MotionDesk

Basics

For MotionDesk 4.8

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

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

© 2001 - 2021 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

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

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

Contents

Reference Information 47

About This Document

Contents

This document introduces you to MotionDesk. It provides basic information of MotionDesk 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 MotionDesk

Where to go from here

Information in this section

Features of MotionDesk

Introduction

In a simulation, it is difficult to infer the movement of a mechanical system, such as a car, simply from plots displaying forces or angles. Especially when the user interacts with the simulation, as in a driving simulator, it is extremely important to give him/her a more realistic feedback.

There are three mechanisms to enhance the realism of a simulation:

- Give the user realistic visual feedback, so that he/she can see what is going on.
- Give the user realistic acoustic feedback, like what he/she would hear in similar situations.
- Give the user a perception of acceleration in driving and flight simulations, so that he/she cannot only see and hear, but also feel, that the vehicle is moving.

  If you want to reach this highest level of realism, you need a motion platform.

The following describes how you can use MotionDesk to create a realistic visual feedback of your simulation.

MotionDesk has been developed as a complement to the dSPACE tool chain to visualize the movement of mechanical objects in the 3-D world. MotionDesk can visualize mechanical parts like vehicles or robotic arms that move in a virtual world and are simulated with dSPACE simulators. If your PC is fast enough, the latency time between simulation and visualization is low, which makes the system capable of "man-in-the-loop" simulation. For a presentation of simulation results, the simulation data can be recorded and replayed later.

Creating a virtual world is simple. A wide range of 3-D objects for vehicle simulation is available in a 3-D object library. Library expansion is easy as the 3-D object geometries are described in the COLLADA standard. The objects are assembled in the **3-D View** to create a scene. You drag them from the **Library Browser** into the **3-D View** and move them to the correct positions with the mouse or via the properties pane. The scene looks more realistic through the use of modern rendering techniques such as texture mapping and shaders.

Once created, the 3-D scene comes to life in MotionDesk in the 3-D view. MotionDesk gets the motion data from a dSPACE simulator and moves the movable objects in accordance with the data. To get the right view of the scene, observers can be defined with varied behaviors, for example, static in the scene or following a moving object.

You can also incorporate sensors, for example, camera, laser, lidar, and radar sensors into a sensor simulation architecture. You can add and configure the sensors in MotionDesk and download the scene with the sensor data to an application that prepares and renders the sensor data.

A material database is also available to configure the materials for the 3-D objects and assign them to the pixels of the 3-D object images. This information is important for sensor simulation.

MotionDesk is not restricted to automobiles. Other typical applications for 3-D animation are robotics and flight simulation.

Note

As of MotionDesk 4.8 in Release 20-B, MotionDesk is installed separately in the `dSPACE MotionDesk <release>` folder. It is not installed in the `dSPACE RCP and HIL <release>` folder.

MotionDesk Blockset supported platforms

MotionDesk and the **MotionDesk Blockset** support the following platforms:

- Simulink
- SCALEXIO Systems
- MicroAutoBox III
- VEOS and VEOS on Linux

RTI platforms

- DS1006
- DS1007
- MicroLabBox (DS1202)
- MicroAutoBox II (DS1401)

Note

To work with Sensor Simulation, you must use the Model and Sensor Interface Blockset. For more information, refer to Basics of the Model and Sensor Interface Blockset (Model and Sensor Interface Blockset Manual 🕮).

MotionDesk Licenses

Introduction

MotionDesk consists of several modules, each protected by its own licenses, so you can order only the modules which you need for your work. If you need some extra MotionDesk functionalities, check if the appropriate license is available.

Licenses

The following table shows the licenses and the modules which are protected.

License	Protected Modules
MotionDesk	Basic module for MotionDesk The module is required for each MotionDesk version. It allows you to perform the following actions: ▪ Handling projects and experiments ▪ Creating and editing a scene ▪ Visualizing a scene using motion data coming from all supported simulation platforms
MotionDesk Animated Characters	The license allows you to use animated characters (humans and animals) in the scene.
MotionDesk Sensors	To work with sensors in Sensor Simulation, you must activate the sensor licenses for each sensor type used. When you activate the license, you can download a scene to the SensorSim application and display the sensor image data for the simulation in the sensor composition window. Sensor licenses for Sensor Simulation ▪ MDD_SENSOR_CAMERA: To use camera or fish-eye sensors ▪ MDD_SENSOR_LIDAR: To use laser or lidar sensors ▪ MDD_SENSOR_RADAR: To use radar sensors For more information, refer to Overview of Sensor Simulation (Sensor Simulation Overview 🕮)

The dSPACE objects library can be installed and used without a license.

How to Start and Shut Down MotionDesk

Objective

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

Part 1

To start MotionDesk

1 In the Start menu, select dSPACE MotionDesk <release> – dSPACE MotionDesk <release>.

Tip

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

Result

MotionDesk opens.

Part 2

To shut down MotionDesk

1 On the File ribbon, click Exit.

Result

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

Related topics

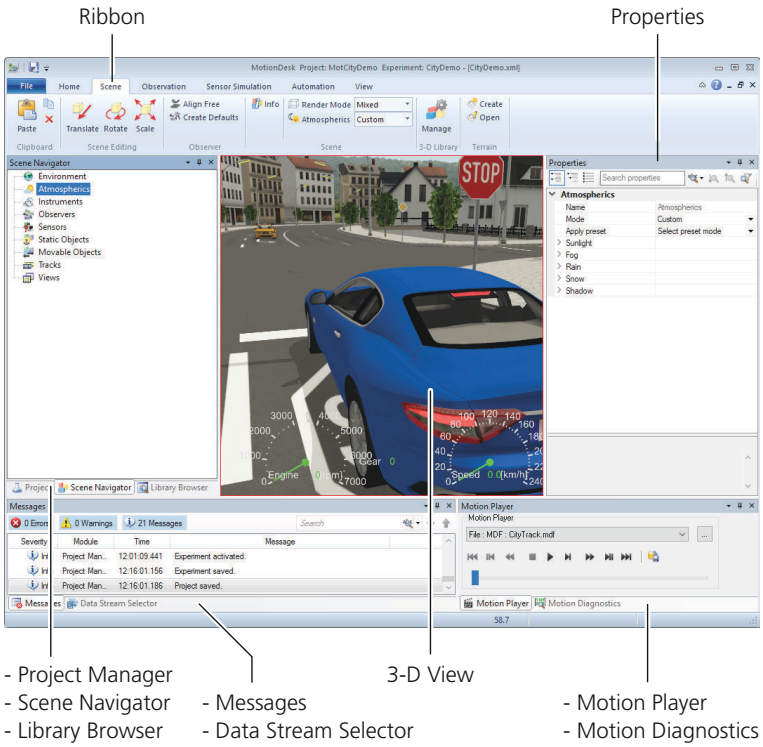
References

Create Shortcut (MotionDesk Project and Experiment Management 🕮)
Exit 54

User Interface of MotionDesk

Graphical overview

The main window offers the basic environment for all other elements and tools provided by MotionDesk.



You can switch each element on or off individually.

Ribbon

The ribbon provides access to MotionDesk's functions and commands.

Overview Most of MotionDesk's components provide commands in the ribbon for quick access to frequently used commands, see the following table shows.

Ribbon	Ribbon Group	Description
File	–	To work with MotionDesk projects and experiments.
Home	Simulation	To start and stop the simulation.
	Motion Player	To control a replay of the animation.
	Save As	To create an MDF file or video of an animated scene.
	Views	To set up the 3-D View.
	3-D Library	To open the Library Manager for managing customer objects.
	Materials	To open the Material Database Editor to add custom materials and extended properties for use with 3-D objects.
	Multi-PC	To configure and start scene synchronization manually.
	Platform	To select the data source that provides the motion data.
Scene	Clipboard	To copy, paste, or remove objects in the scene.
	Scene Editing	To modify the position, orientation or size of an object.
	Observer	To work with the observers.
	Scene	To specify the render mode and atmospherics.
	3-D Library	To open the Library Manager for managing customer objects.
	Terrain	To generate a terrain 3-D object.
Observation	Clipboard	To access standard commands, for example, copy, cut or paste for creating instruments.
	Create Instrument	To create instruments.
	Layer Instrument	To specify the layer of instruments.
	Views	To manage the 3-D view.
	Observer	To work with the observers.
Sensor Simulation	Connections	To work with the connections to the SensorSim application instances.
	Sensors	To work with sensors in the scene.
	Scene	To download the scene to the running SensorSim application instances.
Automation	User Function	To configure and start user functions.
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 MotionDesk's windows.
	Full Screen	To enable full screen mode or stereo full screen.

Basics For general information of the ribbon, refer to Basics on Ribbons on page 19.

