



QUADERA Developer Reference

Help information for VSTA development

Table of Contents

Class Library Reference	1
AnalogInput Class	2
AnalogInput Members	2
AnalogInput Properties	3
Application Property	3
ApplyCharacteristic Property	3
Index Property	3
IOType Property	4
Name Property	4
Parent Property	4
Unit Property	4
Value Property	4
AnalogInputChangedEventArgs Class	5
AnalogInputChangedEventArgs Members	5
AnalogInputChangedEventArgs Properties	5
AnalogInput Property	5
AnalogInputs Class	6
AnalogInputs Members	6
AnalogInputs Methods	6
this Indexer	7
this Indexer (String)	7
this Indexer (int)	7
AnalogInputs Properties	7
Application Property	8
Count Property	8
HardwareInstalled Property	8
Parent Property	8
AnalogOutput Class	9
AnalogOutput Members	9
AnalogOutput Properties	9
Application Property	10
Index Property	10
IOType Property	10
Name Property	10
Parent Property	10
Value Property	11
AnalogOutputChangedEventArgs Class	11

AnalogOutputChangedEventArgs Members	11
AnalogOutputChangedEventArgs Properties	11
AnalogOutput Property	12
AnalogOutputs Class	12
AnalogOutputs Members	12
AnalogOutputs Methods	13
this Indexer	13
this Indexer (String)	13
this Indexer (int)	13
AnalogOutputs Properties	14
Application Property	14
Count Property	14
HardwareInstalled Property	14
Parent Property	14
Application Class	15
Application Members	15
Application Methods	16
CompressAllDevices Method	16
CompressDevice Method	16
CompressDevice Method (Device)	16
CompressDevice Method (string)	17
Delay Method	17
SendMessage Method	17
Application Properties	18
AutoApplyCalibrations Property	18
Devices Property	18
MeasureProjects Property	19
Parent Property	19
PopupMessageManager Property	19
UnitsConverter Property	19
Version Property	19
WorkspaceHeight Property	20
WorkspaceWidth Property	20
Device Class	20
Device Members	20
Device Methods	23
ClearMSAMasses Method	25
ClearTotalPressure Method	25
Connect Method	25
Disconnect Method	25
DoCalibrationGSC Method	25

DoCalibrationMSC Method	26
DoCalibrationOffset Method	27
DoCalibrationZeroGas Method	27
DoGetActiveFilament Method	28
DoGetIonSourceActiveParameterSet Method	28
DoGetIonSourceV Method	28
DoGetIonSourceV1 Method	28
DoGetIonSourceV2 Method	29
DoGetIonSourceV3 Method	29
DoGetIonSourceV4 Method	29
DoGetIonSourceV5 Method	29
DoGetIonSourceVoltage Method	30
DoMSA Method	30
DoSetActiveFilament Method	31
DoSetEmissionCurrent Method	31
DoSetIonSourceActiveParameterSet Method	31
DoSetIonSourceV Method	32
DoSetIonSourceV1 Method	32
DoSetIonSourceV2 Method	32
DoSetIonSourceV3 Method	33
DoSetIonSourceV4 Method	33
DoSetIonSourceV5 Method	33
DoSetIonSourceVoltage Method	34
GetCommonSemVoltage Method	34
GetEPIOffsetValueRange05 Method	34
GetEPIOffsetValueRange06 Method	34
GetEPIOffsetValueRange07 Method	35
GetEPIOffsetValueRange08 Method	35
GetEPIOffsetValueRange09 Method	35
GetEPIOffsetValueRange10 Method	35
GetEPIOffsetValueRange11 Method	36
GetEPIOffsetValueRange12 Method	36
GetFilamentState Method	36
GetFineTuneMasses Method	36
GetMSAMeasuredMasses Method	37
GetMSAStatus Method	37
GetOPCItem Method	37
GetSemState Method	37
GetTotalPressureState Method	37
GetTotalPressureValue Method	38
IO Method	38
MSAActualMasses Method	38

NewMeasureProject Method	39
NewMeasureProject Method (String)	39
NewMeasureProject Method (String, String)	39
ResetFineTuneActualMasses Method	40
SetCommonSemVoltage Method	40
SetFilamentState Method	40
SetFineTuneActualMasses Method	40
SetSemState Method	42
SetTotalPressure Method	42
SetTotalPressureState Method	42
Start Method	43
Start Method ()	43
Start Method (DeviceStartMode)	43
Stop Method	43
Device Properties	43
AnalogInputs Property	44
AnalogOutputs Property	44
Application Property	44
AttachedState Property	45
ConnectedState Property	45
DeviceIP Property	45
DeviceMacAddress Property	45
DevicePath Property	46
DigitalInputs Property	46
DigitalOutputs Property	46
IsSimulator Property	46
Name Property	47
Parent Property	47
PartialPressureAvailable Property	47
RunningState Property	47
Type Property	47
Device Events	47
AnalogInputChanged Event	48
AnalogOutputChanged Event	48
DeviceAttachedStateChanged Event	49
DeviceConnectionStateChanged Event	49
DeviceError Event	50
DeviceFilamentStateChanged Event	50
DeviceRecipeChanged Event	50
DeviceSemStateChanged Event	51
DeviceTPValueChanged Event	51
DigitalInputChanged Event	52

DigitalOutputChanged Event	52
RunningStateChanged Event	52
DeviceError Class	53
DeviceError Members	53
DeviceError Properties	54
Code Property	54
Description Property	54
Type Property	54
DeviceErrorEventArgs Class	54
DeviceErrorEventArgs Members	55
DeviceErrorEventArgs Properties	55
DeviceError Property	55
Devices Class	55
Devices Members	56
Devices Methods	56
this Indexer	56
this Indexer (int)	56
this Indexer (string)	57
Devices Properties	57
Application Property	57
Count Property	57
Parent Property	58
DeviceAttachedState Enumeration	58
DeviceAttachedStateChangedEventArgs Class	58
DeviceAttachedStateChangedEventArgs Members	59
DeviceAttachedStateChangedEventArgs Properties	59
AttachedState Property	59
DeviceConnectedState Enumeration	59
DeviceConnectionStateChangedEventArgs Class	60
DeviceConnectionStateChangedEventArgs Members	60
DeviceConnectionStateChangedEventArgs Properties	61
ConnectedState Property	61
DeviceErrorType Enumeration	61
DeviceFilamentState Enumeration	62
DeviceFilamentStateChangedEventArgs Class	62
DeviceFilamentStateChangedEventArgs Members	62
DeviceFilamentStateChangedEventArgs Properties	63
FilamentState Property	63

DeviceRunningState Enumeration	63
DeviceSemState Enumeration	64
DeviceSemStateChangedEventArgs Class	64
DeviceSemStateChangedEventArgs Members	64
DeviceSemStateChangedEventArgs Properties	65
SemState Property	65
DeviceStartMode Enumeration	65
DeviceTotalPressureState Enumeration	66
DeviceTPValueChangedEventArgs Class	66
DeviceTPValueChangedEventArgs Members	67
DeviceTPValueChangedEventArgs Properties	67
Value Property	67
DeviceType Enumeration	67
DigitalInput Class	68
DigitalInput Members	68
DigitalInput Properties	68
Application Property	69
Index Property	69
IOType Property	69
Name Property	69
Parent Property	69
Value Property	70
DigitalInputChangedEventArgs Class	70
DigitalInputChangedEventArgs Members	70
DigitalInputChangedEventArgs Properties	71
DigitalInput Property	71
DigitalInputs Class	71
DigitalInputs Members	71
DigitalInputs Methods	72
this Indexer	72
this Indexer (String)	72
this Indexer (int)	72
DigitalInputs Properties	73
Application Property	73
Count Property	73
HardwareInstalled Property	73
Parent Property	74
DigitalOutput Class	74

DigitalOutput Members	74
DigitalOutput Properties	74
Application Property	75
Index Property	75
IOType Property	75
Name Property	75
Parent Property	76
Value Property	76
DigitalOutputChangedEventArgs Class	76
DigitalOutputChangedEventArgs Members	76
DigitalOutputChangedEventArgs Properties	77
DigitalOutput Property	77
DigitalOutputs Class	77
DigitalOutputs Members	77
DigitalOutputs Methods	78
this Indexer	78
this Indexer (String)	78
this Indexer (int)	78
DigitalOutputs Properties	79
Application Property	79
Count Property	79
HardwareInstalled Property	79
Parent Property	80
Values Property	80
IO Class	80
IO Members	80
IO Properties	81
AnalogInputs Property	81
AnalogOutputs Property	81
DigitalInputs Property	82
DigitalOutputs Property	82
IsInstalled Property	82
Type Property	82
IOType Enumeration	82
MeasureProject Class	83
MeasureProject Members	83
MeasureProject Methods	84
Activate Method	84
Backup Method	84
Close Method	85

