AutomationDesk

Accessing ControlDesk

For AutomationDesk 6.5

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

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

How to Contact dSPACE Support

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

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

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

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

Software Updates and Patches

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

Important Notice

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.

© 2017 - 2021 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

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

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

Contents

About This Document 7

Basics and Instructions 9

Overview of the ControlDesk Access Library Elements ... 9
Example of a ControlDesk Access Sequence ... 10
Adding Further ControlDesk API Functionality in AutomationDesk ... 12

Reference Information 15

Automation Blocks ... 16

Convenience ... 16

Application ... 17
CloseControlDesk ... 17
CloseProjectAndExperiment ... 18
OpenProjectAndExperiment ... 19
SaveProjectAndExperiment ... 20
StartControlDesk ... 20

Snapshot ... 21
Snapshot ... 21

Calibration ... 22
GetParameterValue ... 23
ReadVariableValue ... 24
SetParameterValue ... 26
StartOnlineCalibration ... 27
StopOnlineCalibration ... 28
SwitchMemoryPage ... 28
WriteVariableValue ... 29

Recording ... 31
AddSignalsToMainRecorder ... 31
GetRecordedData ... 32
RemoveUnconnectedSignalsFromMSL ... 34
StartMeasurementAndRecording ... 34
StopMeasurementAndRecording ... 35

Platform ... 36
AddPlatform ... 37
AddPlatformByID ... 37
AddVariableDescription ... 38
ConfigureDeviceGeneralSettings ... 40
ConnectPlatform ... 41
DisconnectPlatform ... 42
GetMemorySegments ... 43
GetPlatformStates ... 44
RemovePlatform ... 45

Diagnostics ... 46
AddDiagResultDataToReport ... 47
CreateDiagResultDataFromResponses ... 48
DirectClearAllDTCs ... 49
DirectClearDTC ... 50
DirectReadDTCs ... 51
DirectReadDTCsWithEnvData ... 52
DirectReadEnvironmentData ... 53
GetDiagPlatform ... 54
GetActiveLogicalLink ... 55
DirectExecuteHexService ... 56
DirectExecuteService ... 57
DirectExecuteServiceUsingCustomPDU ... 59

Basic Functions ... 60

Common ... 60
ActiveExperiment ... 60
Application ... 61
Platform (Common) ... 61
Platforms (Common) ... 62

Diagnostic ... 63
ActiveDiagnosticsDatabase ... 66
ActiveLogicalLink ... 66
ActiveLogicalLinks ... 67
ControlPrimitive ... 68
ControlPrimitives ... 69
CtrlPrimitiveRequestParameter ... 69
CtrlPrimitiveRequestParameters ... 71
CtrlPrimitiveResponse ... 71
CtrlPrimitiveResponseParameter ... 72
CtrlPrimitiveResponseParameters ... 73

CtrlPrimitiveResponseRequestParameter ... 74
CtrlPrimitiveResponseRequestParameters ... 75
CtrlPrimitiveResponses ... 76
Job ... 77
JobRequestParameter ... 77
JobRequestParameters ... 79
JobResponse ... 79
JobResponseParameter ... 80
JobResponseParameters ... 81
JobResponseRequestParameter ... 82
JobResponseRequestParameters ... 83
JobResponses ... 84
Jobs ... 84
LogicalLinks ... 85
LogicalLinkSelection ... 86
Platform (Diagnostics) ... 86
Platforms (Diagnostics) ... 87
PPSetRequestParameter ... 88
PPSetRequestParameters ... 89
PPSetResponse ... 90
PPSetResponseParameter ... 91
PPSetResponseParameters ... 92
PPSetResponseRequestParameter ... 92
PPSetResponseRequestParameters ... 93
ProtocolParameterSet ... 94
SelectedVehicle ... 95
Service ... 96
ServiceRequestParameter ... 96
ServiceRequestParameters ... 98
ServiceResponse ... 99
ServiceResponseParameter ... 99
ServiceResponseParameters ... 101
ServiceResponseRequestParameter ... 101
ServiceResponseRequestParameters ... 102
ServiceResponses ... 103
Services ... 104
Vehicle ... 105
Vehicles ... 105
VehicleSelection ... 106

Automation 107

Basics on Automating the Access to ControlDesk..... 107

Index 109

About This Document

Content

This document gives you information on how to access ControlDesk via AutomationDesk.

Required knowledge

Working with AutomationDesk requires:

- Basic knowledge in handling the PC and the Microsoft Windows operating system.
- Basic knowledge in developing applications or tests.
- Basic knowledge in handling the external device, which you control remotely via AutomationDesk.

dSPACE provides trainings for AutomationDesk. For more information, refer to https://www.dspace.com/go/trainings.

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.

Symbol	Description
🕮	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

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

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

Special folders

Some software products use the following special folders:

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

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

or

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

Documents folder A standard folder for user-specific documents.

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

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

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

Accessing dSPACE Help and PDF Files

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

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

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

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

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

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

Basics and Instructions

Where to go from here

Information in this section

Overview of the ControlDesk Access Library Elements 9
Lists AutomationDesk's ControlDesk Access library elements that let you access the ControlDesk API.

Example of a ControlDesk Access Sequence 10
Provides an example of automating access to ControlDesk via its API.

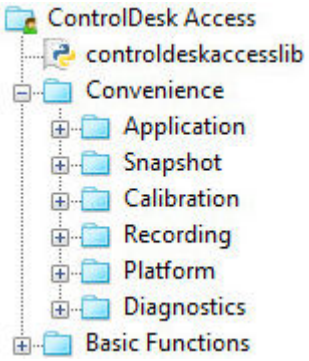
Adding Further ControlDesk API Functionality in AutomationDesk 12
Basic information on implementing a custom library with additional API functionality of ControlDesk.

Overview of the ControlDesk Access Library Elements

Library overview

The AutomationDesk ControlDesk Access library provides some of the functionality of ControlDesk's API to AutomationDesk.

The library folder elements in the AutomationDesk ControlDesk Access library are shown in the illustration below:



The library consists of two sections:

- Convenience

  Provides blocks that include several basic functions from the ControlDesk API in one single block to allow standard access to ControlDesk. For an overview of the folders, see below.

- Basic Functions

  Provides blocks that represent one single function from the ControlDesk API to allow specific access to ControlDesk. For detailed information, refer to Basic Functions on page 60.

Application

The Application folder provides automation blocks to control ControlDesk's user interface. For further information, refer to Application on page 17.

Snapshot

The Snapshot folder provides an automation block to read the current values of variables from the ECU. For further information, refer to Snapshot on page 21.

Calibration

The Calibration folder provides automation blocks to manage your calibration task. For further information, refer to Calibration on page 22.

Recording

The Recording folder provides automation blocks to record measurement data. For further information, refer to Recording on page 31.

Platform

The Platform folder provides automation blocks to manage the access to the platform/device. For further information, refer to Platform on page 36.

Diagnostics

The Diagnostics folder provides automation blocks to manage access to the diagnostic features of ControlDesk. For further information, refer to Diagnostics on page 46.

Related topics

Basics

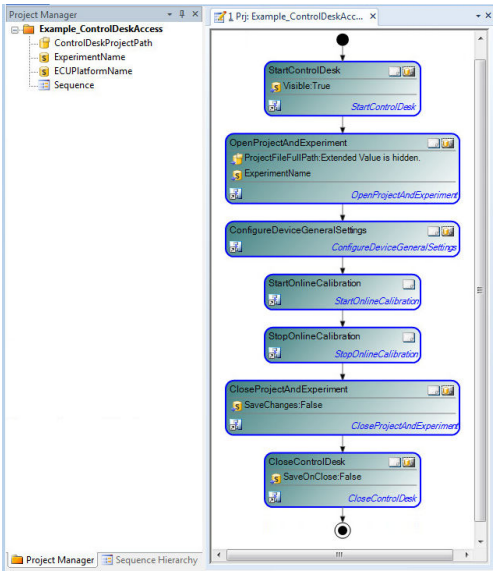
Packaging of AutomationDesk (AutomationDesk Introduction And Overview 🕮)

Example of a ControlDesk Access Sequence

Introduction

The following example contains basic automation blocks and data objects to automate access to ControlDesk via its API.

AutomationDesk sequence



Description of the sequence

The sequence shown in the illustration above contains the following automation blocks:

- StartControlDesk

  To start ControlDesk with its user interface.
- OpenProjectAndExperiment

  To open an experiment of a specified project.
- ConfigureDeviceGeneralSettings

  To configure platform/device properties.
- StartOnlineCalibration

  To start online calibration for all platforms in the active experiment.
- StopOnlineCalibration

  To stop online calibration for all platforms in the active experiment.
- CloseProjectAndExperiment

  To close the currently active project in ControlDesk.
- CloseControlDesk

  To stop the active project and to exit ControlDesk.

Example project

You will find example project `ControlDeskAccessExample.zip`, which uses the ControlDesk Access library, at `<DocumentsFolder>\ControlDesk Access`.

Adding Further ControlDesk API Functionality in AutomationDesk

Introduction

The ControlDesk Access library provides basic features of the automation API of ControlDesk. You can extend AutomationDesk if you need more of the API's functionality. The steps to implement custom automation blocks with more API functionality are described below.

For an example implemented with source code, automation blocks in a custom library, and an AutomationDesk project, refer to ExtensionExample demo on page 13.

Extending AutomationDesk

Writing a Python module To extend AutomationDesk you must first write your own Python module. In this module, you must define Python functions which contain methods of the ControlDesk API. For example:

```
def CheckActiveExperimentExists():
    # Checks if an Experiment is loaded in ControlDesk
    if not GetApplicationObject().ActiveExperiment:
        raise DSControlDeskNGLibraryException( \
            "No active ControlDesk experiment found")
    return True
```

For information on the API methods, refer to Introduction to the ControlDesk Automation API (ControlDesk Automation 🕮).

Note

You must import the ControlDesk Access Library Python module to your own module via `import controldeskaccesslib`, which contains a modified Dispatch method that ensures proper COM handling in AutomationDesk.

Example 1:

```
# Create COM Objects
Application = controldeskaccesslib.GetApplicationObject()
```

Example 2:

```
raise controldeskaccesslib.DSControlDeskNGLibraryException( \
        "RemoveVariableDescription: No PlatformName specified")
```

Tip

You can import the `mainlibrary` module to get the following information in AutomationDesk's Message Viewer:

- `LogError(Stringtext)`
- `LogInfo(Stringtext)`
- `LogWarning(Stringtext)`

Example:

```
mainlibrary.LogInfo( \
          "AddProjectAndExperiment, Project name: %s, \
          Experiment name: %s"%(ProjectName, ExperimentName))
```

Creating a custom library with specific automation blocks With your Python module, you can create new automation blocks and save them to a custom library.

First, create a new custom library and integrate your Python module to it. Refer to How to Integrate Python Sources (AutomationDesk Basic Practices 🕮).

For the new automation block, use a **Serial** block in which you place an **Exec** block. In the Serial block, you can create data objects to parameterize your specific automation block. Importing your Python module (`import <name_of_your_python_module>`) in the **Editor** of the Exec block makes all your implemented functions available. Now you can write Python code that uses one or more of these functions (see Using Python in AutomationDesk (AutomationDesk Basic Practices 🕮)).

Note

Add the data objects you use in your **Exec** block to the **Serial** block to create an interface for the variables (see Serial (AutomationDesk Basic Practices 🕮)).

Drag your new automation block to your custom library.

For further information, refer to How to Create a Block Template (AutomationDesk Basic Practices 🕮).

Tip

To view the source code of each ControlDesk Access Library block, double-click it and open the **Editor** of the contained **Exec** block.

Now you can use the automation blocks of your custom library together with the automation blocks of the ControlDesk Access Library to run automation tasks in ControlDesk.

ExtensionExample demo

You will find the ExtensionExample demo at `<DocumentsFolder>\ControlDesk Access`.

The following files are relevant:

- Demo project **ControlDeskAccess_ExtensionExample.zip** to be imported in AutomationDesk.

  This demo project is based on the **CalDemo** of ControlDesk and is executable on a simulated ECU.
- Custom library **Custom Ext ControlDesk Access.zip** to be imported in AutomationDesk providing automation blocks that use the functions of the implemented Python module. The library has to be loaded before executing the demo project.

  For information on importing a custom library, refer to How to Import a Custom Library from a ZIP Archive (AutomationDesk Basic Practices 🕮).
- Python module **controldesklib_extension.py** providing additional functionality such as reloading the variable description file. It is stored in the custom library archive and extracted to the custom library path after importing it.

Related topics

Basics

Working with Custom Libraries (AutomationDesk Basic Practices 🕮)

Reference Information

Automation Blocks

Where to go from here

Information in this section

Convenience 16
Provides convenience blocks that include several basic functions from the ControlDesk API in one single block to control ControlDesk's graphical user interface.

Basic Functions 60
Provides basic blocks that represent one single function from the ControlDesk API to control ControlDesk's graphical user interface.

Convenience

Using ControlDesk Access library features in Python scripts

You can use functions and other definitions of the ControlDesk Access library in Python scripts after you imported the `controldeskaccesslib` module to the current namespace.

Where to go from here

Information in this section

Application 17
Provides automation blocks to control ControlDesk's user interface.

Snapshot 21
Provides an automation block to read the current values of variables from the ECU.

Calibration 22
Provides automation blocks to manage your calibration task.

Recording 31
Provides automation blocks to record measurement data.

Platform 36
Provides automation blocks to manage the access to the platform/device.

Diagnostics 46
Provides automation blocks to control ControlDesk's diagnostics interface.

Application

Where to go from here

Information in this section

CloseControlDesk 17
To close the user interface of ControlDesk.

CloseProjectAndExperiment 18
To close the currently active project in ControlDesk.

OpenProjectAndExperiment 19
To open an experiment of a specified project.

SaveProjectAndExperiment 20
To save the active project and experiment in ControlDesk.

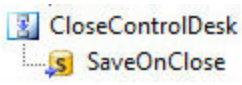
StartControlDesk 20
To start ControlDesk with its user interface visible or hidden.

Information in other sections

Application Handling (ControlDesk Automation 🕮)

CloseControlDesk

Graphical representation



Purpose

To stop the active project and to exit ControlDesk.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
SaveOnClose	In	String	"False"	Specifies whether the active application is stored before closing ControlDesk: ▪ True: active project will be saved ▪ False: ControlDesk will be closed without saving

For more information, refer to Application Handling (ControlDesk Automation 🕮).

Related topics

Examples

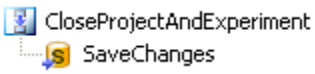
Example of a ControlDesk Access Sequence 10

References

CloseProjectAndExperiment 18
OpenProjectAndExperiment 19
SaveProjectAndExperiment 20
StartControlDesk 20

CloseProjectAndExperiment

Graphical representation



Purpose

To close the currently active project in ControlDesk.