Controlbars

MotionDesk contains several controlbars.

Data Stream Selector Shows a list of all data streams which can be assigned to movable objects and instruments.

Library Browser Lists all 3-D objects which you can use for creating a scene.

Messages Shows all messages provided by MotionDesk.

Motion Diagnostics Displays information on rotation and angle values for a selected object.

Motion Player Contains the elements for controlling replay.

Project Manager Provides a specific view on the project and experiments.

Properties Lists all the properties of an item select on the Scene Navigator.

Scene Navigator Provides a specific view on the scene components.

User Function Output Provides access to the output of external tools added to the Tools menu and the User Functions toolbar.

Tip

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

Status bar

The Status bar displays explanatory texts for the command icons of the toolbar or menu items, and the frame rate.

3-D View

The 3-D View displays the MotionDesk scene in a window. It can be divided into up to 4 subwindows and each sub-window can be configured to display the scene from a different perspective. The views are selected in the ribbon or in the Scene Navigator.

You can also enable a two-dimensional birds's-eye view of the scene from above using the properties pane.

The 3-D View can also be displayed in full screen mode on one or more screens. The selected view is shown in the full screen.

For more information, refer to 3-D View Properties (MotionDesk Scene Animation 🕮).

Related topics

Basics

Customizing the User Interface 17

Customizing the User Interface

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** controlbar
- 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 View** 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

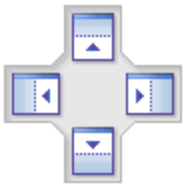
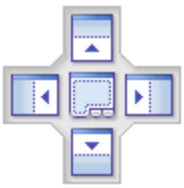
To customize the screen arrangement

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.
	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....64
Auto Hide....65
Dock/Docked....70
Float in Main Window....72
Float/Floating....71

Basics on Ribbons

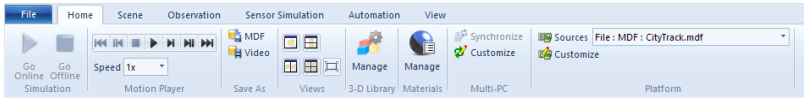
Introduction

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.

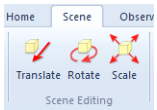
The ribbon consists of several ribbon tabs, see the following example of the Home ribbon.



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 Scene Editing ribbon group in MotionDesk as an example.

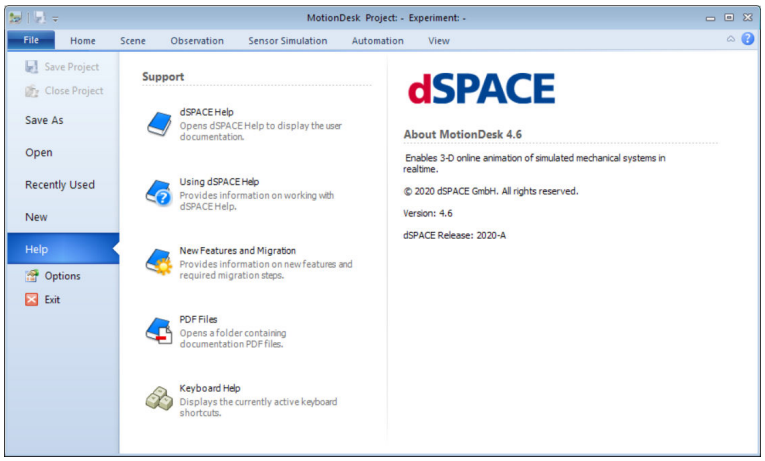


Dialog launcher A dialog launcher is an optional element of a ribbon group that lets you open a dialog related to that ribbon group.

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 21.

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.....69

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..... 19

References

Customize Quick Access Toolbar / More commands..... 68

Basics on User Functions

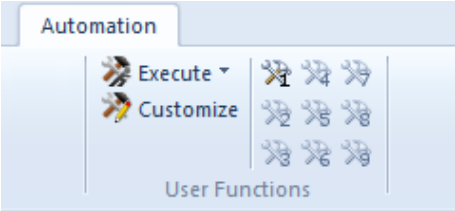
Introduction

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

User functions

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

Related topics

HowTos

How to Add External Programs as User Functions 23

References

Customize (User Functions) 50

How to Add External Programs as User Functions

Objective

You can add shortcuts to external applications to MotionDesk.

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 22

References

Customize (User Functions) 50

General Information on 3-D Graphics

Introduction

The following topics give some basic information on 3-D visualization. If you are familiar with 3-D graphics, you can skip the following topics.

Where to go from here

Information in this section

Information in other sections

Basics of 3-D Graphics

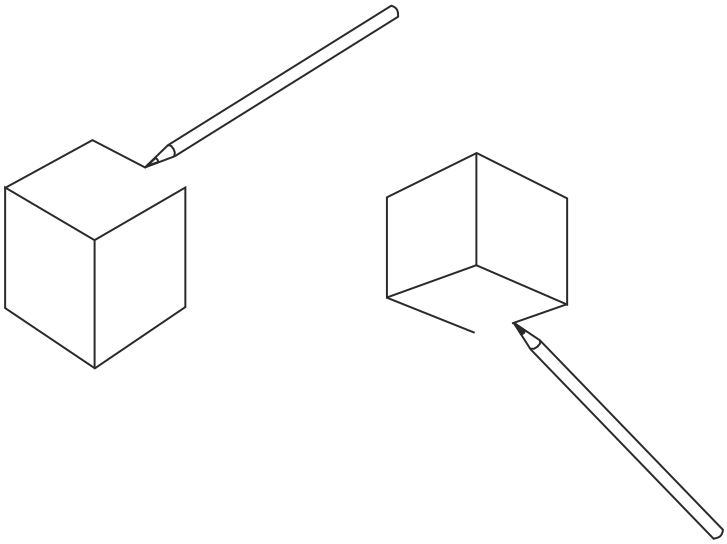
Where to go from here

Information in this section

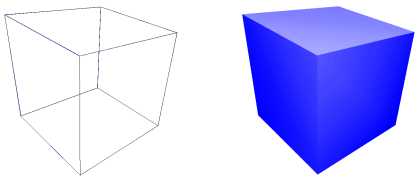
2-D Versus 3-D Graphics

Differences

The main difference between 2-D and 3-D graphics is depth. 2-D graphics can be drawn so that they look like they are 3-D, but if you want to change the perspective or viewpoint in any way, you have to redraw the object from scratch, see the following illustration.



A 3-D graphic has to be constructed. Every 3-D object consists of several polygons defined by their coordinate values (x, y, z points). To get a surface for an object, you have to define the reference points (P1, P2, P3) for the polygons. The polygons have a color and can be given a texture (bitmap), see the following illustration.

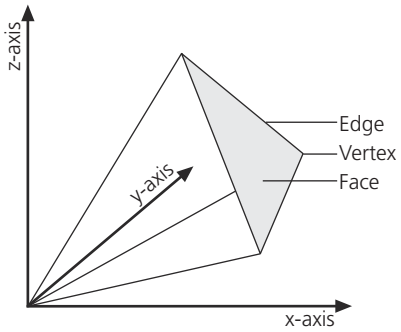


Once the 3-D objects are created, you can view them from different angles and distances. The graphics are redrawn by the computer according to the viewpoint from which each object is seen. Light effects are also included. You can see brighter or darker sides of the objects, depending on the position of the lights.

Basic Elements Vertex, Edge, and Face

Three-edged polygon

The simplest basic component of a 3-D object is a three-edged polygon (triangle), see the following illustration:



A vertex is a point where any number of lines come together and connect to each other. Each of the lines forms a boundary, or in other words, an edge. Several lines connected together are a polyline. If a polyline is closed (start and end points are the same), it is called a polygon. The area enclosed by the edges is called a face.

The illustration above shows a three-edged polygon with three vertices. Four-sided polygons (quads) are also used as basic components. However, there is no limit to the number of sides a polygon can have.

The coordinates of the vertices define the position of each face component. The color (or a texture) can be defined for the face. By placing several polygons edge to edge, you get a 3-D object.

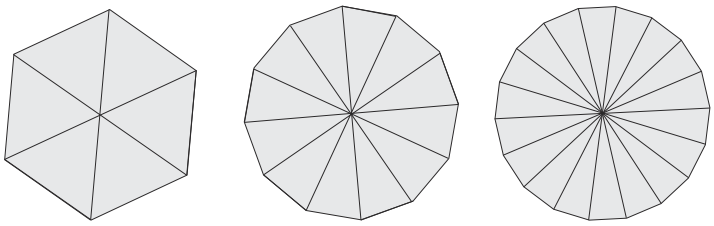
3-D Objects

Introduction

3-D objects are the basic components of a virtual world. They describe the shape (geometry) of mechanical bodies.

