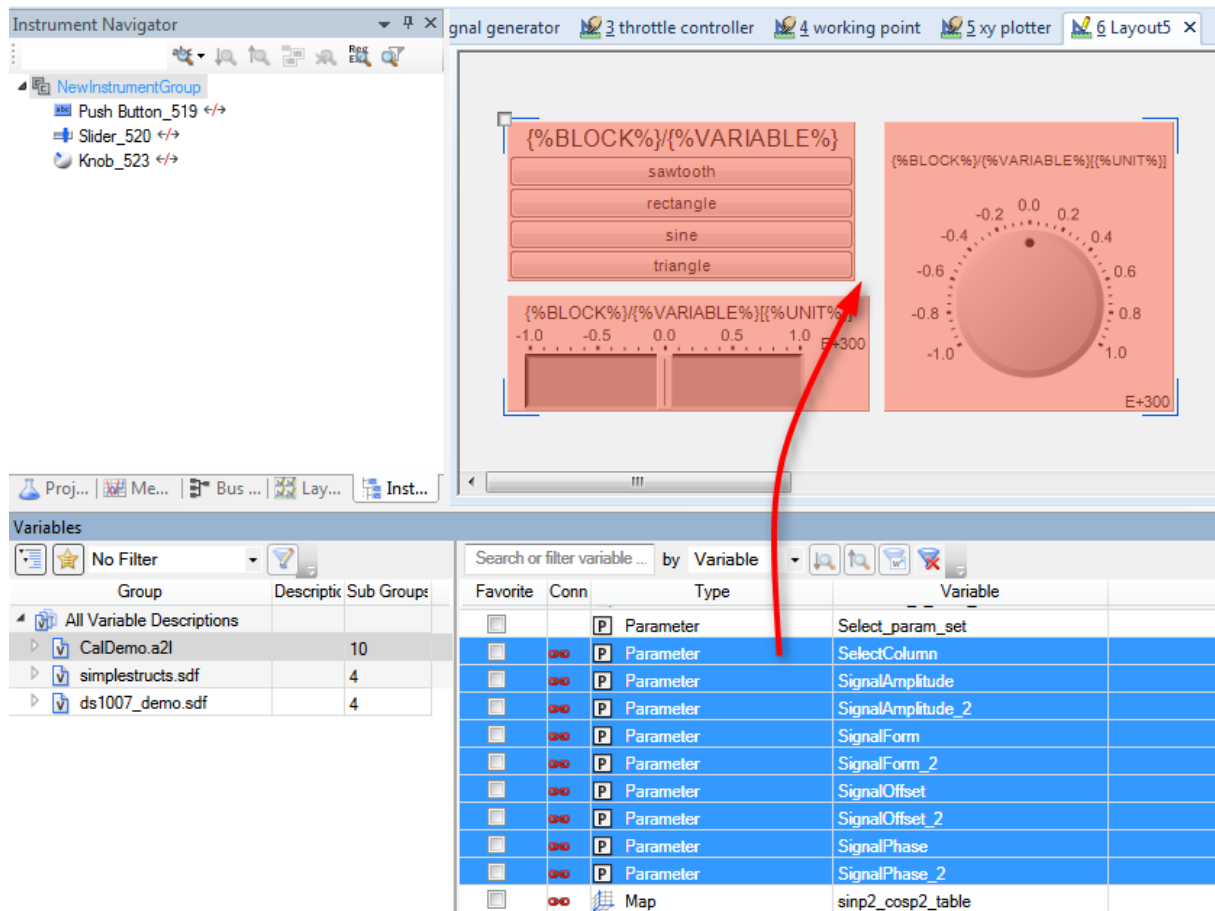
## Result

You have created a new custom instrument consisting of an instrument group with a custom connection assignment for each instrument in the group.



If you now drag a group of variables to the new instrument group, ControlDesk searches for variables that match the connection assignment. If the dragged variable group offers a matching variable for an instrument in the instrument group, it is connected. Non-matching variables are ignored.

Before you drop the variable group, the instruments that can be connected are highlighted.



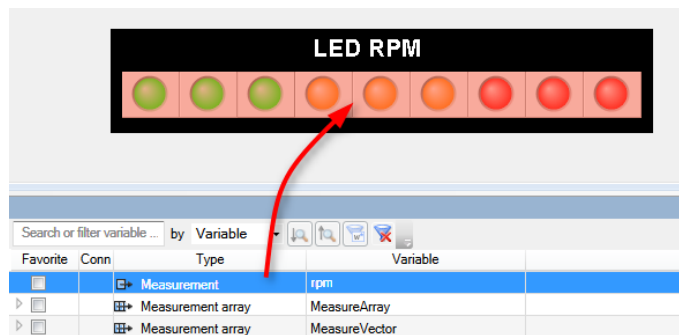
As long as connection assignment is enabled for an instrument group, you can drag variables only to the instrument group as a whole, not to an individual instrument in the instrument group.

### Example of connecting one variable to an instrument group

Suppose you want to create an LED panel that visualizes an rpm value. After you have created an instrument group with the desired number of LEDs and connected all of them to the same rpm variable, you have to specify the value range that each LED must display.



You can now reuse this custom LED instrument. Dragging the rpm variable on it creates all the necessary connections in one step.



### Tip

If you change the connection assignment to the regular expression `$*`, all variables available for the instrument are accepted. You can use this to connect rpm variables independent of their naming. However, the value ranges of the rpm variable must match the LED settings.

If you select Configure Connection Assignment for an instrument without variable connection, ControlDesk creates a connection assignment with the regular expression `$*` automatically.

### Related topics

### References

[Connection Assignment Properties \(ControlDesk Instrument Handling\)](#)

## How to Copy Variables Between Instruments

### Objective

You can copy variables between instruments or within an instrument which can display multiple variables.

### Basics

You can copy variables from an instrument to another instrument or within an instrument such as a Variable Array. This method is the same as assigning variables to instruments via the Variables controlbar. The advantage of copying variables is that you can place them on several instruments quickly without having to search for them in the Variables controlbar.

### Preconditions

- An instrument with the variables to be copied must be created on a layout.
- Layout editing must be unlocked. Refer to [Locked Mode](#) on page 82.

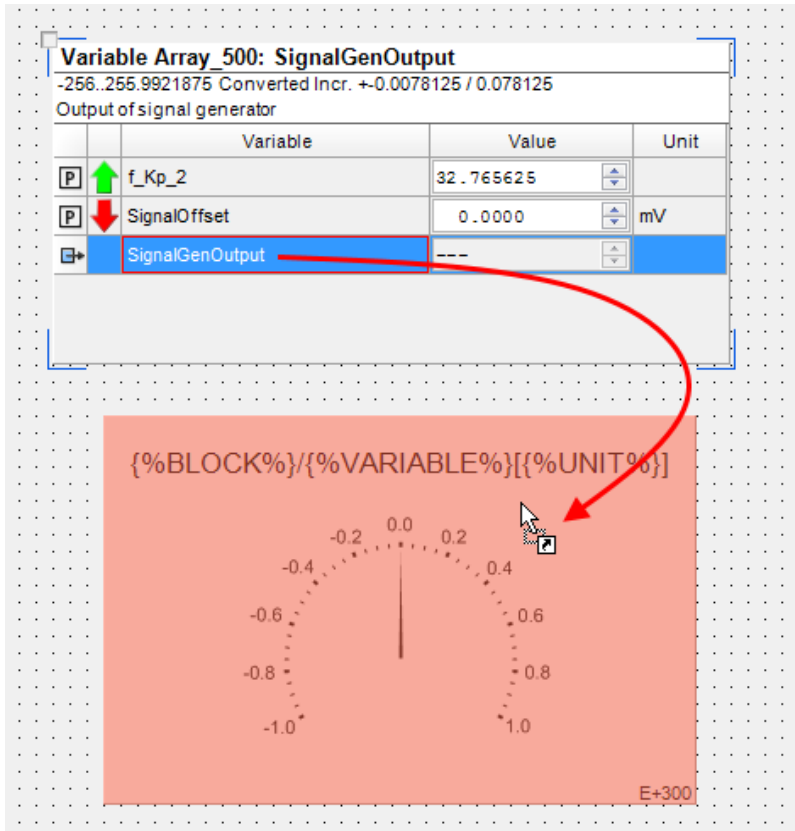
### Possible methods

- You can copy variables via drag and drop (only from Plotter, Table Editor, Variable Array). Refer to [Method 1](#) on page 42.
- You can copy variables via the context menu of an instrument. Refer to [Method 2](#) on page 43.

### Method 1

#### To copy variables (using drag and drop)

- 1 In an instrument, select the variable(s) to be copied.
- 2 Drag and drop it/them to another instrument.



It is/they are displayed in both instruments.

#### Tip

Pressing the right mouse button while dropping selected variables or whole instruments opens a context menu for specifying the insert behavior.

Depending on the drop position (instrument title, instrument variables, free area on a layout), you can, for example:

- Replace an existing variable in the target instrument
- Connect variables to a new instrument
- Copy an instrument with or without its variable connections
- Connect variables as additional write variables

---

**Method 2****To copy variables (via the context menu of the instrument)**

- 1 If the instrument contains more than one variable, select the variables to be copied.
- 2 From the context menu of the instrument, select **Variable(s) - Copy to Instrument** and specify the desired instrument.
- 3 The selected variable are displayed in the new instrument.

---

**Result**

You copied variables from one instrument to another.

---

**Related topics****References**

[Append Variable \(ControlDesk Instrument Handling !\[\]\(3e2231b1ad3ca8da8658228c00dd08e0\_img.jpg\)\)](#)  
[Connect as Additional Write Variable\(s\) \(ControlDesk Instrument Handling !\[\]\(96a82dd1250f57fd139c5f3b80c9d977\_img.jpg\)\)](#)  
[Connect to New Instrument \(ControlDesk Instrument Handling !\[\]\(3fd2f8db37e12aa5bbcaf4dfbd320f6c\_img.jpg\)\)](#)  
[Copy Here \(ControlDesk Instrument Handling !\[\]\(e0326b4151c040481d9d39947d969a7a\_img.jpg\)\)](#)  
[Copy Here Without Connections \(ControlDesk Instrument Handling !\[\]\(6f6ed6fc9826399634bf9176c1715113\_img.jpg\)\)](#)  
[Insert Before \(ControlDesk Instrument Handling !\[\]\(e7912652fed497d4cbe94d194939c368\_img.jpg\)\)](#)  
[Move Here \(ControlDesk Instrument Handling !\[\]\(902290fb460f6ccd802f6345b2e21605\_img.jpg\)\)](#)  
[Replace Variable \(ControlDesk Instrument Handling !\[\]\(e1fdced8dd9edfb61394a92c24dbca67\_img.jpg\)\)](#)

# Arranging Instruments on a Layout

## Where to go from here

## Information in this section

### Selecting and Positioning Instruments on a Layout.....44

You can use various methods to select and position instruments on a layout.

### How to Optimize the Instrument Arrangement.....46


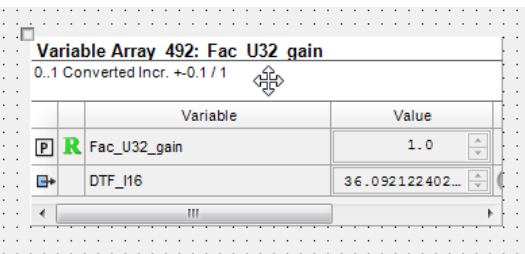
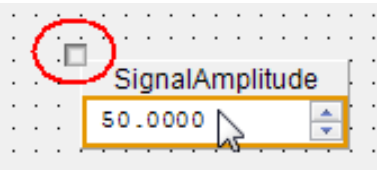
You can arrange instruments manually using the mouse. ControlDesk provides a number of commands to accelerate and optimize the arranging of instruments.

