ControlDesk

Variable Management

For ControlDesk 7.4

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

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

How to Contact dSPACE Support

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

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

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

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

Software Updates and Patches

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

Important Notice

Contents

About This Document 7

Basics and Instructions 9

Introduction to Variable Descriptions ... 10
- Variable Descriptions Supported by ControlDesk ... 10
- Handling Variable Descriptions ... 15
- Basics on Importing A2L Files of Version 1.7 ... 16
- How to Add a Variable Description to a Platform/Device ... 20
- Basics on Variable Types ... 24
- Accessing 64-Bit Integer Variables ... 27
- Ethernet Bus Monitoring Device: Information Required in the Variable Description File ... 29

Handling Variables in the Variables Controlbar ... 30
- Basics of the Variables Controlbar ... 30
- How to Filter the Variable List ... 33
- How to Use the Function Buttons for Searching and Filtering Variables ... 37
- How to Search for a Variable Using Incremental Search ... 41

Source and Converted Mode ... 43
- How to Display a Value in Converted or Source Mode ... 43
- Basics on Variables Using Conversion Tables ... 44

Defining Calculated Variables ... 49

Introduction to Calculated Variables ... 49
- Basics on Defining Calculated Variables ... 49
- Basics of Computing Calculated Variables in ControlDesk ... 51
- How to Define a Calculated Variable ... 55
- How to Write Calculated Variable Values to Parameters and Writable Measurements ... 60

Defining Formulas for Calculated Variables ... 63
- Basics of Formulas for Calculated Variables ... 63
- How to Define a Formula for Calculated Variables ... 67
- Examples of Defining a Formula for Calculated Variables ... 71

Exporting and Importing ... 78
How to Export and Import Calculated Variables ... 78
How to Export and Import Template Formulas for Calculated Variables ... 80

Reference Information 83

Variable Management Properties ... 84
Properties of Filter Conditions in the Edit Filter Dialog ... 84
Properties of Variable Descriptions in the Project Controlbar ... 85
Properties of Variables (Variables Controlbar) ... 86

Variable Management Commands ... 91
Activate (Variable Filter) ... 93
Activate Variable Description ... 94
Add to Measurement Signal List ... 96
Add Variable Description ... 97
Add Variable to Favorites List ... 100
Add/Edit Combined Filter – Filter 1-n / Edit (Variable Filter) ... 101
Add/Edit Combined Filter – New / Create Variable Filter ... 102
Apply Combined Filter – Filter 1-n ... 103
Calculated Variables - Copy ... 104
Calculated Variables - Edit ... 105
Calculated Variables - Export Selected Items ... 106
Calculated Variables - Import ... 107
Calculated Variables - New ... 108
Calculated Variables - Remove Selected Items ... 109
Case-Sensitive Search ... 110
Collapse All ... 111
Customize Columns (Variable List) ... 111
Deactivate (Variable Filter) ... 113
Edit Filter Dialog ... 114
Expand All ... 115
Favorite List ... 115
Navigate to Referenced Variable ... 116
No Filter ... 117
Properties (Variable Description) ... 118
Properties for Calculated Variables Dialog ... 118
Reload (Variable Description) ... 125
Remove (Variable Filter) ... 127
Remove All (Favorites List) ... 128
Remove from Measurement Signal List ... 128
Remove Variable Description ... 129

Remove Variable from Favorites List..... 130
Rename (Variable Filter)..... 130
Replace (Variable Description)..... 131
Save Connected Variables in Label List..... 133
Select Variable Dialog..... 133
Standard Filter..... 135
Tree View..... 137
Uncheck All..... 138
Variable Properties / Properties..... 138
Variables..... 142
Variables Page..... 146
Wildcard Filter..... 150

Automation 153

Programming ControlDesk Automation..... 154
Automating Platform Management and Variable Management..... 154
Automating Direct Variable Access..... 160

Variable Description Handling..... 167
Variable Description-Related Interfaces..... 167

Variable Access Handling..... 169
Variable Access Handling Interfaces..... 169
Global Variable-Related Interfaces..... 169
Variable Group-Related Interfaces..... 171

Troubleshooting 175

Variables Removed from the Measurement Signal List When Activating Another Variable Description..... 175
Problems with A2L Files..... 176
Problems with AML Files..... 178

Limitations 185

General Limitations for Variable Management..... 185
Limitations for SDF Files..... 186
Limitations for A2L Files..... 189
Limitation for DBC Files..... 194
Limitations for Calculated Variables..... 195

Glossary 197

Index 235

About This Document

Content

This document introduces you to ControlDesk's variable management.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	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.

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

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

Basics and Instructions

Where to go from here

Information in this section

Introduction to Variable Descriptions.....10
Handling Variables in the Variables Controlbar.....30
Source and Converted Mode.....43
Defining Calculated Variables.....49

Introduction to Variable Descriptions

Where to go from here

Information in this section

Variable Descriptions Supported by ControlDesk 10
A variable description specifies the variables of a platform/device. It also specifies the interface used to connect the platform/device.

Handling Variable Descriptions 15
Gives an overview of handling variable descriptions in a ControlDesk project.

Basics on Importing A2L Files of Version 1.7 16
ControlDesk supports the import of A2L files of version 1.7. There are some specifics.

How to Add a Variable Description to a Platform/Device 20
To access the variables and parameters of dSPACE real-time hardware, a VPU, or an ECU, ControlDesk must know the corresponding variable description.

Basics on Variable Types 24
A table lists all the variable types that ControlDesk displays in variable lists.

Accessing 64-Bit Integer Variables 27
Gives an overview of accessing 64-bit integer variables in ControlDesk.

Ethernet Bus Monitoring Device: Information Required in the Variable Description File 29
The variable description to be added to an Ethernet Bus Monitoring device must contain specific information.

Variable Descriptions Supported by ControlDesk

Introduction

A variable description specifies the variables of a platform/device. It also specifies the interface type used to connect the platform/device and how the interface is to be configured.

ControlDesk supports different types of variable descriptions. What variable description type is used depends on the type of the platform/device.

A2L files

An A2L variable description file contains the description of the variables available for measurement, calibration, and stimulation on an ECU in the standardized ASAM MCD-2MC format. It also describes the interface used to connect to the ECU. For details on the ASAM MCD-2MC format, refer to http://www.asam.net.

A2L files are typically provided by the vendor of the ECU software.

Interface description data Each ECU calibration or bypass interface used to connect the ECU has its own parameters, which must be specified in the ECU's A2L file. For information on this interface description data (`IF_DATA`), and the data format the interface-specific information must comply with, refer to Basics on Interface Description Data (Interface Description Data Reference 🕮).

Data format of interface description data (AML specification) The interface-specific information in the `IF_DATA` entry must comply with a particular format, which is specified in ASAM MCD-2 MC Metalanguage (AML). The AML specification must also be contained in the ECU's A2L file.

dSPACE provides AML files that you can include in the A2L file of your ECU. These files contain the AML specification you need for calibrating or measuring your ECU or RapidPro system via ControlDesk platforms/devices.

You will find AML files in the following folder: `.Main\bin\a2l\aml` in your ControlDesk installation

Related platforms/devices ControlDesk supports A2L files in connection with the following measurement and calibration devices:

- CCP Device
- DCI-GSI2 Device
- XCP on CAN Device
- XCP on Ethernet Device

Supported standards ControlDesk supports the following versions of the ASAM MCD-2MC standard:

- 1.4
- 1.5
- 1.6
- 1.7

  For details on importing version 1.7 A2L files, refer to Basics on Importing A2L Files of Version 1.7 on page 16.

If you encounter any problems when you add an A2L file to a device, refer to Problems with A2L Files on page 176 and Problems with AML Files on page 178, or inquire at support@dspace.de for details.

For information on limitations, refer to Limitations for A2L Files on page 189.

ECU Image file An ECU Image file contains the data of the parameters in the ECU application, and optionally the code of the ECU application. It may be stored as a HEX, MOT, or S19 file. When you add a measurement and calibration device to an experiment, you can also add an ECU Image file to that device. If you do so, ControlDesk initially fills the mirrored memory with the contents of that ECU Image file. ControlDesk can handle ECU Image files that contain code and data or only data.

Tip

If the A2L file contains the EEPROM (**EPK**) identifier, ControlDesk uses the identifier for checking consistency. Refer to Consistency Checks (EPK Checks) (ControlDesk Platform Management 🕮).

AUTOSAR system description files

Related platforms/devices ControlDesk supports AUTOSAR system description (ARXML) files according to the AUTOSAR system template in connection with the following devices:

- CAN Bus Monitoring Device
- Ethernet Bus Monitoring Device

Note

The variable description to be added to an Ethernet Bus Monitoring device must contain specific information. Refer to Ethernet Bus Monitoring Device: Information Required in the Variable Description File on page 29.

- LIN Bus Monitoring Device

Supported standards ControlDesk supports the following standard versions of the AUTOSAR Classic Platform (CP) system template:

- CP R19-11
- CP 4.4
- CP 4.3.1
- CP 4.3.0
- CP 4.2.2
- CP 4.2.1
- CP 4.1.1 ... 4.1.3
- CP 4.0.3
- CP 3.2.1 ... 3.2.3
- CP 3.1.4

DBC files

Related platforms/devices ControlDesk supports DBC files in connection with the following device:

- CAN Bus Monitoring Device

For information on limitations, refer to Limitation for DBC Files on page 194.

FIBEX files

Related platforms/devices ControlDesk supports FIBEX files in connection with the following devices:

- CAN Bus Monitoring Device
- LIN Bus Monitoring Device

Supported file versions ControlDesk supports the following file versions:

- Version 4.1.0, 4.1.1, 4.1.2
- Version 3.1.0, 3.1.1
- Version 3.0.0

LDF files

Related platforms/devices ControlDesk supports LDF files in connection with the following device:

- LIN Bus Monitoring Device

Supported protocol versions ControlDesk supports the following protocol versions:

- 2.1
- 2.0
- 1.3

ODX database files

Open Diagnostic Data Exchange (ODX), the ASAM MCD-2 D (ISO 22901) standard, describes communication with ECUs via diagnostic interfaces. ControlDesk's diagnostic support is based on ODX database files.

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).

Related platforms/devices ControlDesk supports ODX database files in connection with the following device:

- ECU Diagnostics Device

Supported standards ControlDesk supports the following ODX database standards:

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

Variable description for the ECU Diagnostics device ControlDesk lets you generate a variable description for an ECU Diagnostics device from the related ODX database. The variable description of an ECU Diagnostics device specifies the measurement variables and parameters of an ECU which can be accessed via ControlDesk's diagnostics interface.

SDF files

System description (SDF) files specify dSPACE real-time and offline simulation applications, and their parameters and measurement variables. An SDF file bundles TRC files and additional information for the application. The SDF file references the relevant TRC files, which contain the variable descriptions. An SDF file is created automatically when you build the application.

Variable descriptions without initial parameter values Variable descriptions of platforms often do not contain initial parameter values. You therefore cannot calibrate the parameters of the application for these platforms as long as ControlDesk is offline.

Parameter calibration is possible only if you start online calibration: ControlDesk then uploads the parameter values from the dSPACE real-time hardware or VEOS and lets you change the values.

Applications without a variable description (Applicable to the SCALEXIO and VEOS platforms only) In some cases, there is no variable description for an application. For example, if the offline simulation application to be loaded to VEOS contains does not contain an environment VPU, there is no SDF file related to that application.

In this case, you can add the application directly to the platform in the experiment. This allows you to let ControlDesk automatically load and start the offline simulation application when you start online calibration or start measuring. Refer to Real-Time Application / Offline Simulation Application - Add to Experiment (ControlDesk Platform Management 🕮).

Variable description for handcoded applications To experiment with a *handcoded application*, you have to edit the application's SDF file yourself.

For information on the SDF file syntax, refer to SDF File Syntax (RTI and RTI-MP Implementation Reference 🕮).

Related platforms/devices ControlDesk supports SDF files in connection with 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 (ControlDesk Platform Management 🕮)
- Multiprocessor System Platform
- SCALEXIO Platform
- VEOS Platform

For information on limitations, refer to Limitations for SDF Files on page 186.

Related topics

Basics

Handling Variable Descriptions 15

HowTos

How to Add a Variable Description to a Platform/Device 20

References

Limitation for DBC Files 194
Limitations for A2L Files 189
Limitations for SDF Files 186

Handling Variable Descriptions

Introduction

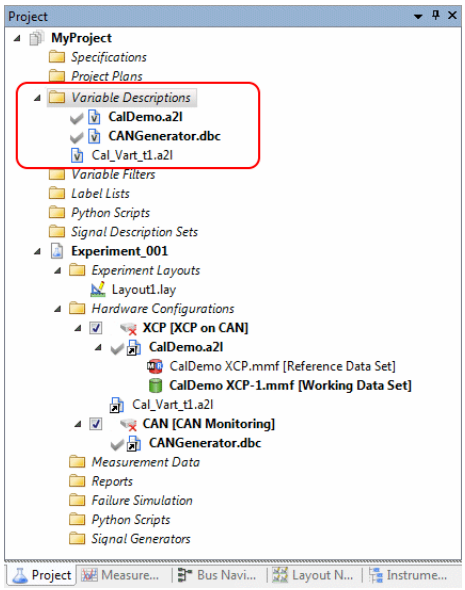
Gives an overview of handling variable descriptions in a ControlDesk project.

Adding variable descriptions

For instructions on adding variable description to a platform/device, refer to How to Add a Variable Description to a Platform/Device on page 20.

Project-global variable description

In ControlDesk, variable descriptions are managed project-globally. All the variable descriptions used in the experiments of a ControlDesk project are stored in the Variable Descriptions folder of the project.



Each time you assign a variable description to a platform/device, you actually assign a reference to the project-global variable description to the platform/device. In the **Project** controlbar, a reference to a project-global variable description is marked by an additional symbol.

Active variable description

You can manage multiple variable descriptions for one platform/device in ControlDesk, but only one of them can be active at a time. In the **Project** controlbar, all the variable descriptions used in a ControlDesk project are stored in the **Variable Descriptions** folder of the project. A variable description that is assigned to a platform/device is listed under the platform/device node as a reference to one of these project-global variable descriptions. The currently active variable description of a platform/device is displayed with a symbol.

If one or more processors in a DS1006-based multiprocessor system application are disabled, the variable descriptions of the disabled processors are grayed out in the **Project** controlbar. In the **Variables** controlbar, variable descriptions of disabled processors are unavailable. For details, refer to Working with Multiprocessor Systems with Optional Processors (ControlDesk Platform Management).

Related topics

Basics

Variable Descriptions Supported by ControlDesk 10

HowTos

How to Add a Variable Description to a Platform/Device 20

Basics on Importing A2L Files of Version 1.7

Introduction

ControlDesk supports the import of A2L files of version 1.7. There are some specifics.

Structures

Version 1.7 of the ASAM MCD-2 MC standard introduces the keywords `TYPEDEF_<xyz>` and `STRUCTURE_COMPONENT` to define structures in a variable description. A structure 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/or struct arrays as elements.

Type definitions

Type definitions are templates of objects and contain default settings for an object.

- Type definitions can be used to build parameters or measurement variables by instantiating them. This allows you to create various objects of the same type with common properties. This minimizes duplications of redundant information.
- Type definitions can be used to build structures by referencing type definitions in structure components. This allows you to create single objects consisting of various different parameters or measurement variables.

Overview of new keywords

The following table lists important keywords that are introduced in version 1.7 of the ASAM MCD-2 MC standard and keywords whose processing in ControlDesk has changed.

Keyword	General Description	Supported	Usage in ControlDesk
`ADDRESS_TYPE`	The type of ECU determines which address width can be defined.	Yes	In version 1.7, `ADDRESS_TYPE` is also supported for measurements and structs.
`ASAP2_VERSION`	The ASAM MCD 2-MC version. This entry is mandatory as of version 1.6.1.	Yes	ControlDesk evaluates the version entry if available. ControlDesk does not reject the import on the basis of a specific or missing version entry. However,the specific ASAM MCD 2-MC version plays a key role in the correct processing of some keywords, such as MATRIX_DIM. Refer to Conflicts if ASAP2_VERSION is not defined on page 19.
`AXIS_DESCR`	Axis description within an adjustable object. As of version 1.7, array elements can also be used as the input quantity.	Partly	Array elements cannot be used as the input quantity.
`BLOB`	New variable type. A binary large object (`BLOB`) is an array of bytes with no further semantic interpretation, and no conversion method or record layout.	No	
`BYTE_ORDER`	Either the Motorola or Intel format or a mix of both can be specified as the byte order.	Partly	Mixed byte orders (`MSB_FIRST_MSW_LAST`, `MSB_LAST_MSW_FIRST`) are not supported.
`CONSISTENT_EXCHANGE`	Indicates that a structure is to be handled as one complete object.	No	This keyword is ignored. The structure is imported.
`CONVERSION`	This keyword is related to `OVERWRITE` (not supported).	No	
`ENCODING`	As of version 1.7, strings are also supported.	No	Processing stays the same.
`INPUT_QUANTITY`	This keyword is related to `OVERWRITE` (not supported).	No	

Keyword	General Description	Supported	Usage in ControlDesk
INSTANCE	Instantiates an object that is specified via a type definition, such as TYPEDEF_STRUCTURE.	Partly	Only the following variable types can be instantiated: ▪ String ▪ Value (scalar parameter) ▪ Measurement ▪ Measurement array ▪ Struct ▪ Struct array
LIMITS	This keyword is related to OVERWRITE (not supported).	No	
MATRIX_DIM	Describes the dimensions of multi-dimensional objects. As of version 1.7, the interpretation has changed.	Partly	Only the following variable types are supported: ▪ Value blocks (with a maximum of 2 dimensions) ▪ Measurement array (with a maximum of 3 dimensions) ▪ Struct array (with a maximum of 3 dimensions) Refer to Conflicts if ASAP2_VERSION is not defined on page 19.
MODEL_LINK	Can be used to reference a software model.	No	
NUMBER	Number of available bytes in ECU memory for a CHARACTERISTIC of the type ASCII.	Yes	As introduced with version 1.6, NUMBER should not to be used with objects other than strings. For other objects use MATRIX_DIM instead. Refer to Conflicts if ASAP2_VERSION is not defined on page 19.
NO_AXIS_PTS_X/ _Y/ _Z/ _4/ _5	Description of the axis point count of a curve, map, cube, or value block.	Partly	Only curve and map are supported. Cube and value block are not supported. Processing stays the same.
OVERWRITE	Can be used at INSTANCE to overwrite a property specified in the referenced type definition.	No	
STATIC_ADRESS_OFFSETS	Indicates that the start addresses of axes and function values do not change when removing or inserting axis points.	No	
STRUCTURE_COMPONENT	Specifies the component of a structure by referencing a type definition. Components that reference measurement objects must not be mixed with components that reference adjustable objects.	Partly	Only the following variable types are supported. ▪ String ▪ Value (scalar parameter) ▪ Value block (with a maximum of 2 dimensions) ▪ Measurement ▪ Measurement array (with a maximum of 3 dimensions) ▪ Struct

Keyword	General Description	Supported	Usage in ControlDesk
			▪ Struct array
THIS	This keyword is related to OVERWRITE (not supported).	No	
TRANSFORMER TRANSFORMER_IN_OBJECTS TRANSFORMER_OUT_OF_OBJECTS	These keywords specify the transformation of variables.	No	
TYPEDEF_AXIS	Type definition of an AXIS_PTS object (axis points). This type definition can be used as a template for AXIS_PTS objects via INSTANCE or referenced in a structure via TYPEDEF_STRUCTURE.	Partly	Instantiated axis objects can be referenced in an axis description (AXIS_DESCR) for a common axis (COM_AXIS). Due to the limitations of STRUCTURE_COMPONENT, TYPEDEF_AXIS cannot be referenced in structures.
TYPEDEF_BLOB	Type definition of a binary large object object (BLOB). This type definition can be used as a template for BLOB objects via INSTANCE or referenced in a structure via TYPEDEF_STRUCTURE.	No	
TYPEDEF_CHARACTERISTIC	Type definition of a CHARACTERISTIC object. This type definition can be used as a template for CHARACTERISTIC objects via INSTANCE or referenced in a structure via TYPEDEF_STRUCTURE.	Partly	Only the following variable types are supported: ▪ String ▪ Value (scalar parameter) ▪ Value block (with a maximum of 2 dimensions)
TYPEDEF_MEASUREMENT	Type definition of a MEASUREMENT object. This type definition can be used as a template for MEASUREMENT objects via INSTANCE or referenced in a structure via TYPEDEF_STRUCTURE.	Partly	Only the following type definitions are supported: ▪ Measurement variable (with a maximum of 3 dimensions)
TYPEDEF_STRUCTURE	Type definition of a STRUCTURE object. The components of a structure are specified by referencing type definitions via STRUCTURE_COMPONENTs.	Yes	Due to the limitations for STRUCTURE_COMPONENT,only components that reference the following variable types are supported: ▪ String ▪ Value (scalar parameter) ▪ Value block (with a maximum of 2 dimensions) ▪ Measurement ▪ Measurement array (with a maximum of 3 dimensions) ▪ Struct ▪ Struct array

Conflicts if ASAP2_VERSION is not defined

ControlDesk evaluates the version of ASAM MCD-2 MC file via the ASAP2_VERSION keyword, if it is available in the variable description. ControlDesk does not reject the import on the basis of a specific or missing version entry. However, the specific ASAM MCD 2-MC version plays a key role in the correct processing of some keywords.

MATRIX_DIM The evaluation of the `MATRIX_DIM` keyword differs according to the ASAM MCD 2-MC version that is used.

- Version 1.6 and earlier:

  This keyword is supported only for value blocks with a maximum of 2 dimensions. However, the third dimension (`zDim`) must be always be 1, see the following example:

  ```
  MATRIX_DIM 5 2 1
  ```

  If both keywords, `MATRIX_DIM` and `NUMBER`, are defined for a value block but the values are not compatible, the `MATRIX_DIM` values are used (a warning is displayed).
- Version 1.7:

  This keyword is used for value blocks, measurement arrays, and struct arrays.

  If this keyword is defined for a variable type with a maximum of 2 dimensions, do not set more than 2 dimensions. In contrast to version 1.6, you must not set the parameter of the third dimension (`zDim`) to 1, see the following example:

  ```
  MATRIX_DIM 5 2
  ```

  The `NUMBER` keyword should not to be used with objects other than strings.

NUMBER

- Version 1.6 and earlier:

  The `NUMBER` keyword can be used to define the number of values or parameters of a value block or a string. As of version 1.6, it is recommended to use `MATRIX_DIM` for value blocks.
- Version 1.7:

  The `NUMBER` keyword should not be used with objects other than strings.

Related topics

HowTos

How to Add a Variable Description to a Platform/Device 20

References

Limitations for A2L Files 189

How to Add a Variable Description to a Platform/Device

Objective

To access the variables and parameters of dSPACE real-time hardware, a VPU, or an ECU, ControlDesk must know the corresponding variable description.

Consistency check of A2L and ECU Image file

Whenever you add an A2L file and an ECU Image file to a platform/device, ControlDesk automatically checks whether the files are consistent. Refer to Consistency Checks (EPK Checks) (ControlDesk Platform Management 🕮).

Variable descriptions

For platforms, the variable description also specifies the corresponding real-time application or VPU. For further information, refer to Basics on Handling Simulation Applications (ControlDesk Platform Management 🕮) and Basics on Offline Simulation Applications (ControlDesk Platform Management 🕮).

Restrictions

- You cannot reload or replace a variable description if the variable description is added to platforms/devices in multiple experiments of a project.

  As a workaround, add a new variable description to a platform/device in the affected experiment of a project.
- Variable descriptions of platforms often do not contain initial parameter values. You therefore cannot calibrate the application parameters for these platforms if online calibration has not started yet. Refer to No offline calibration if variable description has no initial data set (ControlDesk Platform Management 🕮).
- The instructions below do not apply to ECU Diagnostics device. The variable description of an ECU Diagnostics device is generated from the ODX database during device configuration. Refer to How to Configure an ECU Diagnostics Device (ControlDesk Platform Management 🕮).

Preconditions

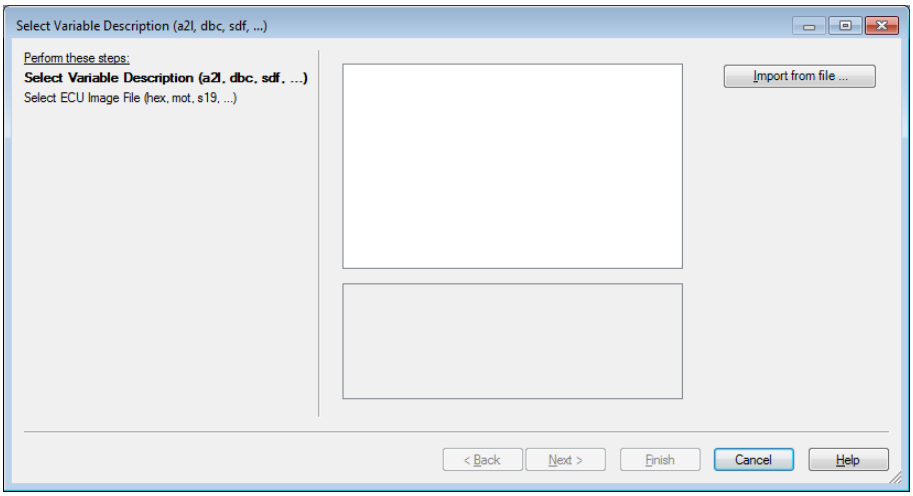
- The platform/device must already be available in the experiment. Refer to How to Add a Platform/Device to an Experiment (ControlDesk Platform Management 🕮).
- Online calibration must be stopped for the selected platform/device. Refer to Go Offline (ControlDesk Calibration and Data Set Management 🕮) or Stop Online Calibration (for Single Platform/Device) (ControlDesk Calibration and Data Set Management 🕮).
- For the XIL API MAPort platform, there are further preconditions. Refer to How to Register and Configure an XIL API MAPort Platform (ControlDesk Platform Management 🕮).
- For the variable description of an Ethernet Bus Monitoring device, there are further preconditions. Refer to Ethernet Bus Monitoring Device: Information Required in the Variable Description File on page 29.

Method

To add a variable description to a platform/device

1. Select the platform/device in the **Project** controlbar.
2. If online calibration is running on the platform/device, select **Stop Online Calibration** from the context menu.
3. From the context menu of the platform/device, select **Add Variable Description**.

The Project Wizard opens, starting with the Select Variable Description dialog similar to the following one:



4 Select or import the variable description file by clicking Import from file.

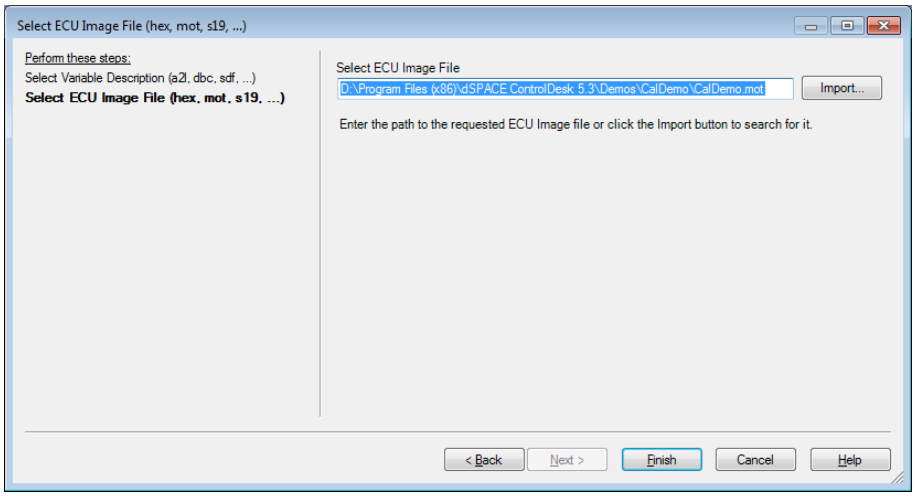
ControlDesk opens a standard Windows open dialog for you to choose one of the following files:

Platform/Device Type	Supported Variable Description File Type[1)]
Platforms (except for XIL API MAPort platforms)	SDF
XIL API MAPort platforms	MAPort configuration file type
Measurement and calibration devices	A2L
CAN Bus Monitoring devices[2)]	▪ DBC ▪ FIBEX ▪ AUTOSAR system description (ARXML)
Ethernet Bus Monitoring devices	AUTOSAR system description
LIN Bus Monitoring devices[2)]	▪ LDF ▪ FIBEX ▪ AUTOSAR system description (ARXML)

1) For information on the supported file versions, refer to Variable Descriptions Supported by ControlDesk on page 10.

2) If an AUTOSAR or FIBEX file contains the description of multiple clusters, ControlDesk lets you select the cluster to be imported.

5 For measurement and calibration devices, click **Next >**.
The **Select ECU Image File** dialog opens.



6 Enter or select the ECU Image file which contains the initial parameter values for the application.

⚠ WARNING

If the ECU Image file of a device does not provide initial parameter values for the entire memory region described by the memory segments of the device, ControlDesk sets the remaining parameter values to 0 in the mirrored memory and in the initially created data sets.
Do not download such data sets to the connected device hardware when you start online calibration, since this may cause property damage or even personal injury in connection with the connected system. Upload the parameter values from the connected device hardware instead.

7 Click **Finish** to close the dialog and activate the new variable description.

8 If you added memory segments to, or manually edited memory segments of, the previously selected variable description, you are asked whether you want to apply them to the current variable description. If you click **Yes**, they are added to the current variable description, but the default memory segments of the current variable description remain unchanged. If you click **No**, they are not added to the new variable description, but remain in the experiment for the previously selected variable description.

9 If you added a variable description to an XCP or CCP device and if the **Read XCP settings from A2L** or **Read CCP settings from A2L** property is selected for the device, the service configuration information is taken from the new variable description. If there are inconsistencies between the configuration information that is currently used by the device and the information contained in the new A2L file, a dialog opens for you to specify whether to use the current value or the one from the new A2L file for each inconsistent ECU interface setting.

Result

The selected variable description is added to the project's variable descriptions folder and a reference to it is added to the selected platform/device. For a platform, the corresponding application is automatically loaded if online calibration is started.

Next steps

You can configure the platform/device for which you added a variable description. Refer to:

- Handling Devices (ControlDesk Platform Management 🕮)
- Handling Platforms (ControlDesk Platform Management 🕮)

Related topics

Basics

Handling Variable Descriptions 15
Variable Descriptions Supported by ControlDesk 10

References

Add Variable Description 97
CCP Settings Advanced Properties (ControlDesk Platform Management 🕮)
XCP/GSI2 Settings Advanced Properties (ControlDesk Platform Management 🕮)

Basics on Variable Types

Overview

The table below lists all the variable types that ControlDesk displays in variable lists.

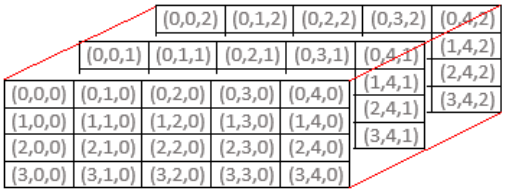
Variable Type	Description and Variable Symbol[1), 2)]	Example
Calculated variable	A scalar variable that can be measured and recorded, and that is derived from one or more *input variables*. The following input variable types are supported: ▪ Measurement variables ▪ Single elements of measurement arrays 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.	Scalar variable `cv`: `cv = a * b + c` (0)

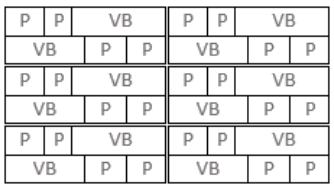
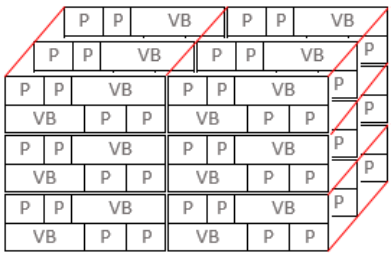
Variable Type	Description and Variable Symbol[1), 2)]	Example
Common axis	A parameter that consists of a 1-dimensional array containing axis points. A common axis can be referenced by one or more curves and/or maps. Calibrating the data points of a common axis affects all the curves and/or maps referencing the axis. Common axes are represented by the symbol.	1-dimensional array for axis points: `a[5] = {1,3,5,7,9}` 1 | 3 | 5 | 7 | 9

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.
		❶ 1-dimensional array (x-axis):
		`x[5] = {1,3,5,7,9}`
		❷ 1-dimensional array (data points):
		`d[5] = {3,5,2,2,1}`
		X | 1 | 3 | ❶ | 7 | 9
		3 | 5 | ❷ | 2 | 1

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.
		❶ 1-dimensional array (x-axis):
		`x[3] = {2,4,6}`
		❷ 1-dimensional array (y-axis):
		`y[5] = {1,3,5,7,9}`
		❸ 2-dimensional array (data points):
		`d[3][5] = {3,5,2,2,1`
		`6,1,3,4,2`
		`2,1,4,3,2}`
		X/Y | 1 | 3 | ❷ | 7 | 9
		2 | 3 | 5 | 2 | 2 | 1
		❶ | 6 | 1 | ❸ | 4 | 2
		6 | 2 | 1 | 4 | 3 | 2

Measurement	A scalar variable that can be measured, including individual elements of a measurement array. Measurement variables are represented by the symbol.	Scalar variable `a`: (0)
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. A 1-, 2-, or 3-dimensional array of measurement variables. In variable lists, ControlDesk displays entries for the	▪ 1-dimensional array `a[5]`: (0) | (1) | (2) | (3) | (4) ▪ 2-dimensional array `a[3][5]`: (0,0) | (0,1) | (0,2) | (0,3) | (0,4) (1,0) | (1,1) | (1,2) | (1,3) | (1,4) (2,0) | (2,1) | (2,2) | (2,3) | (2,4)

Variable Type	Description and Variable Symbol[1), 2)]	Example
	measurement array itself and for each array element. Measurement arrays are represented by the symbol.	▪ 3-dimensional array `a[3][4][5]`: 
Value	A scalar parameter, as well as the individual elements of a value block. Scalar parameters are represented by the P symbol.	Scalar variable `a`: `a = 3` 
String	A text variable in ASCII format. Strings are represented by the symbol.	`a = "text"` 
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 with five parameters (P), five value blocks (VB), and one struct (Strct) as elements. 

Struct array	An array of homogeneous struct variables.
	Struct arrays are represented by the symbol.
		▪ 2-dimensional struct array containing 6 homogeneous structs with parameters (P) and value blocks (VB):
		`a[3][2]`
		 
		▪ 3-dimensional struct array containing 18 homogeneous structs with parameters (P) and value blocks (VB):
		`a[3][3][2]`
		 

Variable Type	Description and Variable Symbol[1), 2)]	Example

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.
		▪ 1-dimensional array `a[5]`:
		`a[5] = {3,5,2,2,1}`
		3 5 2 2 1
		▪ 2-dimensional array `a[3][5]`:
		`a[3][5] = {3,5,2,2,1`
		`6,1,3,4,2`
		`2,1,4,3,2}`
		3 5 2 2 1
		6 1 3 4 2
		2 1 4 3 2

1) Displayed in the Variables controlbar

2) If an SDF/TRC variable description contains parameters tagged as `INITONLY`, they are visualized by an added pin, for example: . The `INITONLY` flag is used, for example, to mark fixed parameters in SDF/TRC files based on Functional Mock-up Units (FMUs).

Related topics

Basics

Basics of Placing Variables on a Layout (ControlDesk Layouting)
Instruments and the Variable Types they can Visualize (ControlDesk Instrument Handling)

Accessing 64-Bit Integer Variables

Introduction

Gives an overview of accessing 64-bit integer variables in ControlDesk.

Accessing 64-bit integer variables via ControlDesk instruments

The following table shows the ControlDesk instruments that support 64-bit integer variables (int64 and uint64 data types) in source mode. The support depends on whether you use the instrument via the ControlDesk user interface or via automation:

Instrument	Instrument Usage	
	Via the Graphical User Interface	Via the Instrument Automation Interface[1)]
Browser	—	✓ (read)
Display	✓ (read)	✓ (read)
Frame	—	✓ (read)

Instrument	Instrument Usage	
	Via the Graphical User Interface	**Via the Instrument Automation Interface[1)]**
Numeric Input	✓ (read and write)	✓ (read and write)
Variable Array	✓ (read and write)	✓ (read and write)

1) For information on accessing a variable value via an instrument automation interface, refer to Connecting variables to instruments (ControlDesk Automation 🕮).

Note

- If you connect 64-bit integer variables to instruments that do not support them, the precision is limited to 52 bit.
- Bitmasks are not supported for 64-bit integer variables. Bitmask settings are ignored.
- A range check is not supported for 64-bit integer variables. Range limitations (weak limits) are ignored.

Accessing 64-bit integer variables via direct variable access

Using automation, you can access variables directly without having to use instrument interfaces. Via direct variable access, you can read and write 64-bit integer variables in source mode.

Note

- Bitmasks are not supported for 64-bit integer variables. Bitmask settings are ignored.
- A range check is not supported for 64-bit integer variables. Range limitations (weak limits) are ignored.