Creating 3-D objects

The 3-D objects are composed of polygons, for example, triangles. The quality of a 3-D object depends on the number of polygons used to build it, see the following illustration:

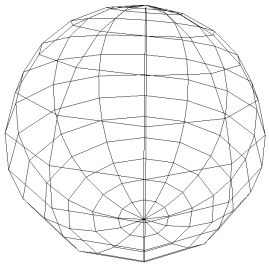


In the illustration, the number of polygons increases from left to right. You can see that the more polygons are used, the smoother the objects look. However, if you use more polygons, the required computation time will also increase. The left object consists of only 6 polygons, whereas the right object consists of 18.

Displaying 3-D objects

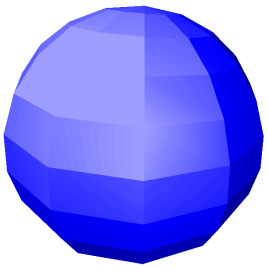
There are different modes for displaying 3-D objects.

Wire frame mode Wire frame mode draws the object using lines to represent the edges of the polygons, which makes it resemble a sculpture made of wire mesh. You can see the inner form of the object and have access to individual vertices. Because every edge is visible (even the ones on the far side of the object), this mode can sometimes be confusing. The remedy is to hide the edges on the far side. The graphics card can process the wire frame render mode very fast. See the following illustration.

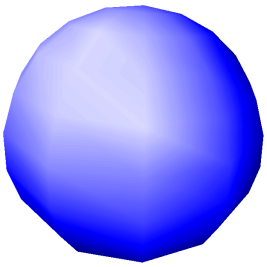


Hidden line mode Hidden line mode draws the edges as in wire frame mode, but only the ones that would be visible to the user if the object were opaque. The result is much less confusing to the eye, but it requires more computation time. MotionDesk does not support this mode.

Flat shaded mode Flat shaded mode gives a rough view of the surface and color of the object. The objects appear faceted, but the effects of lighting can be seen, see the following illustration created with MotionDesk.



Smooth shaded mode Smooth shaded mode shows the surface of the object with color and smoothing. The object looks accurate but it requires more computation time than the flat shaded mode, see the following illustration created with MotionDesk.



As the illustration above shows, using smooth shaded mode can provide a smooth display of an object. A large number of polygons is not always necessary.

Smooth textured mode Smooth textured mode shows the surface mapped with images. The object looks very realistic, but enormous computing power is necessary. This is provided by hardware-accelerated graphics cards.

The more accurate or detailed the display mode, the longer it takes to redraw the display when something has changed. This can add up to a long processing time, especially with complex models or a scene with a lot of objects.

Lights and Shadows

Lights

To get brightness into a virtual world, you need lights. There are four main kinds of lights.

Point lights Point lights are like bare bulbs that radiates light equally in all directions away from a light source.

Spotlights Spotlights are conical light flows, for example, the light of a torch. They can be used to highlight portions of an object.

Distant lights Distant lights are directional sources from a long distance. Their light rays are parallel, like the sun.

Ambient light Ambient light is present everywhere in a virtual world. It illuminates every object in the virtual world equally.

MotionDesk uses a distant light for sunlight and a point light for headlights. The sunlight illuminates the whole scene like the sun. Headlights are used with the observers to illuminate the area for the observers. Both light types are specified with Atmospherics properties.

Shadows

In the real world shadows are casted when lights illuminate. For greater realism, this should also be the same in a virtual world. The shadows of 3-D objects must be calculated according to the light sources.

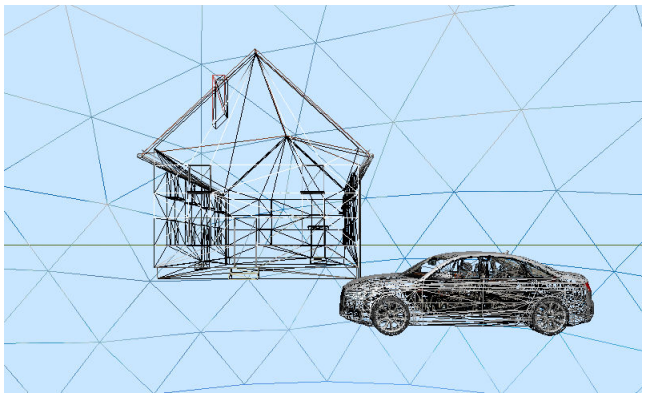
In MotionDesk, the shadows of 3-D objects are calculated according to the specified sunlight. As shadow calculation requires computation power, the calculation can reduce the frame rate of the animation. If the frame rate

becomes too slow, you can disable shadow calculation for single 3-D objects or the whole virtual world.

3-D Scene

Introduction

A virtual world (also called a 3-D scene) is an arrangement of 3-D objects in the 3-D view. Every object is separate from the others and can be placed anywhere in the scene: car body, wheels, houses, trees, etc. See the following illustration created with MotionDesk.



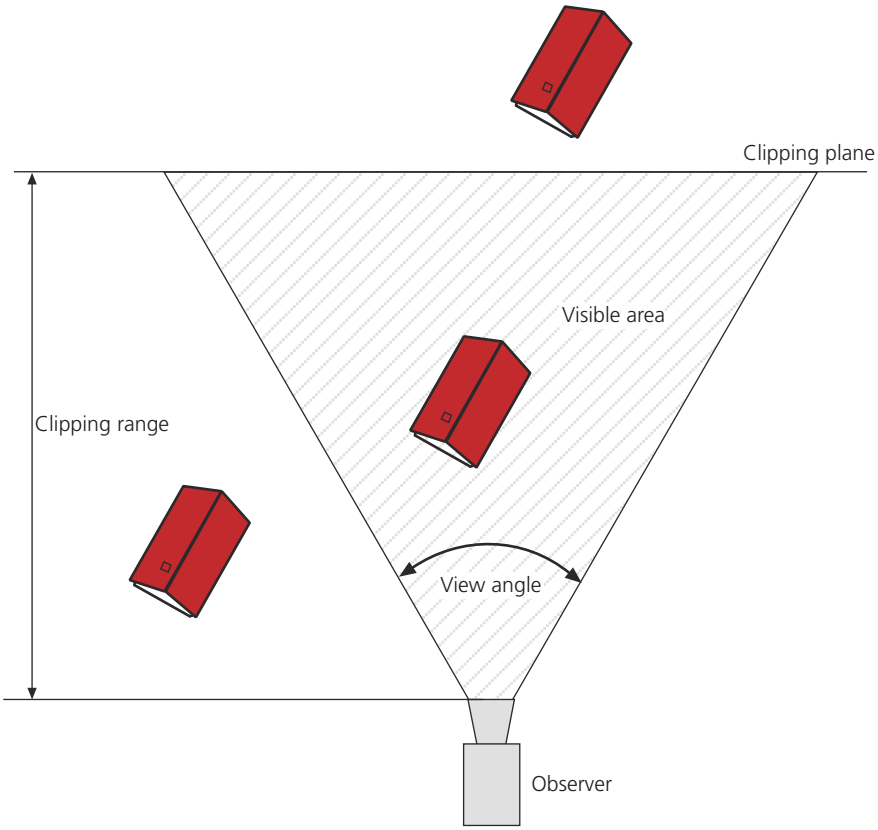
In the illustration above, the scene is displayed in wire frame mode. About 12,300 polygons are used for this example. It looks more realistic in the smooth textured mode, see the following illustration created with MotionDesk.

To explore the scene, you can look at it from different viewpoints, see Viewpoints, Clipping Plane on page 30.

Viewpoints, Clipping Plane

Viewpoint

A viewpoint is a position in a virtual world that represents the user's eye position (like a camera). In MotionDesk the user's eye position is defined as an observer. Surrounding the viewpoint at a perpendicular angle is the clipping plane, an imaginary flat panel that defines the limits of the user's sight. Only objects inside the visible area are seen and everything else is clipped off, see the following illustration.



You can place the observer anywhere in the scene in the 3-D view to get a view of the 3-D objects from different angles or distances. Only objects in front of the observer and in front of the clipping plane are displayed.

Rendering 3-D Graphics

Introduction

Rendering is the process by which the 3-D data is transformed to a 2-D representation, for example, the screen of your display. There are different

rendering methods, which differ in the quality they provide and the processing power they require.

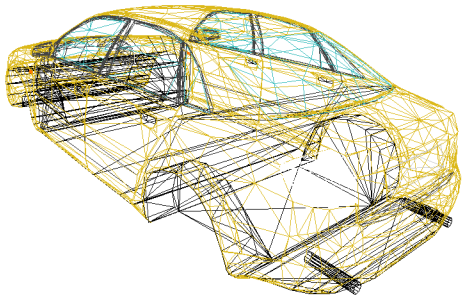
Where to go from here

Information in this section

Rendering 3-D Objects with Wire Frames

Wire frames

The following illustration shows a car rendered with wire frames.



