ModelDesk

Basics

For ModelDesk 5.5

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

Contents

This document introduces you to ModelDesk. It provides basic information of ModelDesk and describes its frame work.

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

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 ModelDesk

Features of ModelDesk

Introduction	The following introduction gives you an overview of the features of ModelDesk.
Parameterizing ASMs	Automotive Simulation Models (ASM) are open Simulink models for real-time and Simulink simulation. You can use the models for hardware-in-the-loop testing of electronic control units or for early validation of controller algorithms by Simulink simulation during the design phase. For you to enter individual parameter values, ModelDesk has parameter pages with illustrations for each component. You can enter parameters in controls beside the components. Table parameters can be visualized as 3-D graphs and modified using the Table Editor. To navigate to the parameter pages, ModelDesk has navigation pages. The navigation pages visualize model components and their subsystems. They are represented by a hierarchy. The model components to be parameterized can be selected from the top level. You see the vehicle model and can browse through its systems, guided by graphical representations of the modeled components. You can use ModelDesk's Processing component to calculate parameter values based on measurement data and other parameter.
Project management	ModelDesk's **Project Navigator** provides a means of organizing and managing large-scale model parameterization projects. You can create and assign parameter files to each model component, for example, differential, tires, road. Complete vehicle parameter sets can be created and managed. You can select existing parameter files from a parameter pool and apply them by drag & drop.
Platform management	ModelDesk has a **Platform Manager** from which you can access real-time hardware or VEOS to load and control the real-time applications or offline simulation applications.
Supported platforms and simulation modes	ModelDesk can parameterize three types of simulation models: ▪ Real-time simulation A simulation running on a real-time system: ▪ DS1006 Processor Board ▪ DS1007 PPC Processor Board ▪ DS1202 MicroLabBox

- SCALEXIO Processing Unit
- DS6001 Processor Board (processor board for SCALEXIO systems)
- MicroAutoBox II
- MicroAutoBox III

- Simulink simulation
  - A simulation running in MATLAB®/Simulink®
- Offline simulation
  - A simulation running in VEOS on the same host PC and a remote PC

Processing

The Processing component provides a convenient way to parameterize simulation models on the basis of measurements. This is especially useful if you want to parameterize engine models with data measured on a test bench.

Road Generator

The Road Generator is a component of ModelDesk. It has a graphical user interface for defining road networks with road elements and junctions. Each road element and junction can have individual lateral and longitudinal profiles, and there are various options for configuring the road surfaces and scenery along the road. There are different views in the Road Generator that let you parameterize the different elements of a road network. The results of the settings are visible immediately in 1-D and 2-D previews. The Road Generator also creates a 3-D geometry file to be used for animation in MotionDesk.

Scenario Editor

Scenarios define the stimulations and movements of the ASM vehicle and the movements of *fellow vehicles (fellows)* relative to the *ASM vehicle*. You can create scenarios with the Scenario Editor.

In ModelDesk 4.6 and below, this component was called Traffic Editor because this component was used to define only the movement of fellows (traffic scenarios).

ModelDesk Testing

The testing component lets you specify and execute tests in ModelDesk. Logical test cases specify the environment of the tests and are based on a parameter set, road, and scenario. Logical test cases contain concrete test cases. A concrete test case is used to specify concrete values of the variables that are modified during the tests. The test results are validated and reported.

Traffic Object Manager

You can create and manage traffic objects with the Traffic Object Manager. Traffic objects are used as traffic participants in scenarios or as static objects, for example, traffic signs, in road networks.

Plot Manager

You can use the Plot Manager to visualize signals of the simulation model and to manage the simulation results. It provides layouts containing up to four plotters where you can drag signals for monitoring. The monitored signals can be

saved. So you can compare signals of different simulation runs, for example, to analyze the results of modified parameter values.

Custom libraries

If you want to add your own Simulink blocks to an Automotive Simulation Model (ASM), you must store these blocks in custom libraries. ModelDesk lets you integrate these custom libraries and visualize custom blocks from a custom library integrated in an ASM. You can enter parameter values in parameter pages generated for the custom blocks and integrated in the hierarchy.

Automating

The ModelDesk automation interface allows you to automize ModelDesk using scripts. Automation with Python scripts is described in detail.

Overview of ModelDesk

Where to go from here

Information in this section

How to Start and Shut Down ModelDesk

Objective

After installing the dSPACE software, you can start ModelDesk via the Microsoft Windows Start menu.

Part 1

To start ModelDesk

1 In the Start menu, select All Programs – dSPACE Modeling Software <x.y> – dSPACE ModelDesk <x.y>.

Tip

You can create a shortcut to an experiment on the desktop. This starts ModelDesk and loads the project and experiment automatically. Refer to Create Shortcut (ModelDesk Project and Experiment Management 🕮).

Result

ModelDesk opens.

Part 2

To shut down ModelDesk

1 In the File ribbon, click Exit.

Result

ModelDesk shuts down. Before shutting down, ModelDesk asks whether you want to save the project if it was changed.

Related topics

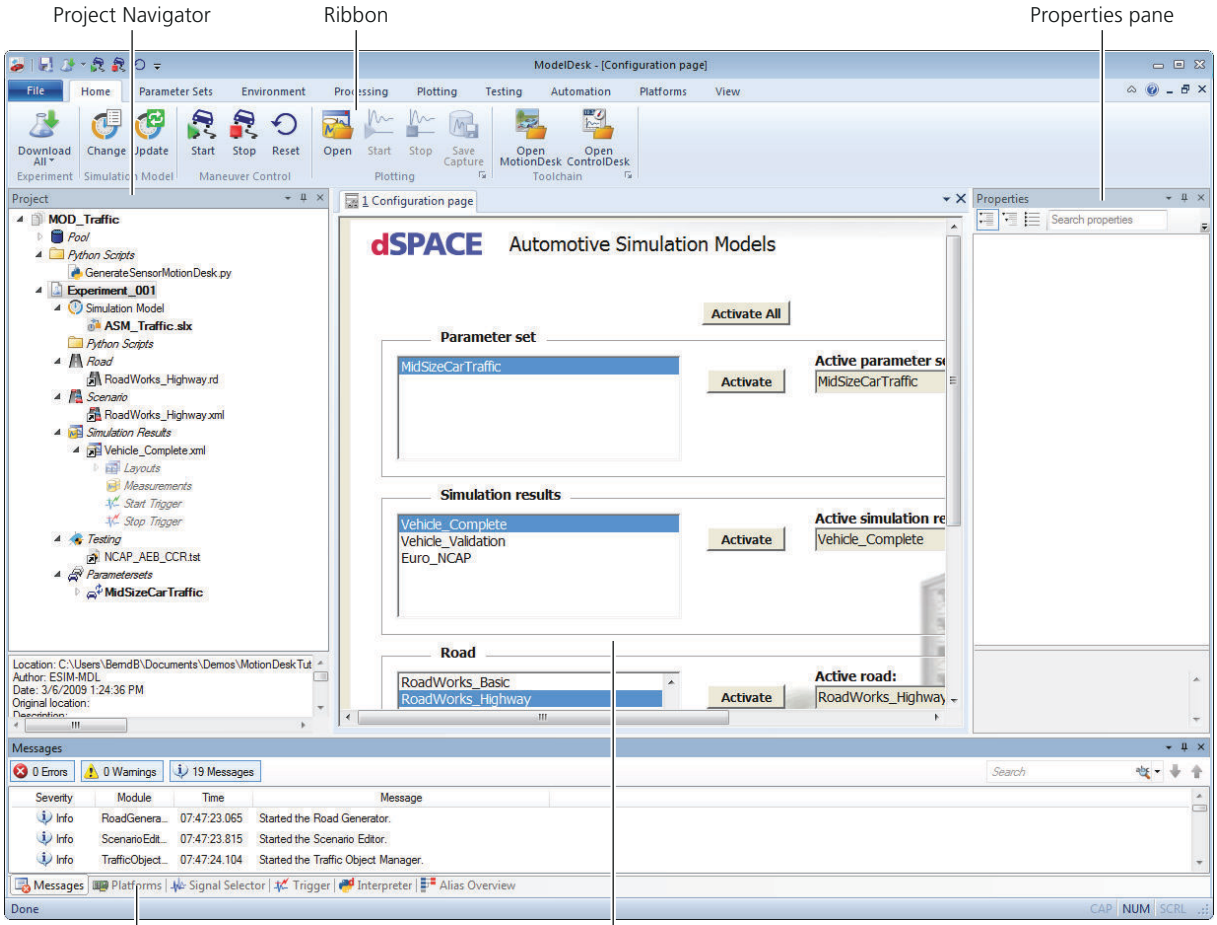
References

Create Shortcut (ModelDesk Project and Experiment Management 🕮)
Exit....49
Start Page....61

User Interface of ModelDesk

Graphical overview

ModelDesk's main window offers the basic environment for all other elements and tools provided by ModelDesk.



- Messages
- Platforms
- Signal Selector
- Trigger
- Interpreter
- Alias Overview
- User Functions Output

Ribbon

The ribbon provides access to ModelDesk's functions and commands. Most of ModelDesk's components provide commands in the ribbon for quick access to frequently used commands, see the following table.

Ribbon	Ribbon Group	Description
File	–	To work with ModelDesk projects and experiments.