For information on direct variable access, refer to Automating Direct Variable Access on page 160.

Related topics

Basics

Automating Direct Variable Access..... 160
Automating Parameter Calibration (ControlDesk Automation 🕮)
Automating Visualization of Variables on Layouts (ControlDesk Automation 🕮)

References

Range / IXaRange <<Interface>> (ControlDesk Automation 🕮)
Range Check Properties (ControlDesk Instrument Handling 🕮)
RangeCheck / IViRangeCheck <<Interface>> (ControlDesk Automation 🕮)

Ethernet Bus Monitoring Device: Information Required in the Variable Description File

Introduction

The variable description to be added to an Ethernet Bus Monitoring device must contain specific information.

Static destination ports

The Ethernet Bus Monitoring device does not support *dynamic* destination port assignment. Instead, the destination port must be assigned *statically* in the AUTOSAR variable description file.

The following listing shows an example:

```
<DYNAMICALLY-ASSIGNED>false</DYNAMICALLY-ASSIGNED>
<PORT-NUMBER>30491</PORT-NUMBER>
```

Fixed IP addresses

The Ethernet Bus Monitoring device does not support *dynamic* IP address assignment. Instead, *fixed* IP addresses must be assigned in the AUTOSAR variable description file. This applies to the sender and to the receiver in the Ethernet network.

The following listing shows an example:

```
<NETWORK-ENDPOINT ...
<IPV-4-ADDRESS-SOURCE>FIXED</IPV-4-ADDRESS-SOURCE>
<IPV-6-ADDRESS-SOURCE>FIXED</IPV-6-ADDRESS-SOURCE>
```

Related topics

Basics

Variable Descriptions Supported by ControlDesk 10

HowTos

How to Add a Variable Description to a Platform/Device 20

References

Ethernet Bus Monitoring Device (ControlDesk Platform Management)

Handling Variables in the Variables Controlbar

Where to go from here

Information in this section

Basics of the Variables Controlbar.....30
The ControlDesk **Variables** controlbar displays the variables of the active variable description for each platform/device of the currently active experiment.

How to Filter the Variable List.....33
In the **Variables** controlbar, you can filter the variable list using various filter types.

How to Use the Function Buttons for Searching and Filtering Variables.....37
The **Variables** controlbar offers function buttons to facilitate searching and filtering.

How to Search for a Variable Using Incremental Search.....41
You can find a variable by typing the first characters of one of its column entries.

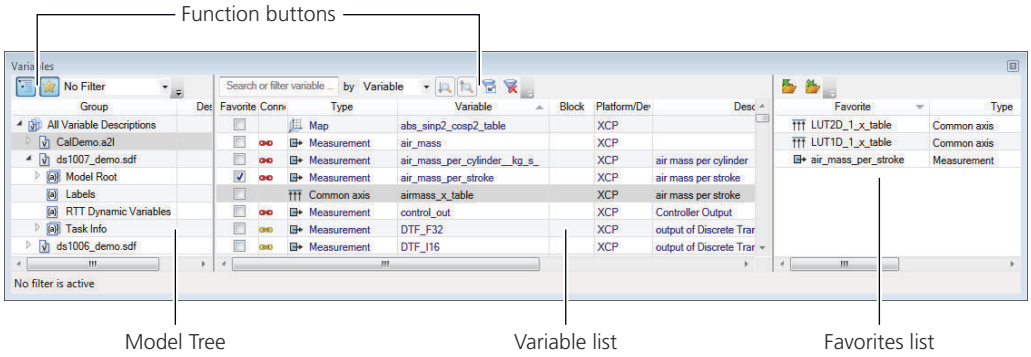
Basics of the Variables Controlbar

Introduction

The ControlDesk **Variables** controlbar displays the variables of the active variable description for each platform/device of the currently active experiment. The variable description file contains information about the variables on the platform/device hardware or VPU. For example, it contains the memory addresses, conversion formulas of parameters and measurement variables and display identifiers (alias names).

Variables controlbar

The Variables controlbar is a controlbar that provides access to the variables of the currently open experiment.



Tree view

The tree view of the Variables controlbar displays the structure of the variable descriptions that are currently active in the experiment. The variable list shows the variables of the selected node. You can enable and disable the tree view. Refer to Tree View on page 137.

Variable list

Lists the variables of the variable descriptions that are currently active in the experiment.

Function buttons

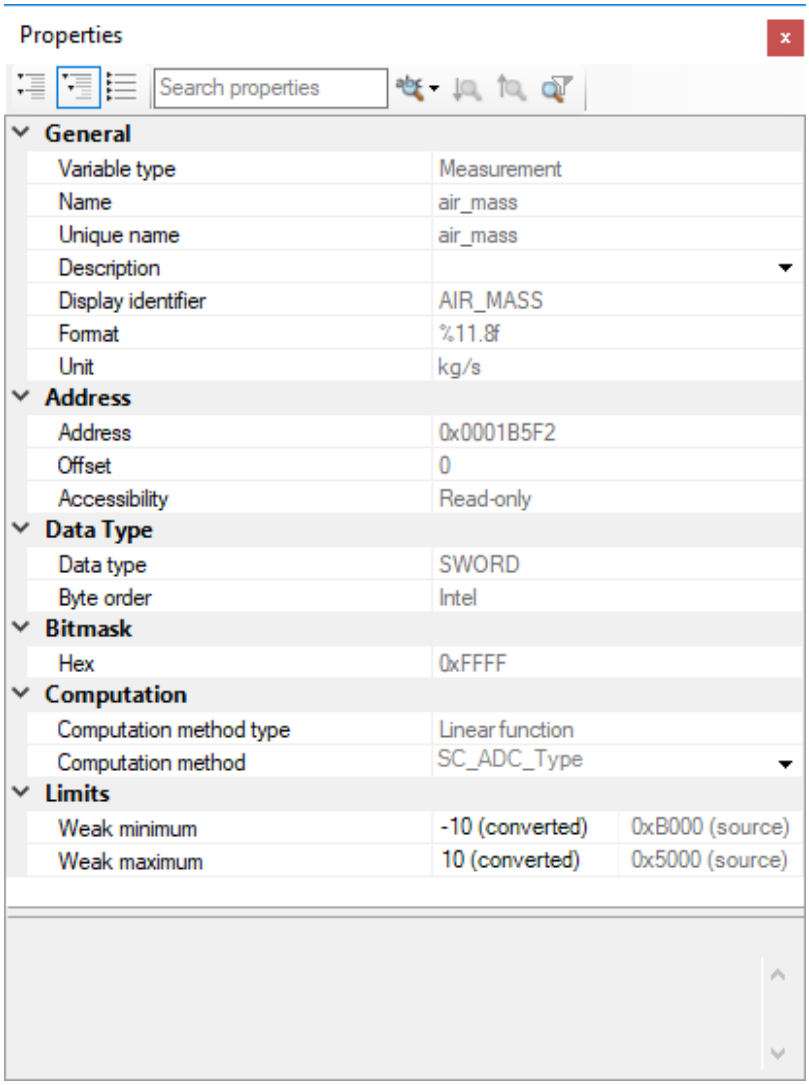
In the Variables controlbar, function buttons are displayed in the header of the variable list. You can use these buttons to filter the variable list and search for specific variables.

Favorites list

In the variable list, you can specify variables as favorites. These variables are then displayed in the favorites list. You can handle the content of the favorites list like a reduced variable list: Context menu commands and dragging options are the same. The names of the contained variables can be exported and imported via text or label list files.

Variable properties

In the Properties controlbar you can view the properties of variables. The illustration below shows an example:



Related topics

Basics

Defining Calculated Variables 49

References

Variables 142

How to Filter the Variable List

Objective

You can filter the variable list by using various filter types.

Basics

You can filter the variable list by using the following filters.

Filter Type	To Filter the Variable List by
Standard filter	▪ One or more single variable types ▪ All parameters ▪ All measurement variables
Wildcard filter	▪ Filter strings that contain wildcards
Combined filter	A filter that contains a combination of the following filter conditions: ▪ Filter strings that contain wildcards ▪ Variable types ▪ Variable states ▪ Memory segments The filter conditions are combined by logical operators (AND, OR) and can be grouped. Each combined filter you create is stored as a variable filter in the **Project** controlbar and can be activated via the **Project** controlbar or via the **Variables** controlbar.

If a filter is active, the color of list entries changes to blue. The **Variables** controlbar's status bar shows the current filter settings.

The following table shows the shortcut keys for filtering.

Shortcut Key	Purpose
Ctrl + **Shift** + **F**	To open the **Standard Filter** dialog.
Ctrl + **Shift** + **P**	To filter by parameters.
Ctrl + **Shift** + **M**	To filter by measurement variables.
Ctrl + **Shift** + **W**	To filter by strings that contain wildcards.
Ctrl + **Shift** + **A**	To disable the current filter and show all variables.

Preconditions

- The experiment must contain a platform/device with a variable description. Refer to How to Add a Variable Description to a Platform/Device on page 20.
- For the **Measurement Data Pool**: At least one measurement data file must be opened in the **Measurement Data Pool**. Refer to How to Open a Measurement Data File in the Measurement Data Pool (ControlDesk Measurement and Recording 🕮).

Possible methods

You can use the following filter types to filter the variable list:

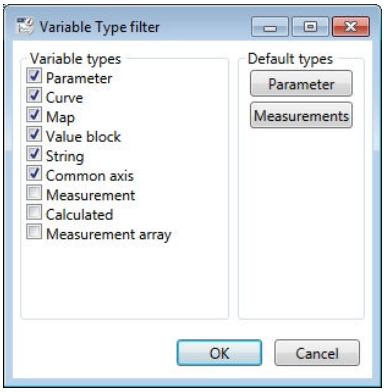
- If you want to filter by variable types, see Method 1 on page 34.
- If you want to filter by variable names including wildcards, see Method 2 on page 34.
- If you want to filter by a combination of filter conditions, see Method 3 on page 35.

Method 1

To filter the variable list by using a standard filter

1 From the context menu of the variable list, select Standard Filter, or press **Ctrl** + **Shift** + **F**.

The Variable Type Filter dialog opens.



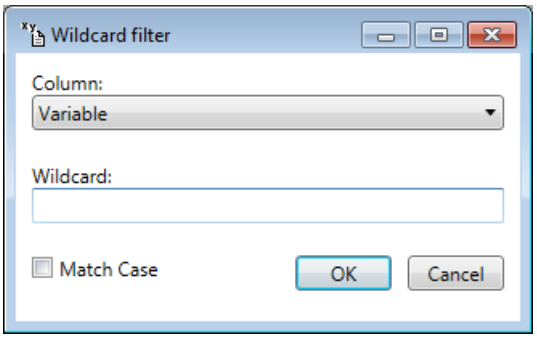
2 Select the variable types you want to filter the variable list by.

Method 2

To filter the variable list by using a wildcard filter

1 From the context menu of the variable list, select Wildcard Filter, or press **Ctrl** + **Shift** + **W**.

The Wildcard Filter dialog opens.



2 Select the column to which you want to apply the wildcard filter.

3 Enter a search string in the Wildcard edit field.

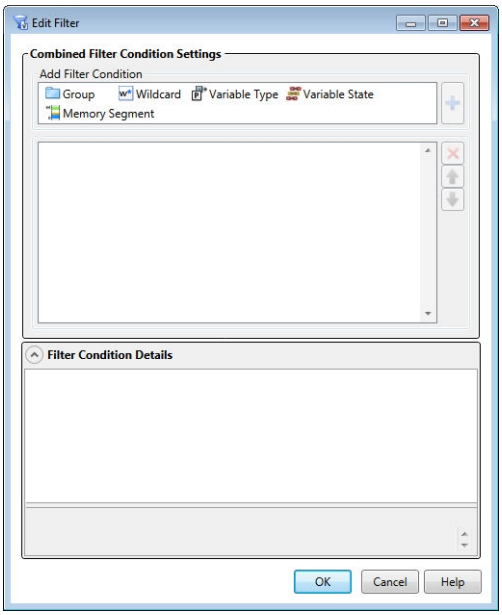
You can use the ? wildcard for one missing letter or the * wildcard for an undefined number of missing letters in the filter string.

Method 3

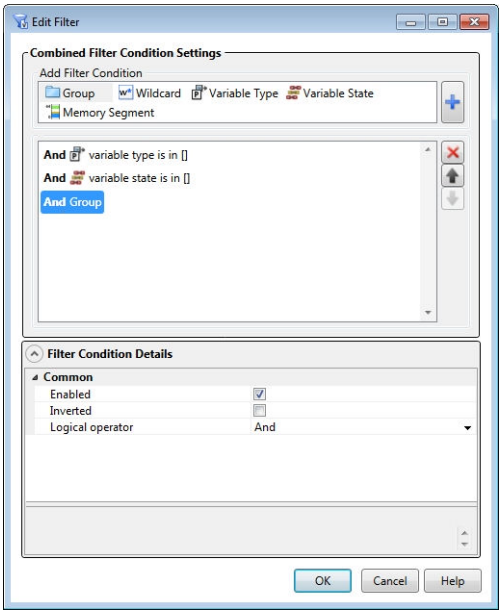
To filter the variable list using a combined filter

1 To create a new combined filter, select Apply/Edit Combined Filter – New from the context menu of the variable list.

The Filter Edit dialog opens.



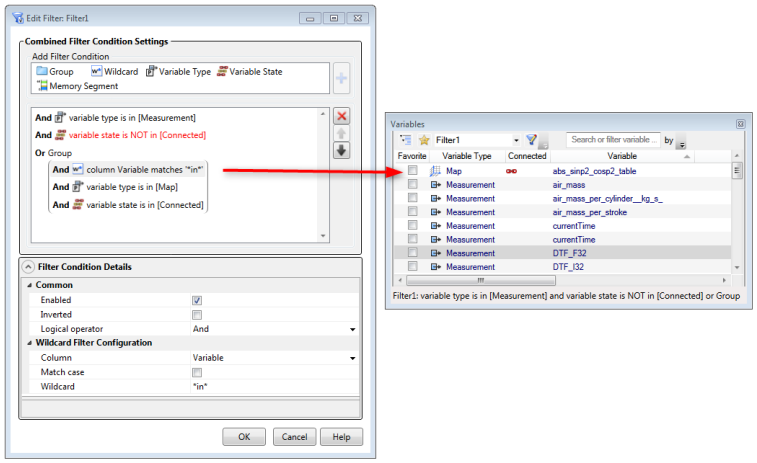
2 Double-click a filter condition or the group symbol to add it to the filter conditions list.



3 Select a filter condition or group element from the filter conditions list to specify its settings in the Filter Condition Details property list. You can also move it up and down via the buttons next to the list.

4 To add an element to a group in the filter condition list, select the group element and then double-click a filter condition or the group symbol to add it to the group.

The following illustration shows an example and its result, if the filter is activated.



5 Click **OK** to store the new combined filter in the **Variable Filters** folder of the **Project** controlbar. The filter is named **Filter** plus an incremented number.

6 To activate the new combined filter, select **Apply Combined Filter – <name>** from the context menu of the variable list.

Tip

You can create, rename, edit, activate, and remove combined filters under the **Variable Filters** node of the **Project** controlbar.

Result

The selected filter is active, the color of the entries in the variable list changes to blue. The active filter and the filter conditions are displayed in the status bar of the **Variables** controlbar.

Tip

In the **Variables** controlbar, all the defined filters can be used as predefined filters via the function buttons. Refer to How to Use the Function Buttons for Searching and Filtering Variables on page 37.

Next steps

You can now select the variables that you want to visualize in instruments. If you want to select variables from different tree nodes, use the checkboxes to display them in the favorites list.

Related topics

HowTos

How to Use the Function Buttons for Searching and Filtering Variables..... 37

References

Add/Edit Combined Filter – New / Create Variable Filter..... 102
Apply Combined Filter – Filter 1-n..... 103
Standard Filter..... 135
Wildcard Filter..... 150

How to Use the Function Buttons for Searching and Filtering Variables

Objective

The Variables controlbar offers function buttons to facilitate searching and filtering.

Basics

You can use various filters to reduce the variable list:

- You can activate a predefined filter, such as a standard filter, a wildcard filter, or a combined filter.
- You can enter a search or filter string directly into a text field.

Both filter methods can be used separately or in combination.

Predefined filters

You can filter the variable list by using the following predefined filters.

Filter Type	To Filter the Variable List by
Standard filter	▪ One or more single variable types ▪ All parameters ▪ All measurement variables
Wildcard filter	▪ Filter strings that contain wildcards
Combined filter	A filter that contains a combination of the following filter conditions: ▪ Filter strings that contain wildcards ▪ Variable types ▪ Variable states ▪ Memory segments The filter conditions are combined by logical operators (AND, OR) and can be grouped. Each combined filter you create is stored as a variable filter in the Project controlbar and can be activated via the Project controlbar or via the Variables controlbar.

For more information on predefined filters, refer to How to Filter the Variable List on page 33.

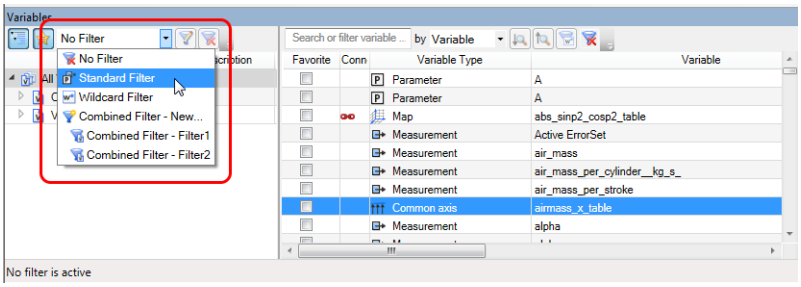
Possible methods

- If you want to filter the variable list by a predefined filter, see Method 1 on page 38.
- If you want to filter the variable list by a string in a column entry, see Method 2 on page 39.
- If you want to search for a string in a column entry, see Method 3 on page 40.

Method 1

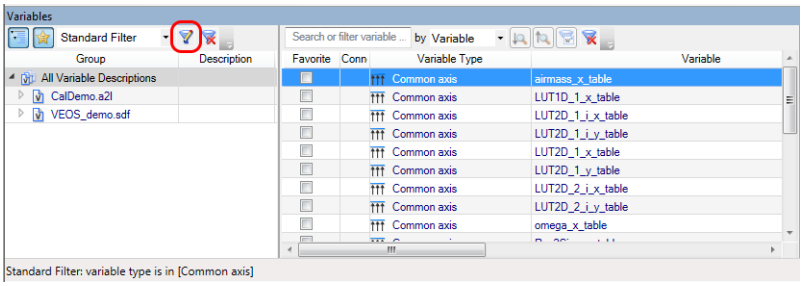
To use the function buttons for predefined filters

1 On the left side of the variable list header, open the filter list and select a filter.



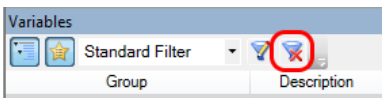
The variables of all the contained variable descriptions are filtered. The result is displayed in blue to show that the variable list is filtered. If a variable node in the tree does not contain any variable after filtering, it is hidden.

2 To change the settings of the selected predefined filter, click the edit filter symbol next to it.



The appropriate filter dialog opens and you can change the settings.

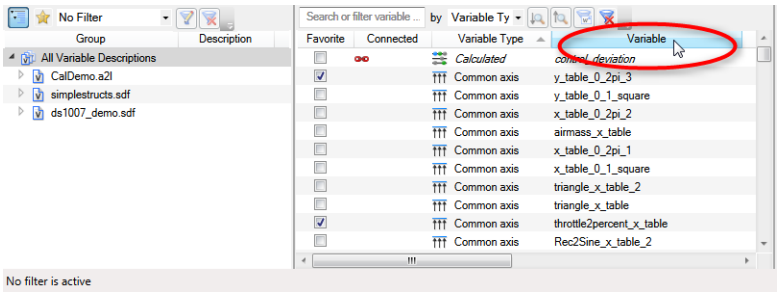
3 To clear a predefined filter, click the No Filter button:



Method 2

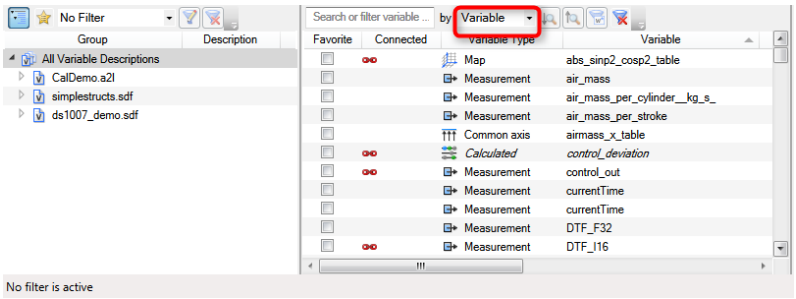
To use the function buttons for filtering by column entries

1 Click the column header of the column that contains the string you want to filter by.



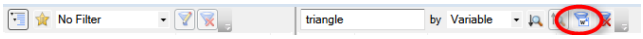
The entries of the column are sorted. This enhances the performance of search processes.

In the variable list header, the selected column is displayed in the column list.



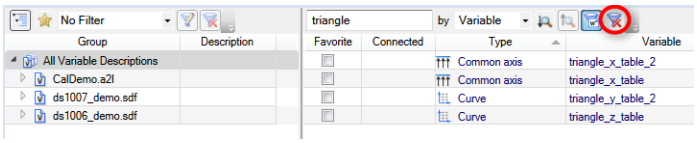
2 Enter a filter string in the text field in the variable list header. You can use the ? wildcard for one missing letter or the * wildcard for an undefined number of missing letters in the filter string.

3 Click the Wildcard Filter button.



The variables in the variable list are filtered. The result is displayed in blue to show that the variable list is filtered.

4 To clear the wildcard filter, click the Clear filter button.



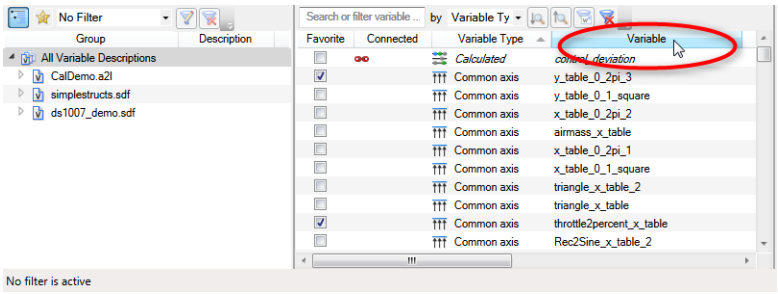
Note

If you have activated a predefined filter before, it remains active until you clear it by selecting No Filter in the filter list.

Method 3

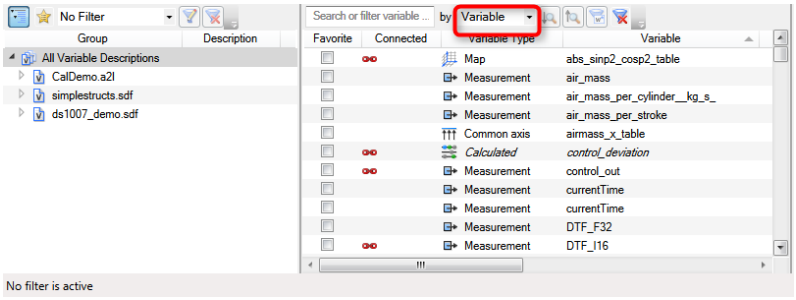
To use the function buttons for searching in column entries

1 Click the column header of the column you want to search in.



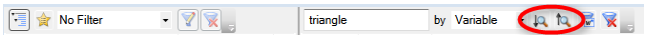
The entries of the column are sorted. This enhances the performance of search processes.

In the variable list header, the selected column is displayed in the column list.



2 Enter a search string in the text field in the variable list header. You can use the ? wildcard for one missing letter or the * wildcard for an undefined number of missing letters in the filter string.

3 Click one of the Find buttons to find the next or the previous occurrence of the search string.



Result

You have filtered the variable list and searched for a specific variable using the function buttons.

Related topics

HowTos

How to Filter the Variable List 33

How to Search for a Variable Using Incremental Search

Objective

You can find a variable by typing the first characters of one of its column entries.

Incremental search

If the variable list has the focus, ControlDesk stores all keystrokes in a buffer. The keystroke sequence is applied to the column that is used to sort the variable list. The first item that matches the current keystroke sequence is selected. A short beep indicates that no item matches the keystroke sequence.

Note

In the variable list of the Variables controlbar, you can search and filter variables via function buttons. In this case, keystrokes are sent to the text field for searching and filtering and you can combine searching and filtering functions to find a variable.

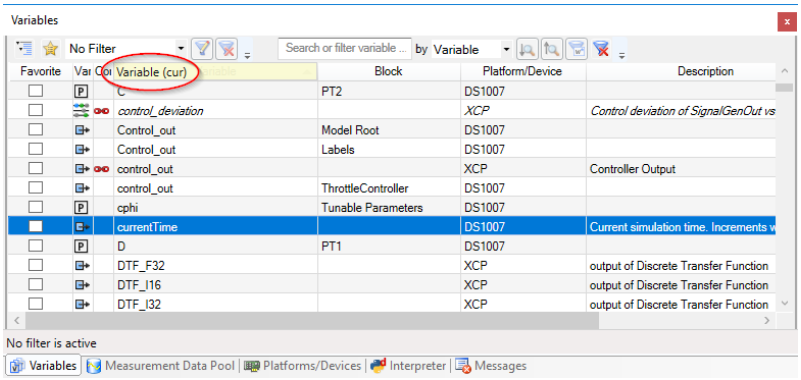
Preconditions

For the Measurement Data Pool: At least one measurement data file must be opened in the Measurement Data Pool. Refer to How to Open a Measurement Data File in the Measurement Data Pool (ControlDesk Measurement and Recording 🕮).

Method

To search for a variable via incremental search

1. In the variable list, click the column header to select the column to search in. The entries are sorted alphabetically.
2. Type the first characters of the entry you want to find. The current keystroke sequence is shown in the variable list header or in the column header.



Tip

You can use the wildcards * and ? in the keystroke sequence.

Result

The first item that matches the current keystroke sequence is selected. A short beep indicates that no item matches the keystroke sequence.

Tip

- To select the next item that matches the keystroke sequence, press **F3**.
  To select the previous one, press **Shift** + **F3**.
- To delete the last keystroke from the keystroke sequence, press **Backspace**. The selection is adapted.
  To delete the whole keystroke sequence, press **Ctrl** + **Backspace** or **Del**. The selection remains.

Next steps

When you have found the desired variable you can select it for visualization and place it on a layout. Refer to How to Place Variables on a Layout via the Variables Controlbar (ControlDesk Layouting 🕮).

Related topics

Basics

Basics of the Variables Controlbar..... 30

HowTos

How to Place Variables on a Layout via the Variables Controlbar (ControlDesk Layouting 🕮)

Source and Converted Mode

Where to go from here

Information in this section

How to Display a Value in Converted or Source Mode.....43
In ControlDesk, a numerical value can be displayed in its 'original format' (hex or source mode) on the hardware/VPU or in a converted form (physical or converted mode).

Basics on Variables Using Conversion Tables.....44
A conversion table specifies the computation of a source value into a converted value in the form of a table. In the case of verbal conversion, the converted value is a string that represents one numerical value or a range of numerical values.

How to Display a Value in Converted or Source Mode

Objective

In ControlDesk, a numerical value can be displayed in its 'original format' on the hardware/VPU (hex or source mode) or in a converted form (physical or converted mode).

Conversion formula or conversion table

The variable description file defines whether a value can be represented in a converted form, for example, a physical unit, and how it is to be converted. You can get information on the conversion in the properties dialog of a variable.

Tip

You can also define custom conversion formulas in the Properties dialog of an instrument. Refer to Custom Value Conversion Properties (ControlDesk Instrument Handling 🕮).

For variables using table conversion, every value which is available on the ECU can be displayed in the source mode, no matter whether the value is defined in the conversion table or not.

Restrictions

To use the conversion mode, a conversion formula or a conversion table must be defined in the variable description. You cannot switch between converted and source mode in the case of ASCII variables.

Method

To display a value in converted or source mode

1 From the context menu of an instrument or instrument row, select Display Values and choose Bin (Source), Dec (Source), Hex (Source) or Physical (Converted) to switch the conversion mode.

What you can select in the context menu depends on the instrument and the specific variable properties defined in the variable description.

Result

The representation of the value in the instrument changes.

Related topics

Basics

Basics on Variables Using Conversion Tables.....44

References

Bin (Source) (ControlDesk Instrument Handling 🕮)
Dec (Source) (ControlDesk Instrument Handling 🕮)
Header Visible Property (ControlDesk Instrument Handling 🕮)
Hex (Source) (ControlDesk Instrument Handling 🕮)
Physical (Converted) (ControlDesk Instrument Handling 🕮)
Variable Properties / Properties.....138

Basics on Variables Using Conversion Tables

Introduction

A conversion table specifies the computation of a source value into a converted value in the form of a table. In the case of verbal conversion, the converted value is a string that represents one numerical value or a range of numerical values.

Variables using conversion tables

Conversion tables are defined in the variable description file. They indicate how a source value is represented in the converted mode. The following kinds of conversion tables are supported.

Variable	Source Value	Converted Value
COMPU_TAB_INTP (variable using tabular conversion with interpolation)	Integer	Double
COMPU_TAB_NOINTP (variable using tabular conversion without interpolation)	Integer	Double
COMPU_VTAB (variable using verbal conversion)	Integer	String
COMPU_VTAB_RANGE (variable using verbal conversion range)	Double interval	String

You can get information on the conversion table in the **Properties** dialog of the variable.

Example This example shows how source values of variables can be converted via conversion tables.

Conversions that cannot be represented as a function (COMPU_TAB_INTP and COMPU_TAB_NOINTP)

Source Value	Converted Value
1	5
4.3	16.8
2	6
4.7	17.2

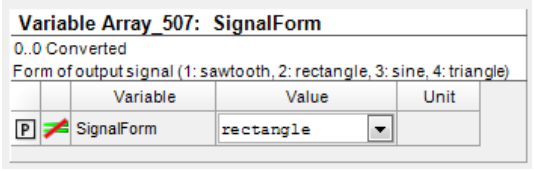
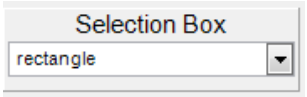
Verbal conversion of values (COMPU_VTAB)

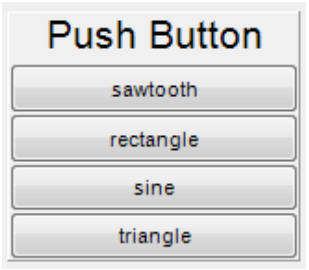
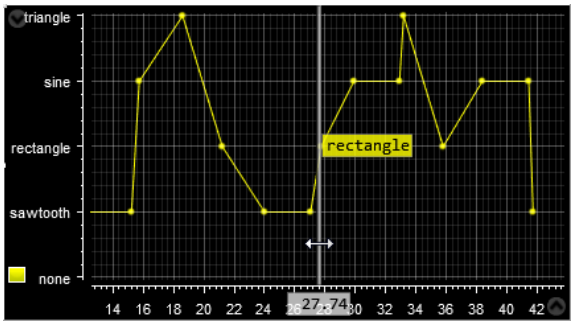
Source Value	Converted Value
0	engine off
1	idling
2	partial load
3	full load

Verbal conversion of value ranges (COMPU_VTAB_RANGE)

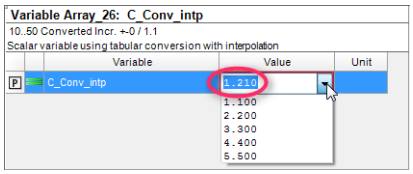
Source Value	Converted Value
0...2	low
3...5	medium
6...8	high

Instruments supporting variables with conversion tables ControlDesk provides several instruments that support variables that use conversion tables. The following table shows some examples:

Instrument	Variable With Verbal Conversion Table
Variable Array	
Selection Box	
Radio Button	

Instrument	Variable With Verbal Conversion Table
Push Button	
Display	
Time Plotter	

Calibrating parameters with conversion tables For changing the value of a parameter that uses a conversion table, some instruments offer a list with the table entries. Depending on the type of variable you can also enter values that are not in the list, as shown in the following illustration:



The following table shows whether the value in the edit field of the drop down list is editable or not:

Variable	Source Mode	Converted Mode
COMPU_TAB_INTP (variable using tabular conversion with interpolation)	Editable	Editable
COMPU_TAB_NOINTP (variable using tabular conversion without interpolation)	Not editable	Not editable
COMPU_VTAB (variable using verbal conversion)	Not editable	Not editable
COMPU_VTAB_RANGE (variable using verbal conversion range)	Editable	Not editable

Variables using multiscalings

Some variable description files contain multiscaling tables. A multiscaling table maps numerical source values to different subscalings. You can get information on the subscalings of a multiscaling in the Properties dialog of the variable.

Example This example shows the combination of a linear scaling and a verbal table range scaling in a multiscaling table.

Linear scaling:

Scaling Name	Conversion Formula
LinConv0210	`conv = source - 2 / 10`

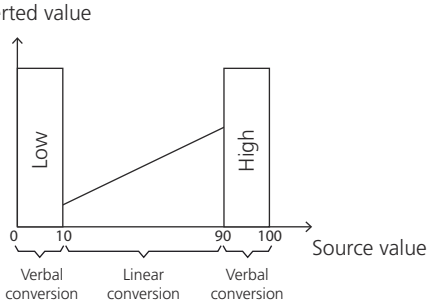
Verbal table range scaling:

Scaling Name	Range	Converted Value
VerbTabRange1090	0…10	Low
	11…90	Medium
	91…100	High

Multiscaling table:

Multiscaling Name	Range	Source Scaling	Converted Value	Example
MultiScaling1090	0...10	VerbTabRange1090	Low	Variable: Device/SignalMulti, Value: low
	11...90	LinConv0210	`conv = source - 2 / 10`	Variable: Device/SignalMulti, Value: 4.2
	91...100	VerbTabRange1090	High	Variable: Device/SignalMulti, Value: high

See the following illustration:



Instruments supporting variables with multiscaling The following instruments support variables with multiscaling:

- Display
- Index Plotter
- Numeric Input

- Time Plotter
- Variable Array

Some instruments do not completely support multiscaling, but can work with a part of the multiscaling table. The button instruments (On/Off Button, Push Button, Radio Button), the Selection Box, and the MultiState Display, for example, analyse the subscalings with conversion tables and display the table entries as buttons or states. Other subscalings of the multiscaling table are ignored.

Related topics

HowTos

How to Display a Value in Converted or Source Mode..... 43

References

Variable Properties / Properties..... 138

Defining Calculated Variables

Where to go from here

Information in this section

Introduction to Calculated Variables.....49
General information on computing and defining of calculated variables.

Defining Formulas for Calculated Variables.....63
The basis for computing a calculated variable is a formula that contains one or more variables or constant values as parameters.

Exporting and Importing.....78
You can transfer calculated variables to another ControlDesk experiment.

Introduction to Calculated Variables

Where to go from here

Information in this section

Basics on Defining Calculated Variables.....49
You can define calculated variables via the Variables controlbar.

Basics of Computing Calculated Variables in ControlDesk.....51
A number of factors affect the computation of calculated variables in ControlDesk.

How to Define a Calculated Variable.....55
To add a calculated variable to the variable list, you must define its properties.

How to Write Calculated Variable Values to Parameters and Writable Measurements.....60
You can write the value of a calculated variable to one or more output parameters.

Basics on Defining Calculated Variables

Introduction

You can define calculated variables via the **Variables** controlbar.

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.

Computation formula

The basis for computing a calculated variable is a formula that contains one or more variables or constant values as parameters.

To define a calculated variable, you have to perform the following steps:

1. Assign an appropriate formula to the calculated variable, either by creating a new formula or by selecting an existing one.
2. Assign input signals (measurement variables, single elements of measurement arrays/value blocks, scalar parameter variables, or existing calculated variables) or constant values to the parameters of the formula.

Note

At least one parameter in the formula must be connected to a signal or parameter.

A new calculated variable is added to the variable list and can be used for measurements and recordings.

Example

The following example shows how to use a formula named '`CV_Formula1`' for a calculated variable named '`CalcVar1`':

```
CV_Formula1 = (Parameter1 * Parameter2 + Parameter3) / 10
```

Assigning two signals and one constant value:

- `Parameter1 = SignalA`
- `Parameter2 = SignalB`
- `Parameter3 = 5`

As a result, `CalcVar1` is calculated like this:

```
CalcVar1 = (SignalA * SignalB + 5) / 10
```

Related topics

Basics

Basics of Computing Calculated Variables in ControlDesk....51
Basics of Formulas for Calculated Variables....63

HowTos

How to Define a Calculated Variable....55

Basics of Computing Calculated Variables in ControlDesk

Introduction

If you measure a calculated variable, a number of factors affect the computation. The following shows this for a measurement in ControlDesk.