Description

You can see all the edges of the model's polygons. This is the fastest rendering method but the 3-D object does not look very realistic.

Rendering 3-D Objects with Texture Mapping

Texture mapping

Texture mapping is the process of adding a graphic pattern from a bitmap image to the polygons of a 3-D scene. Unlike simple shading, which applies colors to

the polygons of the scene, texture mapping applies tiled bitmaps to give a realistic graphical appearance to walls, floors, the sky, and so on. The following illustration shows a car rendered with Gouraud shading and texture mapping.

Description

The car is rendered with Gouraud shading for a smooth representation. In addition, some images are mapped on the car to give it a realistic appearance. The dashboard, rear light and license plate in the illustration above are examples. Rendering with texture mapping takes a lot of processing time. This is provided by modern hardware-accelerated graphics cards. It is the technique with the most realistic effect available in MotionDesk.

Rendering 3-D Objects with Ray Tracing

Ray tracing

Ray tracing provides the highest quality of rendering, where the color and brightness of each pixel on the screen are calculated by casting an imaginary ray backward from the user's viewpoint onto the model. The ray can be bounced off surfaces and bent, just like real light. The result is very realistic, with extremely accurate shadows, reflections and even refraction. This method needs too much processing power, so it is not useful in an animation with MotionDesk and therefore not supported by MotionDesk. It is used to render films offline overnight for presentation purposes.

Rendering Engines

Introduction

A rendering engine supports the rendering of 3-D graphics.

OpenGL

OpenGL is the standard for professional CAD tools and scientific visualization applications. It is a hardware-independent 3-D API (Application Programming Interface) developed by Silicon Graphics. There are also OpenGL drivers that are executed for special cards. MotionDesk uses this interface.

DirectX

DirectX is an API for multimedia applications and games developed by Microsoft. It contains routines for 2-D and 3-D graphics (Direct Draw and Direct 3D), audio samples (Direct Sound), input devices like joysticks (Direct Input) and network connections for games (Direct Play).

Basics on Animation in 3-D with MotionDesk

Introduction

The screen must be redrawn if the viewpoint changes or objects change their positions. A picture of a viewpoint at one point in time is called a frame. To achieve animation, several frames are shown one after the other.

Where to go from here

Information in this section

Movable Objects and Static Objects

Introduction

A MotionDesk scene consists of movable objects and static objects.

Movable Object

When a dynamic mechanical system is visualized, the motions of some objects are simulated. These objects are called movable objects.

The movable objects are the objects that move in the scene. It is therefore not necessary to specify their positions or align them. A movable object gets its data from a simulation or a motion data file. You can also specify offset values for the data that additionally move or rotate the object.

Static object

Other objects are static in the scene and are called static objects.

Static objects are for visualization purposes only. They are ignored in simulations, for example, you can drive right through them.

You need several objects to create a realistic scene. A ground plate and a horizon are essential. To make this basic environment more realistic, you may add more objects like traffic signs, houses and trees for an automotive environment.

Material management

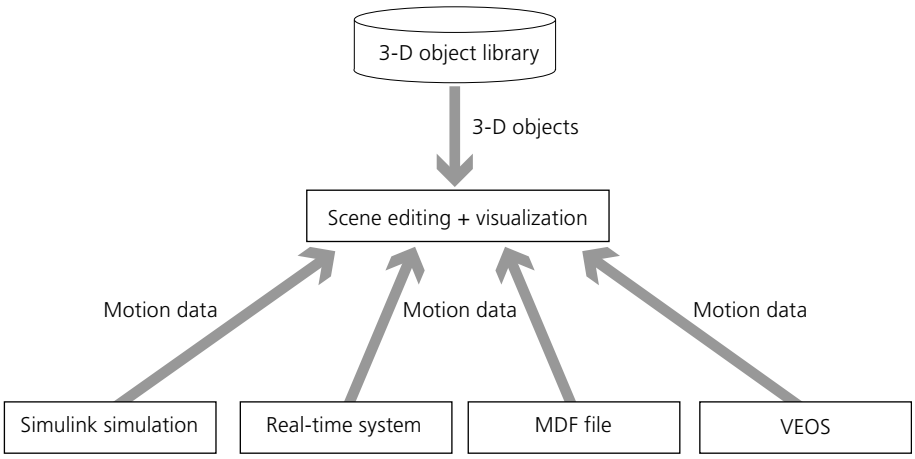
MotionDesk provides a material management database. The materials can be assigned pixel by pixel to the static and movable objects in the object library. The materials in the database can be extended with behavior characteristics, for example, reflection, or transparency values.

The material assignment to static and movable objects is important in sensor simulation when adding sensors to the testing architecture of control units in ADAS systems.

Simulation and Visualization

Introduction

MotionDesk visualizes the movement of objects in a scene (virtual world). The objects are moved according to data calculated in a simulation. The following illustration gives an overview of visualization and the scene creation/editing process.



Data sources

To be able to move objects, their coordinates must be calculated and sent to MotionDesk as motion data. The real-time application must therefore be prepared. You must use blocks of the MotionDesk Blockset in your Simulink model or use MotionDesk-specific code in your handcoded model. Note that handcoded models are not supported for each platform type.

MotionDesk supports several data sources.

Simulink simulation The simulation model is not calculated on a real-time system, but on the host PC. The Simulink model must include additional services that provide the motion data for the visualization. The visualization acquires the motion data and applies it to the movable objects in the 3-D world. The preparations that are necessary to create a PC-based Simulink simulation are almost identical to those for a real-time application.

Real-time system The real-time application, for example, the car dynamics, is calculated on the real-time system. The real-time application must include additional services that provide the motion data for visualization. The visualization process acquires the motion data and applies it to the movable objects in the 3-D world.

MDF file The motion data can be recorded and saved to a motion data file (MDF file). This can be used to replay the visualization. When you replay the visualization, a real-time system is not needed.

VEOS The VEOS platform provides access to one environment VPU that runs in a VEOS simulation. The environment VPU must contain the MotionDesk services that calculate motion data and send them to MotionDesk. The preparations that are necessary to create a VEOS simulation are almost identical to those for a real-time application.

3-D Object Library

The 3-D Object Library that comes with MotionDesk contains 3-D objects for building a scene in the 3-D View. All the objects in it can be used for vehicle dynamic simulation and are eminently suitable for real-time simulation, for example, roads, chassis, wheels. For proper visualization, you have to select objects which fit the simulation model.

This library also includes graphics texturing files. Each pixel in the 3-D object is assigned to materials form the material database for use with sensor simulation.

Scene editing and visualization

In MotionDesk, you can edit and visualize scenes in the 3-D View.

Scene editing In MotionDesk's 3-D View, you can assemble the 3-D objects to create a scene. You have to distinguish between static and movable objects. Static objects include the ground plate, horizon, road, houses, and trees. These objects form the virtual world in which the movable objects are moved. Movable objects include the chassis, wheels, and steering wheels. Only these objects can be animated in MotionDesk.

For more information about editing a scene in the 3-D View, refer to MotionDesk Scene Creation 🕮.

Visualization In the visualization, the animated 3-D scene is shown in the 3-D View on the display. The movable objects defined in the scene in the 3-D View are moved according to the motion data calculated in the real-time application or real-time model, or read from an MDF file.

Frame rate and latency

Frame rate and latency are two parameters that describe the quality of an animation. The time between the scene being computed in the simulation and displayed on the monitor is called the latency. It should be as short as possible, especially in a "man-in-the-loop" scenario.

The number of frames shown in one second is called the frame rate and is stated in fps (frames per second). The frame rate should be as high as possible to get a smooth animation. The higher the frame rate, the smoother the animation and the lower the latencies.

Application Scenarios

Introduction

There are three different application scenarios in the 3-D visualization.

"Man-in-the-loop"

If a human driver "drives" the virtual vehicle, he or she is enclosed in the control loop, for example, controlling the steering angle of a car and getting visual stimuli (responses) from the visualization system. This is called a "man-in-the-loop" (MIL) scenario. The simulation and visualization are done in real time. To give the driver the real driving experience, a high frame rate and low latencies are required. Typical requirements for latencies for MIL applications are lower than 50 ms.

Automatic maneuver

When the driving maneuver is defined in the simulation and there is no need for human online intervention, this is called an automatic maneuver scenario. The simulation and visualization are done in real time. In contrast to a "man-in-the-loop" scenario, the frame rate may be lower and the latency may be higher. It does not matter if there is a small delay in visualizing the changes occurring in the simulation; for example, a latency of 100 ms may be sufficient.

