AutomationDesk

Accessing Real-Time Testing

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 and the contents hereof are subject to change without notice.

Contents

About This Document

Content

This document gives you information on how to access Real-Time Testing 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

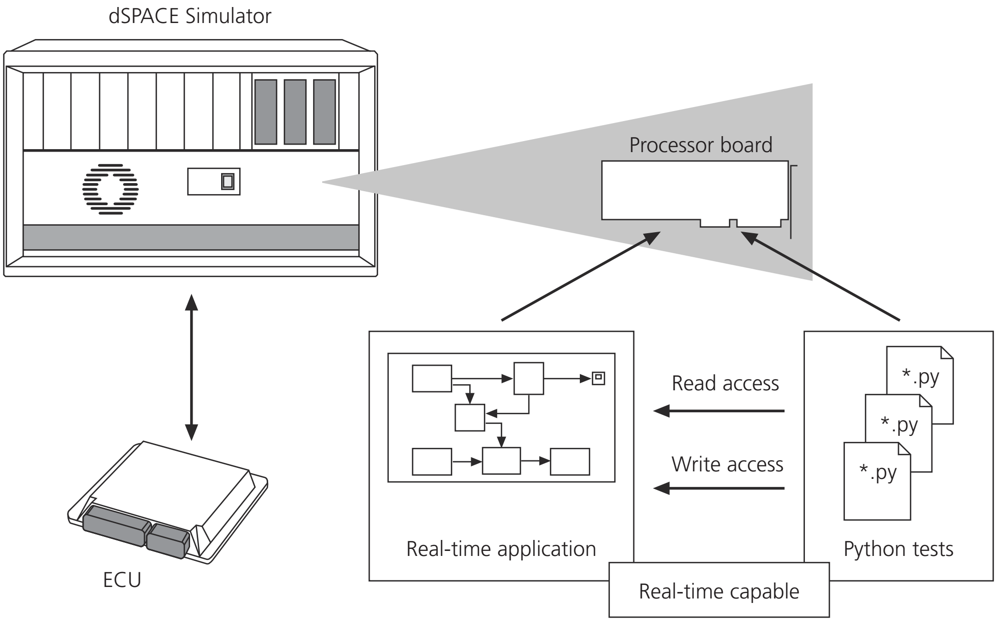
Where to go from here

Information in this section

Basics on Real-Time Testing

Introduction

Real-Time Testing means that you can synchronously execute tests and the real-time application on a real-time time basis for a greater timing precision. You programmed these tests, the Real-Time Testing (RTT) sequences, previously with the Python script language.



You can manage the execution of these RTT sequences with AutomationDesk's Real-Time Testing library.

Note

You must be familiar with Real-Time Testing to work with the Real-Time Testing library in AutomationDesk.

Basics on running RTT sequences

The real-time application and the RTT sequences run on the same processor board. The board must calculate the real-time application and scheduled parts of the RTT sequences in one sampling step. You can manage the RTT sequences from the host PC with AutomationDesk. For further information, refer to Basics on Running RTT Sequences (Real-Time Testing Guide 🕮).

States of RTT sequences

An RTT sequence can have different states, such as *running*, *stopped*, or *paused*. For information on all possible states and their meanings, refer to Basics on Managing RTT Sequences (Real-Time Testing Guide 🕮).

BCG file handling

You have to convert your RTT sequence (*.py) into the signed byte code generated (BCG) file format. Only signed BCG files can be transferred to the real-time platform.

The Real-Time Testing library has a Bytecode Generation folder providing automation blocks to generate and work with BCG files. For detailed information on the blocks, refer to Bytecode Generation on page 26.

Data objects for the connection to the RTT Manager Server

AutomationDesk's Real-Time Testing library provides data objects to instantiate the data objects which you implemented in your RTT source code. You have to use the data objects of the Real-Time Testing library when you structure the project. This provides access to the Real-Time Testing Manager Server.

Maximum number of variables

It is not possible to state precisely the maximum number of variables that you can write during one test step in a Real-Time Testing sequence.

It depends on the following factors:

- Board used (DS1006, MicroAutoBox)
- CPU clock frequency
- Size and turnaround time of the model the real-time application is based on
- Remaining free memory
- Performance of the implemented real-time application

For example, if you use a DS1006 with 1 GHz, writing a variable takes about 0.4 µs. This allows you to write more than one hundred variables in the RTT demo model, which you can find at `C:\Program Files\Common Files\dSPACE\RealTimeTesting\<VersionNumber>\Demos\SampleExperiments`.

You should determine the specific limits for your test environment before you start complex test scenarios.

Limitation on RTT event handling

RTT event handling is not yet supported by AutomationDesk.

DS1006, MicroAutoBox II: required RTT versions

Note

The RTT version used for building the real-time application and the RTT version active on the host PC must be identical to use RTT in connection with AutomationDesk and one of the following dSPACE simulation platforms:

- DS1006
- MicroAutoBox II

SCALEXIO, MicroLabBox, MicroAutoBox III, DS1007, VEOS: required RTT versions

Note

The RTT version of the simulation platform and the RTT version active on the host PC must be identical to use RTT in connection with AutomationDesk and one of the following dSPACE simulation platforms:

- dSPACE real-time simulation platforms:
  - SCALEXIO
  - MicroLabBox
  - MicroAutoBox III
  - DS1007

  The RTT version of a real-time simulation platform depends on the platform's *firmware version*.
- VEOS

  The RTT version of VEOS depends on the *product version*.

The following table shows the mapping between the firmware/product version and the corresponding RTT version:

Host PC		**Compatible Firmware Version**				
dSPACE Release	**Real-Time Testing Version**	**SCALEXIO**	**MicroAutoBox III**	**DS1202 MicroLabBox**	**DS1007**	**VEOS**
RLS2021-A	5.0	5.1 5.0	5.1	2.16	3.16	5.2
RLS2020-B	4.4	5.0	5.0	2.14	3.14	5.1
RLS2020-A	4.3	4.6	4.6	2.12	3.12	5.0
RLS2019-B	4.2	4.5	4.5	2.10	3.10	4.5
RLS2019-A	4.1	4.4	–	2.8	3.8	4.4
RLS2018-B	4.0	4.3	–	2.6	3.6	4.3
RLS2018-A	3.4	4.2	–	2.4	3.4	4.2
RLS2017-B	3.3	4.1	–	2.2	3.2	4.1
RLS2017-A	3.2	4.0	–	2.0	3.0	4.0
RLS2016-B	3.1	3.5	–	1.7	2.6	3.7
RLS2016-A	3.0	3.4	–	1.5	2.4	3.6
RLS2015-B	2.6	3.3	–	1.3	2.2	3.5
RLS2015-A	2.5	3.2	–	–	2.0	3.4
RLS2014-B	2.4	3.1	–	–	–	3.3
RLS2014-A	2.3	3.0	–	–	–	3.2
RLS2013-B	2.2	2.3	–	–	–	3.1

Python version on the platform

Real-Time Testing supports Python with a specific version on the platform, i.e. the tests can use standard modules of this version.

Note that older versions of Real-Time Testing used older Python versions for RTT sequences. Old real-time models and BCG files have to be recompiled with the current version of Real-Time Testing to upgrade them to the required Python.

Real-Time Testing Version	Python Version
Real-Time Testing 1.2 and earlier	Python 2.4.3
Real-Time Testing 1.3 … Real-Time Testing 2.6	Python 2.5.1
Real-Time Testing 3.0	Python 2.7.10
Real-Time Testing 3.1 and later	Python 2.7.11[1]
Real-Time Testing 4.2	Python 3.6.4[2] for VEOS and MicroAutoBox III Python 2.7.11 for all other platforms
Real-Time Testing 4.4 and later	Python 3.6.4[2] for SCALEXIO, VEOS, and MicroAutoBox III Python 2.7.11 for all other platforms

[1] Recompiling the BCG files are not necessary when they are compiled with Python 2.7.10.

[2] Note that all strings in Python 3.6 are unicode objects. You must use the correct encoding. For more information on migrating to this Python version, refer to www.dspace.com/go/Python36Migration.

Python version on the host PC

Real-Time Testing supports Python 3.9 on the host PC.

The activated Real-Time Testing version and the dSPACE software using Real-Time Testing, e.g., AutomationDesk, must support the same Python version.

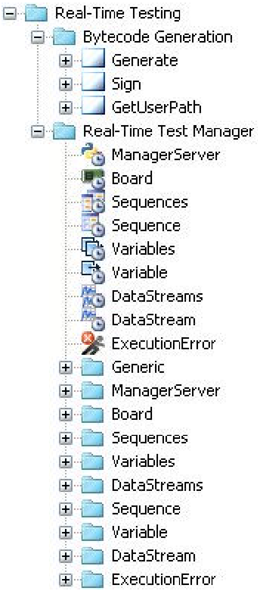
Examples

AutomationDesk demo projects can be found at `<DocumentsFolder>\RealTimeTesting`.

Overview of the Real-Time Testing Library Elements

Introduction

AutomationDesk´s Real-Time Testing library provides the functionality to automate the management of Real-Time Testing (RTT) projects.



- Real-Time Testing
  - Bytecode Generation
    - Generate
    - Sign
    - GetUserPath
  - Real-Time Test Manager
    - ManagerServer
    - Board
    - Sequences
    - Sequence
    - Variables
    - Variable
    - DataStreams
    - DataStream
    - ExecutionError
    - Generic
    - ManagerServer
    - Board
    - Sequences
    - Variables
    - DataStreams
    - Sequence
    - Variable
    - DataStream
    - ExecutionError

Bytecode Generation

The Bytecode Generation folder provides automation blocks to generate and work with RTT sequence files in BCG file format. For further information, refer to Bytecode Generation on page 26.