Export Method	85
Export Method (String)	85
Export Method (String, int, int)	85
SetBounds Method	86
MeasureProject Properties	86
Application Property	87
Device Property	87
Height Property	87
Left Property	87
MeasurementTasks Property	87
MeasureProjectRecipes Property	88
Name Property	88
Parent Property	88
Top Property	88
Type Property	88
Width Property	89
MeasureProjects Class	89
MeasureProjects Members	89
MeasureProjects Methods	90
Open Method	90
this Indexer	90
this Indexer (int)	90
this Indexer (string)	91
MeasureProjects Properties	91
Application Property	91
Count Property	91
Parent Property	92
MeasureProjectType Enumeration	92
MeasurementTask Class	92
MeasurementTask Members	93
MeasurementTask Methods	93
GetCycleData Method	93
GetLastMeasureValue Method	94
GetMeasureTaskStatistics Method	94
GetMeasureTaskStatistics Method (double, double, double, double)	95
GetMeasureTaskStatistics Method (double, double, double, double, DateTime, DateTime)	95
GetMeasureTaskStatistics Method (double, double, double, double, DateTime, DateTime, int)	96
GetMeasureTaskStatistics Method (double, double, double, double, int)	96
MeasurementTask Properties	97
Application Property	97
ApplyBackgroundSubtraction Property	97

























ApplyCharacteristic Property	98
Cycles Property	98
Index Property	98
Name Property	98
Parent Property	98
TaskID Property	99
TaskType Property	99
TaskUnit Property	99
UsePartialPressure Property	99
MeasurementTasks Class	99
MeasurementTasks Members	100
MeasurementTasks Methods	100
this Indexer	100
MeasurementTasks Properties	101
Application Property	101
Count Property	101
Parent Property	101
RunningStateChangedEventArgs Class	101
RunningStateChangedEventArgs Members	102
RunningStateChangedEventArgs Properties	102
RunningState Property	102



Index

a












1 Symbol Reference

Classes

	Name	Description
	AnalogInput (see page 2)	Represents a single analog input controlled by the device.
	AnalogInputChangedEventArgs (see page 5)	Provides data for the AnalogInputChanged event.
	AnalogInputs (see page 6)	Represents the collection of AnalogInput (see page 2) objects controlled by the device. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	AnalogOutput (see page 9)	Represents a single analog output controlled by the device.
	AnalogOutputChangedEventArgs (see page 11)	Provides data for the AnalogOutputChanged event.
	AnalogOutputs (see page 12)	Represents the collection of AnalogOutput (see page 9) objects controlled by the device. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	Application (see page 15)	Represents the QUADERA application.
	Device (see page 20)	Represents a hardware device such as the QMG220 or QMG700.
	DeviceError (see page 53)	Represents error or warning information from the device.
	DeviceErrorEventArgs (see page 54)	Provides data for the DeviceError (see page 55) event.
	Devices (see page 55)	Represents the devices which are currently available in QUADERA.
	DeviceAttachedStateChangedEventArgs (see page 58)	Provides data for the DeviceAttachedStateChanged event.
	DeviceConnectionStateChangedEventArgs (see page 60)	Provides data for the DeviceConnectionStateChanged event.
	DeviceFilamentStateChangedEventArgs (see page 62)	Provides data for the DeviceFilamentStateChanged event.
	DeviceSemStateChangedEventArgs (see page 64)	Provides data for the DeviceSemStateChanged event.
	DeviceTPValueChangedEventArgs (see page 66)	Provides data for the DeviceSemStateChanged event.
	DigitalInput (see page 68)	Represents a single digital input controlled by the device.
	DigitalInputChangedEventArgs (see page 70)	Provides data for the DigitalInputChanged event.
	DigitalInputs (see page 71)	Represents the collection of DigitalInput (see page 68) objects controlled by the device. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	DigitalOutput (see page 74)	Represents a single digital output controlled by the device.
	DigitalOutputChangedEventArgs (see page 76)	Provides data for the DigitalOutputChanged event.
	DigitalOutputs (see page 77)	Represents the collection of DigitalOutput (see page 74) objects controlled by the device. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	IO (see page 80)	Represents an IO component such as the IO700 or IO220.
	MeasureProject (see page 83)	Represents a measurement project.
	MeasureProjects (see page 89)	Represents the collection of open measurement projects in QUADERA.
	MeasurementTask (see page 92)	Represents a measurement task in a measurement project.

	MeasurementTasks (see page 99)	Represents the collection of MeasurementTask (see page 92) objects controlled by the device.
	RunningStateChangedEventArgs (see page 101)	Provides data for the RunningStateChanged event.

Enumerations

	Name	Description
	DeviceAttachedState (see page 58)	Represents the device attached state.
	DeviceConnectedState (see page 59)	Represents the connected state of the device.
	DeviceErrorType (see page 61)	Represents the device error type.
	DeviceFilamentState (see page 62)	Represents the filament state.
	DeviceRunningState (see page 63)	Represents the running state of the device.
	DeviceSemState (see page 64)	Represents the SEM state.
	DeviceStartMode (see page 65)	Specifies the measurement mode of the device.
	DeviceTotalPressureState (see page 66)	Represents the total pressure state.
	DeviceType (see page 67)	Represents the device type.
	IOType (see page 82)	Represents the type of the IO (see page 80) component.
	MeasureProjectType (see page 92)	Represents the type of a measurement project.

Remarks

QUADERA Developer Reference - Version 4.50 (May 2012)

1.1 AnalogInput Class

Represents a single analog input controlled by the device.

File

AnalogInput.cs

C#

```
public class AnalogInput;
```

Visual Basic

```
Public Class AnalogInput
```





Inheritance Hierarchy





AnalogInput

1.1.1 AnalogInput Members

The following tables list the members exposed by AnalogInput.

Public Properties









	Name	Description
	Application (see page 3)	Gets the QUADERA Application object.
	ApplyCharacteristic (see page 3)	Used to have characteristics applied to applicable Tasks.
	Index (see page 3)	Gets the index of this analog input.
	IOType (see page 4)	Gets the IO (see page 80) component type which this analog input belongs to.

	Name (see page 4)	Gets the name of this analog input.
	Parent (see page 4)	Gets the parent AnalogInputs (see page 6) collection object.
	Unit (see page 4)	Gets a string value representing the unit for the Analog Input
	Value (see page 4)	Gets a boolean value representing the state of a analog input.

1.1.2 AnalogInput Properties

The properties of the AnalogInput class are listed here.

Public Properties

	Name	Description
	Application (see page 3)	Gets the QUADERA Application object.
	ApplyCharacteristic (see page 3)	Used to have characteristics applied to applicable Tasks.
	Index (see page 3)	Gets the index of this analog input.
	IOType (see page 4)	Gets the IO (see page 80) component type which this analog input belongs to.
	Name (see page 4)	Gets the name of this analog input.
	Parent (see page 4)	Gets the parent AnalogInputs (see page 6) collection object.
	Unit (see page 4)	Gets a string value representing the unit for the Analog Input
	Value (see page 4)	Gets a boolean value representing the state of a analog input.

1.1.2.1 AnalogInput.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.1.2.2 AnalogInput.ApplyCharacteristic Property

Used to have characteristics applied to applicable Tasks.

C#

```
public bool ApplyCharacteristic;
```

Visual Basic

```
Public Property ApplyCharacteristic() As bool
```

1.1.2.3 AnalogInput.Index Property

Gets the index of this analog input.

C#

```
public int Index;
```

Visual Basic

```
Public Property Index() As Integer
```

Remarks

The index of the first analog input is 1.

1.1.2.4 AnalogInput.IOType Property

Gets the IO (see page 80) component type which this analog input belongs to.

C#

```
public IOType IOType;
```

Visual Basic

```
Public Property IOType() As IOType
```

1.1.2.5 AnalogInput.Name Property

Gets the name of this analog input.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

1.1.2.6 AnalogInput.Parent Property

Gets the parent AnalogInputs (see page 6) collection object.

C#

```
public AnalogInputs Parent;
```

Visual Basic

```
Public Property Parent() As AnalogInputs
```

1.1.2.7 AnalogInput.Unit Property

Gets a string value representing the unit for the Analog Input

C#

```
public String Unit;
```

Visual Basic

```
Public Property Unit() As String
```

1.1.2.8 AnalogInput.Value Property

Gets a boolean value representing the state of a analog input.

C#

```
public float Value;
```

Visual Basic

```
Public Property Value() As float
```

Remarks

This property can only be used when the device is connected.

Example

The following example code displays a message depending on the state of the first analog input.

```
Sub ReadAnalogInputState()
    MsgBox "AnalogInput #1 is:" + Me.Device.AnalogInputs(1).Value.ToString()
End Sub
```

1.2 AnalogInputChangedEventArgs Class

Provides data for the AnalogInputChanged event.