# Selecting and Positioning Instruments on a Layout

## Introduction

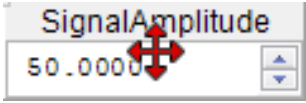
You can use various methods to select and position instruments on a layout.

## Sensitive move area

Example	Description
	Each instrument has a sensitive area used to drag it to a new position. In this area the mouse pointer changes to a move symbol.
	For some instruments, the sensitive area covers the whole instrument, for others it is only a part of the instrument. You can move the Variable Array, for example, by clicking the title bar or header and then dragging the pointer to the new position.
	Each instrument has at least one sensitive move area in the left upper corner: the selection handler, visualized by a square. If you have activated the selection handler via the instrument properties, you can use the square to drag the instrument to a new position.

**Shortcut keys**

You can use the following shortcut keys to move or resize selected instruments.

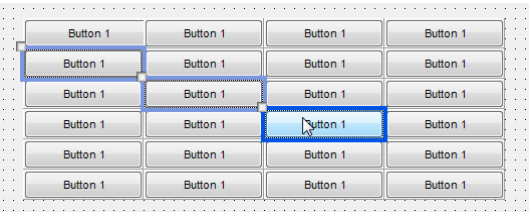
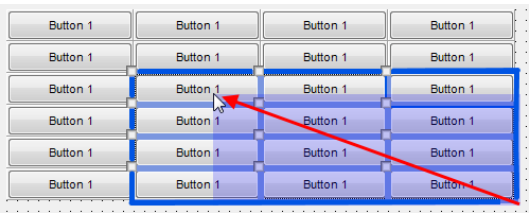
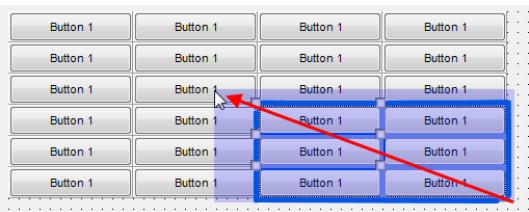
Shortcut Key	Description
<b>Ctrl+Shift+E</b>	To activate/deactivate the keyboard shortcuts for moving and resizing instruments. A symbol is displayed when the keyboard shortcuts are active: 
<b>↑↓↔</b>	To move the selected instruments.
<b>Shift+↑↓↔</b>	To resize the selected instruments.
<b>Tab</b>	To select the next instrument.

**Multiple selection of instruments**

You can select several instruments and move them together. Use the sensitive move area of one selected instrument to drag the complete selection.

**Tip**

Activate selection handlers or selection border styles to visualize the currently selected instruments.

Example	Description
<b>Adding More Selections with Ctrl</b>	
	<p>You can add more selections to existing ones by pressing <b>Ctrl</b> while selecting the next instrument(s).</p> <p>You can add single instruments with a left-click and groups of instruments by dragging the pointer (see below).</p>
<b>Selecting Partially Covered Instruments</b>	
	<p>You can select a group of instruments by clicking outside the group and moving the pointer across the group.</p> <p>All the instruments that are touched by the selection frame are selected.</p> <p>Pressing <b>Ctrl</b> allows you to add more selections.</p>
<b>Selecting Fully Covered Instruments with Shift</b>	
	<p>You can change the selection behavior by pressing <b>Shift</b> while moving the pointer. In this case, only instruments that are fully covered by the selection frame are selected.</p> <p>Pressing <b>Ctrl</b> allows you to add more selections.</p>

## Related topics

### HowTos

[How to Optimize the Instrument Arrangement..... 46](#)

# How to Optimize the Instrument Arrangement

## Objective

You can arrange instruments manually using the mouse. ControlDesk provides a number of commands to accelerate and optimize the arranging of instruments.

## Optimized arranging of instruments

The following commands are available to support the arranging of instruments:

- Align (Left, Right, Top, Bottom, Vertical Center, Horizontal Center)
- Space Evenly (Across, Down)
- Center in View (Vertically, Horizontally)
- Make Same Size (Width, Height, Both)
- Group
- Ungroup

If instruments overlap, you can arrange them with the following commands:

- Bring to Front
- Send to Back

## Automatic sizing

ControlDesk provides the automatic sizing property for some instruments. If enabled, the selected instrument adjusts the size of its display areas automatically.

## Locked mode

If you want to protect a layout from unintentional changes, you can lock it against editing.

## Method

### To optimize the instrument arrangement

- 1** Press the **Ctrl** key and click all instruments that you want to arrange.
- 2** On the **Layouting** ribbon, click **Arrange** and click one of the arrangement icons.
- 3** If you want to save the changes without saving the whole project/experiment, press **Ctrl+S**.

## Result

You arranged the instruments faster and more accurately.

---

**Related topics****References**

Align Bottom.....	53
Align Horizontal Center.....	54
Align Left.....	55
Align Right.....	55
Align Top.....	56
Align Vertical Center.....	57
Bring to Front.....	60
Center Horizontal.....	61
Center Vertical.....	61
Group.....	69
Locked Mode.....	82
Make Same Height.....	83
Make Same Size.....	84
Make Same Width.....	84
Send to Back.....	90
Space Across.....	92
Space Down.....	92
Ungroup.....	95





# Reference Information

## Where to go from here

## Information in this section

<a href="#">Activate (Layout).....</a>	<a href="#">53</a>
To activate a layout.	
<a href="#">Align Bottom.....</a>	<a href="#">53</a>
To align all the instruments selected in the layout with the bottom of the last selected item.	
<i>Not available in operator mode.</i>	
<a href="#">Align Horizontal Center.....</a>	<a href="#">54</a>
To arrange all selected instruments in a horizontal line.	
<i>Not available in operator mode.</i>	
<a href="#">Align Left.....</a>	<a href="#">55</a>
To align all selected instruments in the layout with the left border of the last selected item.	
<i>Not available in operator mode.</i>	
<a href="#">Align Right.....</a>	<a href="#">55</a>
To align all selected instruments in the layout with the right border of the last selected item.	
<i>Not available in operator mode.</i>	
<a href="#">Align Top.....</a>	<a href="#">56</a>
To align all selected instruments in the layout with the top of the last selected item.	
<i>Not available in operator mode.</i>	
<a href="#">Align Vertical Center.....</a>	<a href="#">57</a>
To arrange all selected instruments in a vertical line.	
<i>Not available in operator mode.</i>	
<a href="#">Assign Platforms.....</a>	<a href="#">57</a>
To assign unresolved variable connections to platforms/devices in the experiment.	
<i>Not available in operator mode.</i>	

<b>Autoselect Variable</b> .....	59
To select and highlight the main variable connected to the selected instrument in the Variables controlbar.	
<b>Bring to Front</b> .....	60
To bring the selected instruments in the layout to the front.	
<i>Not available in operator mode.</i>	
<b>Center Horizontal</b> .....	61
To center the selected instruments horizontally in the layout.	
<i>Not available in operator mode.</i>	
<b>Center Vertical</b> .....	61
To center the selected instruments vertically in the layout.	
<i>Not available in operator mode.</i>	
<b>Close (Layout)</b> .....	62
To close one or more open layouts.	
<b>Collapse All</b> .....	62
To collapse all the nodes and subnodes of the tree.	
<b>Create Copy</b> .....	63
To create a new layout as a copy of the currently selected layout.	
<i>Not available in operator mode.</i>	
<b>Create Python Reference</b> .....	64
To create the Python string that references the instrument.	
<i>Not available in operator mode.</i>	
<b>Delete Invalid</b> .....	65
To delete invalid variable-instrument connections.	
<i>Not available in operator mode.</i>	
<b>Expand All</b> .....	65
To expand all the nodes and subnodes of the tree.	
<b>Export Active Layout Connections</b> .....	66
To export the variable connections of the instruments on the currently active layout to a CONX file.	
<b>Export Connections</b> .....	66
To export the variable connections of the instruments on all currently opened layouts to a CONX file.	
<b>Export Layout / Export</b> .....	67
To export the selected layout(s) with all instruments, variable connections, and layout-specific events.	
<i>Not available in operator mode.</i>	
<b>Go to Instrument</b> .....	68
To switch the focus to the associated instrument on the layout.	
<b>Group</b> .....	69
To group selected instruments.	
<i>Not available in operator mode.</i>	


<b>Import Connections.....</b>	<b>70</b>
To import variable connections from a CON or CONX file.	
<b>Import Layout/Import.....</b>	<b>70</b>
To import a layout with all instruments and variable connections.	
<i>Not available in operator mode.</i>	
<b>Insert Layout.....</b>	<b>72</b>
To insert a new empty layout into the active experiment.	
<i>Not available in operator mode.</i>	
<b>Instrument Navigator.....</b>	<b>73</b>
To show the Instrument Navigator.	
<b>Layout Navigator.....</b>	<b>76</b>
To show the Layout Navigator.	
<b>Layout Properties.....</b>	<b>80</b>
To view and edit the properties of the currently active layout.	
<b>Locked Mode.....</b>	<b>82</b>
To lock/unlock layout editing.	
<i>Not available in operator mode.</i>	
<b>Make Same Height.....</b>	<b>83</b>
To resize the selected instruments in the layout to the height of the last selected item.	
<i>Not available in operator mode.</i>	
<b>Make Same Size.....</b>	<b>84</b>
To resize the selected instruments in the layout to the width and height of the last selected item.	
<i>Not available in operator mode.</i>	
<b>Make Same Width.....</b>	<b>84</b>
To resize the selected instruments in the layout to the width of the last selected item.	
<i>Not available in operator mode.</i>	
<b>Move Down.....</b>	<b>85</b>
To move the selected instruments in the layout one layer down.	
<i>Not available in operator mode.</i>	
<b>Move Up.....</b>	<b>86</b>
To move the selected instruments in the layout one layer up.	
<i>Not available in operator mode.</i>	
<b>Open (Layout).....</b>	<b>86</b>
To open one or more closed layouts.	
<b>Paste / Paste Instruments with Connections.....</b>	<b>87</b>
To paste the copied instruments in the current layout with variable connections.	
<i>Not available in operator mode.</i>	

Paste Instruments Without Connections.....	88
To paste the copied instruments in the current layout without variable connections.	
<i>Not available in operator mode.</i>	
Remove (Layout).....	88
To remove one or more layouts from the experiment.	
<i>Not available in operator mode.</i>	
Rename (Layout).....	89
To rename a layout.	
<i>Not available in operator mode.</i>	
Scroll to Instrument.....	90
To scroll the layout to make the selected instrument visible.	
Send to Back.....	90
To send the selected instruments in the layout to the back.	
<i>Not available in operator mode.</i>	
Show Disconnected.....	91
To highlight the instruments which are not connected with a main variable in the current layout.	
Space Across.....	92
To equalize the distance of selected instruments in the layout between the left-most and right-most instruments.	
<i>Not available in operator mode.</i>	
Space Down.....	92
To equalize the distance of selected instruments in the layout between the uppermost and lowest instruments.	
<i>Not available in operator mode.</i>	
Switch Data Source.....	93
To switch the variable connections of all instruments of the selected layout to another source.	
Synchronize (Layout).....	94
To synchronize the local copy of a layout with the layout it is referenced to.	
<i>Not available in operator mode.</i>	
Ungroup.....	95
To ungroup an instrument group.	
<i>Not available in operator mode.</i>	
Visualization Page.....	95
To specify visualization settings.	
Visualize Variables.....	97
To connect variables to instruments and place them on a layout.	
<i>Not available in operator mode.</i>	

## Activate (Layout)

### Access

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>▪ <a href="#">Project</a>  controlbar – Layout</li> <li>▪ Layout Navigator - Layout</li> </ul>
Shortcut key	None
Icon	None

### Purpose

To activate a layout.