Real-Time Test Manager

AutomationDesk's Real-Time Test Manager folder contains RTT-specific data objects and folders with RTT-specific automation blocks.

Data objects There are RTT-specific data objects available (ManagerServer, Board, etc.) which are the interfaces to the Real-Time Test Manager Server. For further information, refer to Data Objects of the Real-Time Test Manager Folder on page 30.

Generic The Generic folder provides automation blocks which you can use in conjunction with the Real-Time Test Manager data objects. For further information, refer to Generic on page 37.

ManagerServer The ManagerServer folder provides specific automation blocks to access the Real-Time Test Manager Server. For further information, refer to ManagerServer on page 39.

Board The Board folder provides automation blocks to manage real-time platform-specific tasks. For further information, refer to Board on page 42.

Sequences The Sequences folder provides automation blocks to manage collections of RTT sequences. For further information, refer to Sequences on page 45.

Variables The Variables folder provides automation blocks to manage collections of RTT variables. For further information, refer to Variables on page 55.

DataStreams The DataStreams folder provides automation blocks to manage collections of RTT datastreams. For further information, refer to DataStreams on page 59.

Sequence The Sequence folder provides automation blocks to manage single RTT sequences. For further information, refer to Sequence on page 63.

Variable The Variable folder provides automation blocks to manage single RTT variables. For further information, refer to Variable on page 76.

DataStream The DataStream folder provides automation blocks to manage single RTT datastreams. For further information, refer to DataStream on page 82.

ExecutionError The ExecutionError folder provides automation blocks to manage information on the last errors occurring during sequence execution. For further information, refer to ExecutionError on page 84.

Related topics

Basics

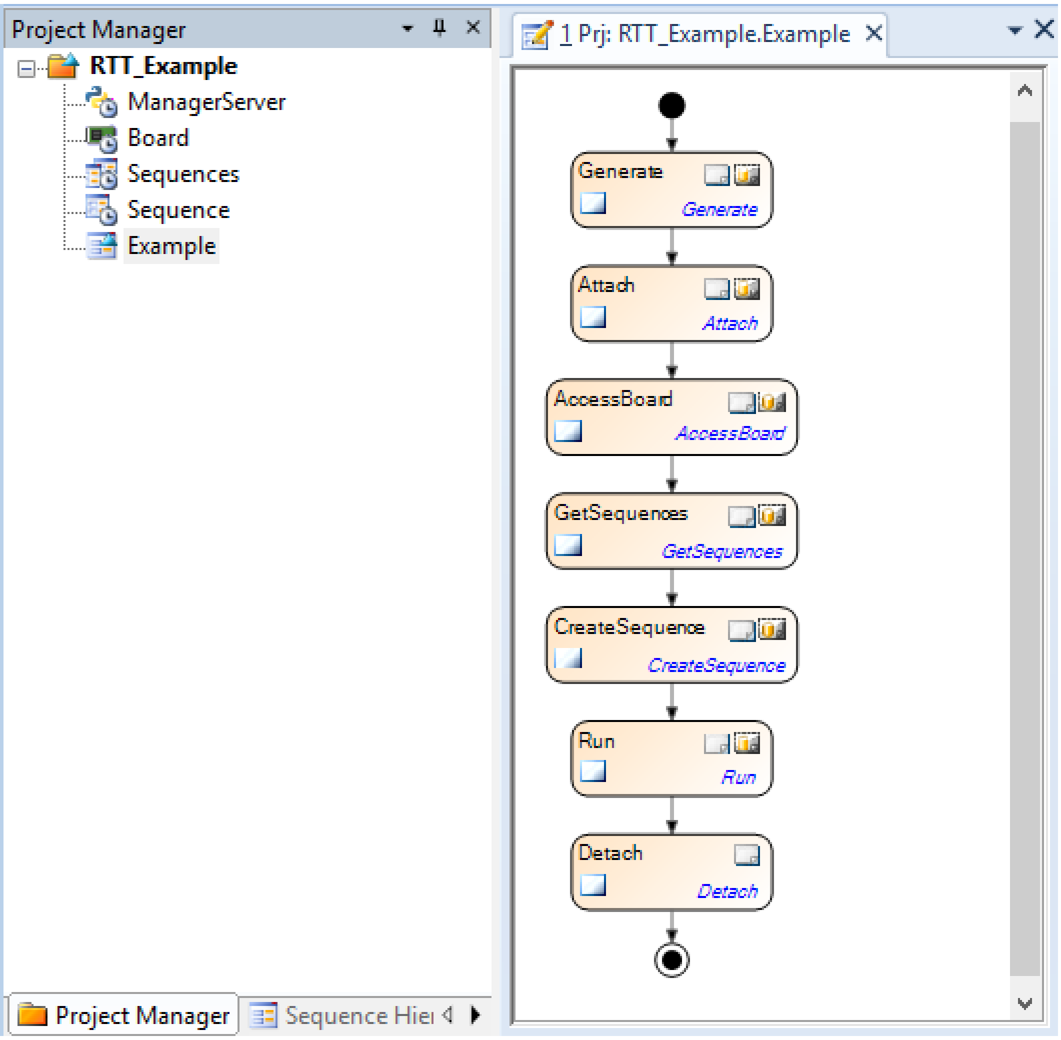
Packaging of AutomationDesk (AutomationDesk Introduction And Overview 🕮)

Example of a Real-Time Testing Project

Introduction

The following example contains the essential automation blocks and data objects to automate the management of an RTT project.

AutomationDesk sequence



Description of the sequence

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

- Generate

  To generate and sign a BCG file from your Python test file.
- Attach

  To attach the automation project to the Real-Time Testing Manager Server.
- AccessBoard

  To access a real-time platform which is registered in AutomationDesk or ControlDesk (for example: DS1006, or MicroAutoBox).
- GetSequences

  To access the Sequences (collection) from the real-time platform.

- CreateSequence

  To create a new RTT sequence on the real-time platform.
- Run (Executing the RTT sequence)

  To start an RTT sequence on the real-time platform.
- Detach

  To release the Real-Time Test Manager Server.

Demo projects

Further AutomationDesk demo projects can be found at `<DocumentsFolder>\RealTimeTesting`.

How to Use the Data Objects of a Real-Time Testing Project

Objective

Before you create an automation sequence for handling a Real-Time Testing project, you have to set up the data structure.

Project-specific data objects

You have to add project-specific data objects. It is recommended to place the data objects in the Project Manager. The parameters of the automation blocks must be parameterized by referencing these project-specific data objects. These data objects are the connection to Real-Time Testing Manager Server.

Precondition

Before you can execute a real-time test, the real-time hardware must be installed and configured correctly. You must also register it, for example, by using AutomationDesk's Platform Manager. For further information, refer to Registering and Managing dSPACE Platforms (AutomationDesk Accessing Simulation Platforms 🕮).

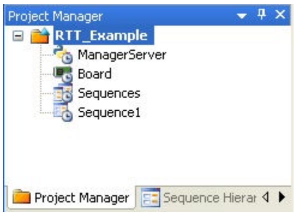
Method

How to use the data objects of a Real-Time Testing project

1. Create a new AutomationDesk project.
2. Drag the following data objects from the Real-Time Testing library to the Project Manager:
   - ManagerServer
   - Board
   - Sequences
   - Sequence
3. Drag all data objects to your AutomationDesk project which you programmed in your RTT sequence(s).

Result

You have created project-specific data objects in the Project Manager. These are basic requirements for working with the Real-Time Testing library.



Next steps

An RTT sequence is implemented in the Python programming language. It must be converted into a byte code generator (BCG) file. Only BCG files can be downloaded to the real-time platform. The next step gives you instructions on generating BCG files. Refer to How to Generate a BCG File on page 18.

Related topics

References

Real-Time Test Manager.. 29

How to Generate a BCG File

Objective

An RTT sequence is implemented in the Python programming language. It must be converted into a signed byte code generator (BCG) file. Only signed BCG files can be downloaded to the real-time platform.

Tip

Using the automation blocks in the Bytecode Generation folder is optional. If you still have a signed BCG file, proceed with How to Initialize an RTT Project on page 20.
You must of course generate a new BCG file if you change the script in the Python file.

Preconditions

You have programmed an RTT sequence in Python. For information on programming RTT sequences, refer to Real-Time Testing Guide 📖.

Method

How to generate a BCG file

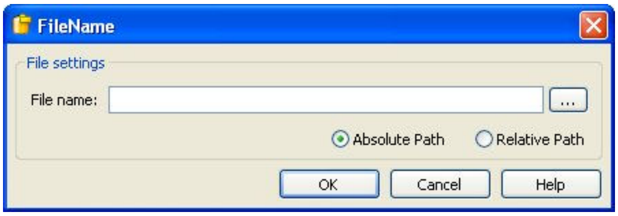
1 Choose **New Sequence** from the AutomationDesk project element's context menu to add a new sequence element to the Project Manager.

2 Double-click the sequence element to open it in the Sequence Builder.

3 Click the Real-Time Testing tree node and the Bytecode Generation tree node in the Library Browser to get access to the automation blocks.

4 Drag a **Generate** block from the Library Browser (Bytecode Generation folder) to the Sequence Builder.

5 Double-click the **FileName** data object in the Data Object Editor to open the dialog.



6 Enter the path where the Python file is stored on your computer and the file name.

Result

The **Generate** automation block generates a signed BCG file if you execute the sequence.

Note

The generated BCG file is usually placed in the same folder as the source file. If the folder already contains a BCG file with the same name, that file is replaced if it is older than the source file of the new BCG file. If the file is read-only, an exception is raised.