Description

The CloseProjectAndExperiment automation block closes project and experiment, ControlDesk is still running. To close ControlDesk use the CloseControlDesk automation block.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
SaveChanges	In	String	"False"	Specifies whether the currently active project is stored before it is closed: ▪ True: active project will be saved ▪ False: active project will not be saved

For more information, refer to Application Handling (ControlDesk Automation 🕮).

Related topics

Examples

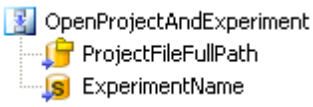
Example of a ControlDesk Access Sequence 10

References

CloseControlDesk 17
OpenProjectAndExperiment 19
SaveProjectAndExperiment 20
StartControlDesk 20

OpenProjectAndExperiment

Graphical representation



Purpose

To open an experiment of a specified project.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ProjectFileFullPath	In	File	" "	Specifies the path to the ControlDesk project file (CDP).
ExperimentName	In	String	" "	Specifies the name of the experiment.

For more information, refer to Application Handling (ControlDesk Automation 🕮).

Related topics

Examples

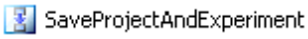
Example of a ControlDesk Access Sequence 10

References

CloseControlDesk 17
CloseProjectAndExperiment 18
SaveProjectAndExperiment 20
StartControlDesk 20

SaveProjectAndExperiment

Graphical representation



Purpose

To save the active project and experiment in ControlDesk.

Description

The active project and experiment is saved to the path and with the file name that you specified in ControlDesk.

Data objects

None

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

CloseControlDesk 17
CloseProjectAndExperiment 18
OpenProjectAndExperiment 19
StartControlDesk 20

StartControlDesk

Graphical representation

Purpose

To start ControlDesk with its user interface visible or hidden.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Visible	In	String	"True"	Specifies how the user interface of ControlDesk starts: ▪ True: visible ▪ False: hidden

For more information, refer to Application Handling (ControlDesk Automation 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

CloseControlDesk 17
CloseProjectAndExperiment 18
OpenProjectAndExperiment 19
SaveProjectAndExperiment 20

Snapshot

Where to go from here

Information in this section

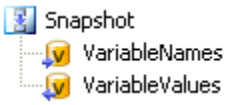
Snapshot 21
To read variables by ControlDesk's snapshot function.

Information in other sections

Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮)

Snapshot

Graphical representation



Purpose

To read the values of variables with ControlDesk's snapshot function.

Description

Taking a snapshot allows you to read the current values of variables from the ECU and save them in a comma-separated values (CSV) file. Before taking a snapshot you have to specify the variables to be saved by selecting a label list file.

Note

As of ControlDesk 7.1, you can no longer take snapshots. If you use the Snapshot block in AutomationDesk, an exception occurs. Use the ReadVariableValue block instead.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
VariableNames	In	Variant	None	Specifies a single value or a list of variable names from which you want to take a snapshot of. For example: ['air_mass', 'Model Root/SigGenOut']
VariableValues	Out	Variant	None	Contains a variable value or values of a variable list. For example: [12, 76.67]

For more information, refer to Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

Calibration

Where to go from here

Information in this section

GetParameterValue 23
To return the value of a specified parameter.

ReadVariableValue 24
To read the value of a specified variable.

SetParameterValue.....26
To write a value to a specified parameter.

StartOnlineCalibration.....27
To start the online calibration.

StopOnlineCalibration.....28
To stop the online calibration.

SwitchMemoryPage.....28
To switch between the reference page and working page of the corresponding ECU.

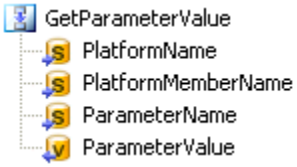
WriteVariableValue.....29
To write a value to a specified variable.

Information in other sections

Event Management Handling (ControlDesk Automation 🕮)

GetParameterValue

Graphical representation



Purpose

To return the value of a specified parameter.

Note

- With ControlDesk 7.0, data set handling was modified. Direct access to parameters and parameter values of a data set via ControlDesk automation is no longer supported. If you use the **GetParameterValue** block in AutomationDesk, an exception occurs. Use the **ReadVariableValue** block instead.
- With ControlDesk 6.4 and earlier, you need at least one data set in ControlDesk to execute this block. For information on data sets, refer to Managing Data Sets (ControlDesk Calibration and Data Set Management 🕮).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the platform/device used in ControlDesk. For example: XCP
PlatformMemberName	In	String	" "	Optional: Specifies the name of the platform member of MC or MP systems.
ParameterName	In	String	" "	Specifies the name of the parameter whose value you want to get. For example: SignalOffset
ParameterValue	Out	Variant	None	Contains the return value of the specified parameter. For example: ▪ Scalar: [Value]; ▪ Curve: [[XAxisValue], [Value]]; ▪ Map: [[XAxisValue], [YAxisValue], [Value]]

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

SetParameterValue 26
StartOnlineCalibration 27
StopOnlineCalibration 28
SwitchMemoryPage 28

ReadVariableValue

Graphical representation

Purpose

To read the value of a specified variable.

Description

The variable you want to read the value from has to be specified by the platform name and its unique name. For a multicore or a multiprocessor system, you additionally have to specify the platform member name. The format of the value is specified by the ValueType data object. You can use the value's original format, for example, a hexadecimal value, or a converted physical value. For the conversion, a conversion table or conversion formula is to be specified in the variable description. For further information, refer to How to Display a Value in Converted or Source Mode (ControlDesk Variable Management 🕮).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform to be read the variable value from. For example: `MyDS1005`
PlatformMemberName	In	String	" "	Specifies the platform that is a member of a multicore or multiprocessor (MC/MP) system. This data object need not be parameterized for a single-processor system. For example: `Platform_2`
VariableUniqueName	In	String	" "	Specifies the name and the model path of the variable to be read. For example: `Model Root/SubSystem/Gain/Out`
ValueType	In	String	"Converted"	Specifies in which format the value is to be returned with the VariableValue data object: ▪ Source The value is used in its original format. ▪ Converted The value is used in the converted format.
VariableValue	Out	Variant	None	Contains the read variable value in the format specified by the ValueType data object.

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

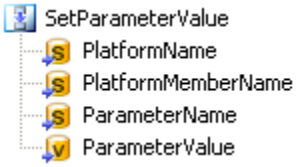
Example of a ControlDesk Access Sequence 10

References

AddVariableDescription 38
WriteVariableValue 29

SetParameterValue

Graphical representation



Purpose

To write a value to a specified parameter.

Note

- With ControlDesk 7.0, data set handling was modified. Direct access to parameters and parameter values of a data set via ControlDesk automation is no longer supported. If you use the **SetParameterValue** block in AutomationDesk, an exception occurs. Use the **WriteVariableValue** block instead.
- With ControlDesk 6.4 and earlier, you need at least one data set in ControlDesk to execute this block. For information on data sets, refer to Managing Data Sets (ControlDesk Calibration and Data Set Management 🕮).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the platform/device used in ControlDesk. For example: XCP
PlatformMemberName	In	String	" "	Optional: Specifies the name of the platform member of MC or MP systems.
ParameterName	In	String	" "	Specifies the name of the parameter, whose value you want to modify. For example: SignalOffset
ParameterValue	In	Variant	None	Specifies the values to be set. For example: ▪ Scalar: [Value]; ▪ Curve: [[XAxisValue], [Value]]; ▪ Map: [[XAxisValue], [YAxisValue], [Value]]

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

GetParameterValue 23
StartOnlineCalibration 27
StopOnlineCalibration 28
SwitchMemoryPage 28

StartOnlineCalibration

Graphical representation

Purpose

To start online calibration for all platforms in the active experiment.

Description

The devices go online with the predefined online behavior state.

Data objects

None

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

GetParameterValue 23
SetParameterValue 26
StopOnlineCalibration 28
SwitchMemoryPage 28

StopOnlineCalibration

Graphical representation

Purpose

To stop the online calibration for all platforms in the active experiment.

Data objects

None

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

GetParameterValue 23
SetParameterValue 26
StartOnlineCalibration 27
SwitchMemoryPage 28

SwitchMemoryPage

Graphical representation

Purpose

To switch between the reference page and working page of the corresponding ECU.

Note

You need at least one data set and one reference data set in ControlDesk to execute this block. For information on data sets, refer to Managing Data Sets (ControlDesk Calibration and Data Set Management 🕮).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the platform/device used in ControlDesk. For example: XCP
MemoryPage	In	String	"WorkingPage"	Specifies the memory page you want to activate. Two settings are possible: ▪ WorkingPage ▪ ReferencePage

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

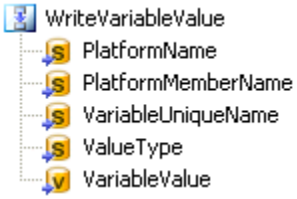
Example of a ControlDesk Access Sequence 10

References

GetParameterValue 23
SetParameterValue 26
StartOnlineCalibration 27
StopOnlineCalibration 28

WriteVariableValue

Graphical representation



Purpose

To write a value to a specified variable.

Description

The variable you want to write the value to has to be specified by the platform name and its unique name. For a multicore or a multiprocessor system, you additionally have to specify the platform member name. The format of the value is specified by the ValueType data object. You can use the value's original format, for example, a hexadecimal value, or a converted physical value. For the conversion, a conversion table or conversion formula is to be specified in the

variable description. For further information, refer to How to Display a Value in Converted or Source Mode (ControlDesk Variable Management 🕮).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform to write the variable value to. For example: `MyDS1005`
PlatformMemberName	In	String	" "	Specifies the platform that is a member of a multicore or multiprocessor (MC/MP) system. This data object need not be parameterized for a single-processor system. For example: `Platform_2`
VariableUniqueName	In	String	" "	Specifies the name and the model path of the variable to be written. For example: `Model Root/SubSystem/Gain/Out`
ValueType	In	String	"Converted"	Specifies in which format the value is to be written with the VariableValue data object: ▪ Source The value is used in its original format. ▪ Converted The value is used in the converted format.
VariableValue	In	Variant	None	Specifies the variable value to be written. The write format is specified by the ValueType data object.

For more information, refer to Event Management Handling (ControlDesk Automation 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddVariableDescription 38
ReadVariableValue 24

Recording

Where to go from here

Information in this section

AddSignalsToMainRecorder 31
To add signals to the main recorder of ControlDesk.

GetRecordedData 32
To return the recorded data from the specified file as a dictionary.

RemoveUnconnectedSignalsFromMSL 34
To remove unconnected signals from the measurement signal list.

StartMeasurementAndRecording 34
To start recording immediately.

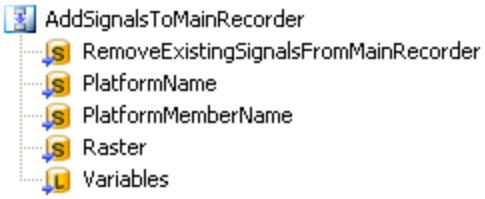
StopMeasurementAndRecording 35
To stop measurement and recording in ControlDesk.

Information in other sections

Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮)

AddSignalsToMainRecorder

Graphical representation



Purpose

To add signals to the main recorder of ControlDesk.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
RemoveExistingSignals\FromMainRecorder	In	String	"True"	Specifies whether all existing signals stored in the main recorder are removed before the specified signals are added:

Name	In / Out	Type	Default Value	Description
				▪ True: All stored signals are removed. ▪ False: The signals in the main recorder are not removed.
PlatformName	In	String	" "	Specifies the name of the platform/device you want to use. For example: ds1005
PlatformMemberName	In	String	" "	Optional: Specifies the name of the platform member of MC or MP systems.
Raster	In	String	" "	Specifies the measurement raster. A measurement raster defines how often a value of a variable is captured. For example: 5 ms
Variables	In	List	[]	Specifies the values to be set. For example: [u'Model Root/ThrottleController/act_pos/Out1', u'Model Root/ThrottleController/PT2/Out1', u'Model Root/SignalGenerator/Λ SignalGenerator/SignalForm/Value', u'Model Root/SignalGenerator/SignalGenerator/Λ SignalOffset/Value']

For more information, refer to Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮).

Related topics

Examples

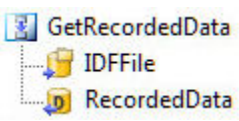
Example of a ControlDesk Access Sequence..... 10

References

GetRecordedData..... 32
RemoveUnconnectedSignalsFromMSL..... 34
StartMeasurementAndRecording..... 34
StopMeasurementAndRecording..... 35

GetRecordedData

Graphical representation



Purpose

To return the recorded data from the specified file as a dictionary.

Description

This block internally uses the `measurementdataapilib` Python module installed with ControlDesk. Because of the migration from Python 2.7 to Python 3.6 in dSPACE Release 2018-B, it is required to use only dSPACE software that supports the same Python version. If you use AutomationDesk 6.0 or later with ControlDesk 6.3 or earlier, an error message is displayed.

Note

As of ControlDesk 7.2, you can no longer record data in the IDF format. If you use the GetRecordedData or StopMeasurementAndRecording blocks in AutomationDesk with an IDF file specified for the IDFFile data object, an exception occurs. Use the MF4, CSV, or MAT file formats for the IDFFile data object instead. For best performance, use the MF4 file format, because the data is recorded in this format. The other file formats are created by converting the MF4 data.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
IDFFile	In	File	" "	Specifies the path and the name of the file with the recorded data. For example, files of the following formats are supported: ▪ MDF files ▪ CSV files ▪ MAT files
RecordedData	Out	Dictionary	{}	Contains the recorded data of the specified file. For example: {"XCP_5ms": "control_out": [1,2,1,2,1], ...}

For more information, refer to Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

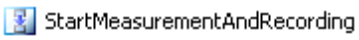
References

AddSignalsToMainRecorder 31
RemoveUnconnectedSignalsFromMSL 34
StartMeasurementAndRecording 34
StopMeasurementAndRecording 35

RemoveUnconnectedSignalsFromMSL

Graphical representation	
Purpose	To remove unconnected signals from the measurement signal list.
Description	The measurement signal list is a list containing the variables to be included in subsequent measurements and recording. A signal in the list is marked as unconnected, if it is not connected to an instrument and visualized on a layout in ControlDesk.
Data objects	None For more information, refer to Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮).
Related topics	Examples Example of a ControlDesk Access Sequence ..... 10 References AddSignalsToMainRecorder ..... 31 GetRecordedData ..... 32 StartMeasurementAndRecording ..... 34 StopMeasurementAndRecording 35

StartMeasurementAndRecording

Graphical representation	
Purpose	To start recording in ControlDesk immediately.
Data objects	None

Related topics

Examples

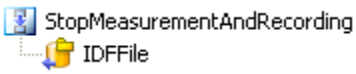
Example of a ControlDesk Access Sequence 10

References

AddSignalsToMainRecorder 31
GetRecordedData 32
RemoveUnconnectedSignalsFromMSL 34
StopMeasurementAndRecording 35

StopMeasurementAndRecording

Graphical representation



Purpose

To stop measurement and recording in ControlDesk.