Ribbon	Ribbon Group	Description
Home	Experiment	To download all the settings done in ModelDesk and active to the simulation model (parameter values, road model, maneuver, and scenario)
	Simulation Model	To update or change the simulation model.
	Maneuver Control	To control the maneuver.
	Plotting	To start or stop plotting, or save the capture.
	Toolchain	To start MotionDesk or ControlDesk.
Parameter Sets	Parameter Set	To work with the parameter set.
	Simulation Model	To update or change the simulation model.
	Navigation	To navigate one level up when you navigate to navigation and parameter pages.
	Custom Libraries	To register, unregister, or update custom libraries.
Environment	Road	To work with the road models specified with the Road Generator.
	Scenario	To work with the scenarios specified with the Scenario Editor.
	Traffic Objects	To manage the traffic objects.
	Traffic Driver	To manage traffic driver objects.
	File	To save the active document.
	Scene Synchronization	To synchronize the settings done in ModelDesk (road model, scenery, traffic objects) with the scene in MotionDesk.
	Environment	To configure settings of the environment model for the maneuver, road, traffic, and testing.
Processing	Configuration	To configure the processing.
	Measurement Type	To configure measurement types.
	Measurement Data	To configure measurement data.
	File	To save the files.
	Parameter Functions	To import or export the parameter mapping.
Plotting	Configuration	To configure the plotting.
	Plotting	To start or stop plotting, or save the capture.
Testing	Test	To specify and execute test cases.
Automation	Python Scripts	To manage the Python scripts.
	Interpreter	To run a script or import modules.
	Edit Script	To work with the Source Code Editor
	User Functions	To configure and start user functions.
Platforms	Platform Management	To manage the simulation platforms.

Ribbon	Ribbon Group	Description
View	Controlbar	To hide or show controlbars and to reset the view.
	Show	To show the log file, start page or status bar and to specify the workbook mode.
	Window	To manage ModelDesk's windows.
	Road Views	To reset the window of the Road Generator.

ControlBars

ModelDesk contains several controlbars.

User Functions Output A controlbar that provides access to the output of external tools added to the Automation ribbon.

Message Viewer A controlbar displaying a history of all error and warning messages that occur during work with ModelDesk.

Alias Overview A controlbar that gives an overview of all the alias names defined in the active experiment.

Properties A controlbar providing access to the properties of, for example, platforms, road, and scenarios.

Signal Selector A controlbar that is used to select model variables to be plotted in ModelDesk or for triggering the plot using the Plotting component.

Trigger A controlbar that is used to specify start and end trigger when you plot signals in ModelDesk using the Plotting component.

Platforms A software component represented by a controlbar. It provides functions to handle devices, platforms, and the applications assigned to the platforms.

Project Navigator A software component represented by a controlbar. It provides access to projects and experiments and all the files they contain.

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

Tip

Controlbars can be hidden or shown using the Switch Controlbars command.

Status bar

The status bar displays the current state of ModelDesk.

Working area

The working area allows you to view and edit experiments, roads, scenarios, and test cases. It is also used to configure parameter sets and parameterize models via the parameter pages.

Related topics

Basics

Basics on Ribbons....26
Basics on Working with the Graphical User Interface....24

References

Switch Controlbars....85

ModelDesk Elements

Introduction

Some of the terms used in ModelDesk and its documentation are explained below.

Project

A project manages different experiments. One model can be added to each experiment for parameterization. The project holds the experiments related to the parameterization tasks, and the Pool is filled with relevant parameter files for the entire project.

Experiment

An experiment is the basis for carrying out the parameterization of a model. It allows you to manage all the files related to this task, such as

- Simulation model
- Parameter sets
- Roads
- Scenarios
- Simulation results
- Tests

Pool

The Pool is the container for all the project-specific files such as parameter, road and scenario files. From there, they are available for the entire project and linked to the experiments belonging to it.

Simulation model

You can add several model types (real-time model, Simulink model and VEOS model) to one experiment, and use them for simulation. Although a ModelDesk

	experiment can contain several model types, you can work with only one model type – the active one – at a time. To work with the other model type, you have to activate it first. The application which corresponds to a model is executed on a suitable platform. If a platform is assigned to a model, the platform name is added in brackets to the item of the simulation model in the Project Navigator. If no platform is assigned to the model, the icon of the simulation model has an exclamation mark.
Parameter set	A parameter set contains all the parameters of a vehicle variant. Before you set values of individual parameters, you must configure a parameter set. For example, parameter set configuration provides drivetrain settings including front-, rear-, and all-wheel drive.
Navigation page	Navigation pages group and link parameter pages in a hierarchy corresponding to the model structure. By using the navigation pages, you can select the parameter page from the top level and parameterize the model component.
Parameter page	For parameter entry, ModelDesk has parameter pages with illustrations for each model component. You can enter parameters in edit fields beside the illustrations. Table parameters are visualized as 3-D graphs and modified by the Table Editor.
Roads	Roads are built in ModelDesk's graphical Road Generator. You can build simple road elements having different lanes, surfaces, profiles, or sceneries. The road elements can be connected to a road network using junctions. This gives you a flexible road environment for testing different kinds of ECU's.
Scenario	Scenarios define the stimulations and movements of the ASM vehicle. You can specify only a maneuver of the ASM vehicle or specify traffic situations by defining additional traffic participants (fellows). You can define the movements of fellows with absolute values or relative to the ASM vehicle. You can create scenarios with the Scenario Editor.
Test	Testing allows you to specify and execute tests within ModelDesk. You can define tests and vary the environment for the ASM vehicle. Testing starts the simulation and validates the simulation results on basis of evaluation function.

Related topics

Basics

Starting with ModelDesk

Introduction

This overview shows the workflow for parameterizing and helps you start your project.

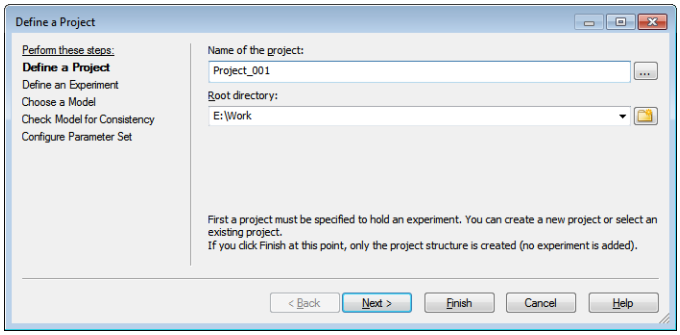
Workflow for parameterizing models

If you want to start parameterizing, you must first access the simulation platform, organize your project, and make the necessary settings:

1. Access the simulation platform where the simulation model is executed. If the platform is not already registered at the host PC or the application is not loaded, you can use ModelDesk to do this. Refer to Basics of Platform Manager (ModelDesk Platform Management 🕮).
2. Define a new project as the container of the experiment and all the project files belonging to it. Refer to How to Define a Project (ModelDesk Project and Experiment Management 🕮).
3. Define a new experiment. An experiment is the basis for carrying out the parameterization for one added model, and allows you to manage all the files and parameter sets for it. Refer to How to Define an Experiment (ModelDesk Project and Experiment Management 🕮).
4. Choose a simulation model (real-time application for real-time hardware, Simulink model for MATLAB/Simulink, or offline simulation application for VEOS) and check it for consistency. Refer to How to Choose a Model and Initialize the Consistency Check (ModelDesk Parameterizing 🕮).
5. Configure a parameter set. A parameter set contains all the parameters of a model including its subsystems, and you must configure it before you can specify individual parameters. Refer to How to Configure a Parameter Set (ModelDesk Parameterizing 🕮).
6. You can start parameterizing the model:
   - To modify parameters of the simulation model, refer to ModelDesk Parameterizing 🕮.
   - To calculate parameter values, refer to ModelDesk Processing 🕮.
   - To create a road, refer to ModelDesk Road Creation 🕮.
   - To specify a scenario (maneuver of the ASM vehicle and movements of traffic participants), refer to ModelDesk Scenario Creation 🕮.
7. You can use a plotter to display and compare the simulation results. Refer to Basics of the Plot Manager (ModelDesk Plotting 🕮).
8. You can specify and execute tests within ModelDesk. Refer to Basics of ModelDesk Testing (ModelDesk Testing 🕮).

Project wizard

ModelDesk's project wizard leads you through all the necessary steps for defining a project and experiment, choosing a model, and configuring a parameter set. The wizard consists of 5 dialogs, one for each step, which you go through by clicking **Next**. The illustration below shows the **Define a Project** dialog, as an example.



Related topics

Basics

Features of ModelDesk 10

References

Project Wizard (ModelDesk Project and Experiment Management 🕮)

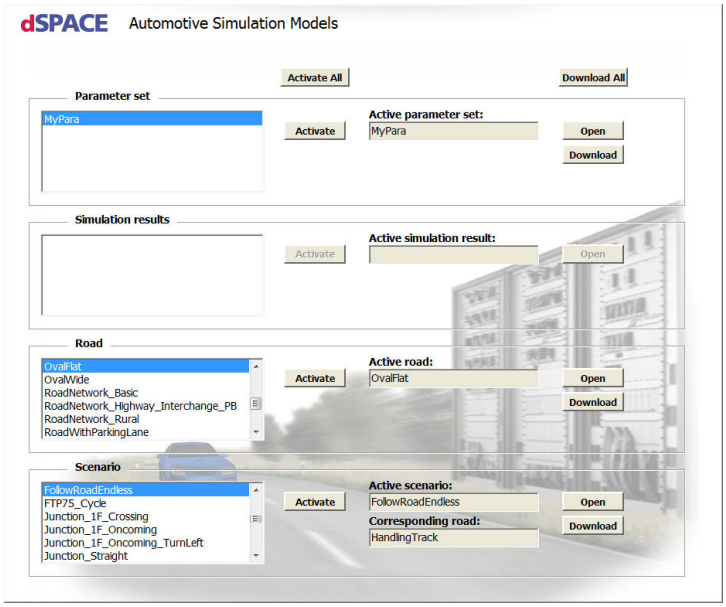
Managing an Experiment

Introduction

When your experiment contains several parameter sets, roads, and scenarios you can activate, open or download one of them by one central page.

Configuration page

The Configuration page is displayed when you double-click the Experiment node in the Project Navigator.



This page shows all roads, scenarios, and parameter sets which are available in your experiment. You can use this page to activate, open or download one of them.

Related topics

References

Configuration Page (ModelDesk Project and Experiment Management 🕮)

Changing the Background Image of the Configuration Page

Introduction

You can change the background image of the Configuration page.