You can download only signed BCG files to the real-time platform. BCG files generated with Real-Time Testing 1.0 or the RTT_OPERATOR license of Real-Time Testing are unsigned. You can sign these files with the **Sign** automation block after generation.

Next steps

You can now proceed with initializing your RTT project. Refer to How to Initialize an RTT Project on page 20.

Related topics

References

How to Initialize an RTT Project

Objective

Each sequence that you build for an RTT project contains some common blocks.

Preconditions

- You have created the required data objects in the Project Manager, refer to How to Use the Data Objects of a Real-Time Testing Project on page 17.
- You have added an AutomationDesk sequence to your project and a signed BCG file is available (see How to Generate a BCG File on page 18).

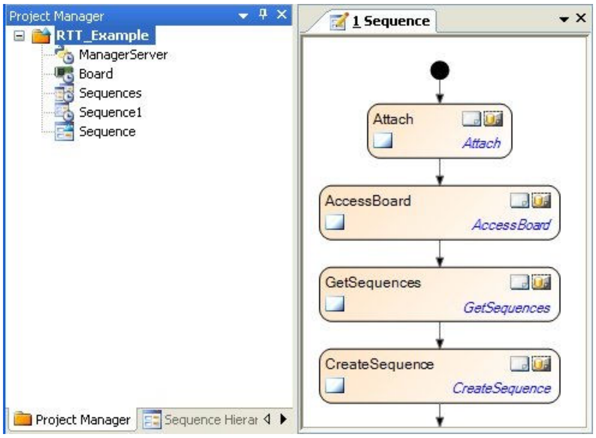
Method

How to initialize an RTT project

1. Drag an **Attach** block from the Library Browser (ManagerServer folder) to the Sequence Builder. This block starts the Real-Time Testing Manager Server, if it was not already started by another real-time task outside AutomationDesk, and attaches your automation project to the server.
2. Set the project-specific **ManagerServer** data object as a reference to the block's ManagerServer data object.
3. Drag an **AccessBoard** block from the Library Browser (ManagerServer folder) to the Sequence Builder for instantiating the objects for the real-time hardware.
4. Set the project-specific **ManagerServer** and **Board** data objects as a reference to the block's data objects.
5. Click the Value edit field of the **BoardName** data object and enter the name of the used dSPACE board.
6. Drag a **GetSequences** block from the Library Browser (Board folder) to the Sequence Builder to get access to the Sequences (collection) from the real-time platform.
7. Set the project-specific **Board** and **Sequences** data objects as a reference to the block's data objects.
8. Drag a **CreateSequence** block from the Library Browser (Sequences folder) to the Sequence Builder to create a new RTT sequence on the real-time platform.
9. Set the project-specific **Sequences** and **Sequence** data objects as a reference to the block's data objects.
10. Double-click the **FileName** data object in the Data Object Editor to open the dialog.
11. Enter the path where the BCG file is stored on your computer and the file name.
12. Add appropriate automation blocks to your AutomationDesk project if you programmed class descriptions in your RTT sequence(s), such as Variables or DataStreams.

Result

When you execute the automation sequence, the Real-Time Test Manager Server is started, if it was not already started by another real-time task outside AutomationDesk, and the data objects are attached to it.



Next steps

You can now add an automation block to the sequence to execute the created RTT sequence on the real-time platform. Refer to How to Execute a Real-Time Testing Project on page 21.

How to Execute a Real-Time Testing Project

Objective

AutomationDesk can start the RTT sequence execution on the real-time platform.

Preconditions

The real-time project is initialized (see How to Initialize an RTT Project on page 20).

Method

How to execute a Real-Time Testing project

1 Drag a Run block from the Library Browser (Sequence folder) to the already existing sequence in the Sequence Builder.

2 Set the project-specific Sequence data object as a reference to the block.

3 Optionally, you can add further data objects to the Run block to start it with arguments.

Result

When you execute the sequence, the RTT sequence is executed on the real-time platform.

> **Tip**
>
> Running RTT sequences can be paused, continued, or stopped. There are automation blocks for this. For further information, refer to:
>
> - Sequences on page 45
> - Sequence on page 63

Next steps

You can now release the Real-Time Testing Manager Server. Refer to How to Detach a Real-Time Testing Project on page 22.

Related topics

References

Continue	64
ContinueAll	46
New Data Object (AutomationDesk Basic Practices 📖)	
Pause	72
PauseAll	51
Run	74
RunAll	52

How to Detach a Real-Time Testing Project

Objective

To release the Real-Time Testing Manager Server and sign off the project-specific data objects.

> **Note**
>
> Before the execution of the RTT sequence finishes, you must detach all used project-specific data objects you attached when initializing the project. Otherwise the execution of subsequent RTT projects might be faulty.

Method

How to detach a Real-Time Testing project

1. Drag a Detach block from the Library Browser (Generic folder) to the Sequence Builder.
2. Choose New Data Object from the block's context menu to define the project-specific data object to be detached. Specify the data objects used in your project.

Note

When you detach the ManagerServer data object, all other data objects are still attached to the RTT Manager Server. This can cause faults when you execute your AutomationDesk project again. It is recommended to detach all the data objects used. The `Detach` method is executed in the same order as the added data objects (from top to bottom). The ManagerServer data object must be the last.

Result

When you execute the sequence, the Real-Time Testing Manager Server and all data objects are detached.

Related topics

References

Reference Information

Where to go from here

Information in this section

Automation Blocks

Where to go from here

Information in this section

Bytecode Generation

Introduction

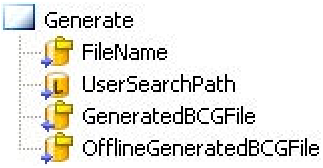
You can manage RTT sequences with the Bytecode Generation automation blocks.

Where to go from here

Information in this section

Generate

Graphical representation



Generate
- FileName
- UserSearchPath
- GeneratedBCGFile
- OfflineGeneratedBCGFile

Purpose

To generate and sign an RTT sequence file (BCG file).

Description

You can download only signed BCG files to the real-time platform.

> **Note**
>
> Signing an RTT sequence is only possible with the ADD_STANDARD license of AutomationDesk, which includes the RTT_DEVELOPER license of Real-Time Testing.

The generated BCG file is usually placed in the same folder as the source file. If the folder already contains a BCG file with the same name, that file is replaced if it is older than the source file of the new BCG file. If the file is read-only, an exception is raised.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
FileName	In	File	" "	Contains the file name of the Python file in which the RTT sequence is implemented.
UserSearchPath	In	List	[]	Contains a list of paths to the folders which contain the user-defined modules that are imported into the Python file. If you import compiled Python moduls (PYC files), they must be compiled with the Python version according to the RTT version. Refer to Basics on Real-Time Testing on page 9.
GeneratedBCGFile	Out	File	" "	Contains the generated BCG file.
OfflineGeneratedBCGFile	In	File	" "	Lets you specify the generated BCG file to be used in offline operation mode.

Related topics

HowTos

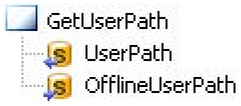
How to Generate a BCG File.......... 18

References

GetUserPath.......... 28
Sign.......... 28

GetUserPath

Graphical representation



GetUserPath
UserPath
OfflineUserPath

Purpose

To get the user path of an RTT sequence file (BCG file).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
UserPath	Out	String	" "	Contains a list of paths to folders including user modules which are imported into the Python file.
OfflineUserPath	In	String	" "	Lets you specify the user path to be used in offline operation mode.

Related topics

HowTos

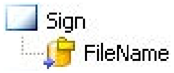
How to Generate a BCG File.. 18

References

Generate.. 26
Sign.. 28

Sign

Graphical representation



Sign
FileName

Purpose

To sign an RTT sequence file (BCG file).

Description

You can download only signed BCG files to the real-time platform.

Generated BCG files can be unsigned for the following reasons:

- The BCG files were generated with Real-Time Testing 1.0.
- The BCG files were generated with the RTT_OPERATOR license of Real-Time Testing.

You can sign a BCG file with this block after generation.

> **Note**
>
> Signing an RTT sequence is only possible with the ADD_STANDARD license of AutomationDesk, which includes the RTT_DEVELOPER license of Real-Time Testing.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
FileName	In	File	" "	Contains the BCG file to sign.

Related topics

HowTos

How to Generate a BCG File.. 18

References

Generate.. 26
GetUserPath... 28

Real-Time Test Manager

Introduction

AutomationDesk's Real-Time Test Manager folder contains RTT-specific data objects and subfolders with RTT-specific automation blocks.

Where to go from here

Information in this section

Information in other sections

rttmanagerlib Module (Real-Time Testing Library Reference 📖)
The Real-Time Test Manager Server handles the RTT sequences on the host PC and creates them on the simulation platform.

Data Objects of the Real-Time Test Manager Folder

Introduction

There are RTT-specific data objects which are the interfaces to the Real-Time Test Manager Server.

Where to go from here

Information in this section

Board (Data Object)

Graphical representation

Board

Purpose

To access the real-time platform on which the RTT sequence is executed.

Description

The Board data object is used to reference the data objects of corresponding automation blocks. It connects AutomationDesk to the real-time platform, which must be registered, for example, by ControlDesk's Platform Manager.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project.....17

References

Board.....42
Detach.....91
ManagerServer (Data Object).....33

DataStream (Data Object)

Graphical representation

DataStream

Purpose

To access a datastream of an RTT sequence.

Description

The DataStream data object is used to reference the data objects of automation blocks.

A datastream is a MAT file whose data is replayed to variable objects in your RTT sequence. Data is streamed from the host PC to the real-time system during the run time of the RTT sequence. For further information on datastreaming, refer to Data Replay in RTT Sequences (Real-Time Testing Guide 📖).