Note

As of ControlDesk 7.2, you can no longer record data in the IDF format. If you use the **GetRecordedData** or **StopMeasurementAndRecording** blocks in AutomationDesk with an IDF file specified for the **IDFFile** data object, an exception occurs. Use the MF4, CSV, or MAT file formats for the **IDFFile** data object instead. For best performance, use the MF4 file format, because the data is recorded in this format. The other file formats are created by converting the MF4 data.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
IDFFile	Out	File	" "	Contains the path and the name of the last measurement data file. For example, files of the following formats are supported: ▪ MDF files ▪ CSV files ▪ MAT files

For more information, refer to Measurement and Recording Handling (ControlDesk Measurement and Recording 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddSignalsToMainRecorder 31
GetRecordedData 32
RemoveUnconnectedSignalsFromMSL 34
StartMeasurementAndRecording 34

Platform

Where to go from here

Information in this section

AddPlatform 37
To add a platform to the active experiment.

AddPlatformByID 37
To add a platform by its ID to the active experiment.

AddVariableDescription 38
To add a variable description to the specified platform.

ConfigureDeviceGeneralSettings 40
To configure the platform/device properties.

ConnectPlatform 41
To connect the specified platform.

DisconnectPlatform 42
To disconnect the specified platform.

GetMemorySegments 43
To return the memory segments of the specified platform/device.

GetPlatformStates 44
To return various states of the platform/device.

RemovePlatform 45
To remove a platform from the active experiment.

Information in other sections

Platform Handling (ControlDesk Platform Management 🕮)

AddPlatform

Graphical representation

Purpose

To add a platform to the active experiment.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the new platform/device name. For example: XCP
PlatformType	In	String	" "	Specifies the type of the platform/device to be added. For example: XCPonCAN, GME, ds1005...

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddPlatformByID 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
DisconnectPlatform 42
GetMemorySegments 43
GetPlatformStates 44
RemovePlatform 45

AddPlatformByID

Graphical representation

Purpose

To add a platform by its ID to the active experiment.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the new platform/device name. For example: XCP
PlatformTypeID	In	Int	0	Specifies the type ID of the platform/device to be added. For valid enumeration values, refer to PlatformType <<Enumeration>> (ControlDesk Automation 🕮).

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

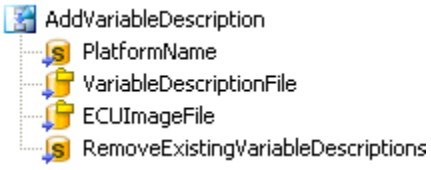
Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
DisconnectPlatform 42
GetMemorySegments 43
GetPlatformStates 44
RemovePlatform 45

AddVariableDescription

Graphical representation



Purpose

To add a variable description to the specified platform.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform/device name.
VariableDescriptionFile	In	File	None	Specifies the path and the name of the variable description file to be used. For example: `..\..\..\TestData\CalDemo\CalDemo.a2l`
ECUImageFile	In	File	None	Specifies the path and the name of the ECU image file to be used. For example: `..\..\..\TestData\CalDemo\CalDemo.mot`
RemoveExistingVariable\Descriptions	In	String	"True"	Specifies whether the variable descriptions are removed: ▪ True: existing variable descriptions will be deleted ▪ False: existing variable descriptions will not be removed

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

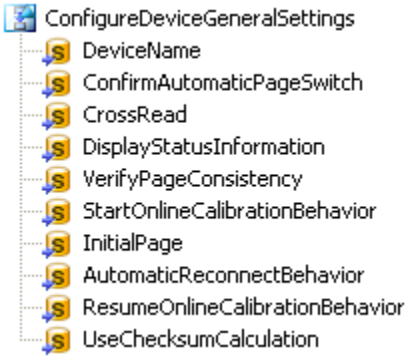
Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
DisconnectPlatform 42
GetMemorySegments 43
GetPlatformStates 44
RemovePlatform 45

ConfigureDeviceGeneralSettings

Graphical representation



Purpose

To configure the platform/device properties.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DeviceName	In	String	" "	Specifies the platform/device name. For example: XCP
ConfirmAutomaticPageSwitch	In	String	" "	Specifies whether the user has to confirm when automatic page switching during online calibration is started.
CrossRead	In	String	" "	Specifies whether a consistency check is performed for each write operation from ControlDesk to the ECU.
DisplayStatusInformation	In	String	" "	Specifies whether message boxes display status information for the selected platform/device.
VerifyPageConsistency	In	String	" "	Specifies whether ControlDesk is to perform a data consistency check after up/downloading the calibration memory contents when online calibration is started.
StartOnlineCalibrationBehavior	In	String	" "	Specifies whether ControlDesk's default behavior during online calibration is started. Note: Do not set the value of the **StartOnlineCalibration** parameter to "Prompt User". This can cause an exception when online calibration is started and the memory pages are different.
IntialPage	In	String	" "	Specifies which memory page is the active page in ControlDesk and on the ECU after online calibration is started or after an automatic reconnection.

Name	In / Out	Type	Default Value	Description
AutomaticReconnectBehavior	In	String	" "	Specifies ControlDesk's default behavior when the unplugged state is detected for the selected platform/device.
ResumeOnlineCalibration\ Behavior	In	String	" "	Specifies ControlDesk's default behavior when online calibration is started and measurement is resumed after ControlDesk has performed an automatic reconnection to the platform/device hardware.
UseChecksumCalculation	In	String	" "	Specifies whether a checksum calculation is performed.

Note

If you leave a value of a data object blank, this property will not be changed and the value that was set before is used in ControlDesk.

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

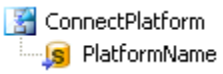
Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
AddVariableDescription 38
ConnectPlatform 41
DisconnectPlatform 42
GetMemorySegments 43
GetPlatformStates 44
RemovePlatform 45

ConnectPlatform

Graphical representation



Purpose

To connect the specified platform.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform/device name. For example: XCP

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
DisconnectPlatform 42
GetMemorySegments 43
GetPlatformStates 44
RemovePlatform 45

DisconnectPlatform

Graphical representation

Purpose

To disconnect the specified platform.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform/device name. For example: XCP

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
GetMemorySegments 43
GetPlatformStates 44
RemovePlatform 45

GetMemorySegments

Graphical representation

Purpose

To return the memory segments of the specified platform/device.

Description

The memory of an ECU or RapidPro system is usually divided into calibration memory segments containing the calibratable parameters.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform/device name.
MemorySegments	Out	List	[]	Contains a list of memory segments of the platform/device. For example: [{'Description': u'REFPAGE', 'EndAddress': 93218, 'StartAddress': 90112, 'Size': 3107, 'Type': 0, 'Name': u'REFPAGE'},...]

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

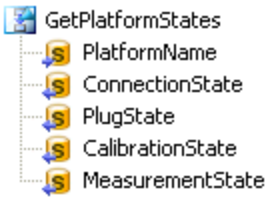
Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
DisconnectPlatform 42
GetPlatformStates 44
RemovePlatform 45

GetPlatformStates

Graphical representation



Purpose

To return various states of the platform/device.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the platform/device name.
ConnectionState	Out	String	" "	Contains the current connection state to indicate if the platform/device is connected (True/False).
PlugState	Out	String	" "	Contains the physical connection state of the platform/device (True/False).
CalibrationState	Out	String	" "	Contains the calibration state of the platform/device (True/False).
MeasurementState	Out	String	" "	Contains the measurement state of the platform/device (True/False).

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

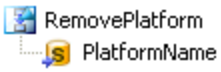
Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
DisconnectPlatform 42
GetMemorySegments 43
RemovePlatform 45

RemovePlatform

Graphical representation



Purpose

To remove a platform from the active experiment.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the platform to be removed. For example: XCP

For more information, refer to Platform Handling (ControlDesk Platform Management 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddPlatform 37
AddPlatformByID 37
AddVariableDescription 38
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
DisconnectPlatform 42

GetMemorySegments....43
GetPlatformStates....44

Diagnostics

Where to go from here

Information in this section

AddDiagResultDataToReport....47
To add diagnostics result data to the report.

CreateDiagResultDataFromResponses....48
To create diagnostics result data from the ResponsesObject data object.

DirectClearAllDTCs....49
To delete all trouble code entries from the fault memory.

DirectClearDTC....50
To delete a trouble code entry from the fault memory.

DirectReadDTCs....51
To read diagnostic trouble codes.

DirectReadDTCsWithEnvData....52
To read diagnostic trouble codes together with the environment data.

DirectReadEnvironmentData....53
To read the environment data of a specific diagnostic trouble code.

GetDiagPlatform....54
To get the diagnostics platform.

GetActiveLogicalLink....55
To get the active logical link.

DirectExecuteHexService....56
To execute the hex service.

DirectExecuteService....57
To execute the specified service directly.

DirectExecuteServiceUsingCustomPDU....59
To execute the specified service using custom PDU.

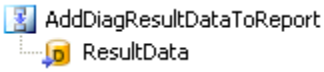
Information in other sections

ControlDesk ECU Diagnostics
The ControlDesk ECU Diagnostics Module lets you communicate with an ECU via the diagnostic interface.

ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮)

AddDiagResultDataToReport

Graphical representation



Purpose

To add diagnostics result data to the report.

Description

You can use the ResultData data object of the following automation blocks as input:

- DirectExecuteHexService on page 56
- DirectExecuteService on page 57
- DirectExecuteServiceUsingCustomPDU on page 59
- CreateDiagResultDataFromResponses on page 48
- DirectClearAllDTCs on page 49
- DirectClearDTC on page 50
- DirectReadDTCs on page 51
- DirectReadDTCsWithEnvData on page 52
- DirectReadEnvironmentData on page 53

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ResultData	In	Dictionary	{}	Specifies the result data after executing the service.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

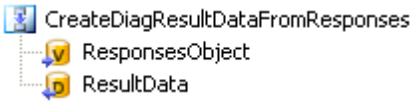
Example of a ControlDesk Access Sequence 10

References

CreateDiagResultDataFromResponses 48
DirectExecuteHexService 56
DirectExecuteService 57
DirectExecuteServiceUsingCustomPDU 59
GetActiveLogicalLink 55
GetDiagPlatform 54

CreateDiagResultDataFromResponses

Graphical representation



Purpose

To create diagnostics result data from the ResponsesObject data object.

Description

The contents of the ResponsesObject data object can come from the following Basic Functions automation blocks:

- ExecuteService, refer to Service on page 96.
- ExecuteHexService, refer to ActiveLogicalLink on page 66.
- ExecuteJob, refer to Job on page 77.
- ExecuteControlPrimitive, refer to ControlPrimitive on page 68.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ResponsesObject	In	Variant	None	Specifies the COM object of the responses.
ResultData	Out	Dictionary	{}	Contains the result data created from the COM object of the response.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

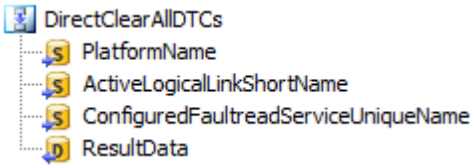
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47
DirectExecuteHexService 56
DirectExecuteService 57
DirectExecuteServiceUsingCustomPDU 59
GetActiveLogicalLink 55
GetDiagPlatform 54

DirectClearAllDTCs

Graphical representation



Purpose

To delete all trouble code entries from the fault memory.

Description

The ConfiguredFaultreadServiceUniqueName data object is optional. If the value is an empty string, the default element of the collection is taken.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.
ConfiguredFaultreadServiceUniqueName	In	String	" "	Specifies a unique name of the configured fault read service (can be an empty string).
ResultData	Out	Dictionary	{}	Contains the result data after clearing all diagnostic trouble codes (DTC).

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

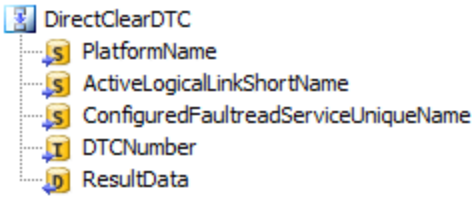
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47

DirectClearDTC

Graphical representation



Purpose

To delete a trouble code entry from the fault memory.

Description

The ConfiguredFaultreadServiceUniqueName data object is optional. If the value is an empty string, the default element of the collection is taken.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.
ConfiguredFaultreadServiceUniqueName	In	String	" "	Specifies a unique name of the configured fault read service (can be an empty string).
DTCNumber	In	Integer	0	Specifies the number of the DTC to be cleared.

Name	In / Out	Type	Default Value	Description
ResultData	Out	Dictionary	{}	Contains the result data after clearing the diagnostic trouble codes (DTC).

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

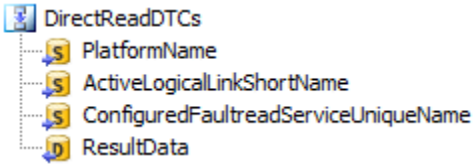
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47

DirectReadDTCs

Graphical representation



Purpose

To read diagnostic trouble codes.

Description

The ConfiguredFaultreadServiceUniqueName data object is optional. If the value is an empty string, the default element of the collection is taken.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.
ConfiguredFaultreadServiceUniqueName	In	String	" "	Specifies a unique name of the configured fault read service (can be an empty string).

Name	In / Out	Type	Default Value	Description
ResultData	Out	Dictionary	{}	Contains the result data after reading the diagnostic trouble codes (DTC).

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

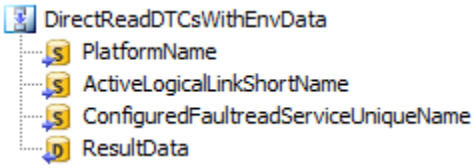
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47

DirectReadDTCsWithEnvData

Graphical representation



Purpose

To read diagnostic trouble codes together with the environment data.

Description

The ConfiguredFaultreadServiceUniqueName data object is optional. If the value is an empty string, the default element of the collection is taken.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.
ConfiguredFaultreadServiceUniqueName	In	String	" "	Specifies a unique name of the configured fault read service (can be an empty string).

Name	In / Out	Type	Default Value	Description
ResultData	Out	Dictionary	{}	Contains the result data after reading all diagnostic trouble codes (DTC).

