

ModelDesk

# Processing

For ModelDesk 5.5

Release 2021-A – May 2021

## How to Contact dSPACE

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

## 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](mailto:support@dspace.de)

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

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

## Software Updates and Patches

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

## Important Notice

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

© 2006 - 2021 by:  
dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

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

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

# Contents

## About This Document 9

## Basics and Instructions 11

Basics and Workflow.....	12
Basics of Processing.....	12
Workflow for Processing.....	14
Preparing Measurement Data for Processing.....	16
Raw Data, Measurement Types, and Measurement Data.....	16
How to Specify Measurement Types.....	18
How to Get Measurement Data by Mapping Raw Data.....	21
How to Calculate Measurement Data.....	25
Executing Processing.....	30
Managing Measurement, Setting, Function, and Additional Function Files.....	30
How to Structure the Pool for Setting and Function Files.....	32
Working with the Processing Configuration Pane.....	33
How to Map Setting and Function Files to Parameters.....	35
How to Specify the Execution Sequence of Function Files.....	38
How to Specify the Plotting Details.....	39
How to Start Calculating the Parameter Values.....	40

## Reference Information 43

Processing Commands.....	44
Activate (Measurement Data).....	45
Append Measurement Type Variable / Append.....	46
Close Plots.....	46
Configure.....	47
Create Folder.....	48
Export Mapping.....	48
Import Mapping.....	49
Insert (Measurement Type).....	50
New (Measurement Data).....	51
New (Measurement Type).....	51
New Conversion File (Processing).....	52
Open.....	53

Open from Pool (Measurement Data).....	53
Open from Pool (Measurement Type).....	54
Rename Folder.....	55
Remove (Processing).....	56
Remove Folder.....	56
Save (Processing).....	57
Save All (Processing).....	58
Save As (Processing).....	58
Processing Dialogs, Panes, and Pages.....	60
Add Measurement Type Variable.....	60
Configure Measurement Types for Experiment.....	61
Measurement Data.....	62
Measurement Data Variable Mapping Import Conflicts.....	65
Measurement Type Pane.....	66
Measurement Types Page.....	68
Measurement Type Container Page.....	68
Processing Page.....	69
Processing Configuration.....	69
Specify Raw Data – Measurement Data Source.....	76
Processing Properties.....	77
Properties of Additional Functions.....	77
Measurement Data Properties.....	78
Measurement Type Properties.....	80
Parameter Properties.....	80
Parameter Page Properties.....	82
Variable Mapping Details.....	84

## Automation 87

Programming ModelDesk Automation.....	88
Automatic Processing.....	88
Overview of the Object Model for Processing.....	92
Classes for Processing.....	94
ActiveMeasurementData.....	95
Class Description (ActiveMeasurementData).....	96
ChangeConversionFile.....	97
Close.....	98
CreateCopy.....	99
Save.....	99
SpecifyRawData.....	100

ActiveMeasurementTypes.....	101
Class Description (ActiveMeasurementTypes).....	101
Item.....	102
AdditionalFunction.....	102
Class Description (AdditionalFunction).....	103
Execute.....	104
AdditionalFunctions.....	104
Class Description (AdditionalFunctions).....	105
Add.....	105
Item.....	106
Remove.....	107
ConversionFile.....	108
Class Description (ConversionFile).....	108
Save.....	109
ConversionRule.....	109
Class Description (ConversionRule).....	109
ConversionRules.....	111
Class Description (ConversionRules).....	111
Find.....	112
Item.....	112
MappingConfiguration.....	113
Class Description (MappingConfiguration).....	113
AutoMap.....	114
Item.....	115
MappingVariable.....	116
Class Description (MappingVariable).....	116
MapVariable.....	117
RemoveMapping.....	118
MeasurementConfiguration.....	118
Class Description (MeasurementConfiguration).....	118
Item.....	119
MeasurementFunction.....	120
Class Description (MeasurementFunction).....	120
DeleteValues.....	121
Execute.....	122
MeasurementType.....	122
Class Description (MeasurementType).....	123
Close.....	124

CreateCopy.....	124
Save.....	125
MeasurementTypeConfiguration.....	126
Class Description (MeasurementTypeConfiguration).....	126
ActivateMeasurementType.....	127
AddMeasurementType.....	128
DeactivateMeasurementType.....	128
RemoveMeasurementType.....	129
MeasurementTypeContainer.....	130
Class Description (MeasurementTypeContainer).....	130
ActivateMeasurementData.....	131
AddMeasurementData.....	132
DeactivateMeasurementData.....	132
RemoveMeasurementData.....	133
MeasurementTypeVariable.....	133
Class Description (MeasurementTypeVariable).....	134
MeasurementTypeVariables.....	134
Class Description (MeasurementTypeVariables).....	135
Append.....	136
ChangeOrder.....	136
Item.....	137
Remove.....	138
PlotConfiguration.....	138
Class Description (PlotConfiguration).....	139
ClosePlots.....	139
ParameterProcessingInfo.....	140
Class Description (ParameterProcessingInfo).....	140
Execute.....	141
ParameterRecordProcessingInfo.....	142
Class Description (ParameterRecordProcessingInfo).....	142
Execute.....	143
ParameterSetProcessing.....	144
Class Description (ParameterSetProcessing).....	144
Execute.....	145
ProcessingConfiguration.....	145
Class Description (ProcessingConfiguration).....	145

RawData.....	146
Class Description (RawData).....	146
ReloadRawData.....	147
RawDataVariables.....	148
Class Description (RawDataVariables).....	148
Item.....	149
RawDataVariable.....	149
Class Description (RawDataVariable).....	150
Enumerations for Processing.....	150
Enumerations for Processing.....	150
 Index.....	 153













# About This Document

## Contents

This document introduces you to the calculation of parameter values based on measurement in MotionDesk (named processing).

## Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
 <b>DANGER</b>	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
 <b>WARNING</b>	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
 <b>CAUTION</b>	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
 <b>NOTICE</b>	Indicates a hazard that, if not avoided, could result in property damage.
 <b>Note</b>	Indicates important information that you should take into account to avoid malfunctions.
 <b>Tip</b>	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

## Naming conventions

dSPACE user documentation uses the following naming conventions:

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

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

---

## Special folders

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

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

or

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

**Documents folder** A standard folder for user-specific documents.

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

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

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

---

## Accessing dSPACE Help and PDF Files


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

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

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

**dSPACE Help (Web)** You can access the Web version of dSPACE Help at [www.dspace.com/go/help](http://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

<a href="#">Basics and Workflow.....</a>	<a href="#">12</a>
To get basic information and an overview of the workflow.	
<a href="#">Preparing Measurement Data for Processing.....</a>	<a href="#">16</a>
Before you can calculate the parameter values in ModelDesk, you must prepare raw data so that it can be used as measurement data. You must specify measurement types which contain the variables for the calculation.	
<a href="#">Executing Processing.....</a>	<a href="#">30</a>
When the measurement data is specified and the function and settings files are written, you can execute Processing.	

# Basics and Workflow

## Where to go from here

## Information in this section

<a href="#">Basics of Processing.....</a>	<a href="#">12</a>
Provides basic information on the Processing component.	
<a href="#">Workflow for Processing.....</a>	<a href="#">14</a>
Provides an overview of processing.	

## Basics of Processing

### Processing

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

### Features

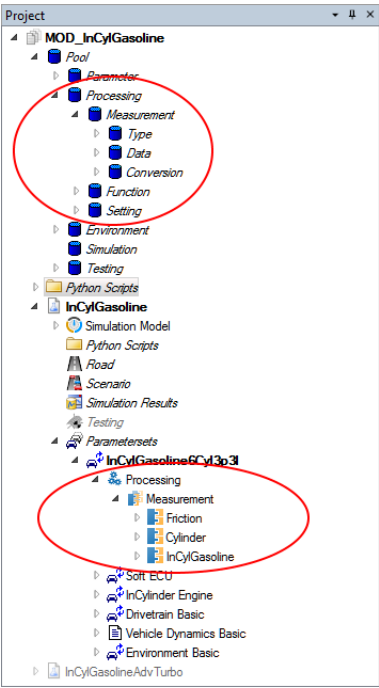
ModelDesk's Processing component helps you to prepare the measurement data. You can adapt raw data from the measurements to the measurement data used for parameter calculation. As an alternative to using raw data, you can calculate the measurement data in MATLAB. The parameter values are calculated in MATLAB by using setting and function files. Then ModelDesk can write the calculated values to the parameter pages of the model.

ModelDesk's Processing component provides the following features:

- Loading Excel files containing the raw data, for example, data measured on an engine test bench
- Specifying measurement types containing variables that are used for the calculation of the parameter values
- Mapping raw data to the variables of a measurement type and adapting their units to get measurement data that can be used in the calculation
- Mapping functions to the variables of a measurement type
- Handling several variations of measurement types configured in the experiment
- Handling several instances of measurement data files, one of them can be activated for each measurement type in every parameter set
- Managing M files for MATLAB. M files are used for:
  - Function files to calculate variables of a measurement type
  - Setting files to set initial values for calculating parameter values
  - Function files to calculate the parameter values
  - Additional functions for parameter-independent calculations

- Triggering MATLAB to calculate parameter values by using M files managed by ModelDesk. The sequence of the M file calculation is specified in ModelDesk.
- Storing setting and function files in ModelDesk's Pool

The following illustration shows the Project Navigator with the elements for processing.



Related topics

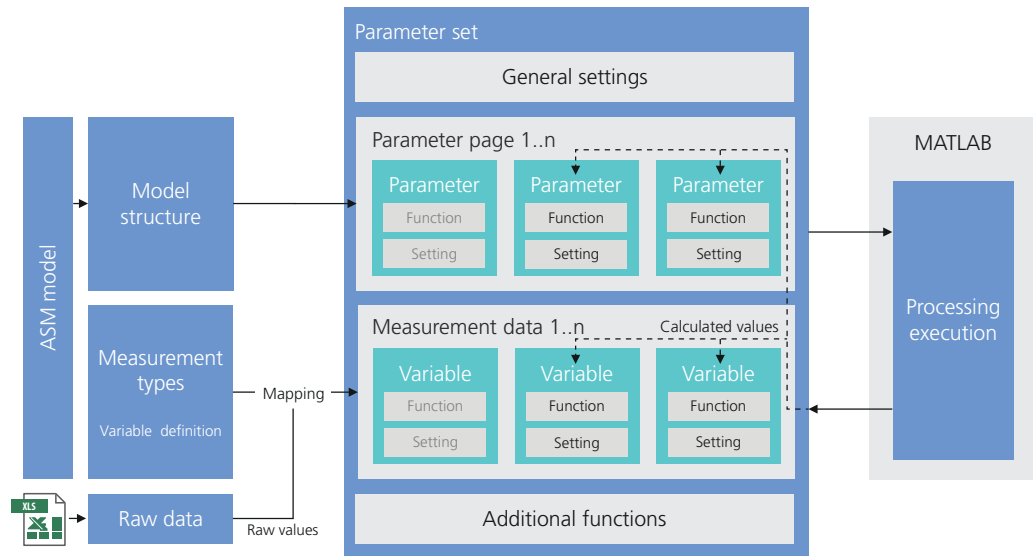
Basics

Workflow for Processing.....	14
------------------------------	----

## Workflow for Processing

### Overview

The following illustration shows the processing workflow.



### Workflow

The following steps are necessary to get measurement data:

1. Store the measured values in Excel™ files in XLSX format. ModelDesk can read data from an Excel file when it contains the data in the required format. Refer to [Raw Data, Measurement Types, and Measurement Data](#) on page 16.
2. Specify measurement types with the variables that are used for the calculation in MATLAB. Refer to [How to Specify Measurement Types](#) on page 18.
3. Specify the measurement data.
  - To do this, map the raw data to the variables of a measurement type. For each mapping, you can specify a factor and offset to adapt the units of the raw data and variables. Refer to [How to Get Measurement Data by Mapping Raw Data](#) on page 21.
  - As an alternative, you can write and map functions to the variables of a measurement type. Refer to [How to Calculate Measurement Data](#) on page 25.

The following steps are necessary for processing:

1. Create the files that calculate the parameter values in MATLAB. Four kinds of files can be used: a general settings file, setting files, function, and additional function files. ModelDesk can manage these files in the Pool. For details on managing the files, refer to [Managing Measurement, Setting, Function, and Additional Function Files](#) on page 30.

For details on how to write the setting and function files, refer to [ModelDesk Processing \(ASM User Guide\)](#).

- 2. Map the setting and function files to the parameter. You can map one setting file and one function file to a parameter. For instructions, refer to [How to Map Setting and Function Files to Parameters](#) on page 35.
- 3. Specify the execution sequence of the M files. This is important if parameter values depend on other calculated parameter values. For details, refer to [How to Specify the Execution Sequence of Function Files](#) on page 38.
- 4. Start the processing execution. You can evaluate one parameter value or additional function, the parameters of one parameter page, or all the parameters of the model. MATLAB executes the selected M files. For instructions, refer to [How to Start Calculating the Parameter Values](#) on page 40.

Related topics

Basics	
Basics of Processing.....	12

# Preparing Measurement Data for Processing

## Introduction

Before you can calculate the parameter values in ModelDesk, you must prepare raw data so that it can be used as measurement data. You must specify measurement types which contain the variables for the calculation.

## Where to go from here

## Information in this section

[Raw Data, Measurement Types, and Measurement Data..... 16](#)  
Provides information on raw data, measurement types, and measurement data.

[How to Specify Measurement Types..... 18](#)  
To prepare raw data for processing, you must specify measurement types.

[How to Get Measurement Data by Mapping Raw Data..... 21](#)  
To get measurement data, you can map the variables of raw data to measurement type variables. The units can be adjusted.

[How to Calculate Measurement Data..... 25](#)  
To get measurement data, you can calculate their values using MATLAB functions.

# Raw Data, Measurement Types, and Measurement Data

## Introduction

Measurement data contains the values of raw data which is prepared by using measurement types.

## Raw data

Raw data is data that is measured on a physical system under consideration and stored in a file. Raw data must be provided to ModelDesk in the form of an Excel™ file in XLSX format. The Excel file must have a specific structure.

During measurement, several variables are measured and stored in a file. To use them for processing, the file must be converted into the Excel file format. To be used in ModelDesk, the Excel file must comply with the following rules:

- One Excel sheet must contain all the variables data that is used for one measurement type.
- One row must contain names of the variables.
- One row must contain units of the variables.
- The measurement values must be in columns.



- The rows containing names and units must come before the rows containing the measurement values.

**Example** The following table shows an example of raw data:

Name Data 1	Name Data 2	Name Data 3	...
Unit Data 1	Unit Data 2	Unit Data 3	...
Value 1 Data 1	Value 1 Data 2	Value 1 Data 3	...
Value 2 Data 1	Value 2 Data 2	Value 2 Data 3	...
Value 3 Data 1	Value 3 Data 2	Value 3 Data 3	...
...	...	...	...

### Measurement type

A measurement type is a collection of variables which are specified for the calculation of parameter values. The variables have the following attributes:

- Name
- Unit
- Description
- Default value
- Lower limit
- Upper limit

### Measurement data

Measurement data is a collection of the variables with measured, calculated, or default values. These variables are used in the calculation during processing. The individual measurement data must all have the same number of variables.

The values of the measurement variables can come from 3 different sources:

- You can use values that were measured before. To do this, map raw data coming from measurements to variables specified with a measurement type (see above). To adjust their units, you can specify offset values and factors.
- You can use measurement functions to calculate the values. A measurement function contains MATLAB M files to calculate the variables (function file) and set initial values (setting file), a priority to specify the execution order and a flag for activation. To calculate the measurement variables, write the M files and map them to the measurement variables.

All of the measurement functions must return the same number of values so that the variables can be used for processing. If you use calculated and measured values, the functions must return the same number of values as measured values.

You can activate or deactivate the use of functions.

- If measured or calculated values are not available, the default values specified for the measurement type variable are used.

The following table shows the cases in which the sources are used for processing.

Function Assigned	Function Active	Raw Data Mapped	Existing Calculated Values	Values Used for Processing
✓	✓	✓	✓	Calculated
✓	✓	–	✓	Calculated
✓	–	✓	✓	Calculated
–	–	✓	✓	Calculated
✓	–	–	✓	Calculated
–	–	–	✓	Calculated
✓	✓	✓	–	Default
✓	✓	–	–	Default
✓	–	✓	–	Default
✓	–	–	–	Default
–	–	–	–	Default
–	–	✓	–	Raw Data

## Related topics

### Basics

Basics of Processing..... 12

## How to Specify Measurement Types

### Objective

To prepare raw data for processing, you must specify measurement types.

### Specifying measurement types manually

When you specify the measurement type variables, you can insert or append new variables. The position at which you use the command affects the order of the variables.

#### Tip

The mapping of the measurement type variables to the variables of the raw data is easier when the order of the variables is equal.  
If variable name and variable unit of the measurement type are identical to raw name and raw unit used in the Excel file, ModelDesk can assist you when the variables are mapped.

You can copy, cut, paste, or delete variables. Because names must be unique, you must modify the name when you copy & paste a variable.

The name of the measurement type is specified when it is saved.

**Restrictions** When you create the measurement type variables, you must consider the following restrictions:

- The properties of a variable cannot be modified after the variable was created. If a property, such as the unit, is not correct, you must delete the variable and insert a new one.
- The order of the variables cannot be modified. You must create the variables in the correct order.

### Specifying measurement types with Excel or word processing program

It is not necessary to specify all the variables manually. You can copy & paste the variables and their properties from Excel or a word processing program to the table. So it is possible to specify several variables at once.

Data must be in a table form: One row for each variable, the columns must contain the variable properties in the following order:

- Variable name (MATLAB-compatible)
- Unit
- Default value
- Lower limit
- Upper limit
- Description

**Example** The following table shows an example of two variables:

Variable1	m	1	0	1000	Description of variable 1
Variable2	grad	1	0	360	Description of variable 2

### Possible methods

There are two different methods to specify the variables:

- Copy the variable properties from Excel or word processing program, refer to [Method 1](#) on page 19.
- Specify all variable of the measurement types manually, refer to [Method 2](#) on page 20.


### Method 1

#### To specify measurement types via copy & paste

- 1** On the Processing ribbon, click Measurement Type – New.  
ModelDesk creates a new measurement type and opens the Measurement Type pane.
- 2** In Excel or a word processing program, select a table containing the variable properties and copy them to the Clipboard.
- 3** In ModelDesk on the Measurement Type pane, open the context menu and select Paste.

The Add Measurement Type Variable dialog opens and shows the properties of the first variable.

- 4 In the Add Measurement Type Variable dialog, modify the properties if necessary.

If  is displayed in the edit field of a property, the value is invalid. To get information on the reason, move the mouse pointer to the symbol.

- 5 Click OK to confirm the settings for this variable.

#### Tip

If you select the Show only on conflict option, the Add Measurement Type Variable dialog opens only when a property of a variable is invalid.

ModelDesk shows the properties of the next variable.

- 6 When all variables are specified, go to the Processing ribbon and click File – Save As.

The Specify File Name dialog opens for you to specify the name of the measurement type.

## Method 2


### To specify measurement types manually

- 1 On the Processing ribbon, click Measurement Type – New.  
ModelDesk creates a new measurement type and opens the Measurement Type pane.

- 2 On the Measurement Type pane, open the context menu and select Append or Insert.

The Add Measurement Type Variable dialog opens.

- 3 Specify the name and properties of the variable. For details of the properties, refer to [Add Measurement Type Variable](#) on page 60.

- 4 If  is displayed in the edit field of a property, the value is invalid. To get information on the reason, move the mouse pointer to the symbol.
- 5 Repeat the previous steps to add more variables to the measurement type.
- 6 When all variables are specified, go to the Processing ribbon and click File – Save As.  
The Specify File Name dialog opens for you to specify the name of the measurement type.

Result	You have specified a measurement type. The Properties pane shows the properties of the measurement type.
Related topics	<div>References<div>Append Measurement Type Variable / Append..... 46 Insert (Measurement Type)..... 50 Open from Pool (Measurement Type)..... 54</div></div>

## How to Get Measurement Data by Mapping Raw Data

Objective	To get measurement data, you can map the variables of raw data to measurement type variables. The units can be adjusted.
Overview	<p>To get measurement data, you must perform the following tasks:</p> <ol style="list-style-type: none"><li>1. Select a measurement type.</li><li>2. Create or open a measurement data file.</li><li>3. Select the Excel™ file which contains raw data.</li><li>4. Map the raw data variable to the measurement type. You can also import the mapping of an existing measurement data file.</li><li>5. Specify the conversion parameter to adapt the variable's units.</li><li>6. Save the measurement data file to specify its name.</li></ol> <p>You can also calculate the variables of a measurement data file. Refer to <a href="#">How to Calculate Measurement Data</a> on page 25.</p>
Adapting the units	<p>When you map the raw data variables to measurement type variables, you can specify conversion parameters to adapt their units. The parameters are a factor and an offset which are used in the following formula:</p> $\text{Measurement\_Type\_Variable} = \text{Factor} \cdot \text{Raw\_Data\_Variable} + \text{Offset}$

ModelDesk compares the strings of the units of a measurement type variable with those of a raw data variable to decide which conversion parameters to use. So you have to specify the conversion parameters only once for a pair of units but you must also specify conversion parameters if different writing styles are used for the units. The following table shows some examples.

Variable Unit	Raw Unit	Factor	Offset
m	mm	0,001	0
m	cm	0,01	0
m2	m <sup>2</sup>	1	0
[m]	m	1	0
[m s]	m/s	1	0
[km h]	[m s]	3,6	0

### Preconditions

- A measurement type must be specified, refer to [How to Specify Measurement Types](#) on page 18.
- An Excel file with raw data must be available, refer to [Raw Data, Measurement Types, and Measurement Data](#) on page 16.