Background image

The Configuration page has a background image. The default image file can be replaced by any other image file. The image file must have the following properties:

- Name: `Config_page.jpg`
- Format: JPG
- Size: 930 × 794

The image file must be copied to `<RCP_HIL_InstallationPath>\ASM\MODHtml\Config_page.jpg`.

Related topics

References

Configuration Page (ModelDesk Project and Experiment Management 🕮)

Basics on Working with the Graphical User Interface

Introduction

ModelDesk's user interface provides some features that are common to all components. You can modify the user interface to best fit your usage.

Where to go from here

Information in this section

How to Customize the Screen Arrangement

Objective

The screen arrangement defines which panes are displayed and how they are arranged. The first time you execute the application, it starts with its default screen arrangement, which you can modify.

Screen modifications

The screen arrangement contains information about:

- Display states and positions of the toolbars
- Display states and positions of the panes such as the **Project Navigator**
- Pane settings, such as the docking state

Saving the screen arrangement

All the modifications you made to the screen are automatically saved to the current screen arrangement when you exit the application. You cannot save them explicitly.

Resetting the screen arrangement

To discard all the modifications that you made in the screen arrangement, you can reset it to its default. Use the **Reset to Default** command from the **View – Controlbar** ribbon to make the user interface look like it did the first time you started the application.

Commands for customizing the screen arrangement

The application provides various commands, such as the **Floating** command, to modify the state of each pane. You can execute all of these commands quickly and flexibly via mouse. The following instructions describe how to do this.

Method

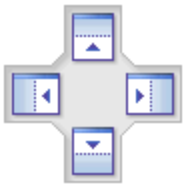
To customize the screen arrangement

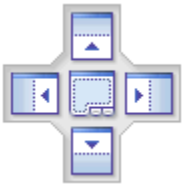
1. Move the mouse pointer onto the title bar of the pane whose position you want to change.

   If you want to move a tabbed pane, you must select its tab instead of the title bar.

2. Drag the pane to another position while holding the left mouse button down.

   The docking state of the pane is automatically changed to *floating* and the screen displays *docking stickers* that you can use to specify the new position.

Docking Sticker	Description
	The pane is docked to the top of your application's main window.
	The pane is docked to the bottom of your application's main window.
	The pane is docked to the left of your application's main window.
	The pane is docked to the right of your application's main window.
	The pane is docked to the top, bottom, left, or right of your application's working area.

Docking Sticker	Description
	The pane is docked above, below, to the left, or to the right of the selected pane. If you drag the mouse onto the middle docking sticker, the pane is docked as a new page.

3 Move the mouse pointer onto a docking sticker. When the area of the new position is displayed, release the left mouse button.

Result

The component is moved to the new position in the user interface and docked to another component.

If you release the mouse button anywhere except on a docking sticker, the docking state of the pane remains floating.

Tip

If you want to change the order of pane tabs, you can drag them to new positions.

Related topics

References

Allow Docking....73
Auto Hide....74
Dock/Docked....79
Float in Main Window....81
Float/Floating....80

Basics on Ribbons

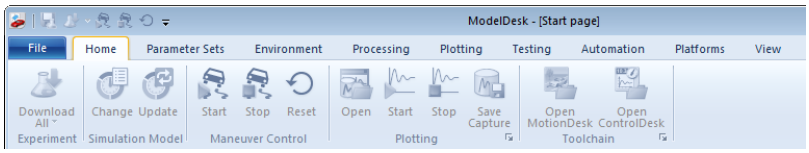
Introduction

Ribbons are user interface elements that provide access to common commands and dialogs.

Ribbon

The ribbon organizes and groups commands of a program. The ribbon is located at the top of the user interface.

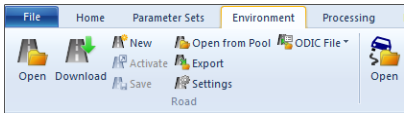
The ribbon consists of several ribbon tabs, see the following example of the Home ribbon.



Ribbon group

A ribbon group is a part of a tabbed ribbon. It consists of a set of related commands.

The following illustration shows the Road ribbon group in ModelDesk as an example.

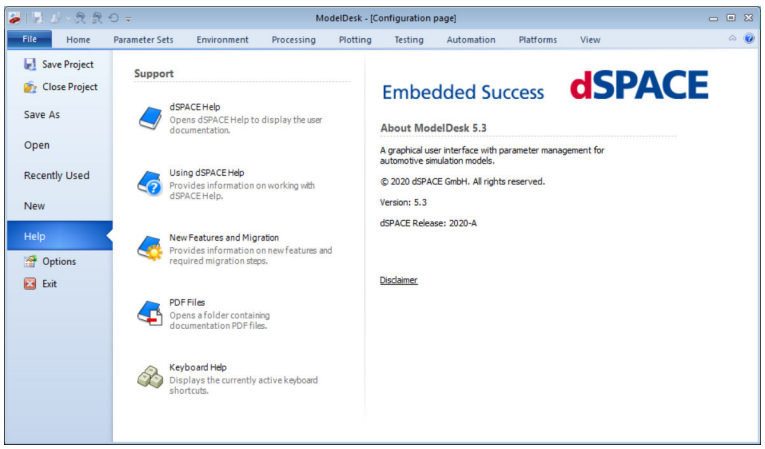


Dialog launcher A dialog launcher is an optional element of a ribbon group that lets you open a dialog related to that ribbon group.

Backstage view

The Backstage view provides basic commands of a software program, such as Save, Open, Close, Exit. The Backstage view can be identified by the colored ribbon tab. When the Backstage view is opened, it covers the entire user interface.

The following illustration shows the Backstage view of the Help ribbon group as an example.



Quick Access Toolbar

The Quick Access Toolbar is an easy way to call commands. You can customize it to provide the commands you use most frequently.

For information on how to add or remove commands to/from the Quick Access Toolbar, refer to How to Customize the Quick Access Toolbar on page 28.

Navigating the ribbon

You can navigate the ribbon via mouse and via keyboard.

Navigation via mouse You can navigate the ribbon with the mouse scroll wheel.

Navigation via keyboard If you want to navigate the ribbon via keyboard, press the **Alt** key. Each command in the Quick Access Toolbar and each ribbon tab then is marked by an access key.

If you then press one of the ribbon tab access keys, each command in the selected ribbon is also marked by an access key.

You can remove the access keys by pressing **Alt** again.

Related topics

References

Customize Ribbon 78

How to Customize the Quick Access Toolbar

Objective

You can customize the Quick Access Toolbar to display the commands that you use frequently and you can specify its position.

Method

To customize the Quick Access Toolbar

1 On the Quick Access Toolbar, click – More Commands.
The Quick Access Toolbar page of the Customize dialog is displayed.

2 On the Quick Access Toolbar page you can add and remove commands, reset the toolbar and specify the position of the Quick Access Toolbar.
- To add a command to the Quick Access Toolbar, open the Choose commands from list and select a ribbon. Then select a command from the list on the left and click Add.

Tip

To add a command to the Quick Access Toolbar, you can also select **Add to Quick Access Toolbar** in the context menu of a command in a ribbon.

- To remove a command from the Quick Access Toolbar, select the command in the list on the right and click **Remove**.

Tip

To remove a command from the Quick Access Toolbar, you can also select **Remove from Quick Access Toolbar** in the context menu of a command icon in the Quick Access Toolbar.

- To reset the Quick Access Toolbar to the factory default, click **Reset**.
- To show the Quick Access Toolbar below the ribbon, select **Show Quick Access Toolbar below the ribbon**.

3 Click **Close** to save the changes.

Result

You have customized the Quick Access Toolbar.

Related topics

Basics

Basics on Ribbons.....26

References

Customize Quick Access Toolbar / More commands.....77

Customizing Tables

Introduction

Some components of ModelDesk use tables that you can customize according to your needs.

Note that customizing is available only for some tables, and features may be disabled for specific tables.

Basics

The customization commands for are available in the context menu of the column header. If the described command is not visible, the corresponding feature is disabled.

Adapting column widths

You can adapt the width of the columns by dragging the column header border left or right and by using the following commands:

Command	Purpose
Best Fit	To optimize the width of one column.
Best Fit (all columns)	To optimize the widths of all columns.

Sorting rows

You can sort the rows by using the following commands:

Command	Purpose
Sort Ascending	To sort the displayed elements in ascending order.
Sort Descending	To sort the displayed elements in descending order.
Clear Sorting	To return to the default order from previously sorted data elements.

Grouping rows

You can group the rows by using the following commands:

Command	Purpose
Group by This Column	To group the rows according to the entries of the column.
Ungroup	To revoke grouping the rows according to the entries of the selected column.
Show Group Panel	To show the group panel. You can select or clear column headers by dragging them to or from the group panel. If you drag multiple column headers to the group panel, the resulting row grouping is arranged hierarchically.
Hide Group Panel	To hide the group panel.
Full Expand	To fully expand all the rows.
Full Collapse	To collapse all the rows.

Hiding columns

You can hide the rows by using the following commands:

Command	Purpose
Show Column Chooser	To show the column chooser. The column chooser lets you add a column to the table and opens a dialog displaying the columns that can be added to the table. To add a column, drag it from the dialog to the column header. To remove a column from the table, drag its header below the table.
Hide Column Chooser	To hide the column chooser.

Command	Purpose
Hide <xyz> Column	To hide the <xyz> column.
Group Summary Editor[1)]	To display summary information for selected items in each group row.

1) The command is available via the context menu of column headers that are moved to the group panel.

Filtering contents

You can use filters to reduce the number of entries displayed by using the following commands:

Command	Purpose
Filter Editor	To specify a filter. Refer to How to Specify and Use a Filter on page 31.
Clear Filter	To clear the filter.

How to Specify and Use a Filter

Objective

If a table has a lot of entries, you can use a filter to reduce the number of entries displayed. A filter can be used for example, on the Surface and Processing Configuration pane of the Road Generator.