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project.................... 17

References

DataStream.................... 82
DataStreams.................... 59
DataStreams (Data Object).................... 32
Detach.................... 91

DataStreams (Data Object)

Graphical representation

DataStreams

Purpose

To access a collection of datastreams of an RTT sequence.

Description

The DataStreams data object is used to reference the data objects of automation blocks. For further information on datastreaming, refer to Data Replay in RTT Sequences (Real-Time Testing Guide 📖).

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project.. 17

References

DataStream... 82
DataStream (Data Object)... 32
DataStreams.. 59
Detach... 91

ExecutionError (Data Object)

Graphical representation

ExecutionError

Purpose

To access an execution error of an RTT sequence.

Description

The ExecutionError data object is used to reference the data objects of automation blocks. It contains information on the last errors occurring during sequence execution.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project.. 17

References

Detach... 91
ExecutionError.. 84

ManagerServer (Data Object)

Graphical representation

ManagerServer

Purpose

To access the Real-Time Test Manager Server.

Description

The ManagerServer data object is used to reference the data objects of automation blocks.

You can attach/detach the data object to/from the RTT Manager Server manually or with specific automation blocks.

Note

When you detach the ManagerServer data object, all other data objects are still attached to the RTT Manager Server. This can cause faults when you execute your AutomationDesk project again. It is recommended to use the Detach automation block at the end of your sequence to detach all data objects used.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project........ 17

References

Attach........ 89
Detach........ 91
ManagerServer........ 39

Sequence (Data Object)

Graphical representation

Sequence

Purpose

To access an RTT sequence.

Description

The Sequence data object is used to reference the data objects of automation blocks.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project 17

References

Continue 89
Detach 91
Pause 93
Run 94
Sequence 63
Sequences (Data Object) 35
Stop 95

Sequences (Data Object)

Graphical representation

Sequences

Purpose

To access a collection of sequences of an RTT sequence.

Description

The Sequences data object is used to reference the data objects of automation blocks.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project 17

References

ContinueAll 90
Detach 91
PauseAll 93
RunAll 95
Sequences 45
StopAll 96

Variable (Data Object)

Graphical representation

Variable

Purpose

To access a variable of an RTT sequence.

Description

The Variable data object is used to reference the data objects of automation blocks.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project.. 17

References

Detach.. 91
Variable.. 76
Variables.. 55

Variables (Data Object)

Graphical representation

Variables

Purpose

To access a collection of variables of an RTT sequence.

Description

The Variables data object is used to reference the data objects of automation blocks.

Related topics

HowTos

How to Use the Data Objects of a Real-Time Testing Project.. 17

References

Detach.. 91
Variables.. 55

Generic

Introduction

The Generic folder provides automation blocks which you can use in conjunction with the Real-Time Test Manager data objects.

Where to go from here

Information in this section

Detach

Graphical representation

Detach

Purpose

To detach one or more data objects from the Real-Time Test Manager Server at the same time.

Description

The Detach automation block does not provide data objects by default. You can define one or more data objects from the Real-Time Testing library via the New Data Object context menu.

The `Detach` method is executed in the same order of the added data objects (from top to bottom).

Data objects

This automation block can manage the following data objects:

Name	In / Out	Type	Default Value	Description
ManagerServer	In	ManagerServer (Data Object)	None	Contains the Real-Time Test ManagerServer to be used.
Board	In	Board (Data Object)	None	Contains the Board to be used.
Sequences	In	Sequences (Data Object)	None	Contains the collection of Sequences to be used.

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence to be used.
Variables	In	Variables (Data Object)	None	Contains the collection of Variables to be used.
Variable	In	Variable (Data Object)	None	Contains the Variable to be used.
DataStreams	In	DataStreams (Data Object)	None	Contains the collection of DataStreams to be used.
DataStream	In	DataStream (Data Object)	None	Contains the DataStream to be used.
ExecutionError	In	ExecutionError (Data Object)	None	Contains the ExecutionError to be used.

Related topics

HowTos

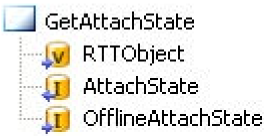
How to Detach a Real-Time Testing Project.......... 22

References

Attach.......... 41
GetAttachState.......... 38

GetAttachState

Graphical representation



GetAttachState
RTTObject
AttachState
OfflineAttachState

Purpose

To get the attach state of the referenced Real-Time Testing library data object.

Description

The required information on the referenced data object is saved to the AttachState data object. You can reference the following data objects via the RTTObject parameter value:

- ManagerServer (Data Object)
- Board (Data Object)
- Sequences (Data Object)
- Sequence (Data Object)
- Variables (Data Object)

- Variable (Data Object)
- DataStreams (Data Object)
- DataStream (Data Object)
- ExecutionError (Data Object)

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
RTTObject	In	Variant	None	Contains the data object on which you want information.
AttachState	Out	Integer	0	Contains the attach state. ▪ 0 = not attached ▪ 1 = is attached
OfflineAttachState	In	Integer	0	Lets you specify the values to be used in offline operation mode.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

Detach... 37
Detach... 91

ManagerServer

Introduction

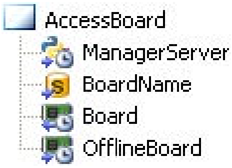
The ManagerServer folder provides specific automation blocks for the ManagerServer data object to get access to the Real-Time Test Manager Server.

Where to go from here

Information in this section

AccessBoard

Graphical representation



AccessBoard
ManagerServer
BoardName
Board
OfflineBoard

Purpose

To access a registered real-time platform (for example, the DS1007) or VEOS.

Description

The Board data object is attached if the AccessBoard automation block is executed.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ManagerServer	In	ManagerServer (Data Object)	None	Contains the ManagerServer to be used.
BoardName	In	String	''	Lets you specify the real-time platform or VEOS to be accessed. The notation is not case-sensitive and depends on the platform type: ▪ For a single processor board (DS1006, MicroAutoBox II), BoardName is the name of the real-time platform displayed in the Platform Manager. ▪ For a multiprocessor system based on the DS1006, you must access each processor board individually by appending an index to the platform name separated by an underscore, for example, `ds1006_2`, `ds1006_3` and so on. ▪ For a SCALEXIO system, BoardName is the IP address of the SCALEXIO Processing Unit and the application name separated by a slash, for example, `192.168.0.15/MyApp`. ▪ For VEOS, BoardName is the IP address of the host PC where VEOS runs and the application name separated by a slash, for example, `127.0.0.1/MyApp` if VEOS runs on the same PC as the Real-Time Test Manager. VEOS is supported for Real-Time Testing 2.0p1 and Real-Time Testing 2.2 and later.
Board	Out	Board (Data Object)	None	Returns the current Board.
OfflineBoard	In	Board (Data Object)	None	Lets you specify the Board to be used in offline operation mode.

Related topics

HowTos

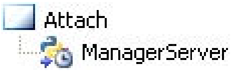
How to Initialize an RTT Project.. 20

References

Attach.. 41
Board.. 42
RealTimeTestManagerServer (Real-Time Testing Library Reference 📖)

Attach

Graphical representation



Attach
ManagerServer

Purpose

To attach the automation task to the Real-Time Testing Manager Server.

Description

The ManagerServer data object must be attached first in your Real-Time Testing automation task.

You can attach your automation sequence manually via the Attach command.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ManagerServer	In	ManagerServer (Data Object)	None	Contains the ManagerServer to be used.

Related topics

HowTos

How to Initialize an RTT Project.. 20

References

AccessBoard.. 40
Detach.. 37
GetAttachState.. 38
RealTimeTestManagerServer (Real-Time Testing Library Reference 📖)

Board

Introduction

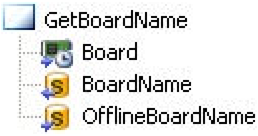
The board folder provides automation blocks to manage real-time platform-specific tasks.

Where to go from here

Information in this section

GetBoardName

Graphical representation



GetBoardName
- Board
- BoardName
- OfflineBoardName

Purpose

To read the name of the board the dSPACE real-time system is based on.

Description

The GetBoardName automation block reads the name of the real-time platform which is registered in AutomationDesk, for example.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Board	In	Board (Data Object)	None	Contains the real-time platform and must be referenced to the Board data object.
BoardName	Out	String	''	Contains the name of the real-time platform.
OfflineBoardName	In	String	''	Lets you specify the name of the real-time platform to be used in offline operation mode.

Related topics

Basics

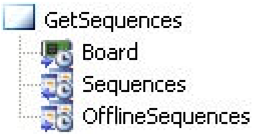
Basics on Real-Time Testing.......... 9

References

AccessBoard.......... 40
Board.......... 42
Board (Real-Time Testing Library Reference 📖)

GetSequences

Graphical representation



GetSequences
- Board
- Sequences
- OfflineSequences

Purpose

To get access to the Sequences (collection) from the real-time platform.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Board	In	Board (Data Object)	None	Contains the real-time platform to be used.
Sequences	Out	Sequences (Data Object)	None	Contains the Sequences (collection) from the real-time platform.
OfflineSequences	In	Sequences (Data Object)	None	Lets you specify the Sequences (collection) to be used in offline operation mode.

Related topics

Basics

Basics on Real-Time Testing..... 9

HowTos

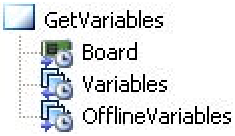
How to Initialize an RTT Project..... 20

References

Board (Real-Time Testing Library Reference 📖)
Sequences..... 45

GetVariables

Graphical representation



GetVariables
Board
Variables
OfflineVariables

Purpose

To get access to the Variables (collection) from the real-time platform.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Board	In	Board (Data Object)	None	Contains the Board to be used.
Variables	Out	Variables (Data Object)	None	Contains the Variables (collection) from the real-time platform.
OfflineVariables	In	Variables (Data Object)	None	Lets you specify the Variables (collection) to be used in offline operation mode.