### Method

#### To get measurement data by mapping raw data

- 1 You can create new measurement data or use an existing measurement data file:

To create new measurement data:

1. On the Processing ribbon, click **Measurement Data – New**.

ModelDesk opens the **Specify File Name** dialog and opens the folder in the Pool that contains the measurement types of the ModelDesk project.

2. In the **Specify File Name** dialog, select a measurement type and click **Open**.

ModelDesk opens a **Measurement Data** pane and loads the selected measurement type. The variables of the measurement type are listed in the table on the right-hand side.

To use an existing measurement data file:

1. On the Processing ribbon, click **Measurement Data – Open from Pool**.

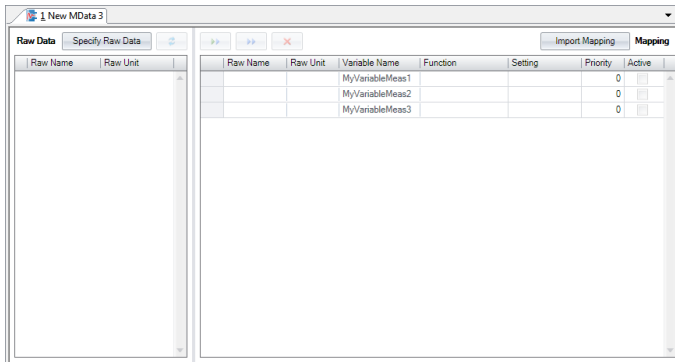
ModelDesk opens the **Specify File Name** dialog and opens the folder in the Pool that contains the measurement types of the ModelDesk project.

2. In the **Specify File Name** dialog, select the measurement type used for the measurement data and click **Open**.

The Specify File Name dialog shows all measurement data files that are based on the selected measurement type.

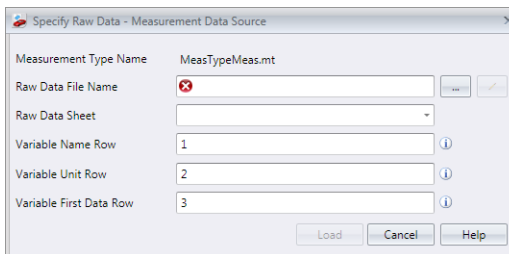
3. Select the measurement data file.

ModelDesk opens a Measurement Data pane and loads the selected measurement data file. The variables of the measurement type are listed in the table on the right-hand side. When a raw data file is assigned to the measurement data file, the raw data variables are listed on the left-hand side.



- 2 On the Measurement Data pane, click Specify Raw Data.

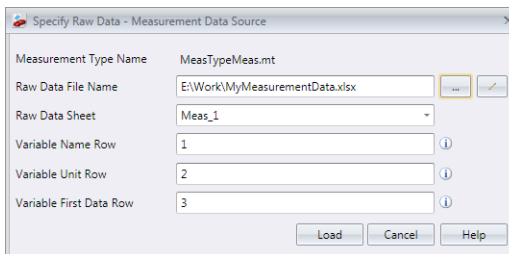
ModelDesk opens the Specify Raw Data dialog.



- 3 In the Specify Raw Data dialog, click the Browse button and select the Excel file which contains the raw data.

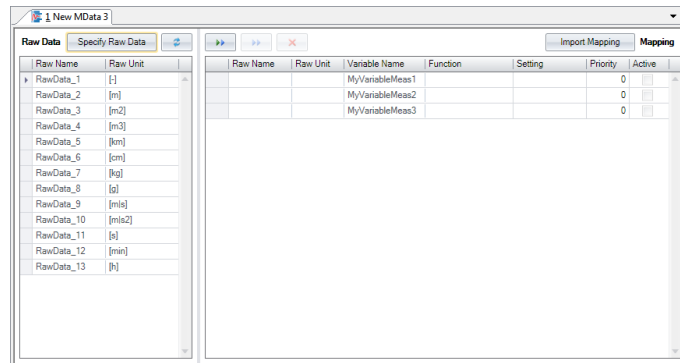
ModelDesk reads the selected Excel file and prepares the dialog with the read data.

- 4 Select the worksheet and specify the row numbers.






## 5 Click Load.

ModelDesk loads the Excel file and lists the variable found in the Excel file on the left-hand side of the Measurement Data pane.

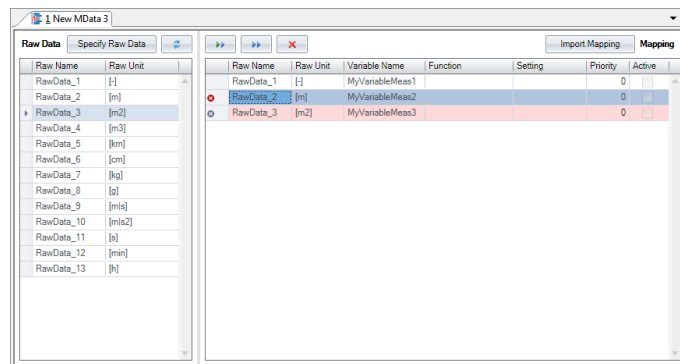


## 6 Map raw data variables to measurement type variables. You have several options:

- If raw name and raw unit are identical to variable name and variable unit, ModelDesk can automatically do the mapping: Open the context menu of the pane and choose **Auto Map All** or click .
- Select their entries one after another and click .
- Drag a raw data variable to a measurement variable.
- If the order of raw data variables and measurement type variables is equal, you can map several variables at once: Multi-select the variables in both tables and click .
- If a measurement data with the necessary mapping already exists, you can import its mapping: Click **Import Mapping**.

A dialog opens for you to select the measurement data file.

The name and unit of the raw data variable appear in the table on the right-hand side.



## 7 In some cases, it is necessary to specify the conversion parameters.

If the columns are hidden, you can make them visible:

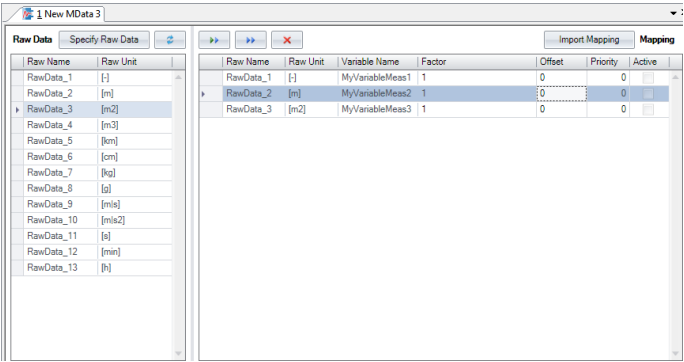
1. Open the context menu on the table header and select **Show Column Chooser**.



The column chooser contains all the hidden properties of the table.  
2. Click Factor and Offset.

ModelDesk adds both properties to the table.  
If the units of the mapped variables differ, specify the conversion parameter (Factor and Offset). If only their spelling differs, enter Factor = 1 and Offset = 0.

8 Repeat the previous steps for all the other variables that are necessary for the calculation.



9 If all variables are mapped, go to the Processing ribbon and click File – Save As or Save.  
If the measurement data file has no name, the Specify File Name dialog opens for you to specify the name.

**Result** Measurement data is created. The Properties pane shows the properties of the measurement data.

Related topics

Basics

[Customizing Tables \(ModelDesk Basics\)](#)

References

Measurement Data.....	62
Open from Pool (Measurement Data).....	53
Specify Raw Data – Measurement Data Source.....	76

# How to Calculate Measurement Data

**Objective** To get measurement data, you can calculate their values using MATLAB functions.

## Overview

To get measurement data, you must perform the following tasks:

1. Select a measurement type.
2. Create or open a measurement data file.
3. Write and assign functions to variables of the measurement data file.
4. Save the measurement data file to specify its name.

## Basics

You can use MATLAB functions to calculate the values for the variables of the measurement data file. ModelDesk provides templates you can use to create new functions. You can assign the functions and setting functions to the variables in the **Measurement Data** pane.

## Number of sampling points

You can use calculated values and measured values for variables of the same measurement data. However, all the variables must have the same number of values (raw data measuring points or sampling points) for processing.

If you mapped raw data to measurement data, the number of measuring points of the raw data defines the number of values that are required for processing. If you want to add calculated variables to the measurement data, their number of sampling points must match the number of measuring points.

If raw data is not mapped to measurement data, the number of sampling points of the functions defines the number of values that are used for processing. Each function must calculate the defined number of values. If a function returns a different number of sampling points, the calculated values are ignored and the default values are used for processing.

You can specify the number of values with the **Number** property of **Measurement Data**. Another option is to specify "0" for the **Number** property. Then its value is calculated with the first evaluation of a function and set once. If you want to modify the number of sampling points, you must delete the calculated values. If you set the **Dynamic Size** property, the number of sampling points is set after each processing execution. However, only those calculated values are used for processing whose number of sampling points matches the value in **Number**.

## Preconditions

- A measurement type must be specified. Refer to [How to Specify Measurement Types](#) on page 18.
- If you use existing measurement data, it must be active. Refer to [Activate \(Measurement Data\)](#) on page 45.

## Method

### To calculate measurement data

- 1 You can create new measurement data or use an existing measurement data file:

To create new measurement data:

1. On the **Processing** ribbon, click **Measurement Data – New**.

ModelDesk opens the **Specify File Name** dialog and opens the folder in the Pool that contains the measurement types of the ModelDesk project.

2. In the **Specify File Name** dialog, select a measurement type and click **Open**.

ModelDesk opens a **Measurement Data** pane and loads the selected measurement type. The variables of the measurement type are listed in the table on the right-hand side.

To use existing measurement data:

1. On the **Processing** ribbon, click **Measurement Data – Open from Pool**.

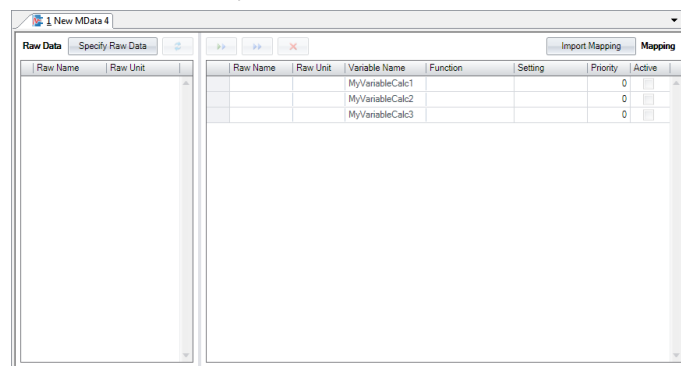
ModelDesk opens the **Specify File Name** dialog and opens the folder in the Pool that contains the measurement types of the ModelDesk project.

2. In the **Specify File Name** dialog, select the measurement type used for the measurement data and click **Open**.

The **Specify File Name** dialog shows all measurement data files that are based on the selected measurement type.

3. Select the measurement data file.

ModelDesk opens a **Measurement Data** pane and loads the selected measurement data. The variables of the measurement type are listed in the table on the right-hand side. When a raw data file is assigned to the measurement data, the raw data variables are listed on the left-hand side.



- 2 To assign function and setting files, the table must display the corresponding columns. If the columns are hidden, you can make them visible:

1. Open the context menu on the table header and select **Show Column Chooser**.

The column chooser contains all the hidden properties of the table.

2. Click **Function**, **Setting**, and **Priority**.

ModelDesk adds the properties to the table.

- 3 Click in the **Function** cell of the variable to be calculated.

When the cell is selected, a **Browse** button and an **Edit** button appear.

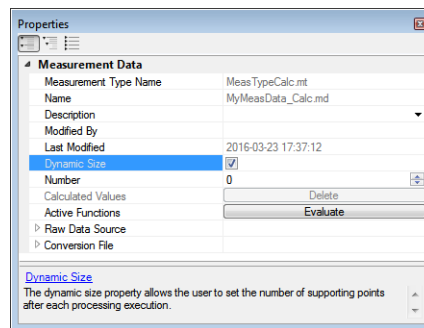
- 4 Click the **Edit** button.

The **Specify File Name** dialog opens.

- 5 Specify a file name and click **Save**.  
ModelDesk creates an M file with the specified name. If MATLAB is not running, it is started. Then the M file is opened in the MATLAB Editor. The M file contains the function template.
- 6 In the MATLAB Editor, implement the calculation in the function. To calculate values for ModelDesk, use the `asm_proc` function. This function implements the interface between ModelDesk and MATLAB. Refer to [asm\\_proc Function \(ASM User Guide\)](#).
- 7 Repeat the previous steps for the setting function.
- 8 In ModelDesk, in the **Measurement Data** pane, specify the **Priority** value to define the execution order.
- 9 After you implemented the function and setting files, you can start the evaluation in ModelDesk.

In ModelDesk, in the **Measurement Data** pane, click in the area above the table of the measurement variables or click **Show Measurement Data** in the properties of **Variable Mapping Details**.

The **Properties** panes shows the properties of the measurement data.




- 10 In the **Properties** pane, select the **Dynamic Size** property and specify **0** for **Number**.  
When **Dynamic Size** is selected, the value for **Number** is automatically set after each processing execution.
- 11 Click **Evaluate** to start the evaluation of all the activated functions of the measurement data.  
ModelDesk starts processing.
- 12 Observe the results and correct the function if necessary:
  - Because **Dynamic Size** is selected, the value of **Number** shows the number of sampling points of the first calculated function.
  - The **Data Source** property of each variable mapping shows which values are used. If the value is "Default", no calculated values are used.
  - Check the log window. If a function returns a different number of sampling points than specified in **Number**, an error message is displayed.
- 13 Repeat the previous steps for all the other variables to be calculated.  
Note that when **Dynamic Size** is selected and the value for **Number** has changed, all the calculated variables that exist are deleted. Only those calculated values are used for processing whose number of sampling points matches the value in **Number**.

- 14** To save the measurement data file, go to the Processing ribbon and click Measurement Data – Save As or Save.  
If the measurement data file has no name, the Specify File Name dialog opens for you to specify the name.

**Result** Measurement data is created. The Properties pane shows the properties of the measurement data.

**Related topics**

Basics

Customizing Tables (ModelDesk Basics )

References

Measurement Data..... 62

Specify Raw Data – Measurement Data Source..... 76

# Executing Processing

**Introduction** When the measurement data is specified and the function and settings files are written, you can execute Processing.

## Where to go from here

## Information in this section

<a href="#">Managing Measurement, Setting, Function, and Additional Function Files.....</a>	<a href="#">30</a>
The parameter values are calculated in MATLAB by using M files, which are managed in ModelDesk.	
<a href="#">How to Structure the Pool for Setting and Function Files.....</a>	<a href="#">32</a>
Function and Setting files are stored in the Pool of a ModelDesk project folders. You can create subfolders in these folders to structure the storage of the files.	
<a href="#">Working with the Processing Configuration Pane.....</a>	<a href="#">33</a>
The Processing Configuration pane is used for configuring the settings for processing.	
<a href="#">How to Map Setting and Function Files to Parameters.....</a>	<a href="#">35</a>
You can map setting and function files to parameters of the model in the Processing Configuration pane.	
<a href="#">How to Specify the Execution Sequence of Function Files.....</a>	<a href="#">38</a>
You can specify the sequence in which the function files are executed during processing.	
<a href="#">How to Specify the Plotting Details.....</a>	<a href="#">39</a>
MATLAB can generate plots to visualize the parameter values during processing. You can specify the details of the plots.	
<a href="#">How to Start Calculating the Parameter Values.....</a>	<a href="#">40</a>
When measurement data is specified and setting and function files are mapped to the parameters, the parameter values can be calculated.	

## Managing Measurement, Setting, Function, and Additional Function Files

**Introduction** The parameter values are calculated in MATLAB by using setting and function files, which are managed in ModelDesk.

**M files** The calculation of the parameter values is performed in MATLAB by using M files. There are 5 types of M files.

**Measurement function files** M files that calculate the variables of a measurement type. These M files can be used instead of mapping raw data.

**General Settings file** An M file that contains the general settings and that is used for all the function M files. For details on writing general settings file, refer to [General Settings File \(ASM User Guide !\[\]\(d84e7ea36f695d92cb39ec32c307ac93\_img.jpg\)](#)).

**Setting files** M files that have settings for the calculation and output of function files. You can assign one setting file to each parameter of the model in ModelDesk.

**Function files** M files that perform the calculation. You can assign one function file to each parameter of the model in ModelDesk. The function files calculate the parameter values in MATLAB and makes the results available to ModelDesk. ModelDesk reads the parameter values and writes them to the parameter pages. ASM provides several methods for this, refer to [asm\\_proc Function \(ASM User Guide !\[\]\(9dfdaff1d86ba3c1f8353b4d1b61b8c5\_img.jpg\)](#)).

**Additional function files** M Files that perform calculation independently from any parameter. It is not necessary to assign additional function files to parameters.

For information on how to write the function M files, refer to [ModelDesk Processing \(ASM User Guide !\[\]\(8d0f0e0fe25b320c33272c52aec1fbca\_img.jpg\)](#)).

---

## Managing M files

ModelDesk manages the M files (function, setting, and additional function files) within the ModelDesk project.

**Creating M files** You can create M files in ModelDesk. When you create an M file, ModelDesk opens a template for it and starts the MATLAB Editor.

**Editing setting and function files** You can start editing M files in ModelDesk. When you select an M file for editing, ModelDesk loads the file to the MATLAB Editor.

**Handling M files** ModelDesk stores the M files in the Pool of the ModelDesk project so that they can be used by all the experiments of the project. The Pool contains the Processing folder with the Function and Setting subfolders to store the files. You can create more subfolders under the Function and Setting subfolders to structure file storage, refer to [How to Structure the Pool for Setting and Function Files](#) on page 32.

You can export the M files to ModelDesk archives and import them from there. So it is possible to reuse M files in other ModelDesk projects.

## Related topics

### Basics

[asm\\_proc Function \(ASM User Guide !\[\]\(3d8c13c92b853674f749aac6fa869926\_img.jpg\)](#))  
[General Settings File \(ASM User Guide !\[\]\(ce455c990c00145a2dda1d9a310cb682\_img.jpg\)](#))  
[ModelDesk Processing \(ASM User Guide !\[\]\(de9e6664b8ceb5519927d73e240a55d9\_img.jpg\)](#))

### HowTos

[How to Structure the Pool for Setting and Function Files..... 32](#)

## How to Structure the Pool for Setting and Function Files

### Objective

Function and Setting files are stored in the Pool of a ModelDesk project in the **Pool\Processing\Function** and **Pool\Processing\Setting** folders. You can create subfolders in these folders to structure the storage of the files.

### Restrictions

You can only create, rename, or remove folders under the Function and Setting folder in the Pool.

### Method

#### To structure the Pool for setting and function files

- 1** To create a folder:  
 In the Project Navigator, open the context menu of the Function, Setting, or a subfolder of them and select **Create Folder**.  
 ModelDesk opens the **Create Folder** dialog.  
 In the **Create Folder** dialog specify a folder name and click **OK**.
- 2** To rename a folder:  
 Open the context menu of the folder and select **Rename**.  
 ModelDesk opens the **Rename Folder** dialog.  
 In the **Rename Folder** dialog specify a new folder name and click **OK**.
- 3** To delete a folder and all contained subfolders and files:  
 Open the context menu of the folder and select **Remove Folder**.  
 ModelDesk displays a warning message.  
 To confirm the warning, click **Yes**.

### Result

You have structured the folders for setting and function files.



Related topics

References

- [Create Folder \(ModelDesk Basics !\[\]\(38441ceaa711016e0bf2ad46ad394ff4\_img.jpg\)](#))
- [Remove Folder \(ModelDesk Basics !\[\]\(6e027340d4263908f264926b1ad81c5e\_img.jpg\)](#))
- [Rename Folder \(ModelDesk Basics !\[\]\(781510d64f329bf3c880acf086e884d6\_img.jpg\)](#))

# Working with the Processing Configuration Pane

Introduction

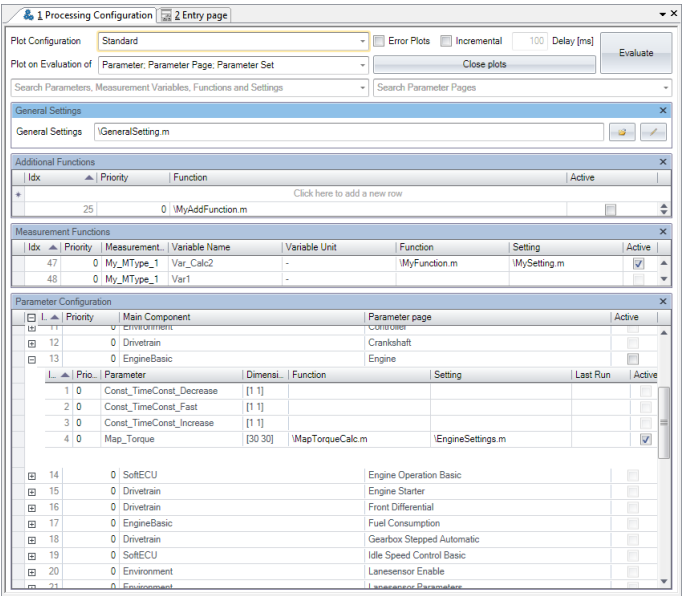
The Processing Configuration pane is used for configuring the settings for processing.

Basics on the Processing Configuration pane

- You can configure all the settings for the processing operation in the Processing Configuration pane:
- Select an M file with the general settings.
  - Select the M files with functions for parameter-independent calculations.
  - Select the M files with the functions and settings for the parameters.
  - Specify the execution sequence of the functions.
  - Specify the data that is displayed in plots.
  - Start the calculation of one parameter, all activated parameters of a parameter page, or all activated parameter pages.