Computing calculated variables

During a measurement in ControlDesk, the input values that are used for computing a calculated variable depend on the following factors:

- The selected measurement rasters of the input variables
- The selected measurement raster of the calculated variable
- The times at which ControlDesk receives data packages from the platform/device hardware or VPU

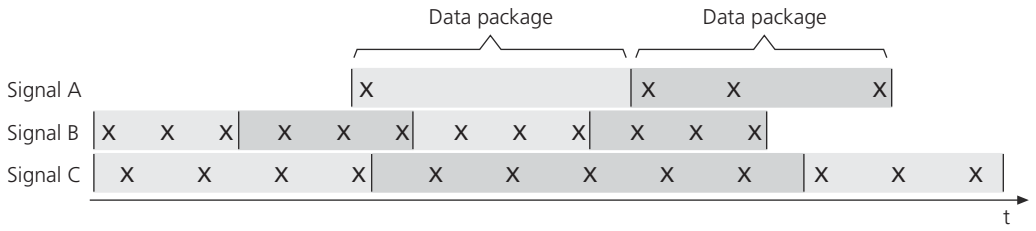
To compute calculated variables, it is sufficient to observe the input variables. The variables do not have to be part of a running measurement. For more information on observing variables, refer to Observing Variables (ControlDesk Measurement and Recording).

Data packages

ControlDesk receives the data of signals at different points in time, bundled in data packages of unequal size.

The following illustration shows data packages of three input signals. The measurement of signal A is triggered by events (nonequidistant data points). Signal B and signal C are measured periodically (equidistant data points).

A data point is represented by an x, and the start and end of a data package are represented by a vertical line.



Selecting a measurement raster for the calculated variable

You can select input variables from different platforms/devices. The platform/device that is associated to the variable description you "add" a calculated variable to, determines the measurement rasters that are available for the calculated variable.

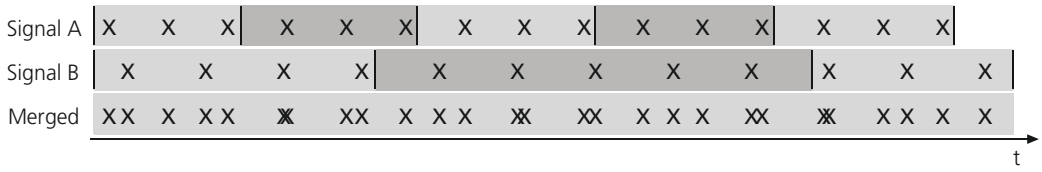
Under this precondition, you can select one of the following raster types for measuring calculated variables.

Measurement raster of one of the input signals You can select a *measurement raster of one of the input signals* and use it for the calculated variable. In this case you make this raster the master raster and the values of the calculated variable get the same time stamps as the values of the selected input signal.

Tip

If the measurement raster you select for the calculated variable is synchronous, the time stamps of the computed values are also synchronous to the task on the hardware or VPU.

Merged raster ControlDesk provides the *merged raster* for calculated variables. In this raster, the time stamps of all input variables are merged in a new raster.



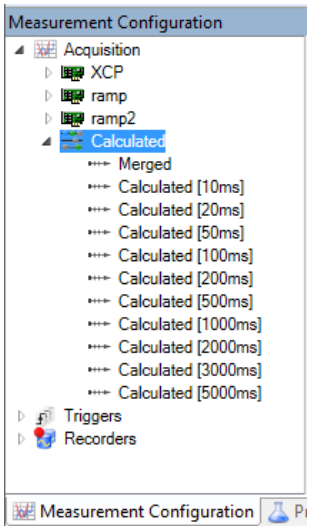
Polling rasters for calculated variables If none of the input signals (or the merged raster) offers the desired measurement raster, you can select a predefined *polling raster for calculated variables*. If you select a polling raster for a calculated variable, ControlDesk treats this raster as the master raster when calculating.

Note

Polling rasters are asynchronous measurement rasters. They do not correspond to a task running on an ECU or real-time system.

If you select a polling raster that is equal to a raster of an input variable, ControlDesk uses the raster of the input variable instead.

The following illustration shows ControlDesk's **Measurement Configuration** with the merged raster and the polling rasters for calculated variables.



For details on measurement rasters, refer to Basics on Measurement Rasters (ControlDesk Measurement and Recording 🕮).

Visualization of a calculated variable

To compute a value of a calculated variable, ControlDesk waits until it has received data from all input signals at the time stamps of the master raster. Therefore, if you connect a calculated variable to an instrument on a layout, the visualization in the instrument might be delayed, because ControlDesk ensures that all the data relevant to calculation has been received.

Computation principle

ControlDesk checks the following two points to specify whether new values of a calculated variable can be computed:

Valid data from all signals To compute a value for a calculated variable, ControlDesk needs at least one time stamp, at which it has received valid data from all input signals. As long as this condition is not true, the data is held in memory and ControlDesk waits for the next data package of an input variable.

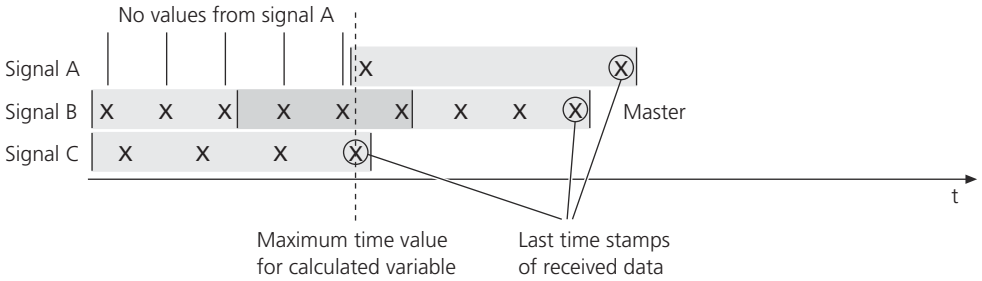
Maximum time stamp for the calculated variable Each time ControlDesk has received a new data package, it checks the last time stamp of each signal. The lowest of these last time stamps determines the maximum point in time for which a value of the calculated variable can be computed with the received data.

Example

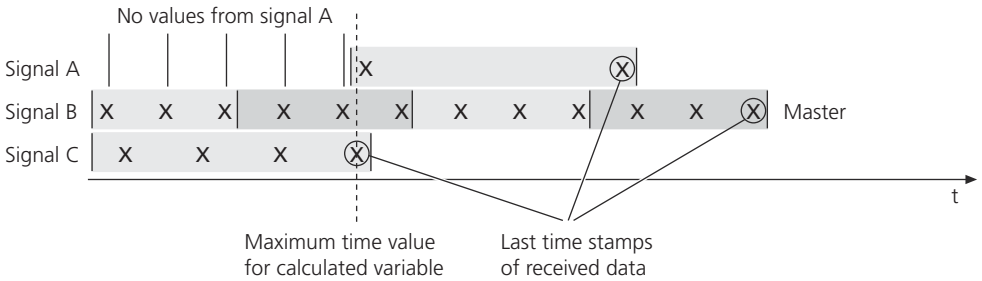
The following examples show how ControlDesk determines the values for computing a calculated variable.

Each example shows one new received data package. The raster of signal B is selected as the raster of the calculated variable. The raster of signal B therefore is the master raster.

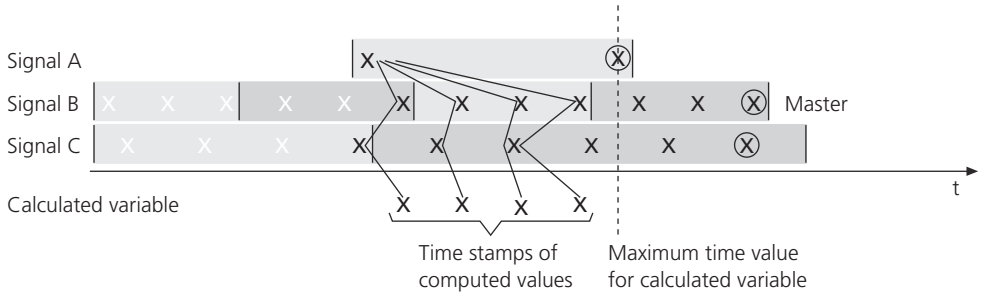
Five data packages received – no computation ControlDesk cannot compute a value for the calculated variable, because it has not received valid data from signal A at the maximum time value which is limited by signal C.



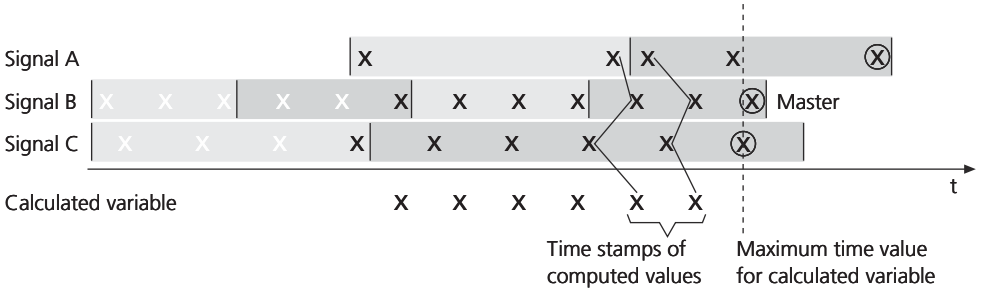
Next data package received – no computation ControlDesk has received a new data package, but still cannot compute a value for the calculated variable, because it has not received valid data from signal A at the maximum time value which is still limited by signal C.



Next data package received – first computation ControlDesk has received a new data package. Four values for the calculated variable are computed. Values that cannot be considered for the computation are represented by a white cross. The time stamps of the computed values correspond to the values of signal B, because signal B is the master raster.



Next data package received – next computation ControlDesk has received a new data package. Two more values are computed for the calculated variable.



Related topics

Basics

Basics on Defining Calculated Variables....49

HowTos

How to Define a Calculated Variable....55
How to Write Calculated Variable Values to Parameters and Writable Measurements....60

How to Define a Calculated Variable

Objective

To add a calculated variable to the variable list, you must define its properties.

Properties of a calculated variable

As well as the "usual" variable properties such as the name, you must assign a computation formula to the calculated variable and specify signals (and/or constant values) to be used as input.

For basic information on defining calculated variables, refer to Basics on Defining Calculated Variables on page 49.

Restrictions

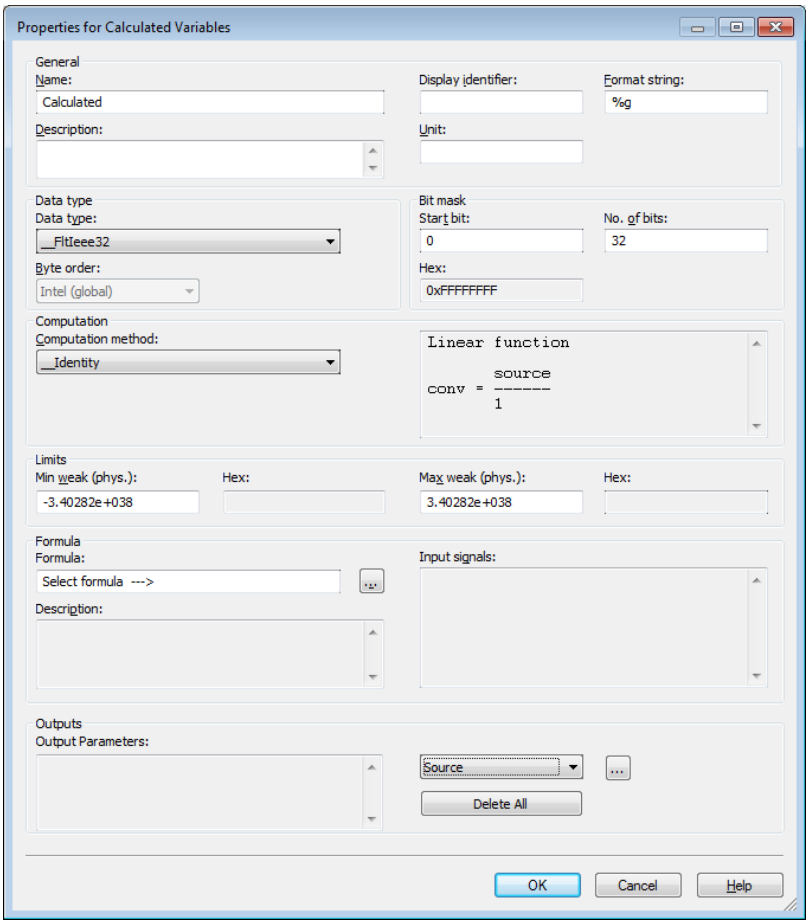
You cannot use multidimensional measurement variables as input signals for calculated variables.

Method

To define a calculated variable

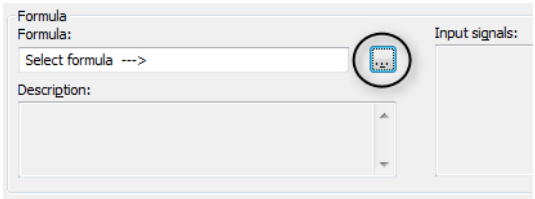
1 In the Variables controlbar, right-click a variable and select Calculated Variables – New.

The Properties for Calculated Variables dialog opens.

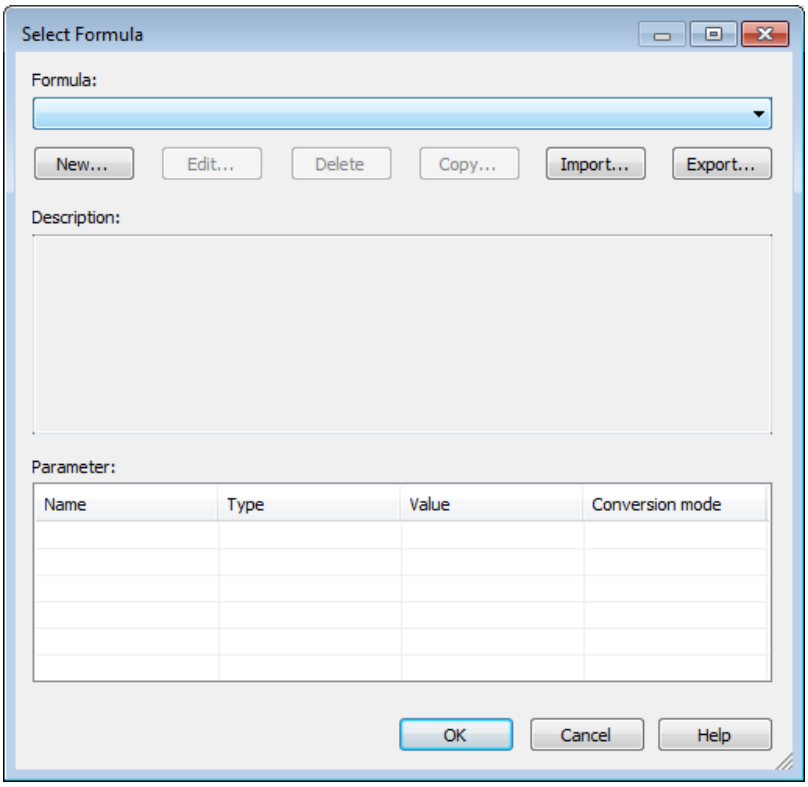


2 Enter or select the properties of the calculated variable.

3 Click the Browse button next to the Formula field to select a formula for the calculated variable.



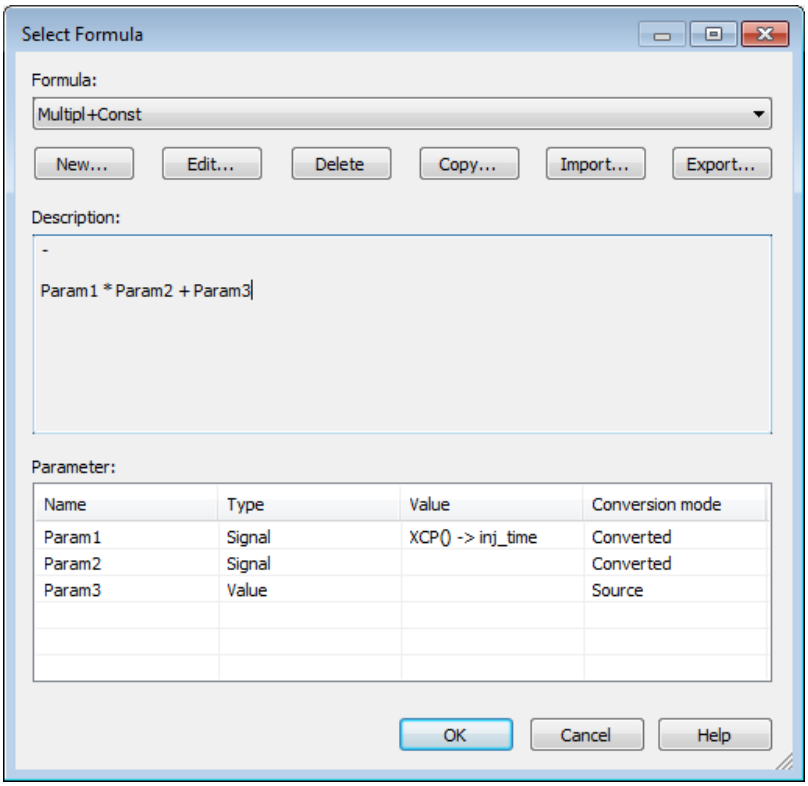
The Select Formula dialog opens.



4 From the Formula list, select a formula.

To define a new formula or edit an existing one, click New, Edit, or Copy. Refer to How to Define a Formula for Calculated Variables on page 67.

5 In the **Parameter** table, specify the connection of the parameters in the formula.



- To assign a signal to a parameter, select **Signal** in the **Type** column, click into the **Value** field and click the Browse button to select a variable from the variable list displayed in the **Select Variable** dialog. The variable list in the dialog is prefiltered and shows only variables that can be used as input

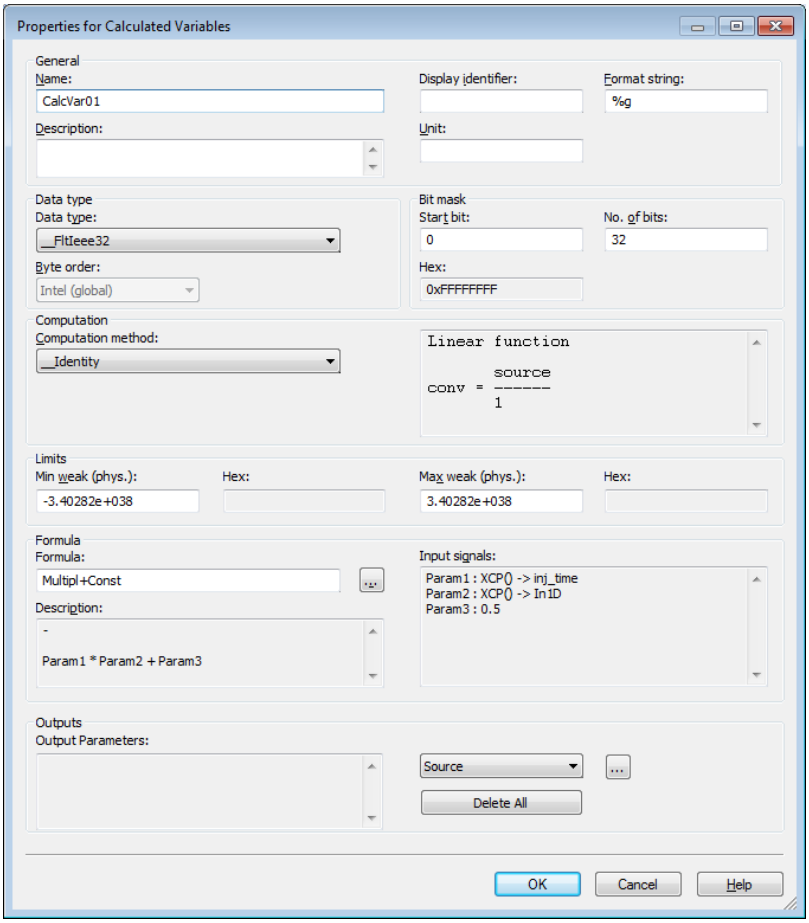
signals for calculated variables. For more information on the dialog, refer to Select Variable Dialog on page 133.

- To assign a constant value to a parameter, select **Value** in the **Type** column and enter a constant value in the **Value** column.

By default, signal values are specified as converted values. You can change the specification to source values in the **Conversion Mode** column. The specification for constant values cannot be changed.

6 Click **OK** to confirm your settings.

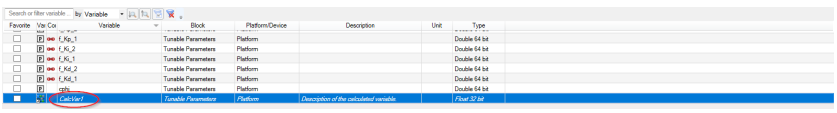
The **Select Formula** dialog closes and the selected formula and input signals are displayed in the **Properties for Calculated Variables** dialog.



7 Click **OK** to close the **Properties for Calculated Variables** dialog.

Result

You have defined a new calculated variable, which is added on the root node level of the variable description. If you have selected a subnode, a reference to the calculated variable is added to it. The new variable is displayed at the end of the variable list.



Next step

If you want to define output parameters for the calculated variable, refer to How to Write Calculated Variable Values to Parameters and Writable Measurements on page 60.

Related topics

Basics

Basics on Defining Calculated Variables 49

References

Calculated Variables - New 108

How to Write Calculated Variable Values to Parameters and Writable Measurements

Objective

You can write the value of a calculated variable to one or more *output parameters*.

Method

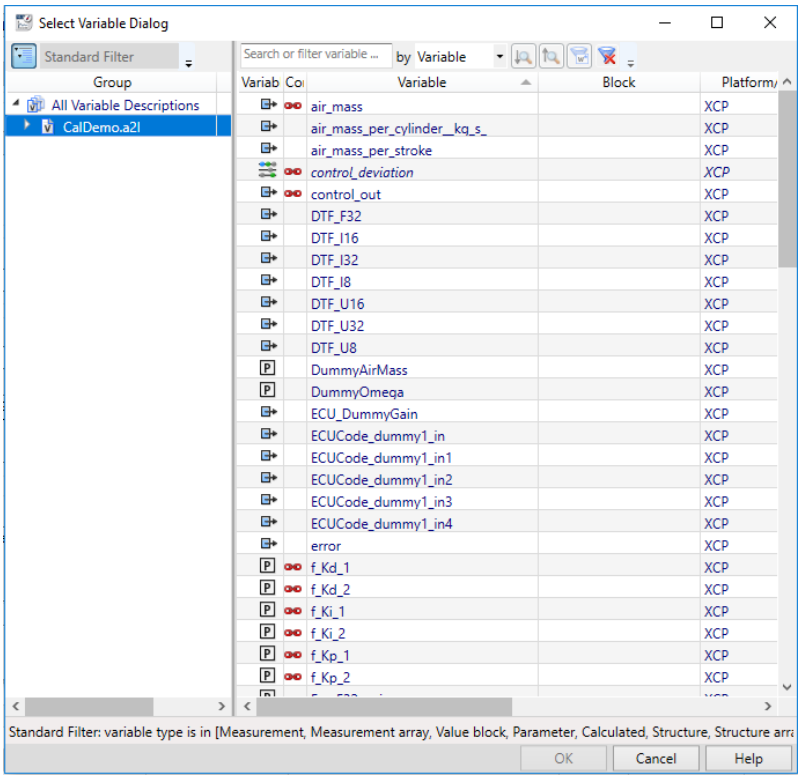
To write calculated variable values to parameters and writable measurements

1. From the context menu of the calculated variable you want to assign an output parameter to, select **Calculated variables - Edit** (refer to Calculated Variables - Edit on page 105).

   ControlDesk opens the **Properties for Calculated Variables** (refer to Properties for Calculated Variables Dialog on page 118) dialog.

2. Select **Source** or **Converted** next to the Output parameters list to specify the display mode of the value, then click the Browse button.

The Select Variable dialog opens.



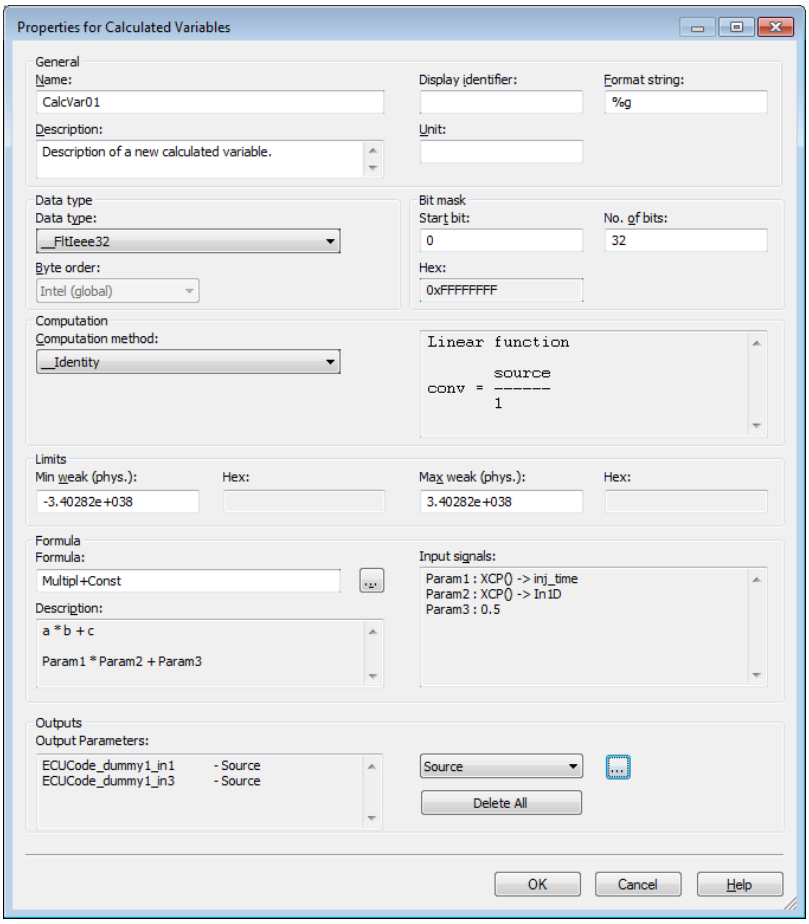
3 Select a parameter or writable measurement to be used as an output parameter.

Note

The output parameter must not be one of the calculated variable's input variables or part of their references. This creates circular references.

4 Click OK to close the Select Variable dialog. You can select further output parameters by clicking the Browse button again.

All the selected variables are displayed in the Output parameters list.



5 Click OK to close the Properties for Calculated Variables dialog.

Result

The value of the calculated variable is written to the memory address of the selected output parameters.

- If a measurement is running, the value is written according to the measurement raster of the calculated variable.
- If no measurement is running, the value is written each time it changes.

Related topics

Basics

Basics of Formulas for Calculated Variables 63

References

Calculated Variables - Edit 105
Properties for Calculated Variables Dialog 118

Defining Formulas for Calculated Variables

Where to go from here

Information in this section

Basics of Formulas for Calculated Variables..... 63
For calculated variables, you can use existing formulas or define new ones.

How to Define a Formula for Calculated Variables..... 67
You can add new formulas to the list of available formulas for calculated variables.

Examples of Defining a Formula for Calculated Variables..... 71
In the **Formula Editor** dialog you can add new formulas to the list of available formulas for calculated variables. Below are some examples.

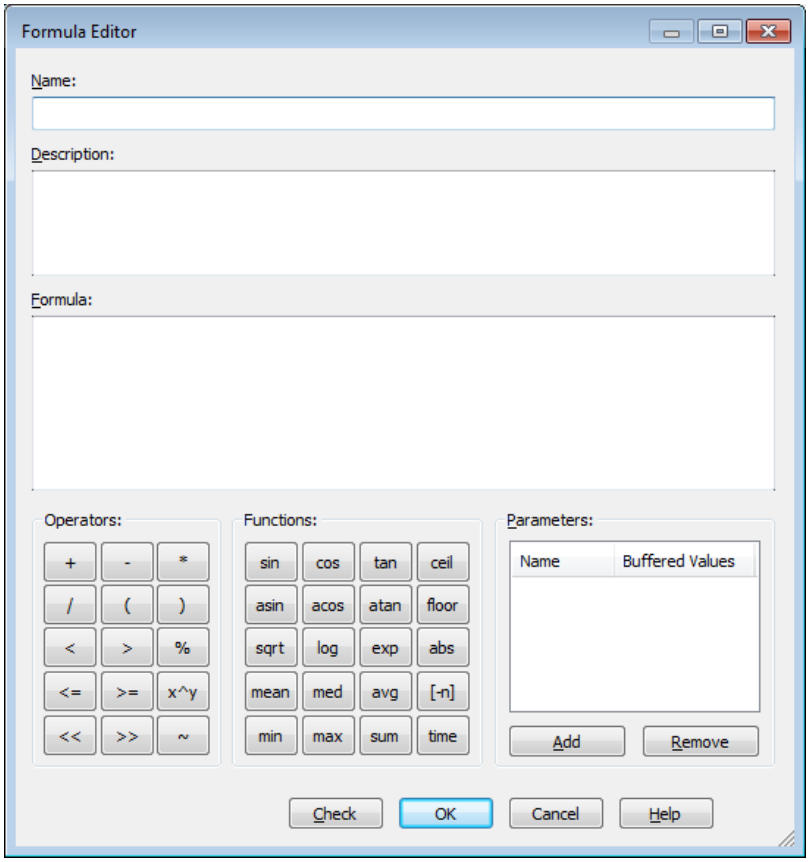
Basics of Formulas for Calculated Variables

Introduction

For calculated variables, you can use existing formulas or define new ones.

Formula Editor

You can add new formulas to the list of formulas or edit existing ones via the Formula Editor.



For instructions, refer to How to Define a Formula for Calculated Variables on page 67.

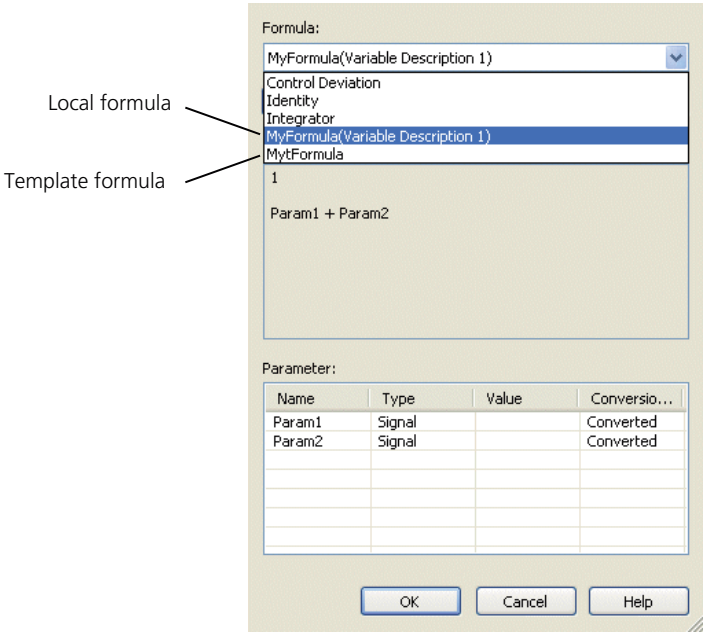
Local and template formulas

Local formulas Each formula is saved *locally* in the variable description containing a calculated variable that uses the formula.

Template formulas In addition, when you create a new formula, it is also stored as a *template formula* in a user-specific template formula pool. This allows you to reuse the formula in other variable descriptions.

Local and template formulas in the Formula Editor Local and template formulas are available in the Formula Editor.

The following illustration shows a locally saved formula and the template formula in the **Formula Editor** as an example.



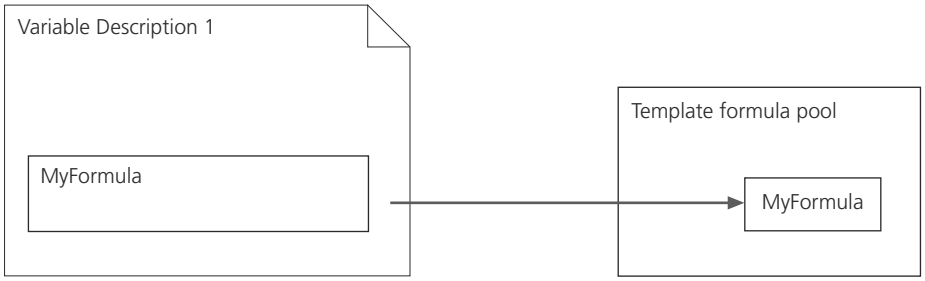
Example Suppose you work with one computer and already created a formula in a variable description (*variable description 1*), this formula is saved locally in variable description 1 and also stored in your template formula pool.

If you update the formula in another variable description (*variable description 2*) on the same computer, the changed formula is saved under the same name in variable description 2 and overwrites the existing template formula in the template formula pool.

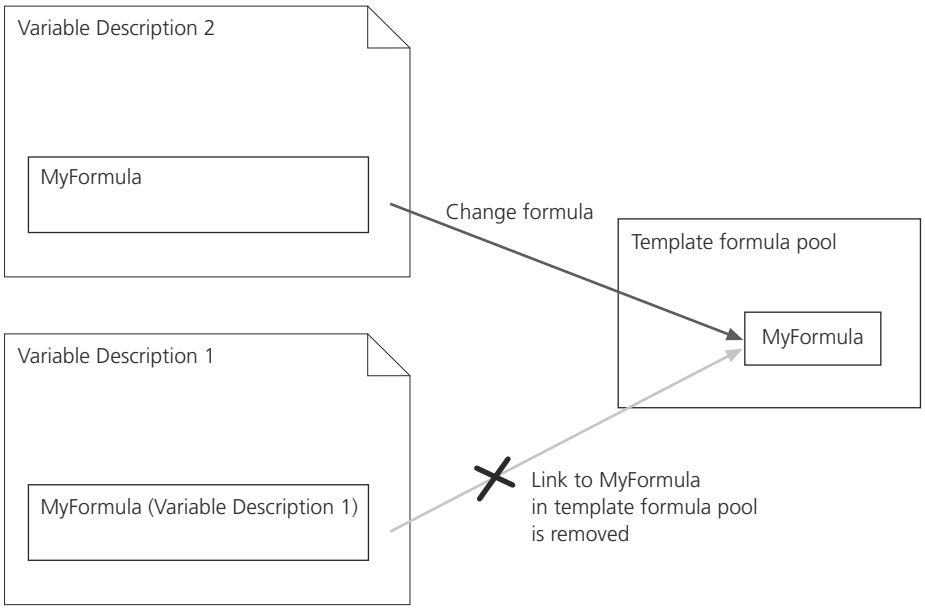
In variable description 1, the locally saved original formula and the updated template formula are both available in the formula editor's list of formulas. The local formula is indicated by the formula name followed by the variable description name.

The following illustration shows the concept of the local formula and the template formula on the same computer.

Step 1: Create MyFormula in Variable Description 1



Step 2: Edit MyFormula in Variable Description 2



Importing formulas If you import a VXF file that contains a formula with the same name as one in the template formula pool but that is different from it, ControlDesk prompts you to overwrite the template formula. A replaced template formula is still available as a local formula. It is indicated by the formula name followed by the name of the variable description in which it was created.

Related topics

HowTos

How to Define a Formula for Calculated Variables....67

Examples

Examples of Defining a Formula for Calculated Variables....71

References

Properties for Calculated Variables Dialog....118

How to Define a Formula for Calculated Variables

Objective

You can add new formulas to the list of available formulas for calculated variables.

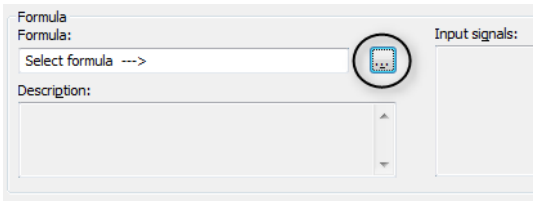
Method

To define a formula for calculated variables

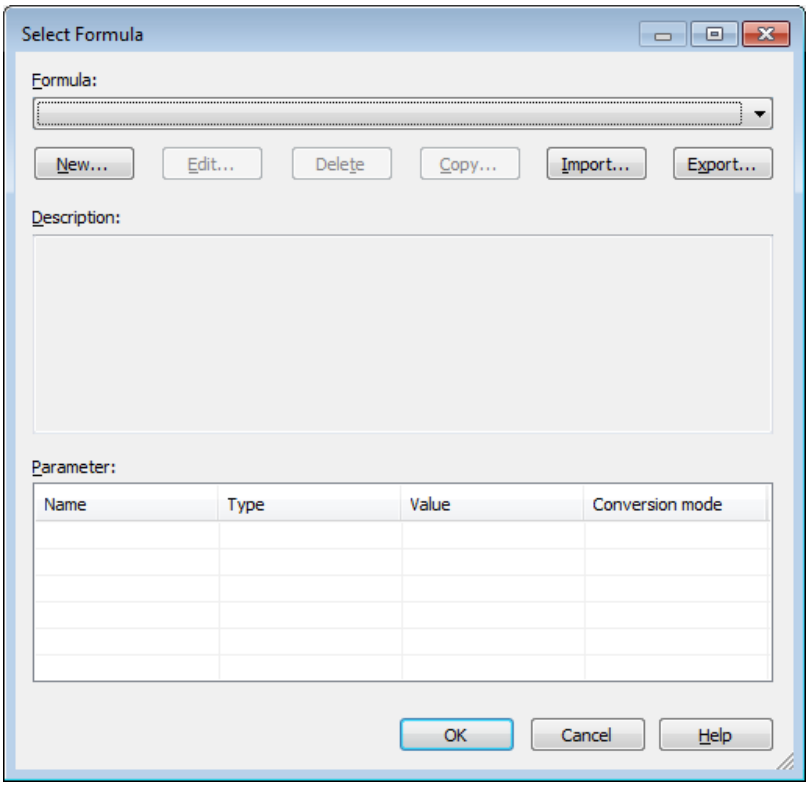
1 Open the Properties for Calculated Variables dialog.