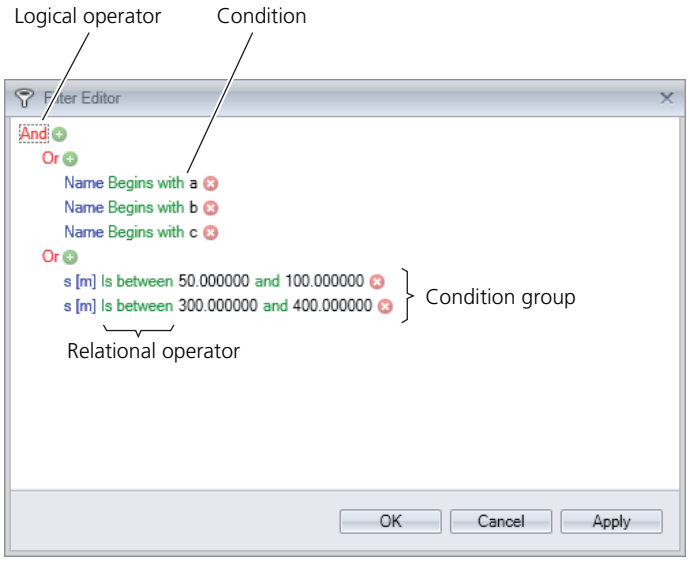
Possible methods

You can use the Filter Editor to specify a filter containing several conditions or use a simple autofilter based on the table entries.

- To specify a filter using the Filter Editor, refer to Basics of the Filter Editor on page 31 and Method 1 on page 33.
- To use an autofilter, refer to Method 2 on page 34.

Basics of the Filter Editor

A filter can consist of several conditions that are combined by logical operations. You can group conditions to create a multistage filter. In a condition, a column header of the table is compared with a specified value. An entry is displayed only if it fulfills the filter criteria. The following illustration shows an example.



Logical operator An operator that compares several conditions or condition groups. The result is either true or false. The Filter Editor provides four logical operators (AND, OR, NOR, NAND). Logical operators are displayed in red.

Relational operator An operator that compares two values. The Filter Editor provides relational operators for mathematical and string comparisons. Relational operators are displayed in green.

Column header The table header column that is used to build a filter condition.

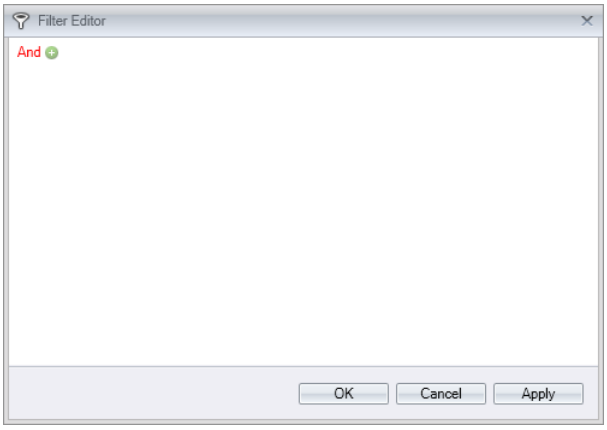
Condition An expression that compares a value of the table column header with a specified value via a relational operator. The result is either true or false.

Condition group A group of two or more conditions that are combined by a logical operator. The result is either true or false. A condition group can also contain a condition group, so you can build a multistage filter.

Method 1

To specify and use a filter using the Filter Editor

1 Click to open the Filter Editor.



The Filter Editor opens. It contains a red word: And. This is a logical operator.

2 Click the red word.

The Filter Editor opens a menu for you use to change the logical operator or add a condition or condition group:

- To change the logical operator, select And, Or, NotAnd, or NotOr.
- To add a condition, select Add Condition.
- To add a condition group, select Add Group.

3 If you have added a condition, specify it:

- Click the blue word to select a column header.
- Click the green word to select a relational operator.
- Click the gray word(s) to specify the values.
- To delete a condition, click it first and then or press - or **Del**.

4 Repeat the previous steps until the filter is complete.

5 Click Apply to apply the filter.

When the filter is active, only entries that meet the filter criteria are displayed. If you are not satisfied with the result, you can repeat the previous steps.

6 Click OK

The Filter Editor dialog is closed, and the filter is active. The active filter is displayed in the line below the table.

7 To clear the filter, clear the checkbox or click .

Cleared filters are not deleted. They are still available.

8 To use a defined filter, click the down arrow to open the filter list and select the filter.

Method 2

To specify and use a filter via autofilters

1. Click the filter symbol in the column header.
   A list of all the column entries opens.
2. Select the entries to display.
3. Click outside the list.

Result

Fewer entries are displayed.

Related topics

HowTos

How to Customize the Screen Arrangement 24

References

Edit Filter 48

Basics on User Functions

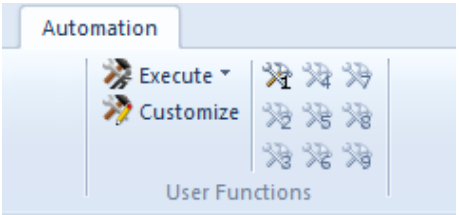
Introduction

You can specify external applications as user functions so that they can be started via ModelDesk's ribbon.

User functions

ModelDesk allows you to embed external applications as *user functions*. A user function is available in the Automation ribbon as a button in the User Functions ribbon group.

For example, you can add Windows' notepad to ModelDesk as a user function.



To configure a user function, you have to specify an executable file. Files with the EXE, COM, or BAT extension, for example, are executable files. ModelDesk starts the corresponding application in a separate process.

If user functions write to the standard output, the User Functions Output controlbar displays the output.

Additionally to the command, you can specify arguments that are used when the executable file is called.

Tip

If you work with ModelDesk and MotionDesk, you can create a user function and its shortcut. This makes it possible to start MotionDesk using a shortcut in ModelDesk.
When you also specify the MotionDesk project and experiment as arguments for the user function, the specified project and experiment is loaded after you start MotionDesk. The required arguments can easily be copied from the desktop shortcut to the experiment, which you can create in MotionDesk using the Create Shortcut command.

Related topics

HowTos

How to Add External Programs as User Functions..... 35

References

Create Shortcut (MotionDesk Project and Experiment Management 🕮)
Customize (User Functions)..... 44

How to Add External Programs as User Functions

Objective

You can add shortcuts to external applications to ModelDesk.

Method

To add external programs as user functions

1. On the Automation ribbon, click User Functions – Customize.
   The User Functions dialog opens.
2. In the dialog, click  to add a new user function.
3. Specify a name for the user function.

Tip

To apply an accelerator (underlined character in the menu), prefix the character in the user function name with an ampersand (&), for example, Start &Editor.
Press **Alt** to visualize accelerators in the menu.

4 Enter a description and select an executable file and an image for the new user function.

5 Configure further settings, such as additional arguments or an initial directory.

6 Click **OK** to confirm your settings and close the dialog.

Result

You have added an external program or a script as user function. A new button with the specified icon appears in the **Automation – User Functions** ribbon group.

Related topics

Basics

Basics on User Functions 34

References

Customize (User Functions) 44

Reference Information

Where to go from here

Information in this section

Commands and Dialogs of the Basic Interface

Where to go from here

Information in this section

About ModelDesk

Access

Ribbon	File – Help
Context menu of	None
Shortcut key	None
Icon	None

Purpose

To display information about the ModelDesk version installed on your system.

Result

Here you can see ModelDesk's version number.

Dialog settings

Disclaimer Opens the Disclaimer dialog containing general warnings concerning the usage of ModelDesk. By default, the Disclaimer dialog is displayed, when you start ModelDesk. If you clear the Always show this warning during start-up option, it is not displayed at the next start-ups.

Related topics

References

dSPACE Help 42

Best Fit (all columns)

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Segments pane ▪ Lanes pane ▪ Surface pane ▪ Scenery pane
Shortcut key	None
Icon	None

Purpose

To adjust all columns of a table for optimal display.

Result The entire content of the columns of the table is displayed properly.

Description If the contents of columns is not displayed entirely, you can use the **Best Fit (all columns)** command to adjust the column width.

Best Fit

Access You can access this command via:

Ribbon	None
Context menu of	▪ Segments pane ▪ Lanes pane ▪ Surface pane ▪ Scenery pane
Shortcut key	None
Icon	None

Purpose To adjust a selected column of a table for optimal display.

Result The content of the selected column is displayed properly.

Description If the content of a selected column is not completely displayed, you can use the **Best Fit** command to adjust the width of the column.

Clear Filter

Access You can access this command via:

Ribbon	None
Context menu of	None
Shortcut key	None
Icon	

Purpose To clear a filter.

Result

The filter is cleared. All elements are shown in the table.

Related topics

HowTos

How to Specify and Use a Filter 31

References

Edit Filter 48

dSPACE Help

Access

You can access this command via:

Ribbon	File – Help
Context menu of	None
Shortcut key	**F1**
Icon	
Others	(at the right side of the ribbon bar)

Purpose

To open the user documentation of ModelDesk.

Result

The user documentation of ModelDesk opens.

Related topics

References

Using dSPACE Help 63

Copy

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Parameter page ▪ Interpreter ▪ Source Code Editor
Shortcut key	**Ctrl+C**
Icon	None

Purpose

To copy the current selection to the Clipboard.

Related topics

References

Paste....57

Cut

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Parameter page ▪ Interpreter ▪ Source Code Editor
Shortcut key	**Ctrl+X**
Icon	None

Purpose

To cut the current selection to the Clipboard.

Result

This can be applied to text by using the Paste command.

Related topics

References

Paste....57

Customize (User Functions)

Access

You can access this command via:

Ribbon	Automation – User Functions
Context menu of	None
Shortcut key	None
Icon	

Purpose

To add external functions to the User Functions ribbon group

Result

The user function is added to the User Functions ribbon group.

Description

Adding external functions gives you a quick and easy access to any function you might need while working with ModelDesk.