Contents of the Processing Configuration pane

The following illustration shows the Processing Configuration pane:



The Processing Configuration pane has several areas and tables:

- *Plot Configuration*: This area contains the user interface for controlling the plotting. You can specify which plots are generated during processing execution.
- *Search field*: The search field lets you filter all the tables below. When you enter a string, only the parameters, measurement variables, function files, and setting files whose names contain this string are displayed.
- *Search Parameter Pages field*: The search field lets you filter all the tables below. When you enter a string, only the parameter pages whose names contain this string are displayed.
- *General Settings*: This field lets you select or edit the general settings file.
- *Additional Functions*: This table contains all the additional functions. You can add new functions, activate or deactivate listed functions, or specify their priority.
- *Measurement Functions*: This table contains all the measurement variables of the active measurement data file that are assigned to functions.
- *Parameter Configuration*: This table contains all the parameter pages and parameters. You can assign functions and setting files to parameters in this table.

## Searching the tables

You can search the tables for additional functions, measurement functions, and parameter configurations. When you specify a string in the search field, only the entries that match the search string are displayed.

To search for parameter pages, you can specify a string in the Search Parameter Pages field.

To clear the filters, click the  button in the search field.

## Working with tables

The tables for additional functions, measurement functions, and parameter configurations have some features that you can use to facilitate your work.

**Minimize tables** You can minimize tables that you do not need. To minimize a table, click the close button. Then the table is minimized and the corresponding tab is displayed at the left side of the pane.

**Maximize tables** You can maximize tables when they are minimized. To maximize a table, click the corresponding tab at the left side of the pane.

**Customize tables** You can sort or group the rows of a table, and hide columns. Refer to [Customizing Tables \(ModelDesk Basics !\[\]\(6a9b39b98eb945faa14c645ec99e4eaa\_img.jpg\)](#)).

**Filter** In addition filtering via the search field, you can filter each table separately. Refer to [How to Specify and Use a Filter \(ModelDesk Basics !\[\]\(9c2e8d1b5bd77cb5c9f83b7a9cff79fd\_img.jpg\)](#)).


## Related topics

### HowTos

[How to Map Setting and Function Files to Parameters.....](#) 35

How to Specify the Execution Sequence of Function Files.....	38
How to Specify the Plotting Details.....	39
How to Start Calculating the Parameter Values.....	40
References	
Processing Configuration.....	69

## How to Map Setting and Function Files to Parameters

Objective	You can map setting and function files to parameters of the model in the Processing Configuration pane.
Basics on setting and function files	<p>For basic information on managing setting and function files, refer to <a href="#">Managing Measurement, Setting, Function, and Additional Function Files</a> on page 30.</p> <p>For information on programming setting and function files, refer to <a href="#">ModelDesk Processing (ASM User Guide </a>).</p>
Exporting and importing parameter function mappings	After you finished mapping of setting and function files to parameters, you can export the settings to a MAT file. This MAT file can be imported to other projects. Refer to <a href="#">Export Mapping</a> on page 48 and <a href="#">Import Mapping</a> on page 49.
Possible methods	<p>ModelDesk has a Processing Configuration pane that lists all the parameter pages and parameters of the parameter set in a table. You can use this pane for mapping the setting and function files to the parameters. Furthermore, you can use the Properties pane for mapping when a parameter page or parameter is selected.</p> <ul style="list-style-type: none"><li>▪ To map setting and function files to parameters on the Processing Configuration pane, refer to <a href="#">Method 1</a> on page 36.</li><li>▪ To map setting and function files to parameters on a parameter page, refer to <a href="#">Method 2</a> on page 37.</li></ul>

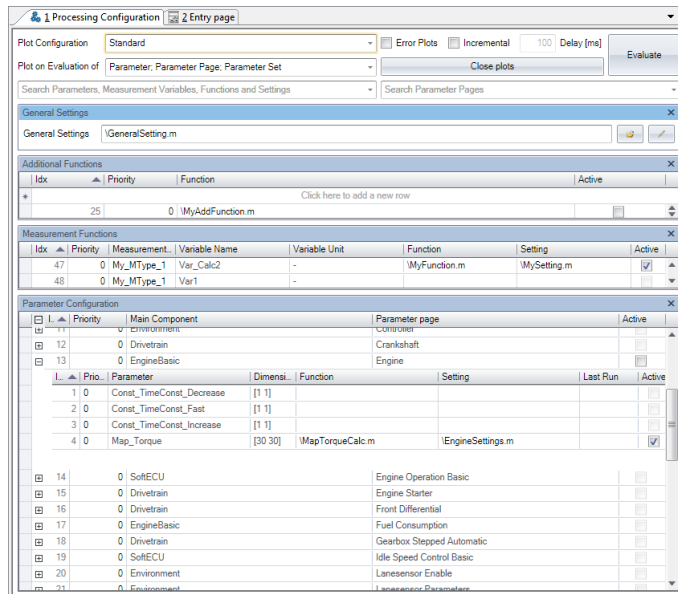
## Method 1

### To map setting and function files to parameters


- 1 On the Processing ribbon, click Configuration – Open.  
The Processing Configuration pane opens.

#### Tip

You can also open the pane using the Project Navigator:  
In the Project Navigator, open the context menu of the active parameter set and select Processing Configuration.



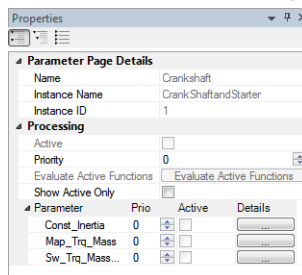
- 2 In the Search field, enter a part of the parameter name to which you want to map the setting and function files.  
The parameters are filtered according to the entered search string.
- 3 To select an existing function or setting file for a parameter,
  1. Click in the Function or Setting cell.  
ModelDesk opens a Specify File Name dialog.
  2. Select the file and click Open.
- 4 To create a new function or setting file,
  1. Click in the empty Function or Setting cell.  
ModelDesk opens a Specify File Name dialog.
  2. In the Specify File Name dialog, select or create a folder for the file.  
Function files must be stored under the **Pool/Processing/Function** folder or subfolder. Setting files must be stored under the **Pool/Processing/Setting** folder or subfolder.
  3. Specify a file name and click Save.  
ModelDesk creates a file and opens the file in the MATLAB Editor. The file contains a template of a setting or function for the selected parameter type.


4. In the MATLAB Editor, implement the function or setting in the file. For details on the implementation, refer to [ModelDesk Processing \(ASM User Guide\)](#).
- 5 To edit an existing function or setting file, click  in the Function or Setting cell.  
ModelDesk opens the MATLAB Editor and loads the file.
- 6 Repeat the previous steps for another parameter.

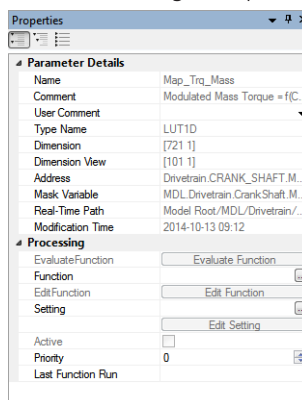
## Method 2


### To map setting and function files to parameters via Properties pane

- 1 In the Project Navigator, navigate to the parameter page.  
When the parameter page is selected, the Properties pane displays its properties, see the following example.



- 2 To select the parameter, click  on its entry in the Properties pane.  
When the parameter is selected, the Properties pane displays its properties, see the following example.



- 3 Click  in the Function or Setting property.  
ModelDesk opens a Specify File Name dialog.
- 4 In the Specify File Name dialog, select the setting or function file.
- 5 To edit an existing function or setting file, click Edit Function or Edit Setting.  
ModelDesk opens the MATLAB Editor and loads the file.
- 6 Repeat the previous steps for another parameter.

<b>Result</b>	Function and setting files are mapped to the parameters.
<b>Next steps</b>	You can specify the sequence in which the files are executed, refer to <a href="#">How to Specify the Execution Sequence of Function Files</a> on page 38.
<b>Related topics</b>	<div>References</div> <div>Processing Configuration..... 69</div>

## How to Specify the Execution Sequence of Function Files

<b>Objective</b>	To specify the sequence in which the function files are executed during processing.
<b>Execution sequence</b>	<p>MATLAB executes the function files one after the other during processing. When function files use parameter values that are calculated by other function files, the execution sequence must be observed so the results are consistent. It is therefore necessary to specify the execution sequence.</p> <p>You can specify the execution sequence on the <b>Processing Configuration</b> pane. The parameter pages, parameters, and additional functions have <b>Priority</b> properties that you can use to specify the execution sequence of the mapped function files. Functions with lower priority values are calculated before functions with higher priority values. For example, priority -1 is calculated before priority 0, priority 5 is calculated before priority 10. When the priority values are equal, the names of the parameter or additional functions are used to build the execution sequence.</p> <p>The Idx properties of parameter pages, parameters, and additional functions displays the position in the actual execution sequence.</p>

<b>Method</b>	<p><b>To specify the execution sequence of the function files</b></p> <ol style="list-style-type: none"> <li>On the Processing ribbon, click Configuration – Open. ModelDesk opens the Processing Configuration pane. The pane lists all the parameter pages and parameters of the model.</li> <li>To specify the execution sequence of parameter pages and additional functions, perform the following steps. <ol style="list-style-type: none"> <li>In the Priority edit field of a parameter page or additional function, enter a value. If you enter a lower value, the mapped function files are executed</li> </ol> </li> </ol>
---------------	---

earlier during processing. If you enter a higher value, the mapped function files are executed later.

2. Repeat the previous step for other parameter pages or additional functions until their execution sequence is correct.
- 3 To specify the execution sequence of the parameters on a parameter page, perform the following steps.
  1. Click the plus sign in a row of a parameter page to open the table with its parameters.
  2. In the Priority edit field of a parameter, enter a value. If you enter a low value, the mapped function files are executed earlier during processing. If you enter a high value, the mapped function files are executed later.
  3. Repeat the previous step for other parameters until the execution sequence is correct.

**Result**

The execution sequence is specified.

If the execution sequence is incorrect so that a parameter value is used in a function file before it the parameter is calculated, a warning occurs during processing.

**Next steps**

You can start processing, refer to [How to Start Calculating the Parameter Values](#) on page 40.

**Related topics****References**

[Processing Configuration.....](#) 69

## How to Specify the Plotting Details

**Objective**

MATLAB can generate plots to visualize the parameter values during processing. You can specify the details of the plots.

**Method****To specify the plotting details**

- 1 On the Processing ribbon, click Configuration – Open to open the Processing Configuration pane.
- 2 In Plot Configuration, select an item to specify the kind of signals to be plotted.
- 3 In Plot on Evaluation of, select the items to specify when the signals are plotted.

- 4 To create error plots during processing, select **Error Plots**.
- 5 To create incremental plots, select **Incremental** and specify a delay value.

**Result** You have specified the plotting details.

## Related topics

## References

<a href="#">Close Plots.....</a>	<a href="#">46</a>
<a href="#">Processing Configuration.....</a>	<a href="#">69</a>

# How to Start Calculating the Parameter Values

**Objective** When measurement data is specified and setting and function files are mapped to the parameters, the parameter values can be calculated.

**Activating measurement data** A ModelDesk project can manage several measurement type documents and measurement data documents. You can select one measurement type for your experiment. You must activate one measurement data document of the selected measurement type for processing.

**Activating function files** A function file is used for processing only when it is activated. You can only activate a parameter if a function file is mapped.

**Preconditions**

- Measurement data must be available, refer to [How to Get Measurement Data by Mapping Raw Data](#) on page 21 and [How to Calculate Measurement Data](#) on page 25.
- Function and setting M files must be written and mapped to the parameters, refer to [How to Map Setting and Function Files to Parameters](#) on page 35.

## Method

### To start calculating the parameter values

- 1 In the Project Navigator, select Experiment – Parametersets – parameter set – Processing – Measurement – Measurement type to open the Measurement Type Container page.
- 2 On the Measurement Type Container page, select measurement data.
- 3 Click **Activate**.  
The selected measurement data is used for processing.



- 4 In the Processing ribbon, click Configuration – Open to open the Processing Configuration pane.
- 5 On the Processing Configuration pane, select the Active option for all the parameter pages and parameters that are to be calculated.

Tip

You can also activate or deactivate the calculation of a parameter, a parameter page, or all the parameters using commands of the context menu.

- 6 Start processing. You can start processing for all the activated parameters, the parameters of a parameter page, or a single parameter:
  - To start processing for all the activated parameters within all the activated parameter pages of a parameter set, click Evaluate.
  - To start processing for all the parameters of a parameter page, open the context menu of the parameter page and select Evaluate Functions.
  - To start processing for a single parameter, open the context menu of the parameter and select Evaluate Function.

**Result** MATLAB calculates the parameter values and writes them to the parameter pages of the parameter set.

Related topics

References

Processing Configuration..... 69



# Reference Information

Where to go from here

Information in this section

<a href="#">Processing Commands.....</a>	<a href="#">44</a>
To manage the processing configuration.	
<a href="#">Processing Dialogs, Panes, and Pages.....</a>	<a href="#">60</a>
To configure the settings for processing.	
<a href="#">Processing Properties.....</a>	<a href="#">77</a>
To specify or read properties of measurement data, parameter pages, and parameters.	

# Processing Commands

## Introduction

There are various commands for configuring the processing operation. These commands are accessible via the Processing ribbon and context menu of the Project Navigator and the panes for processing.

## Where to go from here

## Information in this section

<a href="#">Activate (Measurement Data).....</a>	<a href="#">45</a>
To activate the opened measurement data file.	
<a href="#">Append Measurement Type Variable / Append.....</a>	<a href="#">46</a>
To append a new variable in the variable list of a measurement type.	
<a href="#">Close Plots.....</a>	<a href="#">46</a>
To close all MATLAB plots that are created during processing.	
<a href="#">Configure.....</a>	<a href="#">47</a>
To select measurement types for the processing operation in the experiment.	
<a href="#">Create Folder.....</a>	<a href="#">48</a>
To create a folder under the selected folder.	
<a href="#">Export Mapping.....</a>	<a href="#">48</a>
To export the parameter function mapping (assignment of the M files to the model parameters) to a MATLAB file.	
<a href="#">Import Mapping.....</a>	<a href="#">49</a>
To import a parameter function mapping (assignment of the M files to the model parameters).	
<a href="#">Insert (Measurement Type).....</a>	<a href="#">50</a>
To insert a new measurement type after the selected measurement type variable.	
<a href="#">New (Measurement Data).....</a>	<a href="#">51</a>
To create new measurement data.	
<a href="#">New (Measurement Type).....</a>	<a href="#">51</a>
To create a new measurement type.	
<a href="#">New Conversion File (Processing).....</a>	<a href="#">52</a>
To create a new conversion file.	
<a href="#">Open.....</a>	<a href="#">53</a>
To open the Processing Configuration dialog.	
<a href="#">Open from Pool (Measurement Data).....</a>	<a href="#">53</a>
To open a measurement data file from the Pool.	
<a href="#">Open from Pool (Measurement Type).....</a>	<a href="#">54</a>
To open a measurement type file from the Pool.	

<a href="#">Rename Folder.....</a>	<a href="#">55</a>
To rename the selected folder.	
<a href="#">Remove (Processing).....</a>	<a href="#">56</a>
To remove the selected file from the ModelDesk project and delete it from the file system.	
<a href="#">Remove Folder.....</a>	<a href="#">56</a>
To remove the selected folder from the ModelDesk project and delete it from the file system.	
<a href="#">Save (Processing).....</a>	<a href="#">57</a>
To save the currently active document of the Processing component.	
<a href="#">Save All (Processing).....</a>	<a href="#">58</a>
To save all open documents of the Processing component.	
<a href="#">Save As (Processing).....</a>	<a href="#">58</a>
To save the currently opened document of the Processing component with a new name.	

## Activate (Measurement Data)

### Access

You can access this command via:

Ribbon	Processing – Measurement Data
Context menu of	None
Shortcut key	None
Icon	None

### Purpose

To activate the opened measurement data file.

### Result

Measurement data is activated.

### Description

A ModelDesk project can have more than one measurement data file of each measurement type. One of it must be selected in the active parameter set for each configured measurement type to execute processing. You can use this command to activate the measurement that is currently opened in the working view.

**Related topics****HowTos**

[How to Calculate Measurement Data.....](#) 25

## Append Measurement Type Variable / Append

**Access**

You can access this command via:

Ribbon	None
Context menu of	Measurement Type pane
Shortcut key	None
Icon	None

**Purpose**

To append a new variable in the variable list of a measurement type.

**Result**

A new variable is added.

**Description**

When you select the command, the Add Measurement Type Variable dialog opens for you to specify the new variable, see [Add Measurement Type Variable](#) on page 60.

**Related topics****References**


[Insert \(Measurement Type\).....](#) 50

## Close Plots

**Access**

You can access this command via:

Ribbon	Processing – Configuration
Context menu of	None
Shortcut key	None

Icon	
Other	Button on the Processing Configuration pane

**Purpose** To close all MATLAB plots that are created during processing.

**Result** All plots are closed.

## Related topics

### HowTos


[How to Specify the Plotting Details.....](#) 39

### References

[Processing Configuration.....](#) 69

# Configure

**Access** You can access this command via:

Ribbon	Processing – Measurement Type
Context menu of	Experiment node in the Project Navigator
Shortcut key	None
Icon	

**Purpose** To select measurement types for the processing operation in the experiment.

**Result** The command opens the **Configure Measurement Types for Experiment** dialog for you to select measurement types for the experiment.

## Related topics

### HowTos

[How to Specify Measurement Types.....](#) 18

### References

[Configure Measurement Types for Experiment.....](#) 61

## Create Folder

### Access

You can access this command via:

Ribbon	None
Context menu of	Project Navigator under the Pool – Processing – Function and Setting folders
Shortcut key	None
Icon	None

### Purpose

To create a folder under the selected folder.

### Result

A new folder is created.

### Description

You can use this command to structure the Function and Setting folders in the Pool of a ModelDesk project. You can only create folders under the Function and Setting folders.

### Create Folder dialog

Lets you specify the name of the folder.

### Related topics

#### HowTos

[How to Structure the Pool for Setting and Function Files.....](#) 32


#### References

[Remove Folder.....](#) 56  
[Rename Folder.....](#) 55

## Export Mapping

### Access




You can access this command via:

Ribbon	Processing – Parameter Functions
Context menu of	None
Shortcut key	None
Icon	



<b>Purpose</b>	To export the parameter function mapping (assignment of the M files to the model parameters) to a MATLAB file.
<b>Result</b>	The function mapping is saved to a MATLAB file. This file contains the assignment of the function and setting M files to the model parameters and sequence in which the M files are calculated.
<b>Description</b>	<p>A <b>Save As</b> dialog opens for you to specify the path and the file name. When you have specified a file name, the function mapping is saved.</p> <p>You can import the function mapping using the <b>Import Mapping</b> command.</p>
<b>Related topics</b>	<p><b>Basics</b></p> <p><a href="#">Managing Measurement, Setting, Function, and Additional Function Files.....</a> 30</p> <p><b>HowTos</b></p> <p><a href="#">How to Map Setting and Function Files to Parameters.....</a> 35</p> <p><b>References</b></p> <p><a href="#">Import Mapping.....</a> 49</p>

## Import Mapping

<b>Access</b>	<p>You can access this command via:</p> <table border="1"> <tr> <td>Ribbon</td><td>Processing – Parameter Functions</td></tr> <tr> <td>Context menu of</td><td>None</td></tr> <tr> <td>Shortcut key</td><td>None</td></tr> <tr> <td>Icon</td><td></td></tr> </table>	Ribbon	Processing – Parameter Functions	Context menu of	None	Shortcut key	None	Icon	
Ribbon	Processing – Parameter Functions								
Context menu of	None								
Shortcut key	None								
Icon									
<b>Purpose</b>	To import a parameter function mapping (assignment of the M files to the model parameters).								
<b>Result</b>	The function mapping is imported.								

**Description** An Open dialog opens for you to select a MATLAB file that was exported before and has a function mapping (see [Export Mapping](#) on page 48).

**Related topics****Basics**

[Managing Measurement, Setting, Function, and Additional Function Files.....](#) 30

**HowTos**

[How to Map Setting and Function Files to Parameters.....](#) 35

**References**

[Measurement Data Variable Mapping Import Conflicts.....](#) 65

## Insert (Measurement Type)

**Access**

You can access this command via:

Ribbon	None
Context menu of	Measurement Type pane
Shortcut key	None
Icon	None

**Purpose**

To insert a new measurement type after the selected measurement type variable.

**Result**

A new variable is inserted.

**Description**

When you select the command, the **Add Measurement Type Variable** dialog opens for you to specify the new variable, see [Add Measurement Type Variable](#) on page 60.


**Related topics****References**

[Append Measurement Type Variable / Append.....](#) 46

## New (Measurement Data)

### Access

You can access this command via:

Ribbon	Processing – Measurement Data
Context menu of	Project Navigator in the Processing folder
Shortcut key	None
Icon	

### Purpose

To create new measurement data.

### Result

New measurement data is created.

### Description

To create measurement data, raw data is mapped to variables of a measurement type. When you select the **New** command from the main menu, the **Specify File Name** dialog opens for you to select the measurement type file the new measurement data should belong to. When you select the **New** command from the context menu of a measurement type element in the Project Navigator, the selected measurement type is used. When the measurement type is selected, the **Measurement Data** pane opens for you to select the raw data file and map data to the variables.