In the Variables controlbar, the Properties for Calculated Variables dialog opens when you edit an existing calculated variable or define a new one.

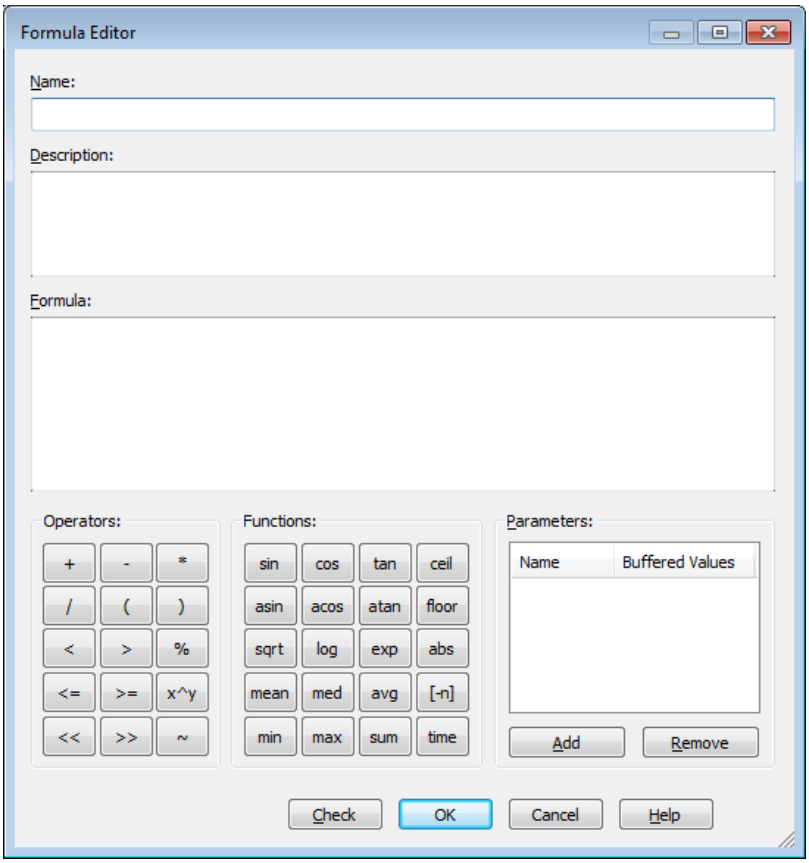
2 Click the Browse button next to the Formula field.



The Select Formula dialog opens.



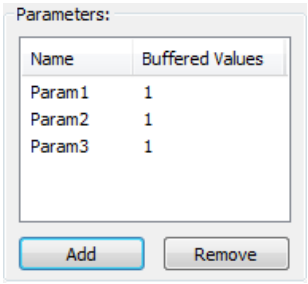
3 In the Select Formula dialog, click New. The Formula Editor dialog opens.



4 Enter a unique name for the formula and, if desired, a description text for it.

The formulas for calculated variables are stored globally for all projects on the host PC. The formula name must therefore be unique on your host PC.

5 Below the Parameters list, click Add until you have created the desired number of parameters for your formula.



Double-click a parameter to insert it into your formula in the Formula field.

To rename a parameter, select it and press **F2**.

6 In the Formula field, enter operators and values to complete the formula. You can use the buttons for operators and functions below the field.

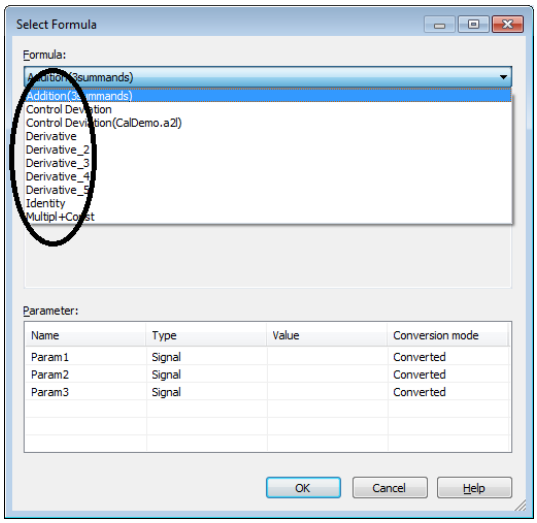
7 To check the syntax of the formula, click Check.

8 Click OK to confirm your settings and close the dialog.

The syntax of the formula is checked when the dialog is closed. If a syntax error is detected, the dialog stays open and you can correct the formula.

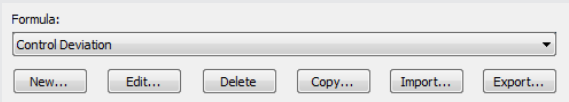
Result

A new formula for calculated variables is added to the list of available formulas in the Select Formula dialog.



Tip

- If you want to edit an existing formula or get a copy of an existing formula as a template, click Edit or Copy in the Select Formula dialog.



- You can enlarge your template formula pool by importing formulas and share your formulas by exporting them. Refer to How to Export and Import Template Formulas for Calculated Variables on page 80.

Note

If you edit an existing formula, make sure that the changes do not cause problems for calculated variables which reference it.

Next steps

You can now use the formula for calculated variables. Refer to How to Define a Calculated Variable on page 55.

Related topics

Basics

Basics on Defining Calculated Variables....49

Examples

Examples of Defining a Formula for Calculated Variables....71

References

Properties for Calculated Variables Dialog....118

Examples of Defining a Formula for Calculated Variables

Introduction

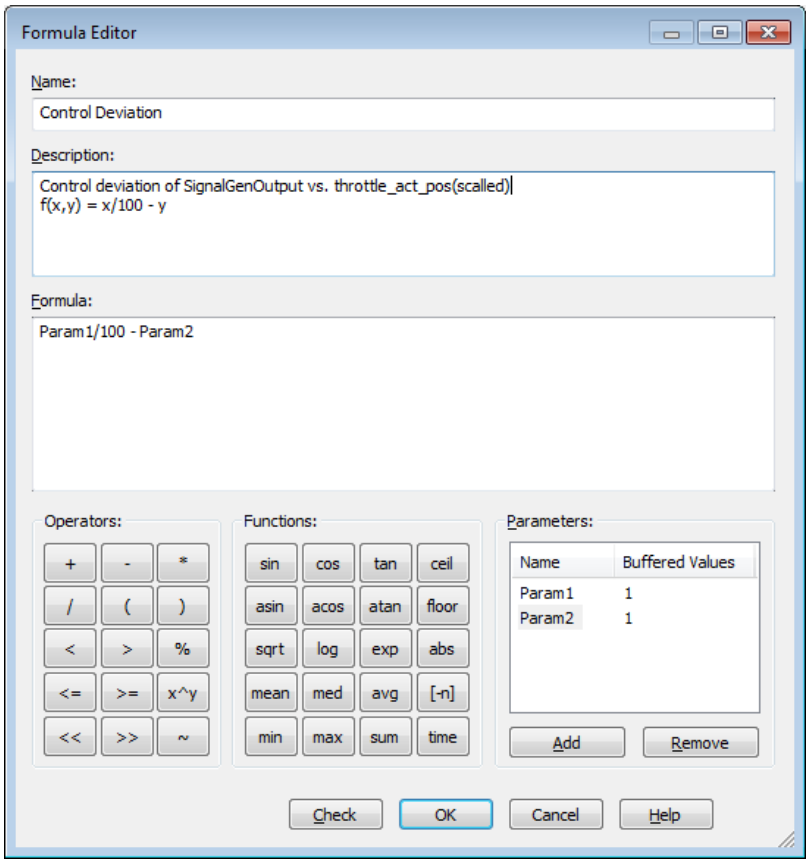
In the Formula Editor dialog you can add new formulas to the list of available formulas for calculated variables. Below are some examples.

Formula with 2 parameters

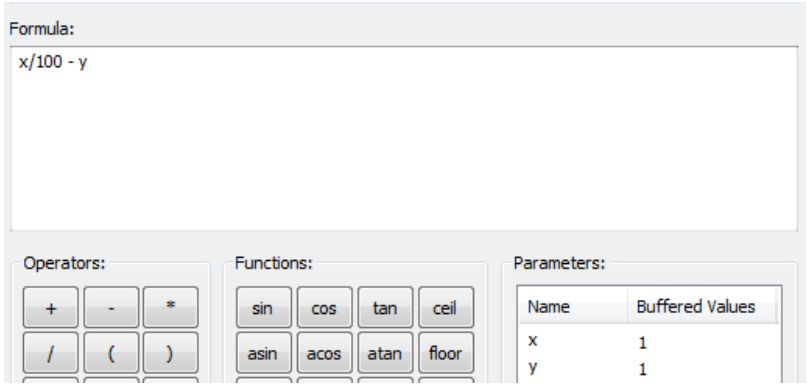
Imagine you have to read the values of 2 signals as input for the following function:

$f(x,y) = x/100 - y$

In the Formula Editor dialog, you have to create a formula with two parameters as shown in the following illustration.



If you want to rename the parameters in the formula, select them in the Parameters list, then click them once.

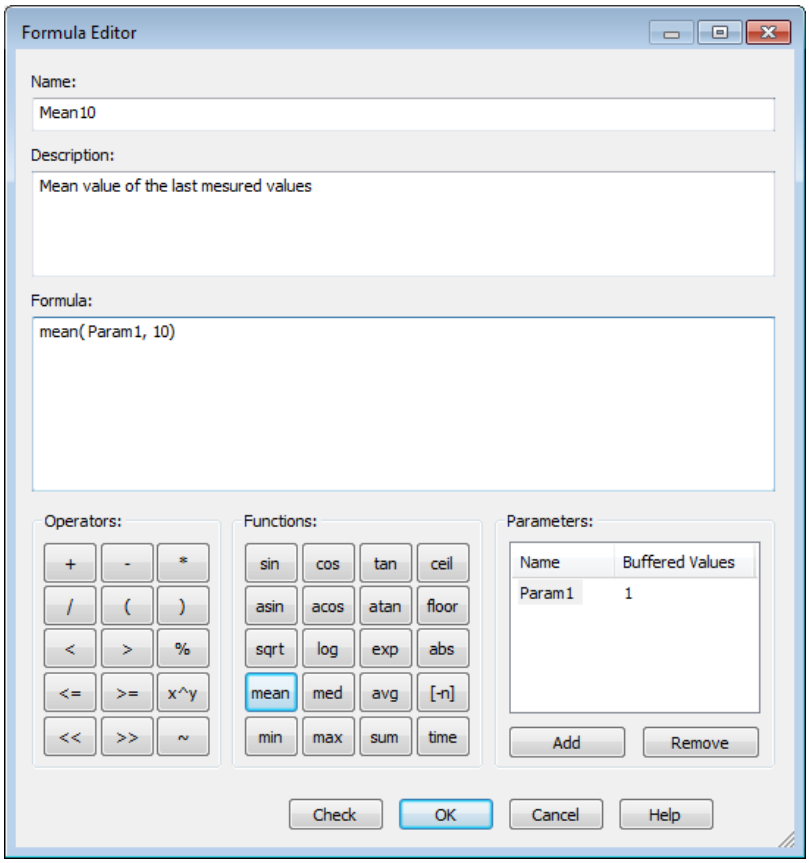


Formula with statistical functions

Imagine you want to calculate the mean value of the last 10 values of a signal. The formula for this function has a length parameter which lets you specify the number of consecutive values in the past to take into account for calculation:

f(x) = mean(x, <length>)

In this case the length must be set to 10 (or -10, the sign does not matter). The following illustration shows the formula.



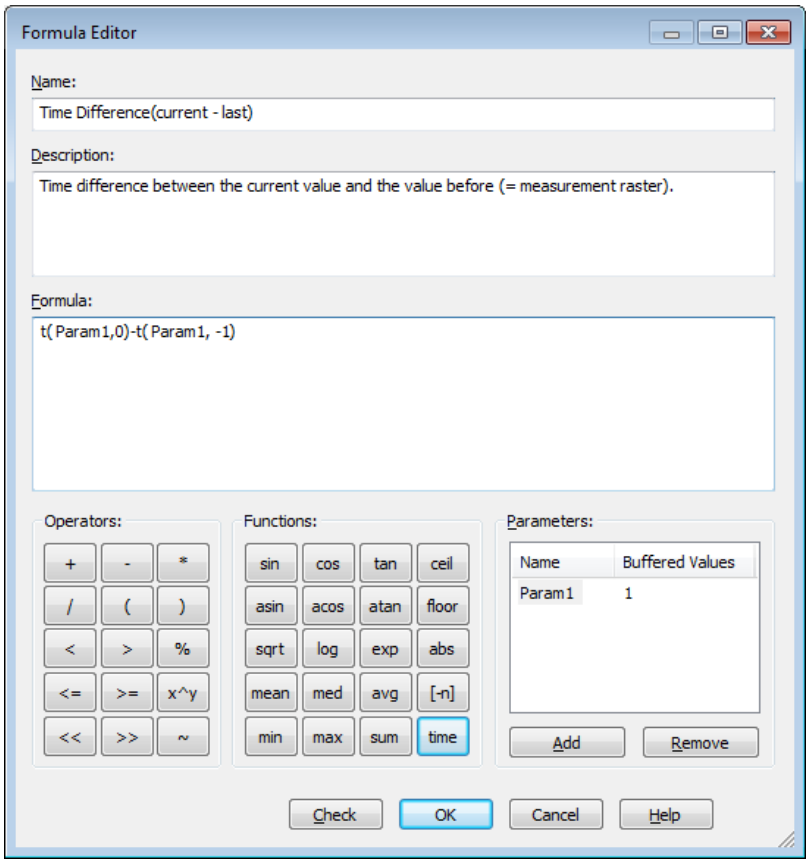
You can also use the following buttons for formulas with statistical functions: **med** (median), **avg** (average), **min** (minimum), **max** (maximum), and **sum** (sum).

Formula for calculating time differences

Imagine you want to calculate the time difference between two values on the timeline of an input signal. The formula for a time function has an index parameter which lets you specify the point on the timeline used for the calculation:

f(x) = t(x, <index>)

The illustration shows the formula for computing the time difference between the current value (index = 0) and the one before (index = 1 or -1).

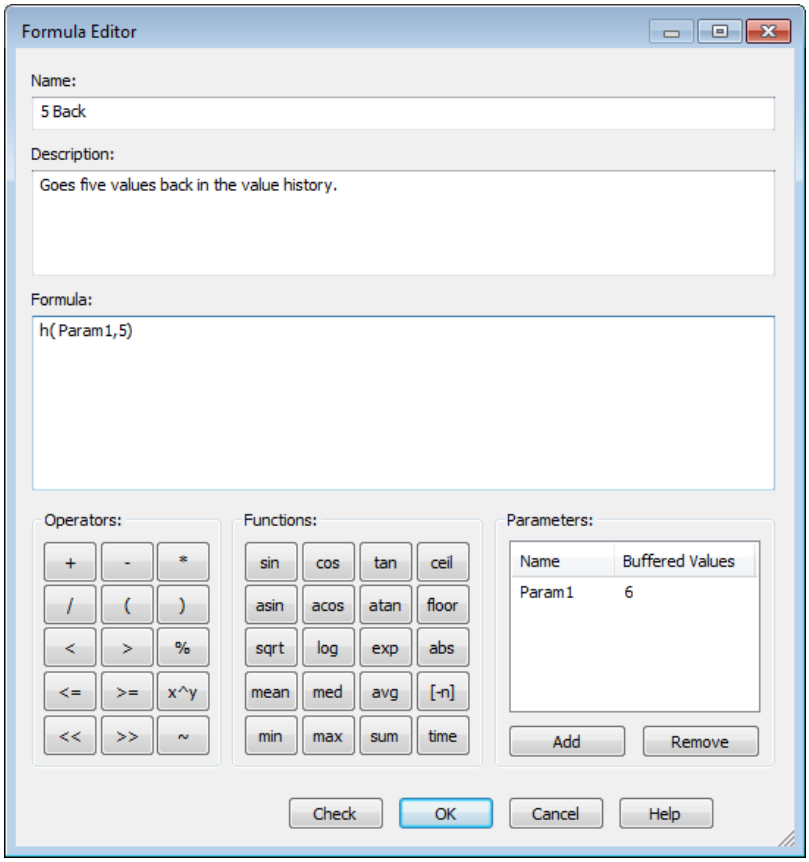


Formula for calculating the value history

If you do not want to calculate the time stamp but the value of an input signal in the past, use a history function with the following syntax:

f(x) = h(x, <index>)

The index parameter lets you specify the number of measurement points you want to trace back to. The illustration shows how to go back 5 values in the value history.

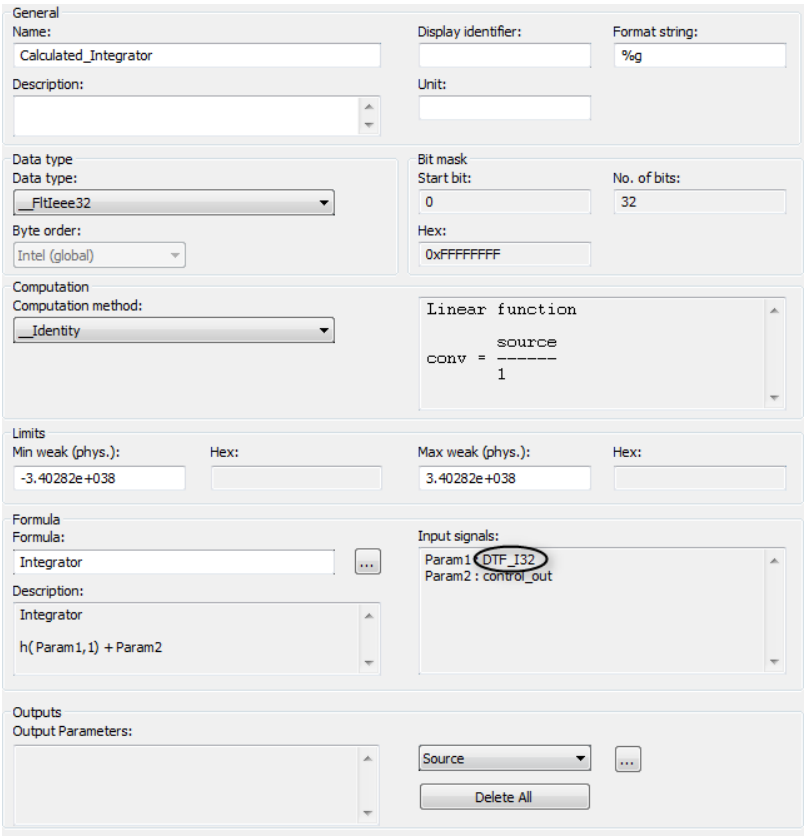


Formula for calculating the Integral of a signal

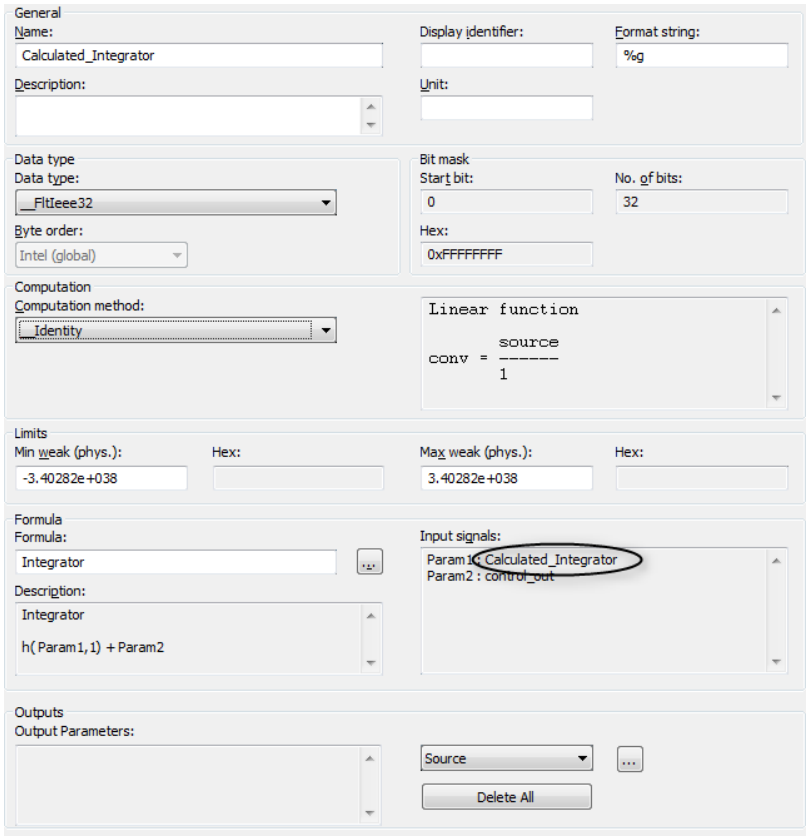
If you want to calculate the integral of a signal, i.e., to calculate the sum of all the last measured values and the current measured value of a signal, you can define an integrator formula. The integrator formula is a function that refers to itself:

$f(x) = h(f(x), 1) + x$

To assign a calculated variable to itself in the formula, you must first use any variable as a placeholder for f(x). When the new calculated variable is available in the Variables controlbar, you must replace this placeholder with the new calculated variable in the formula. The illustration below shows the first step in defining an integrator calculated variable: The DTF_I32 variable is used as a placeholder.



The illustration below shows the second step in defining an integrator calculated variable: The DTF_I32 variable is replaced by the Calculated_Integrator calculated variable.



More examples

For a complete description of all operator and function buttons with an example for each, refer to Properties for Calculated Variables Dialog on page 118.

Related topics

Basics

Basics on Defining Calculated Variables 49

References

Properties for Calculated Variables Dialog 118

Exporting and Importing

Where to go from here

Information in this section

How to Export and Import Calculated Variables..... 78
If you want to use a calculated variable in another experiment, you have to export and import the variable.

How to Export and Import Template Formulas for Calculated Variables..... 80
You can export and import template formulas for calculated variables to enlarge the template formula pool that is available to a user.

How to Export and Import Calculated Variables

Objective

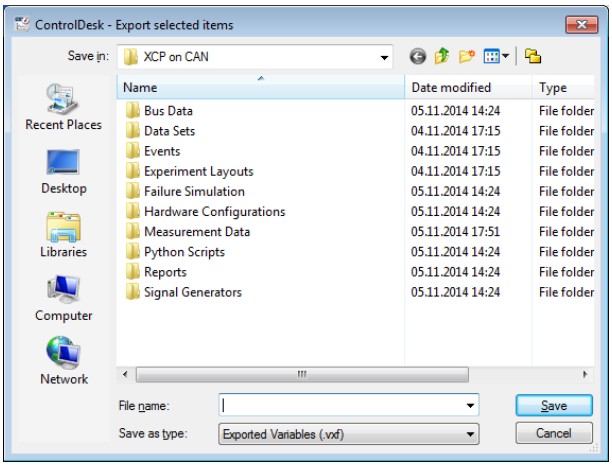
Calculated variables and their formulas are saved, exported, or imported when the experiment containing them is saved, exported or imported. If you want to reuse the calculated variables and/or their formulas in a different experiment, you need to export and import them specifically.

Method

To export and import calculated variables in the Variables Controlbar

1. In the Variables controlbar, select all the calculated variables you want to export. Multiple selection is possible by pressing **Ctrl** or **Shift** when clicking a variable.
2. Right-click one of the selected calculated variables.
3. From the context menu, choose Calculated Variables and select Export selected items.

The Export selected items dialog opens.



4 Specify the name and path of a VXF file to save the selected calculated variables to.

5 Click Save to export the selected calculated variables.

6 To import calculated variables, open the context menu of the Variables controlbar, choose Calculated variables and select Import.

The ControlDesk - Import dialog opens.

7 Select the VXF file you have exported the calculated variables to.

Result

You have exported the selected calculated variables to a VXF file and then imported them. Assigned formulas, type definitions, and scalings are also imported.

- Calculated variables are always imported on the root node level. If they were exported from a subnode, variable references are added if a suitable subnode exists.
- If the project already contains a calculated variable with the same name as the one you try to import, the imported calculated variable is renamed automatically (by adding an underscore and a consecutive number). The existing calculated variable is not changed or overwritten.
- If the project already contains a formula, type definition, or scaling with the same name as one of those you try to import, the new one is not imported. Existing formulas, typedefs, and scalings are not changed or overwritten.

Related topics

Basics

Basics of Formulas for Calculated Variables....63

HowTos

How to Export and Import Template Formulas for Calculated Variables....80

References

Calculated Variables - Export Selected Items....106
Calculated Variables - Import....107

How to Export and Import Template Formulas for Calculated Variables

Objective

You can export and import template formulas for calculated variables to enlarge the template formula pool that is available to a user.

Tip

Calculated variables and their formulas are saved, exported, or imported when the experiment containing them is saved, exported or imported. If you want to reuse the calculated variables and/or their formulas in a different experiment, you need to export and import them specifically.

User-specific template formula pool

The formulas for calculated variables are stored according to the user name for all the projects on a host PC. You can export the template formulas in the template formula pool to a VXF file to transfer them to another host PC or to make them available to another user on the same host PC.

Note

- If you export and import a calculated variable, the assigned formula is also exported and imported (as a local formula). You do not have to export and import the assigned formula separately.
- If you export and import the formulas for calculated variables, they are added as template formulas to the user-specific template formula pool. Local formulas are not exchanged.

For more information on the formula editor and the concept of local and template formulas, refer to Basics of Formulas for Calculated Variables on page 63.

Method

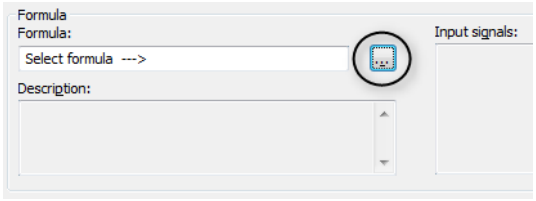
To export and import template formulas for calculated variables

1 Open the Properties dialog for calculated variables.

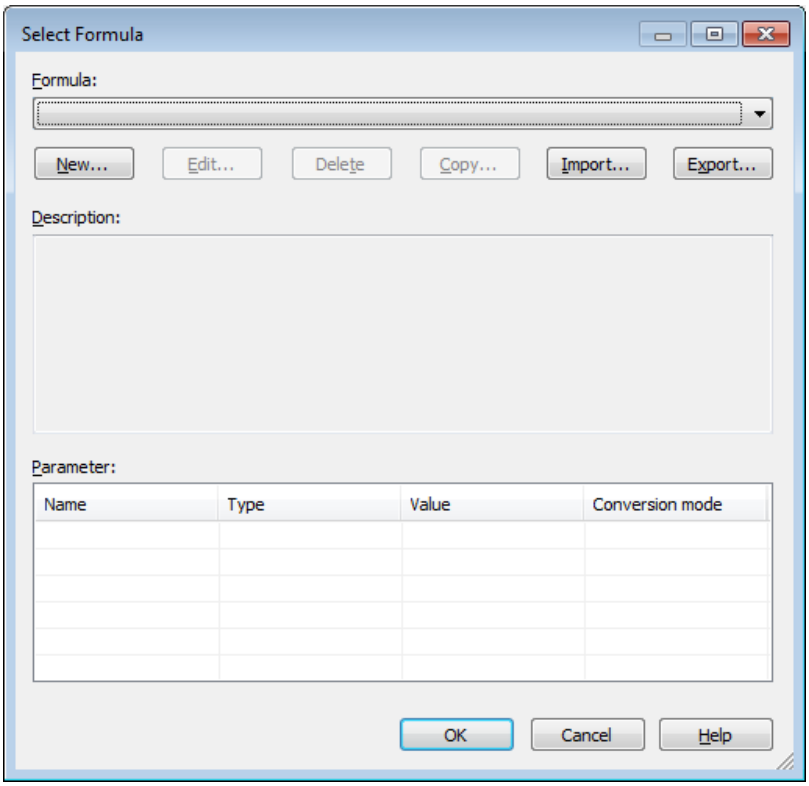
Tip

In the Variables controlbar, the Properties for Calculated Variables dialog opens when you edit an existing calculated variable or define a new one.

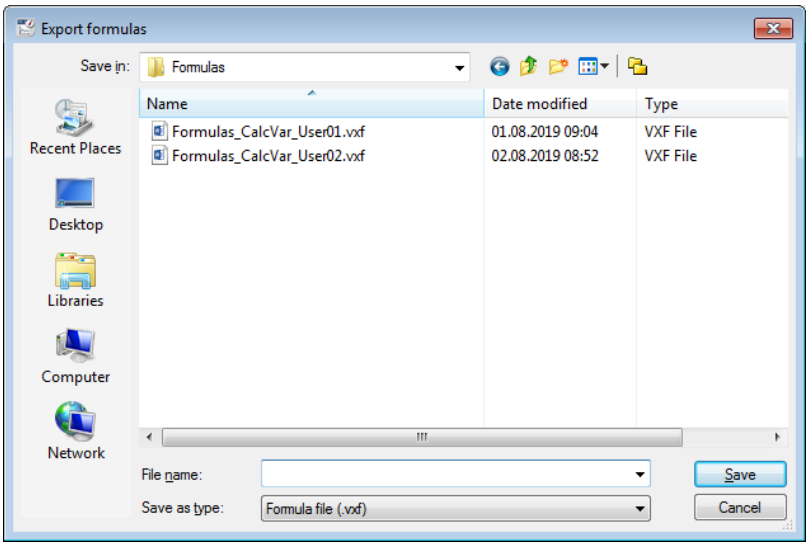
2 Click the Browse button next to the Formula field.



The Select Formula dialog opens.



3 In the Select Formula dialog, click Export.
The Export Formulas dialog opens.



4 Specify the name and path of the VXF file to which you want to save the template formulas.

5 Click Save to export the formulas.

6 To import the formulas, for example, on another host PC, click Import.
The Import Formulas dialog opens.

7 Select the VXF file you have exported the template formulas to.
All the formulas in the VXF file are added to your template formula pool.

Result

You have exported all your template formulas for calculated variables to a VXF file and then imported them to your user-specific template formula pool.

Related topics

Basics

Basics of Formulas for Calculated Variables..... 63

Reference Information

Where to go from here

Information in this section

Variable Management Properties..... 84
Variable Management Commands..... 91

Variable Management Properties

Where to go from here

Information in this section

Properties of Filter Conditions in the Edit Filter Dialog.....84
If you select a filter condition in the Edit Filter dialog, you can specify its properties in the dialog's Filter Condition Details area.

Properties of Variable Descriptions in the Project Controlbar.....85
If you select a variable description in the Project controlbar, its properties are displayed in the Properties controlbar.

Properties of Variables (Variables Controlbar).....86
If you select a variable in the Variables controlbar, its properties are displayed in the Properties controlbar. All the values are read-only except for the weak limits.

Properties of Filter Conditions in the Edit Filter Dialog

Purpose

If you select a filter condition in the Edit Filter dialog, you can specify its properties in the dialog's Filter Condition Details area.

Common

Enabled Lets you enable the selected filter condition.

Inverted Lets you specify to invert the meaning of the selected filter condition.

Logical operator Lets you select AND or OR as the logical operator for the selected filter condition.

Memory Segment Filter Configuration

Memory Segment Lets you select a memory segment for filtering. ControlDesk displays the memory segments of all the variable descriptions in the currently active experiment.

Variable State Filter Configuration

Variable states Lets you select one or more of the following variable states and flags to filter the variable list by.

- States
  - Changed: The variable properties were changed after the related variable description file was added or reloaded.
  - Connected: The variable is connected to an instrument.
  - Measured: The variable is in the measurement signal list.

- New: The variable was created after you added or reloaded the related variable description file.
- Flags
  - Init only: The variable is a fixed parameter. Changing the value of this parameter does not immediately affect the simulation results. The affect occurs only after you stop the simulation and start it again.
  - Read only: The variable is read-only. You cannot change its value.

Variable Type Filter Configuration

Variable types Lets you select one or more variable types to filter the variable list by.

Wildcard Filter Configuration

Column Lets you select a variable list column to which you want to apply the selected wildcard filter.

Match case Lets you specify whether the wildcard filter differentiates between lower and upper case letters.

Wildcard Lets you specify a wildcard string for filtering. You can use the ? wildcard for one missing letter or the * wildcard for an undefined number of missing letters in the wildcard string.

Related topics

HowTos

How to Filter the Variable List 33

Properties of Variable Descriptions in the Project Controlbar

Purpose

If you select a variable description in the Project controlbar, its properties are displayed in the Properties controlbar.

Properties

Note

Which properties are available depends on the specific variable description. Some of the possible properties are described below.

EPROM Identifier (Available only for variable descriptions based on A2L files) Displays the EPROM identifier.

EPROM Identifier Address (Available only for variable descriptions based on A2L files) Displays the address of the EPROM identifier.

Variable description: file date Displays the date when the selected variable description was created.

Variable description: file type Displays the type of the selected variable description.

Variable description: file created by Displays how the selected variable description was created.

Variable description: source Displays the path and file name of the selected variable description.

Related topics

Basics

Introduction to Variable Descriptions..... 10

Properties of Variables (Variables Controlbar)

Purpose

If you select a variable in the Variables controlbar, its properties are displayed in the Properties controlbar. All the values are read-only except for the weak limits.

Address properties

Address Displays the start address of the variable.

Accessibility Displays whether the variable is read-write or read-only.

Symbol Displays the name of the symbol that is assigned to the variable. If a symbol is assigned, it specifies the address of the variable.

Offset If an offset is added to the address of the variable, it is displayed here.

Axis properties

Input quantity Displays the input quantity of the axis.

Monotony Displays the monotonicity of the axis points if defined in the variable description.

Number of axis points Displays the number of points on the axis.

Referenced by (Available only for a common axis) Displays the name of the map or curve that references this axis.

Axis properties (x-axis)

X-axis type Displays the type of the assigned x-axis (standard, fixed, or common axis). Click the Browse button to open the properties dialog of the axis.

X-axis name Displays the name of the assigned axis.

No. of axis points X Displays the number of points on the x-axis.

Input quantity x Displays the input quantity of the x-axis.

Axis properties (y-axis)

Y-axis type Displays the type of the assigned y-axis (standard, fixed, or common axis). Click the Browse button to open the properties dialog of the axis.

Y-axis name Displays the name of the assigned axis.

No. of axis points Y Displays the number of points on the y-axis.

Input quantity y Displays the input quantity of the y-axis.

Bit mask

Bit mask – Start bit Displays the bit offset for the bit mask, counted from the right side of the bit mask.

Bit mask – No. of bits Displays the bit mask width by entering the number of contiguous 1-bits.

Bit mask – Hex Displays the bit mask value in hexadecimal notation.

The following example shows the result of different bit mask properties settings:

Data Type	No. of Bits	Start Bit	Bit Mask	Hex
8-bit	4	0	00001111	F
8-bit	4	2	00111100	3C
16-bit	5	5	0000001111100000	3E0

Computation

Computation method Displays the computation method for converting a source value into a converted value. The conversion rule of the selected computation method is displayed to the right. In case of a multiscaling, the conversion rules of all the subscalings are displayed.

Computation method type Displays the type of the computation method.

Data type properties

Data type Displays the data type (record layout) for the variable.

Byte order Displays whether bytes are stored in Motorola format (big endian, MSB at the memory location with the lowest address) or in Intel format (little endian, LSB at the memory location with the lowest address).

Dimensions (Available only for value blocks and measurement arrays) Displays the number of elements in the array.

- A value block consists of a 1-, or 2-dimensional array of scalar parameters.

- A measurement array consists of a 1-, 2-, or 3-dimensional array of measurement variables.

Fixed-axis properties

Axis points (Available only if a list of axis points is specified in the variable description) Displays the axis points. Empty if the axis points are equidistant and specified via the Offset and Distance properties.

Distance If a distance value is defined for the fixed axis, it is displayed here. If, for example, an offset of 0 and a distance of 3 is defined for an x-axis with 5 axis points, the resulting axis points are: 0, 3, 6, 9, 12. The axis points are computed using the following formula: Xi = Offset + (i - 1) * Distance where i = {1 … NumberOfAxisPointsX}.

Offset If an offset is added to the axis points, it is displayed here.

Shift If a shift value is defined for the fixed axis, it is displayed here. If, for example, an offset of 0 and a shift of 4 is defined for an x axis with 6 axis points, the resulting axis points are: 0, 16, 32, 48, 64, 80. The axis points are computed using the following formula: Xi = Offset + (i - 1) * 2^Shift where i = {1 … NumberOfAxisPointsX}.