Presentation

In a presentation, the simulation and visualization are not in real time. Usually, the simulation results are stored in a file and the graphic is animated offline. The focus is on a high quality of graphical representation and on independence from the simulator.

System Overview

Introduction

Depending on the application scenario, there are different requirements for the hardware of your visualization system.

Simulink simulation

You can visualize your simulation without a connection to a real-time system, for example, in an early development phase. Refer to Simulink Simulation on page 38.

1-PC solution

If your visualization task does not need much processing power, working with 1 PC may be sufficient. Refer to 1-PC Solution on page 39.

Multi-PC solution

You can share the tasks of calculating the model and visualizing the motion data by working with 2 or more PCs. This also enables you to place different views on separate displays. Refer to Multi-PC Solution on page 40.

Comparing the visualization systems

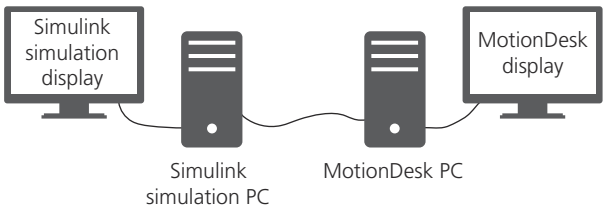
The following table describes the difference between the systems.

Characteristic	Simulink Simulation	1-PC Solution	Multi-PC Solution
Suitable application	Early development phase	Automatic maneuver	Man-in-the-loop
Frame rate	Standard	Standard	High
Latency	High	High	Low
Architecture	Visualization and PC-based simulation run on one or two PCs	Visualization and real-time simulation run on one PC	The motion data from the real-time simulation is transmitted to one or more PCs via Ethernet

Simulink Simulation

Introduction

The following illustration gives an overview of a Simulink simulation.



If you are in an early development stage, it may be sufficient to run your simulation as a PC-based Simulink simulation without any real-time hardware. In this case you can visualize the simulation in MotionDesk if your PC has a network adapter. As you have to use the same network sender block as in a multi-PC Solution, it is quite easy to switch from a PC-based Simulink simulation to a multi-PC solution later on.

Depending on your settings in the dialog of the network sender block, you can use one or two PCs for simulation and visualization.

1 PC If you use one PC, it has two tasks: it calculates the simulation and visualizes it via MotionDesk. You need a PC with high performance, especially for 3-D visualization. The PC must have a network adapter, but it does not need to be connected to a network.

2 PCs If you use two PCs, the Simulink simulation runs on one PC and MotionDesk runs on the other. The performance is better for complex Simulink simulations.

Related topics

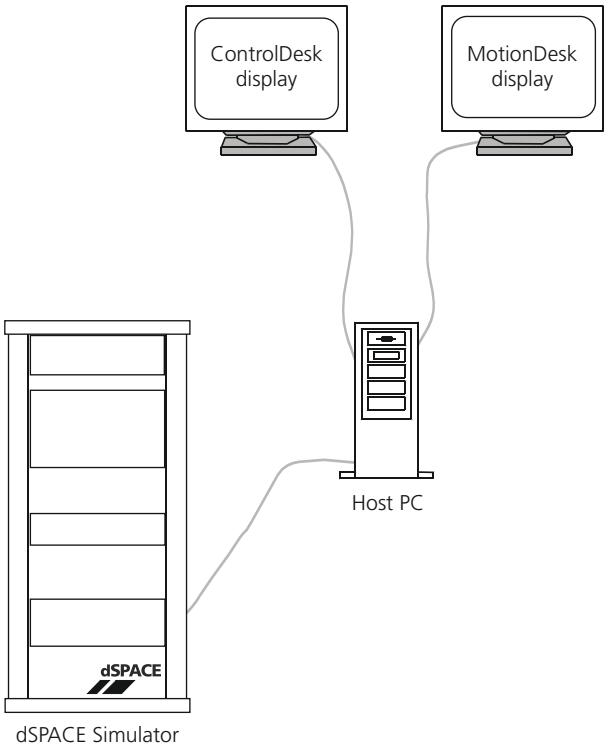
Basics

Setting Up Visualization for a Simulink Simulation (MotionDesk Calculating and Streaming Motion Data 🕮)

1-PC Solution

Introduction

The following illustration gives an overview of a 1-PC solution.

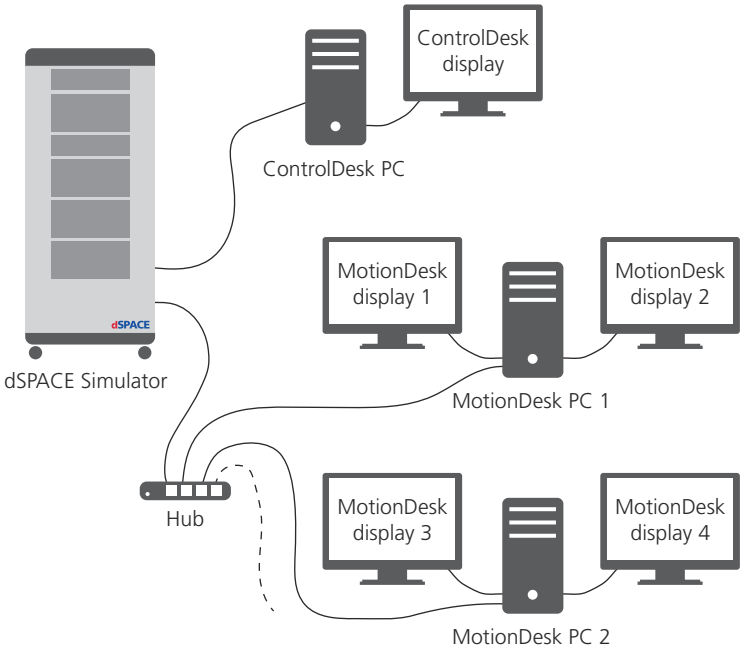


A dSPACE Simulator Full-Size is shown in the illustration above. However, there are different types of dSPACE Simulator depending on your simulation

	requirements: different processor boards (DS1006, or DS1007) installed in the host PC or dSPACE simulator, MicroAutoBox, and SCALEXIO system.
dSPACE Simulator	The simulator calculates the real-time model for a hardware-in-the-loop test. The real-time model must include the 3-D services that provide the motion data for MotionDesk. The simulator is connected to a host PC via dSPACE link boards or via Ethernet. The control and motion data is transferred to the host PC through these connections.
Host PC	In a 1-PC solution, the host PC has two tasks: controlling the simulation and visualizing the simulation. The simulation is controlled by ControlDesk, and the 3-D visualization is done by MotionDesk. You need a PC with high performance, especially for 3-D visualization.
Displaying ControlDesk and MotionDesk	ControlDesk and MotionDesk should each have their own display. The control display and the animation can then be viewed at the same time. If you work with one display, you have to toggle the ControlDesk and MotionDesk window or arrange the windows so that both are visible on one screen. If you work with two displays, your host PC needs a 2-channel graphics card to serve two displays. Generally, a graphics card with two outputs has a lower performance than a graphics card with one output.

Multi-PC Solution

Introduction	You can use several PCs for visualizing the same scene from different point of views. The following illustration gives an overview of a multi-PC solution.



A dSPACE Simulator Full-Size is shown in the illustration above. However, there are different types of dSPACE simulator, depending on your simulation requirements.

For the multi-PC solution a dSPACE simulator with a DS1006 or DS1007 processor board or a SCALEXIO system is required.

dSPACE simulator based on DS1006

If the real-time hardware is a simulator based on a DS1006 processor board, it is normally connected to the ControlDesk PC via dSPACE link boards. The control data is transmitted via this connection to control the simulation with ControlDesk. The dSPACE simulator is also connected to one or more MotionDesk PCs via Ethernet (10 Mbit/s), using the MotionDesk Multi-PC Interface Kit and a network hub. The translation matrix is calculated on the simulator to produce the motion data. The motion data is then transferred to the MotionDesk PCs via the Ethernet connection.

dSPACE simulator based on DS1007 or SCALEXIO system

If the real-time system is a dSPACE simulator based on DS1007 or SCALEXIO system, it is connected to the ControlDesk PC via Ethernet. The control data is transmitted via this connection to control the simulation with ControlDesk. The DS1007 or SCALEXIO system is also connected to one or more MotionDesk PCs via Ethernet. The translation matrix is calculated on the DS1007 or SCALEXIO system to produce the motion data. The motion data is then transferred to the MotionDesk PCs via the Ethernet connection.

ControlDesk PC

ControlDesk runs on one PC to control the simulation. A standard PC is sufficient.