### Related topics

#### HowTos

[How to Get Measurement Data by Mapping Raw Data.....](#) 21


#### References

[Measurement Data.....](#) 62

## New (Measurement Type)

### Access

You can access this command via:

Ribbon	Processing – Measurement Type
Context menu of	None
Shortcut key	None
Icon	

<b>Purpose</b>	To create a new measurement type.
<b>Result</b>	A measurement type is created.
<b>Description</b>	The <b>Measurement Type</b> pane opens. The pane contains a table of variables of the measurement type. When you create a new measurement type, the table is empty. You can add and specify the measurement type variables on the <b>Measurement Type</b> pane.
<b>Related topics</b>	<p>HowTos</p> <p><a href="#">How to Specify Measurement Types..... 18</a></p> <p>References</p> <p><a href="#">Measurement Type Pane..... 66</a></p>

## New Conversion File (Processing)

<b>Access</b>	<p>You can access this command via:</p> <table border="1"> <tr> <td>Ribbon</td><td>None</td></tr> <tr> <td>Context menu of</td><td>None</td></tr> <tr> <td>Shortcut key</td><td>None</td></tr> <tr> <td>Icon</td><td>None</td></tr> <tr> <td>Property</td><td>File Path property of the <b>Measurement Data</b> pane</td></tr> </table>	Ribbon	None	Context menu of	None	Shortcut key	None	Icon	None	Property	File Path property of the <b>Measurement Data</b> pane
Ribbon	None										
Context menu of	None										
Shortcut key	None										
Icon	None										
Property	File Path property of the <b>Measurement Data</b> pane										
<b>Purpose</b>	To create a new conversion file.										
<b>Result</b>	A new conversion file is created.										
<b>Description</b>	The conversion file is stored in <b>Pool – Processing – Measurement – Conversion</b> .										

---

**Create File dialog** Lets you specify the name of the conversion file.

---

## Related topics

### References


[Measurement Data Properties..... 78](#)

# Open

---

## Access

You can access this command via:

Ribbon	Processing – Measurement Type
Context menu of	Experiment node in the Project Navigator
Shortcut key	None
Icon	

## Purpose

To open the Processing Configuration dialog.

## Result

The command opens the Processing Configuration dialog for you to configure the processing.

## Related topics

### References


[Processing Configuration..... 69](#)

# Open from Pool (Measurement Data)

---




## Access

You can access this command via:

Ribbon	Processing – Measurement Data
Context menu of	Project Navigator in the Processing folder
Shortcut key	None
Icon	

<b>Purpose</b>	To open a measurement data file from the Pool.
<b>Result</b>	The selected measurement data file is opened.
<b>Description</b>	<p>When you select the <b>Open from Pool</b> command from the ribbon, ModelDesk opens the <b>Specify File Name</b> dialog with the path of the Pool. You can first select the measurement type and then a measurement data file to use in the current experiment.</p> <p>When you select the <b>Open from Pool</b> command from the context menu of the <b>Project Navigator</b>, ModelDesk opens the <b>Specify File Name</b> dialog with the path to the selected measurement type folder. You can select a measurement data file to use in the current experiment.</p>
<b>Related topics</b>	<p><b>Basics</b></p> <p><a href="#">Working with the Processing Configuration Pane..... 33</a></p> <p><b>HowTos</b></p> <p><a href="#">How to Calculate Measurement Data..... 25</a></p>

## Open from Pool (Measurement Type)

<b>Access</b>	<p>You can access this command via:</p> <table border="1"> <tr> <td>Ribbon</td><td>Processing – Measurement Type</td></tr> <tr> <td>Context menu of</td><td>None</td></tr> <tr> <td>Shortcut key</td><td>None</td></tr> <tr> <td>Icon</td><td></td></tr> </table>	Ribbon	Processing – Measurement Type	Context menu of	None	Shortcut key	None	Icon	
Ribbon	Processing – Measurement Type								
Context menu of	None								
Shortcut key	None								
Icon									
<b>Purpose</b>	To open a measurement type file from the Pool.								
<b>Result</b>	The selected measurement type file is opened.								
<b>Description</b>	ModelDesk opens the <b>Specify File Name</b> dialog with the path of the Pool. You can select a measurement type file to use in the current experiment.								

Related topics

HowTos

[How to Specify Measurement Types.....](#) 18

Rename Folder

Access

You can access this command via:

Ribbon	None
Context menu of	Project Navigator under the Pool – Processing – Function and Setting folders
Shortcut key	None
Icon	None

Purpose

To rename the selected folder.

Description

You can only rename folders under the Function and Setting folders.

Result

The selected folder is renamed.

Rename Folder dialog

Lets you specify a new name for the folder.

Related topics

HowTos

[How to Structure the Pool for Setting and Function Files.....](#) 32

References

[Create Folder.....](#) 48

[Remove Folder.....](#) 56

## Remove (Processing)

### Access

You can access this command via:

Ribbon	None
Context menu of	Project Navigator in the Processing folder
Shortcut key	None
Icon	None

### Purpose

To remove the selected file from the ModelDesk project and delete it from the file system.

### Description

You can remove the M files, measurement types, measurement data files, and conversion file using this command.

### Result

The selected file is deleted.

### Related topics

#### HowTos

<a href="#">How to Map Setting and Function Files to Parameters.....</a>	<a href="#">35</a>
<a href="#">How to Specify Measurement Types.....</a>	<a href="#">18</a>
<a href="#">How to Structure the Pool for Setting and Function Files.....</a>	<a href="#">32</a>

## Remove Folder

### Access

You can access this command via:

Ribbon	None
Context menu of	Project Navigator under the Pool – Processing – Function and Setting folders
Shortcut key	None
Icon	None




### Purpose

To remove the selected folder from the ModelDesk project and delete it from the file system.




<b>Description</b>	You can only remove folders under the Function and Setting folders.
<b>Description</b>	When a folder is removed, its subfolders and M files are also removed.
<b>Result</b>	The selected folder is deleted.
<b>Related topics</b>	<p><b>HowTos</b></p> <p><a href="#">How to Structure the Pool for Setting and Function Files.....</a> 32</p> <p><b>References</b></p> <p><a href="#">Create Folder.....</a> 48</p> <p><a href="#">Rename Folder.....</a> 55</p>

## Save (Processing)

<b>Access</b>	<p>You can access this command via:</p> <table border="1"> <tr> <td>Ribbon</td><td>Processing – File</td></tr> <tr> <td>Context menu of</td><td>None</td></tr> <tr> <td>Shortcut key</td><td>None</td></tr> <tr> <td>Icon</td><td></td></tr> </table>	Ribbon	Processing – File	Context menu of	None	Shortcut key	None	Icon	
Ribbon	Processing – File								
Context menu of	None								
Shortcut key	None								
Icon									
<b>Purpose</b>	To save the currently active document of the Processing component.								
<b>Result</b>	The currently active document is saved.								
<b>Related topics</b>	<p><b>HowTos</b></p> <p><a href="#">How to Get Measurement Data by Mapping Raw Data.....</a> 21</p> <p><a href="#">How to Map Setting and Function Files to Parameters.....</a> 35</p> <p><a href="#">How to Specify Measurement Types.....</a> 18</p>								

How to Specify the Plotting Details.....	39
References	
Save All (Processing).....	58

# Save All (Processing)

Access	You can access this command via:	
	Ribbon	Processing – File
	Context menu of	None
	Shortcut key	None
	Icon	

Purpose	To save all open documents of the Processing component.
---------	---

Result	All the opened documents of the Processing component are saved.
--------	---

Related topics	HowTos
	How to Get Measurement Data by Mapping Raw Data.....21
	How to Map Setting and Function Files to Parameters.....35
	How to Specify Measurement Types.....18
	How to Specify the Plotting Details.....39
	References
	Save (Processing).....57

# Save As (Processing)

Access	You can access this command via:	
	Ribbon	Processing – File
	Context menu of	None

Shortcut key	None
Icon	

**Purpose** To save the currently opened document of the Processing component with a new name.

**Result** The currently opened document is saved with the new name.

**Related topics**

HowTos

- How to Get Measurement Data by Mapping Raw Data..... 21
- How to Map Setting and Function Files to Parameters..... 35
- How to Specify Measurement Types..... 18
- How to Specify the Plotting Details..... 39

References

- Save (Processing)..... 57
- Save All (Processing)..... 58

## Processing Dialogs, Panes, and Pages

### Introduction

There are several dialogs, panes, and pages for configuring the processing operation.

### Where to go from here

### Information in this section

<a href="#">Add Measurement Type Variable.....</a>	<a href="#">60</a>
To specify a variable for a measurement type.	
<a href="#">Configure Measurement Types for Experiment.....</a>	<a href="#">61</a>
To select measurement types for the processing operation in the experiment.	
<a href="#">Measurement Data.....</a>	<a href="#">62</a>
To select the file containing raw data and map data to variables or to map functions for calculating the variable values to have measurement data for processing.	
<a href="#">Measurement Data Variable Mapping Import Conflicts.....</a>	<a href="#">65</a>
To select the mappings to import although conflicts occur.	
<a href="#">Measurement Type Pane.....</a>	<a href="#">66</a>
To configure a measurement type.	
<a href="#">Measurement Types Page.....</a>	<a href="#">68</a>
To select and open a measurement type.	
<a href="#">Measurement Type Container Page.....</a>	<a href="#">68</a>
To activate and open a measurement data file.	
<a href="#">Processing Page.....</a>	<a href="#">69</a>
To open the Measurement Types page or the Processing Configuration pane.	
<a href="#">Processing Configuration.....</a>	<a href="#">69</a>
To configure and start processing.	
<a href="#">Specify Raw Data – Measurement Data Source.....</a>	<a href="#">76</a>
To select raw data to be used as measurement data.	

## Add Measurement Type Variable

### Access

The dialog opens when you add a variable to a measurement type:

- Using the Append Measurement Type Variable / Append or Insert (Measurement Type) command

- By copying data from Excel and pasting it to a measurement type
- By copying data from a word processing program and pasting it to a measurement type (variable properties must be separated by tabulators (\t) and the variables must be separated by line breaks (\r\n))

---

**Purpose** To specify a variable for a measurement type.

---

**Dialog settings**

**Name** Lets you specify the variable name. The name must be unique in the measurement type and comply with the rules for variable names in MATLAB.

**Unit** Lets you specify the unit of the variable.

**Default Value** Lets you specify the default value of the variable.

**Lower Limit** Lets you specify the lower limit of the variable. You can enter a numerical value or "-Infinity".

**Upper Limit** Lets you specify the upper limit of the variable. You can enter a numerical value or "Infinity".

**Description** Lets you specify a description for the variable.

**Show only on conflict** Lets you specify to open the dialog only when an added variable causes conflicts.

---

## Related topics

### HowTos

[How to Specify Measurement Types.....](#) 18

### References


[Append Measurement Type Variable / Append.....](#) 46

[Insert \(Measurement Type\).....](#) 50

## Configure Measurement Types for Experiment

### Access

You can access this command via:

Ribbon	Processing – Measurement Type
Context menu of	None
Shortcut key	None
Icon	

The dialog opens when you select the Configure command.

<b>Purpose</b>	To select measurement types for the processing operation in the experiment.
----------------	---

<b>Description</b>	The dialog has a table that contains all the measurement types available in the Pool. Select the measurement types that you want to use for processing.
--------------------	---

<b>Related topics</b>	<div>References</div> <div> <a href="#">Configure.....47</a> </div>
-----------------------	---


## Measurement Data


<b>Access</b>	You can open the pane by creating a new measurement data file by using <b>New (Measurement Data)</b> or opening an existing measurement data file on the <b>Measurement Type Container Page</b> page.
---------------	---

<b>Purpose</b>	To select the file containing raw data and map data to variables, or to map functions for calculating the variable values to have measurement data for processing.
----------------	--

<b>Description</b>	<p>You can perform several tasks in the <b>Measurement Data</b> pane:</p> <ul style="list-style-type: none"> <li>▪ To use raw data as measurement data, you can select raw data, map raw data to variables, and convert their units.</li> <li>▪ To calculate values to use them as measurement data, you can use the function and setting files, and specify the execution order of the measurement functions.</li> </ul>
--------------------	---

**Raw data selection** Raw data must be available in an Excel file. When you specify raw data, you must select the Excel file and specify the used worksheet and the rows where the variable name, unit, and values are stored. When raw data is loaded, the **Raw Data** table lists all the names and units of the selected worksheet.

**Mapping** When raw data is loaded, you can map raw data to the variables of the measurement type. Select a row in the **Raw Data** table and a row in the **Mapping** table and click  (**Map Variable**). ModelDesk enters the raw name and raw unit into the **Mapping** table and compares the raw unit with the unit of the variable. If they differ, you must use a conversion rule.

To delete a mapping, select a row on the **Mapping** table and click  (Delete Mapping). ModelDesk removes the raw data from the row.

**Unit conversion** The units of raw data can differ from those of a variable. You can specify conversion parameters to adapt the values. ModelDesk compares the strings of raw data unit and variable unit to decide which conversion parameters to use. The conversion parameters are a **Factor** and **Offset** according the following formula:

$$\text{Variable\_Value} = (\text{Raw\_Data\_Value} \cdot \text{Factor}) + \text{Offset}$$

For a pair of raw data unit and variable unit, you can specify only one conversion rule (set of conversion parameters).

Note that ModelDesk compares the strings of the units, so conversion rules are also required when the units are written in different styles.

The conversion rules are stored in a conversion file in the Pool of the ModelDesk project. The Pool can have multiple conversion files. The conversion file that is used for processing is selected in the **File Path** property. Using export and import from the Pool, you can reuse them in other ModelDesk projects.

**Using functions and setting files** Instead of using raw data, you can calculate the values to use them for processing. The function files that calculate the values, and the setting files that can be used for settings can be specified in the **Measurement Data** pane. You can create function and setting files, and select and edit existing function and setting files.


For processing with the measurement data, all the functions must return the same number of values. If you use calculated values and raw data for processing, the number of calculated values must match the number of values of the raw data.


To determine the execution order, you can specify priorities. Functions with a low priority are executed before functions with a higher priority.


When a function is mapped, the first column of the Mapping table contains *fx*. It is written in green when the function is active.


Refer to [How to Calculate Measurement Data](#) on page 25.

## Dialog settings


**Specify Raw Data**  Opens the Specify Raw Data – Measurement Data Source dialog to select the Excel file and specify raw data, refer to [Specify Raw Data – Measurement Data Source](#) on page 76.

**Reload Raw Data**  Reloads the raw data of the Excel file. The button is disabled unless you have specified the raw data. The command is useful when raw data is changed after it was specified in ModelDesk.

**Auto Map All**  Maps automatically the data in the Raw Data table to variable in the Mapping table. To do this, ModelDesk compares the names and units of data and variables. If they are identical, they are mapped.

**Map Variable**  Maps the data selected in the Raw Data table to the variable selected in the Mapping table. After mapping, ModelDesk compares

the units of data and variable. If they differ, you must specify a factor and offset for the conversion, see [Unit conversion](#) on page 63.

**Delete Mapping**  Deletes the mapping from the variable selected in the Mapping table.

**Import Mapping** Imports the mapping of another measurement data file. If you have mapped the variables of raw data and a measurement type of a similar measurement data file before, you can reuse the mapping.

**Raw Data table** Lists the raw name and raw unit of the data in the Excel file.

**Mapping table** Lists the variables of the measurement type. The table has the following columns:

Column <sup>1)</sup>	Description
Raw Name <sup>2)</sup>	Name of raw data after mapping
Raw Unit <sup>2)</sup>	Unit of raw data after mapping
Variable Name	Name of the variable specified for the measurement type
Variable Unit	Unit of the variable specified for the measurement type
Factor <sup>2)</sup>	Factor for adapting the units of raw data and the variable, see <a href="#">Unit conversion</a> on page 63
Offset <sup>2)</sup>	Offset value for adapting the units of raw data and the variable, see <a href="#">Unit conversion</a> on page 63
Default Value	Default value of the variable
Function <sup>3)</sup>	Function file that calculates the variable values. When you click the cell, a Browse button and edit button appear. Click the Browse button to select an existing function. Click the edit button to create a new function file or edit an existing function file in the MATLAB Editor.
Setting <sup>3)</sup>	Setting file that is called before calculating the variable values. When you click the cell, a Browse button and edit button appear. Click the Browse button to select an existing function. Click the edit button to create a new setting file or edit an existing setting file in the MATLAB Editor.
Priority <sup>3)</sup>	Priority for specifying the execution order.
Active <sup>3)</sup>	Displays whether the measurement function is active. If the measurement function is active, the referenced function is executed during processing execution.

<sup>1)</sup> By default, not all the listed columns are displayed. For instructions on displaying hidden columns, refer to [Customizing Tables \(ModelDesk Basics\)](#).

<sup>2)</sup> The column is required when you map raw data to a variable.

<sup>3)</sup> The column is required when you calculate the values for a variable.



Commands	<p>The pane has a context menu with the following commands.</p> <p><b>Map Variable</b> Maps the data selected in the Raw Data table to the variable selected in the Mapping table. After mapping, ModelDesk compares the units of data and variable. If they differ, you must specify a factor and offset for the conversion, refer to <a href="#">Unit conversion</a> on page 63.</p> <p><b>Auto Map All</b> Maps automatically the data in the Raw Data table to variable in the Mapping table. To do this, ModelDesk compares the names and units of data and variables. If they are identical, they are mapped.</p> <p><b>Delete Mapping</b> Deletes the mapping from the variable selected in the Mapping table.</p> <p><b>Evaluate Function</b> Lets you evaluate the function in MATLAB.</p> <p><b>Edit Function</b> Lets you edit the function file in the MATLAB Editor.</p> <p><b>Create Function</b> Creates a function file based on a standard template and opens it in the MATLAB Editor.</p> <p><b>Edit Setting</b> Lets you edit the setting file in the MATLAB Editor.</p> <p><b>Create Settings</b> Creates a setting file based on a standard template and opens it in the MATLAB Editor.</p>
----------	--

Related topics	<p>HowTos</p> <table><tr><td><a href="#">How to Calculate Measurement Data.....</a></td><td><a href="#">25</a></td></tr><tr><td><a href="#">How to Get Measurement Data by Mapping Raw Data.....</a></td><td><a href="#">21</a></td></tr></table> <p>References</p> <table><tr><td><a href="#">Measurement Data Properties.....</a></td><td><a href="#">78</a></td></tr><tr><td><a href="#">Measurement Type Container Page.....</a></td><td><a href="#">68</a></td></tr><tr><td><a href="#">New (Measurement Data).....</a></td><td><a href="#">51</a></td></tr><tr><td><a href="#">Variable Mapping Details.....</a></td><td><a href="#">84</a></td></tr></table>	<a href="#">How to Calculate Measurement Data.....</a>	<a href="#">25</a>	<a href="#">How to Get Measurement Data by Mapping Raw Data.....</a>	<a href="#">21</a>	<a href="#">Measurement Data Properties.....</a>	<a href="#">78</a>	<a href="#">Measurement Type Container Page.....</a>	<a href="#">68</a>	<a href="#">New (Measurement Data).....</a>	<a href="#">51</a>	<a href="#">Variable Mapping Details.....</a>	<a href="#">84</a>
<a href="#">How to Calculate Measurement Data.....</a>	<a href="#">25</a>												
<a href="#">How to Get Measurement Data by Mapping Raw Data.....</a>	<a href="#">21</a>												
<a href="#">Measurement Data Properties.....</a>	<a href="#">78</a>												
<a href="#">Measurement Type Container Page.....</a>	<a href="#">68</a>												
<a href="#">New (Measurement Data).....</a>	<a href="#">51</a>												
<a href="#">Variable Mapping Details.....</a>	<a href="#">84</a>												

## Measurement Data Variable Mapping Import Conflicts

Access	The dialog opens when you click Import Mapping on the Processing Configuration pane.
Purpose	To select the mappings to import although conflicts occur.

**Description** When you import mappings from another measurement data file, conflicts can occur. This dialog lists the conflicts and lets you import mappings although conflicts exist.

Select the mapping to be imported in spite of the reported conflicts.

#### Dialog settings

**Variable Name** Displays the name of the variable as specified in the measurement type.

**Current Raw Data Name** Displays the name of raw data to which the measurement type variable is currently mapped.

**Current Function** Displays the name of the function file to which the measurement type variable is currently mapped.

**Current Setting** Displays the name of the setting file to which the measurement type variable is currently mapped.

**Import Raw Name** Displays the name of raw data to which the measurement type variable is mapped in the file which is used for import.

**Import Function** Displays the name of the function file to which the measurement type variable is mapped in the file that is used for import.

**Import Setting** Displays the name of the setting file to which the measurement type variable is mapped in the file that is used for import.

#### Related topics

##### HowTos

How to Calculate Measurement Data.....	25
How to Get Measurement Data by Mapping Raw Data.....	21

##### References

Processing Configuration.....	69
-------------------------------	----

## Measurement Type Pane

**Access** You can open the pane via the **Configure** command.

**Purpose** To configure a measurement type.

**Description** The Measurement Type pane has a table that lists all the variables which are specified for the measurement type. The table displays the properties of the

variables. The Name and Unit properties cannot be modified. The other properties can be modified in the table.

The pane has a context menu with the following commands.