Formula properties

The properties of calculated variables in the Properties controlbar are read-only. You can edit a calculated variable via its context menu in the Variables controlbar.

Formula (Available only for calculated variables) Displays the name of the formula that is assigned to the calculated variable.

Description (Available only for calculated variables) Displays the selected formula and its description text.

Input Signals (Available only for calculated variables) Displays the names of the variables that are used as input signals in the selected formula.

General properties

Name Displays the name of the variable.

Display identifier Displays the display identifier of the variable. The display identifier is used for output in graphical user interfaces. If not specified, the name of the variable is used. Whether the display identifier is to be used in the GUI or not can usually be configured in the global options of a tool.

In ControlDesk, you can find this setting on the Visualization page of the ControlDesk Options dialog.

Format Displays the Printf declaration for the displaying of a numerical value (if specified).

The following table shows some of the expressions that are available:

Printf String	Description
`%d`	Decimal integer
`%i`	Decimal integer

Printf String	Description
%u	Unsigned integer
%o	Octal integer
%x, %X	Hexadecimal integer
%b	Binary integer
%f	Floating point
%e, %E	Exponential floating point
%g, %G	Floating point of the formats %f or %e

For more information on format strings, refer to a standard C programming language reference.

Description Displays the description text for the variable (if specified).

Length (Available only for string variables) Displays the maximum number of characters of the selected string variable.

Parent (Available only for standard axes) Displays the name of the map or curve that contains the selected axis.

Unique name Displays the unique name of the variable. The unique name contains the name and the path information of a variable. If no path information is available in the variable description as, for example, in A2L files, Name and Unique name of a variable do not differ.

Unit Displays the unit for the converted value (physical value) of the variable, for example, kg or rad/s (if specified).

Variable type Displays the variable type.

Reference (Available only for a fixed axis) Displays the map or curve that references this axis.

Limits properties (hard limits)

Hard minimum Displays the lower hard limit as the converted and source values.

Hard maximum Displays the upper hard limit as the converted and source values.

Limits properties (weak limits)

Weak minimum (Mandatory. If not specified, a default value is automatically generated.) Displays the lower weak limit as the converted and source values. Lets you specify the converted value. You can only enter values that are inside the hard limits.

Weak maximum (Mandatory. If not specified, a default value is automatically generated.) Displays the upper weak limit as the converted and source values. Lets you specify the converted value. You can only enter values that are inside the hard limits.

Outputs properties

Output parameters (Available only for calculated variables) Displays parameters and writable measurements that are used as output variables. For each output parameter, displays whether the source or converted value is used.

Struct/Struct array properties

(Available only for structs and struct arrays.)

Struct size in bytes Displays the size of a struct or struct array in bytes.

Dimensions Displays the number of structs in a struct array.

- X and Y for a 2-dimensional struct array
- X, Y, and Z for a 3-dimensional struct array

Related topics

Basics

Basics of the Variables Controlbar 30
Handling Variables in the Variables Controlbar 30

Variable Management Commands

Where to go from here

Information in this section

Activate (Variable Filter)....93
To filter the variable list of the **Variables** controlbar according to the selected variable filter.

Activate Variable Description....94
To activate a variable description for the respective platform/device.

Add to Measurement Signal List....96
To measure a variable without visualizing it on a layout.
Not available in operator mode.

Add Variable Description....97
To add a new variable description, and (if applicable) a new ECU Image file, to a platform/device.
Not available in operator mode.

Add Variable to Favorites List....100
To remove the selected variable(s) from the favorites list.

Add/Edit Combined Filter – Filter 1-n / Edit (Variable Filter)....101
To edit a combined filter to filter the variable list using a combination of filter conditions.

Add/Edit Combined Filter – New / Create Variable Filter....102
To create a new combined filter. A combined filter is used to filter the variable list using a combination of filter conditions.

Apply Combined Filter – Filter 1-n....103
To filter the variable list using a combination of filter conditions.

Calculated Variables - Copy....104
To copy a calculated variable.
Not available in operator mode.

Calculated Variables - Edit....105
To edit the properties of a calculated variable.
Not available in operator mode.

Calculated Variables - Export Selected Items....106
To export the selected calculated variables to a VXF file.
Not available in operator mode.

Calculated Variables - Import....107
To import a VXF file with calculated variables.
Not available in operator mode.

Calculated Variables - New....108
To define a new calculated variable.
Not available in operator mode.

Calculated Variables - Remove Selected Items....109
To remove all selected calculated variables.
Not available in operator mode.

Case-Sensitive Search....110
To enable differentiation between upper and lower case letters for search operations.

Collapse All....111
To collapse all the subnodes of a node in the tree.

Customize Columns (Variable List)....111
To add columns to or remove them from the variable list.

Deactivate (Variable Filter)....113
To deactivate a variable filter.

Edit Filter Dialog....114
To specify a combined filter for the variable list of the **Variables** controlbar.

Expand All....115
To expand all the subnodes of a node in the tree.

Favorite List....115
To open or close the **Favorite List**.

Navigate to Referenced Variable....116
To select the referenced variable in the variable list.

No Filter....117
To disable all filters.

Properties (Variable Description)....118
To view the properties of the selected variable description.

Properties for Calculated Variables Dialog....118
Lets you configure the properties of a calculated variable.

Reload (Variable Description)....125
To reload the variable description of the selected platform/device.

Remove (Variable Filter)....127
To remove the selected variable filter.

Remove All (Favorites List)....128
To remove all variables from the Checked Variables list.

Remove from Measurement Signal List....128
To remove the selected variable(s) from the measurement signal list.
Not available in operator mode.

Remove Variable Description....129
To remove a variable description from the experiment.
Not available in operator mode.

Remove Variable from Favorites List....130
To remove the selected variable(s) from the favorites list.

Rename (Variable Filter)..... 130
To rename the selected variable filter.

Replace (Variable Description)..... 131
To replace the variable description of the selected platform/device.

Save Connected Variables in Label List..... 133
To save a label list containing the names of the variables that are connected to instruments.

Select Variable Dialog..... 133
To select a variable from a filtered variable list.

Standard Filter..... 135
To filter the variable list by variable types.

Tree View..... 137
To toggle between flat and hierarchy views of the variables.

Uncheck All..... 138
To clear all checkboxes in the variable list.

Variable Properties / Properties..... 138
To view the properties of a variable (and edit its weak limits).

Variables..... 142
To show the Variables controlbar.

Variables Page..... 146
To specify settings for the Variables controlbar's user interface and the hierarchy view mode.

Wildcard Filter..... 150
To filter the variable list using filter strings that contain wildcards.

Information in other sections

Handling Variables in the Variables Controlbar..... 30

Activate (Variable Filter)

Access

You can access this command via:

Ribbon	None
Context menu of	Project controlbar – variable filter
Shortcut key	None
Icon	None

Purpose

To filter the variable list of the **Variables** controlbar according to the selected variable filter.

Result

The selected filter is applied to the variable list, which now shows the variables that match the stored combination of filter conditions. The tree view is reduced to the relevant nodes, empty nodes are hidden.

Related topics

HowTos

How to Filter the Variable List 33

References

Add/Edit Combined Filter – New / Create Variable Filter 102
Deactivate (Variable Filter) 113
Remove (Variable Filter) 127
Rename (Variable Filter) 130

Activate Variable Description

Access

This command is available only if an experiment is active and an inactive reference to a variable description is selected. Online calibration must be stopped for the selected platform/device. You can access this command via:

Ribbon	None
Context menu of	**Project** controlbar – platform/device – variable description (reference)
Shortcut key	None
Icon	None

Purpose

To activate a variable description for the respective platform/device.

Result

The previously active variable description is deactivated, the selected one is activated, and the variable list is updated to the active variable description.

In the **Project** controlbar, an active variable description is displayed in bold letters with the  symbol next to it.

Description

If you add a new variable description via the Add Variable Description command, it is stored in the `Variable Descriptions` folder of the project. A reference to this project-global variable description is inserted into the project tree under the platform/device node. The new variable description is automatically activated. To change back to an older one later on, you have to activate it manually.

Restoration of variable connections ControlDesk tries to reactivate connections between variables and instruments when you activate another variable description. For reactivation, the variables in the activated variable description must have the same name and data type. For details, refer to Basics of Placing Variables on a Layout (ControlDesk Layouting 🕮).

Note

The 🔗 symbol in the Measurement Configuration and in the Variables controlbar shows if a variable is not visualized in an instrument but is in the measurement signal list. If you activate another variable description *that does not contain this variable*, and then reactivate the original variable description later on, the variable is no longer in the measurement signal list.

To avoid this, visualize the variable in a plotter before you activate another variable description.

When you reactivate the original variable description, ControlDesk restores the connection between the variable and the plotter, and adds the variable to the measurement signal list again.

XCP/CCP service configuration information when activating another variable description If you work with an XCP or CCP device for which the Read XCP settings from A2L or Read CCP settings from A2L property is selected, the XCP/CCP service configuration information is taken from the new variable description. If there are inconsistencies between the ECU interface settings that are currently used by the device and the settings contained in the new variable description, ControlDesk opens a dialog for you to specify whether to use the current value or the value from the new variable description for each of the settings concerned.

Related topics

References

Add Variable Description....97
CCP Settings Advanced Properties (ControlDesk Platform Management 🕮)
Go Offline (ControlDesk Calibration and Data Set Management 🕮)
Stop Online Calibration (for Single Platform/Device) (ControlDesk Calibration and Data Set Management 🕮)
Variable Management Commands....91
XCP/GSI2 Settings Advanced Properties (ControlDesk Platform Management 🕮)

Add to Measurement Signal List

Access

You can access this command via:

Ribbon	None
Context menu of	**Variables** controlbar - variable list
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To measure a variable without visualizing it on a layout.

Result

The variables are shown in the **Measurement Configuration** controlbar and in the **Variables** controlbar with a yellow chain symbol. The yellow chain symbol indicates that a variable is measured without visualization on a layout.

Description

Selected variables (either highlighted or with activated checkbox) can be added to the measurement signal list. In the **Variables** controlbar, checked variables are also shown in the **Favorites** list. If several variables are selected or checked, all of them are added to the measurement signal list.

If you want to measure a variable and to visualize it in on a layout, you have to select Visualize Variables (ControlDesk Layouting 🕮). Visualization on a layout is indicated by a red chain symbol.

Note

You can even add variables to the measurement signal list during a running measurement, but they will not be measured until 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.

Related topics

HowTos

How to Activate Variables for Measurement (ControlDesk Measurement and Recording 🕮)
How to Search for a Variable Using Incremental Search....41

References

Measurement Configuration (ControlDesk Measurement and Recording 🕮)
Remove from Measurement Signal List....128
Variable Management Commands....91

Add Variable Description

Access

This command is available only if an experiment is active and a platform/device is selected. Online calibration must be stopped for the selected platform/device. You can access this command via:

Ribbon	None
Context menu of	Project controlbar – platform/device
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To add a new variable description, and (if applicable) a new ECU Image file, to a platform/device.

Result

ControlDesk successively opens the Select Variable Description dialog and the Select ECU Image File dialog, which allow you to add a variable description for the respective platform/device.

- If you added memory segments to or manually edited memory segments of the previously selected variable description, you are asked whether you want to apply them to the new variable description.

- If you work with an XCP or CCP device for which the **Read XCP settings from A2L** or **Read CCP settings from A2L** property is selected, the XCP/CCP service configuration information is taken from the new variable description. If there are inconsistencies between the ECU interface settings that are currently used by the device and the settings contained in the new variable description, ControlDesk opens a dialog for you to specify whether to use the current value or the value from the new variable description for each of the settings concerned.

The new variable description is activated and the variable list is updated to the new variable description.

Description

A new variable description is stored in the `Variable Descriptions` folder of the project. A reference to this project-global variable description is inserted into the project tree under the platform/device node. It is automatically activated and ControlDesk tries to reactivate connections between variables and instruments. To change to an older variable description later on, you have to activate it manually via the Activate Variable Description command of the reference.

Note

The symbol in the **Measurement Configuration** and in the **Variables** controlbar shows if a variable is not visualized in an instrument but is in the measurement signal list. If you activate another variable description *that does not contain this variable*, and then reactivate the original variable description later on, the variable is no longer in the measurement signal list.
To avoid this, visualize the variable in a plotter before you activate another variable description.
When you reactivate the original variable description, ControlDesk restores the connection between the variable and the plotter, and adds the variable to the measurement signal list again.

Note

There is a migration issue in connection with double slashes in SDF/TRC variable paths in projects initially created with ControlDesk 5.0 or earlier. Refer to Double slashes in SDF/TRC variable paths (ControlDesk Introduction and Overview 🕮).

Select Variable Description dialog

To select a variable description for a platform/device.

Variable description list Displays the variable descriptions (A2L, DBC, SDF, and LDF files) available in the `Variable Descriptions` folder of the ControlDesk project.

Variable descriptions that do not match the selected platform/device type or that are already assigned to another platform/device of the current experiment are not displayed in the list.

Import/Import from file Opens a dialog for you to import a variable description file (A2L, DBC, SDF, LDF file, …). The imported variable description file is added to the list of variable descriptions in the `Variable Descriptions` folder of the ControlDesk project.

If an AUTOSAR or FIBEX file contains the description of multiple clusters, ControlDesk lists all contained clusters. From this list, you can select the one that you want to import. Incompatible clusters are grayed out.

Tip

In the ControlDesk Options dialog, you can specify import options for variable descriptions in the SDF and FIBEX formats. Refer to Variables Page on page 146.

Remove (Available in the context menu of the variable descriptions in the list) Removes the selected variable description from the list of variable descriptions. Only variable descriptions that are not assigned to any platform/device in the project can be removed.

Variable description info field Displays information on the variable description selected in the variable description list.

< Back Opens the previous dialog.

Next > Opens the next dialog.

Finish Confirms the selection and closes the dialog without adding an ECU Image file to the selected variable description. To add an ECU Image file to the selected variable description, click Next > instead.

Cancel Closes the dialog without saving any of your settings.

Select ECU Image File dialog

(Available only for measurement and calibration devices) To specify an ECU Image file for the selected variable description.

Select ECU Image file Enter the path and name of the ECU Image file (HEX, MOT, S19, … file) or select it via the BROWSE button.

ControlDesk can handle ECU Image files that contain code and data or only data.

< Back Opens the previous dialog.

Finish Confirms the specified configuration and closes the dialog.

Cancel Closes the dialog without saving any of your settings.

Related topics

HowTos

How to Add a Variable Description to a Platform/Device 20

References

Activate Variable Description 94
CCP Settings Advanced Properties (ControlDesk Platform Management 🕮)
Go Offline (ControlDesk Calibration and Data Set Management 🕮)
Stop Online Calibration (for Single Platform/Device) (ControlDesk Calibration and Data Set Management 🕮)
XCP/GSI2 Settings Advanced Properties (ControlDesk Platform Management 🕮)

Add Variable to Favorites List

Access

You can access this command via:

Ribbon	None
Context menu of	Variables controlbar
Shortcut key	None
Icon	None

Purpose

To remove the selected variable(s) from the favorites list.

Result

The selected variables are removed from the favorites list and the checkboxes in the variable list are cleared.

Related topics

References

Favorite List 115
Remove Variable from Favorites List 130

Add/Edit Combined Filter – Filter 1-n / Edit (Variable Filter)

Access

You can access this command via:

Ribbon	None

Context menu of	▪ Add/Edit Combined Filter – Filter 1-n command:
	  ▪ Variable list
	  ▪ Hierarchy tree
	▪ Edit (Variable Filter) command:
	Project controlbar - Variable Filters - variable filter

Shortcut key	None
Icon	

Purpose

To edit a combined filter to filter the variable list using a combination of filter conditions.

Result

ControlDesk opens the Edit Filter dialog for you to edit a combined filter.

Edit Filter dialog

Displays a list with the defined filter conditions and lets you configure their settings.

Combined Filter Condition Settings Lets you add a new group or filter condition to the filter conditions list. The following buttons are available.

Button	Description
	Adds the selected element (group or filter condition) to the bottom of the filter condition list. If a group is selected in the filter condition list, the selected element is added to the bottom of the group.
	Moves the selected element down in the filter condition list/group.
	Moves the selected element up in the filter condition list/group.
	Deletes the selected element.

Filter Condition Details Lets you specify the properties of the selected filter condition or group. For details, refer to Properties of Filter Conditions in the Edit Filter Dialog on page 84.

Related topics

HowTos

How to Filter the Variable List....33

References

Activate (Variable Filter)....93
Add/Edit Combined Filter – New / Create Variable Filter....102
Apply Combined Filter – Filter 1-n....103
Remove (Variable Filter)....127
Rename (Variable Filter)....130

Add/Edit Combined Filter – New / Create Variable Filter

Access

You can access this command via:

Ribbon	None

Context menu of	▪ **Add/Edit Combined Filter – New** command:
	  ▪ Variable list
	  ▪ Hierarchy tree
	▪ **Create Variable Filter** command:
	  Project controlbar - **Variable Filters** node

Shortcut key	None
Icon	New Add/Edit

Purpose

To create a new combined filter. A combined filter is used to filter the variable list using a combination of filter conditions.

Result

The filter is applied to the variable list, which now shows the variables that match the combination of filter conditions.

The following results depend on the context in which the filter is created or displayed:

- The new combined filter is stored in the **Variable Filters** folder of the **Project** controlbar.
- The tree view of the variable list is reduced to the relevant nodes, empty nodes are hidden.
- If the filter is active, the color of the variable list entries is blue.
- Above the columns, a header is displayed showing the current filter setting, for example, 'Filter: Wildcard (air*)'.

Description

ControlDesk opens the Edit Filter dialog for you to specify a combined filter.

Edit Filter dialog

Displays a list with the defined filter conditions and lets you configure their settings.

Combined Filter Condition Settings Lets you add a new group or filter condition to the filter conditions list. The following buttons are available.

Button	Description
	Adds the selected element (group or filter condition) to the bottom of the filter condition list. If a group is selected in the filter condition list, the selected element is added to the bottom of the group.
	Moves the selected element down in the filter condition list/group.
	Moves the selected element up in the filter condition list/group.
	Deletes the selected element.

Filter Condition Details Lets you specify the properties of the selected filter condition or group. For details, refer to Properties of Filter Conditions in the Edit Filter Dialog on page 84.

Related topics

HowTos

How to Filter the Variable List....33

References

Activate (Variable Filter)....93
Apply Combined Filter – Filter 1-n....103
Remove (Variable Filter)....127
Rename (Variable Filter)....130

Apply Combined Filter – Filter 1-n

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	

Purpose

To filter the variable list using a combination of filter conditions.

Result

The selected filter is applied to the variable list, which now shows the variables that match the stored combination of filter conditions. The tree view is reduced to the relevant nodes, empty nodes are hidden.

Description

The displayed filters are stored in the **Variable Filters** folder of the Project controlbar.

Related topics

HowTos

How to Filter the Variable List 33
How to Use the Function Buttons for Searching and Filtering Variables 37

References

Add/Edit Combined Filter – Filter 1-n / Edit (Variable Filter) 101
Add/Edit Combined Filter – New / Create Variable Filter 102
Remove (Variable Filter) 127
Rename (Variable Filter) 130

Calculated Variables - Copy

Access

This command is available only if a calculated variable is selected. You can access it in the **Variables** controlbar via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To copy a calculated variable.

Result

In the variable list, ControlDesk creates a new calculated variable with the same properties as the selected one.

Description

You can use the copy to easily create a calculated variable similar to the selected one.

Related topics

Basics

Basics of Formulas for Calculated Variables 63

References

Variable Management Commands 91

Calculated Variables - Edit

Access

This command is available only if a calculated variable is selected. You can access it in the **Variables** via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To edit the properties of a calculated variable.

Result

ControlDesk opens the **Properties for Calculated Variables** dialog and lets you change the variable properties. Refer to Properties for Calculated Variables Dialog on page 118.

Description

You can change the properties of the selected calculated variable.

Related topics

Basics

Basics of Formulas for Calculated Variables....63

HowTos

How to Define a Calculated Variable....55

References

Variable Management Commands....91

Calculated Variables - Export Selected Items

Access

This command is available only if a calculated variable is selected. You can access it in the Variables via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To export the selected calculated variables to a VXF file.

Description

The calculated variables are exported together with the associated formulas.

Tip

Calculated variables and their formulas are saved, exported, or imported when the experiment containing them is saved, exported or imported. If you want to reuse the calculated variables and/or their formulas in a different experiment, you need to export and import them specifically.

Related topics

Basics

Basics of Formulas for Calculated Variables....63

HowTos

How to Export and Import Calculated Variables....78

References

Variable Management Commands....91

Calculated Variables - Import

Access

You can access this command in the **Variables** controlbar via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To import a VXF file with calculated variables.

Result

The calculated variables are imported to the active variable description.

Description

The name of the imported variable is extended by a # and a number, if a calculated variable with the same name already exists on the same level of the variable description.

When importing a calculated variable, ControlDesk tries to find input signals via the device name and the variable name. If the search fails, ControlDesk starts a

new search, using the variable name only. The following list shows the search sequences for devices and for platforms.

- Device: device + variable name, variable name only
- Platform: platform + path + variable name, path + variable name, variable name only

Tip

The mapping of input signals and variables is stored in ControlDesk's log file. You can use ControlDesk's **Messages** controlbar to display it.

If no variable is found, the calculated variable is marked as disabled in the plotter legend:  .

Related topics

Basics

Basics of Formulas for Calculated Variables....63

HowTos

How to Export and Import Calculated Variables....78

References

Variable Management Commands....91

Calculated Variables - New

Access

You can access this command in the **Variables** controlbar via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To define a new calculated variable.

Result

ControlDesk opens the Properties dialog for a calculated variable and lets you specify the variable properties. Refer to Properties for Calculated Variables Dialog on page 118.

Description

In the **Variables** controlbar, the calculated variable (or a reference to it) is added under the selected node of the variable list.

Related topics

Basics

Basics of Formulas for Calculated Variables....63

HowTos

How to Define a Calculated Variable....55

References

Variable Management Commands....91

Calculated Variables - Remove Selected Items

Access

This command is available only if a calculated variable is selected. You can access it in the **Variables** controlbar via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To remove all selected calculated variables.

Related topics

Basics

Basics of Formulas for Calculated Variables....63

References

Variable Management Commands....91

Case-Sensitive Search

Access

You can access this command via:

Ribbon	None
Context menu of	Variable list
Shortcut key	None
Icon	None

Purpose

To enable differentiation between upper and lower case letters for search operations.

Description

If case-sensitive search is disabled, you will find the "MAP1" string with the "map1" search string. If it is enabled, you must enter "MAP1" to find it.

Related topics

HowTos

How to Search for a Variable Using Incremental Search....41

References

Variable Management Commands....91

Collapse All

Access

You can access this command via:

Ribbon	None
Context menu of	**Variables** controlbar
Shortcut key	None
Icon	None

Purpose

To collapse all the subnodes of a node in the tree.

Related topics

References

Expand All..... 115

Customize Columns (Variable List)

Access

You can access this command via:

Ribbon	None
Context menu of	Variable list (column header)
Shortcut key	**Ctrl+Shift+C**
Icon	None

Purpose

To add columns to or remove them from the variable list.

Result

The added columns appear in the variable list, the removed columns are hidden.

Description

Whether or not the column contains helpful information depends on how the variable is described in the variable description file.

Customize Columns dialog

The columns list contains all the columns that can be displayed in the variable list. You can add a column to the variable list by selecting its checkbox. To remove a column, clear its checkbox.

Note

Which of the following dialog elements and options are available depends on the context in which you opened the dialog.

Column list The following columns are available:

Column	Description
Address	Displays the base address of the variable on the hardware or VPU.
Block	Displays the name of the block that contains the variable (real-time model).
Computation Method	Displays the name of the computation method that is used for converting a source value into a converted value.
Connected	Displays a symbol indicating the connection status of the variable. ▪  indicates that the variable is connected to an instrument. ▪  indicates that the variable is not connected to an instrument, but is in the measurement signal list. ▪ If no symbol is displayed, the variable is not connected to an instrument and not in the measurement signal list.
Data Type	Displays the name of the data type used in the variable description (typedef alias name, for example the name of the record layout in an A2L file).
Description	Displays the comment that is assigned to the item in the variable description.
Favorite	▪ Variable list: Lets you specify the variable as a favorite variable by activating a checkbox. All selected variables are visible in the favorites list. ▪ Favorites list: Displays the name of the variable.
Group	(Available only in the tree view) Displays the name of the group node in the tree.
Limits	Displays the weak limits in the format <minimum> | <maximum>. If no weak limits are defined, the limits of the data type are displayed. If the variable list is sorted by limits, the minimum value is evaluated first. The minimum and maximum are displayed as converted values. Negative values are therefore possible for unsigned integer data types depending on the computation method.
Offset	If an offset is added to the address of the variable, it is displayed here.
Origin	Displays the Simulink label of the variable, and the variable's path in the Simulink model.
Path	Displays the path of the variable.
Platform/Device	Displays the platform/device name.
Referenced Variable	If the variable references another variable, the path of the referenced variable is displayed here.
Size	Displays the dimensions of the variable, which is especially necessary for nonscalar variables.
Start Bit	Displays the bit offset for the bit mask, counted from the right side of the bit mask.
Sub Groups	(Available only in the tree view) Displays the number of subgroups a group has on the next lower tree level.
Symbol	Displays the name of the symbol that is assigned to the variable. If a symbol is assigned, it specifies the address of the variable.
Type	Displays the data type of the variable.
Unit	Displays the unit of the variable.

Column	Description
Variable	Displays the name of the variable.
Variable Description	(Available only in the favorites list) Displays the name of the variable description the variable originates from.
Variable Type	Displays the type of the variable.

You can use the following commands/settings to configure the columns.

Hide Lets you hide the column in the variable list.

Move Down Moves the selected column down the list (to the right in the variable list).

Move Up Moves the selected column up the list (to the left in the variable list).

Show Lets you display the column in the variable list.

Visible Lets you display/hide the column in the variable list by activating a checkbox.

Width / Width of the selected column (in pixels) Lets you enter a width for the selected column (in pixels).

Related topics

References

Variable Management Commands.....91

Deactivate (Variable Filter)

Access

This command is available only if the selected variable filter is currently active. You can access this command via:

Ribbon	None
Context menu of	Project controlbar – (active) variable filter
Shortcut key	None
Icon	None

Purpose

To deactivate a variable filter.

Related topics

HowTos

How to Filter the Variable List 33

References

Add/Edit Combined Filter – New / Create Variable Filter 102
Remove (Variable Filter) 127
Rename (Variable Filter) 130

Edit Filter Dialog

Access

This dialog opens when you select one of the following commands:

- Add/Edit Combined Filter – New / Create Variable Filter on page 102
- Add/Edit Combined Filter – Filter 1-n / Edit (Variable Filter) on page 101

Edit Filter dialog

Displays a list with the defined filter conditions and lets you configure their settings.

Combined Filter Condition Settings Lets you add a new group or filter condition to the filter conditions list. The following buttons are available.

Button	Description
	Adds the selected element (group or filter condition) to the bottom of the filter condition list. If a group is selected in the filter condition list, the selected element is added to the bottom of the group.
	Moves the selected element down in the filter condition list/group.
	Moves the selected element up in the filter condition list/group.
	Deletes the selected element.

Filter Condition Details Lets you specify the properties of the selected filter condition or group. For details, refer to Properties of Filter Conditions in the Edit Filter Dialog on page 84.

Related topics

HowTos

How to Filter the Variable List....33

References

Activate (Variable Filter)....93
Apply Combined Filter – Filter 1-n....103
Remove (Variable Filter)....127
Rename (Variable Filter)....130

Expand All

Access

You can access this command via:

Ribbon	None
Context menu of	**Variables** controlbar
Shortcut key	None
Icon	None

Purpose

To expand all the nodes and subnodes of the tree.

Related topics

References

Collapse All....111

Favorite List

Access

You can access this command via:

Ribbon	None
Context menu of	**Variables** controlbar
Shortcut key	None
Icon	None

Purpose

To open or close the Favorite List.

Description

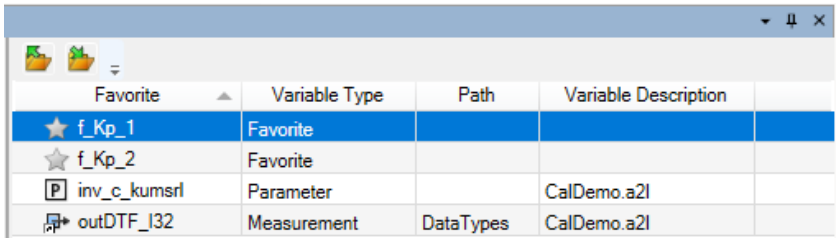
The Favorite List displays all the variables that are selected in the Favorite column of the variable list.

The following function buttons are available.

	Opens a dialog to import the variables of a text (TXT) or label list (LAB) file to the favorites list.
	Opens a dialog to export the variables of the favorites list to a text (TXT) or label list (LAB) file.

- If an imported variable in the Favorite list is not part of the variables list, it gets the Label variable type and the  icon. If you load a variable description with an appropriate variable, the favorite status is reestablished.
- If a variable in the Favorite list has no associated variable description, e.g., because the platform or device was removed from the experiment, it gets the  icon.

You can customize the Favorite list via the header's context menu to add the Variable Description column that shows associated variable descriptons.



Favorite	Variable Type	Path	Variable Description
f_Kp_1	Favorite		
f_Kp_2	Favorite		
inv_c_kumsrl	Parameter		CalDemo.a2l
outDTF_I32	Measurement	DataTypes	CalDemo.a2l

Related topics

References

Add Variable to Favorites List..... 100
Remove Variable from Favorites List..... 130

Navigate to Referenced Variable

Access

You can access this command via:

Ribbon	None
Context menu of	Variables controlbar
Shortcut key	**Ctrl+R**
Icon	None

Purpose

To select the referenced variable in the variable list.

Description

If a variable is a reference to a variable residing in another node of the variable description, the variable's icon has an additional ↗ symbol. The path of the referenced variable is displayed in the **Referenced Variable** column of the variable list.

Result

The referenced variable is selected in the variable list.

Related topics

Basics

Basics of Placing Variables on a Layout (ControlDesk Layouting 🕮)

No Filter

Access

You can access this command in the **Variables** controlbar and in the **Select Variables** dialog via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	`Ctrl+Shift+A`
Icon	

Purpose

To disable all filters.

Result

All variables are shown in the variable list.

Related topics

HowTos

How to Filter the Variable List.....33

How to Use the Function Buttons for Searching and Filtering Variables..... 37

References

Variable Management Commands..... 91

Properties (Variable Description)

Access

You can access this command via:

Ribbon	None
Context menu of	**Project** controlbar – variable description
Shortcut key	**Enter**
Icon	None

Purpose

To view the properties of the selected variable description.

Result

The variable description properties are displayed in the **Properties** controlbar. You can also change the properties using the controlbar.

For reference information on the properties, refer to Properties of Variable Descriptions in the Project Controlbar on page 85.

Related topics

References

Properties of Variable Descriptions in the Project Controlbar..... 85

Properties for Calculated Variables Dialog

Access

This dialog opens when you create or edit a calculated variable.

Purpose

Lets you configure the properties of a calculated variable.

Result

The new calculated variable or the changes are added to the variable list.

General

Name (Mandatory) Lets you specify the name of the variable. The name you enter must not contain the following character: /.

Note

You can specify a variable name containing space characters. Exporting a variable with a name containing space characters to an A2L file, however, is not possible.

Display identifier (Optional) Lets you specify the DISPLAY_IDENTIFIER of the variable. The display identifier is used for output in graphical user interfaces. If not specified, the name of the variable is used. Whether the display identifier is to be used in the graphical user interface or not can usually be configured in the global options of a tool. You can find this setting on the Visualization page of the ControlDesk Options dialog.

Format string (Optional) Lets you specify to display a numerical value using a Printf declaration of the C programming language. The default format string is %g.

The following table shows some of the expressions that are available:

Printf String	Description
%d	Decimal integer
%i	Decimal integer
%u	Unsigned integer
%o	Octal integer
%x, %X	Hexadecimal integer
%b	Binary integer
%f	Floating point
%e, %E	Exponential floating point
%g, %G	Floating point of the formats %f or %e

For more information on format strings, refer to a standard C programming language reference.

Description (Optional) Lets you specify a description text for the variable.

Unit (Optional) Lets you specify a unit for the converted value (physical value) of the variable, for example, kg or rad/s.

Data type

Data type (Mandatory) Lets you select a data type (record layout) for the variable.

If you select another record layout, the limit properties (see below) are automatically adapted.

Byte order (Mandatory) Lets you select whether to store bytes in Motorola format (big endian, MSB at the memory location with the lowest address) or in Intel format (little endian, LSB at the memory location with the lowest address).

Bit mask

Bit mask – Start bit (Mandatory) Lets you enter a bit offset for the bit mask, counted from the right side of the bit mask.

Bit mask – No. of bits (Mandatory) Lets you specify the bit mask width by entering the number of contiguous 1-bits.

Bit mask – Hex (Read-only) Displays the bit mask value in hexadecimal notation.

The following example shows the result of different bit mask properties settings:

Data Type	No. of Bits	Start Bit	Bit Mask	Hex
8-bit	4	0	00001111	F
8-bit	4	2	00111100	3C
16-bit	5	5	0000001111100000	3E0

Computation

Computation method (Mandatory) Lets you select a computation method for converting a source value into a converted value. The conversion rule of the selected computation method is displayed to the right.

In addition to the computation methods defined in the variable description file, the Variable Editor provides the default computation method __Identity. Select this computation method if the converted and the source value have to be equal.

If you select another computation method, the limit properties (see below) are automatically adapted.

Limits (weak limits)

Min weak (phys.) (Mandatory. If not specified, a default value is automatically generated.) Lets you specify the lower weak limit of the physical value. You can only enter values that are inside the hard limits.

Hex (Read-only) Displays the source value in hexadecimal notation.

Max weak (phys.) (Mandatory. If not specified, a default value is automatically generated.) Lets you specify the upper weak limit of the physical value. You can only enter values that are inside the hard limits.

Hex (Read-only) Displays the source value in hexadecimal notation.

Note

The limits are automatically adapted, if you perform one of the following actions:

- You select another record layout or change the data type of a structure element.
- You select another computation method or change the conversion rule.

Formula

Formula (Mandatory) Displays the name of the formula that is assigned to the calculated variable. Click the Browse button to select a formula via the Select Formula dialog. You can also use the **Select Formula** dialog to delete, copy, and edit formulas.

Input signals Displays the names of the variables that are used as input signals in the selected formula.

Description Displays the selected formula and its description text.

Outputs

Output parameters Lets you specify parameters and writable measurements as output variables. The drop-down list lets you specify whether to use the source or converted value for the next output variable you add. Click the Browse button to select a parameter or a writable measurement via the **Select Variable** dialog. For more information on the dialog, refer to Select Variable Dialog on page 133.

Select Formula dialog

Lets you select a formula for the calculated variable and specify the parameters.

Formula Lets you select a formula.

New Opens the Formula Editor dialog, which lets you define a new formula.

Edit Opens the Formula Editor dialog to change the selected formula.

Delete Deletes the selected formula.

Copy Opens the Formula Editor dialog with a copy of the selected formula.

Import Opens the **Import Formulas** dialog for you to select an VXF file containing exported formulas.

Export Opens the **Export Formulas** dialog for you to export all the formulas in the **Formula** list.

Description Displays the selected formula and its description text.

Parameter table Displays the connections of variables or constants to the parameters of the formula and lets you edit them.

Column	Description
Name	Displays the name of the parameter.
Type	Lets you select the type of the connection: ▪ Select **Signal** to connect a measurement variable, scalar parameter variable, or existing calculated variable to the parameter. Click the **Value** column and use the **Browse** button to select or change a connection. Clicking the **Browse** button opens the **Select Variable** dialog for you to select a variable to assign it to the parameter of the formula. The variable list in the dialog is filtered and shows only variables that can be used as input signals for calculated variables: ▪ Measurement variable (can also be a subelement of a measurement array or a structure)

Column	Description
	▪ Scalar parameter (can also be a subelement of a value block or a structure) ▪ Calculated variable For more information on the dialog, refer to Select Variable Dialog on page 133. ▪ Select **Value** to connect a constant value to the parameter. Enter the constant value in the Value column.
Value	Lets you change the connected measurement variable or enter a constant value for the parameter (depending on the selection in the **Parameter – Type** column).
Conversion	Lets you specify whether the source value or the converted value of a signal is to be used for computing the calculated variable. For constant values, **Parameter – Conversion** is set to Source (read-only).

Note

The conversion that you select in the **Parameter – Conversion** column is part of the formula for the calculated variable. The formula of the computation method is applied to the total result of the variable computation afterwards.

Formula Editor dialog

Lets you define a new formula that can be assigned to calculated variables.

Name Lets you enter the name of the formula. The formulas for calculated variables are stored globally for all projects on a host PC.

The formula name must therefore be unique on your host PC.

Description Lets you enter a description text for the formula.

Formula Lets you enter the formula. You can use the buttons below the field to enter operators and functions. You must use the Parameter list below the field to enter a parameter.

Operators Click an operator to insert it into the formula at the position of the text cursor.