MotionDesk PC

MotionDesk runs on the other PC or PCs to visualize the simulation. Visualization requires a high-performance PC that has a suitable high-end graphics card. Additionally, the MotionDesk PC should have 2 network adapters, one for the company network and one especially for the visualization tasks.

Displaying ControlDesk and MotionDesk

ControlDesk and MotionDesk should each have their own display. The control display and the animation can then be viewed at the same time. If you work with one display, you have to toggle the ControlDesk and MotionDesk window or arrange the windows so that both are visible on one screen. If you work with two displays, your host PC needs a 2-channel graphics card to serve two displays.

Generally, a graphics card with two outputs has a lower performance than a graphics card with one output.

Related topics

Basics

Setting Up a Multi-PC Solution (MotionDesk Calculating and Streaming Motion Data 🕮)

Working with the MotionDesk Demos

Introduction	Provides an overview on working with the MotionDesk demos provided with the installation.

Working with the MotionDesk Demos

Introduction	You can open and use the MotionDesk Demos provided with the installation to learn how to work with MotionDesk.
Overview	You can start working with MotionDesk and create a project. You can also open and work with the demo projects in MotionDesk that are provided with the installation. The demos are installed in the `<MotionDesk_InstallationPath>\Demos\MotionDesk` folder. For each of the demos, the simulation is only available using the motion data file (MDF). Demo models for supported simulation platforms are no longer delivered. The demos delivered with the MotionDesk installation are described as follows.
ADAS Instruments	The ADAS Instruments demo contains a MotionDesk project that displays a road network scene with an ego-vehicle including vehicle instruments, for example, for speed, gear, and engine revolutions. You can open the MotionDesk project and play the animation using a motion data file.
Automotive	The automotive demo contains a MotionDesk project that displays a road network scene with an ego-vehicle and fellows. You can open the MotionDesk project and play the animation using a motion data file. Refer to the following lessons in the MotionDesk tutorial: ▪ Lesson 3: Animating the Scene with Data from a File (MotionDesk Tutorial 🕮) ▪ Lesson 6: Working with a Predefined Road from ModelDesk (MotionDesk Tutorial 🕮)
Character	The character demo contains a MotionDesk project that displays a road network scene with an ego-vehicle, fellows, and animated characters. You can play the animation using a motion data file. Refer to the following lesson in the MotionDesk tutorial: ▪ Lesson 8: Using Objects in a MotionDesk Animation (MotionDesk Tutorial 🕮)

City

The city demo contains a MotionDesk project that displays a road network in a city scene with an ego-vehicle and fellows. The city network was created by ModelDesk. You can open the MotionDesk project and play the animation using a motion data file.

Scene Generation

The scene generation demo contains a MotionDesk project that displays a road network scene with various scenery types, for example, country and city sceneries. The road network was created by ModelDesk. It contains an ego-vehicle, fellows, and other static objects. You can play the animation using a motion data file.

For more information on ModelDesk and generating roads, refer to the following:

- Introduction to the Road Generator (ModelDesk Road Creation 🕮)

Sensor Simulation

The Sensor Simulation demo contains examples and tools to help you work with Sensor Simulation using MotionDesk.

Sensor Simulation demos

- **Roadworks highway**: Contains a MotionDesk project that displays a highway roadworks scene with an ego-vehicle that has attached sensors. The roadworks network was created by ModelDesk. You can play the animation using a motion data file and view the sensor composition window.

  Refer to the following lesson in the MotionDesk tutorial: Lesson 1: First Steps in Sensor Simulation (Sensor Simulation Tutorial 🕮).
- **Camera postprocessing**: Contains examples of how to postprocess sensor simulation data.

  Refer to Example of Postprocessing the Images for a Camera and Fish-Eye Sensor (MotionDesk Sensor Simulation Control 🕮).

Note

The demos are also available in the `<SensorSimulation_InstallationPath>\Demos\SensorSimulation\SensorSimulation` folder.

Shared memory access

The shared memory access demo contains tools are used with Sensor Simulation. You can save the sensor data output calculated by each SensorSim application instance to a dedicated shared memory storage area. You can then access and read the shared memory using a DLL library to further process the data and share it with external applications. You can also configure the delivered shared memory synchronizer to synchronize the output of multiple SensorSim applications to shared memory.

For more information on shared memory and the demo tools available, refer to the following:

- Shared Memory Access (Sensor Simulation Manual 🕮)

Note

The shared memory access tools are also available in the `<SensorSimulation_InstallationPath>\Demos\SensorSimulation\SharedMemoryAccess` folder. This folder also contains the synchronizer tool.

Related topics

Basics

Overview of Lessons (MotionDesk Tutorial 🕮)
Overview of Lessons (Sensor Simulation Tutorial 🕮)
Tutorial (ASM User Guide 🕮)

HowTos

How to Open a Project and Experiment (MotionDesk Project and Experiment Management 🕮)
How to Replay an Animation (MotionDesk Scene Animation 🕮)

Reference Information

Where to go from here

Information in this section

Basic Interface

Introduction

MotionDesk's Basic Interface provides various commands and dialogs, which are accessible via the menu bar. The provided dialogs and commands are:

Where to go from here

Information in this section

Start page..... 60
To display the start page.

About MotionDesk

Access

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

Purpose

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

Result

Here you can see MotionDesk's version number.

Related topics

References

dSPACE Help..... 49

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 MotionDesk.

Result

The user documentation of MotionDesk opens.

Related topics

References

Using dSPACE Help.....60

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 MotionDesk.

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....22

HowTos

How to Add External Programs as User Functions....23

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 79.

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 79.

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 56

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 22

HowTos

How to Add External Programs as User Functions 23

References

Customize (User Functions) 50
User Functions Output 59

Exit

Access

You can access this command via:

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

Purpose

To exit the current MotionDesk session.

Result

MotionDesk ends the current session.

Description

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

Related topics

HowTos

How to Start and Shut Down MotionDesk 12

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 49
- Using dSPACE Help on page 60
- New Features and Migration on page 56
- PDF Files on page 58
- Keyboard Help on page 55

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

MotionDesk 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 MotionDesk.

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.

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.

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 79.

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 79.

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 51

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.

Status bar

Access

You can access this command via:

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

Purpose

To show or hide the status bar at the bottom the main window.

Result

If there is a checkmark to the left, the status bar is shown at the bottom of the main window. If there is no checkmark, it is hidden.

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) 50
Execute (User Function) 53

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.

Start page

Access

You can access this 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.

Description

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 Project + Experiment/New Experiment Lets you define a new project or a new experiment.

Open Project + Experiment Lets you open an existing project and experiment.

Recent Lets you open one of the recently opened projects.

Close page after project has loaded Lets you specify to automatically close the Start Page after you have opened a project.

Show page on startup Lets you specify to display the Start Page on every start of MotionDesk.

Basic Practices/Advanced Practices Lets you access MotionDesk documentation.

Related topics

References

New Project + Experiment/New Experiment (MotionDesk Project and Experiment Management 🕮)
Open Project + Experiment (MotionDesk Project and Experiment Management 🕮)

Window Handling

Introduction

MotionDesk's window handling provides various commands and dialogs, which are accessible via the menu bar and the context menus of the MotionDesk components:

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

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 MotionDesk's main window, the component is docked if MotionDesk 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 72

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 MotionDesk's working area.

Result

All minimized windows are arranged horizontally at the bottom of MotionDesk'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 Options Page on page 83.

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 MotionDesk'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.

Tip

The most important window in MotionDesk is the 3-D View. You can change the number and arrangement of the views in the 3-D View via the One View, Tile Horizontally, Tile Vertically, and Four Views commands.

Related topics

References

Four Views (MotionDesk Scene Animation 🕮)
One View (MotionDesk Scene Animation 🕮)
Tile Horizontally (MotionDesk Scene Animation 🕮)
Tile Vertically (MotionDesk Scene Animation 🕮)

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 MotionDesk.

Result All unchanged windows are closed.

Description

The 3-D View cannot be closed.

Related topics

References

Close All But This....67
Close Window....67

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....66
Close Window....67

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....66
Close All But This....67

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

MotionDesk 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 21

References

Reset View 74

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..... 19

References

Reset View..... 74

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.
- You cannot dock the **3-D View** window.

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.

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 64
Dock/Docked 70
Float/Floating 71

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.

More Windows

Access

You can access this command via:

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

Purpose

To arrange windows in MotionDesk's working area as you require them.

Result

Windows are arranged in MotionDesk's working area according to your settings.

Description

The Arrange Windows dialog opens for you to arrange all the windows available in MotionDesk's working area.

Select Window Lets you select the windows whose settings you want to change.

Activate Lets you activate the selected window.

Close Window Lets you close the selected window.