Command	Purpose
Copy	To copy the properties of a variable to the Clipboard.
Cut	To delete a variable from the table and copy its properties to the Clipboard.
Paste	<p>To paste one or more variables with their properties from the Clipboard to the table. Before the variable is inserted in the table, the <b>Add Measurement Type Variable</b> dialog opens so you can modify the variable's properties. Normally, a modification of the name is necessary, because the name must be unique within a measurement type.</p> <p>To copy data in the Clipboard for pasting, you can copy a variable from this table or copy data from Excel or a word processing program.</p> <p>In Excel the variable properties must specified in columns. Several variables can be specified in rows.</p> <p>In a word processing program the variable properties must be separated by tabulators (\t) and the variables must be separated by line breaks.</p> <p>The following properties must be specified for a variable:</p> <ul style="list-style-type: none"> <li>▪ Variable name (MATLAB-compatible)</li> <li>▪ Unit</li> <li>▪ Default value</li> <li>▪ Lower limit</li> <li>▪ Upper limit</li> <li>▪ Description</li> </ul>
Delete	To delete a variable from the table.
Insert	<p>To insert a new variable into the table at the selected position. Before the variable is inserted, the <b>Add Measurement Type Variable</b> dialog opens so you can specify the variable's properties. The name and unit can only be specified in this dialog.</p>
Append	<p>To append a new variable to the table. Before the variable is appended, the <b>Add Measurement Type Variable</b> dialog opens so you can specify the variable's properties. The name and unit can only be specified in this dialog.</p> <p>The command is also available in the menu bar, see <a href="#">Append Measurement Type Variable / Append</a> on page 46.</p>

**Related topics****Basics**[Raw Data, Measurement Types, and Measurement Data.....](#) 16**HowTos**[How to Specify Measurement Types.....](#) 18**References**[Add Measurement Type Variable.....](#) 60[Configure.....](#) 47

## Measurement Types Page

**Access**

You can open the page by clicking the **Processing – Measurement** node in the **Project Navigator**.

**Purpose**

To select and open a measurement type.

**Description**

The **Measurement Types** page lists all the measurement types that are stored in the Pool of the ModelDesk project. Select one and click **Open** to open it in a **Measurement Type Pane** pane.

**Related topics****References**[Measurement Type Pane.....](#) 66

## Measurement Type Container Page

**Access**

You can open the page by clicking a measurement type under the **Processing** node in the **Project Navigator**.

**Purpose**

To activate and open a measurement data file.

Result	A measurement data file is activated or opened in the Measurement Data pane.
Related topics	<div>References<div>Measurement Data..... 62</div></div>

## Processing Page

Access	You can open the page by clicking the Processing node in the Project Navigator.
Purpose	To open the Measurement Types page or the Processing Configuration pane.
Description	<ul style="list-style-type: none"><li>▪ Click Measurement to open the Measurement Types page, refer to <a href="#">Measurement Types Page</a> on page 68.</li><li>▪ Click Processing Configuration to open the Processing Configuration pane, refer to <a href="#">Processing Configuration</a> on page 69.</li></ul>
Related topics	<div>Basics<div>Basics of Processing..... 12</div></div> <div>References<div>Measurement Types Page..... 68</div><div>Processing Configuration..... 69</div></div>

## Processing Configuration

Access	You can access this command via:	
	Ribbon	Processing
	Context menu of	Project Navigator – Parameter set

Shortcut key	None
Icon	

**Purpose** To configure and start processing.

**Result** The parameter values are calculated and written to the parameter files.

**Description** Refer to [Working with the Processing Configuration Pane](#) on page 33.

**Dialog settings**

**Plot Configuration** Lets you specify the details of plotting. When the parameters are calculated, MATLAB plots can be created. You can select the data to be plotted in Plotting details.

**Plot on Evaluation of** Lets you enable or disable the creation of MATLAB plots during the evaluation of the functions. You can enable the MATLAB plot creation for single parameters, the parameters of a parameter page, the whole parameter set, additional functions, and measurement functions.


**Error Plots** Lets you enable or disable the creation of error plots during the evaluation of the functions.

**Incremental** Lets you enable or disable the creation of incremental plots during the evaluation of the functions. When this option is enabled, you must specify the delay value in ms.

**Close plots** Closes all the MATLAB plots which are created during processing.



**Evaluate** Starts the calculation of all the functions of the active parameter pages and active parameters.

**Search field** Lets you filter for parameter names and file names of M files for processing functions and settings. When you enter a filter string, the table is instantly filtered.

Click  to delete the filter.

**General settings**



The general settings file is calculated before any of the activated parameter functions. You can use this file to specify general parameter values.

**General Settings** Lets you specify the M file that performs settings which are valid for all the functions of the processing operation. You can select the M file via  and start the MATLAB Editor for editing via .

**Additional Functions** Additional functions are MATLAB functions that are calculated independently of the parameters during processing. You can create several additional functions

and edit them with the MATLAB Editor. By using the **Priority** property, you can specify the calculation sequence of the additional functions.

**Additional functions table** The columns of the table contain the properties of the additional functions.

Property	Description
Idx	Displays the index of the additional functions. The index shows the calculation sequence of the additional functions. ModelDesk sets the indices automatically according to the specified priorities. You can modify the sequence by specifying the <b>Priority</b> properties. ModelDesk calculates the sequence depending on the specified priorities of the additional functions, measurement functions, and parameter pages.
Priority	Lets you specify a priority for the calculation sequence of the additional functions. Additional functions with lower priority values are calculated before additional functions with higher priority values.
Function	Lets you specify the name of the M file that is used as additional function. When you click the cell, two buttons appear. To browse for an existing M file, click  . To edit an M file in the MATLAB Editor, click  . If the M file was already specified, it is loaded to the MATLAB Editor. If no M file is specified, a dialog opens for you to specify the path and name. Then the MATLAB Editor opens with a template for the function file.
Active	Lets you enable or disable the calculation of the additional function.

**Commands of the Additional Functions table** The table has a context menu with the following commands:





Command	Purpose
Evaluate Function	Lets MATLAB calculate the selected additional function.
Activate All Functions	Lets you activate all the additional functions for processing.
Deactivate All Functions	Lets you deactivate the additional function for processing.
Delete Function	Lets you delete the selected additional function.
Edit Function	Lets you edit the selected additional function in the MATLAB Editor.
Create Function	Creates a function file based on a standard template and opens it in the MATLAB Editor.

## Measurement Functions

Measurement functions are MATLAB functions that calculate values for variables of a measurement type. You can use such functions instead of raw data. You can create measurement functions and edit them with the MATLAB Editor. By using

the Priority property, you can specify the calculation sequence of the measurement functions.

**Measurement Functions table** The columns of the table contain the properties of the measurement functions.

Property	Description
Idx	Displays the index of the measurement functions. The index shows the calculation sequence of the measurement functions. ModelDesk sets the indices automatically according to the specified priorities. You can modify the sequence by specifying the individual Priority properties.
Priority	ModelDesk calculates the sequence depending on the specified priorities of the additional functions, measurement functions, and parameter pages. Lets you specify a priority for the calculation sequence of the measurement functions. Measurement functions with lower priority values are calculated before measurement functions with higher priority values.
Measurement Type	Displays the measurement type to which the variable belongs.
Variable Name	Displays the name of the measurement variable.
Variable Unit	Displays the unit of the measurement variable.
Function	Lets you specify the name of the M file that is used as a measurement function. When you click the cell, two buttons appear. To browse for an existing M file, click  . To edit an M file in the MATLAB Editor, click  . If the M file was already specified, it is loaded to the MATLAB Editor. If no M file is specified, a dialog opens for you to specify its path and name. Then the MATLAB Editor opens with a template for the function file.
Setting	Lets you specify the name of the M file that is used for setting initial values. To browse for an existing M file, click  . To edit an M file in the MATLAB Editor, click  . If the M file was already specified, it is loaded to the MATLAB Editor. If no M file is specified, a dialog opens for you to specify its path and name. Then the MATLAB Editor opens with a template for the setting M file.
Active	Lets you enable or disable the calculation of the measurement function.



**Commands of the Measurement Functions table** The table has a context menu with the following commands:

Command	Purpose
Evaluate Function	Lets MATLAB calculate the selected function.
Activate All Functions	Lets you activate all the measurement functions for processing.
Deactivate All Functions	Lets you deactivate the measurement function for processing.
Show Measurement Data	Displays the measurement data in the Measurement Data pane.
Edit Function	Lets you edit the selected measurement function in the MATLAB Editor.
Create Function	Creates a measurement function file based on a standard template and opens it in the MATLAB Editor.
Edit Settings	Lets you edit the selected setting file in the MATLAB Editor.
Create Settings	Creates a setting M file based on a standard template and opens it in the MATLAB Editor.

## Parameter Configuration

The pane has a table containing all the parameter pages of the model that are suitable for processing.

**Parameter page table** The columns of the table contain the properties of the parameter pages.



Property	Description
Idx	Displays the index of the parameter page. The index shows the calculation sequence of the parameter pages. ModelDesk sets the indices automatically according to the specified priorities and parameter page names. You can modify the sequence by specifying the Priority properties. The calculation sequence depends on the specified priorities of the additional functions, measurement functions, and parameter pages first and their names afterwards.
Priority	Lets you specify a priority for the calculation sequence of the parameter pages. Parameter pages with lower priorities are calculated before parameter pages with higher priorities.
Main Component	Displays the main component to which the parameter page belongs.
Parameter Page	Displays the name of the parameter page.
Active	Lets you enable or disable the processing of all the variables that belong to the parameter page.



**Commands of the parameter page table** The table has a context menu with the following commands:

Command	Purpose
Activate All Parameter Pages	Lets you activate the processing of all the parameter pages.
Activate All Parameters	Lets you activate the processing of all the parameters on the selected parameter page.
Deactivate All Parameter Pages	Lets you deactivate the processing of all the parameter pages.
Deactivate All Parameters	Lets you deactivate the processing of all the parameters on the selected parameter page.
Evaluate Functions	Lets you start to calculate the functions of all the activated parameters of the selected parameter page explicitly, regardless of the value of the Active property of the parameter page.
Show Parameter Page	Lets you open the parameter page in the parameter set.

When you click the plus sign in a row of a parameter page, the parameters of the parameter page with the following properties are displayed.

**Parameter table** The columns of the table contain the parameter properties of the selected parameter page.

Property	Description
Idx	Displays the index of the parameter. The index shows the calculation sequence of the parameters of a parameter page. ModelDesk sets the indices automatically according to the specified priorities and the parameter names. You can modify the sequence by specifying the Priority properties.
Priority	Lets you specify a priority for processing the parameter. Parameters with lower priority values are calculated before parameters with higher priority values.
Parameter	Displays the parameter name.
Dimension	Displays the dimension of the parameter.
Function	Lets you specify the name of the M file that is used to calculate the parameter value. When you click the cell, two buttons appear. To browse for an existing M file, click  . To edit an M file in the MATLAB Editor, click  . If the M file was already specified, it is loaded to the MATLAB Editor. If no M file is specified, a dialog opens for you to specify the path and name. Then the MATLAB Editor opens with a template for the function M file.
Setting	Lets you specify the name of the M file that is used for setting processing parameters.

Property	Description
	<p>To browse for an existing M file, click .</p> <p>To edit an M file in the MATLAB Editor, click . If the M file was already specified, it is loaded to the MATLAB Editor. If no M file is specified, a dialog opens for you to specify the path and name. Then the MATLAB Editor opens with a template for the setting M file.</p>
Last Run	Displays the time when the parameter function was last executed.
Active	<p>Lets you enable or disable the processing of the parameter function. A parameter is calculated only if it and its parameter page are activated.</p> <p>You can only activate a parameter for processing when a function is specified for it.</p>

**Commands of the parameter table** The table has a context menu with the following commands:

Command	Purpose
Activate All Page Parameters	Lets you activate all the parameters for processing.
Create Function	Creates a function M file based on a standard template and opens it in the MATLAB Editor.
Create Settings	Creates a setting M file based on a standard template and opens it in the MATLAB Editor.
Deactivate All Page Parameters	Lets you deactivate all the parameters for processing.
Edit Function	Lets you edit the function M file in the MATLAB Editor.
Edit Settings	Lets you edit the setting M file in the MATLAB Editor.
Evaluate Function	Lets you start the calculation of the parameter explicitly, regardless of the value of the Active property.
Show Parameter	Lets you open the page in the parameter set the parameter belongs to.

## Related topics

### HowTos

How to Specify the Execution Sequence of Function Files.....	38
How to Specify the Plotting Details.....	39
How to Start Calculating the Parameter Values.....	40

## Specify Raw Data – Measurement Data Source

<b>Access</b>	The dialog opens when you click <b>Specify Raw Data</b> on a <b>Measurement Data</b> pane.
<b>Purpose</b>	To select raw data to be used as measurement data.
<b>Description</b>	Raw data that is to be used for processing must be available in an Excel™ file in XLSX format. In this dialog, you can select the Excel file and specify the worksheet and cells where raw data is stored.
<b>Dialog settings</b>	<p><b>Measurement Type Name</b> Displays the name of the measurement type to which raw data is mapped.</p> <p><b>Raw Data File Name</b> Lets you select the Excel file in XLSX format that contains raw data.</p> <p><b>Raw Data Sheet</b> Lets you select the worksheet containing raw data to be used. The Excel file must be selected before you can select the worksheet.</p> <p><b>Variable Name Row</b> Lets you specify the number of the row containing the variable name.</p> <p><b>Variable Unit Row</b> Lets you specify the number of the row containing the unit.</p> <p><b>Variable First Data Row</b> Lets you specify the row number of the cell which contains the first data point.</p>

### Related topics

#### Basics

[Raw Data, Measurement Types, and Measurement Data..... 16](#)

#### HowTos

[How to Get Measurement Data by Mapping Raw Data..... 21](#)

#### References

[Measurement Data..... 62](#)

# Processing Properties

Introduction	The Properties pane shows properties for measurement data, parameter pages, and parameters.
--------------	---

Where to go from here	Information in this section
-----------------------	-----------------------------

<a href="#">Properties of Additional Functions.....</a>	<a href="#">77</a>
To get the properties of additional functions.	
<a href="#">Measurement Data Properties.....</a>	<a href="#">78</a>
To read the properties of measurement data.	
<a href="#">Measurement Type Properties.....</a>	<a href="#">80</a>
To read the properties of a measurement type.	
<a href="#">Parameter Properties.....</a>	<a href="#">80</a>
To read the properties of parameters.	
<a href="#">Parameter Page Properties.....</a>	<a href="#">82</a>
To read the properties of the parameters of parameter pages.	
<a href="#">Variable Mapping Details.....</a>	<a href="#">84</a>
To read the properties of variable mapping.	

## Properties of Additional Functions

Purpose	To get the properties of additional functions.
---------	--

Description	When you select an additional function on the Processing Configuration pane, the Properties pane displays the properties of the additional function. The properties give you information on the additional function and let you specify settings for processing.
-------------	--

Additional function	<b>Create Function</b> Creates a function and opens it in MATLAB.
	<b>Evaluate Functions</b> Lets MATLAB calculate the selected additional function.
	<b>Function</b> Displays the path and name of the M file or lets you select another M file as an additional function. The path is relative to the <b>Function</b> folder of the Pool.

**Edit Functions** Lets you edit the M file of the additional function in the MATLAB Editor.

**Active** Lets you enable or disable the additional function for processing.

**Priority** Lets you specify a priority for modifying the calculation sequence of the additional functions during processing.

**Last Function Run** Displays the date and time when the selected additional function was last executed.

**User Comment** Lets you enter a comment for the selected additional function.

## Related topics

### Basics

Managing Measurement, Setting, Function, and Additional Function Files.....	30
Working with the Processing Configuration Pane.....	33

## Measurement Data Properties

**Purpose** To read the properties of measurement data.

**Description** When measurement data is selected, the **Properties** pane displays its properties, the properties of the associated Excel file containing raw data, and the properties of the conversion file.

<b>Measurement Data properties</b>	<b>Active Functions</b>	Lets you evaluate all active functions.
	<b>Calculated Values</b>	Lets you delete existing calculated values for all the variables.
	<b>Description</b>	Lets you specify a description for the measurement data.
	<b>Dynamic Size</b>	Lets you define the number of supporting points to be set once by each processing execution.
	<b>Last Modified</b>	Displays the date and time when the settings were last modified.
	<b>Measurement Type Name</b>	Displays the name of the measurement type file which is used as the basis for the measurement data.

**Modified By** Displays the name of the user who modified the settings of the measurement data.

**Name** Displays the name of the measurement data file.

**Number** Displays the number of raw data measurement values if raw data is loaded. Otherwise, it displays the number of supporting points for all variables. If it is set to 0, it can be set dynamically once during the next processing execution.

#### Raw Data Source properties

**Data Row** Displays the number of the Excel sheet row that contains the first data entry.

**File** Displays the path and name of the Excel file that is used as raw data source.

**Measurements** Displays the number of data entries on the Excel sheet.

**Name Row** Displays the number of the row that contains the variable name.

**Sheet** Displays the name of the sheet in the Excel file that contains the raw data.

**Unit Row** Displays the number of the row that contains the unit of the variable.

#### Conversion File properties

**File Path** Lets you select a conversion file or create a new conversion file. Refer to [New Conversion File \(Processing\)](#) on page 52.

**Description** Lets you specify a description for the conversion file.

**Is Modified** Displays the modification state.

**Last Modified** Displays the date and time when the conversion file was last modified.

**Modified By** Displays the name of the user who modified the conversion file settings.

**Name** Displays the name of the conversion file.

#### Related topics

##### HowTos

<a href="#">How to Calculate Measurement Data.....</a>	<a href="#">25</a>
<a href="#">How to Get Measurement Data by Mapping Raw Data.....</a>	<a href="#">21</a>

##### References

<a href="#">Measurement Data.....</a>	<a href="#">62</a>
---------------------------------------	--------------------

## Measurement Type Properties

<b>Purpose</b>	To read the properties of measurement type.		
<b>Description</b>	When a measurement type is selected, the <b>Properties</b> pane displays its properties.		
<b>Measurement Type properties</b>	<b>Description</b>	Lets you specify a description for the measurement type.	
	<b>Last Modified</b>	Displays the date and time when the settings were last modified.	
	<b>Modified By</b>	Displays the name of the user who modified the settings of measurement type.	
	<b>Name</b>	Displays the name of the measurement type file.	

### Related topics

#### HowTos

[How to Specify Measurement Types.....](#) 18

## Parameter Properties

<b>Purpose</b>	To read the properties of parameters.		
<b>Description</b>	When you click a display name of a parameter on a parameter page , the <b>Properties</b> pane displays the properties of the parameter page. The properties give you information on the parameter and let you specify settings for processing.		



---

**Parameter Details**

**Address** Displays the parameter address including the instance ID. You can use the parameter address to access any parameter in processing functions or in the ModelDesk automation.

**Comment** Displays a description of the parameter.

**Dimension** Displays the maximum dimension specified for the parameter.

**Dimension View** Displays the current dimension of the parameter.

**Extrapolation Type** Displays the extrapolation type.

**Mask variable** Displays the name and path of the mask variable associated with the parameter. The property is empty if the parameter is a supporting parameter.

**Modification Time** Displays the time when the parameter values was last modified.

**Name** Displays the parameter name.

**Real-Time Path** Displays the name and path of the parameter in the variable description file. The property is empty if the parameter is a supporting parameter.

**Show Page Overview** Shows the parameter page that contains the parameter. Refer to [Parameter Page Properties](#) on page 82.

**Type name** Displays the type of the parameter.

**User Comment** Lets you specify a comment about the parameter.

---

**Processing properties**

**Active** Lets you enable the parameter function. If a parameter function is disabled, it is not calculated during processing.

**Create Function** Lets you create an M script to use it as a processing function. When you click the button, a standard dialog opens for you to specify the path and name for the M script. When the name is specified, ModelDesk creates an M script that contains a copy of the processing function template and opens it in the MATLAB Editor.

**Create Setting** Lets you create an M script which is used as setting function. When you click the button, a standard dialog opens for you to specify the path and name for the M script. When the name is specified, ModelDesk creates an

M script containing a copy of the setting template and opens it in the MATLAB Editor.

**Edit Function** Starts the MATLAB Editor and loads the function script for editing.

**Edit Setting** Starts the MATLAB Editor and loads the setting M script for editing.

**Evaluate Function** Lets you execute the processing function of the parameter.

**Function** Lets you specify the M script which is used as processing function. You can enter the name and path of the M script or click the Browse button and select it.

**Last function run** Displays the date and time of the last function run that modified the value of the parameter.

**Priority** Lets you specify the priority of the function. The priorities of all function M scripts determine the execution order of the function calculation. Refer to [How to Specify the Execution Sequence of Function Files](#) on page 38.

**Setting** Lets you specify the M script which is used for the settings of the processing operation. You can enter the name and path of the M script or click the Browse button and select it.

## Related topics

### Basics

[Working with the Processing Configuration Pane..... 33](#)

## Parameter Page Properties

### Purpose

To read the properties of the parameters of parameter pages.

### Description

When you select a parameter page in the **Project Navigator**, the **Properties** pane displays the properties of the parameter page. The properties give you information on the parameter page and let you specify settings for the processing.

**Parameter Page Details****File Name** Displays the name of the XML parameter file.**Instance ID** Displays the instance ID of the parameter page. For details of instance ID and instance name, refer to [Parameterizing ASM Blocks of the Same Type \(ModelDesk Parameterizing !\[\]\(4729e517bc6a7cd81c8025b9646574fb\_img.jpg\)](#)).**Instance Name** Displays the instance name of the parameter page.**Name** Displays the name of the parameter page.**Processing properties****Active** Lets you activate the processing for all the parameters of the parameter page.**Evaluate Active Functions** Starts the execution of all the active parameter's processing functions.**Parameter** Lists all the parameters of the parameter page with some properties.

Name	Description
Parameter	Displays the parameter name. The parameter name is the name of the parameter that is used in the real-time model and usually differs from the name that is displayed on the parameter page.
Priority	Lets you specify the priority of the parameter's processing function. The priority specifies the order of the calculation of the parameter functions. A low priority number is processed before a higher priority number. For example, priority 1 is executed before priority 2.
Active	Lets you specify the active state of the parameter's processing function. A processing function must be active to be calculated during processing.
Details	Displays the properties of the parameter in the Properties pane. For a description of the parameter's properties, refer to <a href="#">Parameter Properties</a> on page 80.