### Result

The selected layout is activated in ControlDesk's working area.

### Related topics


#### Basics

[Basics of Handling Layouts.....](#) 10

## Align Bottom

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	<b>Ctrl+Shift+↓</b> (Available only if enabled via <b>Ctrl+Shift+E</b> )
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To align all the instruments selected in the layout with the bottom of the last selected item.

---

**Description** The last selected item is marked by a black frame.

---


**Related topics****HowTos**

[How to Optimize the Instrument Arrangement..... 46](#)

## Align Horizontal Center

**Access**

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To arrange all selected instruments in a horizontal line.

**Result**

The instruments you selected in the layout are aligned with the horizontal center of the last selected item.

**Description**

The last selected item is marked by a black frame.


**Related topics****HowTos**

[How to Optimize the Instrument Arrangement..... 46](#)

## Align Left

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	<b>Ctrl+Shift+←</b> (Available only if enabled via <b>Ctrl+Shift+E</b> )
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To align all selected instruments in the layout with the left border of the last selected item.

### Description

The last selected item is marked by a black frame.

### Related topics


#### HowTos

[How to Optimize the Instrument Arrangement..... 46](#)

## Align Right

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	<b>Ctrl+Shift+→</b> (Available only if enabled via <b>Ctrl+Shift+E</b> )
Icon	

#### Note

This command is not available in operator mode.

---

**Purpose** To align all selected instruments in the layout with the right border of the last selected item.

---

**Description** The last selected item is marked by a black frame.

---

**Related topics**

HowTos


[How to Optimize the Instrument Arrangement..... 46](#)

## Align Top

---

**Access**

You can access this command via:

Ribbon	Layouting - Arrange
Context menu of	None
Shortcut key	<b>Ctrl+Shift+↑</b> (Available only if enabled via <b>Ctrl+Shift+E</b> )
Icon	

**Note**

This command is not available in operator mode.

---

**Purpose** To align all selected instruments in the layout with the top of the last selected item.

---

**Description** The last selected item is marked by a black frame.

---

**Related topics**

HowTos


[How to Optimize the Instrument Arrangement..... 46](#)



## Align Vertical Center

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To arrange all selected instruments in a vertical line.

### Result

The instruments you selected in the layout are aligned with the vertical center of the last selected item.

### Description

The last selected item is marked by a black frame.

### Related topics


#### HowTos

[How to Optimize the Instrument Arrangement.....](#) 46

## Assign Platforms

### Access

You can access this command via:

Ribbon	Layouting - Connections
Context menu of	Layouts in the <a href="#">Project</a> <sup>?</sup> controlbar
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

---

<b>Purpose</b>	To assign unresolved variable connections to platforms/devices in the experiment.
----------------	---

---

**Description**

**Variable connections and their assignment to platforms/devices** In ControlDesk, a *variable connection* is the connection of a variable to something such as an instrument or a signal of a signal generator.

When you place a variable of a specific variable description on an instrument, the variable connection is implicitly assigned to the platform/device that the variable description belongs to.

**Automatic assignment** When you activate an experiment, ControlDesk automatically resolves the assignment of variable connections to platforms/devices in the experiment if it contains a platform/device which has the same Platform name in experiment property as the originally assigned platform/device.

**No automatic assignment in some cases** There are some cases where ControlDesk cannot resolve the assignment of variable connections to a specific platform/device. For example:

- You import a layout (LAX) file originally exported from another experiment. If the target experiment does not contain a platform/device with the same name as the source experiment's platform/device that the connections of the layout were assigned to, ControlDesk cannot resolve the assignment between the connections to a specific platform/device in the target experiment.
- You want to use an experiment containing layouts with a platform/device whose name differs from the original platform/device's name.

**Manual assignment** If ControlDesk cannot resolve the assignment automatically, you can assign a platform/device in the experiment manually with the Assign Platforms command. You can assign a platform/device either for an individual layout selected in the Project controlbar, or for all the experiment layouts that are currently open.

**Note**

When connections are assigned to a platform/device with another Platform name in experiment, ControlDesk changes the related variable identifiers. If the original Platform name in experiment is XCP, for example, and the new Platform name in experiment is XCP\_2, all the variable identifiers are changed from 'XCP()://<variable\_name>' to 'XCP\_2()://<variable\_name>'.

You may have to adapt your automation scripts accordingly.

When you activate an experiment, ControlDesk automatically resolves the assignment of variable connections to platforms/devices in the experiment if it contains a platform/device which has the same Platform name in experiment property as the originally assigned platform/device.

Tip

To prevent ControlDesk from changing variable identifiers, you can also rename a platform/device via the [Rename Platform/Device \(ControlDesk Platform Management !\[\]\(c507f772dba2b921f86777f01218e570\_img.jpg\)](#)) command, which allows ControlDesk to perform automatic assignment.

Assign Platforms for layouts dialog // Assign Platforms for <layout\_name> dialog

Lists the missing platforms/devices that variable connections cannot be assigned to, and lets you manually select existing platforms/devices in the experiment to assign variable connections to.

- Missing** Displays the Platform name in experiment property of each platform/device that at least one variable connection cannot be resolved for.
- If you opened the dialog via the context menu of a layout selected in the Project controlbar, the missing platforms/devices are displayed for the selected layout.
  - If you opened the dialog via the Layouting ribbon, all the currently open layouts are displayed.

**Existing** Lets you select the Platform name in experiment of a platform/device available in the current experiment to let ControlDesk assign it to the variable connection(s).

Related topics


References

[Rename Platform/Device \(ControlDesk Platform Management !\[\]\(5361750c22c4e047a52f4eac1ec2d4cc\_img.jpg\)](#))

Autoselect Variable

Access

You can access this command via:

Ribbon	Layouting - Instruments
Context menu of	None
Shortcut key	None
Icon	 (Checkbox)

Purpose

To select and highlight the main variable connected to the selected instrument in the Variables controlbar.

**Result**

In the **Variables**, the variable is highlighted as follows:

- In the tree view, the node in which the variable is located is selected.
- In the variable list, the variable is displayed and highlighted.


**Related topics****Basics**

[Basics of Handling Layouts..... 10](#)

## Bring to Front

**Access**

You can access this command via:

Ribbon	Layouting – Layer
Context menu of	Instrument Navigator - instrument node
Shortcut key	<b>Ctrl+Shift+F</b>
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To bring the selected instruments in the layout to the front.

**Result**

The selected instruments will cover the instruments in the background if their borders overlapped before this command was carried out.


**Related topics****HowTos**

[How to Optimize the Instrument Arrangement..... 46](#)

## Center Horizontal

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To center the selected instruments horizontally in the layout.

### Result

The selected instruments are centered in the layout horizontally. If several instruments are in a horizontal line, they will overlap each other.

### Related topics


HowTos

[How to Optimize the Instrument Arrangement..... 46](#)

## Center Vertical

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To center the selected instruments vertically in the layout.

---

**Result** The selected instruments are centered vertically in the layout. If several instruments are in a vertical line, they will overlap each other.

---


**Related topics****HowTos**

[How to Optimize the Instrument Arrangement..... 46](#)

## Close (Layout)

**Access**

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"><li>▪ <a href="#">Project</a>  controlbar – Layout(s)</li><li>▪ Layout Navigator - Layout</li></ul>
Shortcut key	None
Icon	None

**Purpose**

To close one or more open layouts.

**Result**

If you have changed a layout, for example, by scrolling to a new position, ControlDesk prompts you to save it. The name of a closed layout is grayed out in the Project controlbar.

**Related topics****Basics**

[Basics of Handling Layouts..... 10](#)

## Collapse All

**Access**

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"><li>▪ Instrument Navigator</li></ul>

	<ul style="list-style-type: none"> <li>▪ Layout Navigator</li> <li>▪ Bus Navigator</li> </ul>
Shortcut key	None
Icon	None

**Purpose** To collapse all the nodes and subnodes of the tree.

**Result** Only the instruments are displayed in the tree, the connected variables are hidden.

## Related topics

### References

[Expand All..... 65](#)

## Create Copy

**Access** This command is available only for unlocked layouts. You can access it via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>▪ Layout</li> <li>▪ Layout Navigator - Layout</li> </ul>
Shortcut key	None
Toolbar icon	None

### Note

This command is not available in operator mode.

**Purpose** To create a new layout as a copy of the currently selected layout.

**Result** The copied layout contains the same instruments and variable connections as the original layout. It is added to the currently active experiment and saved. The name of the copied layout is **Copy of <Name of original layout>**.

**Description** This command is an easy way to create a layout for viewing recorded data, for example. Multiple selection is possible.

**Related topics****References**[Switch Data Source..... 93](#)

## Create Python Reference

**Access**

You can access this command via:

Ribbon	None
Context menu of	Instrument Navigator - instrument
Shortcut key	None
Icon	None

**Note**

This command is not available in operator mode.

**Purpose**

To create the Python string that references the instrument.

**Result**

ControlDesk creates the Python string that references the selected instrument and copies it to the Clipboard. You can paste the string into your Python code.

Below is an example Python string referencing the **TimePlotter** on **Layout1**.

```
Application.LayoutManagement.Layouts["Layout1"].GetInstrument("TimePlotter")
```

**Related topics****References**[Instrument Navigator..... 73](#)



## Delete Invalid

### Access

You can access this command via:

Ribbon	Layouting – Connections
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To delete invalid variable-instrument connections.

### Related topics

#### Basics

[Basics of Handling Layouts.....](#) 10

## Expand All

### Access

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>▪ Instrument Navigator</li> <li>▪ Layout Navigator</li> <li>▪ Bus Navigator</li> </ul>
Shortcut key	None
Icon	None

### Purpose

To expand all the nodes and subnodes of the tree.

---

**Result** All items are displayed in the tree.

---

**Related topics****References**

[Collapse All..... 62](#)

## Export Active Layout Connections

---

**Access**

You can access this command via:

Ribbon	Layouting - Connections - Export Connections
Context menu of	None
Shortcut key	None
Icon	None

**Purpose**

To export the variable connections of the instruments on the currently active layout to a CONX file.

---

**Result**

The variable connections are exported to the CONX file you specified.

---

**Related topics****HowTos**

[How to Export/Import Variable Connections..... 21](#)

**References**

[Export Connections..... 66](#)

## Export Connections

---

**Access**

You can access this command via:

Ribbon	Layouting - Connections - Export Connections
Context menu of	None

Shortcut key	None
Icon	

**Purpose** To export the variable connections of the instruments on all currently opened layouts to a CONX file.



**Result** The variable connections are exported to the CONX file you specified.

**Note**

Variable connections of Bus Instruments are not exported.

<b>Related topics</b>	<b>HowTos</b>
	<a href="#">How to Export/Import Variable Connections.....</a> 21
	<b>References</b>
	<a href="#">Export Active Layout Connections.....</a> 66 <a href="#">Visualization Page.....</a> 95

## Export Layout / Export

<b>Access</b>	You can access this command via:
Ribbon	Layouting - Layouts
Context menu of	Layout Navigator - Layout <a href="#">Project</a>  controlbar - Selected layout(s)
Shortcut key	None
Icon	
<div><p><b>Note</b></p><p>This command is not available in operator mode.</p></div>	

---

**Purpose** To export the selected layout(s) with all instruments, variable connections, and layout-specific events.

---

**Result** Opens a standard file dialog to specify the file name(s). Each layout is stored as an LAX file. The LAX file also contains the layout's variable connections.