User functions dialog

User Functions Menu Items Lets you select which of the currently available user functions you want to edit.

Description Lets you enter a description for the user function.

Command Lets you select the path and folder to open the program file from.

Arguments Lets you browse for a file or folder argument to pass to the selected user function.

Initial directory Lets you browse for the initial folder to execute the program in.

Image Lets you load, edit and clear an image for the selected user function. You can use images in bitmap or PNG format (16 × 16 pixels) to be used as an icon in the ribbon.

Show window (Applies to console applications) Indicates whether the window of the running process is displayed.

Related topics

Basics

Basics on User Functions 34

HowTos

How to Add External Programs as User Functions 35

Delete

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Parameter page ▪ Source Code Editor
Shortcut key	**Del**
Icon	None

Purpose

To delete the current selection.

Description

Unlike the **Cut** command, the selection is not placed in the Clipboard for further use.

Related topics

References

Cut....43

dSPACE Log

Access

You can access this command via:

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

Purpose

To open the dSPACE Log.

Result

The dSPACE Log is opened in the working area.

Description

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

In contrast, the **Message Viewer** only shows the errors, warnings, information, and advice issued during the current session.

To get the messages of specific products, you can click on the filter panel to open the product filter.

Filter panel

The filter panel lets you filter the message list and search for text in the message list. Refer to Filter Panel on page 50.

Message list

The message list provides the following information for each message:

Information	Description
Date[1)]	The date the message was issued.
Main Module Number[1)]	The main module that issued the message.
Message	The content of the message.
Message Code[1)]	The code of the message.
Message Number[1)]	The number of the message.
Module[1)]	The module that issued the message.
Process ID[1)]	The ID of the process in which the message was issued.
Product[1)]	The name of the product that issued the message.
Severity[1)]	The severity level of the message, indicated by one of the following symbols: ▪ errors ▪ warnings ▪ other messages, i.e., infos, advice, and questions
Submodule Number[1)]	The submodule that issued the message.
Thread ID[1)]	The ID of the thread in which the message was issued.
Time[1)]	The time the message was issued.

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

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

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

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

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

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

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

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

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

Refer to Filter Panel on page 50.

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

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

Related topics

References

Messages 54

Execute (User Function)

Access

You can access the user function under the name you assigned via:

Ribbon	Automation – User Functions
Context menu of	None
Shortcut key	None
Icon	 1), ...

1) You can replace the default icons by other images.

Purpose

To carry out any one of the user functions you added via the Customize command.

Result

The result of these user functions depends on the settings you created.

Description

The defined user functions are available via the Automation ribbon. The first nine user functions are also available via icons.

Related topics

Basics

Basics on User Functions....34

HowTos

How to Add External Programs as User Functions....35

References

Customize (User Functions)....44
User Functions Output....62

Edit Filter

Access

You can access this command via:

Ribbon	None
Context menu of	None
Shortcut key	None
Icon	None
Button	

Purpose

To specify a filter.

Result

If a filter is specified and active, only the entries are displayed that fulfills the filter criteria.

Description

When you click the button, the **Filter Editor** dialog is opened. In this dialog you can specify a filter. A filter can consist of several conditions that can be combined by a logical operation. In a condition, a column header is compared with a specified value. To specify a filter, the colored words can be modified.

Red word Click the red word to select a logical operator or a command for combining conditions. The following logical operator are available:

Logical Operation	Description
And	All combined conditions must be true.
Or	At least one of the combined condition must be true.

Logical Operation	Description
NotAnd	Not all of the combined conditions are true.
NotOr	Neither of the combined conditions are true.

The following commands are available:

Command	Description
Add Condition, , +, or Ins	Lets you add a condition.
Add Group	Lets you add a condition group.
Clear All	Lets you clear all conditions.
	Lets you clear a condition.
Remove group	Lets you remove a group.

Blue word Click the blue word to select a column of the table which value is compared with a specified value.

Green word Click the green word to select a relational operator.

For an instruction on how to specify and use a filter, refer to How to Specify and Use a Filter on page 31

Related topics

References

Clear Filter 41

Exit

Access

You can access this command via:

Ribbon	File
Context menu of	None
Shortcut key	Alt+F4
Icon	None

Purpose

To exit the current ModelDesk session.

Result

ModelDesk ends the current session.

Description

If you made any changes to any open files in ModelDesk, you are prompted to save them before exiting.

Export

Access

You can access this command via:

Ribbon	None
Context menu of	▪ **Project Navigator** – **Parameterset** – parameter set ▪ **Project Navigator** – **Road** – road file ▪ **Project Navigator** – **Scenario** – scenario file
Shortcut key	None
Icon	None

Purpose

To export a parameter set, road, scenario, or fellow type of the active experiment.

Result

ModelDesk exports the selected file to a ZIP archive.

Description

ModelDesk opens the **Save As** dialog that lets you specify a path and file name.

Related topics

HowTos

How to Export Parameter Sets (ModelDesk Parameterizing 🕮)

References

Export.....67
Import Parameter Set (ModelDesk Parameterizing 🕮)

Filter Panel

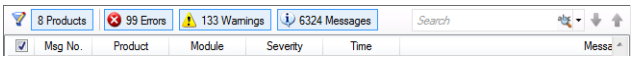
Access

- The filter panel is part of the **Message Viewer**. Refer to Messages on page 54.

  The following illustration shows the **Message Viewer** with the filter panel as an example:

- The filter panel is displayed when the dSPACE Log is opened in the working area. Refer to dSPACE Log on page 45.

The following illustration shows the filter panel as an example:



Purpose

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

Filter panel

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

Edit Product Filter (button - only in dSPACE Log) Lets you specify a list of products for filtering the message list.

To apply the product filter, click the Enable/Disable Product Filter button next to the button.

Enable/Disable Product Filter (Products button - only in dSPACE Log) Lets you enable/disable the product filter.

Show/Hide Errors (Errors button) Lets you display or hide errors.

Show/Hide Warnings (Warnings button) Lets you display or hide warnings.

Show/Hide other Messages (Messages button) Lets you display or hide other messages, i.e., infos, advice, and questions.

Search Lets you enter a text string for searching the message list.

You can use the following wildcards in the text string:

- **?** (wildcard for one character)
- ***** (wildcard for any number of characters)

To mask a wildcard, enter the \ character before the wildcard.

To select the next occurrence of the search string, click next to the edit field, or press the **Enter** key while the search field has the focus.

To select the previous occurrence of the search string, click next to the edit field.

Related topics

References

dSPACE Log....45
Messages....54

Options

Access

You can access this command via:

Ribbon	File
Context menu of	None
Shortcut key	None
Icon	None

Purpose

To view and change the global user settings of ModelDesk.

Result

Opens the **ModelDesk Options** dialog. If you change settings in any of the pages in the **ModelDesk Options** dialog, they apply to this and all subsequent ModelDesk sessions.

The **ModelDesk Options** dialog has the following pages:

- **Display Format Page** (ModelDesk Plotting 🕮)
- **Editor General Page** (ModelDesk Automation 🕮)
- **Instruments Page** (ModelDesk Plotting 🕮)
- **Interpreter Page** (ModelDesk Automation 🕮)
- **Platform Management Page** (ModelDesk Platform Management 🕮)
- **Project Page** (ModelDesk Project and Experiment Management 🕮)
- **Syntax Highlighting Page** (ModelDesk Automation 🕮)
- **Views Page** on page 64

Help

Access

To access this command via:

Ribbon	File
Context menu of	None
Shortcut key	None
Icon	None

Purpose

To get help.

Description

You have access to commands such as:

- dSPACE Help on page 42
- Using dSPACE Help on page 63
- New Features and Migration on page 56
- PDF Files on page 57
- Keyboard Help on page 53

Keyboard Help

Access

You can access this command via:

Ribbon	File – Help
Context menu of	None
Shortcut key	**Ctrl+F1**
Icon	

Purpose

To display the currently active shortcut keys.

Result

ModelDesk opens the Active Keyboard Shortcuts dialog that lists the currently active shortcut keys.

Active Keyboard Shortcuts dialog

Displays a context-dependent list of shortcut keys. They depend on your current working context in ModelDesk.

The shortcut keys displayed in the dialog depend on how the dialog is opened:

- Access via the Help ribbon

  If you access the list via the Help ribbon, ModelDesk displays a list with the shortcut keys that are globally available. They are always available, that is, independent of your current working context in ModelDesk.
- Access via **Ctrl+F1**

  If you access the list via **Ctrl+F1**, ModelDesk displays a context-dependent list of shortcut keys. They depend on your current working context in ModelDesk.

Copy List Lets you copy the list of keyboard shortcuts to the Clipboard. You can paste them, for example, to a text document and print them.

Messages

Access

You can access this command via:

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

Purpose

To show or hide the **Message Viewer**.

Description

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

Filter panel

The filter panel lets you filter the message list and search for text in the message list. Refer to Filter Panel on page 50.

Message list

The message list provides the following information for each message:

Information	Description
Date[1)]	The date the message was issued.
Main Module Number[1)] (Main Module)	The main module that issued the message.
Message	The content of the message.
Message Code[1)]	The code of the message.
Module[1)]	The module that issued the message.
Severity	The severity level of the message, indicated by one of the following symbols: ▪ Errors ▪ Warnings ▪ Other messages, i.e., infos, advice, and questions
Submodule Number[1)] (Submodule)	The submodule that issued the message.
Time[1)]	The time the message was issued.

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

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

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

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

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

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

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

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

The hidden subelements of the selected element are displayed.

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

The subelements of the selected element are hidden.

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

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

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

Refer to Filter Panel on page 50.

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

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

Related topics

References

dSPACE Log 45

New Features and Migration

Access

You can access this command via:

Ribbon	File – Help
Context menu of	None
Shortcut key	None
Icon	

Purpose

To display new features and required migration steps for all the products in the current dSPACE Release.