**Priority** Lets you specify the priority of the parameter page.**Show Active Only** Shows only the parameters whose processing functions are active.**Related topics****Basics**[Working with the Processing Configuration Pane.....](#) 33

## Variable Mapping Details

<b>Purpose</b>	To read the properties of variable mapping.
<b>Description</b>	When a variable on the <b>Measurement Data</b> pane is selected, the <b>Properties</b> pane displays the properties of the variable mapping.
<b>Variable Mapping Details</b>	<p><b>Active</b> Displays whether the measurement function is active. If the measurement function is active, the referenced function is executed during processing execution.</p> <p><b>Calculated Values</b> Deletes existing calculated values.</p> <p><b>Create Function</b> Creates a function file based on a standard template and opens it in the MATLAB Editor.</p> <p><b>Create Setting</b> Creates a setting file based on a standard template and opens it in the MATLAB Editor.</p> <p><b>Data Source</b> Displays the kind of data that is initially written on processing execution. The values can change when functions are executed. Refer to <a href="#">Raw Data, Measurement Types, and Measurement Data</a> on page 16.</p> <p><b>Default Value</b> Displays the default value of the measurement type variable.</p> <p><b>Edit Function</b> Opens the referenced function file in the MATLAB Editor.</p> <p><b>Edit Setting</b> Opens the referenced setting file in the MATLAB Editor.</p> <p><b>Evaluate Function</b> Evaluates the referenced function of this parameter.</p> <p><b>Factor</b> Lets you specify the conversion factor used to convert the raw data values to measurement type values.</p> <p><b>Function</b> Displays the path to the referenced function relative to the Function folder in the project's Pool.</p> <p><b>Measurement Data Name</b> Displays the measurement data name, which corresponds to the file name. This name is used to reference this measurement data in processing functions.</p> <p><b>Measurement Type Name</b> Displays the name of the measurement type that is used for creating this measurement data document.</p> <p><b>Offset</b> Lets you specify the conversion offset used to convert the raw data values to measurement type values.</p>

- Priority** Lets you specify the priority to define the execution order of the measurement function. A low priority number is processed before a higher priority number (e.g., priority 1 is executed before priority 2).
- Raw Name** Displays the name of the raw data variable.
- Raw Unit** Displays the unit of the raw data variable.
- Setting** Displays the path to the referenced settings file relative to the Setting folder in the project's Pool.
- Show Measurement Data** Opens the measurement data properties.
- User Comment** Lets you specify a text which is used to label the measurement variable.
- Variable Name** Displays the name of the measurement type variable.
- Variable Unit** Displays the unit of the measurement type variable.

Related topics

HowTos	
How to Calculate Measurement Data.....	25
How to Get Measurement Data by Mapping Raw Data.....	21
References	
Measurement Data.....	62



# Automation

Where to go from here

Information in this section

<a href="#">Programming ModelDesk Automation.....</a>	<a href="#">88</a>
<a href="#">Classes for Processing.....</a>	<a href="#">94</a>
Provides information on the classes for automation of the processing for parameters.	

# Programming ModelDesk Automation

## Where to go from here

## Information in this section

[Automatic Processing](#).....88

You can automate the processing.

[Overview of the Object Model for Processing](#).....92

The object model overview for processing gives a quick overview of object dependencies.

## Automatic Processing

### Introduction

You can automate the processing.

### Features

ModelDesk's tool automation provides the following features for processing:

- Creating and specifying measurement types
- Creating and specifying measurement data
- Specifying conversion rules.
- Mapping raw data to variables
- Assigning function and setting files to parameters.
- Specifying additional functions
- Configuring the plotting in MATLAB
- Executing the processing for the parameter set, parameter pages, parameters, or additional functions

### Reference information

The following listings show only short examples of the tool automation.

For an overview of the classes, refer to [Overview of the Object Model for Processing](#) on page 92.

For a full description of the classes, refer to [Classes for Processing](#) on page 94.

### Accessing the experiment

The following example shows how you can open a project and activate an experiment. Replace the project name and path by your own project in the script below.



```

from win32com.client import Dispatch
# Start ModelDesk, Load project and activate experiment
Application = Dispatch("ModelDesk.Application")
Application.Visible = True
MyProject = Application.OpenProject(r"C:\ExamplePath\Example_001\Example_001.CDP")
MyExperiments = MyProject.Experiments
MyExperiment = MyExperiments.Item(0)
MyActiveExperiment = MyExperiment.Activate(False)

```

A project is opened and an experiment is accessed in this part. For a description, refer to [Handling Projects and Experiments in Python \(ModelDesk Project and Experiment Management\)](#).

### Creating a measurement type

The following example shows how to create a measurement type.

```

# Get the measurement type configuration
MyMeasurementTypeConfiguration = MyActiveExperiment.MeasurementTypeConfiguration
# Add and activate a new measurement type
MyMeasurementTypeConfiguration.AddMeasurementType("MyMeasurementType")
# Activate the new measurement type
MyMeasurementType = MyMeasurementTypeConfiguration.\
ActivateMeasurementType("MyMeasurementType")
# Add variables to the measurement type
Variable = MyMeasurementType.Variables.Append("MyVar1", "m")
Variable.LowerLimit = 0
Variable.UpperLimit = 100
Variable = MyMeasurementType.Variables.Append("MyVar2", "kg")
Variable.LowerLimit = 0
Variable.UpperLimit = 1000
Variable.Description = "Weight"
# Save and close the measurement type
MyMeasurementType.Save()
MyMeasurementType.Close(True)

```

### Activating measurement data

The following example shows how to select the raw data and create the measurement data.

```

# Get the active parameter set of the experiment
MyActiveParameterSet = MyActiveExperiment.ActiveParameterSet
MyParameterSetProcessing = MyActiveParameterSet.ParameterSetProcessing
MyMeasurements = MyParameterSetProcessing.Measurements
# Get the first measurement type
MyMeasurementTypeContainer = MyMeasurements.Item(0)
# Add a new measurement data
MyMeasurementTypeContainer.AddMeasurementData("CustomMData.md")
# Get the active measurement data
MyActiveMeasurementData = MyMeasurementTypeContainer.\
ActivateMeasurementData("CustomMData.md")
# Get the Excel file containing raw data
RawData = MyActiveMeasurementData.SpecifyRawData(\
"E:\\Work\\Automation\\MyRawData.xlsx", "TestData", 1, 2, 3)
# If the Excel file has been modified, reload data
MyMeasurements = MyActiveMeasurementData.ReloadRawData()

```

```

# Map the raw data variable to measurement variables
MyMappingConfiguration = MyActiveMeasurementData.Mappings
# Try to map the variables automatically (comparing their names and units)
MyMappingConfiguration.AutoMap()
# If automatic mapping does not work, map variables manually
# Get a measurement variable (from measurement type)
MappingVariable = MyMappingConfiguration.Item(0)
# Get a raw data variable (from Excel)
RawDataVariable = RawData.RawDataVariables.Item(0)
# Do the mapping
MappingVariable.MapVariable(RawDataVariable)
# Specify the offset and factor for conversion
MappingVariable.ConversionRule.Offset = 0
MappingVariable.ConversionRule.Factor = 10
# Save the measurement data
MyActiveMeasurementData.Save()
# Close the measurement data
MyActiveMeasurementData.Close(True)

```

## Using measurement functions

The following example shows how to use measurement functions to calculate variables of the measurement type.

```

# Get the active parameter set
MyActiveParameterSet = MyActiveExperiment.ActiveParameterSet
# Get the processing info of the parameter set
MyParameterSetProcessing = MyActiveParameterSet.ParameterSetProcessing
# Get the measurements
MyMeasurements = MyParameterSetProcessing.Measurements
# Get a measurement type that is configured for this experiment
MyMeasurementTypeContainer = MyMeasurements.Item(0)
# Get the active measurement data
MyMeasurementData = MyMeasurementTypeContainer.MeasurementData
# Get a mapping variable
MyVariable = MyMeasurementData.Mappings.Item(0)
# Get the measurement function
MyMeasurementFunction = MyVariable.MeasurementFunction
# Set a function
MyMeasurementFunction.Function = "Example.m"
# You can execute the measurement function
MyMeasurementFunction.Execute()

```

## Specifying the general settings file

The following example shows how to specify general settings file.

```

# Get the active parameter set
MyActiveParameterSet = MyActiveExperiment.ActiveParameterSet
# Get the processing info of the parameter set
MyParameterSetProcessing.Configuration.GeneralSettings = "\\GeneralSetting.m"

```

## Specifying additional functions

The following example shows how to specify additional functions

```
# Get the active parameter set
MyActiveParameterSet = MyActiveExperiment.ActiveParameterSet
# Get the processing info of the parameter set
MyParameterSetProcessing = MyActiveParameterSet.ParameterSetProcessing
# Get the collection of additional functions
MyAdditionalFunctions = MyParameterSetProcessing.Configuration.AdditionalFunctions
# Add an additional function
MyAdditionalFunction = MyAdditionalFunctions.Add("Example.m")
# Specify settings for the additional function
MyAdditionalFunction.Priority = 5
MyAdditionalFunction.Active = True
# You can execute the additional function
MyAdditionalFunction.Execute()
```

## Executing processing

**Single parameter** The following example shows how to execute the processing for a single parameter.

```
# Get an object for the collection of parameter sets
MyParameterSets = MyActiveExperiment.ParameterSets
# Select the first parameter set in the collection
MyParameterSet = MyParameterSets.Item(0)
# Activate the parameter set
MyActiveParameterSet = MyParameterSet.Activate(True)
# Find the variable
MyParameter = MyActiveParameterSet.Find("Environment.AMBIENT.Const_pressure")
# Get the processing info of the parameter
MyParameterProcessingInfo = MyParameter.ParameterProcessingInfo
# Specify the processing setting, e.g., the m file
MyParameterProcessingInfo.Function = "Example.m"
# Execute the processing for the variable
MyParameterProcessingInfo.Execute()
```

**Parameter record** The following example shows how to execute the processing for a parameter record.

```
# Get the active parameter set
MyActiveParameterSet = MyActiveExperiment.ActiveParameterSet
# Get the main component
MyMainComponent = MyActiveParameterSet.MainComponents.Item(0)
# Get the parameter record
MyParameterRecord = MyMainComponent.ParameterRecords.Item(0)
# Get the processing info of the parameter record
MyParameterRecordProcessingInfo = MyParameterRecord.ParameterRecordProcessingInfo
MyParameterRecordProcessingInfo.Execute()
```

**Parameter set** The following example shows how to execute the processing for a parameter set.

```
# Get the active parameter set
MyActiveParameterSet = MyActiveExperiment.ActiveParameterSet
# Get the processing info of the parameter set
MyParameterSetProcessing = MyActiveParameterSet.ParameterSetProcessing
# Execute the processing
ParameterSetProcessing.Execute()
MyActiveExperiment.ActiveParameterSet.ParameterSetProcessing.Execute()
```

## Related topics

## Basics

Basics for Automating ModelDesk (ModelDesk Automation )  
 Handling Projects and Experiments in Python (ModelDesk Project and Experiment Management )

## References

Classes for Processing..... 94  
 Overview of the Object Model for Processing..... 92






## Overview of the Object Model for Processing

## Introduction

The object model overview for processing gives a quick overview of object dependencies.










## Symbols














The following symbols are used in the object model overview:

Symbol	Description
	Method, function
	Attribute (property, class)
	Collection
	Level of dependency (0, 1, 2, ...)
	Read only

## ParameterSetProcessing

The following table gives an overview of the processing object model:

Class	Level
<a href="#">ParameterSetProcessing</a> on page 144	
<a href="#">ProcessingConfiguration</a> on page 145	
<a href="#">PlotConfiguration</a> on page 138	
<a href="#">AdditionalFunctions</a> on page 104	
<a href="#">AdditionalFunction</a> on page 102	
<a href="#">MeasurementConfiguration</a> on page 118	
<a href="#">MeasurementTypeContainer</a> on page 130	
<a href="#">ActiveMeasurementData</a> on page 95	
<a href="#">RawData</a> on page 146	
<a href="#">RawDataVariables</a> on page 148	

Class	Level
<a href="#">RawDataVariable</a> on page 149	
<a href="#">ConversionFile</a> on page 108	
<a href="#">ConversionRules</a> on page 111	
<a href="#">ConversionRule</a> on page 109	
<a href="#">MappingConfiguration</a> on page 113	
<a href="#">MappingVariable</a> on page 116	
<a href="#">MeasurementFunction</a> on page 120	
<a href="#">ActiveMeasurementData</a> on page 95	
<a href="#">MeasurementTypeConfiguration</a> on page 126	
<a href="#">ActiveMeasurementTypes</a> on page 101	
<a href="#">MeasurementType</a> on page 122	
<a href="#">MeasurementTypeVariables</a> on page 134	
<a href="#">MeasurementTypeVariable</a> on page 133	

## Related topics

## References

<a href="#">Classes for Processing.....</a>	94
<a href="#">Enumerations for Processing.....</a>	150

## Classes for Processing

### Purpose

The following classes are used to automate the processing for parameters.

### Where to go from here

### Information in this section

<a href="#">ActiveMeasurementData</a> .....	95
To handle the measurement data.	
<a href="#">ActiveMeasurementTypes</a> .....	101
To handle the active measurement types.	
<a href="#">AdditionalFunction</a> .....	102
To specify and execute an additional function.	
<a href="#">AdditionalFunctions</a> .....	104
To handle the additional functions.	
<a href="#">ConversionFile</a> .....	108
To specify the conversion file.	
<a href="#">ConversionRule</a> .....	109
To specify a conversion rule.	
<a href="#">ConversionRules</a> .....	111
To handle the conversion rules.	
<a href="#">MappingConfiguration</a> .....	113
To configure the mapping of raw data to variables.	
<a href="#">MappingVariable</a> .....	116
To specify mapping for a variable.	
<a href="#">MeasurementConfiguration</a> .....	118
To configure the measurement.	
<a href="#">MeasurementFunction</a> .....	120
To specify and execute a measurement function.	
<a href="#">MeasurementType</a> .....	122
To specify a measurement type.	
<a href="#">MeasurementTypeConfiguration</a> .....	126
To handle the measurement types of an experiment.	
<a href="#">MeasurementTypeContainer</a> .....	130
To handle the measurement data.	
<a href="#">MeasurementTypeVariable</a> .....	133
To specify a variable of a measurement type.	
<a href="#">MeasurementTypeVariables</a> .....	134
To handle all the variables of a measurement type.	

<a href="#">PlotConfiguration</a> .....	138
To configure the plotting.	
<a href="#">ParameterProcessingInfo</a> .....	140
To handle the processing of a parameter.	
<a href="#">ParameterRecordProcessingInfo</a> .....	142
To handle the processing of a parameter record.	
<a href="#">ParameterSetProcessing</a> .....	144
To handle the processing of a parameter set.	
<a href="#">ProcessingConfiguration</a> .....	145
To configure the general settings, plotting, and additional functions for the processing.	
<a href="#">RawData</a> .....	146
To get information on the raw data.	
<a href="#">RawDataVariables</a> .....	148
To handle the variables of the raw data.	
<a href="#">RawDataVariable</a> .....	149
To get a raw data variable.	
<a href="#">Enumerations for Processing</a> .....	150

#### Information in other sections

<a href="#">Overview of the Object Model for Processing</a> .....	92
The object model overview for processing gives a quick overview of object dependencies.	
<a href="#">Automatic Processing</a> .....	88
You can automate the processing.	

## ActiveMeasurementData

**Purpose** To handle the measurement data.

**Where to go from here** Information in this section

<a href="#">Class Description (ActiveMeasurementData)</a> .....	96
To describe the class and its attributes.	

<a href="#">ChangeConversionFile.....</a>	<a href="#">97</a>
To change the conversion file.	
<a href="#">Close.....</a>	<a href="#">98</a>
To close the measurement data.	
<a href="#">CreateCopy.....</a>	<a href="#">99</a>
To create a copy of the measurement data.	
<a href="#">Save.....</a>	<a href="#">99</a>
To save the measurement data.	
<a href="#">SpecifyRawData.....</a>	<a href="#">100</a>
To specify the raw data for the measurement data.	

## Class Description (ActiveMeasurementData)

### Syntax

```
ActiveMeasurementData = MeasurementTypeContainer.Measurement
```

or

```
ActiveMeasurementData = MeasurementTypeContainer.ActivateMeasurementData
```

### Purpose

To handle the measurement data.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the active measurement data.
MtypeName	String	To get the name of the measurement type which is assigned to the measurement data.
Description	String	To get/set a description of the measurement data.
ModifiedBy	String	To get the name of the author which has modified the measurement data.
LastModified	Time	To get the date when the measurement data has been modified the last time.
RawData	RawData <sup>1)</sup>	To get/set the raw data for the measurement data.



Attributes	Type	Purpose
ConversionFile	ConversionFile <sup>2)</sup>	To get/set the conversion file for the measurement data.
Mapping	MappingConfiguration <sup>3)</sup>	To get the mapping configuration.

<sup>1)</sup> Refer to [RawData](#) on page 146.

<sup>2)</sup> Refer to [ConversionFile](#) on page 108.

<sup>3)</sup> Refer to [MappingConfiguration](#) on page 113.

## Methods

The class contains the following methods:

Method	Purpose
ChangeConversionFile	To change the conversion file. Refer to <a href="#">ChangeConversionFile</a> on page 97.
CreateCopy	To create a copy of the measurement data. Refer to <a href="#">CreateCopy</a> on page 99.
Close	To close the measurement data. Refer to <a href="#">Close</a> on page 98.
Save	To save the measurement data. Refer to <a href="#">Save</a> on page 99.
SpecifyRawData	To specify the raw data for the measurement data. Refer to <a href="#">SpecifyRawData</a> on page 100.

## Related topics

### References

[Class Description \(MeasurementTypeContainer\).....](#) 130

# ChangeConversionFile

## Class

ActiveMeasurementData

## Syntax

```
ConversionFile = ActiveMeasurementData.ChangeConversionFile(string
Filename, boolean SaveChanges)
```

## Purpose

To change the conversion file.

## Parameters

The method uses the following parameters:

Parameter	Type	Description
Filename	String	The file name of the conversion file.

Parameter	Type	Description
SaveChanges	Boolean	Specifies whether the current conversion file is saved when the new file is opened.

**Return value**

The method returns the following parameter:

Type	Description
ConversionFile <sup>1)</sup>	The conversion file.

<sup>1)</sup> Refer to [ConversionFile](#) on page 108.

**Related topics****References**

[Class Description \(ActiveMeasurementData\)..... 96](#)

## Close

**Class**

ActiveMeasurementData

**Syntax**

```
ActiveMeasurementData.Close(boolean SaveChanges)
```

**Purpose**

To close the measurement data.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
SaveChanges	Boolean	Specifies to save the modifications when the measurement data is closed.

**Return value**

—

**Related topics****References**

[Class Description \(ActiveMeasurementData\)..... 96](#)

## CreateCopy

**Class** ActiveMeasurementData

**Syntax** `ActiveMeasurementData.CreateCopy(string Filename, boolean OverwriteExisting)`

**Purpose** To create a copy of the measurement data.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
Filename	String	The file name.
OverwriteExisting	Boolean	Specifies whether a file with the same name should be overwritten.

**Return value** —

**Related topics**

References

[Class Description \(ActiveMeasurementData\)..... 96](#)

## Save

**Class** ActiveMeasurementData

**Syntax** `ActiveMeasurementData.Save()`

**Purpose** To save the measurement data.

**Parameters** —

**Return value**

—

**Related topics****References**[Class Description \(ActiveMeasurementData\)..... 96](#)

## SpecifyRawData

**Class**

ActiveMeasurementData

**Syntax**

```
RawData = ActiveMeasurementData.SpecifyRawData(string FilePath, string Sheet, int  
NameRow, int UnitRow, int DataRow)
```

**Purpose**

To specify the raw data for the measurement data.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
FilePath	string	The file path of the Excel file.
Sheet	string	The worksheet of the raw data in the Excel file.
NameRow	integer	The row of the worksheet which contains the names of the variables.
UnitRow	integer	The row of the worksheet which contains the units of the variables.
DataRow	integer	The row of the worksheet which contains the data of the variables.

**Return value**

The method returns the following parameter:

Type	Description
RawData <sup>1)</sup>	The raw data object.

<sup>1)</sup> Refer to [RawData](#) on page 146.**Related topics****References**[Class Description \(ActiveMeasurementData\)..... 96](#)

# ActiveMeasurementTypes

**Purpose** To handle the active measurement types.

**Where to go from here** **Information in this section**

<a href="#">Class Description (ActiveMeasurementTypes).....</a>	<a href="#">101</a>
To describe the class and its attributes.	
<a href="#">Item.....</a>	<a href="#">102</a>
To get a specific measurement type.	

## Class Description (ActiveMeasurementTypes)

**Syntax** `ActiveMeasurementTypes = MeasurementTypeConfiguration.ActiveMeasurementTypes`

**Purpose** To handle the active measurement types.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
Count	long	To get the umber of active measurement types.

**Methods** The class contains the following methods:

Method	Purpose
Item	To get a specific measurement type. Refer to <a href="#">Item</a> on page 102.

**Related topics**

**References**

<a href="#">Class Description (MeasurementTypeConfiguration).....</a>	<a href="#">126</a>
---	---------------------

## Item

**Class** ActiveMeasurementTypes

**Syntax** `ActiveMeasurementType = ActiveMeasurementTypes.Item(variant ID)`

**Purpose** To get a specific measurement type.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
ID	variant	Index of the measurement type to be returned. It can be numeric (starting at 0) or the name of a measurement type.

**Return value** The method returns the following parameter:

Type	Description
MeasurementType <sup>1)</sup>	The specific measurement type object.

<sup>1)</sup> Refer to [MeasurementType](#) on page 122.