If you access the command via the Project controlbar, ControlDesk opens the Browse for Folder dialog to specify an export folder for the LAX files.

**Note**

Exported LAX files can be imported to the same or to a newer ControlDesk version only.

---

**Related topics**

**Basics**

[Switching the Simulation Platform and Reusing Experiment Parts \(ControlDesk Platform Management !\[\]\(ec9132f1d27c8919987d92907322654d\_img.jpg\)](#))

**HowTos**

[How to Export/Import Layouts..... 15](#)

## Go to Instrument

---

**Access**

You can access this command via:

Ribbon	None
Context menu of	Instrument Navigator - instrument node
Shortcut key	Enter
Icon	None

---

**Purpose**

To switch the focus to the associated instrument on the layout.

---

**Result**

The associated instrument gets the focus. If you now press a shortcut key, for example, it belongs to the instrument on the layout and not to the instrument item in the Instrument Navigator.


**Related topics****References**

[Instrument Navigator..... 73](#)

## Group

**Access**

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	Instrument Navigator - instrument node
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To group selected instruments.

**Result**

All selected instruments are combined in a group of instruments.

**Description**

Grouped instruments have a common frame. You can change the properties of the instruments via the **Properties** dialogs, but you cannot resize or shift instruments inside a group.


**Related topics****HowTos**

[How to Optimize the Instrument Arrangement..... 46](#)

## Import Connections

### Access

You can access this command via:

Ribbon	Layouting - Connections
Context menu of	None
Shortcut key	None
Icon	

### Purpose

To import variable connections from a CON or CONX file.

### Result

The variable connections are imported from the CON or CONX file you specified.

#### Note

Variable connections are imported only to currently open layouts of the active experiment. They are not imported to currently closed layouts.

### Related topics

#### HowTos

[How to Export/Import Variable Connections.....](#) 21



#### References

[Export Connections.....](#) 66  
[Visualization Page.....](#) 95

## Import Layout/Import

### Access

You can access this command via:

Ribbon	Layouting - Layouts
Context menu of	<a href="#">Project</a>  controlbar – Experiment Layouts folder or group
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To import a layout with all instruments and variable connections.

**Note**

Only LAX files contain variable connections. If you import a LAY file, you must recreate all the variable connections later on.

**Result**


ControlDesk opens the Import Layout(s) dialog that lets you select the layout to be imported.

**Description**

The imported layout is copied to the Experiment Layouts folder in the Project controlbar. It is automatically renamed if necessary. When the imported layout is loaded, ControlDesk checks all imported variable connections.

**Note**

Exported LAX files can be imported to the same or to a newer ControlDesk version only.

**Referenced files** You can import a layout as a local working copy with a reference to the external layout file by importing it *as link*. You can then synchronize the layout settings via the context menu of the new experiment layout. Refer to [Synchronize \(Layout\)](#) on page 94. In the Project controlbar, a layout that references an external file is visualized by a layout icon that shows a chain: 

ControlDesk compares the layout settings of the local layout file and its reference when an experiment is activated. If it detects a difference, it asks you if you want to update the local file.

**Import Layout(s) dialog**

Lets you import a layout with all instruments and variable connections.

**Note**

Only LAX files contain variable connections. If you import a LAY file, you must recreate all the variable connections later on.

**Import as link** Lets you import the layout as a local working copy that references the selected layout file.

---

**Related topics****HowTos**



[How to Export/Import Layouts.....](#) 15

## Insert Layout

---

**Access**

You can access this command via:

Ribbon	Layouting - Layouts
Context menu of	<a href="#">Project</a>  controlbar – Experiment Layouts folder or group
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

---

**Purpose**

To insert a new empty layout into the active experiment.

---

**Result**

A new empty layout is inserted into the active experiment.

In a layout you can calibrate and measure variables. You can also use it to visualize recorded data.

---

**Related topics****HowTos**


[How to Create a New Layout.....](#) 12



# Instrument Navigator

## Access

You can access this command via:

Ribbon	View – Controlbar - Switch Controlbars
Context menu of	None
Shortcut key	None
Icon	

## Purpose

To show the Instrument Navigator.

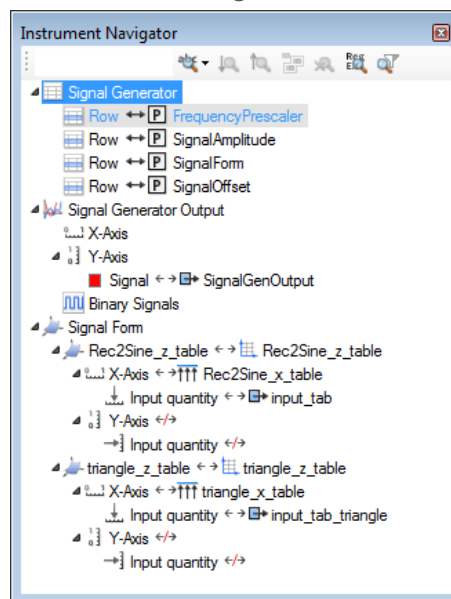
## Result

The Instrument Navigator opens.

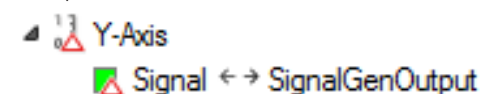
## Description

**Instrument Navigator** A controlbar that gives access to all the instruments and variables that are placed on the currently active layout.

The Instrument Navigator looks like this:






















An invalid connection is visualized by an added warning icon, see the following example:



Data connections are invalid if the source or the target of the connection is removed from the application or Simulink simulation, or if an internal error occurs.

**Function buttons** The following function buttons are available below the Instrument Navigator header.

Function Button	Description
	Lets you set the search scope to all text fields. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Lets you limit the search scope to instrument names. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Lets you limit the search scope to variable names. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Highlights all unconnected variables. Click ▼ to change the search scope.
	Highlights all invalid connections. Click ▼ to change the search scope.
	Selects the previous occurrence of the search string.
	Selects the next occurrence of the search string.
	Selects all the occurrences of the search string.
	Stops the search and deletes the search string.
	Lets you search with regular expressions. For information on regular expressions, see Regular expressions.
	Lets you filter the search results. If activated, only the matching results and their parent objects are displayed.
	If necessary, ControlDesk scrolls the layout to make the selected item visible.
	On the layout, ControlDesk highlights all the instruments that contain items of a search result.

**Wildcard search** You can use implicit and explicit wildcards in the filter string.

- **Implicit wildcards:** If you type a search string without wildcards, an implicit \* is set at the beginning and at the end of the string.  
For example: **or** finds the same elements as **\*or\***.

- **Explicit wildcards:** If you enter at least one wildcard, implicit wildcards are disabled. All wildcards must be set explicitly.
  - You can use the `*` wildcard for any number of missing letters in the filter string.  
For example: `*or`, `or*`, and `*or*` specify different searches.
  - If the filter string contains at least one explicit `*` wildcard, you can also use the `?` wildcard for one missing letter in the filter string.  
For example: `*_?_table` finds *triangle\_z\_table* and *triangle\_x\_table*.

**Regular expressions** The following table gives you an overview of the expressions you can use if regular expressions are activated in the toolbar.

Regular Expression	Purpose	Example
.	Matches any single character.	<code>ho.se</code> matches horse and house.
\<	Matches the start of a word.	<code>\&lt;art</code> matches artist but not start.
\>	Matches the end of a word.	<code>\&gt;art</code> matches start but not artist.
[...]	Matches a single character that is contained within the brackets. For example, <code>[abc]</code> matches "a", "b", or "c". <code>[a-z]</code> specifies a range which matches any lowercase letter from "a" to "z".	<code>a[ur]t</code> matches auto or artist but not alternate.
[^...]	Matches a single character that is not contained within the brackets. For example, <code>[^abc]</code> matches any character other than "a", "b", or "c". <code>[^a-z]</code> matches any single character that is not a lowercase letter from "a" to "z".	<code>a[^ur]t</code> matches alternative but not automobile or artist.
^	Matches the start of a line.	<code>^dig</code> matches digit at the start of a line.
\$	Matches the end of a line.	<code>git\$</code> matches digit at the end of a line.
*	Matches the preceding character zero or more times.	<code>di*git</code> matches dgit, digit, diigit, diiigit, etc.
+	Matches the preceding character one or more times.	<code>di+git</code> matches digit, diigit, diiigit, etc.
\x	Allows you to use a character x that would otherwise have a special meaning.	<code>2\+2&lt;5</code> matches 2+2<5.

**Positioning the controlbar** By default, the Instrument Navigator is located at the left side of the ControlDesk user interface.


You can shift controlbars to any position inside the working area or dock them at the border of the working area. For instructions on positioning controlbars, refer to [How to Customize the Screen Arrangement \(ControlDesk User Interface Handling !\[\]\(a03a7eb2f4046e1d3c76772003e549ea\_img.jpg\)](#)).

**Related commands**

The Instrument Navigator provides the following commands:

Purpose	Refer to
To bring the selected instruments in the layout to the front.	<a href="#">Bring to Front</a> on page 60
To collapse all the nodes and subnodes of the tree.	<a href="#">Collapse All</a> on page 62
To link Python code to a specific event in ControlDesk.	<a href="#">Configure Events</a> ( <a href="#">ControlDesk Automation</a>  )
To copy the selected instruments to the Clipboard.	<a href="#">Copy (Instrument)</a> ( <a href="#">ControlDesk Instrument Handling</a>  )
To create the Python string that references the instrument.	<a href="#">Create Python Reference</a> on page 64
To cut the selected instruments to the Clipboard.	<a href="#">Cut (Instrument)</a> ( <a href="#">ControlDesk Instrument Handling</a>  )
To delete the selected instruments.	<a href="#">Delete (Instrument)</a> ( <a href="#">ControlDesk Instrument Handling</a>  )
To expand all the nodes and subnodes of the tree.	<a href="#">Expand All</a> on page 65
To switch the focus to the associated instrument on the layout.	<a href="#">Go to Instrument</a> on page 68
To group selected instruments.	<a href="#">Group</a> on page 69
To view and edit the properties of an instrument.	<a href="#">Properties</a> (refer to <a href="#">Instrument Properties</a> ( <a href="#">ControlDesk Instrument Handling</a>  ))
To scroll the layout to make the selected instrument visible.	<a href="#">Scroll to Instrument</a> on page 90
To send the selected instruments in the layout to the back.	<a href="#">Send to Back</a> on page 90
To copy the selected variables to the Clipboard.	<a href="#">Copy / Variables - Copy</a> ( <a href="#">ControlDesk Instrument Handling</a>  )
To cut the selected variables in the current instrument.	<a href="#">Variables - Cut</a> ( <a href="#">ControlDesk Instrument Handling</a>  )
To delete the selected variables.	<a href="#">Variables - Delete</a> ( <a href="#">ControlDesk Instrument Handling</a>  )
To paste variables from the Clipboard to an instrument.	<a href="#">Variables - Paste/Paste Variables</a> ( <a href="#">ControlDesk Instrument Handling</a>  )

**Related topics****Basics**


[Instruments and the Variable Types they can Visualize](#) ([ControlDesk Instrument Handling](#) )

## Layout Navigator

**Access**

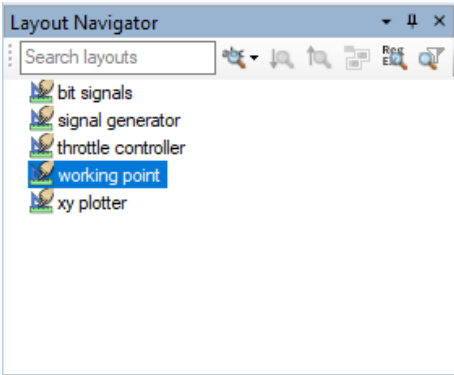
You can access this command via:

Ribbon	View – Controlbars - Switch Controlbars
Context menu of	None
Shortcut key	None













Icon	
Purpose	To show the Layout Navigator.
Result	The Layout Navigator opens.