Result

dSPACE Help opens with New Features and Migration 🕮 displayed. Navigate to the specific product information to read about the new features of a specific product. If there are migration steps required, the necessary steps are described.

Open – File or Project

Access

You can access this command via:

Ribbon	File
Context menu of	None
Shortcut key	None
Icon	None

Purpose

To open a file or project.

Result

The standard Open dialog opens for you to select the path, folder, file type, and name of the file to be opened.

Tip

The last files that were opened can be seen in the File – Recent Files submenu. This list is updated each time you save or open another file.

Dialog settings

Look in Lets you select the path and folder to open the file from.

File name Displays the name of the selected file.

Files of type Lets you select the type of file to open.

Related topics

References

New (Script File) (ModelDesk Automation 🕮)
Save Project 59

Paste

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Parameter page ▪ Interpreter ▪ Source Code Editor
Shortcut key	`Ctrl+V`
Icon	None

Purpose

To paste the Clipboard contents into the current window.

Result

If the Clipboard contents are valid for the current window, they are pasted.

Related topics

References

Copy 43

PDF Files

Access

You can access this command via:

Ribbon	File – Help
Context menu of	None

Shortcut key	None
Icon	

Purpose

To open a folder containing documentation PDF files of the current dSPACE Release.

Save

Access

You can access this command via:

Ribbon	▪ Environment – File ▪ Environment – Road – ODIC File ▪ Automation – Python Scripts – Script File
Context menu of	None
Shortcut key	None
Icon	None

Purpose

To save the active document.

Result

The currently active document is saved.

Save As

Access

You can access this command via:

Ribbon	▪ Environment – File ▪ Environment – Road – ODIC File ▪ Automation – Python Scripts – Script File
Context menu of	None
Shortcut key	None
Icon	None

Purpose To save a document under a new name.

Result The currently active document is saved.

Save Project

Access You can access this command via:

Ribbon	File
Context menu of	Project Manager – project
Shortcut key	**Ctrl+Shift+S**
Icon	
Others	Quick Access Toolbar

Purpose To save the active project.

Result The currently active project is saved.

Related topics

References

Open – File or Project 56

Save Project As

Access You can access this command via:

Ribbon	File – Save As
Context menu of	None
Shortcut key	None
Icon	None

Purpose To save the selected project under a new file name.

Result

This allows you to insert the file name you desire. The project is saved under the file name and path you specified. The project name in the **Project Navigator** is changed accordingly.

Dialog settings

Enter a new name Lets you specify the new name of the project.

Select All

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Interpreter ▪ Source Code Editor
Shortcut key	**Ctrl+A**
Icon	None

Purpose

To mark all of the entries in the Interpreter or Source Code Editor.

Result

The entire text is marked.

Description

This lets you copy all the entire text from the Interpreter or Source Code Editor to the Clipboard at once.

Show Dependencies

Access

You can access this command via:

Ribbon	None
Context menu of	**Project Navigator** ▪ Pool – parameter file ▪ Road – road file ▪ Scenario – scenario file ▪ Objects – traffic object file
Shortcut key	None
Icon	None

Purpose To display the items which reference the selected file.

Result ModelDesk opens the **Project Manager** dialog.

Project Manager dialog Lists the items in the **Project Navigator** which reference the parameter, road, or scenario file hierarchically.

Start Page

Access You can access the command via:

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

Purpose To display the **Start** page.

Result The **Start** page is displayed in the working area.

Start page The **Start** page allows you to open an existing project or create a new one, for example. The **Start** page also provides quick access to the product documentation.

New ASM Project Lets you define a new project and experiment on the basis of a ModelDesk project that belongs to an ASM demo model. Refer to New ASM Project (ModelDesk Project and Experiment Management 🕮).

New Project + Experiment/New Experiment Lets you define a new project or a new experiment. Refer to New Project + Experiment/New Experiment (ModelDesk Project and Experiment Management 🕮).

Open Project + Experiment Lets you open an existing experiment. Refer to Open Project + Experiment (ModelDesk Project and Experiment Management 🕮).

Open File or Project To open a Python script in the Python Editor, or to open a ModelDesk project (CDP) file. Refer to Open Project + Experiment (ModelDesk Project and Experiment Management 🕮).

Recent Lets you open one of the most recent experiments that were open.

Reset Sort Direction (Available from the context menu of the column header of the list of experiments) You can click the column headers to sort the experiments in ascending or descending order according to a column. To remove this sorting you can reset the sort direction to this default, which a sorting based on the time the experiments were last opened, starting with the most recently opened experiment.

Open (Available from the context menu of an entry in the list of experiments) Lets you open the selected experiment.

Remove from List (Available from the context menu of an entry in the list of experiments) Lets you delete the selected item from the list of experiments.

Clear Recent Projects + Experiments List (Available from the context menu of an entry in the list of experiments) Lets you clear the list of experiments.

Close page after project has loaded Lets you specify whether to close the Start page after project load.

Show page on startup Lets you specify whether to display the Start page on program start.

Basic Practices/Advanced Practices/New Features and Migration Lets you display the ModelDesk documentation.

Related topics

HowTos

How to Start and Shut Down ModelDesk 13

User Functions Output

Access

You can display the User Function Output controlbar via:

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

Purpose

To show the User Functions Output controlbar.

Result

The User Functions Output controlbar is displayed.

Description

The User Functions Output controlbar provides access to the output of user function tools added to the user interface.

Related commands

The User Functions Output controlbar provides the following commands:

Copy To copy the entries in the user functions output viewer to the Clipboard.

Select All To select all the entries in the User Functions Output controlbar.

Clear All To clear all the entries in the User Functions Output controlbar.

Font To specify the font of the displayed output.

Related topics

References

Customize (User Functions)..... 44
Execute (User Function)..... 47

Using dSPACE Help

Access

You can access this command via:

Ribbon	File – Help
Context menu of	None
Shortcut key	None
Icon	

Purpose

To get information on working with dSPACE Help.

Result

dSPACE Help opens. It provides information on general handling and instructions on using dSPACE Help.

Views Page

Access

This page is part of the ModelDesk Options dialog.

Purpose

To alter the settings for workbook tabs and controlbar tabs.

Dialog settings

Workbook tabs position If you have opened several sequences and other windows, you have a better overview of them when you use the workbook mode. In this mode, each opened sequence and other windows are represented by a tab. With this setting, you can specify whether the tabs are displayed at the *top* or the *bottom* of the working area.

Controlbar tabs layout You can customize the window layout of ModelDesk. If you have docked several components to a window, it provides tabs to switch between them. For example, the Interpreter and the Message Viewer are combined in one window providing controlbar tabs for each other. With this setting, you can specify the layout of the controlbar tabs:

Layout	Description
Auto-size	Each controlbar tab contains the symbol and the name of the component that it represents. The whole component name is displayed. If the window is too small for displaying all controlbar tabs, you must scroll to the hidden ones.
Compressed	Only the active controlbar tab contains the symbol and name of the component that it represents. The other controlbar tabs are displayed only by their symbols.
Size to Fit	Each controlbar tab contains the symbol and the name of the component that it represents. However, the component names are shortened so that all controlbar tabs are accessible without scrolling.

Show shortcut keys in tooltips If you select this option, the tooltip of a selected command contains information on its shortcut key, if available.

Animate auto hiding of controlbars If you select this option, auto hiding of a window is done with an animation.

Drag controlbars to float Lets you specify whether a docked controlbar can be changed to the floating state by dragging the controlbar's title bar.

Related topics

References

Auto Hide....74
Options....52
Workbook Mode....87

Commands for Pool Management

Overview

ModelDesk's Pool Management provides the following commands, which are accessible via the context menu of the Project Navigator:

Where to go from here

Information in this section

Create Folder

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Project Navigator – Pool – Processing – Function ▪ Project Navigator – Pool – Processing – Setting
Shortcut key	None
Icon	None

Purpose

To create a new folder.

Result

A folder is created.

Description

You can create folders under the Pool - Processing - Function or Setting node of the project. These folders can have subfolders, so you can structure the elements hierarchically under these nodes.

Related topics

HowTos

How to Structure the Pool for Setting and Function Files (ModelDesk Processing 🕮)

References

Remove Folder 69
Rename Folder 70

Import

Access

You can access this command via:

Ribbon	None
Context menu of	Project Navigator – Pool
Shortcut key	None
Icon	None

Purpose

To import archived ModelDesk files to the Pool.

Description

You can import:

- Parameter files
- Road files
- Scenario files
- Traffic driver files
- Traffic object files
- Configuration files
- Layout configuration files
- Measurement files
- Trigger definition files
- Measurement type files
- Measurement data files
- Conversion file

- Function and setting file
- OpenDRIVE files

ModelDesk lets you import XML files, individually or grouped in a ZIP archive. ModelDesk imports the archived files to the Pool and arranges them according to the model structure automatically. If you import files to the Pool for the first time, ModelDesk opens the **Parameter file import** dialog for you to choose how ModelDesk should handle the import.

If a layout file is imported, ModelDesk tries to import the corresponding connection file (CDC file) from the same folder to get information on the data connection between real-time application and plotters of the layout. Note that layouts without data connections cannot be used in ModelDesk.

Parameter file import dialog

Shows the file name and lets you choose what would you like to do with it.

Rename Lets you rename the file.

Don't import The currently displayed file is not imported, the dialog displays the next file or closes the dialog if no other file exists.

Replace existing Replaces the existing file by the currently displayed file. The dialog displays the next file or closes the dialog if no other file exists.

Never replace existing Lets you specify that ModelDesk never replaces existing files.

Always replace existing Lets you specify that ModelDesk always replaces existing files.

Related topics

HowTos

How to Import Parameter Files (ModelDesk Parameterizing 🕮)

References

Export 67

Export

Access

You can access this command via:

Ribbon	None
Context menu of	▪ **Project Navigator** – Pool ▪ **Project Navigator** – Pool - all the Pool nodes

Shortcut key	None
Icon	None

Purpose

To export ModelDesk files stored in the Pool.