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

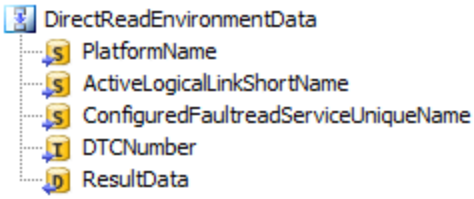
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47

DirectReadEnvironmentData

Graphical representation



Purpose

To read the environment data of a specific diagnostic trouble code.

Description

The ConfiguredFaultreadServiceUniqueName data object is optional. If the value is an empty string, the default element of the collection is taken.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.

Name	In / Out	Type	Default Value	Description
ConfiguredFaultreadServiceUniqueName	In	String	" "	Specifies a unique name of the configured fault read service (can be an empty string).
DTCNumber	In	Integer	0	Specifies the number of the DTC to be read.
ResultData	Out	Dictionary	{}	Contains the result data after reading the diagnostic trouble codes (DTC).

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

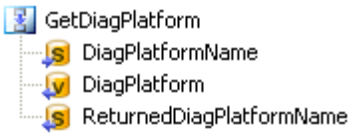
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47

GetDiagPlatform

Graphical representation



Purpose

To get the diagnostics platform.

Description

This block provides direct acccess to the diagnsotics platform object of the current active experiment.

Data object DiagPlatformName is optional. If the value is an empty string, the diagnostics platform found first is returned.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DiagPlatformName	In	String	" "	Specifies the name of the diagnostics platform (can be an empty string).
DiagPlatform	Out	Variant	None	Contains the object of the diagnostics platform.
ReturnedDiagPlatformName	Out	String	" "	Contains the name of the returned diagnostics platform.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

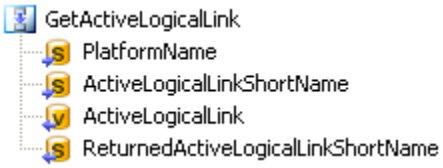
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47
CreateDiagResultDataFromResponses 48
DirectExecuteHexService 56
DirectExecuteService 57
DirectExecuteServiceUsingCustomPDU 59
GetActiveLogicalLink 55

GetActiveLogicalLink

Graphical representation



Purpose

To get the active logical link.

Description

The ActiveLogicalLinkShortName data object is optional. If the value is an empty string, the active logical link found first is returned.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform
ActiveLogicalLinkShortName	In	String	" "	Specifies the name of the active logical link (can be an empty string)
ActiveLogicalLink	Out	Variant	None	Contains the object of the active logical link.
ReturnedActiveLogicalLinkShortName	Out	String	" "	Contains the short name of the returned active logical link.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

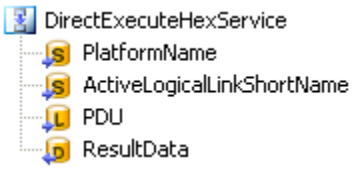
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47
CreateDiagResultDataFromResponses 48
DirectExecuteHexService 56
DirectExecuteService 57
DirectExecuteServiceUsingCustomPDU 59
GetDiagPlatform 54

DirectExecuteHexService

Graphical representation



Purpose

To execute the hex service.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.
PDU	In	List	[]	Specifies the list of the protocol data unit (PDU) data. The list can contain integer values or hex values as a string. For example: [34, 241, 144]| ["0x22", "0xf1", "0x90"]
ResultData	Out	Dictionary	{}	Contains the result data in PDU format after executing the service. This data object should be used as input for the AddDiagResultDataToReport automation block.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

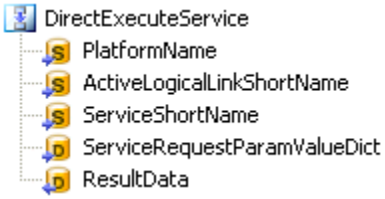
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47
CreateDiagResultDataFromResponses 48
DirectExecuteService 57
DirectExecuteServiceUsingCustomPDU 59
GetActiveLogicalLink 55
GetDiagPlatform 54

DirectExecuteService

Graphical representation



Purpose

To execute the specified service directly.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	" "	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	" "	Specifies the short name of the active logical link.
ServiceShortName	In	String	" "	Specifies the short name of the service.
ServiceRequestParamValueDict	In	Dictionary	{}	Specifies the dictionary of the request parameter to set. Note: If you do not set a value, the request parameter contains the value which was set before. Example: {}| {RequestParameterPath1 : RequestParameterValue1, RequestParameterPath2 : RequestParameterValue2}
ResultData	Out	Dictionary	{}	Contains the result data after executing the service. This data object should be used as input for the AddDiagResultDataToReport automation block.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

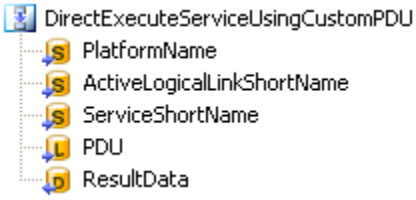
Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47
CreateDiagResultDataFromResponses 48
DirectExecuteHexService 56
DirectExecuteServiceUsingCustomPDU 59
GetActiveLogicalLink 55
GetDiagPlatform 54

DirectExecuteServiceUsingCustomPDU

Graphical representation



Purpose

To execute the specified service using custom Protocol Data Unit (PDU).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
PlatformName	In	String	""	Specifies the name of the diagnostics platform.
ActiveLogicalLinkShortName	In	String	""	Specifies the short name of the active logical link.
ServiceShortName	In	String	""	Specifies the short name of the service.
PDU	In	List	[]	Specifies the list of the PDU data. The list can contain integer values or hex values as a string. For example: [34, 241, 144] or ["0x22", "0xf1", "0x90"]
ResultData	Out	Dictionary	{}	Contains the result data after executing the service. This data object should be used as input for the AddDiagResultDataToReport automation block.

For more information, refer to ECU Diagnostics Handling (ControlDesk ECU Diagnostics 🕮).

Related topics

Examples

Example of a ControlDesk Access Sequence 10

References

AddDiagResultDataToReport 47
CreateDiagResultDataFromResponses 48
DirectExecuteHexService 56
DirectExecuteService 57
GetActiveLogicalLink 55
GetDiagPlatform 54

Basic Functions

Where to go from here

Information in this section

Common....60
Diagnostic....63

Common

Where to go from here

Information in this section

ActiveExperiment....60
To provide access to the currently active experiment in ControlDesk.

Application....61
To provide methods and properties of the Application object of ControlDesk.

Platform (Common)....61
To provide access to a platform in ControlDesk.

Platforms (Common)....62
To provide access to all platforms in ControlDesk.

ActiveExperiment

Purpose

To provide access to the currently active experiment in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment
```

Description

The ActiveExperiment object can be created as a Variant data object by using the **GetActiveExperiment** automation block, refer to Application on page 61.

The following automation blocks use the instantiated **ActiveExperiment** data object:

Automation Block	Description
GetPlatforms	To get the collection of platforms in ControlDesk. The block instantiates the **Platforms** data object (see Platforms (Common) on page 62).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Application

Purpose

To provide methods and properties of the Application object of ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application
```

Description

The Application object can be created as Variant data object by using the **GetApplication** automation block which gets the Application object from ControlDesk.

The following automation blocks use the instantiated Application data object:

Automation Block	Description
GetActiveExperiment	To get the currently active experiment in ControlDesk. If no active experiment is available, null is returned, otherwise the **ActiveExperiment** data object is instantiated.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Platform (Common)

Purpose

To provide access to a platform in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform
```

Description

The Platform object can be created as a Variant data object by using the **GetPlatformByIndex** or the **GetPlatformByName** automation block, refer to Platforms (Common) on page 62.

The following automation blocks use the instantiated **Platform** data object:

Automation Block	Description
GetPlatformType	To get the type of the platform. The block instantiates the **PlatformType** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Platforms (Common)

Purpose

To provide access to all platforms in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms
```

Description

The Platforms object can be created as a Variant data object by using the **GetPlatforms** automation block, refer to ActiveExperiment on page 60.

The following automation blocks use the instantiated **Platforms** data object:

Automation Block	Description
CountPlatforms	To get the number of platforms. The block instantiates the **NoOfPlatforms** data object.
GetPlatformByIndex	To get the platform object by its index. The block instantiates the **Platform** data object (see Platform (Diagnostics) on page 86).
GetPlatformByName	To get the platform object by its name. The block instantiates the **Platform** data object (see Platform (Diagnostics) on page 86).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Diagnostic

Where to go from here

Information in this section

ActiveDiagnosticsDatabase....66
To provide access to the currently active diagnostics database in ControlDesk.

ActiveLogicalLink....66
To provide access to the currently active logical link in ControlDesk.

ActiveLogicalLinks....67
To provide access to all active logical links in ControlDesk.

ControlPrimitive....68
To provide access to a control primitive in ControlDesk.

ControlPrimitives....69
To provide access to all control primitives in ControlDesk.

CtrlPrimitiveRequestParameter....69
To provide access to a control primitive request parameter in ControlDesk.

CtrlPrimitiveRequestParameters....71
To provide access to all control primitive request parameters in ControlDesk.

CtrlPrimitiveResponse....71
To provide access to control primitive response in ControlDesk.

CtrlPrimitiveResponseParameter....72
To provide access to a control primitive response parameter in ControlDesk.

CtrlPrimitiveResponseParameters....73
To provide access to control primitive response parameters in ControlDesk.

CtrlPrimitiveResponseRequestParameter....74
To provide access to the corresponding control primitive request parameter of a response object in ControlDesk.

CtrlPrimitiveResponseRequestParameters....75
To provide access to corresponding control primitive request parameters of a response object in ControlDesk.

CtrlPrimitiveResponses....76
To provide access to control primitive responses in ControlDesk.

Job....77
To provide access to a diagnostics job in ControlDesk.

JobRequestParameter....77
To provide access to a diagnostics job request parameter in ControlDesk.

JobRequestParameters..... 79
To provide access to diagnostics job request parameters in ControlDesk.

JobResponse..... 79
To provide access to a diagnostics job response in ControlDesk.

JobResponseParameter..... 80
To provide access to a diagnostics job response parameter in ControlDesk.

JobResponseParameters..... 81
To provide access to diagnostics job response parameters in ControlDesk.

JobResponseRequestParameter..... 82
To provide access to the corresponding diagnostics job request parameter of a response object in ControlDesk.

JobResponseRequestParameters..... 83
To provide access to corresponding diagnostics job request parameters of a response object in ControlDesk.

JobResponses..... 84
To provide access to diagnostics job responses in ControlDesk.

Jobs..... 84
To provide access to all diagnostics jobs in ControlDesk.

LogicalLinks..... 85
To provide access to all logical links in ControlDesk.

LogicalLinkSelection..... 86
To provide access to locigal link selection in ControlDesk.

Platform (Diagnostics)..... 86
To provide access to a platform in ControlDesk.

Platforms (Diagnostics)..... 87
To provide access to all platforms in ControlDesk.

PPSetRequestParameter..... 88
To provide access to the request parameter of the specific control primitive ProtocolParameterSet (COMPARAM) in ControlDesk.

PPSetRequestParameters..... 89
To provide access to a collection of protocol parameter set request parameters (COMPARAMs)in ControlDesk.

PPSetResponse..... 90
To provide access to protocol parameter set response in ControlDesk.

PPSetResponseParameter..... 91
To provide access to a protocol parameter set response parameter in ControlDesk.

PPSetResponseParameters..... 92
To provide access to a collection of protocol parameter set response parameters in ControlDesk.

PPSetResponseRequestParameter....92
To provide access to the corresponding protocol parameter set request parameter of a response object in ControlDesk.

PPSetResponseRequestParameters....93
To provide access to corresponding protocol parameters set request parameters of a response object in ControlDesk.

ProtocolParameterSet....94
To provide access to protocol parameter set in ControlDesk.

SelectedVehicle....95
To provide access to a selected vehicle in ControlDesk.

Service....96
To provide access to a service in ControlDesk.

ServiceRequestParameter....96
To provide access to a service request parameter in ControlDesk.

ServiceRequestParameters....98
To provide access to all service request parameters in ControlDesk.

ServiceResponse....99
To provide access to a service response in ControlDesk.

ServiceResponseParameter....99
To provide access to a service response parameter in ControlDesk.

ServiceResponseParameters....101
To provide access to service response parameters in ControlDesk.

ServiceResponseRequestParameter....101
To provide access to the corresponding service request parameter of a response object in ControlDesk.

ServiceResponseRequestParameters....102
To provide access to corresponding service request parameters of a response object in ControlDesk.

ServiceResponses....103
To provide access to all service responses in ControlDesk.

Services....104
To provide access to all diagnostic services in ControlDesk.

Vehicle....105
To provide access to a vehicle in ControlDesk.

Vehicles....105
To provide access to all vehicles in ControlDesk.

VehicleSelection....106
To provide access to the vehicle selection in ControlDesk.

ActiveDiagnosticsDatabase

Purpose

To provide access to the currently active diagnostics database in ControlDesk.

Path

The following shows the path of the Basic Functions in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase
```

Description

The ActiveDiagnosticsDatabase object can be created as a Variant data object by using the GetActiveDiagnosticsDatabase automation block, refer to Platform (Diagnostics) on page 86.

The following automation blocks use the instantiated ActiveDiagnosticsDatabase data object:

Automation Block	Description
GetVehicleSelection	To get the vehicle selection. The block instantiates the VehicleSelection data object (see VehicleSelection on page 106).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ActiveLogicalLink

Purpose

To provide access to the currently active logical link in ControlDesk.

Path

The following shows the path of the Basic Functions in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink
```

Description

The ActiveLogicalLink object can be created as a Variant data object by using the GetActiveLogicalLinkByIndex or the GetActiveLogicalLinkByName automation block, refer to ActiveLogicalLinks on page 67.