Cascade Lets you cascade the selected windows in MotionDesk's working area.

Tile Horizontally Lets you arrange the selected windows in MotionDesk's working area horizontally.

Tile Vertically Lets you arrange the selected windows in MotionDesk's working area vertically.

Minimize Lets you minimize the selected windows.

Reset View

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 MotionDesk was first installed.

Result

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

Description

The controlbars of MotionDesk, 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 MotionDesk. You can use this command to reset the user interface to its default.

Related topics

References

Customize Quick Access Toolbar / More commands 68
Customize Ribbon 69

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.....19

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 MotionDesk:

- Data Stream Selector
- Library Browser
- Messages
- Motion Diagnostics
- Motion Player
- Project
- Properties
- Scene Navigator
- 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 MotionDesk 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.

Tip

The most important window in MotionDesk is the 3-D View. You can change the number and arrangement of the views in the 3-D View via the One View, Tile Horizontally, Tile Vertically, and Four Views commands.

Related topics

HowTos

How to Adjust the 3-D View Windows (MotionDesk Scene Animation 🕮)

References

Four Views (MotionDesk Scene Animation 🕮)
One View (MotionDesk Scene Animation 🕮)
Tile Horizontally (MotionDesk Scene Animation 🕮)
Tile Vertically (MotionDesk Scene Animation 🕮)

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 MotionDesk 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.

Tip

The most important window in MotionDesk is the 3-D View. You can change the number and arrangement of the views in the 3-D View via the One View, Tile Horizontally, Tile Vertically, and Four Views commands.

Related topics

HowTos

How to Adjust the 3-D View Windows (MotionDesk Scene Animation 🕮)

References

Four Views (MotionDesk Scene Animation 🕮)
One View (MotionDesk Scene Animation 🕮)
Tile Horizontally (MotionDesk Scene Animation 🕮)
Tile Vertically (MotionDesk Scene Animation 🕮)

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 MotionDesk's working area.

Result

If there is a checkmark to the left, the Workbook mode is enabled in MotionDesk'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 **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....64
Cascade....65
Tile Horizontally....76
Tile Vertically....77
Views Options Page....83

Dialogs and Pages

Where to go from here

Information in this section

Filter Panel

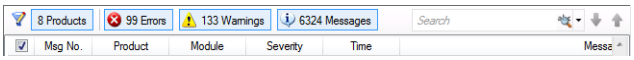
Access

- The filter panel is part of the **Message Viewer**. Refer to Messages on page 56.

  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 51.

  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 51
Messages 56

MotionDesk Options

Access

You can modify the MotionDesk Options via:

Ribbon	File – Options
Context menu of	None
Shortcut key	**Ctrl+Shift+G**
Icon	

Purpose

To view and change the global user settings of MotionDesk.

Result

The MotionDesk Options dialog is open

Description

If you change a setting in any of the pages in the **MotionDesk Options** dialog, they apply to the current and all subsequent MotionDesk sessions.

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

- **Graphics Capabilities Options Page** (MotionDesk Scene Animation 🕮)
- **Project Options Page** (MotionDesk Project and Experiment Management 🕮)
- **Render Options Page** (MotionDesk Scene Animation 🕮)
- **Road Complexity Options Page** (MotionDesk Scene Creation 🕮)
- **Road Generation Options Page** (MotionDesk Scene Creation 🕮)
- **Scenery Complexity Options Page** (MotionDesk Scene Creation 🕮)
- **Sensor Simulations Options Page** (MotionDesk Sensor Simulation Control 🕮)
- **Single Fullscreen Options Page** (MotionDesk Scene Animation 🕮)
- **Stereo Fullscreen Options Page** (MotionDesk Scene Animation 🕮)
- **User Settings Page** on page 81
- **Views Options Page** on page 83
- **Visualization Options Page** (MotionDesk Scene Animation 🕮)
- **Vivid Texturing Options Page** (MotionDesk Scene Creation 🕮)

User Settings Page

Access

This page is part of the **MotionDesk Options** dialog.

Purpose

To export, import, or reset user specific settings.

Description

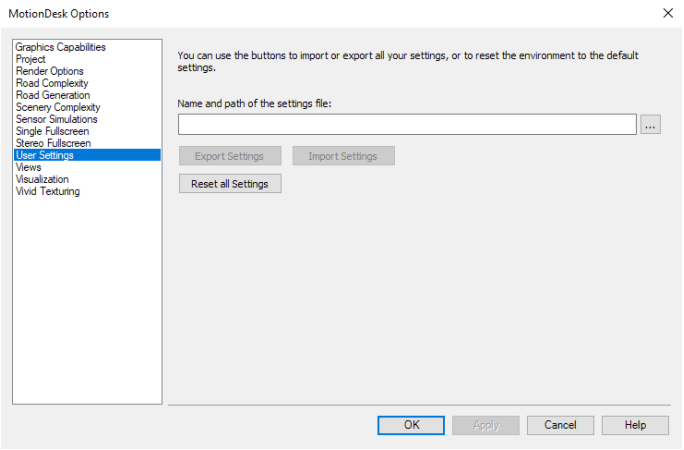
MotionDesk stores all the information that has to be saved for the next session as environment settings in a UCD file. This information is read out when MotionDesk is started again and ensures that user-specific settings, for example, a dialog window in a different position, can be restored.

The **Import and Export Settings** dialog allows you to save the current state of the environment settings manually, for example in the following use cases:

- You are experimenting with different settings and want to save a favorite one.
- You want to transfer the environment settings to a MotionDesk installation on another PC. In this case the PCs must be as similar as possible. For example, screen solutions must not differ.

You can also reset the environment settings to the state of the first installation.

MotionDesk user settings dialog



Dialog settings

Export settings Lets you save the current environment settings. Specify the export file in the Name and path of the settings file text box.

Import settings Lets you restore exported environment settings. Specify the import file in the Name and path of the settings file text box.

Note

You must restart MotionDesk to let the environment settings take effect. All the changes made to MotionDesk settings between closing the Import and Export Settings dialog and restarting MotionDesk will get lost.

Reset all settings Lets you reset the environment settings to the state of the first installation. You must restart MotionDesk to let the initial environment settings take effect.

Note

If you reset the environment settings, all the user-specific settings, even all the defined users themselves, get lost.

Name and path of the settings file Lets you specify the path and name of the export or import file. Click the Browse button to browse through the file system.

Related topics

References

MotionDesk Options....80

Views Options Page

Access

This page is part of the MotionDesk Options dialog.

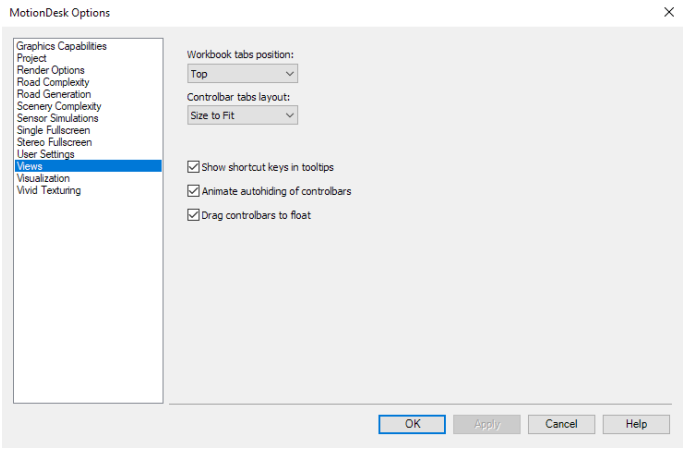
Purpose

To alter the settings for workbook tabs and controlbar tabs.

Description

You can change the size and layout of the MotionDesk tabs and controlbars.

MotionDesk views dialog



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 (refer to Workbook Mode on page 78). 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 MotionDesk's working area.

Controlbar tabs layout You can customize the window layout of MotionDesk. If you have docked several components to a window, it provides tabs to switch between them. 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.

Layout	Description
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. For details on auto hiding, refer to Auto Hide on page 65.

Related topics

References

MotionDesk Options....80

Automation

Where to go from here

Information in this section

Information in other sections

MotionDesk Automation
Provides information on how you can use the automation interface of MotionDesk.

Classes for Handling MotionDesk

Where to go from here

Information in this section

Application

Purpose

To access MotionDesk.

Where to go from here

Information in this section

Class Description (Application)

Syntax

```
from win32com.client import Dispatch
MyApplication = Dispatch("MotionDesk.Application")
```

Purpose

To access MotionDesk.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveProject	ActiveProject[1)]	To access the active MotionDesk project.
ConnectionManagement	ConnectionManager[2)]	To get and set the sensor simulation application instance connections.
LibraryManagement	LibraryManager[3)]	To access the custom object library.
ProductVersion	String	To get the version of MotionDesk.
SimulationManagement	SimulationManager[4)]	To access simulation management to start and stop the animation.
Visible	Boolean	To get/set the visible state of the graphical user interface. True if it is visible.
WindowManagement	WindowManager[5)]	To get/set the view mode of MotionDesk (normal or full screen).