Operator Button	Formula Example	Description
+	Param1 + Param2	Adds the operands.
-	Param1 - Param2	Subtracts the right-hand operand from the left-hand operand.
*	Param1 * Param2	Multiplies the operands.
/	Param1 / Param2	Divides the left-hand operand by the right-hand operand.
( )	Param1 * (Param2 + Param3)	Specifies the order of operations.
<	Param1 < 0 ? Param1 : Param2	Checks if the value of the left operand is less than the value of the right operand, if yes the condition becomes true.
>	Param1 > 0 ? Param1 : Param2	Checks if the value of the left operand is greater than the value of the right operand, if yes the condition becomes true.
%	Param2 % 5	Performs a modulo operation. Divides the left-hand operand by the right-hand operand and returns the remainder.

Operator Button	Formula Example	Description
<=	Param1 <= 0 ? Param1 : Param2	Checks if the value of the left operand is less than or equal to the value of the right operand, if yes the condition becomes true.
>=	Param1 >= 0 ? Param1 : Param2	Checks if the value of left operand is greater than or equal to the value of the right operand, if yes the condition becomes true.
x^y (button)	Math.Pow((double)Param1 ,2)	Performs an exponential calculation on the operands. The operands must have the double data type to pass the formula check. If you click the x^y button, Math.Pow(,2) for an exponential calculation is inserted in the Formula field. If you want to use ^ as bitwise XOR operator, enter it manually (see below).
x^y (entered manually)	((Int32)Param1) ^ ((Int32)Param2)[1)] Example: Param1 = 0011 1100 Param2 = 0000 1101 Result = 0011 0001	Performs a bitwise XOR operation. A bit is set in the result if it is set in one of the operands but not the other.
<<	((Int32)Param1) << 2[1)] Example: Param1 = 0011 1100 Result = 1111 0000	Performs a bitwise left-shift operation. The value of the left operand is moved left by the number of bits specified by the right operand.
>>	((Int32)Param1) >> 2[1)] Example: Param1 = 0011 1100 Result = 0000 1111	Performs a bitwise right-shift operation. The value of the left operand is moved right by the number of bits specified by the right operand.
~	~((Int32)Param1)[1)] Example: Param1 = 0011 1100 Result = 1100 0011	Performs a bitwise complement operation. Inverts all the bits in the binary representation of the value (swapping 0's for 1's and vice versa).

[1)] The operands of a bitwise operation must have the Int32 data type to pass the formula check.

Functions Click a function button to enter it in the formula at the position of the text cursor.

Statistical and history functions have an index parameter that lets you specify the included values. The following table shows the available function buttons.

Function Button	Formula Example	Description
abs	abs(Param1)	Computes the absolute value of the input signal.
acos	acos(Param1)	Computes the arccos value of the input signal.
asin	asin(Param1)	Computes the arcsine value of the input signal.
atan	atan(Param1)	Computes the arctangent value of the input signal.
avg	average(Param1, 4)	Computes the average value of a number of values of an input signal. The length value specifies the number of most recent values of the input signal to be used in the calculation (in the example: the last 4 values of the input signal). The length value must be greater than 0 and must not exceed 100.

Function Button	Formula Example	Description
ceil	ceil(Param1)	Computes the smallest integer that is higher than or equal to the input signal.
cos	cos(Param1)	Computes the cosine value of the input signal.
exp	exp(Param1)	Computes the exponential value of the input signal.
floor	floor(Param1)	Computes the highest integer that is less than or equal to the input signal.
log	log(Param1)	Computes the logarithm of the input signal.
mean	mean(Param1, 4)	Computes the arithmetic middle value of a number of values of an input signal (moving average). The length value specifies the number of most recent values of the input signal to be used in the calculation (in the example: the last 4 values of the input signal). The length value must be greater than 1 and must not exceed 100.
med	median(Param1, 7)	Computes the median value of a number of most recent values of an input signal. The median is the number in the middle of a set of numbers; that is, half the numbers have values that are greater than the median, and half have values that are less. If there is an even number of numbers in the set, the median is the average of the two numbers in the middle. The length value specifies the number of most recent values of the input signal to be used in the calculation (in the example: the last 7 values of the input signal). The length value must be greater than 1 and must not exceed 100.
max	max(Param1,5)	Computes the maximum value of a number of values of an input signal. The length value specifies the number of most recent values of the input signal to be used in the calculation (in the example: the last 5 input signals). The length value must be greater than 1 and must not exceed 100.
min	min(Param1, 5)	Computes the minimum value of a number of values of an input signal. The length value specifies the number of most recent values of the input signal to be used in the calculation (in the example: the last 5 values of the input signals). The length value must be greater than 1 and must not exceed 100.
sin	sin(Param1)	Computes the sine value of the input signal.
sqrt	sqrt(Param1)	Computes the square root of the input signal.
sum	sum(Param1, 10)	Computes the sum of a number of values of an input signal. The length value specifies the number of most recent values of the input signal to be used in the calculation (in the example: the last 10 input signals). The length value must be greater than 1 and must not exceed 100.
tan	tan(Param1)	Computes the tangent value of the input signal.
time	t(Param1, 5)	Uses the value on an input signal's time line (the timestamp at a given index). The index value specifies the number of values to go back in the value history (in the example: the sixth value back in the value history). The index is zero-based, so the latest or current value has index 0. The index value must not exceed 100.

Function Button	Formula Example	Description
[- n]	h(Param1, 3)	Uses the value of an input signal in the past (value history). The index value specifies the number of measurement points to trace back in the value history (in the example: the fourth value back in the value history). The index is zero-based, so the latest or current value has index 0. The index value must not exceed 100.

Tip

The sign of the index parameter does not matter. You can use a negative sign, if you want to emphasize going back in value history, for example: `t(Param1, -5)`.

Parameters Displays a list of parameters. Double-click a parameter to insert it into the formula at the position of the text cursor. To change the name of a parameter, select it and click it once. If you use the parameter in a statistical or a value history function the number of values to be buffered is displayed.

Parameters – Add Adds a new parameter to the list of parameters.

Parameters – Remove Removes a parameter from the list. This does not remove a parameter from a formula.

Check Checks the syntax of the formula and performs a calculation test.

OK Checks the syntax of the formula and performs a calculation test. Saves the formula and closes the dialog.

Cancel Closes the dialog without saving the formula.

Related topics

References

Visualization Page (ControlDesk Layouting 🕮)

Reload (Variable Description)

Access

This command is available only if an experiment is active and an active reference to a variable description is selected. Online calibration must be stopped for the selected platform/device. You can access it via:

Ribbon	None
Context menu of	Project controlbar – platform/device – variable description (reference)
Shortcut key	None
Icon	None

Purpose

To reload the variable description of the selected platform/device.

Description

Reimports the active variable description, using the path of the initially imported variable description file. Variable connections of instruments are restored (if possible). To reload/replace a variable description, the experiment must be saved.

Tip

If ControlDesk cannot find the initially imported variable description, it displays an error message that contains the source path. If you know that the variable description has a new path, use Replace (Variable Description) on page 131 to reach your destination faster.

Reloading the variable description of a platform/device actually means reloading the associated project-global variable description. Reloading a variable description that is referenced in more than one experiment affects all the references to it in all the experiments of a ControlDesk project.

If you added memory segments to or manually edited memory segments of the variable description, you are asked whether you want to apply them to the reloaded variable description. If you click **Yes**, they are added to the variable description again. Otherwise, they are removed from the variable description for the current experiment.

If you work with an XCP or CCP device for which the **Read XCP settings from A2L** or **Read CCP settings from A2L** property is selected, the XCP/CCP service configuration information is taken from the new variable description. If there are inconsistencies between the ECU interface settings that are currently used by the device and the settings contained in the new variable description, ControlDesk opens a dialog for you to specify whether to use the current value or the value from the new variable description for each of the settings concerned.

This command makes it easy to keep the variable description up to date. In prototyping experiments, for example, you can use this command to handle the frequent changes in the platform's variable description.

Tip

ControlDesk can detect a new real-time application image if it was built using an RTI version from dSPACE Release 7.4 or later. In the **ControlDesk Options** dialog, you can specify to reload the variable description file automatically in such cases. For details, refer to Variables Page on page 146.

Restoring data sets when reloading/replacing a variable description In ControlDesk's **Options** dialog, you can specify whether to remove or to restore data sets automatically after reloading or replacing a variable description. Refer to Data Set Management Page (ControlDesk Calibration and Data Set Management 🕮).

Note

There is a migration issue in connection with double slashes in SDF/TRC variable paths in projects initially created with ControlDesk 5.0 or earlier. Refer to Double slashes in SDF/TRC variable paths (ControlDesk Introduction and Overview 🕮).

Related topics

References

CCP Settings Advanced Properties (ControlDesk Platform Management 🕮)
Go Offline (ControlDesk Calibration and Data Set Management 🕮)
Replace (Variable Description)..... 131
Stop Online Calibration (for Single Platform/Device) (ControlDesk Calibration and Data Set Management 🕮)
Variable Management Commands..... 91
XCP/GSI2 Settings Advanced Properties (ControlDesk Platform Management 🕮)

Remove (Variable Filter)

Access

You can access this command via:

Ribbon	None
Context menu of	Project controlbar – variable filter
Shortcut key	None
Icon	None

Purpose

To remove the selected variable filter.

Related topics

HowTos

How to Filter the Variable List..... 33

References

Activate (Variable Filter)..... 93
Add/Edit Combined Filter – New / Create Variable Filter..... 102
Apply Combined Filter – Filter 1-n..... 103
Rename (Variable Filter)..... 130

Remove All (Favorites List)

Access

You can access this command via:

Ribbon	None
Context menu of	Variables controlbar - favorites list
Shortcut key	**Delete** (if all list entries are selected)
Icon	None

Purpose

To remove all variables from the Favorites list.

Result

All variables are removed from the Favorites list and the checkboxes in the variable list are cleared.

Related topics

References

Variable Management Commands.....91

Remove from Measurement Signal List

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Variables controlbar - variable list ▪ Measurement Configuration controlbar - measurement signal list ▪ Measurement Configuration controlbar - recorder signal list
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To remove the selected variable(s) from the measurement signal list.

Result

The selection is removed from the measurement signal list in the **Measurement Configuration** controlbar.

Description

All the variables you have selected (either by highlighting or by activating the checkbox) are removed from the measurement signal list. The checkboxes in the variable list are then cleared.

You can also remove variables from the measurement signal list via the context menu of the measurement signal list. Refer to Remove (Variable) (ControlDesk Measurement and Recording 🕮).

Related topics

HowTos

How to Activate Variables for Measurement (ControlDesk Measurement and Recording 🕮)

References

Add to Measurement Signal List....96
Measurement Configuration (ControlDesk Measurement and Recording 🕮)
Variable Management Commands....91

Remove Variable Description

Access

You can access this command via:

Ribbon	None
Context menu of	▪ **Project** controlbar – active experiment – platform/device – variable description reference (not activated) ▪ **Project** controlbar - **Variable Descriptions** – variable description (not activated in an experiment)
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To remove a variable description from the experiment.

Result

The reference to the variable description is removed from the experiment. If the variable description is not referenced by other items in the project, it is also removed from the project and the associated variable description file(s) are deleted from the file system.

Related topics

References

Variable Management Commands....91

Remove Variable from Favorites List

Access

You can access this command via:

Ribbon	None
Context menu of	Variables controlbar
Shortcut key	**Delete**
Icon	None

Purpose

To remove the selected variable(s) from the favorites list.

Result

The selected variables are removed from the Favorites list and the checkboxes in the variable list are cleared.

Related topics

References

Add Variable to Favorites List....100
Favorite List....115

Rename (Variable Filter)

Access

You can access this command via:

Ribbon	None
Context menu of	Project controlbar – variable filter

Shortcut key	**F2**
Icon	None

Purpose

To rename the selected variable filter.

Description

The listed filters are displayed as combined filters in the Variables controlbar.

Result

The Rename Variable Filter dialog opens for you to rename the selected variable filter and the corresponding VFE file.

Rename Variable Filter dialog

To specify another name for a combined variable filter.

Enter a unique name for the variable filter Lets you change the name of the variable filter. The name you enter must not contain any of the following characters: `* ? | < > : / \ "`.

Related topics

HowTos

How to Filter the Variable List 33

References

Activate (Variable Filter) 93
Apply Combined Filter – Filter 1-n 103
Remove (Variable Filter) 127

Replace (Variable Description)

Access

This command is available only if an experiment is active and an active reference to a variable description is selected. Online calibration must be stopped for the selected platform/device. You can access it via:

Ribbon	None
Context menu of	Project controlbar – platform/device – variable description (reference)
Shortcut key	None
Icon	None

Purpose

To replace the variable description of the selected platform/device.

Description

Replaces the active variable description with a new one. The new variable description file can have a different path and a different name than the initially imported one but it must be suitable for the selected platform/device. Variable connections of instruments are restored (if possible). To reload/replace a variable description, the experiment must be saved.

Replacing the variable description of a platform/device actually means replacing the associated project-global variable description. Replacing a variable description that is referenced in more than one experiment affects all the references to it in all the experiments of a ControlDesk project.

If you added memory segments to, or manually edited memory segments of, the variable description, you are asked whether you want to apply them to the replaced variable description. If you click **Yes**, they are added to the variable description again. Otherwise, they are removed from the variable description for the current experiment.

Restoring data sets when reloading/replacing a variable description In ControlDesk's **Options** dialog, you can specify whether to remove or to restore data sets automatically after reloading or replacing a variable description. Refer to Data Set Management Page (ControlDesk Calibration and Data Set Management 🕮).

Note

There is a migration issue in connection with double slashes in SDF/TRC variable paths in projects initially created with ControlDesk 5.0 or earlier. Refer to Double slashes in SDF/TRC variable paths (ControlDesk Introduction and Overview 🕮).

Related topics

References

Go Offline (ControlDesk Calibration and Data Set Management 🕮)
Reload (Variable Description)..... 125
Stop Online Calibration (for Single Platform/Device) (ControlDesk Calibration and Data Set Management 🕮)

Save Connected Variables in Label List

Access

You can access this command via:

Ribbon	None
Context menu of	▪ **Variables** controlbar - variable list ▪ **Variables** controlbar - hierarchy tree
Shortcut key	None
Icon	None

Purpose

To save a label list containing the names of the variables that are connected to instruments.

Result

ControlDesk opens a standard **Save As** dialog for you to save the label list.

Related topics

HowTos

How to Activate Variables for Measurement (ControlDesk Measurement and Recording 🕮)

References

Variable Management Commands.....91

Select Variable Dialog

Access

This dialog opens when you choose to select a variable in one of the following commands or dialogs:

- Trigger Rules (ControlDesk Measurement and Recording 🕮)
- Properties for Calculated Variables Dialog on page 118
- Select Formula dialog on page 121
- Variable(s) - Add (ControlDesk Instrument Handling 🕮)

Purpose

Lets you select a variable from a variable list. The variable list resembles the list in the **Variables** controlbar and offers similar functions for searching and filtering.

Dialog settings

Note

Which of the following dialog elements and options are available depends on the context in which you opened the dialog.

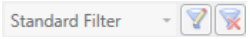
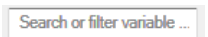
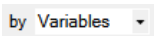
Variable list The variable list displays the variables of the variable description. A Subset of the variable information is displayed in predefined columns.

Predefined Columns	Description
Variable Type	Displays the type of the variable.
Connected	Displays a symbol indicating the connection status of the variable. ▪ indicates that the variable is connected to an instrument. ▪ indicates that the variable is not connected to an instrument. The variable is in the measurement signal list. ▪ If no symbol is displayed, the variable is not connected to an instrument. The variable is not in the measurement signal list.
Variable	Displays the name of the variable.
Block	Displays the name of the block that contains the variable (real-time model).
Platform/Device	Displays the platform/device related to the variable.
Description	Displays a description of the variable if available.
Unit	Displays the unit of the variable if available.
Type	Displays the format type of the variable.

You can configure the columns to customize the information to be displayed. For details, refer to Customize Columns (Variable List) on page 111.

You can change the alternating row color. Refer to Variables Page on page 146.

Function buttons The following function buttons are available below the header of the Variables controlbar.

Function Button	Description
	Shows or hides the tree view on the left side of the Variables controlbar.
	In the filter list, the Standard Filter is selected. It is specified to display all variable types.
	Lets you enter a text string for searching or filtering the variable list. You can select the column that is used for filtering in the column list to the right.
	Lets you select the column that is used for searching or filtering the variable list by a text string.
	Selects the next occurrence of the text string.

Function Button	Description
	Selects the previous occurrence of the text string.
	Enables a wildcard filter that filters the variable list by the string that is currently entered in the text string field. The wildcard filter is active until the button is pressed again.
	Clears the text string filter settings.

Related Commands The dialog provides the following commands:

Purpose	Refer to
Context menu of the variable list and the tree	
To collapse all the subnodes of a node in the tree.	Collapse All on page 111
To expand all the subnodes of a node in the tree.	Expand All on page 115
To view the properties of a variable (and edit its weak limits).	Properties (refer to Variable Properties / Properties on page 138)
Context menu of the variable list header	
To add columns to or remove them from the variable list.	Customize Columns (Variable List) on page 111

Related topics

References

Trigger Rules (ControlDesk Measurement and Recording 🕮)
Variables 142

Standard Filter

Access You can access this command via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	Whether the following shortcuts are available depends on the context in which you opened the dialog: ▪ **Ctrl+Shift+F** (to open the dialog) ▪ **Ctrl+Shift+P** (to filter by parameters) ▪ **Ctrl+Shift+M** (to filter by measurement variables)
Icon	

Purpose

To filter the variable list by variable types.

Result

The filter is applied to the variable list, which now shows the variables that match the filter conditions.

The following results depend on the context in which the filter is created or displayed:

- The tree view of the variable list is reduced to the relevant nodes, empty nodes are hidden.
- If the filter is active, the color of the variable list entries is blue.
- Above the columns, a header is displayed showing the current filter setting, for example, 'Filter: Parameter, Curve, Map'.

Dialog settings

Note

Which of the following dialog elements and options are available depends on the context in which you opened the dialog.

Variable types Select one or more variable types to filter the variable list by.

Default Types – Parameter Preselects all the parameter types in the Variable types list. You can change the preselection before you click OK to activate the filter.

Default Types – Measurements Preselects all the measurement types in the Variable types list. You can change the preselection before you click OK to activate the filter.

Related topics

HowTos

How to Filter the Variable List....33
How to Use the Function Buttons for Searching and Filtering Variables....37

References

Variable Management Commands....91

Tree View

Access

You can access this command in the **Variables** controlbar via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	`Ctrl+H`
Icon	None

Purpose

To toggle between flat and hierarchy views of the variables.

Result

Opens or closes a the tree view, which displays the variables according to the structure in the related variable description.

Description

In the tree view, variables are displayed according to the structure of the variable description files. The variable list shows the variables of the selected node and its subnodes.

You can change the view via the **Variables** page of the **ControlDesk Options** dialog:

- If you enable Collect variables from subgroups (refer to Variables Page on page 146), the variables of all subgroups are displayed at the selected node.
- If you enable Collect variables from subgroups in root (refer to Variables Page on page 146), all variables of all subgroups are displayed at the root node.

Note

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

Related topics

Basics

Basics of the Variables Controlbar....30

References

Variable Management Commands....91

Uncheck All

Access

You can access this command via:

Ribbon	None
Context menu of	Measurement Data Pool: ▪ Variable list ▪ Hierarchy tree
Shortcut key	None
Icon	None

Purpose

To clear all checkboxes in the variable list.

Description

This command also removes all the variables from the Checked Variables list.

Related topics

References

Variable Management Commands..... 91

Variable Properties / Properties

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Variable list ▪ All instruments (plotters via legend)
Shortcut key	▪ `Alt+Enter` ▪ `Ctrl+I`
Icon	None

Purpose

To view the properties of a variable (and edit its weak limits).

Description

In ControlDesk, the variable properties are read-only, except for the weak limits.

Tip

To edit the properties of a calculated variable in ControlDesk, you must open the Properties dialog via **Calculated Variables – Edit**. Refer to Calculated Variables - Edit on page 105.

Variable Properties dialog

This dialog displays information about a variable. The content of the dialog varies depending on the type of the variable and the entries for a specific variable in the variable description.

The following properties are involved.

General

Name Displays the name of the variable.

Display identifier Displays the DISPLAY_IDENTIFIER of the variable. The display identifier is used for output in graphical user interfaces. If not specified, the name of the variable is used. Whether the display identifier is to be used in the GUI or not can usually be configured in the global options of a tool.

In ControlDesk, you can find this setting on the **Visualization** page of the **ControlDesk Options** dialog.

Format string Displays the Printf declaration for the displaying of a numerical value (if specified).

The following table shows some of the expressions that are available:

Printf String	Description
`%d`	Decimal integer
`%i`	Decimal integer
`%u`	Unsigned integer
`%o`	Octal integer
`%x, %X`	Hexadecimal integer
`%b`	Binary integer
`%f`	Floating point
`%e, %E`	Exponential floating point
`%g, %G`	Floating point of the formats `%f or %e`

For more information on format strings, refer to a standard C programming language reference.

Description Displays the description text for the variable (if specified).

Unit Displays the unit for the converted value (physical value) of the variable, for example, kg or rad/s (if specified).

Address

Address Displays the start address of the variable.

Symbol Displays the name of the symbol that is assigned to the variable. If a symbol is assigned, it specifies the address of the variable.

Offset If an offset is added to the address of the variable, it is displayed here.

Assign Symbol This function is available only in the Variable Editor.

Read-only Displays whether the variable is read-only or writable.

Remove Assignment This function is available only in the Variable Editor.

Data type

Data type Displays the data type (record layout) for the variable.

Byte order Displays whether bytes are stored in Motorola format (big endian, MSB at the memory location with the lowest address) or in Intel format (little endian, LSB at the memory location with the lowest address).

Dimensions (X,Y,Z) (Available only for value blocks and measurement arrays) Displays the number of elements in the array.

- A value block consists of a 1-, or 2-dimensional array of scalar parameters.
- A measurement array consists of a 1-, 2-, or 3-dimensional array of measurement variables.

Bit mask

Bit mask – Start bit Displays the bit offset for the bit mask, counted from the right side of the bit mask.

Bit mask – No. of bits Displays the bit mask width by entering the number of contiguous 1-bits.

Bit mask – Hex Displays the bit mask value in hexadecimal notation.

The following example shows the result of different bit mask properties settings:

Data Type	No. of Bits	Start Bit	Bit Mask	Hex
8-bit	4	0	00001111	F
8-bit	4	2	00111100	3C
16-bit	5	5	0000001111100000	3E0

Computation

Computation method Displays the computation method for converting a source value into a converted value. The conversion rule of the selected computation method is displayed to the right.

In case of a multiscaling, the conversion rules of all the subscalings are displayed.

Limits (weak limits)

Min weak (phys.) (Mandatory. If not specified, a default value is automatically generated.) Lets you specify the lower weak limit of the physical value. You can only enter values that are inside the hard limits.

Hex (Read-only) Displays the source value in hexadecimal notation.

Max weak (phys.) (Mandatory. If not specified a default value is automatically generated.) Lets you specify the upper weak limit of the physical value. You can only enter values that are inside the hard limits.

Hex (Read-only) Displays the source value in hexadecimal notation.

Limits (hard limits)

Min hard (phys.) Displays the lower hard limit of the physical value. You can only enter values that are outside the weak limits.

Hex Displays the source value in hexadecimal notation.

Max hard (phys.) Displays the upper hard limit of the physical value. You can only enter values that are outside the weak limits.

Hex Displays the source value in hexadecimal notation.

Axis (x-axis)

X-axis type Displays the type of the x-axis (standard, fixed, or common axis). Click the Browse button to open the properties dialog of the axis.

X-axis name Displays the name of the common axis (if this is the axis type).

No. of axis points X Displays the number of points on the x-axis.

Input quantity Displays the input quantity.

Axis (y-axis)

Y-axis type Displays the type of the y-axis (standard, fixed, or common axis). Click the Browse button to open the properties dialog of the axis.

Y-axis name Displays the name of the common axis (if this is the axis type).

No. of axis points Y Displays the number of points on the y-axis.

Input quantity Displays the input quantity.

Struct/Struct array

(Available only for structs and struct arrays.)

Struct size in bytes Displays the size of a struct or struct array in bytes.

Dimensions (X, Y, Z) Displays the number of structs in a struct array.

- X and Y for a 2-dimensional struct array
- X, Y, and Z for a 3-dimensional struct array

Related topics

References

Variable Management Commands.....91

Variables

Access

You can access this command via:

Ribbon	View – Controlbar – Switch Controlbars
Context menu of	None
Shortcut key	**Alt+Shift+3**
Icon	

Purpose

To show the Variables controlbar.

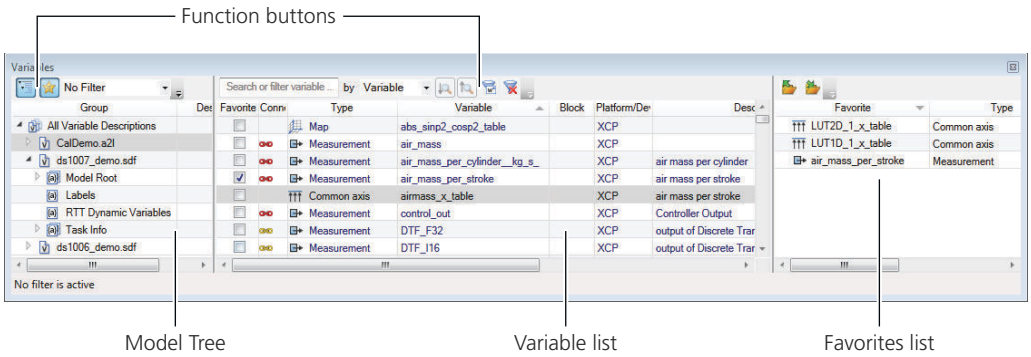
Result

The Variables controlbar opens.

Description

A controlbar that provides access to the variables of the currently open experiment.

The Variables controlbar consists of the following areas:



Tree view In the tree view, the structure of all the variable description files in the active experiment is displayed. ControlDesk lets you toggle between the flat and the tree view of the variables. Refer to Tree View on page 137.

Note

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

Variable list The variable list displays the variables of the variable description. A Subset of the variable information is displayed in predefined columns.

Predefined Columns	Description
Variable Type	Displays the type of the variable.
Connected	Displays a symbol indicating the connection status of the variable. ▪ indicates that the variable is connected to an instrument. ▪ indicates that the variable is not connected to an instrument. The variable is in the measurement signal list. ▪ If no symbol is displayed, the variable is not connected to an instrument. The variable is not in the measurement signal list.
Variable	Displays the name of the variable.
Block	Displays the name of the block that contains the variable (real-time model).
Platform/Device	Displays the platform/device related to the variable.
Description	Displays a description of the variable if available.
Unit	Displays the unit of the variable if available.
Type	Displays the format type of the variable.

You can configure the columns to customize the information to be displayed. For details, refer to Customize Columns (Variable List) on page 111.

You can change the alternating row color. Refer to Variables Page on page 146.

Favorites List Lists the variables that have activated checkboxes in the Favorite column of the variable list. You can export and import variables of a favorites list as text (TXT) files or label (LAB) files.

Function buttons The following function buttons are available below the header of the Variables controlbar.

Function Button	Description
	Shows or hides the tree view on the left side of the Variables controlbar.
	Shows or hides the favorites list on the right side of the Variables controlbar.

Function Button	Description
No Filter	Opens a filter list that lets you select one of the following filters: ▪ Standard Filter ▪ Wildcard Filter ▪ Combined Filter If the tree view is enabled, only the nodes with matching elements are visible. If you select **No filter**, all filters (filter list and text string filter) are disabled.
	Opens the dialog that corresponds to the filter selected in the filter list.
Search or filter variable ...	Lets you enter a text string for searching or filtering the variable list. You can select the column that is used for filtering in the column list to the right.
by Variables	Lets you select the column that is used for searching or filtering the variable list by a text string.
	Selects the next occurrence of the text string.
	Selects the previous occurrence of the text string.
	Enables a wildcard filter that filters the variable list by the string that is currently entered in the text string field. The wildcard filter is active until the button is pressed again. You can create a two-stage filter by first selecting a filter from the filter list and then activating a wildcard filter via the text string field.
	Clears the text string filter settings.
	(Available only if the favorites list is enabled) Opens a dialog to import the variables of a text (TXT) or label list (LAB) file to the favorites list.
	(Available only if the Favorites list is enabled) Opens a dialog to export the variables of the favorites list to a text (TXT) or label list (LAB) file.

Positioning the controlbar By default, the **Variables** controlbar is located at the bottom of the ControlDesk working area.

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 🕮).

Related commands

The Variables controlbar provides the following commands:

Purpose	Refer to
To measure a variable without visualizing it on a layout.	Add to Measurement Signal List on page 96
To enable differentiation between upper and lower case letters for search operations.	Case-Sensitive Search on page 110
To collapse all the subnodes of a node in the tree.	Collapse All on page 111
To copy the selection to the Clipboard.	Copy (ControlDesk User Interface Handling 🕮)
To connect variables to instruments and place them on a layout.	Visualize Variables (ControlDesk Layouting 🕮)
To edit the properties of a calculated variable.	Calculated Variables - Edit on page 105
To expand all the subnodes of a node in the tree.	Expand All on page 115
To export the selected calculated variables to a VXF file.	Calculated Variables - Export Selected Items on page 106
To add columns to or remove them from the variable list.	Customize Columns (Variable List) on page 111
To import a VXF file with calculated variables.	Calculated Variables - Import on page 107
To define a new calculated variable.	Calculated Variables - New on page 108
To disable all filters.	No Filter on page 117
To remove all selected calculated variables.	Calculated Variables - Remove Selected Items on page 109
To remove the selected variable(s) from the measurement signal list.	Remove from Measurement Signal List on page 128
To sort rows alphabetically in ascending order by the selected column.	Sort Ascending (ControlDesk User Interface Handling 🕮)
To sort rows alphabetically in descending order by the selected column.	Sort Descending (ControlDesk User Interface Handling 🕮)
To filter the variable list by variable types.	Standard Filter on page 135
To toggle between flat and hierarchy views of the variables.	Tree View on page 137
To view the properties of a variable (and edit its weak limits).	Variable Properties / Properties on page 138
To create a new combined filter. A combined filter is used to filter the variable list using a combination of filter conditions.	Add/Edit Combined Filter – New / Create Variable Filter on page 102
To filter the variable list using filter strings that contain wildcards.	Wildcard Filter on page 150

Related topics

Basics

Handling Variables in the Variables Controlbar.....30

Variables 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 settings for the Variables controlbar's user interface and the hierarchy view mode.

Dialog settings

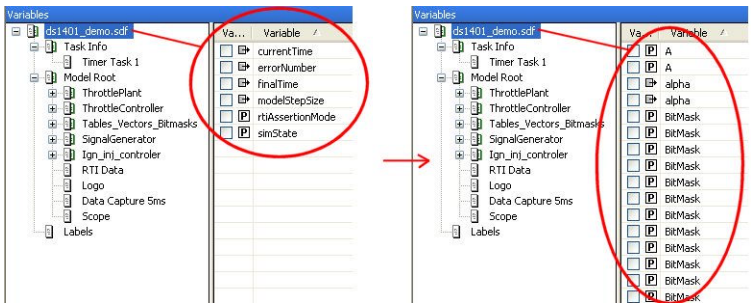
Alternating row color Lets you specify the background color for all even-numbered rows in a variable or item list, for example, in the Variables controlbar.

Ask user to transfer calculated variables to new variable description Lets you specify to ask the user to transfer calculated variables if you reload the variable description. If deactivated, the calculated variables are always transferred.

Collect variables from subgroups in root Lets you specify to display all the variables of a variable description in the variable list when you select the root node in the hierarchy tree.

This option is only for those variable descriptions that have a fixed block structure for the variables, such as SDF files. It does not matter for variable descriptions that have a flat list of variables and only assign groups to them, such as A2L files.

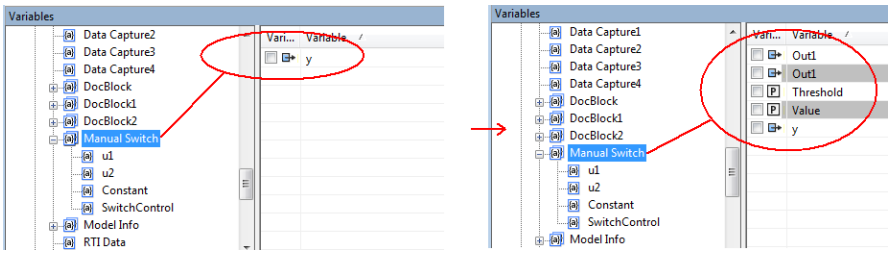
If a variable description has a fixed block structure, selecting a root node usually displays only variables that are not members of a block. If you activate Collect variables from subgroups in root, all the variables are displayed.



Collect variables from subgroups Lets you specify to also display the variables of the subgroups when you select a group node in the hierarchy tree.

- If you activate this option, the variables of the subgroups of the group node selected in the hierarchy tree are also displayed.
- If you deactivate this option, the variables of the subgroups of the group node selected in the hierarchy tree are not displayed.

The following illustration shows a variable description when the option is deactivated (left part of the illustration) and activated (right part of the illustration) as an example.



Rebuild variable description action Lets you specify how ControlDesk reacts when a new simulation application has been built and a new variable description is available.

ControlDesk can detect:

- Real-time applications rebuilt with RTI or ConfigurationDesk
- Offline simulation applications rebuilt with SystemDesk, VEOS Player or TargetLink

ControlDesk can detect a rebuilt simulation application only if it was built using a dSPACE product from dSPACE Release 7.4 or later.

Each rebuild produces a new application image. An application image contains all the files that are created when you build a simulation application. It particularly includes the variable description file and the new simulation application that can be downloaded to the simulation platform.

You have the following options to specify ControlDesk's reaction when it detects a rebuilt variable description:

Option	Description
Do nothing (ignore)	ControlDesk ignores the rebuilt simulation application and does not reload the variable description.
Ask user	ControlDesk asks you how to react when it detects the rebuilt simulation application. You can ignore the rebuild process or reload the variable description.
Reload variable description automatically	ControlDesk automatically reloads the variable description when it detects the rebuilt simulation application. You can specify two additional options: ▪ **Download application image** - Lets you specify to automatically download the simulation application to the simulation platform.

Option	Description
	▪ Force offline mode - Lets you specify to stop online calibration automatically to make it possible to reload the variable description and to download the simulation application to the simulation platform (if enabled).

Import Options

Opens a Properties dialog for you to specify import options for selected variable descriptions:

Purpose	Property Page
To specify import options for variable description files in the A2L format.	A2L Import Options page
To specify import options for variable description files used with bus monitoring devices.	Bus Import Options page
To specify import options for variable description files in the SDF format.	SDF Import Options page

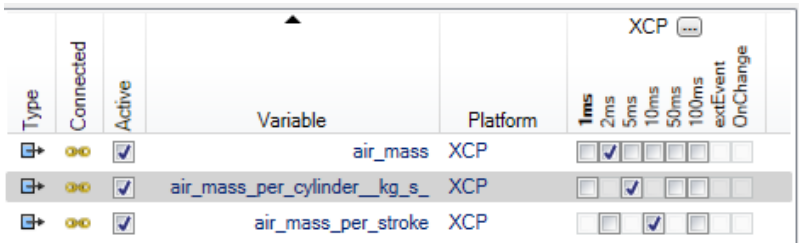
A2L Import Options page

To specify import options for variable description files in the A2L format.

General options – Support IF Data at variables Lets you specify to use variable-specific interface description data (IF_DATA). If enabled, measurement and recording with ControlDesk support variable-specific measurement rasters, which can be defined in the IF_DATA of DCI-GSI 2 and XCP devices of the A2L file.

The available rasters (AVAILABLE_EVENT_LIST) and the default rasters (DEFAULT_EVENT_LIST) of a variable are defined. If more than one default raster is defined, ControlDesk uses the first default raster. The device-specific default raster setting in the Measurement Configuration is overruled.

The following example shows how variables with variable-specific measurement rasters are displayed in the Measurement Configuration, immediately after they have been dropped to it:



- The three variables have different available rasters. Only the available rasters of a variable are selectable.
- The tree variables have different default rasters (2 ms, 5 ms, 10 ms). In the illustration, the default rasters are selected, because the variables have just been dropped to the measurement configuration. The device-specific default raster (1 ms, written in bold letters) was overruled.

In the variable description, different variable-specific raster settings can be defined for individual transport layers. If you change the transport layer in a

project, the new raster settings are used for subsequent measurement configurations. Existing measurement configurations for connected variables have to be adapted manually, if necessary.

General options – Ignore variables at address zero (address 0x0) Lets you specify to ignore variables at the address `0x0` during the import of an A2L file.

In early ECU development stages, ECU variables that are not implemented yet can have the address `0x0` in the A2L file of the related ECU application. If you ignore these ECU variables during the A2L file import, they cannot be mistakenly calibrated.

Bus Import Options page

To specify import options for variable description files for PC-based bus interfaces.

General options – Autorename format Lets you specify how to automatically rename objects to avoid duplicate names using a printf declaration from the C programming language. The default format string is %s(#%d), which copies the name of the original object and adds a number to it which is automatically increased if necessary. Automatic renaming is applied if a non-unique name is encountered.