The following automation blocks use the instantiated **ActiveLogicalLink** data object:

Automation Block	Description
ExecuteHexService	To execute a hex service with a specified protocol data unit (PDU). The block instantiates the **ServiceResponses** data object (see ServiceResponses on page 103).
GetCalibrationStateOfActiveLL	To get the calibration state of the active logical link. The block instantiates the **CalibrationState** data object.
GetConnectionStateOfActiveLL	To get the connection state of the active logical link. The block instantiates the **ConnectionState** data object.
GetControlPrimitives	To get the control primitives collection object. The block instantiates the **ControlPrimitives** data object (see ControlPrimitives on page 69).
GetLongNameOfActiveLL	To get the long name of the active logical link. The block instantiates the **LongName** data object.
GetProtocolParameterSet	To get the protocol parameter set object. The block instantiates the **ProtocolParameterSet** data object (see ProtocolParameterSet on page 94).
GetServices	To get the services collection object. The block instantiates the **Services** data object (see Services on page 104).
GetShortNameOfActiveLL	To get the short name of the active logical link. The block instantiates the **ShortName** data object.
GetSingleECUJobs	To get the single ECU jobs collection object. The block instantiates the **Jobs** data object (see Jobs on page 84).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ActiveLogicalLinks

Purpose

To provide access to all active logical links in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks
```

Description

The ActiveLogicalLinks object can be created as a Variant data object by using the **GetActiveLogicalLink** automation block, refer to LogicalLinkSelection on page 86.

The following automation blocks use the instantiated **ActiveLogicalLinks** data object:

Automation Block	Description
CountActiveLogicalLinks	To get the number of active logical links in the collection. The block instantiates the **NoOfActiveLogicalLinks** data object.
GetActiveLogicalLinkByIndex	To get the active logical link object by its index. The block instantiates the **ActiveLogicalLink** data object (see ActiveLogicalLink on page 66).
GetActiveLogicalLinkByName	To get the active logical link object by its name. The block instantiates the **ActiveLogicalLink** data object (see ActiveLogicalLink on page 66).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ControlPrimitive

Purpose

To provide access to a control primitive in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive
```

Description

The ControlPrimitive object can be created as a Variant data object by using the **GetControlPrimitiveByIndex** or the **GetControlPrimitiveByName** automation block, refer to ControlPrimitives on page 69.

The following automation blocks use the instantiated **ControlPrimitive** data object:

Automation Block	Description
ExecuteControlPrimitive	To execute a control primitive. The block instantiates the **CtrlPrimitiveResponses** data object (see CtrlPrimitiveResponses on page 76).
GetCtrlPrimitiveRequestParameters	To get the control primitive request parameters collection object. The block instantiates the **CtrlPrimitiveRequestParameters** data object (see CtrlPrimitiveRequestParameters on page 71).
GetLongNameOfControlPrimitive	To get the long name of the control primitive. The block instantiates the **LongName** data object.
GetShortNameOfControlPrimitive	To get the short name of the control primitive. The block instantiates the **ShortName** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ControlPrimitives

Purpose

To provide access to all control primitives in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives
```

Description

The ControlPrimitives object can be created as a Variant data object by using the **GetControlPrimitives** automation block, refer to ActiveLogicalLink on page 66.

The following automation blocks use the instantiated **ControlPrimitives** data object:

Automation Block	Description
CountControlPrimitives	To get the number of control primitives. The block instantiates the **NoOfControlPrimitives** data object.
GetControlPrimitiveByIndex	To get the control primitive object by its index. The block instantiates the **ControlPrimitive** data object (see ControlPrimitive on page 68).
GetControlPrimitiveByName	To get the control primitive object by its name. The block instantiates the **ControlPrimitive** data object (see ControlPrimitive on page 68).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveRequestParameter

Purpose

To provide access to a control primitive request parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\
CtrlPrimitiveRequestParameters\CtrlPrimitiveRequestParameter
```

Description

The CtrlPrimitiveRequestParameter object can be created as a Variant data object by using the **GetCtrlPrimitiveRequestParameterByIndex** or the **GetCtrlPrimitiveRequestParameterByPath** automation block, refer to CtrlPrimitiveRequestParameters on page 71.

The following automation blocks use the instantiated **CtrlPrimitiveRequestParameter** data object:

Automation Block	Description
GetDefaultValueOfCtrlPrimitiveRequestParameter	To get the default value of the control primitive request parameter. The block instantiates the **DefaultValue** data object.
GetLongNameOfCtrlPrimitiveRequestParameter	To get the long name of the control primitive request parameter. The block instantiates the **LongName** data object.
GetPathOfCtrlPrimitiveRequestParameter	To get the path of the control primitive request parameter. The block instantiates the **Path** data object.
GetSemanticOfCtrlPrimitiveRequestParameter	To get the semantics of the control primitive request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfCtrlPrimitiveRequestParameter	To get the short name of the control primitive request parameter. The block instantiates the **ShortName** data object.
GetTypeOfCtrlPrimitiveRequestParameter	To get the type of the control primitive request parameter. The block instantiates the **Type** data object.
GetUnitOfCtrlPrimitiveRequestParameter	To get the unit of the control primitive request parameter. The block instantiates the **Unit** data object.
GetValueOfCtrlPrimitiveRequestParameter	To get the value of the control primitive request parameter. The block instantiates the **Value** data object.
GetValueRestrictionsOfCtrlPrimitiveRequestParameter	To get the value restrictions of the control primitive request parameter. The block instantiates the **ValueRestrictions** data object.
IsCtrlPrimitiveRequestParameterReadOnly	To check whether the control primitive request parameter is read-only. The block instantiates the **IsReadOnly** data object.
SetValueOfCtrlPrimitiveRequestParameter	To set the value of the control primitive request parameter.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveRequestParameters

Purpose

To provide access to all control primitive request parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\
CtrlPrimitiveRequestParameters
```

Description

The CtrlPrimitiveRequestParameters object can be created as a Variant data object by using the **GetCtrlPrimitiveRequestParameters** automation block, refer to ControlPrimitive on page 68.

The following automation blocks use the instantiated **CtrlPrimitiveRequestParameters** data object:

Automation Block	Description
CountCtrlPrimitiveRequestParameters	To get the number of the control primitive request parameters in the collection. The block instantiates the **NoOfCtrlPrimitiveRequestParameters** data object.
GetCtrlPrimitiveRequestParameterByIndex	To get the control primitive request parameter object by its index. The block instantiates the **CtrlPrimitiveRequestParameter** data object (see CtrlPrimitiveRequestParameter on page 69).
GetCtrlPrimitiveRequestParameterByPath	To get the control primitive request parameter object by its path. The block instantiates the **CtrlPrimitiveRequestParameter** data object (see CtrlPrimitiveRequestParameter on page 69).
ResetCtrlPrimitiveRequestParametersToDefault	To reset the control primitive request parameters to default.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveResponse

Purpose

To provide access to control primitive response in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\CtrlPrimitiveResponses\
CtrlPrimitiveResponse
```

Description

The CtrlPrimitiveResponse object can be created as a Variant data object by using the **GetCtrlPrimitiveResponseByIndex** automation block, refer to CtrlPrimitiveResponses on page 76.

The following automation blocks use the instantiated **CtrlPrimitiveResponse** data object:

Automation Block	Description
GetCtrlPrimitiveResponseParameters	To get the control primitive response parameters collection object. The block instantiates the **CtrlPrimitiveResponseParameters** data object (see CtrlPrimitiveResponseParameters on page 73).
GetCtrlPrimitiveResponseRequestParameters	To get the control primitive response request parameters collection object. The block instantiates the **CtrlPrimitiveResponseRequestParameters** data object (see CtrlPrimitiveResponseRequestParameters on page 75).
GetCtrlPrimitiveResponseTimeStamp	To get the time stamp of the control primitive response. The block instantiates the **ResponseTimeStamp** data object.
GetCtrlPrimitiveResponseType	To get the type of the control primitive response. The block instantiates the **ResponseType** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveResponseParameter

Purpose

To provide access to a control primitive response parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\CtrlPrimitiveResponses\
CtrlPrimitiveResponse\CtrlPrimitiveResponseParameters\
CtrlPrimitiveResponseParameter
```

Description

The CtrlPrimitiveResponseParameter object can be created as a Variant data object by using the **GetCtrlPrimitiveResponseParameterByIndex** or the **GetCtrlPrimitiveResponseParameterByPath** automation block, refer to CtrlPrimitiveResponseParameters on page 73.

The following automation blocks use the instantiated **CtrlPrimitiveResponseParameter** data object:

Automation Block	Description
GetLongNameOfCtrlPrimitiveResponseParameter	To get the long name of the control primitive response parameter. The block instantiates the **LongName** data object.
GetPathOfCtrlPrimitiveResponseParameter	To get the path of the control primitive response parameter. The block instantiates the **Path** data object.
GetSemanticOfCtrlPrimitiveResponseParameter	To get the semantics of the control primitive response parameter. The block instantiates the **Semantics** data object.
GetShortNameOfCtrlPrimitiveResponseParameter	To get the short name of the control primitive response parameter. The block instantiates the **ShortName** data object.
GetTypeOfCtrlPrimitiveResponseParameter	To get the type of the control primitive response parameter. The block instantiates the **Type** data object.
GetUnitOfCtrlPrimitiveResponseParameter	To get the unit of the control primitive response parameter. The block instantiates the **Unit** data object.
GetValueOfCtrlPrimitiveResponseParameter	To get the value of the control primitive response parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveResponseParameters

Purpose

To provide access to control primitive response parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\CtrlPrimitiveResponses\
CtrlPrimitiveResponse\CtrlPrimitiveResponseParameters
```

Description

The CtrlPrimitiveResponseParameters object can be created as a Variant data object by using the **GetCtrlPrimitiveResponseParameters** automation block, refer to CtrlPrimitiveResponse on page 71.

The following automation blocks use the instantiated **CtrlPrimitiveResponseParameters** data object:

Automation Block	Description
CountCtrlPrimitiveResponseParameters	To get the number of control primitive response parameters. The block instantiates the **NoOfCtrlPrimitiveResponseParameters** data object.
GetCtrlPrimitiveResponseParameterByIndex	To get the control primitive response parameter object by its index. The block instantiates the **CtrlPrimitiveResponseParameter** data object (see CtrlPrimitiveResponseParameter on page 72).
GetCtrlPrimitiveResponseParameterByPath	To get the control primitive response parameter object by its path. The block instantiates the **CtrlPrimitiveResponseParameter** data object (see CtrlPrimitiveResponseParameter on page 72).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveResponseRequestParameter

Purpose

To provide access to the corresponding control primitive request parameter of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\CtrlPrimitiveResponses\
CtrlPrimitiveResponse\CtrlPrimitiveResponseRequestParameters\
CtrlPrimitiveResponseRequestParameter
```

Description

The CtrlPrimitiveResponseRequestParameter object can be created as a Variant data object by using the **GetCtrlPrimitiveResponseRequestParameterByIndex** or the **GetCtrlPrimitiveResponseRequestParameterByPath** automation block, refer to CtrlPrimitiveResponseRequestParameters on page 75.

The following automation blocks use the instantiated **CtrlPrimitiveResponseRequestParameter** data object:

Automation Block	Description
GetLongNameOfCtrlPrimitiveResponseRequestParameter	To get the long name of the control primitive response request parameter. The block instantiates the **LongName** data object.

Automation Block	Description
GetPathOfCtrlPrimitiveResponseRequestParameter	To get the path of the control primitive response request parameter. The block instantiates the **Path** data object.
GetSemanticOfCtrlPrimitiveResponseRequestParameter	To get the semantics of the control primitive response request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfCtrlPrimitiveResponseRequestParameter	To get the short name of the control primitive response request parameter. The block instantiates the **ShortName** data object.
GetTypeOfCtrlPrimitiveResponseRequestParameter	To get the type of the control primitive response request parameter. The block instantiates the **Type** data object.
GetUnitOfCtrlPrimitiveResponseRequestParameter	To get the unit of the control primitive response request parameter. The block instantiates the **Unit** data object.
GetValueOfCtrlPrimitiveResponseRequestParameter	To get the value of the control primitive response request parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveResponseRequestParameters

Purpose

To provide access to corresponding control primitive request parameters of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\CtrlPrimitiveResponses\
CtrlPrimitiveResponse\CtrlPrimitiveResponseRequestParameters
```

Description

The CtrlPrimitiveResponseRequestParameters object can be created as a Variant data object by using the **GetCtrlPrimitiveResponseRequestParameters** automation block, refer to CtrlPrimitiveResponse on page 71.

The following automation blocks use the instantiated **CtrlPrimitiveResponseRequestParameters** data object:

Automation Block	Description
GetCtrlPrimitiveResponseRequestParameterByIndex	To get the control primitive response request parameter object by its index. The block instantiates the

Automation Block	Description
GetCtrlPrimitiveResponseRequestParameterByPath	CtrlPrimitiveResponseRequestParameter data object (see CtrlPrimitiveResponseRequestParameter on page 74). To get the control primitive response request parameter object by its path. The block instantiates the CtrlPrimitiveResponseRequestParameter data object (see CtrlPrimitiveResponseRequestParameter on page 74).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

CtrlPrimitiveResponses

Purpose

To provide access to control primitive responses in ControlDesk.

Path

The following shows the path of the Basic Functions in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ControlPrimitives\ControlPrimitive\CtrlPrimitiveResponses
```

Description

The CtrlPrimitiveResponses object can be created as a Variant data object by using the ExecuteControlPrimitive automation block, refer to ControlPrimitive on page 68.

The following automation blocks use the instantiated CtrlPrimitiveResponses data object:

Automation Block	Description
CountCtrlPrimitiveResponses	To get the number of the control primitve responses. The block instantiates the NoOfCtrlPrimitiveResponses data object.
GetCtrlPrimitiveNameFromCtrlPrimitiveResponses	To get the name of the control primitive. The block instantiates the ControlPrimitiveName data object.
GetCtrlPrimitiveResponseByIndex	To get the control primitive response by its index. The block instantiates the CtrlPrimitiveResponse data object (see CtrlPrimitiveResponse on page 71).
GetExecutionTimeStampFromCtrlPrimitiveResponses	To get the execution time stamp of the control primitive. The block instantiates the ExecutionTimeStamp data object.
GetLogicalLinkNameFromCtrlPrimitiveResponses	To get the name of the logical link. The block instantiates the LogicalLinkName data object.

Automation Block	Description
GetRequestStatusFromCtrlPrimitiveResponses	To get the request status from control primitive responses. The block instantiates the **RequestStatus** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Job

Purpose

To provide access to a diagnostics job in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job
```

Description

The Job object can be created as a Variant data object by using the **GetJobByIndex** or the **GetJobByName** automation block, refer to Jobs on page 84.