**Description**      **Layout Navigator**      A controlbar that gives access to all the layouts of the currently active experiment.

The Instrument Navigator looks like this:



**Function buttons**      The following function buttons are available below the Layout Navigator header.

Function Button	Description
	Lets you set the search scope to all text fields. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Lets you limit the search scope to layout names. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Lets you limit the search scope to open layouts. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Lets you limit the search scope to closed layouts. Enter the search string in the edit field and click  or  to find the

Function Button	Description
	previous or the next matching item. To change the search scope, click ▼.
	Lets you limit the search scope to layout groups. Enter the search string in the edit field and click  or  to find the previous or the next matching item. To change the search scope, click ▼.
	Data connections are invalid if the source or the target of the connection is removed from the application or Simulink simulation, or if an internal error occurs.
	Selects the previous occurrence of the search string.
	Selects the next occurrence of the search string.
	Selects all the occurrences of the search string.
	Stops the search and deletes the search string.
	Lets you search with regular expressions. For information on regular expressions, see Regular expressions.
	Lets you filter the search results. If activated, only the matching results and their parent objects are displayed.

**Wildcard search** You can use implicit and explicit wildcards in the filter string.

- Implicit wildcards: If you type a search string without wildcards, an implicit \* is set at the beginning and at the end of the string.  
For example: **or** finds the same elements as **\*or\***.
- Explicit wildcards: If you enter at least one wildcard, implicit wildcards are disabled. All wildcards must be set explicitly.
  - You can use the \* wildcard for any number of missing letters in the filter string.  
For example: **\*or**, **or\***, and **\*or\*** specify different searches.
  - If the filter string contains at least one explicit \* wildcard, you can also use the ? wildcard for one missing letter in the filter string.  
For example: **\*\_?\_table** finds *triangle\_z\_table* and *triangle\_x\_table*.

**Regular expressions** The following table gives you an overview on the expressions you can use if regular expressions are activated in the toolbar.

Regular Expression	Purpose	Example
.	Matches any single character.	<b>ho.se</b> matches horse and house.
\<	Matches the start of a word.	<b>\&lt;art</b> matches artist but not start.
\>	Matches the end of a word.	<b>\&gt;art</b> matches start but not artist.
[...]	Matches a single character that is contained within the brackets. For example, <b>[abc]</b> matches "a", "b", or "c".	<b>a[ur]t</b> matches auto or artist but not alternate.



Regular Expression	Purpose	Example
[^...]	[a-z] specifies a range which matches any lowercase letter from "a" to "z". Matches a single character that is not contained within the brackets. For example, [^abc] matches any character other than "a", "b", or "c". [^a-z] matches any single character that is not a lowercase letter from "a" to "z".	a[ur]t matches alternative but not automobile or artist.
^	Matches the start of a line.	^dig matches digit at the start of a line.
\$	Matches the end of a line.	git\$ matches digit at the end of a line.
*	Matches the preceding character zero or more times.	di*git matches dgit, digit, diigit, diiigit, etc.
+	Matches the preceding character one or more times.	di+git matches digit, diigit, diiigit, etc.
\x	Allows you to use a character x that would otherwise have a special meaning.	2\+2<5 matches 2+2<5.

**Positioning the controlbar** By default, the Layout Navigator is located at the left side of the ControlDesk user interface.

You can shift controlbars to any position inside the working area or dock them at the border of the working area. For instructions on positioning controlbars, refer to [How to Customize the Screen Arrangement \(ControlDesk User Interface Handling !\[\]\(feabb98897b440bc8695a03336a6e2df\_img.jpg\)](#)).

## Related commands

The Layout Navigator provides the following commands:

Purpose	Refer to
To activate a layout.	<a href="#">Activate (Layout)</a> on page 53
To close one or more open layouts.	<a href="#">Close (Layout)</a> on page 62
To create a new layout as a copy of the currently selected layout.	<a href="#">Create Copy</a> on page 63
To export the selected layout(s) with all instruments, variable connections, and layout-specific events.	<a href="#">Export Layout / Export</a> on page 67
To paste the copied instruments in the current layout with variable connections.	<a href="#">Paste / Paste Instruments with Connections</a> on page 87
To paste the copied instruments in the current layout without variable connections.	<a href="#">Paste Instruments Without Connections</a> on page 88
To paste variables from the Clipboard to an instrument.	<a href="#">Variables - Paste/Paste Variables (ControlDesk Instrument Handling </a> )
To view and edit the properties of a layout.	Properties (refer to <a href="#">Properties (Controlbar) (ControlDesk User Interface Handling </a> ))
To remove one or more layouts from the experiment.	<a href="#">Remove (Layout)</a> on page 88

Purpose	Refer to
To rename a layout.	<a href="#">Rename (Layout)</a> on page 89
To switch the variable connections of all the instruments of the selected layout to another source.	<a href="#">Switch Data Source</a> on page 93

## Related topics

### Basics

[Basics of Handling Layouts](#)..... 10

## Layout Properties

### Purpose

To view and edit the properties of the currently active layout.

### Content size properties

**Content size** Lets you specify the size of the layout (in pixels). Open the dialog to enter or select values for width and height. You can enable or disable the fixed layout size. If it is enabled, you can place instruments only within the fixed size. If it is disabled, you can enlarge the layout size by moving an instrument to the outside of the current layout border. You cannot enter or select values that are smaller than the size that is needed for the instruments currently included. If you do so, the layout size is set to the minimum size needed for the instruments. You can intentionally choose this minimum size by selecting **Fit to Min**.

### Grid properties

**Grid enabled** Lets you specify whether a grid is visible in the layout. If enabled, you can specify the values for the width and height of the grid in pixels. If you move or create instruments on a layout with the grid enabled, they will be snapped to it.

**Grid size** Displays the height and width of the grid.

**Grid size - Height** Lets you specify the vertical line spacing of the grid.

**Grid size - Width** Lets you specify the horizontal line spacing of the grid.

### Instruments properties

**Selection handle enabled** Lets you specify whether the instrument's selection handle is visible. The selection handle is a small rectangle that appears in the top left corner of an instrument when you move the pointer over it. Typically, the selection handle is unavailable for a layout in locked mode. When the locked mode is switched off, the selection handle is automatically enabled.



## Layout properties



**Background color** Opens a dialog where you can choose a color. Click a color or enter RGB values to define it.

### Tip

On the Visualization page of the ControlDesk Options dialog, you can specify to use a custom background color for new inserted layouts. Refer to [Visualization Page](#) on page 95.

**Description** Lets you specify a description for the layout.

**Locked** Lets you lock/unlock the layout for editing. To lock layout editing is an appropriate means to protect a layout from unintentional changes. If layout editing is locked you cannot, for example, delete, add, or move instruments. Some actions belonging to the variables of an instrument are also restricted. For example, you cannot add variables via drag & drop.

Symbol	Meaning
	Locked mode
	Unlocked mode

**Layout name** Lets you specify the name of the layout. Changing this name will save the layout under the new name. The names of the layouts in an experiment must be unique.

**Script** Lets you assign Python code to an event of the selected layout. The Code Editor dialog is opened, containing event handler templates for the layout. Refer to [Code Editor Dialog \(ControlDesk Instrument Handling\)](#). For more information on adding a script to a layout, refer to [Adding a Python Script to an Instrument or Layout \(ControlDesk Customization\)](#).

**Source Layout Properties** Lets you display the path to a linked layout as a relative or an absolute path. You create a linked layout by activating Import as link when you import a layout from somewhere else, such as from a layout pool. Refer to [Import Layout/Import](#) on page 70.

**Source Layout – Relative** Lets you specify to display the path of a linked layout in relation to the project's layout folder.


**Source Layout – Path** Displays the path of a linked layout. If Source Layout - Relative is selected, the layout path is displayed in relation to the project's layout folder.

**Related topics****HowTos**[How to Configure the Properties of a Layout..... 12](#)**References**[Properties \(Controlbar\) \(ControlDesk User Interface Handling !\[\]\(339a16584d5da0f0a3ca4e9ec17bf6a1\_img.jpg\)\)](#)

## Locked Mode

**Access**

You can access this command via:

Ribbon	Layouting – Layouts
Context menu of	Layout
Shortcut key	None
Icon	

**Note**



This command is not available in operator mode.

**Purpose**

To lock/unlock layout editing.

**Result**

ControlDesk locks or unlocks the editing of the selected layout.

Symbol	Meaning
	Locked mode
	Unlocked mode

**Description**

To lock layout editing is an appropriate means to protect a layout from unintentional changes. If layout editing is locked you cannot, for example, delete, add, or move instruments. Some actions belonging to the variables of an instrument are also restricted. For example, you cannot add variables via drag and drop.

**Related topics****Basics**

[Basics of Handling Layouts.....](#) 10


**HowTos**

[How to Optimize the Instrument Arrangement.....](#) 46

## Make Same Height

**Access**

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To resize the selected instruments in the layout to the height of the last selected item.

**Description**

The last selected item is marked by a black frame.


**Related topics****HowTos**

[How to Optimize the Instrument Arrangement.....](#) 46

## Make Same Size

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To resize the selected instruments in the layout to the width and height of the last selected item.

### Description

The last selected item is marked by a black frame.

### Related topics


HowTos

[How to Optimize the Instrument Arrangement..... 46](#)

## Make Same Width

### Access

You can access this command via:




Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

<b>Purpose</b>	To resize the selected instruments in the layout to the width of the last selected item.
<b>Description</b>	The last selected item is marked by a black frame.
<b>Related topics</b>	<div>HowTos<div>How to Optimize the Instrument Arrangement..... 46</div></div>


## Move Down

<b>Access</b>	<div>You can access this command via:</div> <table><tr><td>Ribbon</td><td>Layouting – Layer</td></tr><tr><td>Context menu of</td><td>None</td></tr><tr><td>Shortcut key</td><td>None</td></tr><tr><td>Icon</td><td></td></tr></table> <div><b>Note</b><div>This command is not available in operator mode.</div></div>	Ribbon	Layouting – Layer	Context menu of	None	Shortcut key	None	Icon	
Ribbon	Layouting – Layer								
Context menu of	None								
Shortcut key	None								
Icon									
<b>Purpose</b>	To move the selected instruments in the layout one layer down.								
<b>Result</b>	The selected instruments are covered by the instruments on higher layers if their borders overlapped before this command was carried out.								
<b>Related topics</b>	<div>HowTos<div>How to Optimize the Instrument Arrangement..... 46</div></div>								

## Move Up

### Access

You can access this command via:

Ribbon	Layouting – Layer
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To move the selected instruments in the layout one layer up.

### Result

The selected instruments will cover the instruments on lower layers if their borders overlapped before this command was carried out.

### Related topics


#### HowTos

[How to Optimize the Instrument Arrangement..... 46](#)

## Open (Layout)

### Access

You can access this command via:

Ribbon	None
Context menu of	<a href="#">Project</a>  controlbar – layout(s) that are currently closed
Shortcut key	None
Icon	None

### Purpose

To open one or more closed layouts.