File

AnalogInputChangedEventArgs.cs

C#

```
public class AnalogInputChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class AnalogInputChangedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

AnalogInputChangedEventArgs

1.2.1 AnalogInputChangedEventArgs Members

The following tables list the members exposed by AnalogInputChangedEventArgs.

Public Properties

	Name	Description
	AnalogInput (? see page 5)	Gets the AnalogInput object which has changed.

1.2.2 AnalogInputChangedEventArgs Properties

The properties of the AnalogInputChangedEventArgs class are listed here.

Public Properties

	Name	Description
	AnalogInput (? see page 5)	Gets the AnalogInput object which has changed.

1.2.2.1 AnalogInputChangedEventArgs.AnalogInput Property

Gets the AnalogInput object which has changed.

C#

```
public AnalogInput AnalogInput;
```

Visual Basic

```
Public Property AnalogInput() As AnalogInput
```

1.3 AnalogInputs Class

Represents the collection of AnalogInput (see page 2) objects controlled by the device. If Device (see page 20) is not connected or hardware is not installed, the count will be zero

File

AnalogInputs.cs

C#

```
public class AnalogInputs : IEnumerable;
```

Visual Basic

```
Public Class AnalogInputs
    Inherits IEnumerable
```

Inheritance Hierarchy

AnalogInputs

1.3.1 AnalogInputs Members

The following tables list the members exposed by AnalogInputs.

Public Methods

	Name	Description
	this (see page 7)	This is the overview for the this indexer overload.

Public Properties

	Name	Description
	Application (see page 8)	Gets the QUADERA Application object.
	Count (see page 8)	Gets the number of AnalogInput (see page 2) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 8)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 8)	Gets the parent Device (see page 20) object.

1.3.2 AnalogInputs Methods

The methods of the AnalogInputs class are listed here.



Public Methods

	Name	Description
	this (see page 7)	This is the overview for the this indexer overload.

1.3.2.1 this Indexer

This is the overview for the this indexer overload.

Overload List

	Name	Description
	AnalogInputs.this (int) (see page 7)	Gets the AnalogInput (see page 2) object at the specified index in the collection.
	AnalogInputs.this (String) (see page 7)	Gets the AnalogInput (see page 2) object with the specified name in the collection.

1.3.2.1.1 AnalogInputs.this Indexer (String)

Gets the AnalogInput ([see page 2](#)) object with the specified name in the collection.

C#

```
public AnalogInput this[
    String name
];
```

Visual Basic

```
Public Property Item(name as String) as AnalogInput
```

Parameters

Parameters	Description
name	The name of the analog input.

1.3.2.1.2 AnalogInputs.this Indexer (int)

Gets the AnalogInput ([see page 2](#)) object at the specified index in the collection.

C#

```
public AnalogInput this[
    int index
];
```

Visual Basic

```
Public Property Item(index as Integer) as AnalogInput
```

Parameters

Parameters	Description
index	The index of the analog input.





Remarks

The index of the first analog input is 1.

1.3.3 AnalogInputs Properties

The properties of the AnalogInputs class are listed here.

Public Properties

	Name	Description
	Application (see page 8)	Gets the QUADERA Application object.
	Count (see page 8)	Gets the number of AnalogInput (see page 2) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 8)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 8)	Gets the parent Device (see page 20) object.

1.3.3.1 AnalogInputs.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.3.3.2 AnalogInputs.Count Property

Gets the number of AnalogInput ([see page 2](#)) objects in the collection. If Device ([see page 20](#)) is not connected or hardware is not installed, the count will be zero

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.3.3.3 AnalogInputs.HardwareInstalled Property

Indicates whether IO ([see page 80](#)) hardware is installed

C#

```
public bool HardwareInstalled;
```

Visual Basic

```
Public Property HardwareInstalled() As bool
```

1.3.3.4 AnalogInputs.Parent Property

Gets the parent Device ([see page 20](#)) object.

C#

```
public Device Parent;
```

Visual Basic

```
Public Property Parent() As Device
```

1.4 AnalogOutput Class

Represents a single analog output controlled by the device.

File

AnalogOutput.cs

C#

```
public class AnalogOutput;
```

Visual Basic

```
Public Class AnalogOutput
```







Inheritance Hierarchy

AnalogOutput

1.4.1 AnalogOutput Members

The following tables list the members exposed by AnalogOutput.







Public Properties

	Name	Description
	Application (see page 10)	Gets the QUADERA Application object.
	Index (see page 10)	Gets the index of this analog output.
	IOType (see page 10)	Gets the IO (see page 80) component type which this analog output belongs to.
	Name (see page 10)	Gets the name of this analog output.
	Parent (see page 10)	Gets the parent AnalogOutputs (see page 12) collection object.
	Value (see page 11)	Gets or sets a float value representing the state of a analog output.

1.4.2 AnalogOutput Properties

The properties of the AnalogOutput class are listed here.

Public Properties

	Name	Description
	Application (see page 10)	Gets the QUADERA Application object.
	Index (see page 10)	Gets the index of this analog output.
	IOType (see page 10)	Gets the IO (see page 80) component type which this analog output belongs to.
	Name (see page 10)	Gets the name of this analog output.
	Parent (see page 10)	Gets the parent AnalogOutputs (see page 12) collection object.
	Value (see page 11)	Gets or sets a float value representing the state of a analog output.

1.4.2.1 AnalogOutput.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.4.2.2 AnalogOutput.Index Property

Gets the index of this analog output.

C#

```
public int Index;
```

Visual Basic

```
Public Property Index() As Integer
```

Remarks

The index of the first analog output is 1.

1.4.2.3 AnalogOutput.IOType Property

Gets the IO ([see page 80](#)) component type which this analog output belongs to.

C#

```
public IOType IOType;
```

Visual Basic

```
Public Property IOType() As IOType
```

1.4.2.4 AnalogOutput.Name Property

Gets the name of this analog output.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

1.4.2.5 AnalogOutput.Parent Property

Gets the parent AnalogOutputs ([see page 12](#)) collection object.

C#

```
public AnalogOutputs Parent;
```

Visual Basic

```
Public Property Parent() As AnalogOutputs
```

1.4.2.6 AnalogOutput.Value Property

Gets or sets a float value representing the state of a analog output.

C#

```
public float Value;
```

Visual Basic

```
Public Property Value() As float
```

Remarks

This property can only be used when the device is connected.

Example

The following example code sets the state of the first analog output to on.

```
Sub SetAnalogOutputState()  
    Me.Device.AnalogOutputs(1).Value = value  
End Sub
```

1.5 AnalogOutputChangedEventArgs Class

Provides data for the AnalogOutputChanged event.

File

AnalogOutputChangedEventArgs.cs

C#

```
public class AnalogOutputChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class AnalogOutputChangedEventArgs  
    Inherits EventArgs
```

Inheritance Hierarchy

AnalogOutputChangedEventArgs

1.5.1 AnalogOutputChangedEventArgs Members

The following tables list the members exposed by AnalogOutputChangedEventArgs.


Public Properties

	Name	Description
	AnalogOutput (see page 12)	Gets the AnalogOutput object which has changed.

1.5.2 AnalogOutputChangedEventArgs Properties

The properties of the AnalogOutputChangedEventArgs class are listed here.

Public Properties

	Name	Description
	AnalogOutput (see page 12)	Gets the AnalogOutput object which has changed.

1.5.2.1 AnalogOutputChangedEventArgs.AnalogOutput Property

Gets the AnalogOutput object which has changed.

C#

```
public AnalogOutput AnalogOutput;
```

Visual Basic

```
Public Property AnalogOutput() As AnalogOutput
```

1.6 AnalogOutputs Class

Represents the collection of AnalogOutput ([see page 9](#)) objects controlled by the device. If Device ([see page 20](#)) is not connected or hardware is not installed, the count will be zero

File

AnalogOutputs.cs

C#

```
public class AnalogOutputs : IEnumerable;
```

Visual Basic

```
Public Class AnalogOutputs
    Inherits IEnumerable
```


Inheritance Hierarchy

AnalogOutputs





1.6.1 AnalogOutputs Members

The following tables list the members exposed by AnalogOutputs.

Public Methods

	Name	Description
	this (see page 13)	This is the overview for the this indexer overload.


Public Properties

	Name	Description
	Application (see page 14)	Gets the QUADERA Application object.
	Count (see page 14)	Gets the number of AnalogOutput (see page 9) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 14)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 14)	Gets the parent Device (see page 20) object.

1.6.2 AnalogOutputs Methods

The methods of the AnalogOutputs class are listed here.



Public Methods

	Name	Description
	this (see page 13)	This is the overview for the this indexer overload.

1.6.2.1 this Indexer

This is the overview for the this indexer overload.

Overload List

	Name	Description
	AnalogOutputs.this (int) (see page 13)	Gets the AnalogOutput (see page 9) object at the specified index in the collection.
	AnalogOutputs.this (String) (see page 13)	Gets the AnalogOutput (see page 9) object with the specified name in the collection.

1.6.2.1.1 AnalogOutputs.this Indexer (String)

Gets the AnalogOutput ([see page 9](#)) object with the specified name in the collection.

C#