The following automation blocks use the instantiated **Job** data object:

Automation Block	Description
ExecuteJob	To execute a job. The block instantiates the **JobResponses** data object.
GetJobRequestParameters	To get the request parameters collection object. The block instantiates the **JobRequestParameters** data object (see JobRequestParameters on page 79).
GetLongNameofJob	To get the long name of job. The block instantiates the **LongName** data object.
GetShortNameofJob	To get the short name of job. The block instantiates the **ShortName** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobRequestParameter

Purpose

To provide access to a diagnostics job request parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobRequestParameters\JobRequestParameter
```

Description

The JobRequestParameter object can be created as a Variant data object by using the **GetJobRequestParameterByIndex** or the **GetJobRequestParameterByPath** automation block, refer to JobRequestParameters on page 79.

The following automation blocks use the instantiated **JobRequestParameter** data object:

Automation Block	Description
GetDefaultValueOfJobRequestParameter	To get the default value of the job request parameter. The block instantiates the **DefaultValue** data object.
GetLongNameOfJobRequestParameter	To get the long name of the job request parameter. The block instantiates the **LongName** data object.
GetPathOfJobRequestParameter	To get the path of the job request parameter. The block instantiates the **Path** data object.
GetSemanticOfJobRequestParameter	To get the semantics of the job request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfJobRequestParameter	To get the short name of the job request parameter. The block instantiates the **ShortName** data object.
GetTypeOfJobRequestParameter	To get the type of the job request parameter. The block instantiates the **Type** data object.
GetUnitOfJobRequestParameter	To get the unit of the job request parameter. The block instantiates the **Unit** data object.
GetValueOfJobRequestParameter	To get the value of the job request parameter. The block instantiates the **Value** data object.
GetValueRestrictionsOfJobRequestParameter	To get the value restrictions of the job request parameter. The block instantiates the **ValueRestrictions** data object.
IsJobRequestParameterReadOnly	To check whether the job request parameter is read-only. The block instantiates the **IsReadOnly** data object.
SetValueOfJobRequestParameter	To set the value of the job request parameter.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobRequestParameters

Purpose

To provide access to diagnostics job request parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobRequestParameters
```

Description

The JobRequestParameters object can be created as a Variant data object by using the **GetJobRequestParameters** automation block, refer to Job on page 77.

The following automation blocks use the instantiated **JobRequestParameters** data object:

Automation Block	Description
CountJobRequestParameters	To get the number of job request parameters in the collection. The block instantiates the **NoOfJobRequestParameters** data object.
GetJobRequestParameterByIndex	To get the request parameter object by its index. The block instantiates the **JobRequestParameter** data object (see JobRequestParameter on page 77).
GetJobRequestParameterByPath	To get the request parameter object by its path. The block instantiates the **JobRequestParameter** data object (see JobRequestParameter on page 77).
ResetJobRequestParametersToDefault	To reset the job request parameters to default.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobResponse

Purpose

To provide access to a diagnostics job response in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobResponses\JobResponse
```

Description

The JobResponse object can be created as a Variant data object by using the **GetJobResponseByIndex** automation block, refer to JobResponses on page 84.

The following automation blocks use the instantiated **JobResponse** data object:

Automation Block	Description
GetJobResponseParameters	To get the job response parameters collection object. The block instantiates the **JobResponseParameters** data object (see JobResponseParameters on page 81).
GetJobResponseRequestParameters	To get the job response request parameters collection object. The block instantiates the **JobResponseRequestParameters** data object (see JobResponseRequestParameters on page 83).
GetJobResponseTimeStamp	To get the time stamp of the job response. The block instantiates the **ResponseTimeStamp** data object.
GetJobResponseType	To get the type of the job response. The block instantiates the **ResponseType** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobResponseParameter

Purpose

To provide access to a diagnostics job response parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobResponses\JobResponse\JobResponseParameters\
JobResponseParameter
```

Description

The JobResponseParameter object can be created as a Variant data object by using the **GetJobResponseParameterByIndex** or the **GetJobResponseParameterByPath** automation block, refer to JobResponseParameters on page 81.

The following automation blocks use the instantiated **JobResponseParameter** data object:

Automation Block	Description
GetLongNameOfJobResponseParameter	To get the long name of a job response parameter. The block instantiates the **LongName** data object.
GetPathOfJobResponseParameter	To get the path of a job response parameter. The block instantiates the **Path** data object.
GetSemanticOfJobResponseParameter	To get the semantics of a job response parameter. The block instantiates the **Semantics** data object.
GetShortNameOfJobResponseParameter	To get the short name of a job response parameter. The block instantiates the **ShortName** data object.
GetTypeOfJobResponseParameter	To get the type of a job response parameter. The block instantiates the **Type** data object.
GetUnitOfJobResponseParameter	To get the unit of a job response parameter. The block instantiates the **Unit** data object.
GetValueOfJobResponseParameter	To get the value of a job response parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobResponseParameters

Purpose

To provide access to diagnostics job response parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobResponses\JobResponse\JobResponseParameters
```

Description

The JobResponseParameters object can be created as a Variant data object by using the **GetJobResponseParameters** automation block, refer to JobResponse on page 79.

The following automation blocks use the instantiated **JobResponseParameters** data object:

Automation Block	Description
CountJobResponseParameters	To get the number of the job responses parameters. The block instantiates the **NoOfJobResponseParameters** data object.
GetJobResponseParameterByIndex	To get the job response parameter object by its index. The block instantiates the **JobResponseParameter** data object (see JobResponseParameter on page 80).
GetJobResponseParameterByPath	To get the job response parameter object by its path. The block instantiates the **JobResponseParameter** data object (see JobResponseParameter on page 80).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobResponseRequestParameter

Purpose

To provide access to the corresponding diagnostics job request parameter of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobResponses\JobResponse\JobResponseRequestParameters\
JobResponseRequestParameter
```

Description

The JobResponseRequestParameter object can be created as a Variant data object by using the **GetJobResponseRequestParameterByIndex** or the **GetJobResponseRequestParameterByPath** automation block, refer to JobResponseRequestParameters on page 83.

The following automation blocks use the instantiated **JobResponseRequestParameter** data object:

Automation Block	Description
GetLongNameOfJobResponseRequestParameter	To get the long name of a job response request parameter. The block instantiates the **LongName** data object.
GetPathOfJobResponseRequestParameter	To get the path of a job response request parameter. The block instantiates the **Path** data object.
GetSemanticOfJobResponseRequestParameter	To get the semantics of a job response request parameter. The block instantiates the **Semantics** data object.

Automation Block	Description
GetShortNameOfJobResponseRequestParameter	To get the short name of a job response request parameter. The block instantiates the **ShortName** data object.
GetTypeOfJobResponseRequestParameter	To get the type of a job response request parameter. The block instantiates the **Type** data object.
GetUnitOfJobResponseRequestParameter	To get the unit of a job response request parameter. The block instantiates the **Unit** data object.
GetValueOfJobResponseRequestParameter	To get the value of a job response request parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobResponseRequestParameters

Purpose

To provide access to corresponding diagnostics job request parameters of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobResponses\JobResponse\JobResponseRequestParameters
```

Description

The JobResponseRequestParameters object can be created as a Variant data object by using the **GetJobResponseRequestParameters** automation block, refer to JobResponse on page 79.

The following automation blocks use the instantiated **JobResponseRequestParameters** data object:

Automation Block	Description
GetJobResponseRequestParameterByIndex	To get the job response request parameter object by its index. The block instantiates the **JobResponseRequestParameter** data object (see JobResponseRequestParameter on page 82).
GetJobResponseRequestParameterByPath	To get the job response request parameter object by its name. The block instantiates the **JobResponseRequestParameter** data object (see JobResponseRequestParameter on page 82).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

JobResponses

Purpose

To provide access to diagnostics job responses in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs\
Job\JobResponses
```

Description

The JobResponses object can be created as a Variant data object by using the **ExecuteJob** automation block, refer to Job on page 77.

The following automation blocks use the instantiated **JobResponses** data object:

Automation Block	Description
CountJobResponses	To get the number of job responses. The block instantiates the **NoOfJobResponses** data object.
GetExecutionTimeStampFromJobResponses	To get the execution time stamp from job responses. The block instantiates the **ExecutionTimeStamp** data object.
GetJobNameFromJobResponses	To get the name of the job. The block instantiates the **JobName** data object.
GetJobResponseByIndex	To get the job response by its index. The block instantiates the **JobResponse** data object (see JobResponse on page 79).
GetLogicalLinkNameFromJobResponses	To get the name of the job. The block instantiates the **LogicalLinkName** data object.
GetRequestStatusFromJobResponses	To get the status of the request. The block instantiates the **RequestStatus** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Jobs

Purpose

To provide access to all diagnostics jobs in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\Jobs
```

Description

The Jobs object can be created as a Variant data object by using the **GetSingleECUJobs** automation block, refer to ActiveLogicalLink on page 66.

The following automation blocks use the instantiated **Jobs** data object:

Automation Block	Description
CountSingleEcuJobs	To get the number of single ECU jobs. The block instantiates the **NoOfSingleECUJobs** data object.
GetJobByIndex	To get the job object by its index. The block instantiates the **Job** data object (see Job on page 77).
GetJobByName	To get the job object by its name. The block instantiates the **Job** data object (see Job on page 77).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

LogicalLinks

Purpose

To provide access to all logical links in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\LogicalLinks
```

Description

The LogicalLinks object can be created as a Variant data object by using the **GetLogicalLinks** automation block, refer to LogicalLinkSelection on page 86.

The following automation blocks use the instantiated **LogicalLinks** data object:

Automation Block	Description
CountLogicalLinks	To get the number of logical links. The block instantiates the **NoOfLogicalLinks** data object.
GetLogicalLinkByIndex	To get the logical link object by its index. The block instantiates the **LogicalLink** data object (see LogicalLinkSelection on page 86).
GetLogicalLinkByName	To get the logical link object by its name. The block instantiates the **LogicalLink** data object (see LogicalLinkSelection on page 86).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

LogicalLinkSelection

Purpose

To provide access to locigal link selection in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection
```

Description

The LogicalLinkSelection object can be created as a Variant data object by using the **GetLogicalLinkSelection** automation block, refer to SelectedVehicle on page 95.

The following automation blocks use the instantiated **LogicalLinkSelection** data object:

Automation Block	Description
GetActiveLogicalLinks	To get the active logical links collection object. The block instantiates the **ActiveLogicalLinks** data object (see ActiveLogicalLinks on page 67).
GetLogicalLinks	To get the logical links collection object. The block instantiates the **LogicalLinks** data object (see LogicalLinks on page 85).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Platform (Diagnostics)

Purpose

To provide access to a platform in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform
```

Description

The Platform object can be created as a Variant data object by using the **GetPlatformByIndex** or the **GetPlatformByName** automation block, refer to Platforms (Common) on page 62.

The following automation blocks use the instantiated **Platform** data object:

Automation Block	Description
GetActiveDiagnosticsDatabase	To get the active diagnostics database. The block instantiates the **ActiveDiagnosticsDatabase** data object (see ActiveDiagnosticsDatabase on page 66).
SC_GetActiveLogicalLinks	To get the active logical links by shortcut. The block instantiates the **ActiveLogicalLinks** data object (see ActiveLogicalLinks on page 67).
SC_GetLogicalLinkSelection	To get the logical link selection by shortcut. The block instantiates the **LogicalLinkSelection** data object (see LogicalLinkSelection on page 86).
SC_GetVehicleSelection	To get the vehicle selection by shortcut. The block instantiates the **VehicleSelection** data object (see VehicleSelection on page 106).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Platforms (Diagnostics)

Purpose

To provide access to all platforms in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms
```

Description

The Platforms object can be created as a Variant data object by using the **GetPlatforms** automation block, refer to ActiveExperiment on page 60.

The following automation blocks use the instantiated **Platforms** data object:

Automation Block	Description
CountPlatforms	To get the number of platforms. The block instantiates the **NoOfPlatforms** data object.
GetPlatformByIndex	To get the platform object by its index. The block instantiates the **Platform** data object (see Platform (Diagnostics) on page 86).
GetPlatformByName	To get the platform object by its name. The block instantiates the **Platform** data object (see Platform (Diagnostics) on page 86).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetRequestParameter

Purpose

To provide access to the request parameter of the specific control primitive ProtocolParameterSet(COMPARAM) in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetRequestParameters\
PPSetRequestParameter
```

Description

The PPSetRequestParameter object can be created as a Variant data object by using the **GetPPSetRequestParameterByIndex** or the **GetPPSetRequestParameterByPath** automation block, refer to PPSetRequestParameters on page 89.

The following automation blocks use the instantiated **PPSetRequestParameter** data object:

Automation Block	Description
GetDefaultValueOfPPSetRequestParameter	To get the default value of the protocol parameter set request parameter. The block instantiates the **DefaultValue** data object.
GetLongNameOfPPSetRequestParameter	To get the long name of the protocol parameter set request parameter. The block instantiates the **LongName** data object.
GetPathOfPPSetRequestParameter	To get the path of the protocol parameter set request parameter. The block instantiates the **Path** data object.
GetSemanticOfPPSetRequestParameter	To get the semantics of the protocol parameter set request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfPPSetRequestParameter	To get the short name of the protocol parameter set request parameter. The block instantiates the **ShortName** data object.
GetTypeOfPPSetRequestParameter	To get the type of the protocol parameter set request parameter. The block instantiates the **Type** data object.
GetUnitOfPPSetRequestParameter	To get the unit of the protocol parameter set request parameter. The block instantiates the **Unit** data object.
GetValueOfPPSetRequestParameter	To get the value of the protocol parameter set request parameter. The block instantiates the **Value** data object.
GetValueRestrictionsOfPPSetRequestParameter	To get the value restrictions of the protocol parameter set request parameter. The block instantiates the **ValueRestrictions** data object.

Automation Block	Description
IsPPSetRequestParameterReadOnly	To check whether the protocol parameter set request parameter is read-only. The block instantiates the **IsReadOnly** data object.
SetValueOfPPSetRequestParameter	To set the value of the protocol parameter set request parameter.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetRequestParameters

Purpose

To provide access to a collection of protocol parameter set request parameters (COMPARAMs) in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetRequestParameters
```

Description

The PPSetRequestParameters object can be created as a Variant data object by using the **GetPPSetRequestParameters** automation block, refer to ProtocolParameterSet on page 94.

The following automation blocks use the instantiated **PPSetRequestParameters** data object:

Automation Block	Description
CountPPSetRequestParameters	To get the number of the protocol parameter set request parameters. The block instantiates the **NoOfPPSetRequestParameters** data object.
GetPPSetRequestParameterByIndex	To get the protocol parameter set request parameter by its index. The block instantiates the **PPSetRequestParameter** data object (see PPSetRequestParameter on page 88).
GetPPSetRequestParameterByPath	To get the protocol parameter set request parameter by its path. The block instantiates the **PPSetRequestParameter** data object (see PPSetRequestParameter on page 88).
ResetPPSetRequestParametersToDefault	To reset the protocol parameter set request parameters to default.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetResponse

Purpose

To provide access to protocol parameter set response in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetResponse
```

Description

The PPSetResponse object can be created as a Variant data object by using the **UpdateProtocolParameterSet** automation block (in this block: ProtocolParameterSetResponse data object), refer to ProtocolParameterSet on page 94.