### Result

The layouts are opened with all its instruments and variable connections.

**Description** Closed layouts are grayed out in the project tree.  
If you want to open a layout which is not part of the active experiment you must import it. Refer to [Import Layout/Import](#) on page 70.

**Related topics**


Basics

[Basics of Handling Layouts.....](#) 10

## Paste / Paste Instruments with Connections

**Access**

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>Layout</li> <li>Layout Navigator - Layout</li> </ul>
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To paste the copied instruments in the current layout with variable connections.

**Result**

The copied instrument is pasted in the current layout with variable connections.

**Related topics**


Basics

[Placing Variables on a Layout.....](#) 24

## Paste Instruments Without Connections

### Access

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>Layout</li> <li>Layout Navigator - Layout</li> </ul>
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To paste the copied instruments in the current layout without variable connections.

### Result

The copied instrument is pasted in the current layout without variable connections.

### Related topics

#### Basics

[Placing Variables on a Layout..... 24](#)


#### HowTos

[How to Copy Variables Between Instruments..... 41](#)

## Remove (Layout)

### Access

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li><b>Project</b>  controlbar – Layout</li> <li>Layout Navigator - Layout</li> </ul>
Shortcut key	<b>Delete</b>
Icon	None



**Note**

This command is not available in operator mode.

**Purpose**

To remove one or more layouts from the experiment.

**Result**

The selected layouts are removed from the experiment. You should save the experiment afterwards.

You will be asked if you also want the selected layouts to be removed from your file system.

**Note**

Removing layouts from your file system cannot be undone.


**Related topics****Basics**

[Basics of Handling Layouts.....](#) 10

## Rename (Layout)

**Access**

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>▪ <a href="#">Project</a>  controlbar – layout</li> <li>▪ Layout Navigator – layout</li> </ul>
Shortcut key	<b>F2</b>
Icon	None

**Note**

This command is not available in operator mode.

**Purpose**

To rename a layout.

**Result**

This will save the layout under the new name.

**Description** The layout must be opened in the active experiment.

**Related topics**

**Basics**

[Basics of Handling Layouts..... 10](#)

## Scroll to Instrument

**Access**

You can access this command via:

Ribbon	None
Context menu of	Instrument Navigator - instrument node
Shortcut key	None
Icon	None

**Purpose**

To scroll the layout to make the selected instrument visible.

**Result**

If necessary, ControlDesk scrolls the layout to make the selected instrument visible.

**Related topics**


**References**

[Instrument Navigator..... 73](#)

## Send to Back

**Access**

You can access this command via:

Ribbon	Layouting – Layer
Context menu of	Instrument Navigator - instrument node
Shortcut key	<b>Ctrl+Shift+B</b>
Icon	

**Note**

This command is not available in operator mode.

---

**Purpose** To send the selected instruments in the layout to the back.

---

**Result** The selected instruments are covered by the instruments in the foreground if their borders overlapped before this command was carried out.

**Related topics****HowTos**


[How to Optimize the Instrument Arrangement..... 46](#)

## Show Disconnected

---

**Access**

You can access this command via:

Ribbon	Layouting - Connections
Context menu of	None
Shortcut key	None
Icon	

---

**Purpose** To highlight the instruments which are not connected with a main variable in the current layout.

---

**Result** Each instrument without a connected main variable is highlighted in the current layout with a red frame.


**Related topics****Basics**

[Basics of Handling Layouts..... 10](#)

## Space Across

### Access

You can access this command via:

Ribbon	Layout – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To equalize the distance of selected instruments in the layout between the left-most and right-most instruments.

### Related topics


#### HowTos

[How to Optimize the Instrument Arrangement.....](#) 46

## Space Down

### Access

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

#### Note

This command is not available in operator mode.

### Purpose

To equalize the distance of selected instruments in the layout between the uppermost and lowest instruments.

**Related topics****HowTos**

[How to Optimize the Instrument Arrangement..... 46](#)

## Switch Data Source

**Access**

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> <li>Layout</li> <li>Layout Navigator - Layout</li> </ul>
Shortcut key	None
Icon	None

**Purpose**

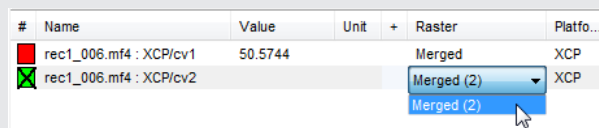
To switch the variable connections of all the instruments of the selected layout to another source.

**Description**

This command provides a submenu that lists up to five open measurement data files and the current measurement if available. The instruments on the layout are connected to the variables in the selected source.

**Note**

If a plotter contains calculated variables and you switch to a data source based on an MF4 file, some signals might be invisible. In contrast to a current measurement, an MF4 file uses incremented numbers to differentiate between identical raster names, for example Merged, Merged(2), Merged(3), .... To make such signals visible again, select the incremented raster name manually in the plotter legend.



#	Name	Value	Unit	+	Raster	Platfo...
	rec1_006.mf4 : XCP/cv1	50.5744			Merged	XCP
	rec1_006.mf4 : XCP/cv2				Merged (2)	XCP

**Related topics****HowTos**

[How to Switch the Data Source of Variable Connections Within a Layout \(ControlDesk Measurement and Recording !\[\]\(eafc244b53721dd1ec133f0772f70fc7\_img.jpg\)\)](#)


**References**

[Create Copy.....63](#)

## Synchronize (Layout)

**Access**

This command is available only if the experiment layout references an external layout that is available in the file system. You can access this command via:

Ribbon	None
Context menu of	<a href="#">Project</a>  controlbar – Experiment Layouts folder
Shortcut key	None
Icon	None

**Note**

This command is not available in operator mode.

**Purpose**

To synchronize the local copy of a layout with the layout it is referenced to.

**Result**

The **Synchronize Data/Files** dialog is opened, where you can export the current layout settings to the referenced external file or import the layout settings.

**Dialog settings**

The dialog displays the name and the date of the last modification of the experiment layout (local working copy file) and the referenced external file.

**Export** Lets you export the settings of the local experiment layout to the referenced external layout.

**Import** Lets you import the settings of the referenced external layout to the local experiment layout.


**Related topics****HowTos**

<a href="#">How to Create a Layout Pool.....</a>	17
<a href="#">How to Export/Import Layouts.....</a>	15

## Ungroup

**Access**

You can access this command via:

Ribbon	Layouting – Arrange
Context menu of	None
Shortcut key	None
Icon	

**Note**

This command is not available in operator mode.

**Purpose**

To ungroup an instrument group.

**Result**

The group is split up into the single instruments. You can handle the instruments separately again.


**Related topics****HowTos**

<a href="#">How to Optimize the Instrument Arrangement.....</a>	46
---	----

## Visualization Page

**Access**

This page is part of the ControlDesk Options dialog.

The dialog can be opened via the **Options Command** (ControlDesk User Interface Handling .

**Purpose**

To specify visualization settings.

**Dialog settings**

**Use display identifier** Indicates whether a variable-specific alias name is used instead of the real variable name.

**Automatically remove unused signals from measurement signal list** Lets you specify whether to remove signals from the measurement signal list in the Measurement Configuration controlbar if they are no longer used on any opened layout or as input variables for a calculated variable or trigger rule.

**Always ask before removing unused signals** Lets you specify whether ControlDesk prompts you to confirm the removing of unused signals in the measurement signal list.

**Export empty connections** Lets you specify to add empty connections to a connection file if no variable connection is defined. This allows you to export/import empty rows of a Variable Array via empty connections, for example.

**Key-based connection import** Lets you specify to import a variable connection only if the exact position of the connected variable in an instrument can be reestablished. If, for example, a variable is connected to the third row of a Variable Array, this connection can only be imported to the third row of an appropriate target Variable Array. If the third row is not available for this variable connection, it is not reestablished.

If key-based connection import is cleared, the connections are imported without regard to the exact position of the variables in an instrument. If necessary, new connections are added to an instrument. Connections that are not part of the connection file are deleted.

**Layout - Custom default background color** Lets you specify to use a custom default color for the layout background. If selected, you can click the color box to choose a background color for newly inserted layouts.

**Related topics****Basics**

[Placing Variables on a Layout..... 24](#)

**HowTos**

[How to Activate Variables for Measurement \(ControlDesk Measurement and Recording !\[\]\(2b17f17ebbacc911bb0ff784ab641779\_img.jpg\)\)](#)

**References**

[Options Command \(ControlDesk User Interface Handling !\[\]\(9db214d549b9aeebe72aa11d3a5c4b1a\_img.jpg\)\)](#)



## Visualize Variables

### Access

In the **Variables** controlbar and in the **Measurement Data Pool**, you can access this command via:

Ribbon	None
Context menu of	Variable list
Shortcut key	<b>Enter</b>
Icon	None

#### Note

This command is not available in operator mode.

### Purpose

To connect variables to instruments and place them on a layout.

### Result

The variable is connected with its default instrument. The default instrument is placed on the currently active layout.

### Description

#### Note

You can place new variables on a layout even during a running measurement, but they will not be measured until the measurement is restarted. ControlDesk therefore asks you if you want to continue the running measurement.

- If you let ControlDesk continue the running measurement, the new variable is not measured until the next restart of the measurement.
- If you let ControlDesk stop and restart the measurement, the new variable is measured. The data in the measurement data buffer is lost.

### Related topics

#### HowTos

- [How to Place Variables on a Layout via the Instrument Selector.....](#) 32
- [How to Place Variables on a Layout via the Variables Controlbar.....](#) 29

#### References

[Measurement Configuration \(ControlDesk Measurement and Recording !\[\]\(19d44b37fb4fa155bf9d60c77a3d3cb2\_img.jpg\)](#))



# Automation

Where to go from here

Information in this section

Programming ControlDesk Automation.....	100
Layout and Instrument Handling.....	110

# Programming ControlDesk Automation

## Where to go from here

## Information in this section

### [Automating Visualization of Variables on Layouts..... 100](#)

Visualizing variables means placing variables on layouts so that they are displayed by instruments.

### [Automating Parameter Calibration..... 107](#)

In ControlDesk, you can calibrate parameters by changing their values in instruments on a layout. ControlDesk offers specialized instruments for editing variable types, for example, the Variable Array for scalar parameters and the Table Editor for nonscalar parameters.

## Information in other sections

### [Tool Automation Demos \(ControlDesk Automation \)](#)

Demonstrate how to automate ControlDesk and use ControlDesk events.

## Automating Visualization of Variables on Layouts

### Introduction

Visualizing variables means placing variables on layouts so that they are displayed by instruments.

Unless otherwise indicated, the program listings below consist of excerpts from the `LayoutAndInstrumentHandling.py` demo script.

### Creating a new layout

The following listing shows how to create a new layout.

```
class MainDemoController(object):
    (...)
    def AddAndRemoveLayout(self):
        (...)
        # Add new Layout.
        newLayout = self.ControlDeskApplication.LayoutManagement.Layouts.Add("NewTestLayout")
```

### Accessing and removing an existing layout

The following listing shows how to access an existing layout and then to remove it from the project and from the hard disk. The parameter of the `Remove` method specifies whether the layout is removed from the project and from the hard disk (`True`) or from the project only (`False`).

```

class MainDemoController(object):
    (...)
    def AddAndRemoveLayout(self):
        (...)
        LAYOUTNAME = "Layout1.lay"
        (...)
        # Check if the layout with the name "Layout1.lay" exists.
        if self.ControlDeskApplication.LayoutManagement.Layouts.Contains(LAYOUTNAME):
            # Get layout with the name "Layout1.lay".
            layout1 = self.ControlDeskApplication.LayoutManagement.Layouts.Item(LAYOUTNAME)
            # Remove layout.
            layout1.Remove(True)

```

## Accessing a layout

You can access layouts in different ways:

**Accessing an open layout** You can access a currently open layout via the `Layouts / IXaLayouts <<Collection>>` collection property of the `LayoutManagement / IXaLayoutManagement <<Interface>>` interface. Closed layouts are *not* included in the `Layouts / IXaLayouts <<Collection>>` collection.