```
public AnalogOutput this[
    String name
];
```

Visual Basic

```
Public Property Item(name as String) as AnalogOutput
```

Parameters

Parameters	Description
name	The name of the analog output.

1.6.2.1.2 AnalogOutputs.this Indexer (int)

Gets the AnalogOutput ([see page 9](#)) object at the specified index in the collection.

C#

```
public AnalogOutput this[
    int index
];
```

Visual Basic

```
Public Property Item(index as Integer) as AnalogOutput
```

Parameters

Parameters	Description
index	The index of the analog output.





Remarks

The index of the first analog output is 1.

1.6.3 AnalogOutputs Properties

The properties of the AnalogOutputs class are listed here.

Public Properties

	Name	Description
	Application (see page 14)	Gets the QUADERA Application object.
	Count (see page 14)	Gets the number of AnalogOutput (see page 9) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 14)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 14)	Gets the parent Device (see page 20) object.

1.6.3.1 AnalogOutputs.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.6.3.2 AnalogOutputs.Count Property

Gets the number of AnalogOutput ([see page 9](#)) objects in the collection. If Device ([see page 20](#)) is not connected or hardware is not installed, the count will be zero

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.6.3.3 AnalogOutputs.HardwareInstalled Property

Indicates whether IO ([see page 80](#)) hardware is installed

C#

```
public bool HardwareInstalled;
```

Visual Basic

```
Public Property HardwareInstalled() As bool
```

1.6.3.4 AnalogOutputs.Parent Property

Gets the parent Device ([see page 20](#)) object.

C#

```
public Device Parent;
```

Visual Basic

```
Public Property Parent() As Device
```

1.7 Application Class

Represents the QUADERA application.

File

Application.cs

C#

```
public class Application;
```

Visual Basic

```
Public Class Application
```





Inheritance Hierarchy

Application










1.7.1 Application Members

The following tables list the members exposed by Application.

Public Methods

	Name	Description
	CompressAllDevices (see page 16)	Compress all Device (see page 20) files.
	CompressDevice (see page 16)	This is the overview for the CompressDevice method overload.
	Delay (see page 17)	Waits for the specified amount of time.
	SendMessage (see page 17)	Sends a message to the All Device (see page 20) Status View for a given Device (see page 20)





Public Properties

	Name	Description
	AutoApplyCalibrations (see page 18)	Global parameter used to force calibration templates to apply when finished running.
	Devices (see page 18)	Gets a reference to the Devices object containing all devices that are currently available in QUADERA.
	MeasureProjects (see page 19)	Gets a reference to the MeasureProjects object containing all measurement projects that are currently open in QUADERA.
	Parent (see page 19)	Gets a reference to the parent object for this class.
	PopupMessageManager (see page 19)	This is PopupMessageManager, a member of class Application.
	UnitsConverter (see page 19)	This is UnitsConverter, a member of class Application.
	Version (see page 19)	Gets the QUADERA version number.
	WorkspaceHeight (see page 20)	Gets the height of the workspace area in the main QUADERA window.
	WorkspaceWidth (see page 20)	Gets the width of the workspace area in the main QUADERA window.

1.7.2 Application Methods

The methods of the Application class are listed here.

Public Methods

	Name	Description
	CompressAllDevices (see page 16)	Compress all Device (see page 20) files.
	CompressDevice (see page 16)	This is the overview for the CompressDevice method overload.
	Delay (see page 17)	Waits for the specified amount of time.
	SendMessage (see page 17)	Sends a message to the All Device (see page 20) Status View for a given Device (see page 20)

1.7.2.1 Application.CompressAllDevices Method

Compress all Device ([see page 20](#)) files.

C#

```
public bool CompressAllDevices();
```

Visual Basic

```
Public Function CompressAllDevices() As bool
```



Returns

True if successful

1.7.2.2 CompressDevice Method

This is the overview for the CompressDevice method overload.

Overload List

	Name	Description
	Application.CompressDevice (Device) (see page 16)	Compress a Device (see page 20) file using Device (see page 20) object.
	Application.CompressDevice (string) (see page 17)	Compress a Device (see page 20) file using path.

1.7.2.2.1 Application.CompressDevice Method (Device)

Compress a Device ([see page 20](#)) file using Device ([see page 20](#)) object.

C#

```
public bool CompressDevice(
    Device dev
);
```

Visual Basic

```
Public Function CompressDevice(
    dev As Device
) As bool
```

Returns

True if successful

1.7.2.2 Application.CompressDevice Method (string)

Compress a Device (see page 20) file using path.

C#

```
public bool CompressDevice(
    string devicePath
);
```

Visual Basic

```
Public Function CompressDevice(
    devicePath As string
) As bool
```

Returns

True if successful

1.7.2.3 Application.Delay Method

Waits for the specified amount of time.

C#

```
public void Delay(
    int numberOfMilliseconds
);
```

Visual Basic

```
Public Function Delay(
    numberOfMilliseconds As Integer
) As void
```

Parameters

Parameters	Description
numberOfMilliseconds	Amount of time to wait.

Returns

No return value

Remarks

The delay time should be at least 10 milliseconds.

1.7.2.4 Application.SendDeviceMessage Method

Sends a message to the All Device (see page 20) Status View for a given Device (see page 20)

C#

```
public void SendDeviceMessage(
    Device device,
    string message
);
```

Visual Basic

```
Public Function SendDeviceMessage(
    device As Device,
    message As string
```

```
) As void
```

Parameters

Parameters	Description
device	The Device (see page 20)
message	The Message to be displayed on ADSV










Returns

No return value

1.7.3 Application Properties

The properties of the Application class are listed here.

Public Properties

	Name	Description
	AutoApplyCalibrations (see page 18)	Global parameter used to force calibration templates to apply when finished running.
	Devices (see page 18)	Gets a reference to the Devices object containing all devices that are currently available in QUADERA.
	MeasureProjects (see page 19)	Gets a reference to the MeasureProjects object containing all measurement projects that are currently open in QUADERA.
	Parent (see page 19)	Gets a reference to the parent object for this class.
	PopupMessageManager (see page 19)	This is PopupMessageManager, a member of class Application.
	UnitsConverter (see page 19)	This is UnitsConverter, a member of class Application.
	Version (see page 19)	Gets the QUADERA version number.
	WorkspaceHeight (see page 20)	Gets the height of the workspace area in the main QUADERA window.
	WorkspaceWidth (see page 20)	Gets the width of the workspace area in the main QUADERA window.

1.7.3.1 Application.AutoApplyCalibrations Property

Global parameter used to force calibration templates to apply when finished running.

C#

```
public bool AutoApplyCalibrations;
```

Visual Basic

```
Public Property AutoApplyCalibrations() As bool
```

Returns

No return value

1.7.3.2 Application.Devices Property

Gets a reference to the Devices object containing all devices that are currently available in QUADERA.

C#

```
public Devices Devices;
```

Visual Basic

```
Public Property Devices() As Devices
```

1.7.3.3 Application.MeasureProjects Property

Gets a reference to the MeasureProjects object containing all measurement projects that are currently open in QUADERA.

C#

```
public MeasureProjects MeasureProjects;
```

Visual Basic

```
Public Property MeasureProjects() As MeasureProjects
```

1.7.3.4 Application.Parent Property

Gets a reference to the parent object for this class.

C#

```
public Application Parent;
```

Visual Basic

```
Public Property Parent() As Application
```

1.7.3.5 Application.PopupMessageManager Property

C#

```
public PopupManager PopupMessageManager;
```

Visual Basic

```
Public Property PopupMessageManager() As PopupManager
```

Remarks

This is PopupMessageManager, a member of class Application.

1.7.3.6 Application.UnitsConverter Property

C#

```
public UnitsConverter UnitsConverter;
```

Visual Basic

```
Public Property UnitsConverter() As UnitsConverter
```

Remarks

This is UnitsConverter, a member of class Application.

1.7.3.7 Application.Version Property

Gets the QUADERA version number.

C#

```
public String Version;
```

Visual Basic

```
Public Property Version() As String
```

1.7.3.8 Application.WorkspaceHeight Property

Gets the height of the workspace area in the main QUADERA window.

C#

```
public int WorkspaceHeight;
```

Visual Basic

```
Public Property WorkspaceHeight() As Integer
```

1.7.3.9 Application.WorkspaceWidth Property

Gets the width of the workspace area in the main QUADERA window.

C#

```
public int WorkspaceWidth;
```

Visual Basic

```
Public Property WorkspaceWidth() As Integer
```

1.8 Device Class

Represents a hardware device such as the QMG220 or QMG700.

File

Device.cs

C#