General options – Enable validation Lets you enable/disable the validation of the objects to be imported. Validation is disabled by default because it increases the import time, particularly if a lot of variables are to be validated.

FIBEX options - Language Lets you specify the language in which multilingual elements, for example, description strings, are to be imported. The abbreviation must correspond to a FIBEX *xml:lang* attribute (i.e. en, de, or ja). If specified language is not used in the FIBEX file or for one of the elements, the language of the first found multilingual element, that does not offer the specified language, determines the import language for all elements of this kind.

If the FIBEX file or any of the elements do not use the specified language, then the import language for these elements is the language of the first multilingual element found without the specified language.

Variable handling – Generate raw byte variables for each CAN message Lets you enable/disable the generation of raw byte variables for each CAN message defined in a variable description file. If enabled, you can use the raw byte variable to send TX messages with a CAN Bus Monitoring device.

SDF Import Options page

To specify import options for variable description files in the SDF format.

Enable validation Lets you enable/disable the validation of the objects to be imported. Validation is disabled by default because it increases the import time, particularly if a lot of variables are to be validated.

Make log file entries for variables with identical address information Lets you specify to log the occurrence of identical address information. Logging this information increases import time.

Trace file parser error output Lets you specify whether to log the short or long (verbose) error information.

Interpret complex as simple variables Lets you switch off the detection of complex variables, e.g., lookup-tables or curve/map variables, and import the simple variables that are specified in the TRC file.

Related topics

References

Options Command (ControlDesk User Interface Handling 🕮)

Wildcard Filter

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Variable list ▪ Hierarchy tree
Shortcut key	`Ctrl+Shift+W`
Icon	

Purpose

To filter the variable list using filter strings that contain wildcards.

Result

The filter is applied to the selected column. The variable list shows the variables that match the filter string.

The following results depend on the context in which the variable list is created or displayed:

- Above the columns, a header is displayed showing the current filter setting, for example, 'Filter: Wildcard (air*)'.
- If the filter is active, the color of the list entries is blue.

Description

You can use the ? wildcard for one missing letter or the * wildcard for an undefined number of missing letters in the filter string.

Dialog settings

Column Lets you select the column to which you want to apply the wildcard filter.

Match case Lets you specify whether the wildcard filter differentiates between lower and upper case letters.

Related topics

HowTos

How to Filter the Variable List....33
How to Use the Function Buttons for Searching and Filtering Variables....37

References

Variable Management Commands....91

Automation

Where to go from here

Information in this section

Programming ControlDesk Automation 154
Variable Description Handling 167
Variable Access Handling 169

Programming ControlDesk Automation

Where to go from here

Information in this section

Automating Platform Management and Variable Management 154
The main tasks of ControlDesk's platform management are to add platforms and/or devices to an experiment, to configure their communication settings, and to add variable descriptions to platforms/devices.

Automating Direct Variable Access 160
You can read and write variables *directly* without having to use instruments.

Information in other sections

Tool Automation Demos (ControlDesk Automation 🕮)
Demonstrate how to automate ControlDesk and use ControlDesk events.

Automating Platform Management and Variable Management

Introduction

The main tasks of ControlDesk's platform management are to add platforms and/or devices to an experiment, to configure their communication settings, and to add variable descriptions to platforms/devices.

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

Registering a platform

The following listing shows you how to register a platform.

Note

To use the listing below in a script to register a platform, you must specify the platform type and further information in the section `Define constants used to register a platform`. The platform does not have to be registered yet.

```
#-----------------------------------------------------------------------------------------------------
# Define constants used to register a platform.
#-----------------------------------------------------------------------------------------------------
# Platform type to register. Uncomment the line with the platform you want to register.
PLATFORMTYPESTRING = None
#PLATFORMTYPESTRING = "DS1006"
#PLATFORMTYPESTRING = "DS1007"
#PLATFORMTYPESTRING = "DS1202"
#PLATFORMTYPESTRING = "MABX"
#PLATFORMTYPESTRING = "SCALEXIO"
#PLATFORMTYPESTRING = "VEOS"
(...)
# Addresses to use to register the platform.
# IP address format: "192.168.140.110"
IPADDRESS = None
PORTADDRESS = 0x300
(...)
class MainDemoController(object):
   def __init__(self):
   (...)
   # The registered platform.
   self.RegisteredPlatform = None
   (...)
   def RegisterPlatform(self):
           if None != PLATFORMTYPESTRING:
              # Show platform window.
              if self.ControlDeskApplication.MainWindow.Windows.Contains(Resources.PlatformTab):
                   self.ControlDeskApplication.MainWindow.Windows.Item(Resources.PlatformTab).Show()
               # Evalute the platform type
               platformType = eval("self.Enums.PlatformType.%s" % PLATFORMTYPESTRING)
               # Create the registration information object.
               registrationInfo = self.ControlDeskApplication.PlatformManagement.\
                                  CreatePlatformRegistrationInfo(PlatformType)
               # Configure the registration information object.
               if self.Enums.PlatformType.MABX == PlatformType and None != IPADDRESS:
                 registrationInfo.NetClient = IPADDRESS
               elif self.Enums.PlatformType.SCALEXIO == PlatformType and None != IPADDRESS:
                 subRegistrationInfo = registrationInfo.RegistrationInfos.Add()
                 subRegistrationInfo.IPADDRESS = IPADDRESS
               else:
                   registrationInfo.PortAddress = PORTADDRESS
                   if None != IPADDRESS:
                       registrationInfo.ConnectionType = self.Enums.InterfaceConnectionType.Net
                       registrationInfo.NetClient = IPADDRESS
                   else:
                       registrationInfo.ConnectionType = self.Enums.InterfaceConnectionType.Bus
               # Register the platform.
               self.RegisteredPlatform = self.ControlDeskApplication.PlatformManagement.\
                                         RegisterPlatform(registrationInfo)
               # Create dictionary with platform information.
               platformInformation = {}
               # Add UniqueName.
               platformInformation['name'] = self.RegisteredPlatform.UniqueName
               # Add ConnectionState.
               platformInformation['displayname'] = self.RegisteredPlatform.DisplayName
```

Adding a registered platform to an experiment

The following listing shows you how to add a registered platform to an experiment. The listing is not part of the `PlatformHandling.py` demo script.

```
class MainDemoController(object):
   def __init__(self):
   (...)
   # The registered platform.
   self.RegisteredPlatform = None
   # The registered platform used in the experiment.
   self.UsedRegisteredPlatform = None
   (...)
   def AddRegisteredPlatform(self):
        if None != self.RegisteredPlatform:
            # Add the existing registered platform to the experiment.
            platforms = self.ControlDeskApplication.ActiveExperiment.Platforms
            if platforms.AddExistingPlatform(self.RegisteredPlatform.UniqueName):
                # Get the platform object from the experiment platforms collection.
                # This object implements the same interface but only on this object
                # properties like VariableDescriptions and Name could be used.
                self.UsedRegisteredPlatform = platforms[1]
```

Adding a platform/device

The following listing shows you how to add an XCP on CAN device and configure some of its properties. The listing is not part of the `PlatformHandling.py` demo script.

```
class MainDemoController(object):
    def __init__(self):
        # The ControlDesk Application object.
        self.ControlDeskApplication = None
        # The enums object.
        self.Enums = None
        (...)
        # The platform used in this demo.
        self.DemoPlatform = None
    def Initialize(self):
        (...)
        # Start ControlDesk.
        self.ControlDeskApplication = Dispatch("ControlDeskNG.Application")
        # Create the enums object.
        self.Enums = Enums(self.ControlDeskApplication)
        (...)
    def CreateAndEditPlatform(self):
        (...)
        # Add XCPonCAN platform.
        self.DemoPlatform = self.ControlDeskApplication.ActiveExperiment.Platforms.Add(
                            self.Enums.PlatformType.XCPonCAN)
        (...)
        # Change the baud rate.
        self.DemoPlatform.CANSettings.BaudRate = 100000
        # Select virtual CAN interface (needed to start the online calibration with the CalDemo device).
        self.DemoPlatform.InterfaceSelection.AvailableChannels.Item(0).Select()
        self.DemoPlatform.GeneralSettings.StartOnlineCalibrationBehavior = self.Enums.\
                                    OnlineCalibrationBehavior.UploadConnectedVariables
```

Editing the processor names in a DS1007-based multiprocessor system

The following listing shows you how to edit the names of the processors in a DS1007-based multiprocessor system. The listing is not part of the `PlatformHandling.py` demo script.

It is presupposed that a DS1007-based multiprocessor system with two DS1007 boards, each of which provides two cores, is registered.

```
class MainDemoController(object):
    (...)
    def EditDS1007ProcessorNames(self):
        dS1007Platform = self.ControlDeskApplication.PlatformManagement.Platforms[0]
        processorNames = dS1007Platform.GetProcessorNames()
        processorNames[0][0].Name = "Master"
        processorNames[0][1].Name = "Slave_B"
        processorNames[1][0].Name = "Slave"
        processorNames[1][1].Name = "Slave_C"
        dS1007Platform.ApplyProcessorNames(processorNames)
```

Adding a variable description

The following listing shows how to add three variable descriptions and ECU image files to an XCP on CAN device and activate the second one.

It also shows how to add a variable description to the registered platform.

```
# Get the root path
SCRIPTPATH = os.path.abspath(os.path.curdir)
if len(sys.argv) > 0:
    SCRIPTPATH = os.path.split(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"))
# Define DS1006 .sdf file path.
DS1006SDFPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\..\..\RTApplications\DS1006\ds1006_demo.sdf"))
# Define DS1401 .sdf file path.
MABXSDFPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\..\..\RTApplications\DS1401\ds1401_demo.sdf"))
# Define SCALEXIO .sdf file path.
SCALEXIOSDFPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\..\..\RTApplications\Scalexio\Scalexio_demo.sdf"))
(...)
class MainDemoController(object):
    def __init__(self):
    (...)
    # The registered platform.
    self.RegisteredPlatform = None
```

```
    # The registered platform used in the experiment.
    self.UsedRegisteredPlatform = None
    (...)
    def AddAndEditVariableDescriptions(self):
        (...)
        # Add three variable descriptions with data sets from CalDemo.
        for index in range(3):
            self.DemoPlatform.VariableDescriptions.AddWithImage(A2LIMAGEPATH, MOTIMAGEPATH)
        (...)
        # Access the second variable description.
        varDes = self.DemoPlatform.VariableDescriptions.Item(1)
        # Activate variable description.
        activeVarDes = varDes.Activate()
        (...)
        if None != self.UsedRegisteredPlatform:
        # Add the matching variable description.
        if self.Enums.PlatformType.DS1006 == self.UsedRegisteredPlatform.Type:
           self.UsedRegisteredPlatform.VariableDescriptions.Add(DS1006SDFPATH)
        elif self.Enums.PlatformType.MABX == self.UsedRegisteredPlatform.Type:
           self.UsedRegisteredPlatform.VariableDescriptions.Add(MABXSDFPATH)
        elif self.Enums.PlatformType.SCALEXIO == self.UsedRegisteredPlatform.Type:
           self.UsedRegisteredPlatform.VariableDescriptions.Add(SCALEXIOSDFPATH)
```

Connecting and online calibration

The following listing shows how to connect to the XCP on CAN device of the CalDemo ECU and then start and stop online calibration.

Note

Do not select `PromptUser` as the `OnlineCalibrationBehavior <<Enumeration>>` property of a platform/device.

If `PromptUser` is selected, a prompt is displayed when differences between parameter values on the hardware and in ControlDesk are detected when online calibration is started. This behavior is not useful in an automation scenario.

```
# Import: The os module is used to start the CalDemo process.
import os
(...)
# Path of the CalDemo exe.
CALDEMOPATH = os.path.abspath(os.path.join(SCRIPTPATH, "..\..\..\CalDemo\CalDemo.exe"))
(...)
class MainDemoController(object):
    (...)
    def ConnectAndStartOnlineCalibration(self):
        # Access the XCPonCAN platform by the item method.
        xcpOnCANPlatform = self.ControlDeskApplication.ActiveExperiment.Platforms.Item(self.DemoPlatform.Name)
        (...)
        # Start CalDemo platform.
        os.startfile(CALDEMOPATH)
        # Connect device.
        XCPOnCANPlatform.Connect()
        (...)
```

```
# Start online calibration.
self.ControlDeskApplication.CalibrationManagement.StartOnlineCalibration()
# Stop online calibration.
self.ControlDeskApplication.CalibrationManagement.StopOnlineCalibration()
# Disconnect platform.
xcpOnCANPlatform.Disconnect()
```

Selecting and focusing variables and variable groups in the Variables controlbar

Selecting and focusing variables The `SelectItems` method of the `VariableBrowserVariableView / IXaVariableBrowserVariableView <<Interface>>` lets you select a list of variables in the Variables controlbar, and focus one of them.

The listing below shows an example:

```
Application.VariablesManagement.VariableBrowser.VariableView.SelectItems('XCP():://control_out', 'XCP():://control_out')
```

For details, refer to VariableBrowserVariableView / IXaVariableBrowserVariableView <<Interface>> (ControlDesk Automation 🕮).

Selecting variable groups The `SelectGroups` method of the `VariableBrowserGroupView / IXaVariableBrowserGroupView <<Interface>>` lets you select a list of variable groups in the Variables controlbar. Via the `SelectRoot` method, you can select the root node.

For details, refer to VariableBrowserGroupView / IXaVariableBrowserGroupView <<Interface>> (ControlDesk Automation 🕮).

Checking whether reloading the variable description is required

The `ActiveVariableDescription / IXaActiveVariableDescription <<Interface>>` provides the `CheckSourceForChanges` method that lets you check whether reloading the active variable description is necessary.

For details, refer to ActiveVariableDescription / IXaActiveVariableDescription <<Interface>> (ControlDesk Automation 🕮).

Related topics

Basics

Basics of Platforms/Devices (ControlDesk Platform Management 🕮)
Handling Variable Descriptions 15
Tool Automation Demos (ControlDesk Automation 🕮)

Automating Direct Variable Access

Introduction

You can read and write variables *directly* without having to use instruments.

Using the *direct variable access* feature, you can:

- Get information on the variables defined in the variable descriptions in a ControlDesk experiment.
- Read and write the values of scalar variables (e.g., scalar parameters and measurement variables) and multidimensional variables (e.g., value blocks, maps, curves and measurement arrays).

  Variable values can be read when online calibration has started. You do not have to start measuring.
- Access a variable via the variable identifier path.
- Analyze complete variable descriptions and collect information from them.
- Add new calculated variables including new formulas, scalings and (simple) data types.

Note

- The following limitations apply to the *direct variable access* feature:
  - The direct variable access feature lets you read/write the *values of specific variables*. However, reading/writing values directly from a memory address without specifying a variable is not supported by the direct variable access feature.
  - You can read and write the values of multidimensional variables such as maps and curves. However, you can access multidimensional variables only as a whole. For example, it is not sufficient to write only the function values of a map. You also have to write all the map's axis values.
  - You can define new data types for a calculated variable. However, you can add only simple data types, not complex ones such as n-dimensional arrays for maps or curves.
- In contrast to other features of ControlDesk's automation interface, the *direct variable access* feature is *not* available via ControlDesk's user interface.

The program listings below consist of excerpts from the `VariableAccessHandling.py` demo script.

Getting information on variables

The following listing shows you how to get the following information on variables of a variable description, and display it in a dialog:

- Unique name of the variable
- Type of the variable (measurement variable, parameter, calculated variable, map,...)
- Information whether the variable is readable
- Information whether the variable is writable

```
# List of variables to show information.
VARIABLESTOSHOW = ['abs_sinp2_cosp2_table', 'air_mass', 'control_deviation', 'control_out', 'f_Kd_1', 'f_Kp_1',\
                   'MeasureArray', 'ParamVector', 'Rec2Sine_x_table', 'Rec2Sine_z_table', 'SignalGenOutput_2']
(...)
class MainDemoController(object):
    def __init__(self):
        # The ControlDesk Application object.
        self.ControlDeskApplication = None
        # The enums for the ControlDesk object model.
        self.Enums = None
       (...)
        # The project root where most parts of the demo are executed..
        self.DemoRootm = None
    def Initialize(self):
        (...)
        # Start ControlDesk.
        self.ControlDeskApplication = Dispatch("ControlDeskNG.Application")
        # Create the enums object.
        self.Enums = Enums(self.ControlDeskApplication)
        (...)
    def ShowVariableInformation(self):
        # Show information in a dialog.
        USERDIALOG.Show("Show information of some variables" , Resources.ShowVariableInformationStep)
        # List with variable information.
        informationLines = []
        # Get the given variables and add information.
        variables = self.ControlDeskApplication.ActiveExperiment.Platforms[0].ActiveVariableDescription.Variables
        for variableName in VARIABLESTOSHOW:
            variable = Variables[variableName]
            informationLines.append("%s : %s, %s, %s" % (variable.Identifier.UniqueName,
                                                         self.Enums.VariableType(variable.Type),
                                                         "Is Readable" if variable.IsReadable else "Not Readable",
                                                         "Is Writable" if variable.IsWritable else "Not Writable"))
        # Create information string.
        information = "\n".join(informationLines)
        # Show information in a Dialog.
        USERDIALOG.Show("Information of some variables" , Resources.ShowVariableInformation % information)
```

Tip

The variable identifier `variable.Identifier.UniqueName` provides information on a variable, such as its unique name or the underlying platform. You can use this information to identify a variable in various dSPACE tools.

An easy way to gather the necessary information on a variable is to paste the connection path into your script when you add the variable. See the following example:

```
variables = self.ControlDeskApplication.ActiveExperiment.
            Platforms[0].ActiveVariableDescription.Variables
variable01 = variables[<connection path>]
```

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.

Reading scalar variables

The following listing shows you how to read the source and converted values of some scalar variables from a platform, and display the values in a dialog:

```
# List of variables to read.
SIMPLEVARIABLESFORREAD = ['air_mass', 'control_out', 'f_Kp_1', 'FrequencyPrescaler', 'Out1D_sin', 'SignalGenOutput',\
                         'throttle_act_pos']
(...)
class MainDemoController(object):
    (...)
    def ReadSimpleVariables(self):
        # Show information in a Dialog.
        variableNames = ", ".join(SIMPLEVARIABLESFORREAD)
        USERDIALOG.Show("Read Simple Variables" , Resources.ReadSimpleVariables % variableNames)
        (...)
        # List with variable value informations.
        informationLines = []
        # Get values for given variables.
        variables = self.ControlDeskApplication.ActiveExperiment.Platforms[0].ActiveVariableDescription.Variables
        for variableName in SIMPLEVARIABLESFORREAD:
            variable = variables.Item(variableName)
            informationLines.append("\n%s : %s | %s" % (variableName, str(variable.ValueConverted),
                                    str(variable.ValueSource)))
        # Create information string.
        information = "\n".join(informationLines)
        # Show information in a Dialog.
        USERDIALOG.Show("Simple Variables Values" , Resources.ShowSimpleVariableValues % information)
```

Writing scalar variables

The following listing shows you how to activate the working data set and write the converted values of some variables to a platform:

```
# List of variables to write.
SIMPLEVARIABLESFORWRITE = [('f_Kp_1', 0.12), ('SignalAmplitude', 55.0), ('SignalForm', 'rectangle')]
(...)
class MainDemoController(object):
    (...)
    def WriteSimpleVariables(self):
        # Show information in a Dialog.
        variableNames = ", ".join([VariableInfo[0] for VariableInfo in SIMPLEVARIABLESFORWRITE])
        USERDIALOG.Show("Write Simple Variables" , Resources.WriteSimpleVariables % variableNames)
        # Activate the working data set.
        self.ControlDeskApplication.ActiveExperiment.Platforms[0].ActiveVariableDescription.DataSets.
                                    WorkingDataSet.Activate()
        # Write converted values for given variables.
        variables = self.ControlDeskApplication.ActiveExperiment.Platforms[0].ActiveVariableDescription.Variables
        for tupel in SIMPLEVARIABLESFORWRITE:
            variable = variables.Item(tupel[0])
            variable.ValueConverted = tupel[1]
```

Reading multidimensional variables

The following listing shows you how to read the source values of some multidimensional variables from a platform, and display the values in a dialog:

```
# List of complex variables to read.
COMPLEXVARIABLESFORREAD = ['abs_sinp2_cosp2_table', 'Rec2Sine_z_table', 'triangle_z_table', 'y_sin_z_table']
(...)
```

```
class MainDemoController(object):
    (...)
    def ReadComplexVariables(self):
        # Show information in a Dialog.
        variableNames = ", ".join(COMPLEXVARIABLESFORREAD)
        USERDIALOG.Show("Read Complex Variables" , Resources.ReadComplexVariables % variableNames)
        # List with variable value informations.
        informationLines = []
        # Get values for given variables.
        variables = self.ControlDeskApplication.ActiveExperiment.Platforms[0].ActiveVariableDescription.Variables
        for variableName in COMPLEXVARIABLESFORREAD:
            variable = variables.Item(variableName)
            informationLines.append("%s :\n" % variableName)
            if variable.Type == self.Enums.VariableType.Curve:
                informationLines.append("Axis : %s\n" % str(variable.Axis.ValueConverted))
            else:
                informationLines.append("X-Axis : %s\n" % str(variable.XAxis.ValueConverted))
                informationLines.append("Y-Axis : %s\n" % str(variable.YAxis.ValueConverted))
                informationLines.append("Function Values : %s" % str(variable.functionValues.ValueConverted))
            # Create information string.
            information = "\n".join(informationLines)
            # Show information in a Dialog.
            USERDIALOG.Show("Complex Variable Values" , Resources.ShowComplexVariableValues % information)
            # Reset the information lines.
            informationLines = []
```

Writing multidimensional variables

The following listing shows you how to read the converted values of some multidimensional variables from a platform, and write the (modified) values back to the platform:

```
# List of complex variables to write.
COMPLEXVARIABLESFORREAD = ['abs_sinp2_cosp2_table', 'Rec2Sine_z_table']
(...)
class MainDemoController(object):
    (...)
    def WriteComplexVariables(self):
        # Show information in a Dialog.
        variableNames = ", ".join(COMPLEXVARIABLESFORREAD)
        USERDIALOG.Show("Write Complex Variables" , Resources.WriteComplexVariables % variableNames)
        # List with variable value informations.
        informationLines = []
        # Read the given variables and increase the function values.
        variables = self.ControlDeskApplication.ActiveExperiment.Platforms[0].ActiveVariableDescription.Variables
        for variableName in COMPLEXVARIABLESFORREAD:
            variable = variables.Item(variableName)
            informationLines.append("%s :\n" % variableName)
            functionValues = variable.functionValues.ValueConverted
            if variable.Type == self.Enums.VariableType.Curve:
                variable.functionValues.ValueConverted = [CurrentValue*0.95 for CurrentValue in functionValues]
            else:
                variable.functionValues.ValueConverted = [[CurrentValue*0.95 for CurrentValue in CurrentRow] \
                                                          for CurrentRow in functionValues]
            informationLines.append("New Function Values : %s" % str(variable.functionValues.ValueConverted))
            # Create information string.
            information = "\n".join(informationLines)
            # Show information in a Dialog.
            USERDIALOG.Show("Complex Variable Values" , Resources.ShowComplexVariableValues % information)
            # Reset the information lines.
            informationLines = []
```

Accessing variables via the ItemByPath method

The following listing shows you how to access a variable via the variable identifier path. The `ItemByPath` method of the `Variables / IXaVariables <<Collection>>` interface is used for this purpose.

```
# Path name of one-dimensional array.
ONEDIMENSIONALARRAYPATH = 'XCP():://ParamVector'
# Path name of multi dimensional array.
MULTIDIMENSIONALARRAYPATH = 'XCP():://MeasureArray'
(...)
class MainDemoController(object):
    (...)
    def ShowItemByPathExample(self):
        #Show Information in a Dialog.
        USERDIALOG.Show("ItemByPath Example", Resources.ItemByPathExampleIntroduction)
        platform = self.ControlDeskApplication.ActiveExperiment.Platforms[0]
        activeVariableDescription = platform.ActiveVariableDescription
        # Access a one-dimensional variable array
        oneDimensionalArray = activeVariableDescription.variables.ItemByPath(ONEDIMENSIONALARRAYPATH)
        itemPath = ONEDIMENSIONALARRAYPATH
        subElements = "\n"
        for subElement in oneDimensionalArray.SubElements:
            subElements += subElement.Identifier.Path + "\n"
        USERDIALOG.Show("One-Dimensional Array", Resources.ShowOneDimensionalArray %(platform.Name, itemPath,
                        subElements))
        # Access a multi-dimensional variable array
        multiDimensionalArray = activeVariableDescription.Variables.ItemByPath(MULTIDIMENSIONALARRAYPATH)
        itemPath = MULTIDIMENSIONALARRAYPATH
        subElements = "\n"
        for subElement in multiDimensionalArray.SubElements:
            subElements += subElement.Identifier.Path + "\n"
        USERDIALOG.Show("Multi Dimensional Array", Resources.ShowMultiDimensionalArray %(platform.Name, itemPath,
                        subElements))
```

Analyzing a variable description

The following listing shows you how to collect information from the variable description of a newly added platform:

```
# Define SCALEXIO .sdf file path.
SCALEXIOSDFPATH = os.path.abspath(os.path.join(ScriptPath, "..\\..\\..\\RTApplications\\Scalexio\\Scalexio_demo.sdf"))
(...)
class MainDemoController(object):
    (...)
    def AnalyzeVariableDescription(self):
        # Stop online calibration.
        self.ControlDeskApplication.CalibrationManagement.StopOnlineCalibration()
        # Add a new platform.
        newPlatform = self.ControlDeskApplication.ActiveExperiment.Platforms.Add(self.Enums.PlatformType.SCALEXIO)
        # Add a variable description.
        activeVariableDescription = newPlatform.VariableDescriptions.Add(SCALEXIOSDFPATH)
        # Analyze the variable description.
        groupCount = 0
        maxMeasurementVariablesInGroupName = ""
        maxMeasurementVariablesInGroup = 0
        maxParameterVariablesInGroupName = ""
        maxParameterVariablesInGroup = 0
        Stack = [activeVariableDescription.RootGroup]
```

```
        while len(Stack) > 0:
            CurrentGroup = Stack.pop()
            groupCount += 1
            # Add all sub variable groups to stack.
            for subGroup in CurrentGroup.Groups:
                Stack.append(subGroup)
            # Iterate through all variables.
            measurementVariablesInGroup = 0
            parameterVariablesInGroup = 0
            for currentVariable in CurrentGroup.Variables:
                if currentVariable.Type == self.Enums.VariableType.Measurement:
                    measurementVariablesInGroup += 1
                elif currentVariable.Type == self.Enums.VariableType.Parameter:
                    parameterVariablesInGroup += 1
            if measurementVariablesInGroup > maxMeasurementVariablesInGroup:
                maxMeasurementVariablesInGroup = measurementVariablesInGroup
                maxMeasurementVariablesInGroupName = CurrentGroup.Name
            if parameterVariablesInGroup > maxParameterVariablesInGroup:
                maxParameterVariablesInGroup = parameterVariablesInGroup
                maxParameterVariablesInGroupName = CurrentGroup.Name
        # Show information in a Dialog.
        USERDIALOG.Show("Variable Description Informations" , Resources.ShowVariableDescriptionInformations % (
                        groupCount, maxMeasurementVariablesInGroupName, maxMeasurementVariablesInGroup, \
                        maxParameterVariablesInGroupName, maxParameterVariablesInGroup))
```

Adding calculated variables

The following listing shows you how to:

1. Define a new formula
2. Create a new formula instance
3. Create a new linear scaling
4. Create a new data type definition
5. Create a new calculated variable based on the newly created elements

```
class MainDemoController(object):
    (...)
    def AddCalculatedVariable(self):
        activeVariableDescription =
           self.ControlDeskApplication.ActiveExperiment.Platforms[1].ActiveVariableDescription
        # Create and set the parameters of the formula.
        newFormula = activeVariableDescription.Formulas.Add("TestFormula")
        newFormula.Formula = "a + b"
        newFormula.FormulaParameters[0].Name = "a"
        newFormula.FormulaParameters.Add("b")
        newFormula.Description = "A new test formula."
        # Create and fill the formula instance.
        newFormulaInstance = newFormula.CreateInstance()
        measurementVariable = activeVariableDescription.Variables.Item(
                              "Model Root/SignalGenerator/SignalGenerator/SignalGenOutput")
        newFormulaInstance.Signals[0].SetValue(
            measurementVariable,
            self.Enums.DependentSignalType.Signal)
        newFormulaInstance.Signals[1].SetValue(3.5, self.Enums.DependentSignalType.Value)
        # Create and set the parameters of a linear scaling.
        newScaling = activeVariableDescription.Scalings.Add("TestScaling", self.Enums.ScalingConversionType.Linear)
        newScaling.Factor = 2.5
        newScaling.Offset = 3.33
```

```
        # Create and set the parameters of a type definition.
        newTypeDefinition = activeVariableDescription.TypeDefinitions.Add("TestTypeDefinition")
        newTypeDefinition.BaseType = self.Enums.DataType.Double
        # Create and set the parameters of a calculated variable.
        newCalculatedVariable = activeVariableDescription.RootGroup.Variables.AddCalculatedVariable(
            "CalculatedVariableTest",
            newScaling,
            newTypeDefinition,
            newFormulaInstance)
```

Related topics

Basics

Getting Started with the Interpreter Controlbar (ControlDesk Automation 🕮)
Tool Automation Demos (ControlDesk Automation 🕮)

References

Variable Access Handling Interfaces..... 169

Variable Description Handling

Variable Description-Related Interfaces

Introduction

A variable description specifies the variables of a platform/device. It also specifies the interface type used to connect the platform/device and how the interface is to be configured.

Description

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

The *ActiveVariableDescription* property of a platform/device returns an object which implements the *IXaActiveVariableDescription* interface. Using this interface, you can get information on the active variable description, and reload/replace the variable description.

The *VariableDescriptions* property of a platform/device returns a collection which implements the *IXaVariableDescriptions* or *IXaImageSupportingVariableDescriptions* interface. Using this interface, you can access all the variable descriptions of a platform/device.

Related interfaces

Interface	Description
IXaActiveVariableDescription (refer to ActiveVariableDescription / IXaActiveVariableDescription <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the active variable description.
IXaVariableDescriptionMetaInformations (refer to VariableDescriptionMetaInformations / IXaVariableDescriptionMetaInformations <<Collection>> (ControlDesk Automation 🕮))	This interface is to access meta-information collection.
IXaImageSupportingVariableDescriptions (refer to ImageSupportingVariableDescriptions / IXaImageSupportingVariableDescriptions <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the image-supporting variable descriptions collection.
IXaVariableDescriptions (refer to VariableDescriptions / IXaVariableDescriptions <<Collection>> (ControlDesk Automation 🕮))	This interface is to access the variable descriptions collection.
IXaVariableDescription (refer to VariableDescription / IXaVariableDescription <<Interface>> (ControlDesk Automation 🕮))	This interface is to access a variable description.

Interface	Description
IXaVariableDescriptionMetaInformations (refer to VariableDescriptionMetaInformations / IXaVariableDescriptionMetaInformations <<Collection>> (ControlDesk Automation 🕮))	This interface is to access meta-information collection.
IXaVariableDescriptionMetaInformation (refer to VariableDescriptionMetaInformation / IXaVariableDescriptionMetaInformation <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing meta information.

Related documentation

Topic	Description
Automating Platform Management and Variable Management on page 154	The main tasks of ControlDesk's platform management are to add platforms and/or devices to an experiment, to configure their communication settings, and to add variable descriptions to platforms/devices.

Variable Access Handling

Where to go from here

Information in this section

Variable Access Handling Interfaces 169
Variable Access handling lets you read and write variable values *directly*, that means, without having to place variables on instruments.
You can also iterate over the variable groups in the variable description and read several properties of a variable.

Global Variable-Related Interfaces 169
You can get all the variables of the active variable description from the global variables collection.

Variable Group-Related Interfaces 171
You can iterate over the variable group tree using the variable group and the child groups.

Variable Access Handling Interfaces

Introduction

Variable Access handling lets you read and write variable values *directly*, that means, without having to place variables on instruments.

You can also iterate over the variable groups in the variable description and read several properties of a variable.

Global Variable-Related Interfaces

Introduction

You can get all the variables of the active variable description from the global variables collection.

Description

Using the *IXaVariables* collection, you can get the variable objects for reading and writing variable values.

Related interfaces

Interface	Description
IXaVariables (refer to Variables / IXaVariables <<Collection>> (ControlDesk Automation 🕮))	Represents the collection of all variables for a variable description.
IXaCalculatedVariable (refer to CalculatedVariable / IXaCalculatedVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the calculated variable.
IXaCommonAxisVariable (refer to CommonAxisVariable / IXaCommonAxisVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the common axis variable.
IXaCurveVariable (refer to CurveVariable / IXaCurveVariable <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the variable type curve.
IXaFixedAxis (refer to FixedAxis / IXaFixedAxis <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the fixed axis.
IXaMapVariable (refer to MapVariable / IXaMapVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access the variable type map.
IXaMatrixVariable (refer to MatrixVariable / IXaMatrixVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to a two-dimensional object, e.g., the function values of a map.
IXaMeasurementArrayVariable (refer to MeasurementArrayVariable / IXaMeasurementArrayVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the MeasurementArray variable type.
IXaMeasurementVariable (refer to MeasurementVariable / IXaMeasurementVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the measurement variable.
IXaParameterVariable (refer to ParameterVariable / IXaParameterVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the variable parameter.
IXaSize2D (refer to Size2D / IXaSize2D <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the dimensions of a 2-D object.
IXaStandardAxis (refer to StandardAxis / IXaStandardAxis <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the standard axis object.
IXaStringVariable (refer to StringVariable / IXaStringVariable <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the string variable.
IXaValueBlockVariable (refer to ValueBlockVariable / IXaValueBlockVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the variable value block.
IXaVariableIdentifier (refer to VariableIdentifier / IXaVariableIdentifier <<Interface>> (ControlDesk Automation 🕮))	This interface is to access a variable identifier.

Interface	Description
IXaVectorVariable (refer to VectorVariable / IXaVectorVariable <<Interface>> (ControlDesk Automation 🕮))	This interface represents a one-dimensional object, e.g,. the function values of a curve.
VariableType (refer to VariableType <<Enumeration>> (ControlDesk Automation 🕮))	All supported types of variables.

Related documentation

Topic	Description
Automating Direct Variable Access on page 160	You can read and write variables *directly* without having to use instruments.

Variable Group-Related Interfaces

Introduction

You can iterate over the variable group tree using the variable group and the child groups.

Description

From the root group of the active variable description, which implements the *IXaVariableGroup* interface, you can iterate over the variable group tree of the active variable description. Each group object has a *Groups* property, which returns an object that collects all child groups and implements the *IXaVariableGroups* interface.

Related interfaces

Interface	Description
IXaActiveVariableDescription (refer to ActiveVariableDescription / IXaActiveVariableDescription <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the active variable description.
IXaVariableGroup (refer to VariableGroup / IXaVariableGroup <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the variable group.
IXaVariableGroups (refer to VariableGroups / IXaVariableGroups <<Collection>> (ControlDesk Automation 🕮))	This interface is to access the variable groups collection.
IXaGroupVariables (refer to GroupVariables / IXaGroupVariables <<Collection>> (ControlDesk Automation 🕮))	Represents the collection of variables in the variable group.

Interface	Description
IXaCalculatedVariable (refer to CalculatedVariable / IXaCalculatedVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the calculated variable.
IXaCommonAxisVariable (refer to CommonAxisVariable / IXaCommonAxisVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the common axis variable.
IXaCurveVariable (refer to CurveVariable / IXaCurveVariable <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the variable type curve.
IXaFixedAxis (refer to FixedAxis / IXaFixedAxis <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the fixed axis.
IXaMapVariable (refer to MapVariable / IXaMapVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access the variable type map.
IXaMatrixVariable (refer to MatrixVariable / IXaMatrixVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to a two-dimensional object, e.g., the function values of a map.
IXaMeasurementArrayVariable (refer to MeasurementArrayVariable / IXaMeasurementArrayVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the MeasurementArray variable type.
IXaMeasurementVariable (refer to MeasurementVariable / IXaMeasurementVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the measurement variable.
IXaParameterVariable (refer to ParameterVariable / IXaParameterVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the variable parameter.
IXaSize2D (refer to Size2D / IXaSize2D <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the dimensions of a 2-D object.
IXaStandardAxis (refer to StandardAxis / IXaStandardAxis <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the standard axis object.
IXaStringVariable (refer to StringVariable / IXaStringVariable <<Interface>> (ControlDesk Automation 🕮))	This interface is to access the string variable.
IXaValueBlockVariable (refer to ValueBlockVariable / IXaValueBlockVariable <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to the variable value block.
IXaVariableIdentifier (refer to VariableIdentifier / IXaVariableIdentifier <<Interface>> (ControlDesk Automation 🕮))	This interface is to access a variable identifier.
IXaVectorVariable (refer to VectorVariable / IXaVectorVariable <<Interface>> (ControlDesk Automation 🕮))	This interface represents a one-dimensional object, e.g,. the function values of a curve.
VariableType (refer to VariableType <<Enumeration>> (ControlDesk Automation 🕮))	All supported types of variables.