**Accessing a closed layout** You can access a currently closed layout via the `Files / IXaFiles <<Collection>>` collection property of the following interfaces:

- `LayoutManagement / IXaLayoutManagement <<Interface>>`
- `ActiveExperiment / IXaActiveExperiment <<Interface>>`

### Note

When you use the `Files / IXaFiles <<Collection>>` collection property of the `ActiveExperiment / IXaActiveExperiment <<Interface>>` interface, you have access to *all the files of the currently active experiment*, not only to layouts.

## Configuring the properties of a layout

The following listing shows how to specify the name, grid width, grid height, and background color of a layout.

```
# Import: Function RGB
# Converts single values for red, green, blue to corresponding color value.
from DemoUtilities import RGB
(...)
class MainDemoController(object):
    (...)
    def AddAndRemoveLayout(self):
        (...)
        # Add new layout.
        newLayout = self.ControlDeskApplication.LayoutManagement.Layouts.Add("NewTestLayout")
        (...)
        # Rename layout.
        newLayout.Rename("RenamedTestLayout")
        # Change grid width and height.
        newLayout.Grid.Width *= 3
        newLayout.Grid.Height *= 3
        # Change background color.
        newLayout.BackgroundColor = RGB(0,255,255)
```

## Exporting/Importing a layout

The following listing shows how to export the active layout to `ExportLayout.lax` and how to import it.

```
# Import: The os module is used for path concatenations.
import os
# Create new root path in the temporary system folder.
PROJECTROOTPATH = os.path.join(tempfile.gettempdir(), "DemoRoot")
(...)
# Path of the exported layout.
EXPORTPATH = os.path.join(PROJECTROOTPATH, "ExportLayout.lax")
(...)
class MainDemoController(object):
    (...)
    def ExportAndImportLayout(self):
        (...)
        # Export active layout.
        self.ControlDeskApplication.LayoutManagement.ActiveLayout.Export(EXPORTPATH)
        (...)
        # Import layout.
        self.ControlDeskApplication.LayoutManagement.Layouts.Import(EXPORTPATH)
```

## Placing instruments on a layout

The following listing shows how to place three instruments on a layout and then align them.

```
# Name of the Knob instrument.
KNOBINSTRUMENTNAME = "TestKnob"
# Name of the Variable Array instrument.
VARIABLE_ARRAY_INSTRUMENT_NAME = "TestVariableArray"
# Name of the Time Plotter instrument.
PLOTTERINSTRUMENTNAME = "TestPlotter"
(...)
```

```

class MainDemoController(object):
    (...)
    def AddLayoutWithInstruments(self):
        (...)
        # Add Time Plotter instrument.
        plotterInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.\
            Instruments.Add("Time Plotter", PLOTTERINSTRUMENTNAME,
                           0, 0, 300, 200)

        # Add Knob instrument.
        knobInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.\
            Instruments.Add("Knob", KNOBINSTRUMENTNAME,
                           310, 100, 200, 200)

        # Add Variable Array instrument.
        arrayInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.\
            Instruments.Add("Variable Array", VARIABLE_ARRAY_INSTRUMENT_NAME,
                           580, 200, 300, 200)

        (...)
        # Select all three instruments.
        plotterInstrument.SelectMulti()
        knobInstrument.SelectMulti()
        arrayInstrument.SelectMulti()
        # Get instrument selection.
        instrumentSelection = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Selection
        # Align instruments to bottom.
        instrumentSelection.AlignTop()
        # Center instrument selection.
        instrumentSelection.CenterInViewHorizontally()
        (...)
        # Deselect all instruments.
        instrumentSelection.DeselectAll()

```

### Accessing and configuring instruments

The following listing shows how to access a Time Plotter and change its position on the layout.

```

PLOTTERINSTRUMENTNAME = "TestPlotter"
(...)
class MainDemoController(object):
    (...)
    def AddLayoutWithInstruments(self):
        (...)
        # Get the Time Plotter instrument and increment its width.
        if self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.Contains(PLOTTERINSTRUMENTNAME):
            plotterInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.\
                Item(PLOTTERINSTRUMENTNAME)
            # Increment the Plotter instrument's width.
            plotterInstrument.Position.Width += 300
            # Move the Plotter to the left.
            plotterInstrument.Position.X = 0

```

#### Tip

You can get the path of an instrument by dragging it from a layout to the Internal Interpreter or to the Python Editor.

**Adding custom data to an instrument** You can add custom data to an instrument. This allows you to add layout generation information to the instrument, such as connection strings, for example.

The Python listing below shows how to specify custom data:

```
instrument = Application.LayoutManagement.ActiveLayout.Instruments.Item(0)
instrument.CustomData = "My custom data"
```

#### Note

Keep in mind, however, that custom data is available only via automation, not via ControlDesk's Properties controlbar when you select the instrument.

**Clearing item collections of instruments** You can clear item collections of instruments such as captions and bands via the **Clear()** method. This allows you, for example, to easily clear the states of a Selection Box as shown in the following Python listing:

```
instrument = Application.LayoutManagement.ActiveLayout.Instruments.Item(0)
instrument.Items.Clear()
```

The following list shows where the **Clear()** method is available:

- `instrument.Captions.Clear()`
- `ScaleUsing.Bands.Clear()`
- `SelectionBox.Items.Clear()`
- `ButtonInstruments.Buttons.Clear()`
- `MultiStateDisplay.States.Clear()`
- `MultiSwitch.States.Clear()`
- `VariableArray.Rows.Clear()`

## Connecting variables to instruments

The following listing shows how to connect variables to instruments. In the listing, the strings for the assignment of the variable connection are stored in the **VARIABLECONNECTIONPATHTEMPLATES** list. To connect a variable to an instrument, a platform/device with a variable description (or a measurement data file) must be available in the active experiment. Variable connections are shown for a Knob, a Variable Array, a Time Plotter, and a Table Editor.

**VARIABLECONNECTIONPATHTEMPLATES syntax** To connect a variable to an instrument, you have to define a specific connection string.

For each connection to be established, you have to specify:

- The platform/device name (or measurement data file name)
- The (optional) raster name in brackets, followed by `://`
- The unique name of the variable

Examples:

- `VARIABLECONNECTIONPATHTEMPLATES = ["XCP()://DummyOmega"]`
- `VARIABLECONNECTIONPATHTEMPLATES = ["XCP(OnChange)://f_Ki_1"]`
- `VARIABLECONNECTIONPATHTEMPLATES = ["Platform(5ms)://ModelRoot/Scope/In2"]`



**Tip**

You can get the connection path of a variable by selecting Copy in the variable's context menu in the Variables controlbar and pasting it to the Internal Interpreter or to the Python Editor.

```
# Get the root path
SCRIPTPATH = os.path.abspath(os.path.curdir)
if len(sys.argv) > 0:
    SCRIPTPATH = os.path.dirname(sys.argv[0])
# Define CalDemo .a2l image path.
A2LIMAGEPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\\..\\..\\CalDemo\\CalDemo.a2l"))
# Define CalDemo .mot image path.
MOTIMAGEPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\\..\\..\\CalDemo\\CalDemo.mot"))
# A list with string representations of some variables used in this demo.
VARIABLECONNECTIONPATHTEMPLATES = [ "%s()//DummyOmega", "%s(5ms)//control_out",
                                       "%s(OnChange)//f_Ki_1", "%s()//abs_sinp2_cosp2_table",
                                       "%s()//airmass_x_table", "%s()//MeasureVector_MASK_0x0100[2]" ]

# Name of the Knob instrument.
KNOBINSTRUMENTNAME = "TestKnob"
# Name of the Variable Array instrument.
VARIABLE_ARRAY_INSTRUMENT_NAME = "TestVariableArray"
# Name of the Time Plotter instrument.
PLOTTERINSTRUMENTNAME = "TestPlotter"
# Name of the Table Editor instrument
TABLEEDITORINSTRUMENTNAME = "TestTableEditor"
(...)
class MainDemoController(object):
    (...)
    def AddPlatformAndConnectVariableToKnobInstrument(self):
        (...)
        # Get Knob instrument.
        knobInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.Item(
            KNOBINSTRUMENTNAME)
        (...)
        # Add XCP on CAN platform.
        self.DemoPlatform = self.ControlDeskApplication.ActiveExperiment.Platform.Add(
            self.Enums.PlatformType.XCPonCAN)
        # Add variable description.
        self.DemoPlatform.VariableDescriptions.AddWithImage(A2LIMAGEPATH, MOTIMAGEPATH)
        # Connect variable via string assignment with syntax:
        # [MeasurementDataFileName:]DeviceName([RasterName])://UniqueName
        knobInstrument.MainVariable = VARIABLECONNECTIONPATHTEMPLATES[0] % self.DemoPlatform.Name
        (...)
    def EditVariableArray(self):
        (...)
        # Get Variable Array instrument.
        variableArrayInst = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.Item(
            VARIABLE_ARRAY_INSTRUMENT_NAME)
        (...)
        # Add new rows associated with variables.
        for index in range(3):
            # Add new row
            newRow = variableArrayInst.Rows.Add()
            # Add variable to row.
            newRow.MainVariable = VARIABLECONNECTIONPATHTEMPLATES[index] % self.DemoPlatform.Name
        (...)
```

```

def EditPlotter(self):
    (...)
    # Get Time Plotter instrument.
    plotterInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.Item(
        PLOTTERINSTRUMENTNAME)
    (...)
    # Configure x-axis.
    xAxis = plotterInstrument.ActivePlot.ActiveXAxis
    (...)
    # Configure y-axes.
    yAxes = plotterInstrument.ActivePlot.YAxes
    # Add new y-axis.
    newYAxis = yAxes.Add()
    (...)
    # Add new signal to axis.
    newSignal = newYAxis.Signals.Add()
    # Connect variable to signal.
    newSignal.MainVariable = VARIABLECONNECTIONPATHTEMPLATES[2] % self.DemoPlatform.Name
    (...)
def CreateAndEditTableEditor(self):
    (...)
    # Add Table Editor instrument.
    tableEditorInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.Add(
        "Table Editor", TABLEEDITORINSTRUMENTNAME, 0, 0, 500, 500)
    (...)
    # Add sub instrument.
    tableEditorInstrument.SubInstruments.Add()
    # Create the event sink for the table editor.
    tableEditorInstrument = DispatchWithEvents(tableEditorInstrument, TableEditorInstrumentEvents)
    # Connect common axis variable to active sub instrument
    tableEditorInstrument.ActiveSubInstrument.MainVariable = VARIABLECONNECTIONPATHTEMPLATES[4] %
self.DemoPlatform.Name

```

## Grouping layouts

The following listing shows how to group layouts, including the following steps:

1. Multiple layouts are added to the Experiment Layouts folder.
2. Multiple groups are added to the Experiment Layouts folder.
3. One sub group is added to the first group.
4. One layout is added to each group.
5. One layout is added to the sub group.

```

class MainDemoController(object):
    (...)
    def GroupLayouts(self):
        (...)
        # Add some layouts to the experiment.
        for index in range(1, 7):
            # Add new layout.
            newLayout = self.ControlDeskApplication.LayoutManagement.Layouts.Add("NewTestLayout_%d" % index)
            newLayout.Window.State = self.Enums.WindowState.Maximized
        (...)
        # Get default folder for layouts.
        defaultFolderLayouts = self.ControlDeskApplication.ActiveExperiment.GetDefaultFolder(self.Enums.FileType.Layout);
        for index in range(1, 5):
            # Add new group.
            defaultFolderLayouts.Groups.Add("Group_%d" % index)

```


```
# Add a sub group to the first group.
subGroup = defaultFolderLayouts.Groups[0].Groups.Add("SubGroup")
(...)
# Get the first three layout files.
layoutFiles = []
for currentLayoutFile in defaultFolderLayouts.Files:
    layoutFiles.append(currentLayoutFile)
# Insert one layout in all groups.
for index in range(3):
    # Insert the layout files.
    defaultFolderLayouts.Groups[index].Files.Insert(layoutFiles[index])
# Insert last layout to the sub group.
subGroup.Files.Insert(layoutFiles[len(layoutFiles) - 1])
(...)
```

**3-D Viewer: Adding roads via MAT file import**

The automation interface of the 3-D Viewer lets you import complete road networks from MAT files created with the ModelDesk Road Generator.