### Related topics

#### References

[Class Description \(ActiveMeasurementTypes\)..... 101](#)

## AdditionalFunction

**Purpose** To specify and execute an additional function.

**Where to go from here** Information in this section

[Class Description \(AdditionalFunction\)..... 103](#)  
To describe the class and its attributes.

[Execute](#)..... 104  
To execute the additional function.

## Class Description (AdditionalFunction)

### Syntax

```
AdditionalFunction = AdditionalFunctions.Item(long Index)
```

or

```
AdditionalFunction = AdditionalFunctions.Add(string FilePath)
```

### Purpose

To specify and execute an additional function.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Function	String	To get/set the name of the m file that is used as function.
Priority	Long	To get/set the priority which specifies the execution sequence.
Active	Boolean	To get/set the active state.
LastFunctionRun	String	To get the time when the function has been calculated the last time.
UserComment	String	To get/set a comment for the function.
ExecutionIndex	Long	To get the position in the sequence when the function is executed.

### Methods

The class contains the following methods:

Method	Purpose
Execute	To execute the additional function. Refer to <a href="#">Execute</a> on page 104.

### Related topics

#### References

[Class Description \(AdditionalFunctions\)](#)..... 105

## Execute

<b>Class</b>	AdditionalFunction
<b>Syntax</b>	<code>AdditionalFunction.Execute()</code>
<b>Purpose</b>	To execute the additional function.
<b>Parameters</b>	–
<b>Return value</b>	–
<b>Related topics</b>	<p>HowTos</p> <p><a href="#">How to Start Calculating the Parameter Values.....</a> 40</p> <p>References</p> <p><a href="#">Class Description (AdditionalFunction).....</a> 103</p>

## AdditionalFunctions

<b>Purpose</b>	To handle the additional functions.
<b>Where to go from here</b>	<p>Information in this section</p> <p><a href="#">Class Description (AdditionalFunctions).....</a> 105 To describe the class and its attributes.</p> <p><a href="#">Add.....</a> 105 To add an additional function.</p> <p><a href="#">Item.....</a> 106 To access an additional function.</p> <p><a href="#">Remove.....</a> 107 To remove an additional function.</p>



## Class Description (AdditionalFunctions)

### Syntax

```
AdditionalFunctions = ProcessingConfiguration.AdditionalFunctions
```

### Purpose

To handle the additional functions.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	long	To get the number of additional functions.
AdditionalFunction	AdditionalFunction <sup>1)</sup>	To get/set the additional functions.

<sup>1)</sup> Refer to [AdditionalFunction](#) on page 102.

### Methods

The class contains the following methods:

Method	Purpose
Add	To add an additional function. Refer to <a href="#">Add</a> on page 105.
Item	To access an additional function. Refer to <a href="#">Item</a> on page 106.
Remove	To remove an additional function. Refer to <a href="#">Remove</a> on page 107.

### Related topics

#### References

[Class Description \(ProcessingConfiguration\)](#)..... 145

## Add

### Class

AdditionalFunctions

### Syntax

```
AdditionalFunctions = AdditionalFunctions.Add(string FilePath)
```

### Purpose

To add an additional function.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
FilePath	string	Specifies the file path of the additional function.

**Return value**

The method returns the following parameter:

Type	Description
AdditionalFunction <sup>1)</sup>	The added additional function.

<sup>1)</sup> Refer to [AdditionalFunction](#) on page 102.

**Related topics****References**

[Class Description \(AdditionalFunctions\).....](#) 105

## Item

**Class**

AdditionalFunctions

**Syntax**

```
AdditionalFunction = AdditionalFunctions.Item(long Index)
```

**Purpose**

To access an additional function.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
Index	long	The index of the additional function.

**Return value**

The method returns the following parameter:

Type	Description
AdditionalFunction <sup>1)</sup>	The specific additional function.

<sup>1)</sup> Refer to [AdditionalFunction](#) on page 102.

---

**Related topics****References**

[Class Description \(AdditionalFunctions\)..... 105](#)

## Remove

---

**Class**

AdditionalFunctions

---

**Syntax**

```
AdditionalFunctions.Remove(long Index)
```

---

**Purpose**

To remove an additional function.

---

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
Index	long	The index of the additional function to be removed.

---

**Return value**

—

---

**Related topics****References**

[Class Description \(AdditionalFunctions\)..... 105](#)

# ConversionFile

**Purpose** To specify the conversion file.

**Where to go from here** **Information in this section**

<a href="#">Class Description (ConversionFile).....</a>	<a href="#">108</a>
To describe the class and its attributes.	
<a href="#">Save.....</a>	<a href="#">109</a>
To save the conversion file.	

## Class Description (ConversionFile)

**Syntax** `ConversionFile = ActiveMeasurementData.ConversionFile`

**Purpose** To specify the conversion file.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
FilePath	String	To get the file path of the conversion file.
Description	String	To get/set a description of the conversion file.
ModifiedBy	String	To get the name of the author who has modified the conversion file.
LastModified	Time	To get the time when the conversion file has been modified the last time.
IsModified	Boolean	To get the information whether the conversion file has been modified.
ConversionRules	ConversionRules <sup>1)</sup>	To get the conversion rules object.

<sup>1)</sup> Refer to [ConversionRules](#) on page 111.

**Methods** The class contains the following methods:

Method	Purpose
Save	To save the conversion file. Refer to <a href="#">Save</a> on page 109.

**Related topics****References**

[Class Description \(ActiveMeasurementData\)..... 96](#)

## Save

**Class**

ConversionFile

**Syntax**`ConversionFile.Save()`**Purpose**

To save the conversion file.

**Parameters**

—

**Return value**

—

**Related topics****References**

[Class Description \(ConversionFile\)..... 108](#)

## ConversionRule

**Purpose**

To specify a conversion rule.

### Class Description (ConversionRule)

**Syntax**`ConversionRule = ConversionRules.Item()`

or

```
ConversionRule = ConversionRules.Find()
```

---

**Purpose**

To specify a conversion rule.

---

**Description**

When you map the raw data variables to measurement type variables, you can specify conversion parameters to adapt their units. The parameters are a factor and an offset which are used in the following formula:

$$\text{Measurement\_Type\_Variable} = \text{Factor} \cdot \text{Raw\_Data\_Variable} + \text{Offset}$$

---

**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
RawUnit	String	To get the unit of the raw data variable.
VariableUnit	String	To get the unit of the measurement type variable.
Factor	String	To get/set the value of the factor.
Offset	String	To get/set the value of the offset.

---

**Methods**

—

---

**Related topics****HowTos**

[How to Get Measurement Data by Mapping Raw Data.....](#) 21

**References**

[Class Description \(ConversionRules\).....](#) 111

# ConversionRules

**Purpose** To handle the conversion rules.

**Where to go from here** **Information in this section**

<a href="#">Class Description (ConversionRules)</a> .....	111
To describe the class and its attributes.	
<a href="#">Find</a> .....	112
To find a conversion rule.	
<a href="#">Item</a> .....	112
To access a conversion rule.	

## Class Description (ConversionRules)

**Syntax** `ConversionRules = ConversionFile.ConversionRules`

**Purpose** To handle the conversion rules.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
Count	long	To get the number of conversion rules.

**Methods** The class contains the following methods:

Method	Purpose
Find	To find a conversion rule. Refer to <a href="#">Find</a> on page 112.
Item	To access a conversion rule. Refer to <a href="#">Item</a> on page 112.

**Related topics**

**References**

<a href="#">Class Description (ConversionFile)</a> .....	108
--	-----

## Find

**Class** ConversionRules

**Syntax** `ConversionRule = ConversionRules.Find(string RawUnit, string VariableUnit)`

**Purpose** To find a conversion rule.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
RawUnit	string	The unit of the raw data
VariableUnit	string	The unit of the variable

**Return value** The method returns the following parameter:

Type	Description
ConversionRule <sup>1)</sup>	The found conversion rule.

<sup>1)</sup> Refer to [ConversionRule](#) on page 109.

**Related topics**

**References**

[Class Description \(ConversionRules\)..... 111](#)

## Item

**Class** ConversionRules

**Syntax** `ConversionRule = ConversionRules.Item(integer Index)`

**Purpose** To access a conversion rule.



**Parameters**

The method uses the following parameters:

Parameter	Type	Description
Index	integer	The index of the conversion rule

**Return value**

The method returns the following parameter:

Type	Description
ConversionRule <sup>1)</sup>	The specific conversion rule

<sup>1)</sup> Refer to [ConversionRule](#) on page 109.

**Related topics****References**

[Class Description \(ConversionRules\).....](#) 111

## MappingConfiguration

**Purpose**

To configure the mapping of raw data to variables.

**Where to go from here****Information in this section**

[Class Description \(MappingConfiguration\).....](#) 113  
To describe the class and its attributes.

[AutoMap.....](#) 114  
To automatically map raw data and variables.

[Item.....](#) 115  
To access a mapping variable object.

## Class Description (MappingConfiguration)

**Syntax**

```
MappingConfiguration = ActiveMeasurementData.Mapping
```

**Purpose** To configure the mapping of raw data to variables.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
Count	long	To get the number of mapping variable objects.

**Methods** The class contains the following methods:

Method	Purpose
AutoMap	To automatically map raw data and variables. Refer to <a href="#">AutoMap</a> on page 114.
Item	To access a mapping variable object. Refer to <a href="#">Item</a> on page 115.

**Related topics**

References

[Class Description \(ActiveMeasurementData\)..... 96](#)

## AutoMap

**Class** MappingConfiguration

**Syntax** `RetVal = MappingConfiguration.AutoMap()`

**Purpose** To automatically map raw data and variables.

**Parameters** —

**Return value** The method returns the following parameter:

Type	Description
long	Number of mapped variables.

**Related topics****References**

[Class Description \(MappingConfiguration\)..... 113](#)

## Item

**Class**

MappingConfiguration

**Syntax**

```
MappingVariable = MappingConfiguration.Item(object Index)
```

**Purpose**

To access a mapping variable object.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
Index	object	Index of the mapping object.

**Return value**

The method returns the following parameter:

Type	Description
MappingVariable <sup>1)</sup>	The mapping variable object.

<sup>1)</sup> Refer to [MappingVariable](#) on page 116.

**Related topics****References**

[Class Description \(MappingConfiguration\)..... 113](#)

# MappingVariable

**Purpose** To specify mapping for a variable.

**Where to go from here**

**Information in this section**

<a href="#">Class Description (MappingVariable).....</a>	<a href="#">116</a>
To describe the class and its attributes.	
<a href="#">MapVariable.....</a>	<a href="#">117</a>
To map a raw data variable.	
<a href="#">RemoveMapping.....</a>	<a href="#">118</a>
To remove a mapping.	

## Class Description (MappingVariable)

**Syntax**

```
MappingVariable = MappingConfiguration.Item()
```

**Purpose** To specify mapping for a variable.

**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
ConversionRule	ConversionRule <sup>1)</sup>	To get the conversion rule.
DataSource	EnumDataSource <sup>2)</sup>	To get the information whether the measurement variable is mapped to a raw data variable or calculated by a measurement function.
MeasurementFunction	MeasurementFunction <sup>3)</sup>	To get the measurement function.
RawName	string	To get the name of the raw data.
RawUnit	string	To get the unit of the raw data.
State	MappingState <sup>4)</sup>	To get the mapping state.
VariableName	string	To get the name of the variable.
VariableUnit	string	To get the unit of the variable.

<sup>1)</sup> Refer to [ConversionRule](#) on page 109.

<sup>2)</sup> Refer to [EnumDataSource](#) on page 150.

<sup>3)</sup> Refer to [MeasurementFunction](#) on page 120.

<sup>4)</sup> Refer to [MappingState](#) on page 151.

**Methods**

The class contains the following methods:

Method	Purpose
MapVariable	To map a raw data variable. Refer to <a href="#">MapVariable</a> on page 117.
RemoveMapping	To remove a mapping. Refer to <a href="#">RemoveMapping</a> on page 118.

**Related topics****References**

[Class Description \(MappingConfiguration\)..... 113](#)

## MapVariable

**Class**

MappingVariable

**Syntax**

```
MappingState = MappingVariable.MapVariable(RawDataVariable RdVar)
```

**Purpose**

To map a raw data variable.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
RdVar	RawDataVariable <sup>1)</sup>	The raw data variable to be mapped.

<sup>1)</sup> Refer to [RawDataVariable](#) on page 149.

**Return value**

The method returns the following parameter:

Type	Description
MappingState <sup>1)</sup>	The mapping state.

<sup>1)</sup> Refer to [MappingState](#) on page 151.

**Related topics****References**

[Class Description \(MappingVariable\)..... 116](#)

## RemoveMapping

<b>Class</b>	MappingVariable
--------------	-----------------

<b>Syntax</b>	<code>MappingVariable.Remove()</code>
---------------	---------------------------------------

<b>Purpose</b>	To remove a mapping.
----------------	----------------------

<b>Parameters</b>	–
-------------------	---

<b>Return value</b>	–
---------------------	---

<b>Related topics</b>	References
-----------------------	------------

<a href="#">Class Description (MappingVariable).....</a>	116
--	-----

## MeasurementConfiguration

<b>Purpose</b>	To configure the measurement.
----------------	-------------------------------

<b>Where to go from here</b>	Information in this section
------------------------------	-----------------------------

<a href="#">Class Description (MeasurementConfiguration).....</a>	118
To describe the class and its attributes.	
<a href="#">Item.....</a>	119
To get a specific measurement type container.	

## Class Description (MeasurementConfiguration)

<b>Syntax</b>	<code>MeasurementConfiguration = ParameterSetProcessing.Measurement</code>
---------------	--

**Purpose** To configure the measurement.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
Count	long	To get the number of measurement type containers.

**Methods** The class contains the following methods:

Method	Purpose
Item	To get a specific measurement type container. Refer to <a href="#">Item</a> on page 119.

**Related topics**

References

[Class Description \(ParameterSetProcessing\).....](#) 144

## Item

**Class** MeasurementConfiguration

**Syntax** `MeasurementTypeContainer = MeasurementConfiguration.Item(variant ID)`

**Purpose** To get a specific measurement type container.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
ID	variant	Index of the measurement type container. The value can be specified as string (name) or integer (index).

**Return value**

The method returns the following parameter:

Type	Description
MeasurementTypeContainer <sup>1)</sup>	The measurement type container.

<sup>1)</sup> Refer to [MeasurementTypeContainer](#) on page 130.

**Related topics****References**

[Class Description \(MeasurementConfiguration\)..... 118](#)

## MeasurementFunction

**Purpose**

To specify and execute a measurement function.

**Where to go from here****Information in this section**

[Class Description \(MeasurementFunction\)..... 120](#)  
To describe the class and its attributes.

[DeleteValues..... 121](#)  
To delete the values calculated by the measurement function.

[Execute..... 122](#)  
To execute the measurement function.

## Class Description (MeasurementFunction)

**Syntax**

```
MeasurementFunction = MappingVariable.MeasurementFunction
```

**Purpose**

To specify and execute a measurement function.



**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
Active	Boolean	To get/set the flag whether the measurement function is active.
ExecutionIndex	Long	To get the execution index.
Function	String	To get/set the name of the m file which is used for the calculation.
Priority	Long	To get/set the priority of the measurement function.
Setting	String	To get/set the name of the m file which is used for the settings.
Values	Double	To get the values

**Methods**

The class contains the following methods:

Method	Purpose
DeleteValues	To delete the values calculated by the measurement function. Refer to <a href="#">DeleteValues</a> on page 121.
Execute	To execute the measurement function. Refer to <a href="#">Execute</a> on page 122.

**Related topics****References**

[Class Description \(MappingVariable\)](#)..... 116

## DeleteValues

**Class**

MeasurementFunction

**Syntax**

```
MeasurementFunction.DeleteValues()
```

**Purpose**

To delete the values calculated by the measurement function.

**Parameters**

—

**Return value**

—

**Related topics****References**[Class Description \(MeasurementFunction\)..... 120](#)

## Execute

**Class**

MeasurementFunction

**Syntax**`MeasurementFunction.Execute()`**Purpose**

To execute the measurement function.

**Parameters**

—

**Return value**

—

**Related topics****References**[Class Description \(MeasurementFunction\)..... 120](#)

## MeasurementType

**Purpose**

To specify a measurement type.

**Where to go from here****Information in this section**[Class Description \(MeasurementType\)..... 123](#)  
To describe the class and its attributes.

<a href="#">Close.....</a>	124
To close the measurement type.	
<a href="#">CreateCopy.....</a>	124
To create a copy of the measurement type.	
<a href="#">Save.....</a>	125
To save the measurement type.	

## Class Description (MeasurementType)

### Syntax

```
MeasurementType = ActiveMeasurementTypes.Item()
```

### Purpose

To specify a measurement type.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	string	To get the name of the measurement type.
LastModified	Time	To get the time when the measurement type has been modified.
ModifiedBy	string	To get the name of the author who has modified the measurement type.
Description	string	To get/set a description for the measurement type.
Variables	MeasurementTypeVariables <sup>1)</sup>	To access the variables of the measurement type.

<sup>1)</sup> Refer to [MeasurementTypeVariables](#) on page 134.

### Methods

The class contains the following methods:

Method	Purpose
Close	To close the measurement type. Refer to <a href="#">Close</a> on page 124.
CreateCopy	To create a copy of the measurement type. Refer to <a href="#">CreateCopy</a> on page 124.
Save	To save the measurement type. Refer to <a href="#">Save</a> on page 125.

**Related topics****References**[Class Description \(ActiveMeasurementTypes\)..... 101](#)

## Close

**Class**

MeasurementType

**Syntax**`MeasurementType.Close(boolean SaveChanges)`**Purpose**

To close the measurement type.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
SaveChanges	Boolean	Specifies whether the modification are saved when the measurement type is closed.

**Return value**

—

**Related topics****References**[Class Description \(MeasurementType\)..... 123](#)

## CreateCopy

**Class**

MeasurementType

**Syntax**`MeasurementType.CreateCopy(string FileName, boolean OverwriteExisting)`

**Purpose** To create a copy of the measurement type.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
FileName	String	Specifies the file name of the copied measurement type.
OverwriteExisting	Boolean	Specifies whether an existing file which has the same name is overwritten.

**Return value** —

**Related topics**

References

[Class Description \(MeasurementType\)..... 123](#)

## Save

**Class** MeasurementType

**Syntax** `MeasurementType.Save()`

**Purpose** To save the measurement type.

**Parameters** —

**Return value** —

**Related topics**

References

[Class Description \(MeasurementType\)..... 123](#)

# MeasurementTypeConfiguration

**Purpose** To handle the measurement types of an experiment.

**Where to go from here** Information in this section

<a href="#">Class Description (MeasurementTypeConfiguration).....</a>	<a href="#">126</a>
To describe the class and its attributes.	
<a href="#">ActivateMeasurementType.....</a>	<a href="#">127</a>
To activate a measurement type.	
<a href="#">AddMeasurementType.....</a>	<a href="#">128</a>
To create a measurement type file and add the measurement type to the configuration.	
<a href="#">DeactivateMeasurementType.....</a>	<a href="#">128</a>
To deactivate a measurement type.	
<a href="#">RemoveMeasurementType.....</a>	<a href="#">129</a>
To remove a measurement type from the configuration and delete the corresponding file.	

## Class Description (MeasurementTypeConfiguration)

**Syntax** `MeasurementTypeConfiguration =  
ActiveExperiment.MeasurementTypeConfiguration`

**Purpose** To handle the measurement types of an experiment.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
AvailableElements	string	To get the names of all the available measurement types.
ActiveMeasurementTypes	ActiveMeasurementTypes <sup>1)</sup>	To get the active measurement types.

<sup>1)</sup> Refer to [ActiveMeasurementTypes](#) on page 101.

**Methods**

The class contains the following methods:

Method	Purpose
ActivateMeasurementType	To activate a measurement type. Refer to <a href="#">ActivateMeasurementType</a> on page 127.
AddMeasurementType	To create a measurement type file and add the measurement type to the configuration. Refer to <a href="#">AddMeasurementType</a> on page 128.
DeactivateMeasurementType	To deactivate a measurement type. Refer to <a href="#">DeactivateMeasurementType</a> on page 128.
RemoveMeasurementType	To remove a measurement type from the configuration and delete the corresponding file. Refer to <a href="#">RemoveMeasurementType</a> on page 129.

**Related topics****References**

[Class Description \(ActiveExperiment\) \(ModelDesk Project and Experiment Management !\[\]\(8d0f0e0fe25b320c33272c52aec1fbca\_img.jpg\)](#))

## ActivateMeasurementType

**Class**

MeasurementTypeConfiguration

**Syntax**

```
MeasurementType = MeasurementTypeConfiguration.ActivateMeasurementType(string MeasurementTypeName)
```

**Purpose**

To activate a measurement type.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementTypeName	string	The name of the measurement type to be activated.

**Return value**

The method returns the following parameter:

Type	Description
MeasurementType <sup>1)</sup>	The active measurement type.

<sup>1)</sup> Refer to [MeasurementType](#) on page 122.

**Related topics****References**[Class Description \(MeasurementTypeConfiguration\)..... 126](#)

## AddMeasurementType

**Class**

MeasurementTypeConfiguration

**Syntax**

```
MeasurementTypeConfiguration.AddMeasurementType(string MeasurementTypeName)
```

**Purpose**

To create a measurement type file and add the measurement type to the configuration.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementTypeName	string	The name of the measurement type to be added.

**Return value**

—

**Related topics****References**[Class Description \(MeasurementTypeConfiguration\)..... 126](#)

## DeactivateMeasurementType

**Class**

MeasurementTypeConfiguration

**Syntax**

```
MeasurementTypeConfiguration.DeactivateMeasurementType(string MeasurementTypeName)
```

**Purpose**

To deactivate a measurement type.



**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementTypeName	string	The name of the measurement type to be deactivated.