```
public class Device;
```

Visual Basic

```
Public Class Device
```

Remarks

With this class the user can connect to and disconnect from a device and also start and stop a measurement. It also provides the object model for digital inputs and outputs.




Inheritance Hierarchy










Device

1.8.1 Device Members

















The following tables list the members exposed by Device.

Public Events

	Name	Description
	AnalogInputChanged (↗ see page 48)	Occurs when an Analog input changes state.
	AnalogOutputChanged (↗ see page 48)	Occurs when an Analog output changes state.
	DeviceAttachedStateChanged (↗ see page 49)	Occurs when a Device (↗ see page 20)'s attached state changes.





	DeviceConnectionStateChanged (see page 49)	Occurs when a Device (see page 20)'s connected state changes.
	DeviceError (see page 50)	Occurs when an error or warning is received from the device.
	DeviceFilamentStateChanged (see page 50)	Occurs when a Device (see page 20)'s attached filament state changes.
	DeviceRecipeChanged (see page 50)	Occurs when a Recipe changes.
	DeviceSemStateChanged (see page 51)	Occurs when a Device (see page 20)'s Sem's state changes.
	DeviceTPValueChanged (see page 51)	Occurs when a Device (see page 20)'s Total Pressure Value changes.
	DigitalInputChanged (see page 52)	Occurs when a digital input changes state.
	DigitalOutputChanged (see page 52)	Occurs when a digital output changes state.
	RunningStateChanged (see page 52)	Occurs when the running state of the device changes.













Public Methods

	Name	Description
	ClearMSAMasses (see page 25)	This is ClearMSAMasses, a member of class Device.
	ClearTotalPressure (see page 25)	Clears hard coded Total Pressure.
	Connect (see page 25)	Connects to the device.
	Disconnect (see page 25)	Disconnects from the device.
	DoCalibrationGSC (see page 25)	Creates a new measurement project based on the specified template.
	DoCalibrationMSC (see page 26)	Creates a new measurement project based on the specified template.
	DoCalibrationOffset (see page 27)	Creates a new measurement project based on the specified template.
	DoCalibrationZeroGas (see page 27)	Creates a new measurement project based on the specified template.
	DoGetActiveFilament (see page 28)	Gets the active filament.
	DoGetIonSourceActiveParameterSet (see page 28)	
	DoGetIonSourceV (see page 28)	Gets the Ion Source V(indx).
	DoGetIonSourceV1 (see page 28)	Gets the Ion Source V1 or Ion Reference voltage
	DoGetIonSourceV2 (see page 29)	Gets the Ion Source V2 or Ion Cathode voltage
	DoGetIonSourceV3 (see page 29)	Gets the Ion Source V3 or Ion Focus voltage
	DoGetIonSourceV4 (see page 29)	Gets the Ion Source V4 or Ion Field Axis voltage
	DoGetIonSourceV5 (see page 29)	Gets the Ion Source V5 or Ion Extract voltage
	DoGetIonSourceVoltage (see page 30)	Gets the Ion Source Voltage by Index. Valid indexes are 1-5 for QMG220 and 1-9 for QMG 700 returns true if index is invalid.
	DoMSA (see page 30)	Initializes Mass Scale Adjust.
	DoSetActiveFilament (see page 31)	Sets the active filament
	DoSetEmissionCurrent (see page 31)	Sets the active filament
	DoSetIonSourceActiveParameterSet (see page 31)	Sets the index of the Ion Source parameter set
	DoSetIonSourceV (see page 32)	Sets the Ion Source V(indx)
	DoSetIonSourceV1 (see page 32)	Sets the Ion Source V5 or Ion Reference voltage. Change in steps of 0.01 V. 105.00...150.00 Default 150
	DoSetIonSourceV2 (see page 32)	Sets the Ion Source V5 or Ion Cathode voltage. Change in steps of 0.01 V. -100.00...0.00 Default -70
	DoSetIonSourceV3 (see page 33)	Sets the Ion Source V5 or Ion Focus voltage. Change in steps of 0.01 V. -30.00...0.00 Default -5

	DoSetIonSourceV4 (see page 33)	Sets the Ion Source V5 or Ion Field Axis voltage. Change in steps of 0.01 V. -15.00...0.00 Default -10
	DoSetIonSourceV5 (see page 33)	Sets the Ion Source V5 or Ion Extract voltage. Change in steps of 0.01 V. -150.00...0.00 Default -50
	DoSetIonSourceVoltage (see page 34)	Sets the Ion Source Voltage by Index. Valid indexes are 1-5 for QMG220 and 1-9 for QMG 700 returns true if index is invalid.
	GetCommonSemVoltage (see page 34)	Gets the common SEM voltage.
	GetEPIOffsetValueRange05 (see page 34)	Gets the Offset Value for Range E-5.
	GetEPIOffsetValueRange06 (see page 34)	Gets the Offset Value for Range E-6.
	GetEPIOffsetValueRange07 (see page 35)	Gets the Offset Value for Range E-7.
	GetEPIOffsetValueRange08 (see page 35)	Gets the Offset Value for Range E-8.
	GetEPIOffsetValueRange09 (see page 35)	Gets the Offset Value for Range E-9.
	GetEPIOffsetValueRange10 (see page 35)	Gets the Offset Value for Range E-10.
	GetEPIOffsetValueRange11 (see page 36)	Gets the Offset Value for Range E-11.
	GetEPIOffsetValueRange12 (see page 36)	Gets the Offset Value for Range E-12.
	GetFilamentState (see page 36)	Gets the filament state.
	GetFineTuneMasses (see page 36)	Method retrieves Fine Tuning Reference Masses and Actual Masses. Returns the number of masses in the arrays
	GetMSAMeasuredMasses (see page 37)	Property returns an array of the actual measured and adjusted masses
	GetMSAStatus (see page 37)	Property returns an array of the Mass Scale Adjust Status Bytes.
	GetOPCItem (see page 37)	Returns a value as read from OPC Namespace. Path's must be defined in Device (see page 20) definition. If path not found False is returned.
	GetSemState (see page 37)	Gets the SEM state.
	GetTotalPressureState (see page 37)	Gets the total pressure state.
	GetTotalPressureValue (see page 38)	Gets the total pressure value.
	IO (see page 38)	Gets the IO object for the given component which contains the external input / output values for this device.
	MSAActualMasses (see page 38)	Property returns current MSA Adjusted mass and its channel.
	NewMeasureProject (see page 39)	Creates a new measurement project based on the specified template.
	ResetFineTuneActualMasses (see page 40)	Method sets Actual masses to Reference masses.
	SetCommonSemVoltage (see page 40)	Sets the common SEM voltage.
	SetFilamentState (see page 40)	Sets the filament state.
	SetFineTuneActualMasses (see page 40)	Method sets the masses used for Mass Scale Adjust tuning.
	SetSemState (see page 42)	Sets the SEM state.
	SetTotalPressure (see page 42)	Sets a fixed total pressure override.
	SetTotalPressureState (see page 42)	Sets the total pressure state.
	Start (see page 43)	Starts measurement with the device.
	Stop (see page 43)	Stops measurement with the device.

Public Properties






















	Name	Description
	AnalogInputs (see page 44)	Gets the AnalogInputs collection object which contains the analog inputs for this device.
	AnalogOutputs (see page 44)	Gets the AnalogOutputs collection object which contains the analog outputs for this device.
	Application (see page 44)	Gets the QUADERA Application object.
	AttachedState (see page 45)	Gets the attached state of the device.

	ConnectedState (see page 45)	Gets the connected state of the device.
	DeviceIP (see page 45)	Property returns the IP of the device.
	DeviceMacAddress (see page 45)	Property returns the MAC of the device.
	DevicePath (see page 46)	This is DevicePath, a member of class Device.
	DigitalInputs (see page 46)	Gets the DigitalInputs collection object which contains the digital inputs for this device.
	DigitalOutputs (see page 46)	Gets the DigitalOutputs collection object which contains the digital outputs for this device.
	IsSimulator (see page 46)	Property returned indicates that a device is a Simulator.
	Name (see page 47)	Gets the name of the device.
	Parent (see page 47)	Gets the parent MeasureProject (see page 83) object.
	PartialPressureAvailable (see page 47)	Gets state for availability of Pressure.
	RunningState (see page 47)	Gets the running state of the device.
	Type (see page 47)	Gets the device type.

1.8.2 Device Methods

The methods of the Device class are listed here.

Public Methods

	Name	Description
	ClearMSAMasses (see page 25)	This is ClearMSAMasses, a member of class Device.
	ClearTotalPressure (see page 25)	Clears hard coded Total Pressure.
	Connect (see page 25)	Connects to the device.
	Disconnect (see page 25)	Disconnects from the device.
	DoCalibrationGSC (see page 25)	Creates a new measurement project based on the specified template.
	DoCalibrationMSC (see page 26)	Creates a new measurement project based on the specified template.
	DoCalibrationOffset (see page 27)	Creates a new measurement project based on the specified template.
	DoCalibrationZeroGas (see page 27)	Creates a new measurement project based on the specified template.
	DoGetActiveFilament (see page 28)	Gets the active filament.
	DoGetIonSourceActiveParameterSet (see page 28)	
	DoGetIonSourceV (see page 28)	Gets the Ion Source V(indx).
	DoGetIonSourceV1 (see page 28)	Gets the Ion Source V1 or Ion Reference voltage
	DoGetIonSourceV2 (see page 29)	Gets the Ion Source V2 or Ion Cathode voltage
	DoGetIonSourceV3 (see page 29)	Gets the Ion Source V3 or Ion Focus voltage
	DoGetIonSourceV4 (see page 29)	Gets the Ion Source V4 or Ion Field Axis voltage
	DoGetIonSourceV5 (see page 29)	Gets the Ion Source V5 or Ion Extract voltage
	DoGetIonSourceVoltage (see page 30)	Gets the Ion Source Voltage by Index. Valid indexes are 1-5 for QMG220 and 1-9 for QMG 700 returns true if index is invalid.
	DoMSA (see page 30)	Initializes Mass Scale Adjust.
	DoSetActiveFilament (see page 31)	Sets the active filament
	DoSetEmissionCurrent (see page 31)	Sets the active filament
	DoSetIonSourceActiveParameterSet (see page 31)	Sets the index of the Ion Source parameter set

◆	DoSetIonSourceV (see page 32)	Sets the Ion Source V(indx)
◆	DoSetIonSourceV1 (see page 32)	Sets the Ion Source V5 or Ion Reference voltage. Change in steps of 0.01 V. 105.00...150.00 Default 150
◆	DoSetIonSourceV2 (see page 32)	Sets the Ion Source V5 or Ion Cathode voltage. Change in steps of 0.01 V. -100.00...0.00 Default -70
◆	DoSetIonSourceV3 (see page 33)	Sets the Ion Source V5 or Ion Focus voltage. Change in steps of 0.01 V. -30.00...0.00 Default -5
◆	DoSetIonSourceV4 (see page 33)	Sets the Ion Source V5 or Ion Field Axis voltage. Change in steps of 0.01 V. -15.00...0.00 Default -10
◆	DoSetIonSourceV5 (see page 33)	Sets the Ion Source V5 or Ion Extract voltage. Change in steps of 0.01 V. -150.00...0.00 Default -50
◆	DoSetIonSourceVoltage (see page 34)	Sets the Ion Source Voltage by Index. Valid indexes are 1-5 for QMG220 and 1-9 for QMG 700 returns true if index is invalid.
◆	GetCommonSemVoltage (see page 34)	Gets the common SEM voltage.
◆	GetEPIOffsetValueRange05 (see page 34)	Gets the Offset Value for Range E-5.
◆	GetEPIOffsetValueRange06 (see page 34)	Gets the Offset Value for Range E-6.
◆	GetEPIOffsetValueRange07 (see page 35)	Gets the Offset Value for Range E-7.
◆	GetEPIOffsetValueRange08 (see page 35)	Gets the Offset Value for Range E-8.
◆	GetEPIOffsetValueRange09 (see page 35)	Gets the Offset Value for Range E-9.
◆	GetEPIOffsetValueRange10 (see page 35)	Gets the Offset Value for Range E-10.
◆	GetEPIOffsetValueRange11 (see page 36)	Gets the Offset Value for Range E-11.
◆	GetEPIOffsetValueRange12 (see page 36)	Gets the Offset Value for Range E-12.
◆	GetFilamentState (see page 36)	Gets the filament state.
◆	GetFineTuneMasses (see page 36)	Method retrieves Fine Tuning Reference Masses and Actual Masses. Returns the number of masses in the arrays
◆	GetMSAMeasuredMasses (see page 37)	Property returns an array of the actual measured and adjusted masses
◆	GetMSAStatus (see page 37)	Property returns an array of the Mass Scale Adjust Status Bytes.
◆	GetOPCItem (see page 37)	Returns a value as read from OPC Namespace. Path's must be defined in Device (see page 20) definition. If path not found False is returned.
◆	GetSemState (see page 37)	Gets the SEM state.
◆	GetTotalPressureState (see page 37)	Gets the total pressure state.
◆	GetTotalPressureValue (see page 38)	Gets the total pressure value.
◆	IO (see page 38)	Gets the IO object for the given component which contains the external input / output values for this device.
◆	MSAActualMasses (see page 38)	Property returns current MSA Adjusted mass and its channel.
◆	NewMeasureProject (see page 39)	Creates a new measurement project based on the specified template.
◆	ResetFineTuneActualMasses (see page 40)	Method sets Actual masses to Reference masses.
◆	SetCommonSemVoltage (see page 40)	Sets the common SEM voltage.
◆	SetFilamentState (see page 40)	Sets the filament state.
◆	SetFineTuneActualMasses (see page 40)	Method sets the masses used for Mass Scale Adjust tuning.
◆	SetSemState (see page 42)	Sets the SEM state.
◆	SetTotalPressure (see page 42)	Sets a fixed total pressure override.
◆	SetTotalPressureState (see page 42)	Sets the total pressure state.
◆	Start (see page 43)	Starts measurement with the device.
◆	Stop (see page 43)	Stops measurement with the device.

1.8.2.1 Device.ClearMSAMasses Method

C#