Related topics

Basics

References

Sequences

Introduction

The Sequences folder provides automation blocks to manage collections of RTT sequences.

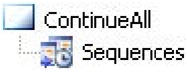
Where to go from here

Information in this section

RemoveSequenceByIndex..54
To remove a sequence from the collection.

ContinueAll

Graphical representation



Purpose

To continue the execution of all the sequences of one Sequences (collection) on a real-time platform.

Description

When you pause the RTT sequences on a real-time platform, you can continue the execution of all the RTT sequences at the point where they were paused.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) to be continued.

Related topics

Basics

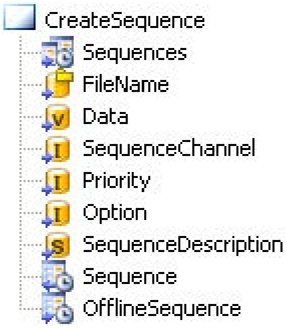
Basics on Real-Time Testing...9

References

PauseAll...51
PauseAll...93
Sequences..45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

CreateSequence

Graphical representation



CreateSequence
Sequences
FileName
Data
SequenceChannel
Priority
Option
SequenceDescription
Sequence
OfflineSequence

Purpose

To create a new RTT sequence on the real-time platform.

Description

Each RTT sequence created on the real-time platform has its own name space. To exchange values between different RTT sequences, you can use the globalvariables module, refer to rttlib.globalvariables Module (Real-Time Testing Library Reference 🕮).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) to which the new created sequence is added.
FileName	In	File	" "	Contains the name of the BCG file which is downloaded to the real-time platform.
Data	In	Variant	None	Python object that is passed to the running RTT sequence.
SequenceChannel	In	Integer	0	Defines when the RTT sequence is executed: ▪ 0 (`rttmanagerlib.constants.scPreComputation` of the rttmanagerlib module): The RTT sequence is executed before the simulation model is calculated by the real-time application. ▪ 1 (`rttmanagerlib.constants.scPostComputation` of the rttmanagerlib module): The RTT sequence is executed after the simulation model is calculated by the real-time application.
Priority	In	Integer	1	Priority of the RTT sequence in a range from 1 to 256 with 1 as the highest priority. The priority specifies the execution order of the RTT sequences. If RTT sequences have the same priority, they are executed in the reverse order in which they are

Name	In / Out	Type	Default Value	Description
				downloaded to the real-time platform. In a sampling step, the most recently created RTT sequence is then executed before older RTT sequences.
Option	In	Integer	0	Not supported in the current version of Real-Time Testing.
SequenceDescription	In	String	" "	User-defined description for the RTT sequence. You can read the description with the GetSequenceDescription block and it is displayed in the standalone user interface Real-Time Test Manager.
Sequence	Out	Sequence (Data Object)	None	Contains the created RTT Sequence.
OfflineSequence	In	Sequence (Data Object)	None	Lets you specify the created RTT Sequence to be used in offline operation mode.

Related topics

HowTos

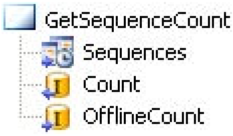
How to Initialize an RTT Project..20

References

Sequences..45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

GetSequenceCount

Graphical representation



Purpose

To get the number of RTT sequences in a specified collection.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) which is read out.
Count	Out	Integer	0	Contains the number of sequences in the Sequences (collection).
OfflineCount	In	Integer	0	Lets you specify a value to be used in offline operation mode.

Related topics

Basics

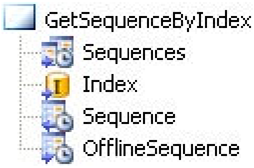
Basics on Real-Time Testing..9

References

Sequences..45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

GetSequenceByIndex

Graphical representation



GetSequenceByIndex
- Sequences
- Index
- Sequence
- OfflineSequence

Purpose

To return an RTT sequence by index.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) which is read out.
Index	In	Integer	0	Contains the index of the Sequence within the Sequences (collection).
Sequence	Out	Sequence (Data Object)	None	Contains the Sequence with the specified index.

Name	In / Out	Type	Default Value	Description
OfflineSequence	In	Sequence (Data Object)	None	Lets you specify the Sequence to be used in offline operation mode. Method of numbering: 0, 1, 2 ...

Related topics

Basics

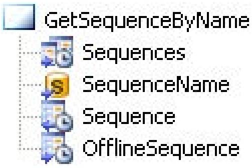
Basics on Real-Time Testing.. 9

References

Sequences.. 45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

GetSequenceByName

Graphical representation



Purpose

To return an RTT sequence by name.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) which is read out.
SequenceName	In	String	" "	Contains the name of the sequence within the Sequences (collection).
Sequence	Out	Sequence (Data Object)	None	Contains the Sequence with the specified name.
OfflineSequence	In	Sequence (Data Object)	None	Lets you specify the Sequence to be used in offline operation mode.

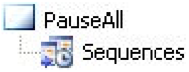
Related topics

Basics

References

PauseAll

Graphical representation



PauseAll
Sequences

Purpose

To pause all the RTT sequences running in the same sampling step on the real-time platform.

Description

The running RTT sequences are paused but not stopped. To continue the sequence execution at the point where it was paused, use the ContinueAll block.

You can also continue the execution of a single RTT sequence with the Continue block.

You can pause a single RTT sequence with the Pause on page 72 block.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Pauses all the sequences in a Sequences (collection).

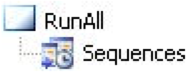
Related topics

Basics

References

RunAll

Graphical representation


RunAll
Sequences

Purpose

To start all the new RTT sequences on the real-time platform in the same sampling step.

Description

When you start the RTT sequences, all the sequences are executed on the real-time platform. RTT sequences that were already executed and do not have the New state are not run again. For information on the possible states, refer to Basics on Real-Time Testing on page 9.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Starts all the sequences in a Sequences (collection).

Related topics

HowTos

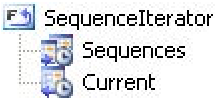
How to Execute a Real-Time Testing Project.. 21

References

Sequences.. 45
Sequences (Collection) (Real-Time Testing Library Reference 🕮)
StopAll.. 53

SequenceIterator

Graphical representation


SequenceIterator
Sequences
Current

Purpose

To execute an operation on every sequence of a Sequences (collection) in succession.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) from which an operation is executed on every sequence in succession.
Current	Out	Sequence (Data Object)	None	Contains the current iteration value. Method of numbering: 0, 1, 2 ...

Related topics

HowTos

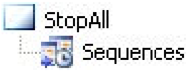
How to Execute a Real-Time Testing Project.. 21

References

Sequences.. 45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

StopAll

Graphical representation



StopAll
Sequences

Purpose

To stop all the RTT sequences running in the same sampling step on the real-time platform.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Stops all the sequences in a Sequences (collection).

Related topics

HowTos

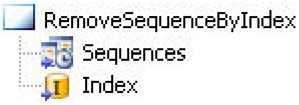
How to Execute a Real-Time Testing Project.. 21

References

RunAll.. 52
Sequences.. 45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

RemoveSequenceByIndex

Graphical representation



RemoveSequenceByIndex
Sequences
Index

Purpose

To remove a sequence from the collection.

Description

The RTT sequence of a real-time platform is deleted regardless of its state. For information on the possible states, refer to Basics on Real-Time Testing on page 9.

> **Note**
>
> You can remove the RTT sequence only if it is detached (see Detach on page 37).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequences	In	Sequences (Data Object)	None	Contains the Sequences (collection) from which the sequence is removed.
Index	In	Integer	0	Index of the sequence to be removed.

Related topics

Basics

Basics on Real-Time Testing 9

References

GetSequenceByIndex 49
Sequences 45
Sequences (Collection) (Real-Time Testing Library Reference 📖)

Variables

Introduction

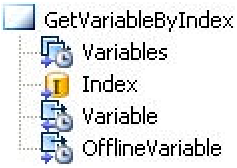
The Variables folder provides automation blocks to manage collections of RTT variables.

Where to go from here

Information in this section

GetVariableByIndex

Graphical representation



GetVariableByIndex
- Variables
- Index
- Variable
- OfflineVariable

Purpose

To return an RTT variable by index.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variables	In	Variables (Data Object)	None	Contains the Variables (collection) which is read out.
Index	In	Integer	0	Contains the index of the variable within the Variables (collection).
Variable	Out	Variable (Data Object)	None	Contains the Variable with the specified index.
OfflineVariable	In	Variable (Data Object)	None	Lets you specify the Variable to be used in offline operation mode.

Related topics

Basics

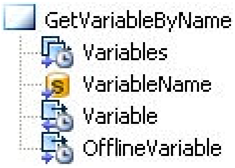
Basics on Real-Time Testing.. 9

References

Variables.. 55
Variables (Collection) (Real-Time Testing Library Reference 🕮)

GetVariableByName

Graphical representation


GetVariableByName
Variables
VariableName
Variable
OfflineVariable

Purpose

To return a variable by name.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variables	In	Variables (Data Object)	None	Contains the Variables (collection) which is read out.
VariableName	In	String	" "	Contains the name of the variable within the Variables (collection).
Variable	Out	Variable (Data Object)	None	Contains the Variable with the specified name.
OfflineVariable	In	Variable (Data Object)	None	Lets you specify the Variable to be used in offline operation mode.