Related documentation

Topic	Description
Automating Direct Variable Access on page 160	You can read and write variables *directly* without having to use instruments.

Troubleshooting

Where to go from here

Information in this section

Variables Removed from the Measurement Signal List When Activating Another Variable Description 175
Variables might be removed from the measurement signal list when another variable description is activated and when the original variable description is reactivated later on.

Problems with A2L Files 176
This topic lists possible warning and error messages concerning invalid A2L syntax statements which may occur during the import process of A2L files in ControlDesk or RTI Bypass Blockset. Each message includes information on why the error occurred, and a solution to the problem.

Problems with AML Files 178
This topic lists possible warning and error messages concerning inconsistencies between interface-specific settings in the A2L file and the AML description.

Variables Removed from the Measurement Signal List When Activating Another Variable Description

Problem

Variables might be removed from the measurement signal list when another variable description is activated and when the original variable description is reactivated later on.

Description

The symbol in the **Measurement Configuration** and in the **Variables** controlbar shows if a variable is not visualized in an instrument but is in the measurement signal list. If you activate another variable description *that*

does not contain this variable, and then reactivate the original variable description later on, the variable is no longer in the measurement signal list.

Solution

To avoid this, visualize the variable in a plotter before you activate another variable description.

When you reactivate the original variable description, ControlDesk restores the connection between the variable and the plotter, and adds the variable to the measurement signal list again.

Related topics

Basics

Basics on Configuring Measurement (ControlDesk Measurement and Recording 🕮)

Problems with A2L Files

Introduction

This topic lists possible warning and error messages concerning invalid A2L syntax statements which may occur during the import process of A2L files in ControlDesk or RTI Bypass Blockset. Each message includes information on why the error occurred, and a solution to the problem.

Improper parameter of the ASAP2_VERSION keyword

Error message VariablesManagement: ERROR 1,0: File <file_name>, line <line_number>, token '1.21'. Open the A2L file and check the syntax at line <line_number>.

Description The parameter of the `ASAP2_VERSION` keyword is of invalid data type.

Example of invalid syntax:

```
ASAP2_VERSION "1.61"
```

The parameter of the `ASAP2_VERSION` keyword is of String data type, which is invalid.

Solution Specify `"Integer"` as the data type for the parameter of the `ASAP2_VERSION` keyword.

Correct syntax:

```
ASAP2_VERSION 1 61 /* Version 1.61 */
```

For details on the `ASAP2_VERSION` keyword, refer to the specification of the ASAM MCD-2 MC standard. See https://www.asam.net.

Improper declaration of the FUNCTION keyword

Error message VariablesManagement: ERROR 1,0: File <file_name>, line <line_number>, token 'FUNCTION'. Open the A2L file and check the syntax at line <line_number>.

Description The declaration of the `FUNCTION` keyword is invalid.

Example of invalid syntax:

```
FUNCTION {
GLOBAL "group generated automatically by a2lgen."
}
```

The short delimiters "`{`" and "`}`" are used, but they are not supported.

Solution Use the delimiters "`/begin`" and "`/end`" for the declaration of the `FUNCTION` keyword.

Correct syntax:

The correct syntax for the above example is:

```
/begin FUNCTION GLOBAL "group generated automatically by
a2lgen."
/end FUNCTION
```

For details on the `FUNCTION` keyword, refer to the specification of the ASAM MCD-2 MC standard.

Identifiers beginning with a Dot character

Warning message VariablesManagement: WARNING 1,0: Warning at <file_name> (<line_number>): Invalid identifier 0.MyExample. You may check the syntax at line <line_number>. Continuing import.

Description The first parameter of the `CHARACTERISTIC` keyword might not be declared correctly.

Example of invalid syntax:

```
/begin CHARACTERISTIC .MyExample
```

The first parameter of the `CHARACTERISTIC` keyword begins with the "." dot character, which is invalid.

Solution Specify "Ident" as the data type for first parameter of the `CHARACTERISTIC` keyword.

As a result, the dot character is deleted or replaced by a valid character.

Note

An identifier beginning with a dot character is recognized by the parser as a number, and is automatically extended by the prefix "0". So an identifier of "Ident" data type is renamed in the data base (see first example below).

Correct syntax:

These are correct statements for the above examples:

```
/begin CHARACTERISTIC 0.MyExample
```

or

```
/begin CHARACTERISTIC MyExample
```

For details on the CHARACTERISTIC keyword, refer to the specification of the ASAM MCD-2 MC standard.

String ending with a single backslash character

Error message VariablesManagement: WARNING 1,0: Warning at <file_name> (<line_number>): string is too long (max 255). You may check the syntax at line <line_number>. Continuing import.

VariablesManagement: ERROR 1,0: File <file_name>, line <line_number>, token <token>. Open the A2L file and check the syntax at line <line_number>.

Description A string in the A2L file is not declared correctly.

Example of invalid syntax:

```
/begin CHARACTERISTIC dummy "backslash at the end \"
```

The second parameter of the CHARACTERISTIC keyword is of "String" data type and ends with a single backslash character, which is invalid.

Solution According to the A2L file specification, the parameter must be a long identifier, and must therefore be an ANSI C compliant C type string.

The backslash character can be used within a string only in combination with either a double quotation mark (\"), or another backslash (\\), as displayed in the following table:

String Declaration	Result After Parsing
"hello \"world\"."	hello "world".
"hello world\\"	hello world\

Correct syntax:

```
/begin CHARACTERISTIC dummy "backslash at the end \\"
```

For details on the CHARACTERISTIC keyword, refer to the specification of the ASAM MCD-2 MC standard.

You can find it at https://www.asam.net.

Related topics

Basics

Variable Descriptions Supported by ControlDesk 10

References

Limitations for A2L Files 189

Problems with AML Files

Introduction

This topic lists possible warning and error messages concerning inconsistencies between interface-specific settings in the A2L file and the AML description.

Missing requested IF_DATA entry

Error message VariablesManagement: The Variable Description '<path><file_name>.a2l' does not contain the IF_DATA block '<interface_name>'.

Description The `IF_DATA` block for the chosen platform/device is not included in the A2L file.

Solution The appropriate `IF_DATA` entry which holds interface-specific information must be added to or included in the A2L file. The format of the interface-specific information is specified in an appropriate AML file, which must also be included in the A2L file.

Missing entire AML entry

Error message VariablesManagement: ERROR 1,0: File '<file_name>', line <line_number>, token 'IF_DATA'. Open the A2L file and check the syntax at line <line_number>.

Description The A2L file contains an `IF_DATA` entry, but the corresponding AML description is missing.

Solution Insert the correct AML description, or include the correct AML file.

Included AML file not found

Error message VariablesManagement: ERROR 1,0: File '<file_name>', line <line_number>, token '<included_file_name>' : File '<included_file_name>' not found. Open the A2L file and check the syntax at line <line_number>.

Description The included AML file is not found due to one of the following reasons:

- The AML file name is wrong.
- The AML file is neither in the same folder as the A2L file that includes it, nor in one of the following folders:
  - `.\Main\bin\a2l\aml` folder of your ControlDesk installation
  - `<RCP_HIL_InstallationPath>\MATLAB\RTIBYPASS\CommonModules\a2l\aml` folder of your RCP and HIL software installation.

Example of incorrect AML file inclusion:

```
/include "test_file.aml"
```

Solution

- Change the AML file name.
- Copy the AML file to the correct folder or to the `.\dsdd\a2l\aml` folder of your ControlDesk installation.

Missing AML definition

Error message VariablesManagement: ERROR 1,0: File '<file_name>', line <line_number> : error: syntax. Open the A2L file and check the syntax at line <line_number>.

Description The AML description is incomplete.

Example of inconsistent A2L and AML file entries:

IF_DATA Entry in the A2L File	AML Description
… K_LINE WuP 0x10 0xF1 …	

The AML description for the K_LINE structure is missing.

Solution The code lines containing the K-LINE structure declaration must be commented out in, or deleted from, the A2L file.

Numerical value description in the AML file, but flag declaration in the IF_DATA entry

Error message VariablesManagement: ERROR 1,0: File '<file_name>', line <line_number>: Number expected. Open the A2L file and check the syntax at line <line_number>.

Description The AML description of the interface parameter and the `IF_DATA` entry in the A2L file are inconsistent.

Example of inconsistent A2L and AML file entries:

IF_DATA Entry in the A2L File	AML Description
… /begin TP_BLOB 0xF1 WuP MSB_FIRST 1 **DATA_ACCESS** …	… block "TP_BLOB" struct { uint; enum {"WuP"=1,"5Baud"=2 }; enum {"MSB_FIRST"=1,"MSB_LAST"=2 }; uint; **ulong;** …

The last displayed parameter is declared as an Enumeration value in the `IF_DATA` entry, which is invalid.

Solution The A2L file must be modified. Declare the parameter as an Unsigned Integer value.

Below are consistent A2L file and AML file entries for the above example:

IF_DATA Entry in the A2L File	AML Description
… /begin TP_BLOB 0xF1 WuP MSB_FIRST 1 **1** …	… block "TP_BLOB" struct { uint; enum {"WuP"=1,"5Baud"=2 }; enum {"MSB_FIRST"=1,"MSB_LAST"=2 }; uint; **ulong;** …

Numerical value description in the AML file, but enumeration value declaration in the IF_DATA entry

Error message VariablesManagement: ERROR 1,0: File '<file_name>', line <line_number>: Number expected. Open the A2L file and check the syntax at line <line_number>.

Description The AML description of the interface parameter and the `IF_DATA` entry in the A2L file are inconsistent.

Example of inconsistent A2L and AML file entries:

IF_DATA Entry in the A2L File	AML Description

…
`/begin TP_BLOB`
`0x210`
`0x1234`
`MSB_FIRST`
…
	…
	`block "TP_BLOB" struct`
	`{`
	`ulong;`
	`uint;`
	`uint;`
	…

The last displayed parameter does not match. The AML description of the parameter expects a numerical value, but the parameter is declared as an enumeration value in the `IF_DATA` entry in the A2L file.

Solution The AML description and the declaration in the `IF_DATA` entry in the A2L file must be made consistent.

There are two ways to solve the problem, depending on your requirements:

- A2L file modification

  The `IF_DATA` entry in the A2L file must be adapted to the AML description. Declare the parameter as an unsigned integer value.

  - Example

    This is a correct A2L file entry for the above example:

IF_DATA Entry in the A2L File	AML Description

…
`/begin TP_BLOB`
`0x210`
`0x1234`
`1`
…
	…
	`block "TP_BLOB" struct`
	`{`
	`ulong;`
	`uint;`
	`uint;`
	…

- AML file modification

  The declaration of the interface parameter in the AML description must be modified. Declare the parameter as an enumeration value.

  - Example

    This is a correct AML file entry for the above example:

IF_DATA Entry in the A2L File	AML Description

…
`/begin TP_BLOB`
`0x210`
`0x1234`
	…
	`block "TP_BLOB" struct`
	`{`
	`ulong;`

IF_DATA Entry in the A2L File	AML Description
MSB_FIRST …	uint; enum {"MSB_FIRST"=1,"MSB_LAST"=2 }; …

Missing parameters in the IF_DATA entry

Error message VariablesManagement: ERROR 1,0: File '<file_name>', line <line_number>: String expected. Open the A2L file and check the syntax at line <line_number>.

Description The AML description of the interface parameters and the `IF_DATA` entry in the A2L file are inconsistent.

Example of inconsistent A2L and AML file entries:

IF_DATA Entry in the A2L File	AML Description

…
/begin SEED_KEY
" "
" "
/end SEED_KEY
…
	…
	block "SEED_KEY" struct
	{
	char[256];
	char[256];
	char[256];
	}
	…

The number of parameters contained in the `SEED_KEY` block differs in the AML description and in the A2L file.

Solution The A2L file must be modified. You must declare one more parameter of String type in the `IF_DATA` entry.

Below are consistent A2L file and AML file entries for the above example:

IF_DATA Entry in the A2L File	AML Description

…
/begin SEED_KEY
" "
" "
" "
/end SEED_KEY
…
	…
	block "SEED_KEY" struct
	{
	char[256];
	char[256];
	char[256];
	}
	…

XCP/XCPplus AML definition differs from ASAM specification

Some A2L files contain AML definitions for XCP or XCPplus that group several `taggedstruct` definitions into one. These AML definitions are not consistent with the AML definition in the XCP protocol specification defined by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.).

Thus, when you import such an A2L file, ControlDesk misinterprets the `IF_DATA` information.

Related topics

Basics

Problems with A2L Files..... 176

Limitations

Where to go from here

Information in this section

General Limitations for Variable Management 185
There are some general limitations for variable management in ControlDesk.

Limitations for SDF Files 186
There are some limitations when working with SDF files in ControlDesk.

Limitations for A2L Files 189
There are some limitations when working with A2L files.

Limitation for DBC Files 194
There is a limitation when working with DBC files.

Limitations for Calculated Variables 195
There are some limitations when working with calculated variables.

General Limitations for Variable Management

Introduction

There are some general limitations for variable management in ControlDesk.

No reloading/replacing of variable descriptions used in several experiments

You cannot reload or replace a variable description if the variable description is added to platforms/devices in multiple experiments of a project.

As a workaround, add a new variable description to a platform/device in the affected experiment of a project.

Limitation for write-only variables

The following limitation applies in connection with *write-only variables*. A write-only variable is a variable that can be written to the platform hardware but not read back.

ControlDesk's **Variables** controlbar does not let you distinguish write-only variables from read/write variables.

Limitations for SDF Files

Introduction

ControlDesk has the following limitations in connection with system description (SDF) files for platforms and the TRC files referenced by SDF files. The following aspects are particularly important if hand-written TRC files are used.

Modifications on TRC file generation in connection with MATLAB R2015b

You have to note some modifications on TRC file generation in connection with MATLAB R2015b.

For more information on TRC file generation and the latest migration instructions, refer to the dSPACE website: http://www.dspace.com/go/trc.

Properties of struct elements are not imported

Properties such as the range, unit and description of elements of structs in an SDF/TRC file are not imported in ControlDesk. These properties are not available in the **Variables** controlbar.

No support for look-up tables in structures

ControlDesk does not support look-up tables in structures.

TRC file must comply with the TRC file syntax

ControlDesk supports SDF files only if the referenced TRC file complies with the syntax of TRC files.

Example If you add an SDF file containing the variable descriptions below to a platform, all the variables will be represented as measurement variables in ControlDesk although the variables `d`, `c` and `m` are parameters. ControlDesk cannot distinguish whether the variables are parameters or measurement variables, since the variable descriptions below do not comply with the syntax of TRC files.

```
group "Model"
   x_disp          flt
   f               flt
   x               flt
   v               flt
   a               flt
   group "Model Parameters"
```

```
      d                   flt
      c                   flt
      m                   flt
   endgroup
endgroup
```

TRC file syntax details For details on the syntax of TRC files, refer to Syntax of the TRC File (RTI and RTI-MP Implementation Reference 🕮).

One variable occurring more than once

You can include several descriptions of the same variable in the SDF file, for example, by describing a parameter as a vector and as a list of vector elements.

Example The following variable description contains the variable dblcucdParamVector: The variable is described as a vector. The individual vector elements are also described:

```
dblcucdParamVector[0..3]
   {
      type: flt(64,IEEE)
      alias: "ParamVector"
      desc: ""
      flags: PARAM
   }
dblcucdParamVector[0]
   {
      type: flt(64,IEEE)
      alias: "Period"
      value: 1.0
      desc: "Parameter to adjust the period of sinus signal"

      flags: PARAM
   }
dblcucdParamVector[1]
   {
      type: flt(64,IEEE)
      alias: "Phaseshifting"
      value: 1.0
      desc: "Parameter to adjust the phase of sinus signal"
      flags: PARAM
   }
dblcucdParamVector[2]
   {
      type: flt(64,IEEE)
      alias: "Offset"
      value: 0.0
      desc: "Parameter to adjust the offset of sinus signal"

      flags: PARAM
   }
```

In ControlDesk's Variables controlbar, the variable `dblcucdParamVector[0]` is represented like this:

```
dblcucdParamVector[0] (ParamVector[0]) …
    …
    …
dblcucdParamVector[0] (Period) …
```

Although the variable `dblcucdParamVector[0]` exists only once, ControlDesk treats it like two different variables, since the variable representations have different names.

If you visualize each variable representation in an instrument individually, ControlDesk does not synchronize the values displayed in the instruments. The values displayed may therefore be inconsistent.

tableData and LookUpTableData TRC variable for Simulink Lookup Table (n-D) blocks The Variables controlbar displays the `tableData` TRC variable created for a Simulink Lookup Table (n-D) block, which has the same address and offset as the `LookUpTableData` TRC variable.

Note

Before you export a data set or create an application image, you should upload all the parameters to ensure consistency for the values of parameters occurring more than once in the same variable description.

Writing bit-masked variables only byte-wise

You can write the value of bit-masked variables only byte-wise, which has the following consequences:

- When you write the value of a bit-masked variable, the values of the masked bits are also written.
- If several bit-masked variables have the same address, writing to one of the variables overwrites the other variables.

Related topics

Basics

General Limitations for Variable Management 185
Problem for Time-Stamped Measurement in Handcoded Applications (ControlDesk Measurement and Recording 🕮)
Problem with Unavailable Measurement Rasters in Handcoded Applications (ControlDesk Measurement and Recording 🕮)

Limitations for A2L Files

Introduction

There are some limitations when working with A2L files.

Common limitations

The following limitations are common to ControlDesk, Variable Editor, and RTI Bypass Blockset.

Variable names containing special characters This limitation applies to special characters for variable names in A2L files:

- Exporting and importing variables whose names contain space characters is not supported.
- Japanese characters are not supported.

Limitation for conversion methods (COMPUx_TAB) The following limitation applies to variables with conversion methods (`COMPU_TAB_INTP`, `COMPU_TAB_NOINTP`, `COMPU_VTAB`, `COMPU_VTAB_RANGE`).

If the source value to be converted is outside the range [-2,147,483,648 ... 2,147,483,647] (`MAX(Long)`), the conversion will be incorrect since ControlDesk and the Variable Editor can work only with `Long` values.

Limitation for the MODULE keyword A2L files with more than one `MODULE` keyword are not supported. Importing such an A2L file results in unpredictable behavior. It is recommended to use only one A2L file for each ECU.

Unsupported A2L keywords The A2L keywords listed below are not supported or supported with limitations by this version of ControlDesk, the Variable Editor and RTI Bypass Blockset. The list also provides information on why the keywords are unsupported and how they are handled. If a keyword contains information that is necessary for the entire block, the entire block will be deleted.

A2L Keyword	Description

`ALIGNMENT_BYTE`
`ALIGNMENT_FLOAT32_IEEE`
`ALIGNMENT_FLOAT64_IEEE`
`ALIGNMENT_LONG`
`ALIGNMENT_WORD`
	If one or more of these keywords are defined for a `RECORD_LAYOUT` block, and if the parameter values of the keywords do not match the default values or the values defined under `MOD_COMMON`, `CHARACTERISTIC` blocks of `CURVE` and `MAP` type and `AXIS_PTS` blocks with such a `RECORD_LAYOUT` referenced are not saved.

`ALTERNATE_CURVES`	ControlDesk does not support `CURVE`s deposited as `ALTERNATE_CURVES`. `RECORD_LAYOUT` blocks with this parameter defined for `FNC_VALUES` are not saved. `CHARACTERISTIC` blocks with such a `RECORD_LAYOUT` referenced as a deposit are not saved.

`AXIS_PTS_Z`
`NO_AXIS_PTS_Z`
`SRC_ADDR_Z`
	If one of these keywords is defined for a `RECORD_LAYOUT` block, the block is not saved. The keywords refer to the z-axis that is defined only for `CUBOID`s.

`AXIS_RESCALE_X`
`AXIS_RESCALE_Y`
`AXIS_RESCALE_Z`
`DIST_OP_X`
	If one of these keywords is defined for a `RECORD_LAYOUT` block, the block is not saved. `CHARACTERISTIC` blocks with such a `RECORD_LAYOUT` referenced as a deposit are not saved.

A2L Keyword	Description
DIST_OP_Y DIST_OP_Z IDENTIFICATION NO_RESCALE_X NO_RESCALE_Y NO_RESCALE_Z OFFSET_X OFFSET_Y OFFSET_Z RIP_ADDR_X RIP_ADDR_Y RIP_ADDR_Z RIP_ADDR_W SHIFT_OP_X SHIFT_OP_Y SHIFT_OP_Z	
BIT_OPERATION	Information provided by this keyword is not saved.
COMPARISON_QUANTITY DEPENDENT_CHARACTERISTIC VIRTUAL_CHARACTERISTIC	If one or more of these keywords are defined for a CHARACTERISTIC block, the block is saved, but without the information provided by the keywords.
CUBOID CUBE_4 CUBE_5	CHARACTERISTICs of the CUBOID, CUBE_4 or CUBE_5 types are not supported by ControlDesk. They are not saved.
CURVE_AXIS CURVE_AXIS_REF	These keywords were introduced with version 1.50 of the ASAM MCD-2 MC standard. Reference curves as normalization axes for maps are not supported. MAPs and CURVEs using this axis type cannot be visualized in ControlDesk.
DATA_SIZE ECU_CALIBRATION_OFFSET S_REC_LAYOUT SYSTEM_CONSTANT	Information provided by these keywords is not saved.
DIFFERENCE	Axis points with DIFFERENCE deposit mode are not supported. If the standard deposit mode for axis points is defined as DIFFERENCE in MOD_COMMON, all the AXIS_PTS and AXIS_DESCR blocks without a defined deposit are regarded as DIFFERENCE. AXIS_PTS blocks with DIFFERENCE deposit and all the CURVEs and MAPs that reference such an axis are not saved. CURVEs and MAPs that have AXIS_DESCR blocks with DIFFERENCE deposit are not saved.
FRAME UNIT VARIANT_CODING	These blocks are not saved.
FUNCTION_LIST	Grouping adjustable objects with the keyword FUNCTION_LIST is not supported. Use the FUNCTION keyword instead. Since version 1.20 of the ASAM MCD-2 MC standard, the FUNCTION keyword provides features to describe functional structures and dependencies.

A2L Keyword	Description
FORMULA	The operators hex(x) and phys(x), which were introduced with version 1.7 of the ASAM MCD-2 MC standard, are not supported. If a formula contains one of these operators, the formula string is not processed further. A warning is displayed.
IF_DATA	IF_DATA blocks defined under AXIS_PTS, CHARACTERISTICs and MEASUREMENTs are not entirely supported. For IF_DATA ASAP1B_CCP, the base address defined under DP_BLOB or KP_BLOB is saved.
INSTANCE	This keyword was introduced with version 1.7 of the ASAM MCD-2 MC standard. Only the following type definitions can be referenced: ▪ TYPEDEF_AXIS ▪ TYPEDEF_CHARACTERISTIC of the VALUE or ASCII type ▪ TYPEDEF_MEASUREMENT with a maximum of 3 dimensions ▪ TYPEDEF_STRUCTURE with a maximum of 2 dimensions If other type definitions are referenced, the block is not saved.
MAP_LIST	If this keyword is defined for a CHARACTERISTIC block, the block is not saved.
MATRIX_DIM	▪ ASAM MCD-2 MC version 1.6 and earlier: This keyword is supported only for CHARACTERISTIC blocks. If it is defined for other blocks, the blocks are not saved. If this keyword is defined for a CHARACTERISTIC block, the block is saved only if it describes a 1-dimensional or 2-dimensional array: the last defined parameter (zDim) must be 1. If both keywords MATRIX_DIM and NUMBER are defined for a CHARACTERISTIC but the values are not compatible, the MATRIX_DIM values are saved (a warning is displayed). ▪ ASAM MCD-2 MC version 1.7: In ControlDesk, this keyword can be used only for the following multi-dimensional object types: ▪ CHARACTERISTIC or TYPEDEF_CHARACTERISTIC of the VAL_BLK type (with a maximum of 2 dimensions) Do not set more than 2 dimensions. In contrast to version 1.6, you must not set the parameter of the third dimension. ▪ MEASUREMENT or TYPEDEF_MEASUREMENT (with a maximum of 3 dimensions) ▪ TYPEDEF_STRUCTURE (with a maximum of 3 dimensions)
REF_MEMORY_SEGMENT	If the keyword is defined for a CHARACTERISTIC, MEASUREMENT or AXIS_PTS block, the block is saved, but without the information provided by the keyword.
REF_UNIT	If the keyword is defined for a COMPU_METHOD block, the block is saved, but without the information provided by the keyword.
RES_AXIS	AXIS_DESCR blocks with this attribute are not supported. CURVEs and MAPs using such AXIS_DESCR blocks are not saved.
STRUCTURE_COMPONENT	This keyword was introduced with version 1.7 of the ASAM MCD-2 MC standard. Only the following type definitions can be referenced: ▪ TYPEDEF_CHARACTERISTIC of the following types:

A2L Keyword	Description
	▪ VALUE ▪ VAL_BLK with a maximum of 2 dimensions ▪ ASCII ▪ TYPEDEF_MEASUREMENT with a maximum of 3 dimensions ▪ TYPEDEF_STRUCTURE with a maximum of 2 dimensions If other type definitions are referenced, the block is not saved.
TYPEDEF_AXIS	This keyword was introduced with version 1.7 of the ASAM MCD-2 MC standard. Instantiated axis objects can be referenced in an axis description (AXIS_DESCR) for a common axis (COM_AXIS). Due to the limitations of STRUCTURE_COMPONENT, TYPEDEF_AXIS cannot be referenced in structures.
TYPEDEF_CHARACTERISTIC	This keyword was introduced with version 1.7 of the ASAM MCD-2 MC standard. Only the following CHARCTERISTIC types are supported: ▪ ASCII ▪ VALUE ▪ VAL_BLK with a maximum of 2 dimensions If the MATRIX_DIM keyword is defined for a VAL_BLK block and specifies more than 2 dimensions, the block is not saved. If it is defined for an ASCII or VALUE block, the block is not saved. If the NUMBER keyword is defined for a VALUE or a VALUE_BLK block, the block is not saved.
TYPEDEF_MEASUREMENT	This keyword was introduced with version 1.7 of the ASAM MCD-2 MC standard. Only the following object types are supported: ▪ MEASUREMENT with a maximum of 3 dimensions If more than 3 dimensions are defined, the block is not saved.
TYPEDEF_STRUCTURE	This keyword was introduced with version 1.7 of the ASAM MCD-2 MC standard. Due to the limitations for STRUCTURE_COMPONENT, only the following type definitions can be referenced: ▪ TYPEDEF_CHARACTERISTIC of the following types: ▪ VALUE ▪ VAL_BLK with a maximum of 2 dimensions ▪ ASCII ▪ TYPEDEF_MEASUREMENT with a maximum of 3 dimensions ▪ TYPEDEF_STRUCTURE with a maximum of 2 dimensions If other type definitions are referenced, the block is not saved.
VIRTUAL	If this keyword is defined for a MEASUREMENT block, the block is not saved.
CALIBRATION_HANDLE_TEXT DEFAULT_VALUE_NUMERIC STATUS_STRING_REF STEP_SIZE	These keywords were introduced in version 1.6 of the ASAM MCD-2 MC standard. If one of these keywords is defined for a block, the block is saved, but without the information provided by the keyword.
CUBE_4 CUBE_5	These keywords were introduced in version 1.6 of the ASAM MCD-2 MC standard. If one of these keywords is defined for a block, the block is not saved.

A2L Keyword	Description
DISCRETE	
BLOB BYTE_ORDER TRANSFORMER TRANSFORMER_IN_OBJECTS TRANSFORMER_OUT_OF_OBJECTS TYPEDEF_BLOB	These keywords were introduced in version 1.7 of the ASAM MCD-2 MC standard. If one of these keywords is defined for a block, the block is not saved.
CONSISTENT_EXCHANGE CONVERSION ENCODING INPUT_QUANTITY LIMITS MODEL_LINK OVERWRITE STATIC_ADRESS_OFFSETS THIS	These keywords were introduced in version 1.7 of the ASAM MCD-2 MC standard. If one of these keywords is defined for a block, the block is saved, but without the information provided by the keyword.

Limitations for A2L files specific to ControlDesk

There are some limitations for A2L files that are specific to ControlDesk:

No support for noncontiguous bit masks ControlDesk supports the calibration of contiguous bit masks. If a bit mask is not contiguous, ControlDesk changes all 0 bits located between the 1 bits. For example, a bitmask defined as `010010100` is changed to `011111100`.

No support for bit masks for 64-bit integer variables ControlDesk does not support bit masks for 64-bit variables. ControlDesk displays these variables, but without the specified bit mask.

Limitation for editing weak parameter limits ControlDesk allows you to edit weak parameter limits. ControlDesk will automatically set the hard limits equal to the weak limits if no hard limits were defined in the A2L file used in the experiment.

There is no limitation if hard and weak limits were defined in the A2L file.

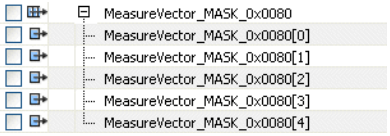
As a workaround, add the A2L file to the platform/device again. For instructions, refer to How to Add a Variable Description to a Platform/Device on page 20.

Limitations for experimenting with pointered parameters

- If the A2L file contains pointered parameters without initial data, ControlDesk's instruments display no values for these parameters after you start online calibration. You can use the Refresh Values (ControlDesk Calibration and Data Set Management 🕮) command to display the parameter values. However, these values are saved neither to a data set nor to the mirrored memory of ControlDesk. As a consequence, no parameter values are displayed when you save, close and reopen the experiment.
- You cannot experiment with pointered parameters from the A2L file of a virtual ECU (V-ECU) using ControlDesk.

Name of array variable must not end with an index in square brackets The name of an array variable must not end with an index in square brackets.

When you add an A2L variable description file with an array variable to a device, ControlDesk generates an array variable for the array and one variable for each array element. As an example, the illustration below shows this for a measurement array:



The names of the variables generated for array elements end with an index in square brackets, which causes problems when the name of the array variable itself ends with an index in square brackets.

Related topics

Basics

Basics on Importing A2L Files of Version 1.7 16
General Limitations for Variable Management 185
Problems with A2L Files 176
Variable Descriptions Supported by ControlDesk 10

Limitation for DBC Files

Introduction

There is a limitation when working with DBC files.

Unsupported signals in DBC files

ControlDesk does not support the following signals in DBC files:

- Signals with the `DBC_ST_EXT_MODE` (extended mode signals)

  - OR -

- Signals longer than 64 bits

When you add a DBC file with such signals to a CAN Bus Monitoring device, the signals are not available in ControlDesk.

Related topics

Basics

General Limitations for Variable Management 185
Variable Descriptions Supported by ControlDesk 10

Limitations for Calculated Variables

Introduction	There are some limitations when working with calculated variables.
At least one assigned signal required	When you define a calculated variable, you have to assign at least one signal to it. You cannot define a calculated variable that is based only on constants.
Measuring in a synchronous raster	A calculated variable can be calculated and measured synchronously to a measurement raster, for example, of an ECU only if at least one of the assigned signals of the calculated variable is measured in the raster concerned. Otherwise, the calculated variable can be measured only in a measurement raster that is not related to a raster, for example, of an ECU.
Root node level for new calculated variables	For devices with variable descriptions in the DBC file format (CAN-based devices), new calculated variables can be defined or imported only on the root node level of the active variable description.
No access to measurement arrays or value blocks	You cannot use measurement arrays or value blocks as input signals for calculated variables. However, you can use subelements of measurement arrays and value blocks as input signals.
Input signals from different platforms/devices	When you load an experiment containing calculated variables that reference *input signals from different platforms/devices (or different multiprocessor system members)*, these calculated variables might be marked as invalid in the log file and/or in instruments. **Workaround** Start measuring after loading such an experiment. This validates the previously invalid calculated variables. Refer to Start Measuring (ControlDesk Measurement and Recording 🕮).
Memory consumption due to continuous measurement	Calculated variables are measured continuously even if they are assigned to a triggered measurement raster. As a result, the memory required by the measurement buffer is large; especially in connection with high-frequency measurement signals. **Workaround** Reduce the size of the measurement buffer. The default size is 60 s. Refer to Acquisition Properties (ControlDesk Measurement and Recording 🕮).
No visualization in the Index Plotter	Calculated variables cannot be visualized in the Index Plotter.

Workaround Visualize calculated variables in a Time Plotter instead of in an Index Plotter.

Measurement rasters of input signals

To measure calculated variables, all the input signals must be measured in the same measurement raster if at least one input signal is measured in a triggered raster.

In other words: Different measurement rasters can be used for the input signals of a calculated variable only if none of the rasters is triggered.

Measurement rasters of input signals from multicore/multiprocessor platforms

When you use a triggered measurement raster for an input signal of a calculated variable, all the other input signals of the calculated variable:

- Must come from the same multicore/multiprocessor node
- Must be measured in the same raster

No measurement of calculated variables via automation

Measuring calculated variables via automation is not supported.

Related topics

Basics

Introduction to Calculated Variables.....49

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 198
A 198
B 199
C 200
D 204
E 208
F 211
G 212
H 212
I 213
K 215
L 215
M 216
N 219
O 219
P 221
Q 223
R 224

S....225
T....228
U....229
V....230
W....232
X....233

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 applications or ECU applications.

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 maps. Calibrating the data points of a common axis affects all the curves and/or maps referencing the axis.

Common axes are represented by the symbol.

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

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

or

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Conversion type The type of a computation method, for example a linear function or a verbal computation method.

Curve A parameter that consists of

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

Curves are represented by the  symbol.

D

DAQ module A hardware module for the acquisition of physical quantities

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

Data logger An object in the Measurement Configuration controlbar that lets you configure a data logging.

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

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

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

- Reference data set
- Sub data set
- Unassigned data set
- Working data set

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

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

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

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

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

Debug interface An ECU interface for diagnostics tasks and flashing.

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

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

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

ControlDesk provides the following devices:

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

Each device usually has a variable description that specifies the device's variables to be calibrated and measured.

Diagnostic interface Interface for accessing the fault memory of an ECU.

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

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

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

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

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

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

Proprietary diagnostics database formats are not supported by ControlDesk.

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

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

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

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

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

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

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

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

Documents folder A standard folder for user-specific documents.

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

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

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

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

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

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

host PC. Unlike the host service, the DsDAQ service lets you perform, for example, triggered measurements with complex trigger conditions.

The following platforms support applications that contain the DsDAQ service:

- DS1007 PPC Processor Board platform
- DS1202 MicroLabBox platform
- MicroAutoBox III platform
- SCALEXIO platform
- VEOS platform
- XIL API MAPort platform

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

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

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

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

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

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

E

ECU Abbreviation of *electronic control unit*.

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

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

ECU diagnostics Functions such as:

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

ControlDesk provides the ECU Diagnostics device device to access ECUs for diagnostic tasks. Communication is via diagnostic protocols 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 configurations.

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 sources and events. 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 events. *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 layouts.

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 instruments 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 instruments. 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 s connected to variables of one or more simulation models.

Layout Navigator A controlbar that displays all opened layout s. It can be used for switching between layouts.

Layout script A Python script used to extend the functionality of a layout .

Leading raster The measurement raster that specifies the trigger settings for the Time Plotter 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 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 variables 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 pages 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 captures 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 pages 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 pages. 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 segments.

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 signals and operation signals.

  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 instruments 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 controlbars 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 variables in instruments:

- Measurement variables 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.

X

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.

A

A2L files
  problems 176
adding a variable description to a platform/device 20
AML files
  problems 178

C

calibrating parameters
  conversion formula 43
  converted mode 43
  displaying a value in converted or source mode 43
  source mode 43
Common Program Data folder 8, 203
ControlDesk Options dialog
  Variables page 146
conversion formula 43
converted mode 43
Customize Columns dialog 111

D

displaying a value in converted or source mode 43
Documents folder 8, 207

F

favorites list 31
filter 33
filtering the variable list 33

H

handling variables in the Variable Browser 30

I

incremental search 41
Instrument Selector 214

L

Local Program Data folder 8, 215

M

Measurement Data Pool 217
merged raster 52
Messages controlbar 218

P

platform management
  adding a variable description to a platform/device 20
  Select ECU Image File dialog 99
  Select Variable Description dialog 98
Platforms/Devices controlbar 222
problems
  A2L files 176
  AML files 178
project and experiment management
  Select ECU Image file 99
  Select Variable Description dialog 98
Project controlbar 223
Project Manager 223
Properties controlbar 223

S

searching for a variable using incremental search 41
Select ECU Image file 99
Select ECU Image File dialog 99
Select Variable Description dialog 98
source mode 43
standard filter 33

T

tree view 31

V

variable list 31
variable management
  Customize Columns dialog 111
  favorites list 31
  filter 33
  filtering the variable list 33
  incremental search 41
  searching for a variable using incremental search 41
  Select ECU Image File dialog 99
  Select Variable Description dialog 98
  standard filter 33
  tree view 31
  variable list 31
  Variable Properties dialog 139
  wildcard filter 33
Variable Properties dialog 139
variables using conversion tables 44
variables using multiscalings 47

W

wildcard filter 33