```
public void ClearMSAMasses();
```

Visual Basic

```
Public Function ClearMSAMasses() As void
```

Remarks

This is ClearMSAMasses, a member of class Device.

1.8.2.2 Device.ClearTotalPressure Method

Clears hard coded Total Pressure.

C#

```
public void ClearTotalPressure();
```

Visual Basic

```
Public Function ClearTotalPressure() As void
```

1.8.2.3 Device.Connect Method

Connects to the device.

C#

```
public void Connect();
```

Visual Basic

```
Public Function Connect() As void
```

Remarks

Connect fails with an error if the measurement project was not created from a template. The Type (see page 47) property can first be checked as shown in the following example code:

```
Sub Connect()  
    If Me.Type = MeasureProjectType.Template Then  
        Me.Device.Connect()  
    End If  
End Sub
```

1.8.2.4 Device.Disconnect Method

Disconnects from the device.

C#

```
public void Disconnect();
```

Visual Basic

```
Public Function Disconnect() As void
```

1.8.2.5 Device.DoCalibrationGSC Method

Creates a new measurement project based on the specified template.

C#

```
public MeasureProject DoCalibrationGSC(
    string templateName,
    string saveMPAs
);
```

Visual Basic

```
Public Function DoCalibrationGSC(
    templateName As string,
    saveMPAs As string
) As MeasureProject
```

Parameters

Parameters	Description
templateName	The Custom filename of the template that is used to create the new measurement project.
saveMPAs	The Custom filename of the new measurement project as it will be saved.

Remarks

The filenames can be absolute or relative. If the filenames are relative then the following directories will be used:

- templateName - the standard template directory for the current device.
- saveName - the standard data directory.

1.8.2.6 Device.DoCalibrationMSC Method

Creates a new measurement project based on the specified template.

C#

```
public MeasureProject DoCalibrationMSC(
    string templateName,
    string saveMPAs,
    bool Coarse
);
```

Visual Basic

```
Public Function DoCalibrationMSC(
    templateName As string,
    saveMPAs As string,
    Coarse As bool
) As MeasureProject
```

Parameters

Parameters	Description
templateName	The Custom filename of the template that is used to create the new measurement project.
saveMPAs	The Custom filename of the new measurement project as it will be saved.
Coarse	Do a coarse calibration?.

Remarks

The filenames can be absolute or relative. If the filenames are relative then the following directories will be used:

- templateName - the standard template directory for the current device.
- saveName - the standard data directory.

1.8.2.7 Device.DoCalibrationOffset Method

Creates a new measurement project based on the specified template.

C#

```
public MeasureProject DoCalibrationOffset(
    string templateName,
    string saveMPAs
);
```

Visual Basic

```
Public Function DoCalibrationOffset(
    templateName As string,
    saveMPAs As string
) As MeasureProject
```

Parameters

Parameters	Description
templateName	The Custom filename of the template that is used to create the new measurement project.
saveMPAs	The Custom filename of the new measurement project as it will be saved.

Remarks

The filenames can be absolute or relative. If the filenames are relative then the following directories will be used:

- templateName - the standard template directory for the current device.
- saveName - the standard data directory.

1.8.2.8 Device.DoCalibrationZeroGas Method

Creates a new measurement project based on the specified template.

C#

```
public MeasureProject DoCalibrationZeroGas(
    string templateName,
    string saveMPAs
);
```

Visual Basic

```
Public Function DoCalibrationZeroGas(
    templateName As string,
    saveMPAs As string
) As MeasureProject
```

Parameters

Parameters	Description
templateName	The Custom filename of the template that is used to create the new measurement project.
saveMPAs	The Custom filename of the new measurement project as it will be saved.

Remarks

The filenames can be absolute or relative. If the filenames are relative then the following directories will be used:

- templateName - the standard template directory for the current device.
- saveName - the standard data directory.

1.8.2.9 Device.DoGetActiveFilament Method

Gets the active filament.

C#