1) Refer to ActiveProject (MotionDesk Project and Experiment Management 🕮).
2) Refer to ConnectionManager (MotionDesk Sensor Simulation Control 🕮).
3) Refer to LibraryManager (MotionDesk Custom Object Library Management 🕮).
4) Refer to SimulationManager (MotionDesk Scene Animation 🕮).
5) Refer to WindowManager on page 92.

Methods

The class contains the following methods:

Method	Purpose
NewProject	To create a new MotionDesk project. Refer to NewProject on page 88.
OpenProject	To open a MotionDesk project. Refer to OpenProject on page 88.
Quit	To exit MotionDesk. Refer to Quit on page 89.

NewProject

Class

Application

Syntax

```
MyApplication.NewProject(ProjectPath, ProjectName, SaveActive)
```

Purpose

To create a new MotionDesk project.

Parameters

The method uses the following parameters:

Parameter	Type	Description
ProjectPath	String	The name of the folder where the project is created.
ProjectName	String	The name of the project.
SaveActive	Boolean	Defines whether to save a project that is open. True if the project is saved (default).

Return value

The method returns the following parameter:

Type	Description
ActiveProject[1)]	Instance of a project class.

[1)] Refer to ActiveProject (MotionDesk Project and Experiment Management 🕮).

Related topics

References

Application..... 86

OpenProject

Class

Application

Syntax

```
MyApplication.OpenProject(ProjectPath, InitialExperiment, SaveActive)
```

Purpose

To open a MotionDesk project.

Parameters

The method uses the following parameters:

Parameter	Type	Description
ProjectPath	String	The path and name of the project to be loaded.
InitialExperiment	String	The name of the experiment to be initially activated. This parameter is optional. If an empty string is specified, only the project is opened.
SaveActive	Boolean	Defines whether to save a project that is open. True if the project is saved (default).

Return value

The method returns the following parameter:

Type	Description
ActiveProject[1)]	Instance of a project class.

[1)] Refer to ActiveProject (MotionDesk Project and Experiment Management 🕮).

Related topics

References

Application.....86

Quit

Class

Application

Syntax

```
MyApplication.Quit(saveAll)
```

Purpose

To exit MotionDesk.

Parameters

The method uses the following parameters:

Parameter	Type	Description
saveAll	Boolean	If True, the MotionDesk project is saved when MotionDesk exits.

Return value

–

Related topics

References

Application....86

View

Purpose

To get or set the settings of a view.

Class Description (View)

Syntax

```
MyBottomLeftView = Views.BottomLeft
MyBottomRightView = Views.BottomRight
MyTopLeftView = Views.TopLeft
MyTopRightView = Views.TopRight
```

Purpose

To get or set the settings of a view.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveObserver	String	To get/set the observer used in the view.
Name	String	To get the name of the view.

Methods

–

Related topics

References

Class Description (ViewManager)....91

ViewManager

Purpose

To get or set the views.

Class Description (ViewManager)

Syntax

```
MyViews = Experiment.VisualizationManagement.Views
```

Purpose

To get or set the settings of the views.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveView	String	To get/set the observer used in the view.
BottomLeft	View[1)]	To get the bottom left view.
BottomRight	View[1)]	To get the bottom right view.
HorizontalSplitterPosition	Integer	To get/set the horizontal splitter position.
TopLeft	View[1)]	To get the top left view.
TopRight	View[1)]	To get the top right view.
VerticalSplitterPosition	Integer	To get/set the vertical splitter position.
ViewMode	ViewModes[2)]	To set the view mode.

1) Refer to View on page 90.
2) Refer to ViewModes on page 93

Methods

–

Related topics

References

Class Description (Experiment) (MotionDesk Project and Experiment Management 🕮)

WindowManager

Purpose To get/set the view mode of MotionDesk (normal or full screen).

Class Description (WindowManager)

Syntax

```
WindowManager = Application.WindowManagement
```

Purpose To get/set the view mode of MotionDesk (normal or full screen).

Attributes The class contains the following attributes:

Attributes	Type	Purpose
MainWindowMode	MainWindowModes[1)]	To get/set the mode of the main window (normal or full screen).

1) Refer to MainWindowModes on page 93

Methods –

Related topics

References

Class Description (Application)..... 87

Enumerations

Enumerations for Handling MotionDesk

Enumerations

The following constants can be used for tool automation.

IdentifierTypes Constants to identify objects.

Value	Description
ByName	Object is identified by its name.
ByIndex	Object is identified by its index.
ByCustomIdentifier	Object is identified by a custom identifier.

MainWindowModes Constants to specify the mode of the main window.

Value	Description
Normal	MotionDesk is displayed a window.
FullScreen	MotionDesk is displayed in full screen.

ViewModes Constants to specify the view mode:

Value	Description
OneView	The scene is displayed in one view.
Horizontal	The scene is displayed in two views, horizontally arranged.
Vertical	The scene is displayed in two views, vertically arranged.
FourViews	The scene is displayed in four views.

Related topics

References

Enumerations for Scene Animation (MotionDesk Scene Animation 🕮)
Enumerations for Scene Creation (MotionDesk Scene Creation 🕮)

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 96.

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 96

Examples

Example of Reading Messages with C# 98
Example of Reading Messages with Python 96

References

MessageReaderSettings Class 105

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..... 94

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 95.

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 96.

```
# 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 96.

```
# 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 104.

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 96.

```
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 94
Supported dSPACE Products and Components 96

References

MessageReaderSettings Class 105

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 95.

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 104.

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API 94

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 102)
Severity	Gets the severity of the message.	Get	Severity (refer to Severity Enumeration on page 107)

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 94

Examples

Example of Reading Messages with C# 98
Example of Reading Messages with Python 96

References

ILogSession Interface 102
Severity Enumeration 107

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 94

Examples

Example of Reading Messages with C# 98
Example of Reading Messages with Python 96

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 105

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 101) >

1) <Type> `Name`: Description

Related topics

Basics

Reading dSPACE Log Messages via the Message Reader API 94

Examples

Example of Reading Messages with C# 98
Example of Reading Messages with Python 96

References

MessageReaderSettings Class 105

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....94

Examples

Example of Reading Messages with C#....98
Example of Reading Messages with Python....96

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....94

Examples

Example of Reading Messages with C#....98
Example of Reading Messages with Python....96

Limitations

General Limitations of MotionDesk

Introduction	The following general limitations apply.
MotionDesk on notebooks	▪ MotionDesk has high requirements with regard to graphics performance and driver quality. In some cases MotionDesk will not operate properly. ▪ The notebook must be equipped with a supported NVIDIA graphics accelerator. There must be only one graphics processing unit on the computer. ▪ MotionDesk does not support notebooks with NVIDIA Optimus technology. For details, refer to http://www.dspace.com/go/mdhwrequ.
Limitations with file paths	MotionDesk does not support the use of the special characters (& : < > ' ") in any file path. Due to several known issues with using Unicode characters, for example, German umlauts, Japanese characters, etc., especially in combination with whitespaces, it is recommended to use only English characters in the file paths and strings in MotionDesk.
Screen savers	Screen savers using the OpenGL rendering engine may cause problems.
Using 4K displays	If you have a 4K display and the 4K resolution is set, you can no longer use the mouse for the dialogs in MotionDesk. To avoid this problem, reduce the resolution of the display.

Troubleshooting (General)

Where to go from here

Information in this section

Information in other sections

Troubleshooting (Scene Creation) (MotionDesk Scene Creation 🕮)

Troubleshooting (Scene Animation) (MotionDesk Scene Animation 🕮)

Internal Error Occurred During Program Start

Problem

MotionDesk does not start because an internal error occurred.

Reason

This error can occur if two graphics drivers are installed on the PC, for example, a graphics driver from NVIDIA and Intel.

Solution	It is recommended to use the NVIDIA graphics driver. You have two solutions: ▪ In the NVIDIA Control Panel, select the NVIDIA processor as preferred graphics processor ▪ In Window's Device Manager, deactivate the Intel graphics driver. In addition, check whether a monitor is connected to the Intel chip. Normally, that is the case when a monitor is connected to the back panel of the motherboard.

Dialogs are Unusable in 4K Resolution

Problem	If you have 4K displays and the 4K resolution is set, the dialogs in MotionDesk cannot be used with the mouse.
Reason	The frame work that is used by MotionDesk does not support the 4K resolution of 4K displays.
Solution	Reduce the resolution of the displays.