**Return value**

—

**Related topics****References**

[Class Description \(MeasurementTypeConfiguration\)..... 126](#)

## RemoveMeasurementType

**Class**

MeasurementTypeConfiguration

**Syntax**

```
MeasurementTypeConfiguration.RemoveMeasurementType(string MeasurementTypeName)
```

**Purpose**

To remove a measurement type from the configuration and delete the corresponding file.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementTypeName	string	The name of the measurement type to be removed.

**Return value**

—

**Related topics****References**

[Class Description \(MeasurementTypeConfiguration\)..... 126](#)

# MeasurementTypeContainer

**Purpose** To handle the measurement data.

**Where to go from here** Information in this section

<a href="#">Class Description (MeasurementTypeContainer).....</a>	<a href="#">130</a>
To describe the class and its attributes.	
<a href="#">ActivateMeasurementData.....</a>	<a href="#">131</a>
To activate a measurement data.	
<a href="#">AddMeasurementData.....</a>	<a href="#">132</a>
To create a measurement data file and add the measurement data to the configuration.	
<a href="#">DeactivateMeasurementData.....</a>	<a href="#">132</a>
To deactivate the active measurement data.	
<a href="#">RemoveMeasurementData.....</a>	<a href="#">133</a>
To remove a measurement data from the configuration and delete the corresponding file.	

## Class Description (MeasurementTypeContainer)

**Syntax** `MeasurementTypeContainer = MeasurementConfiguration.Item()`

**Purpose** To handle the measurement data.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
Name	string	To get the name of the measurement type container.
AvailableElements	strings	To get the names of all the available measurement data
MeasurementData	ActiveMeasurementData <sup>1)</sup>	To get the active measurement data.

<sup>1)</sup> Refer to [ActiveMeasurementData](#) on page 95.

**Methods**

The class contains the following methods:

Method	Purpose
ActivateMeasurementData	To activate a measurement data. Refer to <a href="#">ActivateMeasurementData</a> on page 131.
AddMeasurementData	To create a measurement data file and add the measurement data to the configuration. Refer to <a href="#">AddMeasurementData</a> on page 132.
DeactivateMeasurementData	To deactivate the active measurement data. Refer to <a href="#">DeactivateMeasurementData</a> on page 132.
RemoveMeasurementData	To remove a measurement data from the configuration and delete the corresponding file. Refer to <a href="#">RemoveMeasurementData</a> on page 133.

**Related topics****References**

[Class Description \(MeasurementConfiguration\)](#)..... 118

## ActivateMeasurementData

**Class**

MeasurementTypeContainer

**Syntax**

```
ActiveMeasurementData =
MeasurementTypeContainer.ActivateMeasurementData(string
MeasurementDataName)
```

**Purpose**

To activate a measurement data.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementDataName	string	The name of the measurement data to be activated.

**Return value**

The method returns the following parameter:

Type	Description
ActiveMeasurementData <sup>1)</sup>	The activated measurement data.

<sup>1)</sup> Refer to [ActiveMeasurementData](#) on page 95.

**Related topics****References**

[Class Description \(MeasurementTypeContainer\)..... 130](#)

## AddMeasurementData

**Class**

MeasurementTypeContainer

**Syntax**

```
MeasurementTypeContainer.AddMeasurementData(string MeasurementDataName)
```

**Purpose**

To create a measurement data file and add the measurement data to the configuration.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementDataName	string	The name of the measurement data to be added.

**Return value**

—

**Related topics****References**

[Class Description \(MeasurementTypeContainer\)..... 130](#)

## DeactivateMeasurementData

**Class**

MeasurementTypeContainer

**Syntax**

```
MeasurementTypeContainer.DeactivateMeasurementData()
```

**Purpose**

To deactivate the active measurement data.

**Parameters**

–

**Return value**

–

**Related topics****References**
[Class Description \(MeasurementTypeContainer\)..... 130](#)

## RemoveMeasurementData

**Class**

MeasurementTypeContainer

**Syntax**

```
MeasurementTypeContainer.RemoveMeasurementData(string MeasurementDataName)
```

**Purpose**

To remove a measurement data from the configuration and delete the corresponding file.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
MeasurementDataName	string	The name of the measurement data to be removed.

**Return value**

–

**Related topics****References**
[Class Description \(MeasurementTypeContainer\)..... 130](#)

## MeasurementTypeVariable

**Purpose**

To specify a variable of a measurement type.

## Class Description (MeasurementTypeVariable)

### Syntax

```
MeasurementTypeVariable = MeasurementTypeVariables.Item()
```

OR

```
MeasurementTypeVariable = MeasurementTypeVariables.Append()
```

### Purpose

To specify a variable of a measurement type.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	string	To get the name of the variable.
Unit	string	To get the unit of the variable.
Description	string	To get/set the description of the variable.
DefaultValue	double	To get/set the default value of the variable.
LowerLimit	double	To get/set the lower limit of the variable.
UpperLimit	double	To get/set the upper limit of the variable.

### Methods

—

### Related topics

#### References

[Class Description \(MeasurementTypeVariables\)](#)..... 135

## MeasurementTypeVariables

### Purpose

To handle all the variables of a measurement type.

### Where to go from here

#### Information in this section

[Class Description \(MeasurementTypeVariables\)](#)..... 135

To describe the class and its attributes.

[Append](#)..... 136

To append a variable to the measurement type.

<a href="#">ChangeOrder.....</a>	<a href="#">136</a>
To change the order of variables of the measurement type.	
<a href="#">Item.....</a>	<a href="#">137</a>
To access a specific variable of the measurement type.	
<a href="#">Remove.....</a>	<a href="#">138</a>
To remove a variable from the measurement type.	

## Class Description (MeasurementTypeVariables)

### Syntax

```
MeasurementTypeVariables = MeasurementType.Variables
```

### Purpose

To handle all the variables of a measurement type.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	integer	To get the number of variables.

### Methods

The class contains the following methods:

Method	Purpose
Append	To append a variable to the measurement type. Refer to <a href="#">Append</a> on page 136.
ChangeOrder	To change the order of variables of the measurement type. Refer to <a href="#">ChangeOrder</a> on page 136.
Item	To access a specific variable of the measurement type. Refer to <a href="#">Item</a> on page 137.
Remove	To remove a variable from the measurement type. Refer to <a href="#">Remove</a> on page 138.

### Related topics

#### References

<a href="#">Class Description (MeasurementType).....</a>	<a href="#">123</a>
--	---------------------

## Append

**Class** MeasurementTypeVariables

**Syntax** `MeasurementTypeVariable = MeasurementTypeVariables.Append(string Name, string Unit)`

**Purpose** To append a variable to the measurement type.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
Name	string	The name of the variable to be appended.
Unit	string	The unit of the variable to be appended.

**Return value** The method returns the following parameter:

Type	Description
MeasurementTypeVariable <sup>1)</sup>	The new variable.

<sup>1)</sup> Refer to [MeasurementTypeVariable](#) on page 133.

**Related topics**

**References**

[Class Description \(MeasurementTypeVariables\)..... 135](#)

## ChangeOrder

**Class** MeasurementTypeVariables

**Syntax** `MeasurementTypeVariables.ChangeOrder(integer OldIndex, integer NewIndex)`

**Purpose** To change the order of variables of the measurement type.



**Parameters**

The method uses the following parameters:

Parameter	Type	Description
OldIndex	integer	The old index
NewIndex	integer	The new index

**Return value**

—

**Related topics****References**

[Class Description \(MeasurementTypeVariables\)..... 135](#)

## Item

**Class**

MeasurementTypeVariables

**Syntax**

```
MeasurementTypeVariable = MeasurementTypeVariables.Item(integer Index)
```

**Purpose**

To access a specific variable of the measurement type.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
Index	integer	The index of the variable.

**Return value**

The method returns the following parameter:

Type	Description
MeasurementTypeVariable <sup>1)</sup>	The specific variable

<sup>1)</sup> Refer to [MeasurementTypeVariable](#) on page 133.

**Related topics****References**

[Class Description \(MeasurementTypeVariables\)..... 135](#)

# Remove

**Class** MeasurementTypeVariables

**Syntax** `MeasurementTypeVariables.Remove(integer Index)`

**Purpose** To remove a variable from the measurement type.

**Parameters** The method uses the following parameters:

Parameter	Type	Description
Index	integer	The index of the variable to be removed.

**Return value** –

<b>Related topics</b>	<b>References</b>
	<a href="#">Class Description (MeasurementTypeVariables).....</a> 135

# PlotConfiguration

**Purpose** To configure the plotting.

<b>Where to go from here</b>	<b>Information in this section</b>
	<a href="#">Class Description (PlotConfiguration).....</a> 139 To describe the class and its attributes.
	<a href="#">ClosePlots.....</a> 139 To close all the plots.

## Class Description (PlotConfiguration)

### Syntax

```
PlotConfiguration = ProcessingConfiguration.PlotConfiguration
```

### Purpose

To configure the plotting.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Delay	String	To get the delay value.
ErrorPlots	Boolean	To get the information whether error plots are created.
IncrementalPlots	Boolean	To get the information whether incremental plots are created.
PlotContext	EnumPlotContext <sup>1)</sup>	To get the context of plotting (parameter, parameter page, or parameter set).
PlotType	EnumPlotType <sup>2)</sup>	To get the type of plotting (standard, measurement, measurement label).

<sup>1)</sup> Refer to [EnumPlotContext](#) on page 151.

<sup>2)</sup> Refer to [EnumPlotType](#) on page 151.

### Methods

The class contains the following methods:

Method	Purpose
ClosePlots	To close all the plots. Refer to <a href="#">ClosePlots</a> on page 139.

### Related topics

#### References

[Class Description \(ProcessingConfiguration\)](#)..... 145

## ClosePlots

### Class

PlotConfiguration

### Syntax

```
PlotConfiguration.ClosePlots()
```

---

<b>Purpose</b>	To close all the plots.
----------------	-------------------------

---

<b>Parameters</b>	–
-------------------	---

---

<b>Return value</b>	–
---------------------	---

---

**Related topics****References**

<a href="#">Class Description (PlotConfiguration).....</a>	139
--	-----

## ParameterProcessingInfo

---

<b>Purpose</b>	To handle the processing of a parameter.
----------------	--

---

**Where to go from here****Information in this section**

<a href="#">Class Description (ParameterProcessingInfo).....</a>	140
To describe the class and its attributes.	
<a href="#">Execute.....</a>	141
To execute the processing of the parameter.	

## Class Description (ParameterProcessingInfo)

---

**Syntax**

```
ParameterProcessingInfo = BaseParameterType.Processing
```

---

<b>Purpose</b>	To handle the processing of a parameter.
----------------	--

**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
Function	String	To get/set the m file which is used for calculations in the processing of the parameter.
Setting	String	To get/set the m file which is used for settings in the processing of the parameter.
Active	Boolean	To get/set the active state of the processing of the parameter.
Priority	Long	To get/set the priority for the processing of the parameter.
LastFunctionRun	String	To get the time when the function has been executed last.
ExecutionIndex	Long	To get the actually execution sequence when the function is executed.

**Methods**

The class contains the following methods:

Method	Purpose
Execute	To execute the processing of the parameter. Refer to <a href="#">Execute</a> on page 141.

**Related topics****References**

[Class Description \(BaseParameterType\) \(ModelDesk Parameterizing 📖\)](#)

## Execute

**Class**

ParameterProcessingInfo

**Syntax**

```
ParameterProcessingInfo.Execute()
```

**Purpose**

To execute the processing of the parameter.

**Parameters**

–

**Return value**

–

**Related topics****References**[Class Description \(ParameterProcessingInfo\).....](#) 140

## ParameterRecordProcessingInfo

**Purpose**

To handle the processing of a parameter record.

**Where to go from here****Information in this section**[Class Description \(ParameterRecordProcessingInfo\).....](#) 142  
To describe the class and its attributes.[Execute.....](#) 143  
To execute the processing of the parameter record.

## Class Description (ParameterRecordProcessingInfo)

**Syntax**`ParameterRecordProcessingInfo = ParameterRecord.Processing`**Purpose**

To handle the processing of a parameter record.

**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
Active	Boolean	To get/set the active state of the processing of the parameter record.
Priority	Long	To get/set the priority for the processing of the parameter record.
ExecutionIndex	Long	To get the actually execution sequence when the functions are executed.

**Methods**

The class contains the following methods:

Method	Purpose
Execute	To execute the processing of the parameter record. Refer to <a href="#">Execute</a> on page 143.

**Related topics**

**References**

[Class Description \(ParameterRecord\) \(ModelDesk Parameterizing !\[\]\(313b3b3c8a0c38ad35f0f4cceb5f9abb\_img.jpg\)\)](#)

# Execute

**Class**

ParameterRecordProcessingInfo

**Syntax**

`ParameterRecordProcessingInfo.Execute()`

**Purpose**

To execute the processing of the parameter record.

**Parameters**

—

**Return value**

—

**Related topics**

**References**

[Class Description \(ParameterRecordProcessingInfo\)..... 142](#)

# ParameterSetProcessing

**Purpose** To handle the processing of a parameter set.

**Where to go from here** Information in this section

<a href="#">Class Description (ParameterSetProcessing)</a> .....	144
To describe the class and its attributes.	
<a href="#">Execute</a> .....	145
To execute the processing for the parameter set.	

## Class Description (ParameterSetProcessing)

**Syntax** `ParameterSetProcessing = ActiveParameterSet.Processing`

**Purpose** To handle the processing of a parameter set.

**Attributes** The class contains the following attributes:

Attributes	Type	Purpose
Configuration	ProcessingConfiguration <sup>1)</sup>	To get the processing configuration.
Measurement	MappingConfiguration <sup>2)</sup>	To get the measurement configuration.

<sup>1)</sup> Refer to [ProcessingConfiguration](#) on page 145.

<sup>2)</sup> Refer to [MappingConfiguration](#) on page 113.

**Methods** The class contains the following methods:

Method	Purpose
Execute	To execute the processing for the parameter set. Refer to <a href="#">Execute</a> on page 145.

**Related topics**

References

[ActiveParameterSet \(ModelDesk Parameterizing !\[\]\(4146d17f71dced09c6ad789cacceaa6d\_img.jpg\)\)](#)



## Execute

<b>Class</b>	ParameterSetProcessing
<b>Syntax</b>	<code>ParameterSetProcessing.Execute()</code>
<b>Purpose</b>	To execute the processing for the parameter set.
<b>Parameters</b>	–
<b>Return value</b>	–
<b>Related topics</b>	References <a href="#">Class Description (ParameterSetProcessing)..... 144</a>

## ProcessingConfiguration

<b>Purpose</b>	To configure the general settings, plotting, and additional functions for the processing.
----------------	---

### Class Description (ProcessingConfiguration)

<b>Syntax</b>	<code>ProcessingConfiguration = ParameterSetProcessing.Configuration</code>
<b>Purpose</b>	To configure the general settings, plotting, and additional functions for the processing.

**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
GeneralSettings	string	To get/set the general settings file.
PlotConfiguration	PlotConfiguration <sup>1)</sup>	To get the settings of the plotting.
AdditionalFunctions	AdditionalFunctions <sup>2)</sup>	To get the additional functions.

<sup>1)</sup> Refer to [PlotConfiguration](#) on page 138.

<sup>2)</sup> Refer to [AdditionalFunctions](#) on page 104.

**Methods**

—

**Related topics****References**

[Class Description \(ParameterSetProcessing\)](#)..... 144

## RawData

**Purpose**

To get information on the raw data.

**Where to go from here****Information in this section**

[Class Description \(RawData\)](#)..... 146

To describe the class and its attributes.

[ReloadRawData](#)..... 147

To reload the raw data file.

## Class Description (RawData)

**Syntax**

```
RawData = ActiveMeasurementData.RawData
```

**Purpose**

To get information on the raw data.

**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
FilePath	String	To get the file path of the Excel file.
Sheet	String	To get the worksheet of the raw data.
NameRow	Integer	To get the row of the names.
UnitRow	Integer	To get the row of the units.
DataRow	Integer	To get the row of the data.
Measurements	Integer	To get the measurements.
RawDataVariables	RawDataVariables <sup>1)</sup>	To get the variables of the raw data.

<sup>1)</sup> Refer to [RawDataVariables](#) on page 148.

**Methods**

The class contains the following methods:

Method	Purpose
ReloadRawData	To reload the raw data file. Refer to <a href="#">ReloadRawData</a> on page 147.

**Related topics****References**

[Class Description \(ActiveMeasurementData\).....](#) 96

## ReloadRawData

**Class**

RawData

**Syntax**

```
Measurements = RawData.ReloadRawData()
```

**Purpose**

To reload the raw data file.

**Parameters**

—

**Return value**

The method returns the following parameter:

Type	Description
Integer	Measurements, the number of data points

## Related topics

## References

[Class Description \(RawData\)](#)..... 146

## RawDataVariables

## Purpose

To handle the variables of the raw data.

## Where to go from here

## Information in this section

[Class Description \(RawDataVariables\)](#)..... 148  
To describe the class and its attributes.

[Item](#)..... 149  
To access a raw data variable.

## Class Description (RawDataVariables)

## Syntax

```
RawDataVariables = RawData.RawDataVariables
```

## Purpose

To handle the variables of the raw data.

## Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Long	To get the number of raw data variables.

## Methods

The class contains the following methods:

Method	Purpose
Item	To access a raw data variable. Refer to <a href="#">Item</a> on page 149.

**Related topics****References**

[Class Description \(RawData\).....](#) 146

## Item

**Class**

RawDataVariables

**Syntax**

```
RawDataVariable = RawDataVariables.Item(long Index)
```

**Purpose**

To access a raw data variable.

**Parameters**

The method uses the following parameters:

Parameter	Type	Description
Index	long	The index of the raw data variable.

**Return value**

The method returns the following parameter:

Type	Description
RawDataVariable <sup>1)</sup>	The raw data variable.

<sup>1)</sup> Refer to [RawDataVariable](#) on page 149.

**Related topics****References**

[Class Description \(RawDataVariables\).....](#) 148

## RawDataVariable

**Purpose**

To get a raw data variable.

## Class Description (RawDataVariable)

### Syntax

```
RawDataVariable = RawDataVariables.RawDataVariable
```

### Purpose

To get a raw data variable.

### Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	string	To get the name of the raw data variable.
Unit	string	To get the unit of the raw data variable.
Values	double[]	To get the values of the raw data variable.

### Methods

—

### Related topics

#### References

[Class Description \(RawDataVariables\)..... 148](#)

## Enumerations for Processing

## Enumerations for Processing

### Introduction

You can use predefined constants in the tool automation.

### Enumerations

**EnumDataSource** The following constants are used to specify the data source for a measurement variable:

Value	Description
Default	Default value that is specified for the measurement type variable
Raw	Value of the raw data variable
Calculated	Value that is calculated by the previous evaluation

**EnumPlotContext** The following constants are used to specify the plotting context:

Value	Description
None = 0	None
Parameter = 1	Parameter
ParameterPage = 2	Parameter page
ParameterSet = 4	Parameter set

**EnumPlotType** The following constants are used to specify the plotting type:

Value	Description
None = 0	No plotting
Standard = 1	Standard plotting
Measurement = 2	Plotting of measurement
MeasurementLabel = 3	Plotting of measurement label

**MappingState** The following constants are used specify the mapping state:

Value	Description
Undefined = 0	Undefined
NotMapped = 1	Not mapped
MappedUsingExistingRawDataVariable = 2	Mapped using existing raw data variable
MappedUsingNotExistingRawDataVariable = 3	Mapped using not existing raw data variable
MappedUsingExistingRawDataVariable WithoutConversion = 4	Mapped using existing raw data variable without conversion

## Related topics

## References

[Overview of the Object Model for Processing..... 92](#)





**A**

ActiveMeasurementData class 95  
 ActiveMeasurementTypes class 101  
 Add Measurement Type Variable dialog 60  
 additional function  
   properties 77  
 AdditionalFunction class 102  
 AdditionalFunctions class 104  
 automating  
   processing 88

**C**

Common Program Data folder 10  
 ConversionFile class 108  
 ConversionRule class 109  
 ConversionRules class 111

**D**

Documents folder 10

**L**

Local Program Data folder 10

**M**

mapping variables 21, 25  
 MappingConfiguration class 113  
 MappingVariable class 116  
 measurement data 17  
   properties 78  
 Measurement Data pane 62  
 Measurement Data Variable Mapping Import  
 Conflicts dialog 65  
 measurement function 17  
 measurement type 17  
   properties 80  
   specifying 18  
 Measurement Type Container page 68  
 Measurement Type pane 66  
 Measurement Types page 68  
 MeasurementConfiguration class 118  
 MeasurementFunction class 120  
 MeasurementType class 122  
 MeasurementTypeConfiguration class 126  
 MeasurementTypeContainer class 130  
 MeasurementTypeVariable class 133  
 MeasurementTypeVariables class 134

**O**

object model overview  
   processing 92

**P**

parameter  
   properties 80  
 parameter page  
   properties 82

ParameterProcessingInfo class 140  
 ParameterRecordProcessingInfo class 142  
 ParameterSetProcessing class 144  
 PlotConfiguration class 138  
 processing  
   basics 12  
   workflow 14  
 processing automation 88  
 Processing Configuration pane 33, 70  
 Processing page 69  
 ProcessingConfiguration class 145  
 properties  
   additional function 77  
   measurement data 78  
   measurement type 80  
   parameter 80  
   parameter page 82

**R**

raw data 16  
 RawData class 146  
 RawDataVariable class 149  
 RawDataVariables class 148  
 Remove command 56

**S**

Specify Raw Data – Measurement Data Source  
 dialog 76  
 specifying  
   measurement type 18  
 starting  
   processing 40

**W**

workflow  
   processing 14