Related topics

Basics

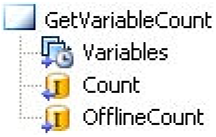
Basics on Real-Time Testing.. 9

References

Variables.. 55
Variables (Collection) (Real-Time Testing Library Reference 📖)

GetVariableCount

Graphical representation



GetVariableCount
Variables
Count
OfflineCount

Purpose

To get the number of items in a collection of variables.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variables	In	Variables (Data Object)	None	Contains the Variables (collection) which is read out.
Count	Out	Integer	0	Contains the number of variables in the Variables (collection).

Name	In / Out	Type	Default Value	Description
OfflineCount	In	Integer	0	Lets you specify a value to be used in offline operation mode.

Related topics

Basics

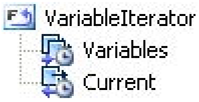
Basics on Real-Time Testing........ 9

References

Sequence........ 63
Variables (Collection) (Real-Time Testing Library Reference 📖)

VariableIterator

Graphical representation



VariableIterator
Variables
Current

Purpose

To execute an operation on every variable of a Variables (collection) in succession.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variables	In	Variables (Data Object)	None	Contains the Variables (collection) from which an operation is executed on every variable in succession.
Current	Out	Variable (Data Object)	None	Contains the current iteration value. Method of numbering: 0, 1, 2 ...

Related topics

Basics

Basics on Real-Time Testing........ 9

References

Variables........ 55
Variables (Collection) (Real-Time Testing Library Reference 📖)

DataStreams

Introduction

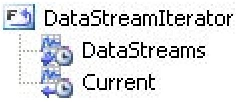
The DataStreams folder provides automation blocks to manage collections of RTT datastreams.

Where to go from here

Information in this section

DataStreamIterator

Graphical representation



DataStreamIterator
DataStreams
Current

Purpose

To execute an operation on every datastream of a DataStreams (collection) in succession.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStreams	In	DataStreams (Data Object)	None	Contains the DataStreams (collection) from which an operation is executed on very DataStream in succession.
Current	Out	DataStream (Data Object)	None	Contains the current iteration value. Method of numbering: 0, 1, 2 ...

Related topics

Basics

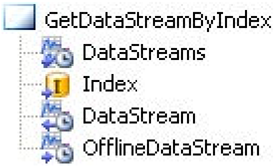
Basics on Real-Time Testing.. 9

References

DataStreams.. 59
rttlib.datastream Module (Real-Time Testing Library Reference 📖)

GetDataStreamByIndex

Graphical representation



GetDataStreamByIndex
DataStreams
Index
DataStream
OfflineDataStream

Purpose

To return an RTT datastream by index.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStreams	In	DataStreams (Data Object)	None	Contains the DataStreams (collection) which is read out.
Index	In	Integer	0	Contains the index of the DataStream within the DataStreams (collection).
DataStream	Out	DataStream (Data Object)	None	Contains the DataStream with the specified index.
OfflineDataStream	In	DataStream (Data Object)	None	Lets you specify the DataStream to be used in offline operation mode.

Related topics

Basics

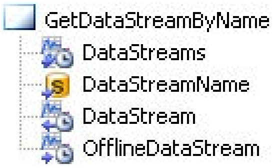
Basics on Real-Time Testing.. 9

References

DataStreams... 59
rttlib.datastream Module (Real-Time Testing Library Reference 📖)

GetDataStreamByName

Graphical representation



GetDataStreamByName
- DataStreams
- DataStreamName
- DataStream
- OfflineDataStream

Purpose

To return an RTT datastream by name.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStreams	In	DataStreams (Data Object)	None	Contains the DataStreams (collection) which is read out.
Index	In	Integer	0	Contains the name of the DataStream within the DataStreams (collection).
DataStream	Out	DataStream (Data Object)	None	Contains the DataStream with the specified name.
OfflineDataStream	In	DataStream (Data Object)	None	Lets you specify the DataStream to be used in offline operation mode.

Related topics

Basics

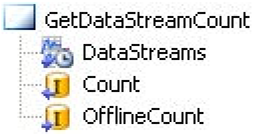
Basics on Real-Time Testing.......... 9

References

DataStreams.......... 59
rttlib.datastream Module (Real-Time Testing Library Reference 📖)

GetDataStreamCount

Graphical representation



GetDataStreamCount
DataStreams
Count
OfflineCount

Purpose

To get the number of items in a collection.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStreams	In	DataStreams (Data Object)	None	Contains the DataStreams (collection) which is read out.
Count	Out	Integer	0	Contains the number of datastreams in the DataStreams (collection).
OfflineCount	In	Integer	0	Lets you specify a value to be used in offline operation mode.

Related topics

Basics

Basics on Real-Time Testing.......... 9

References

DataStreams.......... 59
rttlib.datastream Module (Real-Time Testing Library Reference 📖)

Sequence

Introduction

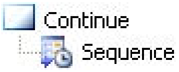
The Sequence folder provides automation blocks to manage single RTT sequences.

Where to go from here

Information in this section

Continue

Graphical representation



Continue
Sequence

Purpose

To continue the RTT sequence execution at the point where it was paused.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence to be continued.

Related topics

Basics

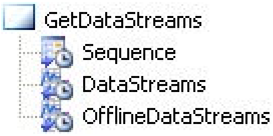
Basics on Real-Time Testing..9

References

Sequence...63
Sequence (Real-Time Testing Library Reference 📖)

GetDataStreams

Graphical representation



GetDataStreams
Sequence
DataStreams
OfflineDataStreams

Purpose

To get the datastreams of a sequence.

Description

A datastream is a MAT file whose data is replayed to variable objects in your RTT sequence. Data is streamed from the host PC to the real-time system during the run time of the RTT sequence. For further information on datastreaming, refer to Data Replay in RTT Sequences (Real-Time Testing Guide 📖).

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want the information.
DataStreams	Out	DataStreams (Data Object)	None	Contains the DataStreams (collection).
OfflineSequenceDataStreams	In	DataStreams (Data Object)	None	Lets you specify the DataStreams (collection) to be used in offline operation mode.

Related topics

Basics

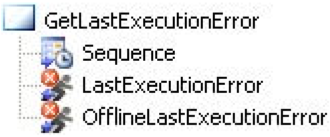
Basics on Real-Time Testing.. 9

References

Sequence... 63
Sequence (Real-Time Testing Library Reference 🕮)

GetLastExecutionError

Graphical representation



Purpose

To get information on errors occurring during sequence execution.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence on which you want information.
LastExecutionError	Out	ExecutionError (Data Object)	None	Contains information on whether an error occurred during sequence execution.

Name	In / Out	Type	Default Value	Description
OfflineLastExecutionError	In	ExecutionError (Data Object)	None	Lets you specify the value to be used in offline operation mode.

Related topics

Basics

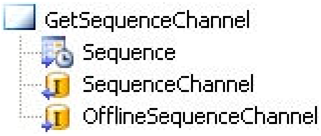
Basics on Real-Time Testing... 9

References

Sequence.. 63
Sequence (Real-Time Testing Library Reference 📖)

GetSequenceChannel

Graphical representation



Purpose

To get the sequence channel.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want the information.
SequenceChannel	Out	Integer	0	Contains information on when the RTT sequence is executed: ▪ 0 (`rttmanagerlib.constants.\scPreComputation` of the rttmanagerlib module): The RTT sequence is executed before the simulation model is calculated by the real-time application. ▪ 1 (`rttmanagerlib.constants.\scPostComputation` of the rttmanagerlib module): The RTT sequence is executed after the simulation model is calculated by the real-time application.

Name	In / Out	Type	Default Value	Description
OfflineSequenceChannel	In	Integer	0	Lets you specify the SequenceChannel to be used in offline operation mode.

Related topics

Basics

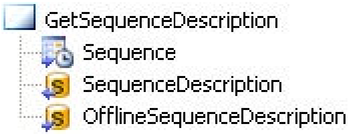
Basics on Real-Time Testing.. 9

References

Sequence.. 63
Sequence (Real-Time Testing Library Reference 📖)

GetSequenceDescription

Graphical representation



GetSequenceDescription
- Sequence
- SequenceDescription
- OfflineSequenceDescription

Purpose

To get the description of an RTT sequence.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want the information.
SequenceDescription	Out	String	" "	Contains the sequence description you specified when creating the RTT Sequence.
OfflineSequenceDescription	In	String	" "	Lets you specify the description of the Sequence to be used in offline operation mode.

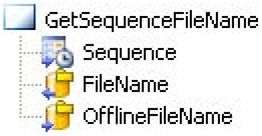
Related topics

Basics

References

GetSequenceFileName

Graphical representation



Purpose

To get the BCG file name of an RTT sequence.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want to read the BCG file name.
SequenceFileName	Out	File	" "	Contains the absolute BCG file name of the RTT sequence and its path.
OfflineSequenceFileName	In	File	" "	Lets you specify the BCG file name to be used in offline operation mode.

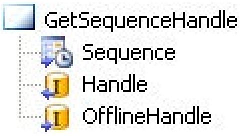
Related topics

Basics

References

GetSequenceHandle

Graphical representation



GetSequenceHandle
Sequence
Handle
OfflineHandle

Purpose

To get the handle identifier value of an RTT sequence on the real-time platform.

Description

The handle identifier value is assigned to each sequence by the hardware platform and is unique.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence you want to handle.
SequenceHandle	Out	Integer	0	Contains the handle identifier value.
OfflineSequenceHandle	In	Integer	0	Lets you specify the value to be used in offline operation mode.