```
public int DoGetActiveFilament();
```

Visual Basic

```
Public Function DoGetActiveFilament() As Integer
```

Returns

The Index

1.8.2.10 Device.DoGetIonSourceActiveParameterSet Method

C#

```
public int DoGetIonSourceActiveParameterSet();
```

Visual Basic

```
Public Function DoGetIonSourceActiveParameterSet() As Integer
```

Returns

The parameter set

Description

Gets the Ion Source Active parameter set

1.8.2.11 Device.DoGetIonSourceV Method

Gets the Ion Source V(indx).

C#

```
public Single DoGetIonSourceV(
    int indx
);
```

Visual Basic

```
Public Function DoGetIonSourceV(
    indx As Integer
) As Single
```

Parameters

Parameters	Description
indx	Index of Voltage To set.

Returns

The Voltage

1.8.2.12 Device.DoGetIonSourceV1 Method

Gets the Ion Source V1 or Ion Reference voltage

C#

```
public Single DoGetIonSourceV1();
```

Visual Basic

```
Public Function DoGetIonSourceV1() As Single
```

Returns

The Ion Reference voltage

1.8.2.13 Device.DoGetIonSourceV2 Method

Gets the Ion Source V2 or Ion Cathode voltage

C#

```
public Single DoGetIonSourceV2();
```

Visual Basic

```
Public Function DoGetIonSourceV2() As Single
```

Returns

The Ion Cathode voltage

1.8.2.14 Device.DoGetIonSourceV3 Method

Gets the Ion Source V3 or Ion Focus voltage

C#

```
public Single DoGetIonSourceV3();
```

Visual Basic

```
Public Function DoGetIonSourceV3() As Single
```

Returns

The Ion Focus voltage

1.8.2.15 Device.DoGetIonSourceV4 Method

Gets the Ion Source V4 or Ion Field Axis voltage

C#

```
public Single DoGetIonSourceV4();
```

Visual Basic

```
Public Function DoGetIonSourceV4() As Single
```

Returns

The Ion Field Axis voltage

1.8.2.16 Device.DoGetIonSourceV5 Method

Gets the Ion Source V5 or Ion Extract voltage

C#

```
public Single DoGetIonSourceV5();
```

Visual Basic

```
Public Function DoGetIonSourceV5() As Single
```

Returns

The Ion Extract voltage

1.8.2.17 Device.DoGetIonSourceVoltage Method

Gets the Ion Source Voltage by Index. Valid indexes are 1-5 for QMG220 and 1-9 for QMG 700 returns true if index is invalid.

C#

```
public bool DoGetIonSourceVoltage(
    int index,
    ref Single value
);
```

Visual Basic

```
Public Function DoGetIonSourceVoltage(
    index As Integer,
    ByRef value As Single
) As bool
```

Parameters

Parameters	Description
index	The index.
value	The voltage.

Returns

The Ion Source voltage

1.8.2.18 Device.DoMSA Method

Initializes Mass Scale Adjust.

C#

```
public void DoMSA(
    long recipeID,
    bool fine,
    int cycles,
    int dwell
);
```

Visual Basic

```
Public Function DoMSA(
    recipeID As long,
    fine As bool,
    cycles As Integer,
    dwell As Integer
) As void
```

Parameters

Parameters	Description
recipeID	recipeID (future)
fine	fine - fine or coarse tune
cycles	cycles - Number of cycles to run
dwell	dwell (future)

1.8.2.19 Device.DoSetActiveFilament Method

Sets the active filament

C#

```
public void DoSetActiveFilament(
    int indx
);
```

Visual Basic

```
Public Function DoSetActiveFilament(
    indx As Integer
) As void
```

Parameters

Parameters	Description
indx	Index of filament. Returns

1.8.2.20 Device.DoSetEmissionCurrent Method

Sets the active filament

C#

```
public void DoSetEmissionCurrent(
    Single current
);
```

Visual Basic

```
Public Function DoSetEmissionCurrent(
    current As Single
) As void
```

Parameters

Parameters	Description
current	Index of filament. Returns

1.8.2.21 Device.DoSetIonSourceActiveParameterSet Method

Sets the index of the Ion Source parameter set

C#

```
public void DoSetIonSourceActiveParameterSet(
    int newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceActiveParameterSet(
    newValue As Integer
) As void
```

Parameters

Parameters	Description
newValue	The new parameter set.

1.8.2.22 Device.DoSetIonSourceV Method

Sets the Ion Source V(indx)

C#

```
public void DoSetIonSourceV(
    int indx,
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceV(
    indx As Integer,
    newValue As Single
) As void
```

Parameters

Parameters	Description
indx	Index of Voltage To set.
newValue	The new voltage.

1.8.2.23 Device.DoSetIonSourceV1 Method

Sets the Ion Source V5 or Ion Reference voltage. Change in steps of 0.01 V. 105.00...150.00 Default 150

C#

```
public void DoSetIonSourceV1(
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceV1(
    newValue As Single
) As void
```

Parameters

Parameters	Description
newValue	The new voltage.

1.8.2.24 Device.DoSetIonSourceV2 Method

Sets the Ion Source V5 or Ion Cathode voltage. Change in steps of 0.01 V. -100.00...0.00 Default -70

C#

```
public void DoSetIonSourceV2(
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceV2(
    newValue As Single
) As void
```

Parameters

Parameters	Description
newValue	The new voltage.

1.8.2.25 Device.DoSetIonSourceV3 Method

Sets the Ion Source V5 or Ion Focus voltage. Change in steps of 0.01 V. -30.00...0.00 Default -5

C#

```
public void DoSetIonSourceV3(
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceV3(
    newValue As Single
) As void
```

Parameters

Parameters	Description
newValue	The new voltage.

1.8.2.26 Device.DoSetIonSourceV4 Method

Sets the Ion Source V5 or Ion Field Axis voltage. Change in steps of 0.01 V. -15.00...0.00 Default -10

C#

```
public void DoSetIonSourceV4(
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceV4(
    newValue As Single
) As void
```

Parameters

Parameters	Description
newValue	The new voltage.

1.8.2.27 Device.DoSetIonSourceV5 Method

Sets the Ion Source V5 or Ion Extract voltage. Change in steps of 0.01 V. -150.00...0.00 Default -50

C#

```
public void DoSetIonSourceV5(
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceV5(
    newValue As Single
) As void
```

Parameters

Parameters	Description
newValue	The new voltage.

1.8.2.28 Device.DoSetIonSourceVoltage Method

Sets the Ion Source Voltage by Index. Valid indexes are 1-5 for QMG220 and 1-9 for QMG 700 returns true if index is invalid.

C#

```
public bool DoSetIonSourceVoltage(
    int index,
    Single newValue
);
```

Visual Basic

```
Public Function DoSetIonSourceVoltage(
    index As Integer,
    newValue As Single
) As bool
```

Parameters

Parameters	Description
index	The index.
newValue	The new voltage.

1.8.2.29 Device.GetCommonSemVoltage Method

Gets the common SEM voltage.

C#

```
public double GetCommonSemVoltage();
```

Visual Basic

```
Public Function GetCommonSemVoltage() As double
```

1.8.2.30 Device.GetEPIOffsetValueRange05 Method

Gets the Offset Value for Range E-5.

C#

```
public double GetEPIOffsetValueRange05();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange05() As double
```

Returns

The value

1.8.2.31 Device.GetEPIOffsetValueRange06 Method

Gets the Offset Value for Range E-6.

C#

```
public double GetEPIOffsetValueRange06();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange06() As double
```

Returns

The value

1.8.2.32 Device.GetEPIOffsetValueRange07 Method

Gets the Offset Value for Range E-7.

C#

```
public double GetEPIOffsetValueRange07();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange07() As double
```

Returns

The value

1.8.2.33 Device.GetEPIOffsetValueRange08 Method

Gets the Offset Value for Range E-8.

C#

```
public double GetEPIOffsetValueRange08();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange08() As double
```

Returns

The value

1.8.2.34 Device.GetEPIOffsetValueRange09 Method

Gets the Offset Value for Range E-9.

C#

```
public double GetEPIOffsetValueRange09();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange09() As double
```

Returns

The value

1.8.2.35 Device.GetEPIOffsetValueRange10 Method

Gets the Offset Value for Range E-10.

C#

```
public double GetEPIOffsetValueRange10();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange10() As double
```

Returns

The value

1.8.2.36 Device.GetEPIOffsetValueRange11 Method

Gets the Offset Value for Range E-11.

C#

```
public double GetEPIOffsetValueRange11();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange11() As double
```

Returns

The value

1.8.2.37 Device.GetEPIOffsetValueRange12 Method

Gets the Offset Value for Range E-12.

C#