The following automation blocks use the instantiated **PPSetResponse** data object:

Automation Block	Description
GetExecutionTimeStampOfPPSetResponse	To get the excution time stamp. The block instantiates the **ExecutionTimeStamp** data object.
GetLogicalLinkNameFromPPSetResponse	To get the logical link name. The block instantiates the **LogicalLinkName** data object.
GetPPSetResponseParameters	To get the protocol parameter set response parameters collection. The block instantiates the **PPSetResponseParameters** data object (see PPSetResponseParameters on page 92).
GetPPSetResponseRequestParameters	To get the protocol parameter set response request parameters collection. The block instantiates the **PPSetResponseRequestParameters** data object (see PPSetResponseRequestParameters on page 93).
GetPPSetResponseTimeStamp	To get the time stamp of the protocol parameter set response. The block instantiates the **ResponseTimeStamp** data object.
GetPPSetResponseType	To get the type of the protocol parameter set response. The block instantiates the **ResponseType** data object.
GetRequestStatusFromPPSetResponse	To get the request status from the protocol parameter set response. The block instantiates the **RequestStatus** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetResponseParameter

Purpose

To provide access to a protocol parameter set response parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetResponse\PPSetResponseParameters\
PPSetResponseParameter
```

Description

The PPSetResponseParameter object can be created as a Variant data object by using the **GetPPSetResponseParameterByIndex** or the **GetPPSetResponseParameterByPath** automation block, refer to PPSetResponseParameters on page 92.

The following automation blocks use the instantiated **PPSetResponseParameter** data object:

Automation Block	Description
GetLongNameOfPPSetResponseParameter	To get the long name of the protocol parameter set response parameter. The block instantiates the **LongName** data object.
GetPathOfPPSetResponseParameter	To get the path of the protocol parameter set response parameter. The block instantiates the **Path** data object.
GetSemanticOfPPSetResponseParameter	To get the semantics of the protocol parameter set response parameter. The block instantiates the **Semantics** data object.
GetShortNameOfPPSetResponseParameter	To get the short name of the protocol parameter set response parameter. The block instantiates the **ShortName** data object.
GetTypeOfPPSetResponseParameter	To get the type of the protocol parameter set response parameter. The block instantiates the **Type** data object.
GetUnitOfPPSetResponseParameter	To get the unit of the protocol parameter set response parameter. The block instantiates the **Unit** data object.
GetValueOfPPSetResponseParameter	To get the value of the protocol parameter set response parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetResponseParameters

Purpose

To provide access to a collection of protocol parameter set response parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetResponse\PPSetResponseParameters
```

Description

The PPSetResponseParameters object can be created as a Variant data object by using the **GetPPSetResponseParameters** automation block, refer to PPSetResponse on page 90.

The following automation blocks use the instantiated **PPSetResponseParameters** data object:

Automation Block	Description
CountPPSetResponseParameters	To get the number of protocol parameter set response parameters. The block instantiates the **NoOfPPSetResponseParameters** data object.
GetPPSetResponseParameterByIndex	To get the protocol parameter set response parameter by its index. The block instantiates the **PPSetResponseParameter** data object (see PPSetResponseParameter on page 91).
GetPPSetResponseParameterByPath	To get the protocol parameter set response parameter by its path. The block instantiates the **PPSetResponseParameter** data object (see PPSetResponseParameter on page 91).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetResponseRequestParameter

Purpose

To provide access to the corresponding protocol parameter set request parameter of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
```

```
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetResponse\
PPSetResponseRequestParameters\PPSetResponseRequestParameter
```

Description

The PPSetResponseRequestParameter object can be created as a Variant data object by using the **GetPPSetResponseRequestParameterByIndex** or the **GetPPSetResponseRequestParameterByPath** automation block, refer to PPSetResponseRequestParameters on page 93.

The following automation blocks use the instantiated **PPSetResponseRequestParameter** data object:

Automation Block	Description
GetLongNameOfPPSetResponseRequestParameter	To get the long name of the protocol parameter set response request parameter. The block instantiates the **LongName** data object.
GetPathOfPPSetResponseRequestParameter	To get the path of the protocol parameter set response request parameter. The block instantiates the **Path** data object.
GetSemanticOfPPSetResponseRequestParameter	To get the semantics of the protocol parameter set response request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfPPSetResponseRequestParameter	To get the short name of the protocol parameter set response request parameter. The block instantiates the **ShortName** data object.
GetTypeOfPPSetResponseRequestParameter	To get the type of the protocol parameter set response request parameter. The block instantiates the **Type** data object.
GetUnitOfPPSetResponseRequestParameter	To get the unit of the protocol parameter set response request parameter. The block instantiates the **Unit** data object.
GetValueOfPPSetResponseRequestParameter	To get the value of the protocol parameter set response request parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

PPSetResponseRequestParameters

Purpose

To provide access to corresponding protocol parameters set response request parameters of a object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet\PPSetResponse\
PPSetResponseRequestParameters
```

Description

The PPSetResponseRequestParameters object can be created as a Variant data object by using the **GetPPSetResponseRequestParameters** automation block, refer to PPSetResponse on page 90.

The following automation blocks use the instantiated **PPSetResponseRequestParameters** data object:

Automation Block	Description
CountPPSetResponseRequestParameters	To get the number of protocol parameter set response request parameters. The block instantiates the **NoOfPPSetResponseRequestParameters** data object.
GetPPSetResponseRequestParameterByIndex	To get the protocol parameter set response request parameter by its index. The block instantiates the **PPSetResponseRequestParameter** data object (see PPSetResponseRequestParameter on page 92).
GetPPSetResponseRequestParameterByPath	To get the protocol parameter set response request parameter by its path. The block instantiates the **PPSetResponseRequestParameter** data object (see PPSetResponseRequestParameter on page 92).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ProtocolParameterSet

Purpose

To provide access to protocol parameter set in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
ProtocolParameterSet
```

Description

The ProtocolParameterSet object can be created as a Variant data object by using the **GetProtocolParameterSet** automation block, refer to ActiveLogicalLink on page 66.

The following automation blocks use the instantiated **ProtocolParameterSet** data object:

Automation Block	Description
GetPPSetActiveValues	To get the protocol parameter set response parameters collection via active values. The block instantiates the **PPSetResponseParameters** data object (see PPSetResponseParameters on page 92).
GetPPSetRequestParameters	To get the protocol parameter set request parameters collection. The block instantiates the **PPSetRequestParameters** data object (see PPSetRequestParameters on page 89).
UpdateProtocolParameterSet	To update the protocol parameter set. The block instantiates the **ProtocolParameterSetResponse** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

SelectedVehicle

Purpose

To provide access to a selected vehicle in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle
```

Description

The SelectedVehicle object can be created as a Variant data object by using the **GetSelectedVehicle** automation block, refer to VehicleSelection on page 106.

The following automation blocks use the instantiated **SelectedVehicle** data object:

Automation Block	Description
GetLogicalLinkSelection	To get the logical link selection object. The block instantiates the **LogicalLinkSelection** data object (see LogicalLinkSelection on page 86).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Service

Purpose

To provide access to a diagnostic service in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service
```

Description

The Service object can be created as a Variant data object by using the **GetServiceByIndex** or the **GetServiceByName** automation block, refer to Services on page 104.

The following automation blocks use the instantiated **Service** data object:

Automation Block	Description
ExecuteService	To execute a service. The block instantiates the **ServiceResponses** data object (see ServiceResponses on page 103).
ExecuteServiceUsingCustomPDU	To execute a service using custom protocol data unit (PDU) data. The block instantiates the **ServiceResponses** data object (see ServiceResponses on page 103).
GetServiceLongName	To get the long name of the service. The block instantiates the **LongName** data object.
GetServiceRequestPDU	To get the service request protocol data unit (PDU) data. The block instantiates the **RequestPDU** data object.
GetServiceRequestParameters	To get the service request parameters collection. The block instantiates the **ServiceRequestParameters** data object (see ServiceRequestParameters on page 98).
GetServiceShortName	To get the short name of the service. The block instantiates the **ShortName** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceRequestParameter

Purpose

To provide access to a service request parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceRequestParameters\
ServiceRequestParameter
```

Description

The ServiceRequestParameter object can be created as a Variant data object by using the **GetServiceRequestParameterByIndex** or the **GetServiceRequestParameterByPath** automation block, refer to ServiceRequestParameters on page 98.

The following automation blocks use the instantiated **ServiceRequestParameter** data object:

Automation Block	Description
GetBitLengthOfServiceRequestParameter	To get the length in bit of the service request parameter. The block instantiates the **BitLength** data object.
GetBitPositionOfServiceRequestParameter	To get the bit position of the service request parameter. The block instantiates the **BitPosition** data object.
GetBytePositionOfServiceRequestParameter	To get the byte position of the service request parameter. The block instantiates the **BytePosition** data object.
GetDefaultValueOfServiceRequestParameter	To get the default value of the service request parameter. The block instantiates the **DefaultValue** data object.
GetLongNameOfServiceRequestParameter	To get the long name of the service request parameter. The block instantiates the **LongName** data object.
GetPathOfServiceRequestParameter	To get the path of the service request parameter. The block instantiates the **Path** data object.
GetSemanticOfServiceRequestParameter	To get the semantics of the service request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfServiceRequestParameter	To get the short name of the service request parameter. The block instantiates the **ShortName** data object.
GetTypeOfServiceRequestParameter	To get the type of the service request parameter. The block instantiates the **Type** data object.
GetUnitOfServiceRequestParameter	To get the unit of the service request parameter. The block instantiates the **Unit** data object.
GetValueOfServiceRequestParameter	To get the value of the service request parameter. The block instantiates the **Value** data object.
GetValueRestrictionsOfServiceRequestParameter	To get the value restrictions of the service request parameter. The block instantiates the **ValueRestrictions** data object.
IsServiceRequestParameterKeyParameter	To check if the service request parameter is a key parameter. The block instantiates the **IsKeyParameter** data object.

Automation Block	Description
IsServiceRequestParameterReadOnly	To check if the service request parameter is read-only. The block instantiates the **IsReadOnly** data object.
SetValueOfServiceRequestParameter	To set a value for the specified ServiceRequestParameter.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceRequestParameters

Purpose

To provide access to all service request parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceRequestParameters
```

Description

The ServiceRequestParameters object can be created as a Variant data object by using the **GetServiceRequestParameters** automation block, refer to Service on page 96.

The following automation blocks use the instantiated **ServiceRequestParameters** data object:

Automation Block	Description
CountServiceRequestParameters	To get the number of service request parameters. The block instantiates the **NoOfServiceRequestParameters** data object.
GetServiceRequestParameterByIndex	To get the request parameter object by its index. The block instantiates the **ServiceRequestParameter** data object (see ServiceRequestParameter on page 96).
GetServiceRequestParameterByPath	To get the request parameter object by its name. The block instantiates the **ServiceRequestParameter** data object (see ServiceRequestParameter on page 96).
ResetServiceRequestParametersToDefault	To reset the service request parameters to default.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceResponse

Purpose

To provide access to a service response in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceResponses\ServiceResponse
```

Description

The ServiceResponse object can be created as a Variant data object by using the **GetServiceResponseByIndex** automation block, refer to ServiceResponses on page 103.

The following automation blocks use the instantiated **ServiceResponse** data object:

Automation Block	Description
GetServiceResponsePDU	To get the service response protocol data unit (PDU) data. The block instantiates the **ResponsePDU** data object.
GetServiceResponseParameters	To get the service response parameters collection object. The block instantiates the **ServiceResponseParameters** data object (see ServiceResponseParameters on page 101).
GetServiceResponseRequestParameters	To get the service response request parameters collection object. The block instantiates the **ServiceResponseRequestParameters** data object (see ServiceResponseRequestParameters on page 102).
GetServiceResponseTimeStamp	To get the time stamp of the service response. The block instantiates the **ResponseTimeStamp** data object.
GetServiceResponseType	To get the type of the service response. The block instantiates the **ResponseType** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceResponseParameter

Purpose

To provide access to a service response parameter in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceResponses\ServiceResponse\
ServiceResponseParameters\ServiceResponseParameter
```

Description

The ServiceResponseParameter object can be created as a Variant data object by using the **GetServiceResponseParameterByIndex** or the **GetServiceResponseParameterByName** automation block, refer to ServiceResponseParameters on page 101.

The following automation blocks use the instantiated **ServiceResponseParameter** data object:

Automation Block	Description
GetBitLengthOfServiceResponseParameter	To get the length in bit of the service response parameter. The block instantiates the **BitLength** data object.
GetBitPositionOfServiceResponseParameter	To get the bit position of the service response parameter. The block instantiates the **BitPosition** data object.
GetBytePositionOfServiceResponseParameter	To get the byte position of the service response parameter. The block instantiates the **BytePosition** data object.
GetLongNameOfServiceResponseParameter	To get the long name of the service response parameter. The block instantiates the **LongName** data object.
GetPathOfServiceResponseParameter	To get the path of the service response parameter. The block instantiates the **Path** data object.
GetSemanticOfServiceResponseParameter	To get the semantics of the service response parameter. The block instantiates the **Semantics** data object.
GetShortNameOfServiceResponseParameter	To get the short name of the service response parameter. The block instantiates the **ShortName** data object.
GetTypeOfServiceResponseParameter	To get the type of the service response parameter. The block instantiates the **Type** data object.
GetUnitOfServiceResponseParameter	To get the unit of the service response parameter. The block instantiates the **Unit** data object.
GetValueOfServiceResponseParameter	To get the value of the service response parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceResponseParameters

Purpose

To provide access to service response parameters in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceResponses\ServiceResponse\
ServiceResponseParameters\ServiceResponseParameter
```

Description

The ServiceResponseParameters object can be created as a Variant data object by using the **GetServiceResponseParameters** automation block, refer to ServiceResponse on page 99.