Description

You can export:

- Parameter files
- Road model files
- Scenario files
- Traffic driver files
- Traffic object files
- Configuration files
- Measurement files
- Trigger definition files
- Measurement type files
- Measurement data files
- Conversion file
- Function and setting file

ModelDesk exports the selected files to a ZIP archive.

Note that it is not possible to export an individual layout. To export layouts, you must export a superordinate node, for example, the Layouts node.

Related topics

HowTos

How to Export Parameter Files (ModelDesk Parameterizing 🕮)

References

Import.....66

Remove (Pool)

Access

You can access the command via:

Ribbon	None
Context menu of	**Project Navigator – Pool** – <element>

Shortcut key	None
Icon	None

Purpose

To remove the selected element.

Description

You can only remove elements that are not linked to an experiment.

Result

ModelDesk checks whether the element is linked to the experiment. If it is not linked, the element is removed.

Related topics

Basics

Working with Parameter Sets (ModelDesk Parameterizing 🕮)

Remove Folder

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Project Navigator – Pool – Processing – Function – created folder ▪ Project Navigator – Pool – Processing – Setting – created folder
Shortcut key	None
Icon	None

Purpose

To remove a created folder.

Result

The folder and its content are removed.

Related topics

HowTos

How to Structure the Pool for Setting and Function Files (ModelDesk Processing 🕮)

References

Create Folder 65
Rename Folder 70

Rename Folder

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Project Navigator – Pool – Processing – Function – created folder ▪ Project Navigator – Pool – Processing – Setting – created folder
Shortcut key	None
Icon	None

Purpose

To rename a created folder.

Result

The folder is renamed.

Description

You cannot rename standard folders.

Related topics

HowTos

How to Structure the Pool for Setting and Function Files (ModelDesk Processing 🕮)

References

Create Folder 65
Remove Folder 69

Commands for Window Handling

Introduction

The ModelDesk user interface consists of several components. The display and the behavior of the components' windows can be customized by the following commands:

Where to go from here

Information in this section

Add to Quick Access Toolbar

Access

You can access this command via:

Ribbon	None
Context menu of	Command in a ribbon
Shortcut key	None
Icon	None

Purpose

To add the selected command to the Quick Access Toolbar.

Result

The command is added to the Quick Access Toolbar.

Related topics

References

Customize Quick Access Toolbar / More commands.....77

Allow Docking

Access

You can access this command via:

Ribbon	None
Context menu of	All controlbars
Shortcut key	None
Icon	None

Purpose

To allow the component to be docked.

Result

When moved within ModelDesk's main window, the component is docked if ModelDesk finds a suitable place for it.

Description

When docked, the component has no window frame but an information header for controlling its state, position, and size.

Related topics

References

Float in Main Window.....81

Arrange Icons

Access

You can access this command via:

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

Purpose

To arrange all minimized windows in ModelDesk's working area.

Result

All minimized windows are arranged horizontally at the bottom of ModelDesk's working area.

Related topics

References

Auto Hide

Access

You can access this command via:

Ribbon	None
Context menu of	Pane's title bar in docked mode
Shortcut key	None
Icon	None
Others	 in the pane's title bar

Purpose

To hide the window automatically if it is not focused.

Description

The **Auto Hide** command is enabled only for windows in docked mode. If you enable the Auto Hide feature, a tab representing the window is created in the user interface. You can click the tab to display the window again. The position of the tab depends on the original position of the window you want to hide. The Auto Hide symbol in the title bar changes to  .

The transition between displayed and hidden window is animated if the **Animate auto hiding of controlbars** option is set. Refer to Views Page on page 64.

Cascade

Access

You can access this command via:

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

Purpose

To cascade all currently open windows in ModelDesk's working area.

Result

The windows are cascaded.

Description

The windows are cascaded diagonally, one on top of the other. You can move them and pull them to any size you desire.

Related topics

References

Close All

Access

You can access this command via:

Ribbon	View – Window
Context menu of	None
Shortcut key	**Ctrl** + **Shift** + **F4**
Icon	

Purpose

To close all windows currently open in ModelDesk.

Result

All unchanged windows are closed.

Description

If you made changes in any of your windows, you are prompted to save each window before it is closed. If you press **Yes**, the windows are saved and closed afterwards. If you press **No**, they are closed without being saved. If you press **Cancel**, only unchanged windows are closed.

Related topics

References

Close All But This....76
Close Window....76

Close All But This

Access

You can access this command via:

Ribbon	None
Context menu of	Workbook tab in workbook mode
Shortcut key	None
Icon	None

Purpose

To close all the windows except for the currently active one.

Result

All the windows other than the active one are closed.

Related topics

References

Close All....75
Close Window....76

Close Window

Access

You can access this command via:

Ribbon	**View – Window**
Context menu of	▪ Window's title bar ▪ Workbook tab in workbook mode

Shortcut key	**Ctrl+F4**
Icon	
Others	in the window's title bar

Purpose

To close the currently active window in the working area.

Result

The currently active window in the working area is closed.

Related topics

References

Close All 75
Close All But This 76

Customize Quick Access Toolbar / More commands

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Commands in the ribbon ▪ Commands in the Quick Access Toolbar
Shortcut key	None
Icon	None

Purpose

To customize the Quick Access Toolbar.

Result

The Quick Access Toolbar is customized according to your settings in the Quick Access Toolbar page.

Description

ModelDesk opens the Quick Access Toolbar page of the Customize dialog, which allows you to customize the Quick Access Toolbar.

Quick Access Toolbar page

To customize the Quick Access Toolbar.

Choose commands from: Lists all the available ribbons. Lets you select a ribbon to display its commands.

Add >> Lets you add the selected command to the Quick Access Toolbar.

<< Remove Lets you remove the selected command from the Quick Access Toolbar.

Reset to Factory Settings Lets you reset the Quick Access Toolbar to the default.

Show Quick Access Toolbar below the ribbon Lets you specify whether to display the Quick Access Toolbar above or below the ribbon.

Related topics

HowTos

How to Customize the Quick Access Toolbar 28

References

Reset to Default 82

Customize Ribbon

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Commands in the ribbon ▪ Commands in the Quick Access Toolbar
Shortcut key	None
Icon	None

Purpose

To customize the ribbon.

Result

The ribbon is customized according to your settings in the Customize Ribbon page.

Description

The Customize Ribbon page of the Customize dialog opens, which allows you to customize the ribbon.

Customize Ribbon page

To customize the Quick Access Toolbar.

Choose commands from: Lists all the available ribbons. Lets you select a ribbon to display its commands.

Add >> Lets you add the selected command to the Quick Access Toolbar.

Remove Lets you remove the selected command from the ribbon.

New Tab Lets you add a new ribbon tab.

New Group Lets you add a new ribbon group.

Rename (Only for tabs and groups added by the user) Lets you rename a ribbon tab or ribbon group.

Reset to Factory Settings Lets you reset the ribbon to the default.

Initially selected tab Lets you specify which ribbon tab is initially open.

Import Lets you import settings of the ribbon.

Export Lets you export the settings of the ribbon.

Related topics

Basics

Basics on Ribbons.....26

References

Reset to Default.....82

Dock/Docked

Access

You can access this command via:

Ribbon	View – Window
Context menu of	▪ Window's title bar in floating mode ▪ Workbook tab in workbook mode
Shortcut key	None
Icon	

Purpose

To place a floating pane at a certain location in the user interface and connect it to its neighbor windows.

Description

The Dock command on the View ribbon is only available for a component that is floating in the main window. If you dock the component, it is moved to a standard location in the user interface and docked to its neighbor windows.

The **Docked** command from the title bar's context menu is available for a component that is in floating state.

If you drag a window and drop it on the selected positioning symbol, it is automatically docked.

Note

- Before you can dock a window, you must select the **Allow Docking** option to enable docking at all.
- This command is only available for windows that support docking.

Float/Floating

Access

You can access this command via:

Ribbon	**View – Window**
Context menu of	▪ Window's title bar in docked mode ▪ Workbook tab in workbook mode
Shortcut key	None
Icon	

Purpose

To make a window movable on the entire screen.

Description

The **Float** command on the **View** ribbon is only available for a component that is floating in the main window. If you float the component, the window type changes, and you can drag it to any place on your screen, even outside of the user interface. The window is not docked to other windows.

The **Floating** command from the title bar's context menu is available for a component that is in docked state.

If you drag a docked window but do not drop it on a positioning symbol, it is automatically in floating state.

Note

- This command is available only for windows that support floating.
- The **Float in Main Window** command makes a window floating, but you can move it only within the main window.
- You cannot float a window in **full screen mode**.

Float in Main Window

Access

You can access this command via:

Ribbon	None
Context menu of	All controlbars
Shortcut key	None
Icon	None

Purpose

To move the controlbar to the main window.

Result

The controlbar is moved to the main window.

Related topics

References

Allow Docking 73
Dock/Docked 79
Float/Floating 80

Full Screen Mode

Access

You can access this command via:

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

Purpose

To maximize all windows currently open and put the active window on top.

Result

All windows are enlarged to their maximum. The window currently active is put on top.

Description

You can return to the normal display mode by clicking the Fullscreen toolbar button or pressing **F11** on the keyboard.

Minimize the Ribbon

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Commands in the ribbon ▪ Commands in the Quick Access Toolbar
Shortcut key	None
Icon	(on the right of the ribbon)  or 

Purpose

To toggle between a minimized and expanded ribbon.

Result

If the ribbon is minimized, only the ribbon tabs are shown on the user interface.

Description

To access a command when the ribbon is minimized, click a ribbon tab. The commands remain on the user interface until you execute a command from the ribbon.

Reset to Default

Access

You can access this command via:

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

Purpose

To set the arrangement of the user interface to the original settings as and when ModelDesk was first installed.

Result

The settings of the user interface are reset to the default.