Related topics

Basics

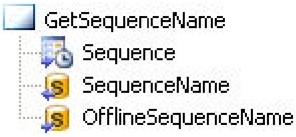
Basics on Real-Time Testing.. 9

References

Sequence.. 63
Sequence (Real-Time Testing Library Reference 📖)

GetSequenceName

Graphical representation



GetSequenceName
Sequence
SequenceName
OfflineSequenceName

Purpose

To get the name of an RTT sequence.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want the information.
SequenceName	Out	String	" "	Contains the sequence's name of the RTT sequence.
OfflineSequenceName	In	String	" "	Lets you specify the sequence's name to be used in offline operation mode.

Related topics

Basics

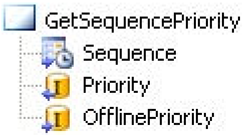
Basics on Real-Time Testing.......... 9

References

Sequence.......... 63
Sequence (Real-Time Testing Library Reference 🕮)

GetSequencePriority

Graphical representation



GetSequencePriority
Sequence
Priority
OfflinePriority

Purpose

To get the RTT sequence's position in the priority list.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want the information.

Name	In / Out	Type	Default Value	Description
Priority	Out	Integer	1	Contains the priority information as a value in the range 1 ... 256 with 1 as the highest priority.
OfflinePriority	In	Integer	0	Lets you specify the value to be used in offline operation mode.

Related topics

Basics

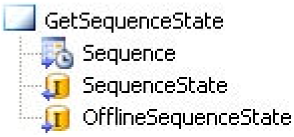
Basics on Real-Time Testing 9

References

Sequence 63
Sequence (Real-Time Testing Library Reference 🕮)

GetSequenceState

Graphical representation



GetSequenceState
Sequence
SequenceState
OfflineSequenceState

Purpose

To get the state of an RTT sequence.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence for which you want the information.
SequenceState	Out	Integer	0	Contains the state of the RTT sequence: ▪ 0 = New RTT sequence was created (`rttmanagerlib.constants.sesNew` of the rttmanagerlib module). ▪ 1 = RTT sequence was executed completely and without errors (`rttmanagerlib.constants.sesTerminated` of the rttmanagerlib module).

Name	In / Out	Type	Default Value	Description
				▪ 2 = Error when creating or executing RTT sequences `rttmanagerlib.constants.sesError` of the rttmanagerlib module). ▪ 3 = RTT sequence was stopped (`rttmanagerlib.constants.sesStopped` of the rttmanagerlib module). ▪ 4 = RTT sequence is running (`rttmanagerlib.constants.sesRunning` of the rttmanagerlib module). ▪ 5 = RTT sequence is paused (`rttmanagerlib.constants.sesPaused` of the rttmanagerlib module).
OfflineSequenceState	In	Integer	0	Lets you specify the sequence's state to be used in offline operation mode.

Related topics

Basics

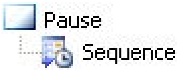
Basics on Real-Time Testing.. 9

References

Sequence.. 63
Sequence (Real-Time Testing Library Reference 🕮)

Pause

Graphical representation



Purpose

To pause a running RTT sequence.

Description

You can continue the paused sequence with the Continue block.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence to be paused.

Related topics

Basics

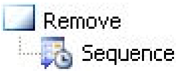
Basics on Real-Time Testing 9

References

Continue 64
Continue 89
ContinueAll 46
Sequence 63
Sequence (Real-Time Testing Library Reference 🕮)
Stop 95

Remove

Graphical representation



Remove
Sequence

Purpose

To remove an RTT sequence from the real-time platform.

Description

When removing the RTT sequence, the `Detach` method is automatically called beforehand.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence to be removed.

Related topics

Basics

Basics on Real-Time Testing 9

References

RemoveSequenceByIndex 54
Sequence 63
Sequence (Real-Time Testing Library Reference 🕮)

Run

Graphical representation

Purpose

To start an RTT sequence on the real-time platform.

Description

When you execute the Run block, the sequence is executed on the real-time platform. An RTT sequence can be started if it has one of the following states:

- New
- Paused
- Stopped
- Terminated

When an RTT sequence is started with the Paused, Stopped, or Terminated state, its namespace is maintained. The sequence is not initialized but immediately starts by executing the MainGenerator function.

You can add data objects to the Run block to pass parameters to the RTT sequence when starting the sequence execution.

You can start all the RTT sequences of a real-time platform which have the New state in one step with the RunAll block.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence to be started.
<run_parameter>	In	Any (Usually String, Int or Float)	–	Contains the arguments for the MainGenerator function.

Related topics

HowTos

How to Execute a Real-Time Testing Project.. 21

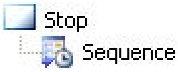
References

Continue.. 64
Continue.. 89
New Data Object (AutomationDesk Basic Practices 📖)
Pause.. 72
Pause.. 93
Sequence.. 63

Sequence (Real-Time Testing Library Reference 🕮)
Stop.......75
Stop.......95

Stop

Graphical representation



Purpose

To stop a running RTT sequence.

Description

When you stop a running RTT sequence, the sequence execution stops but the RTT sequence remains on the real-time platform.

You can stop all the RTT sequences of a real-time platform in one step with the StopAll block.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Sequence	In	Sequence (Data Object)	None	Contains the Sequence to be stopped.

Related topics

Basics

Basics on Real-Time Testing.......9

References

Continue.......64
Run.......94
RunAll.......95
Sequence.......63
Sequence (Real-Time Testing Library Reference 🕮)

Variable

Introduction

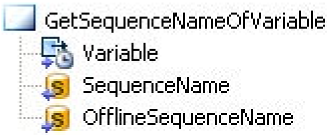
The Variable folder provides automation blocks to manage single RTT variables.

Where to go from here

Information in this section

GetSequenceNameOfVariable

Graphical representation



GetSequenceNameOfVariable
Variable
SequenceName
OfflineSequenceName

Purpose

To get the name of the RTT sequence that contains the RTT variable.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variable	In	Variable (Data Object)	None	Contains the Variable for which you want the information.

Name	In / Out	Type	Default Value	Description
SequenceName	Out	String	" "	Contains the name of the sequence.
OfflineSequenceName	In	String	" "	Lets you specify the name of the sequence to be used in offline operation mode.

Related topics

Basics

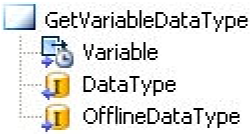
Basics on Real-Time Testing.. 9

References

Variable.. 76
Variable (Real-Time Testing Library Reference 📖)

GetVariableDataType

Graphical representation



GetVariableDataType
Variable
DataType
OfflineDataType

Purpose

To get the data type of an RTT variable.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variable	In	Variable (Data Object)	None	Contains the Variable for which you want the information.
DataType	Out	Integer	0	Contains the data type of the variable.
OfflineDataType	In	Integer	0	Lets you specify the data type to be used in offline operation mode.

Related topics

Basics

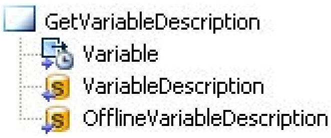
Basics on Real-Time Testing.. 9

References

Variable.. 76
Variable (Real-Time Testing Library Reference 🕮)

GetVariableDescription

Graphical representation



GetVariableDescription
- Variable
- VariableDescription
- OfflineVariableDescription

Purpose

To read the description of an RTT variable.

Data objects

This automation block provides the following data objects:

Name	**In / Out**	**Type**	**Default Value**	**Description**
Variable	In	Variable (Data Object)	None	Contains the Variable for which you want the information.
VariableDescription	Out	String	" "	Contains the description of the variable.
OfflineVariableDescription	In	String	" "	Lets you specify the description to be used in offline operation mode.

Related topics

Basics

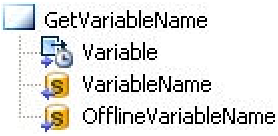
Basics on Real-Time Testing.. 9

References

Variable.. 76
Variable (Real-Time Testing Library Reference 🕮)

GetVariableName

Graphical representation



GetVariableName
- Variable
- VariableName
- OfflineVariableName

Purpose

To read the name of an RTT variable.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variable	In	Variable (Data Object)	None	Contains the Variable for which you want the information.
VariableName	Out	String	" "	Contains the name of the variable.
OfflineVariableName	In	String	" "	Lets you specify the name to be used in offline operation mode.

Related topics

Basics

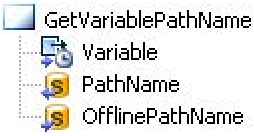
Basics on Real-Time Testing.. 9

References

Variable.. 76
Variable (Real-Time Testing Library Reference 📖)

GetVariablePathName

Graphical representation



GetVariablePathName
- Variable
- PathName
- OfflinePathName

Purpose

To get the path of an RTT variable.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variable	In	Variable (Data Object)	None	Contains the Variable for which you want the information.
PathName	Out	String	" "	Contains the path.
OfflinePathName	In	String	" "	Lets you specify the path to be used in offline operation mode.

Related topics

Basics

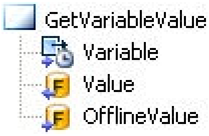
Basics on Real-Time Testing..9

References

Variable..76
Variable (Real-Time Testing Library Reference 📖)

GetVariableValue

Graphical representation



Purpose

To get the value of an RTT variable.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variable	In	Variable (Data Object)	None	Contains the Variable for which you want the information.
Value	Out	Float	0.0	Contains the value of the variable.
OfflineValue	In	Float	0.0	Lets you specify the value to be used in offline operation mode.

Related topics

Basics

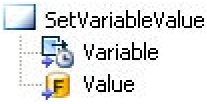
Basics on Real-Time Testing.. 9

References

Variable.. 76
Variable (Real-Time Testing Library Reference 📖)

SetVariableValue

Graphical representation



SetVariableValue
Variable
Value

Purpose

To set a value of an RTT variable.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
Variable	In	Variable (Data Object)	None	Contains the Variable whose value you want to set.
Value	In	Float	0.0	Contains the value to be set.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

Variable.. 76
Variable (Real-Time Testing Library Reference 📖)

DataStream

Introduction

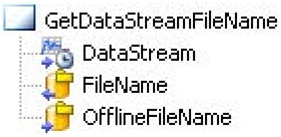
The DataStream folder provides automation blocks to manage single RTT datastreams.