The following automation blocks use the instantiated **ServiceResponseParameters** data object:

Automation Block	Description
CountServiceResponseParameters	To get the service response parameter object by its name. The block instantiates the **NoOfServiceResponseParameters** data object.
GetServiceResponseParameterByIndex	To get the service response parameter object by its index. The block instantiates the **ServiceResponseParameter** data object (see ServiceResponseParameter on page 99).
GetServiceResponseParameterByPath	To get the service response parameter object by its name path. The block instantiates the **ServiceResponseParameter** data object (see ServiceResponseParameter on page 99).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceResponseRequestParameter

Purpose

To provide access to the corresponding service request parameter of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
```

```
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceResponses\ServiceResponse\
ServiceResponseRequestParameters\
ServiceResponseRequestParameter
```

Description

The ServiceResponseRequestParameter object can be created as a Variant data object by using the **GetServiceResponseRequestParameterByIndex** or the **GetServiceResponseRequestParameterByName** automation block, refer to ServiceResponseRequestParameters on page 102.

The following automation blocks use the instantiated **ServiceResponseRequestParameter** data object:

Automation Block	Description
GetBitLengthOfServiceResponseRequestParameter	To get the length in bit of the service response request parameter. The block instantiates the **BitLength** data object.
GetBitPositionOfServiceResponseRequestParameter	To get the bit position of the service response request parameter. The block instantiates the **BitPosition** data object.
GetBytePositionOfServiceResponseRequestParameter	To get the byte position of the service response parameter. The block instantiates the **BytePosition** data object.
GetLongNameOfServiceResponseRequestParameter	To get the long name of the service response request parameter. The block instantiates the **LongName** data object.
GetPathOfServiceResponseRequestParameter	To get the path of the service response request parameter. The block instantiates the **Path** data object.
GetSemanticOfServiceResponseRequestParameter	To get the semantics of the service response request parameter. The block instantiates the **Semantics** data object.
GetShortNameOfServiceResponseRequestParameter	To get the short name of the service response request parameter. The block instantiates the **ShortName** data object.
GetTypeOfServiceResponseRequestParameter	To get the type of the service response request parameter. The block instantiates the **Type** data object.
GetUnitOfServiceResponseRequestParameter	To get the unit of the service response request parameter. The block instantiates the **Unit** data object.
GetValueOfServiceResponseRequestParameter	To get the value of the service response request parameter. The block instantiates the **Value** data object.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceResponseRequestParameters

Purpose

To provide access to corresponding service request parameters of a response object in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceResponses\ServiceResponse\
ServiceResponseRequestParameters
```

Description

The ServiceResponseRequestParameters object can be created as a Variant data object by using the **GetServiceResponseRequestParameters** automation block, refer to ServiceResponse on page 99.

The following automation blocks use the instantiated **ServiceResponseRequestParameters** data object:

Automation Block	Description
CountServiceResponseRequestParameters	To get the service response request parameter object by its name. The block instantiates the **NoOfServiceResponseRequestParameters** data object.
GetServiceResponseRequestParameterByIndex	To get the service response request parameter object by its index. The block instantiates the **ServiceResponseRequestParameter** data object (see ServiceResponseRequestParameter on page 101).
GetServiceResponseRequestParameterByPath	To get the service response request parameter object by its name path. The block instantiates the **ServiceResponseRequestParameter** data object (see ServiceResponseRequestParameter on page 101).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

ServiceResponses

Purpose

To provide access to all service responses in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services\Service\ServiceResponses
```

Description

The ServiceResponses object can be created as a Variant data object by using the **ExecuteService** automation block, refer to Service on page 96.

The following automation blocks use the instantiated **ServiceResponses** data object:

Automation Block	Description
CountServiceResponses	To get the number of service responses. The block instantiates the **NoOfServiceResponses** data object.
GetExecutionTimeStampFromServiceResponses	To get the execution time stamp from the service responses. The block instantiates the **ExecutionTimeStamp** data object.
GetLogicalLinkNameFromServiceResponses	To get the logical link name from the service responses. The block instantiates the **LogicalLinkName** data object.
GetRequestPDUFromServiceResponses	To get the request protocol data unit (PDU) from the service responses. The block instantiates the **RequestPDU** data object.
GetRequestStatusFromServiceResponses	To get the request status from the service responses. The block instantiates the **RequestStatus** data object.
GetServiceNameFromServiceResponses	To get the service name from the service responses. The block instantiates the **ServiceName** data object.
GetServiceResponseByIndex	To get the service response object by its index. The block instantiates the **ServiceResponse** data object (see ServiceResponse on page 99).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Services

Purpose

To provide access to all diagnostic services in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\SelectedVehicle\
LogicalLinkSelection\ActiveLogicalLinks\ActiveLogicalLink\
Services
```

Description

The Services object can be created as a Variant data object by using the **GetServices** automation block, refer to ActiveLogicalLink on page 66.

The following automation blocks use the instantiated **Services** data object:

Automation Block	Description
CountServices	To get the number of services. The block instantiates the **NoOfServices** data object.
GetServiceByIndex	To get the service object by its index. The block instantiates the **Service** data object (see Service on page 96).
GetServiceByName	To get the service object by its name. The block instantiates the **Service** data object (see Service on page 96).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Vehicle

Purpose

To provide access to a vehicle in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\Vehicles\Vehicle
```

Description

The Vehicle object can be created as a Variant data object by using the **GetVehicleByIndex** or the **GetVehicleByName** automation block, refer to Vehicles on page 105.

The following automation blocks use the instantiated **Vehicle** data object:

Automation Block	Description
SelectVehicle	To select the vehicle.

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Vehicles

Purpose

To provide access to all vehicles in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection\Vehicles
```

Description

The Vehicles object can be created as a Variant data object by using the **GetVehicles** automation block, refer to VehicleSelection on page 106.

The following automation blocks use the instantiated **Vehicles** data object:

Automation Block	Description
CountVehicles	To get the number of vehicles. The block instantiates the **NoOfVehicles** data object.
GetVehicleByIndex	To get the Vehicle object by its index. The block instantiates the **Vehicle** data object (see Vehicle on page 105).
GetVehicleByName	To get the Vehicle object by its name. The block instantiates the **Vehicle** data object (see Vehicle on page 105).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

VehicleSelection

Purpose

To provide access to the vehicle selection in ControlDesk.

Path

The following shows the path of the **Basic Functions** in the ControlDesk Access Library:

```
Application\ActiveExperiment\Platforms\Platform\
ActiveDiagnosticsDatabase\VehicleSelection
```

Description

The VehicleSelection object can be created as a Variant data object by using the **GetVehicleSelection** automation block, refer to ActiveDiagnosticsDatabase on page 66.

The following automation blocks use the instantiated **VehicleSelection** data object:

Automation Block	Description
GetSelectedVehicle	To get the selected vehicle object. The block instantiates the **SelectedVehicle** data object (see SelectedVehicle on page 95).
GetVehicles	To get the vehicles collection object. The block instantiates the **Vehicles** data object (see Vehicles on page 105).

For more information, refer to API Reference Information (ControlDesk Automation 🕮).

Automation

Basics on Automating the Access to ControlDesk

Introduction

AutomationDesk provides a COM-based API to automate the handling of AutomationDesk.

Related information

The AutomationDesk COM API provides no specific objects for accessing ControlDesk. You can only use the basic automation features, such as executing a project via script.

For information on the available objects with their properties and methods, refer to Basic Interface (AutomationDesk Automation 🕮).

For basic information and instructions, refer to Basics and Instructions on page 9.

A

ActiveDiagnosticsDatabase
ControlDesk Access library 66
ActiveExperiment
ControlDesk Access library 60
ActiveLogicalLink
ControlDesk Access library 66
ActiveLogicalLinks
ControlDesk Access library 67
AddDiagResultDataToReport
ControlDesk Access library 47
AddPlatform
ControlDesk Access library 37
AddPlatformByID
ControlDesk Access library 37
AddSignalsToMainRecorder
ControlDesk Access library 31
AddVariableDescription
ControlDesk Access library 38
Application
ControlDesk Access library 61

C

CloseControlDesk
ControlDesk Access library 17
CloseProjectAndExperiment
ControlDesk Access library 18
Common Program Data folder 8
ConfigureDeviceGeneralSettings
ControlDesk Access library 40
ConnectPlatform
ControlDesk Access library 41
ControlDesk Access library
ActiveDiagnosticsDatabase 66
ActiveExperiment 60
ActiveLogicalLink 66
ActiveLogicalLinks 67
AddDiagResultDataToReport 47
AddPlatform 37
AddPlatformByID 37
AddSignalsToMainRecorder 31
AddVariableDescription 38
Application 61
CloseControlDesk 17
CloseProjectAndExperiment 18
ConfigureDeviceGeneralSettings 40
ConnectPlatform 41
ControlPrimitive 68
ControlPrimitives 69
CreateDiagResultDataFromResponses 48
CtrlPrimitiveRequestParameter 69
CtrlPrimitiveRequestParameters 71
CtrlPrimitiveResponse 71
CtrlPrimitiveResponseParameter 72
CtrlPrimitiveResponseParameters 73
CtrlPrimitiveResponseRequestParameter 74
CtrlPrimitiveResponseRequestParameters 75
CtrlPrimitiveResponses 76
DirectClearAllDTCs 49
DirectClearDTC 50
DirectExecuteHexService 56
DirectExecuteService 57
DirectExecuteServiceUsingCustomPDU 59
DirectReadDTCs 51
DirectReadDTCsWithEnvData 52
DirectReadEnvironmentData 53
DisconnectPlatform 42
example 10
GetActiveLogicalLink 55
GetDiagPlatform 54
GetMemorySegments 43
GetParameterValue 23
GetPlatformStates 44
GetRecordedData 32
Job 77
JobRequestParameter 77
JobRequestParameters 79
JobResponse 79
JobResponseParameter 80
JobResponseParameters 81
JobResponseRequestParameter 82
JobResponseRequestParameters 83
JobResponses 84
Jobs 84
LogicalLinks 85
LogicalLinkSelection 86
OpenProjectAndExperiment 19
overview 9
Platform (Common) 61
Platform (Diagnostics) 86
Platforms (Common) 62
Platforms (Diagnostics) 87
PPSetRequestParameter 88
PPSetRequestParameters 89
PPSetResponse 90
PPSetResponseParameter 91
PPSetResponseParameters 92
PPSetResponseRequestParameter 92
PPSetResponseRequestParameters 93
ProtocolParameterSet 94
ReadVariableValue 24
RemovePlatform 45
RemoveUnconnectedSignalsFromMSL 34
SaveProjectAndExperiment 20
SelectedVehicle 95
Service 96
ServiceRequestParameter 96
ServiceRequestParameters 98
ServiceResponse 99
ServiceResponseParameter 99
ServiceResponseParameters 101
ServiceResponseRequestParameter 101
ServiceResponseRequestParameters 102
ServiceResponses 103
Services 104
SetParameterValue 26
Snapshot 21
StartControlDesk 20
StartMeasurementAndRecording 34
StartOnlineCalibration 27
StopMeasurementAndRecording 35
StopOnlineCalibration 28
SwitchMemoryPage 28
Vehicle 105
Vehicles 105
VehicleSelection 106
WriteVariableValue 29
ControlPrimitive
ControlDesk Access library 68
ControlPrimitives
ControlDesk Access library 69
CreateDiagResultDataFromResponses
ControlDesk Access library 48
CtrlPrimitiveRequestParameter
ControlDesk Access library 69
CtrlPrimitiveRequestParameters
ControlDesk Access library 71
CtrlPrimitiveResponse
ControlDesk Access library 71
CtrlPrimitiveResponseParameter
ControlDesk Access library 72
CtrlPrimitiveResponseParameters
ControlDesk Access library 73
CtrlPrimitiveResponseRequestParameter
ControlDesk Access library 74
CtrlPrimitiveResponseRequestParameters
ControlDesk Access library 75
CtrlPrimitiveResponses
ControlDesk Access library 76

D

DirectClearAllDTCs
ControlDesk Access library 49
DirectClearDTC
ControlDesk Access library 50
DirectExecuteHexService
ControlDesk Access library 56
DirectExecuteService
ControlDesk Access library 57
DirectExecuteServiceUsingCustomPDU
ControlDesk Access library 59
DirectReadDTCs
ControlDesk Access library 51
DirectReadDTCsWithEnvData
ControlDesk Access library 52
DirectReadEnvironmentData
ControlDesk Access library 53
DisconnectPlatform
ControlDesk Access library 42
Documents folder 8

G

GetActiveLogicalLink
ControlDesk Access library 55
GetDiagPlatform
ControlDesk Access library 54
GetMemorySegments
ControlDesk Access library 43
GetParameterValue
ControlDesk Access library 23

GetPlatformStates
ControlDesk Access library 44
GetRecordedData
ControlDesk Access library 32

J

Job
ControlDesk Access library 77
JobRequestParameter
ControlDesk Access library 77
JobRequestParameters
ControlDesk Access library 79
JobResponse
ControlDesk Access library 79
JobResponseParameter
ControlDesk Access library 80
JobResponseParameters
ControlDesk Access library 81
JobResponseRequestParameter
ControlDesk Access library 82
JobResponseRequestParameters
ControlDesk Access library 83
JobResponses
ControlDesk Access library 84
Jobs
ControlDesk Access library 84

L

Local Program Data folder 8
LogicalLinks
ControlDesk Access library 85
LogicalLinkSelection
ControlDesk Access library 86

O

OpenProjectAndExperiment
ControlDesk Access library 19

P

Platform (Common)
ControlDesk Access library 61
Platform (Diagnostics)
ControlDesk Access library 86
Platforms (Common)
ControlDesk Access library 62
Platforms (Diagnostics)
ControlDesk Access library 87
PPSetRequestParameter
ControlDesk Access library 88
PPSetRequestParameters
ControlDesk Access library 89
PPSetResponse
ControlDesk Access library 90
PPSetResponseParameter
ControlDesk Access library 91
PPSetResponseParameters
ControlDesk Access library 92
PPSetResponseRequestParameter
ControlDesk Access library 92
PPSetResponseRequestParameters
ControlDesk Access library 93
ProtocolParameterSet
ControlDesk Access library 94

R

ReadVariableValue
ControlDesk Access library 24
RemovePlatform
ControlDesk Access library 45
RemoveUnconnectedSignalsFromMSL
ControlDesk Access library 34

S

SaveProjectAndExperiment
ControlDesk Access library 20
SelectedVehicle
ControlDesk Access library 95
Service
ControlDesk Access library 96
ServiceRequestParameter
ControlDesk Access library 96
ServiceRequestParameters
ControlDesk Access library 98
ServiceResponse
ControlDesk Access library 99
ServiceResponseParameter
ControlDesk Access library 99
ServiceResponseParameters
ControlDesk Access library 101
ServiceResponseRequestParameter
ControlDesk Access library 101
ServiceResponseRequestParameters
ControlDesk Access library 102
ServiceResponses
ControlDesk Access library 103
Services
ControlDesk Access library 104
SetParameterValue
ControlDesk Access library 26
Snapshot
ControlDesk Access library 21
StartControlDesk
ControlDesk Access library 20
StartMeasurementAndRecording
ControlDesk Access library 34
StartOnlineCalibration
ControlDesk Access library 27
StopMeasurementAndRecording
ControlDesk Access library 35
StopOnlineCalibration
ControlDesk Access library 28
SwitchMemoryPage
ControlDesk Access library 28

V

Vehicle
ControlDesk Access library 105
Vehicles
ControlDesk Access library 105
VehicleSelection
ControlDesk Access library 106

W

WriteVariableValue
ControlDesk Access library 29