Description

The controlbars of ModelDesk, such as Project Manager, Message Viewer, and all other controlbars and toolbars, can be arranged according to your needs. For example, you can move a controlbar and dock it to another controlbar. All

your modifications are saved when you exit ModelDesk. You can use this command to reset the user interface to its default.

Related topics

References

Customize Quick Access Toolbar / More commands....77
Customize Ribbon....78

Reset Layout

Access

You can access this command via:

Ribbon	View – Road Views
Context menu of	None
Shortcut key	None
Icon	

Purpose

To set the user interface layout to the original settings of the Road Generator when ModelDesk was first installed.

Result

The settings of the user interface are reset to the default.

Description

You can use this command to reset the Road Generator's layout if you have rearranged the panes. Otherwise, all your modifications are saved when you exit.

Related topics

References

Reset to Default....82

Show Quick Access Toolbar Above / Below the Ribbon

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Commands in the ribbon ▪ Commands in the Quick Access Toolbar
Shortcut key	None
Icon	None

Purpose

To show the Quick Access Toolbar above or below the ribbon.

Result

The Quick Access Toolbar is shown above or below the ribbon.

Related topics

Basics

Basics on Ribbons.....26

Status Bar

Access

You can access this command via:

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

Purpose

To show or hide ModelDesk's status bar at the bottom of the main window.

Result

The status bar is shown or hidden.

Switch Controlbars

Access

You can access this command via:

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

Purpose

To show or hide controlbars.

Result

Controlbars are either shown or hidden.

Description

Opens a submenu showing the following controlbars available in ModelDesk:

- Alias Overview
- Interpreter
- Messages
- Platforms
- Project
- Properties
- Signal Selector
- Trigger
- User Functions Output

Tip

To hide a controlbar, you can also use the Close symbol in the controlbar's header.

Switch Windows

Access

You can access this command via:

Ribbon	View – Window
Context menu of	None
Shortcut key	None

Icon	

Purpose

To activate another open window.

Result

The selected window is opened in the working area.

Tile Horizontally

Access

You can access this command via:

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

Purpose

To arrange the currently open windows in ModelDesk horizontally.

Result

The windows are arranged horizontally.

Description

The windows are arranged so that each window has the same size, initially without overlaps. You can move the child windows, pull them to any size you desire, and overlap them.

Related topics

References

Arrange Icons 73
Cascade 75
Tile Vertically 87
Workbook Mode 87

Tile Vertically

Access

You can access this command via:

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

Purpose

To arrange the currently open windows in ModelDesk vertically.

Result

The windows are arranged vertically.

Description

The windows are arranged so that each window has the same size, initially without overlaps. You can move the child windows, pull them to any size you desire, and overlap them.

Related topics

References

Arrange Icons....73
Cascade....75
Tile Horizontally....86

Workbook Mode

Access

You can access this command via:

Ribbon	View – Show
Context menu of	None
Shortcut key	None
Icon	☑ (Checkbox)

Purpose

To enable or disable the Workbook mode for ModelDesk's working area.

Result

If there is a checkmark to the left, the Workbook mode is enabled in ModelDesk's working area.

Description

In the Workbook mode, each separate window in the working area can be activated by a tab at the bottom or the top of the working area. The position of the workbook tabs can be specified in the **ModelDesk Options** dialog. You can sort the tabs by dragging a tab to the desired position. A tab provides a specific context menu.

If there is no checkmark, the Workbook mode is deactivated.

Related topics

References

Arrange Icons....73
Cascade....75
Tile Horizontally....86
Tile Vertically....87
Views Page....64

Automation

Where to go from here

Information in this section

Introduction to the Message Reader API

Where to go from here

Information in this section

Reading dSPACE Log Messages via the Message Reader API

Introduction

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

dSPACE Log

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

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

Note

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

Message Reader API

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

The Message Reader API is available as of dSPACE Release 2020-A. For information on the dSPACE products and components that support the Message Reader API, refer to Supported dSPACE Products and Components on page 92.

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

Supported dSPACE Releases

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

Message Reader API change in dSPACE Release 2021-A

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

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

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

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

Using Message Reader API of ...	
... dSPACE Release 2020-B and Earlier (Python 3.6)	**... dSPACE Release 2021-A and Later (Python 3.9)**

`# Specify products whose messages to read:`
`Settings = MessageReaderSettings()`
`Settings.Products.Add('ControlDesk')`
`Settings.Products.Add('AutomationDesk')`
	`# Specify products whose messages to read:`
	`Settings = MessageReaderSettings()`
	`Settings.SetProducts(['ControlDesk', 'AutomationDesk'])`

Related topics

Basics

Supported dSPACE Products and Components 92

Examples

Example of Reading Messages with C# 94
Example of Reading Messages with Python 92

References

MessageReaderSettings Class 101

Supported dSPACE Products and Components

Supported dSPACE products and components

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

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

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API..... 90

Example of Reading Messages with Python

Introduction

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

Referencing a message reader assembly

You have to reference a `dSPACE.Common.MessageReader.dll` assembly. For information on the location of the assembly, refer to dSPACE.Common.MessageReader.dll on page 91.

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

The following code references and imports the message reader assembly.

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

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

Reading all messages

The following example reads all existing message log files and prints all messages via Python. It is assumed that the message reader assembly is referenced and imported. Refer to Referencing a message reader assembly on page 92.

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

Filtering messages by severity, product, and session

The following example reads and prints messages with a severity of `Error`, `SevereError`, or `SystemError`. Also, only messages of the last sessions of ControlDesk and AutomationDesk are read and printed. It is assumed that the message reader assembly is referenced and imported. Refer to Referencing a message reader assembly on page 92.

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

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

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

Note

The ReadMessages method returns an enumerator which must either read all messages or must be disposed when no longer used. It is not possible to use two enumerators interleaved, only one enumerator may read messages at a time. Refer to MessageReader Class on page 100.

Filtering messages by time

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

The following example reads all messages after a certain start time. It is assumed that the message reader assembly is referenced and imported. Refer to Referencing a message reader assembly on page 92.

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

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

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API 90
Supported dSPACE Products and Components 92

References

MessageReaderSettings Class 101

Example of Reading Messages with C#

Introduction

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

Referencing a message reader assembly

You have to reference a `dSPACE.Common.MessageReader.dll` assembly. For information on the location of the assembly, refer to dSPACE.Common.MessageReader.dll on page 91.

Reading all messages

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

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

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

Filtering messages by severity, product, and session

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

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

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

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

Note

The ReadMessages method returns an enumerator which must either read all messages or must be disposed when no longer used. It is not possible to use two enumerators interleaved, only one enumerator may read messages at a time. Refer to MessageReader Class on page 100.

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API 90

References

dSPACE.Common.MessageHandler.Logging Reference

Where to go from here

Information in this section

ILogMessage Interface

Namespace

`dSPACE.Common.MessageHandler.Logging`

Description

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

Properties

The element has the following properties:

Name	Description	Get/Set	Type
IsStartMessage	Gets a value indicating whether the message is a session start message.	Get	*Boolean*
IsStopMessage	Gets a value indicating whether the message is a session stop message.	Get	*Boolean*
MainModuleNumber	Gets the main module number of the message.	Get	*Integer*
MessageCode	Gets the error code of the message.	Get	*Integer*
MessageText	Gets the text of the message.	Get	*String*
ModuleName	Gets the module name of the message.	Get	*String*
Session	Gets the log session which issued the message.	Get	ILogSession (refer to ILogSession Interface on page 98)
Severity	Gets the severity of the message.	Get	Severity (refer to Severity Enumeration on page 103)

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

Methods

The element has no methods.

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API..... 90

Examples

Example of Reading Messages with C#..... 94
Example of Reading Messages with Python..... 92

References

ILogSession Interface..... 98
Severity Enumeration..... 103

ILogSession Interface

Namespace

`dSPACE.Common.MessageHandler.Logging`

Description

To access information about a message log session.

Properties

The element has the following properties:

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

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

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ToSessionTime	Converts UTC time to time zone used when the session was written.	▪ <*DateTime*> `utcTime`: Specifies the UTC time to convert.	Time in the time zone of the logging session. ▪ DateTime

1) <Type> `Name`: Description

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API....90

Examples

Example of Reading Messages with C#....94
Example of Reading Messages with Python....92

MessageReader Class

Description

To read serialized messages written by dSPACE products.

Constructor

The element has the following constructor:

Name	Description	Parameter[1)]	Returns
MessageReader	Initializes a new instance of the MessageReader class.	▪ <MessageReaderSettings>[2)] `settings`: Settings which allow to specify which sessions and messages are read. Can be null, causing all existing log files to be read.	None

1) <Type> `Name`: Description
2) Refer to MessageReaderSettings Class on page 101

Properties

The element has no properties.

Methods

The element has the following methods:

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

Name	Description	Parameter[1)]	Returns
ReadMessages	Reads the messages written to the log files of the sessions up to now. The messages are returned in chronological order according to their time stamps. **Note** The ReadMessages method returns an enumerator which must either read all messages or must be disposed when no longer used. It is not possible to use two enumerators interleaved, only one enumerator may read messages at a time.	None	Messages read from log file. ▪ IEnumerable< ILogMessage (refer to ILogMessage Interface on page 97) >

1) <Type> `Name`: Description

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API 90

Examples

Example of Reading Messages with C# 94
Example of Reading Messages with Python 92

References

MessageReaderSettings Class 101

MessageReaderSettings Class

Description

To define the settings of a message reader.

Used to filter the log sessions and messages read.

Constructor The element has the following constructor:

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

1) <Type> `Name`: Description

Properties The element has the following properties:

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

Methods

The element has the following methods:

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

1) <Type> `Name`: Description

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API....90

Examples

Example of Reading Messages with C#....94
Example of Reading Messages with Python....92

Severity Enumeration

Description

To specify the severity of a message.

Enumeration values

The enumeration has the following values:

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

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API....90

Examples

Example of Reading Messages with C#....94
Example of Reading Messages with Python....92