Where to go from here

Information in this section

GetDataStreamFileName

Graphical representation



GetDataStreamFileName
DataStream
FileName
OfflineFileName

Purpose

To read the file name of an RTT datastream.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStream	In	DataStream (Data Object)	None	Contains the DataStream for which you want to get the information.
FileName	Out	File	" "	Contains the file name and its path.
OfflineFileName	In	File	" "	Lets you specify the file name to be used in offline operation mode.

Related topics

Basics

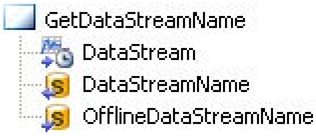
Basics on Real-Time Testing..9

References

Variable..76

GetDataStreamName

Graphical representation



GetDataStreamName
DataStream
DataStreamName
OfflineDataStreamName

Purpose

To read the name of an RTT datastream.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStream	In	DataStream (Data Object)	None	Contains the DataStream for which you want to get the information.
DataStreamName	Out	String	" "	Contains the name of the datastream.
OfflineDataStreamName	In	String	" "	Lets you specify the name of the datastream to be used in offline operation mode.

Related topics

Basics

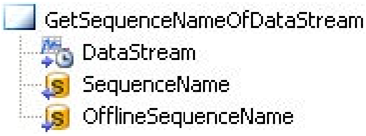
Basics on Real-Time Testing..9

References

Variable..76

GetSequenceNameOfDataStream

Graphical representation



GetSequenceNameOfDataStream
- DataStream
- SequenceName
- OfflineSequenceName

Purpose

To get the name of the RTT sequence that contains the RTT datastream.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
DataStream	In	DataStream (Data Object)	None	Contains the DataStream for which you want to get the information.
SequenceName	Out	String	" "	Contains the name of the sequence.
OfflineSequenceName	In	String	" "	Lets you specify the name of the sequence to be used in offline operation mode.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

Variable.. 76

ExecutionError

Introduction

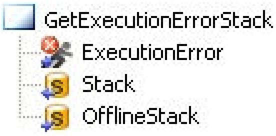
The ExecutionError folder provides automation blocks to manage information on the last errors occurring during sequence execution.

Where to go from here

Information in this section

GetExecutionErrorStack

Graphical representation



Purpose

To get traceback information from a specified error stack.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ExecutionError	In	ExecutionError (Data Object)	None	Contains the error on which you want the information.

Stack	Out	String	" "	Contains the traceback information from a specified error stack.
				For example:
				`Traceback (File "<interactive input>",\ line 1`
				`import os.path.dirname(12)`
				`SyntaxError: invalid syntax`

OfflineStack	In	String	" "	Lets you specify a value to be used in offline operation mode.

Related topics

Basics

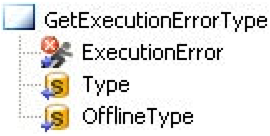
Basics on Real-Time Testing...9

References

ExecutionError...84
ExecutionError (Real-Time Testing Library Reference 🕮)

GetExecutionErrorType

Graphical representation



GetExecutionErrorType
- ExecutionError
- Type
- OfflineType

Purpose

To get the exception type of an error.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ExecutionError	In	ExecutionError (Data Object)	None	Contains the error for which you want the information.
Type	Out	String	" "	Contains the exception type, for example, ZeroDivisionError.
OfflineType	In	String	" "	Lets you specify the value to be used in offline operation mode.

Related topics

Basics

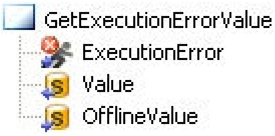
Basics on Real-Time Testing..9

References

ExecutionError..84
ExecutionError (Real-Time Testing Library Reference 📖)

GetExecutionErrorValue

Graphical representation



GetExecutionErrorValue
- ExecutionError
- Value
- OfflineValue

Purpose

To get exception information on an error.

Data objects

This automation block provides the following data objects:

Name	In / Out	Type	Default Value	Description
ExecutionError	In	ExecutionError (Data Object)	None	Contains the error for which you want the information.
Value	Out	String	" "	Contains the value of the error, for example, integer division or modulo by zero.
OfflineValue	In	String	" "	Lets you specify the value to be used in offline operation mode.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

ExecutionError.. 84

ExecutionError (Real-Time Testing Library Reference 📖)

Commands And Dialogs

Where to go from here

Information in this section

Attach

Access

You can access this command via:

Ribbon	None
Context menu of	ManagerServer data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To attach the ManagerServer data object to the Real-Time Testing Manager Server.

Result

The Real-Time Testing Manager Server is started, if it was not already started by another real-time task outside AutomationDesk, and your automation project is attached to it.

Description

The ManagerServer data object must be attached first in your Real-Time Testing automation task. You can do this manually via the Attach command or via the Attach block in your automation sequence.

Related topics

Basics

Basics on Real-Time Testing... 9

References

Attach.. 41
Detach.. 91

Continue

Access

You can access this command via:

Ribbon	None
Context menu of	Sequence data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To continue the RTT sequence execution at the point where it was paused.

Description

You can continue a paused RTT sequence manually via the Continue command or via the Continue block in your automation sequence.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

Continue.. 64
Pause.. 93

ContinueAll

Access

You can access this command via:

Ribbon	None
Context menu of	Sequences data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To continue the execution of all the sequences of one Sequences (collection) on a real-time platform.

Description

You can continue the paused RTT sequences of a Sequences (collection) manually via the ContinueAll command or via the ContinueAll block in your automation sequence.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

ContinueAll.. 46
PauseAll.. 93

Detach

Access

You can access this command via:

Ribbon	None
Context menu of	The following data objects in the Project Manager: ▪ ManagerServer ▪ Board ▪ Sequences ▪ Sequence ▪ Variables ▪ Variable ▪ Datastreams ▪ Datastream ▪ ExecutionError
Shortcut key	None
Icon	None

Purpose

To detach a data object from the Real-Time Testing Manager Server.

Description

You can detach the data object manually via the Detach command or via the Detach block in your automation sequence.

Note

When you detach the ManagerServer data object, all other data objects are still attached to the RTT Manager Server. This can cause faults when you execute your AutomationDesk project again. It is recommended to use the Detach automation block at the end of your sequence to detach all data objects used.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

Detach.. 37

Insert (Real-Time Testing Elements)

Access

You can access the data objects of the Real-Time Testing library via:

Ribbon	Home - Insert - Data Objects
Context menu of	None
Shortcut key	None
Icon	▪ ManagerServer ▪ Board ▪ Sequences ▪ Variables ▪ DataStreams ▪ Sequence ▪ Variable ▪ DataStream ▪ ExecutionError

Purpose

To insert data objects from the Real-Time Testing library using the Home ribbon.

Description

The ribbon commands are enabled or disabled according to context.

You can add data objects to your project in the Project Manager and to specific automation blocks in the Sequence Builder, such as Exec automation blocks. The data object is added to the selected element.

Related topics

Basics

Basics on Real-Time Testing... 9

References

Real-Time Test Manager... 29

Pause

Access

You can access this command via:

Ribbon	None
Context menu of	Sequence data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To pause the execution of an RTT sequence.

Description

You can pause the execution of an RTT sequence manually via the Continue command or via the Pause block in your automation sequence.

Related topics

Basics

Basics on Real-Time Testing.. 9

References

Continue.. 89
Pause... 72

PauseAll

Access

You can access this command via:

Ribbon	None
Context menu of	Sequences data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To pause the execution of all the sequences of one Sequences (collection) on a real-time platform.

Description

You can pause the execution of RTT sequences of a Sequences (collection) manually via the PauseAll command or via the PauseAll block in your automation sequence.

Related topics

Basics

Basics on Real-Time Testing	9

References

ContinueAll	90
PauseAll	51

Run

Access

You can access this command via:

Ribbon	None
Context menu of	Sequence data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To start a sequence on the real-time platform.

Description

You can start an RTT sequence manually via the Run command or via the Run block in your automation sequence.

Related topics

HowTos

How to Execute a Real-Time Testing Project	21

References

Run	74
Stop	95

RunAll

Access

You can access this command via:

Ribbon	None
Context menu of	Sequences data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To start the execution of all sequences of one Sequences (collection) on a real-time platform.

Description

You can start the RTT sequences of a Sequences (collection) manually via the RunAll command or via the RunAll block in your automation sequence.

Related topics

HowTos

How to Execute a Real-Time Testing Project... 21

References

RunAll.. 52
StopAll.. 96

Stop

Access

You can access this command via:

Ribbon	None
Context menu of	Sequence data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To stop a sequence on the real-time platform.

Description

You can stop an RTT sequence manually via the Stop command or via the Stop block in your automation sequence.

Related topics

HowTos

How to Execute a Real-Time Testing Project........21

References

Run........94
Stop........75

StopAll

Access

You can access this command via:

Ribbon	None
Context menu of	Sequences data object in the Project Manager
Shortcut key	None
Icon	None

Purpose

To stop the execution of all the sequences of one Sequences (collection) on a real-time platform.

Description

You can stop the RTT sequences of a Sequences (collection) manually via the StopAll command or via the StopAll block in your automation sequence.

Related topics

HowTos

How to Execute a Real-Time Testing Project........21

References

RunAll........95
StopAll........53

Automation

Basics on Automating the Access to Real-Time Testing

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 Real-Time Testing. 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.