Refer to [Adding Roads via MAT File Import \(ControlDesk Instrument Handling !\[\]\(4729e517bc6a7cd81c8025b9646574fb\_img.jpg\)](#)).

**Related topics**

Basics	
<a href="#">Basics of Placing Variables on a Layout.....</a>	24
<a href="#">Tool Automation Demos (ControlDesk Automation </a> )	

# Automating Parameter Calibration

**Introduction**

In ControlDesk, you can calibrate parameters by changing their values in instruments on a layout. ControlDesk offers specialized instruments for editing variable types, for example, the Variable Array for scalar parameters and the Table Editor for nonscalar parameters.

**Activating working and reference data sets**

The following listing shows how to make the third data set of the active experiment the working data and the second data set the reference data set (excerpts from the `DataSetHandling.py` demo script).

```

class MainDemoController(object):
    (...)
    def Initialize(self):
        (...)
        # Start ControlDesk.
        self.ControlDeskApplication = Dispatch("ControlDeskNG.Application")
        # Define Enums object.
        self.Enums = Enums(self.ControlDeskApplication)
        # Add XCP on CAN platform.
        self.DemoPlatform = self.ControlDeskApplication.ActiveExperiment.Platforms.Add(
            self.Enums.PlatformType.XCPonCAN)
        (...)
    def DefineWorkingAndReferenceSet(self):
        # Get the last created data set
        newDataSet = self.DemoPlatform.ActiveVariableDescription.DataSets.Item(2)
        # Set data set to working set
        newDataSet.MakeWorking()
        # Activate the working data set.
        newDataSet.Activate()
        # Set the second data set as the reference data set.
        self.DemoPlatform.ActiveVariableDescription.DataSets.Item(1).MakeReference()

```

## Changing parameter values

The following listing shows how to change the values of parameters that are connected to a Variable Array (excerpts from the `LayoutAndInstrumentHandling.py` demo script).

```

# Get the root path
SCRIPTPATH = os.path.abspath(os.path.curdir)
if len(sys.argv) > 0:
    SCRIPTPATH = os.path.dirname(sys.argv[0])
# Define CalDemo .a2l image path.
A2LIMAGEPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\\..\\..\\CalDemo\\CalDemo.a2l"))
# Define CalDemo .mot image path.
MOTIMAGEPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\\..\\..\\CalDemo\\CalDemo.mot"))
(...)
# A list with string representations of some variables used in this demo.
VARIABLECONNECTIONSTRINGS = [ "%s()//DummyOmega", "%s(5ms)//control_out",
                                "%s(OnChange)//f_Ki_1", "%s()//abs_sinp2_cosp2_table",
                                "%s()//airmass_x_table", "%s()//MeasureVector_MASK_0x0100[2]" ]
(...)
# Name of the Variable Array instrument.
VARIABLE_ARRAY_INSTRUMENT_NAME = "TestVariableArray"
(...)
class MainDemoController(object):
    (...)
    def AddPlatformAndConnectVariableToKnobInstrument(self):
        (...)
        # Add XCP on CAN platform.
        self.DemoPlatform = self.ControlDeskApplication.ActiveExperiment.
            Platforms.Add(self.Enums.PlatformType.XCPonCAN)
        self.DemoPlatform.VariableDescriptions.AddWithImage(A2LIMAGEPATH, MOTIMAGEPATH)
        (...)

```

```
def EditVariableArray(self):
    # Get Variable Array instrument.
    variableArrayInstrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.Item(
        VARIABLE_ARRAY_INSTRUMENT_NAME)
    (...)
    # Add new rows associated with variables.
    for index in range(3):
        # Add new row
        newRow = variableArrayInstrument.Rows.Add()
        # Add variable to row.
        newRow.MainVariable = VARIABLECONNECTIONSTRINGS[index] % self.DemoPlatform.Name
        # Set value.
        newRow.Value = 3.0 * index
```

### Note

When you try to set the value of a parameter outside the parameter's value range, these changes are discarded without notice. However, you can implement a notification yourself by reading back the parameter value from the instrument.

This applies only when you calibrate a parameter *by changing its value in an instrument on a layout*. It does not apply when you use the *direct variable access feature* (refer to [Automating Direct Variable Access \(ControlDesk Automation !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)](#))).

For information on parameter range limits, refer to [Handling Parameter Range Limits \(ControlDesk Calibration and Data Set Management !\[\]\(74d4806277d7e73349d8e8c0897931e9\_img.jpg\)](#)).

## Related topics

### Basics

[Basics of Calibrating Parameters \(ControlDesk Calibration and Data Set Management !\[\]\(0b5e7e25e8775f7e7e80906ada4f0021\_img.jpg\)](#))  
[Tool Automation Demos \(ControlDesk Automation !\[\]\(740312fd467f47b04cab841ab3868d83\_img.jpg\)](#))

# Layout and Instrument Handling

## Layout- and Instrument-Related Interfaces

### Introduction







The main tasks of ControlDesk's layout and instrument handling are to create and open layouts, create and configure instruments and connect variables to the layouts.

### Description

The Layout Management object implements the *IXaLayoutManagement* interface. The properties of this interface can be used to manipulate the settings of the Layout Management.

A layout document implements the *IXaLayoutDocument* interface. It allows you to get access to the properties of the layout. The *Instruments* property implements the *IViTopLevelInstruments* interface. It allows you to get a collection of all instruments placed on the layout.

### Related interfaces

Interface	Description
IXaLayoutManagement (refer to <a href="#">LayoutManagement / IXaLayoutManagement &lt;&lt;Interface&gt;&gt;</a> (ControlDesk Automation 	This interface is to access visualization including layouts and instruments.
IXaLayoutDocument (refer to <a href="#">LayoutDocument / IXaLayoutDocument &lt;&lt;Interface&gt;&gt;</a> (ControlDesk Automation 	This interface is to access a single currently open layout document.
IViTopLevelInstruments (refer to <a href="#">TopLevelInstruments / IViTopLevelInstruments &lt;&lt;Interface&gt;&gt;</a> (ControlDesk Automation 	Interface for the top level instruments collection.
IViInstrumentLibraries (refer to <a href="#">InstrumentLibraries / IViInstrumentLibraries &lt;&lt;Collection&gt;&gt;</a> (ControlDesk Automation 	Automation interface for instrument libraries.
IViInstrumentDefaults (refer to <a href="#">InstrumentDefaults / IViInstrumentDefaults &lt;&lt;Interface&gt;&gt;</a> (ControlDesk Automation 	Automation interface for instrument defaults.
IXaLayouts (refer to <a href="#">Layouts / IXaLayouts &lt;&lt;Collection&gt;&gt;</a> (ControlDesk Automation 	This interface is to access the list of currently open layout documents.

---

**Related documentation**

Topic	Description
<a href="#">Automating Visualization of Variables on Layouts</a> on page 100	Visualizing variables means placing variables on layouts so that they are displayed by instruments.





# Limitations

## Limitations for Layouting

Introduction

There is a limitation for renaming layouts in ControlDesk.

Renaming layouts

If you rename a ControlDesk layout, you must save the related experiment before closing it or activating another experiment. If you do not save the experiment after you rename an included layout, the experiment cannot find the layout the next time you open the experiment. The connection to the renamed layout is lost.

Visualizing array and struct elements with ambiguous names

If the names of an array or struct element and of a scalar variable are ambiguous, and when you place such an element on an instrument, ControlDesk actually connects the scalar variable with the same name to the instrument. In other words, you cannot connect individual array or struct elements to an instrument if the element names are ambiguous.

**Example** The variable description contains the following variables:

- The **DiagValues** [measurement array](#) with the six elements **DiagValues[0] ... DiagValues[5]**
- The **DiagValues[3]** [scalar measurement variable](#)

When you place the **DiagValues[3]** measurement array element individually on an instrument, ControlDesk actually connects the **DiagValues[3]** scalar measurement variable, not the array element.

Related topics

Basics

<a href="#">Basics of Handling Layouts</a>	10
--	----



# 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.....	116
A.....	116
B.....	117
C.....	118
D.....	122
E.....	126
F.....	129
G.....	130
H.....	130
I.....	131
K.....	133
L.....	133
M.....	134
N.....	137
O.....	137
P.....	139
Q.....	141
R.....	142

S.....	143
T.....	146
U.....	147
V.....	148
W.....	150
X.....	151

## 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\<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](#) <sup>U</sup>s connected to variables of one or more simulation models.

**Layout Navigator** A [controlbar](#) <sup>U</sup> that displays all opened [layout](#) <sup>U</sup>s. It can be used for switching between layouts.

**Layout script** A Python script used to extend the functionality of a [layout](#) <sup>U</sup>.

**Leading raster** The [measurement raster](#) <sup>U</sup> that specifies the [trigger](#) <sup>U</sup> settings for the [Time Plotter](#) <sup>U</sup> 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](#) <sup>U</sup> 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 8, 121
- configuring the properties of a layout 12
- ControlDesk Options dialog
  - Visualization page 95
- copying variables between instruments 41
- creating a new layout 12
- creating layout pools 17
- customizing the connection assignment of variables to instruments 36

**D**

- Documents folder 8, 125

**E**

- exporting layouts 15, 21

**I**

- importing layouts 15, 21
- Instrument Navigator command 73
- Instrument Selector 132
- instruments
  - optimizing the arranging of instruments 46

**L**

- Layout Navigator command 76
- Local Program Data folder 8, 133

**M**

- Measurement Data Pool 135
- Messages controlbar 136

**O**

- optimizing the arranging of instruments 46

**P**

- placing variables on a layout via the Instrument Selector 32
- placing variables on a layout via the Variables Controlbar 29
- Platforms/Devices controlbar 140
- Project controlbar 141
- Project Manager 141
- Properties controlbar 141

**V**

- visualization
  - configuring the properties of a layout 12
  - copying variables between instruments 41
  - creating a new layout 12
  - creating layout pools 17
  - customizing the connection assignment of variables to instrument 36
  - exporting layouts 15
  - exporting variable connections 21

- importing layouts 15
- optimizing the arranging of instruments 46
- placing variables on a layout via the Instrument Selector 32
- placing variables on a layout via the Variables Controlbar 29