```
public double GetEPIOffsetValueRange12();
```

Visual Basic

```
Public Function GetEPIOffsetValueRange12() As double
```

Returns

The value

1.8.2.38 Device.GetFilamentState Method

Gets the filament state.

C#

```
public DeviceFilamentState GetFilamentState();
```

Visual Basic

```
Public Function GetFilamentState() As DeviceFilamentState
```

1.8.2.39 Device.GetFineTuneMasses Method

Method retrieves Fine Tuning Reference Masses and Actual Masses. Returns the number of masses in the arrays

C#

```
public int GetFineTuneMasses(
    ref Single[] refMasses,
    ref Single[] actMasses
);
```

Visual Basic

```
Public Function GetFineTuneMasses(
    ByRef refMasses As Single[],
    ByRef actMasses As Single[]
) As Integer
```

Parameters

Parameters	Description
refMasses	refMasses - Reference masses in amu
actMasses	actMasses - Actual Adjusted masses in amu

1.8.2.40 Device.GetMSAMeasuredMasses Method

Property returns an array of the actual measured and adjusted masses

C#

```
public Single[] GetMSAMeasuredMasses();
```

Visual Basic

```
Public Function GetMSAMeasuredMasses() As Single()
```

1.8.2.41 Device.GetMSAStatus Method

Property returns an array of the Mass Scale Adjust Status Bytes.

C#

```
public UInt16[] GetMSAStatus();
```

Visual Basic

```
Public Function GetMSAStatus() As UInt16()
```

1.8.2.42 Device.GetOPCItem Method

Returns a value as read from OPC Namespace. Path's must be defined in Device (see page 20) definition. If path not found False is returned.

C#

```
public bool GetOPCItem(
    string path,
    ref object obj
);
```

Visual Basic

```
Public Function GetOPCItem(
    path As string,
    ByRef obj As object
) As bool
```

Parameters

Parameters	Description
path	path - The OPC Namespace Path to the OPC Item
obj	obj - the returned value. Must be cast to appropriate type

1.8.2.43 Device.GetSemState Method

Gets the SEM state.

C#

```
public DeviceSemState GetSemState();
```

Visual Basic

```
Public Function GetSemState() As DeviceSemState
```

1.8.2.44 Device.GetTotalPressureState Method

Gets the total pressure state.

C#

```
public DeviceTotalPressureState GetTotalPressureState();
```

Visual Basic

```
Public Function GetTotalPressureState() As DeviceTotalPressureState
```

1.8.2.45 Device.GetTotalPressureValue Method

Gets the total pressure value.

C#

```
public float GetTotalPressureValue();
```

Visual Basic

```
Public Function GetTotalPressureValue() As float
```

Parameters

Parameters	Description
%PAR0%	fPressureValue float

1.8.2.46 Device.IO Method

Gets the IO object for the given component which contains the external input / output values for this device.

C#

```
public IO IO(
    IOType ioType
);
```

Visual Basic

```
Public Function IO(
    ioType As IOType
) As IO
```

Parameters

Parameters	Description
ioType	IO component type.

1.8.2.47 Device.MSAActualMasses Method

Property returns current MSA Adjusted mass and its channel.

C#

```
public void MSAActualMasses(
    ref Single mass,
    ref int channel
);
```

Visual Basic

```
Public Function MSAActualMasses(
    ByRef mass As Single,
    ByRef channel As Integer
) As void
```

Parameters



Parameters	Description
mass	mass

channel	channel
---------	---------

1.8.2.48 NewMeasureProject Method

Creates a new measurement project based on the specified template.

Overload List

	Name	Description
	Device.NewMeasureProject (String) (🔗 see page 39)	Creates a new measurement project based on the specified template.
	Device.NewMeasureProject (String, String) (🔗 see page 39)	Creates a new measurement project based on the specified template.

1.8.2.48.1 Device.NewMeasureProject Method (String)

Creates a new measurement project based on the specified template.

C#

```
public MeasureProject NewMeasureProject(
    String templateName
);
```

Visual Basic

```
Public Function NewMeasureProject(
    templateName As String
) As MeasureProject
```

Parameters

Parameters	Description
templateName	The filename of the template that is used to create the new measurement project.

Remarks

With this overloaded method the measurement project is temporary and will not be saved when closed.

1.8.2.48.2 Device.NewMeasureProject Method (String, String)

Creates a new measurement project based on the specified template.

C#

```
public MeasureProject NewMeasureProject(
    String templateName,
    String fileName
);
```

Visual Basic

```
Public Function NewMeasureProject(
    templateName As String,
    fileName As String
) As MeasureProject
```

Parameters

Parameters	Description
templateName	The filename of the template that is used to create the new measurement project.
fileName	The filename of the new measurement project.

Remarks

The filenames can be absolute or relative. If the filenames are relative then the following directories will be used:

- templateName - the standard template directory for the current device.
- saveName - the standard data directory.

1.8.2.49 Device.ResetFineTuneActualMasses Method

Method sets Actual masses to Reference masses.

C#

```
public void ResetFineTuneActualMasses();
```

Visual Basic

```
Public Function ResetFineTuneActualMasses() As void
```

1.8.2.50 Device.SetCommonSemVoltage Method

Sets the common SEM voltage.

C#

```
public void SetCommonSemVoltage(
    double semVoltage
);
```

Visual Basic

```
Public Function SetCommonSemVoltage(
    semVoltage As double
) As void
```

Parameters

Parameters	Description
semVoltage	Voltage value

1.8.2.51 Device.SetFilamentState Method

Sets the filament state.

C#

```
public void SetFilamentState(
    bool filamentState
);
```

Visual Basic

```
Public Function SetFilamentState(
    filamentState As bool
) As void
```

Parameters

Parameters	Description
filamentState	Boolean (True to switch on, false to switch off)

1.8.2.52 Device.SetFineTuneActualMasses Method

Method sets the masses used for Mass Scale Adjust tuning.

C#

```
public void SetFineTuneActualMasses(
    Single[] refMasses,
    Single[] actualMasses,
    int count
);
```

Visual Basic

```
Public Function SetFineTuneActualMasses(
    refMasses As Single[],
    actualMasses As Single[],
    count As Integer
) As void
```

Parameters

Parameters	Description
refMasses	masses

Example

this example demonstrates how to manipulate the MSA data read, set and clear it

```
'From loading a MID template this is a VSTA MeasurementProject type applet
'Call mp.Device.DoMSA(0, False, 1, 0) to set up MSA course mode
'then connect and Run
```

```
AddHandler mp.Device.RunningStateChanged, AddressOf OnRunningStateChanged
```

```
mp.Device.DoMSA(0, False, 1, 0)
mp.Device.Connect()
mp.Device.Start()
```

```
Private Sub OnRunningStateChanged(ByVal sender As Object, ByVal args As
RunningStateChangedEventArgs)
```

```
    If args.RunningState = DeviceRunningState.Stopped Then
```

```
        'Get array of newly adjusted masses and status associated with each
        Dim masses As Single() = mp.Device.GetMSAMeasuredMasses()
        Dim status As UShort() = mp.Device.GetMSAStatus()
        Dim x As Int16 = 0
```

```
        'to Clear Existing Adjusted Masses
        mp.Device.ClearMSAMasses()
```

```
        Dim SetRefMasses As Single() = Nothing
        Dim SetActualMasses As Single() = Nothing
```

```
        'Get previously saved adjusted masses and their corresponding Reference masses
        Dim count As Int16 = mp.Device.GetFineTuneMasses(SetRefMasses, SetActualMasses)
        'Read the recipe to decide how many tasks, and order they are in.
        'this is used to compare array of masses already saved from previous MSD to
        'Masses adjusted this time
        Dim tsk As RecipeTaskMassMID = mp.MeasureProjectRecipes(0).Tasks(0)
```

```
        'find and assign new adjusted mass to old one
        For y As Int16 = 0 To SetRefMasses.Length
            Dim mass As Single = SetRefMasses(y)
            If mass = tsk.Mass Then
                SetActualMasses(0) = masses(0)
                SetRefMasses(0) = tsk.Mass
            End If
        Next
```

```
        'Set adjusted masses so that they will be used by QMG
        mp.Device.SetFineTuneActualMasses(SetRefMasses, SetActualMasses, 1)
```

```
    End If
End Sub
```

1.8.2.53 Device.SetSemState Method

Sets the SEM state.

C#

```
public void SetSemState(
    bool semState
);
```

Visual Basic

```
Public Function SetSemState(
    semState As bool
) As void
```

Parameters

Parameters	Description
semState	Boolean (True to switch on, false to switch off)

1.8.2.54 Device.SetTotalPressure Method

Sets a fixed total pressure override.

C#

```
public void SetTotalPressure(
    double newPressure
);
```

Visual Basic

```
Public Function SetTotalPressure(
    newPressure As double
) As void
```

Parameters

Parameters	Description
newPressure	newPressure double

1.8.2.55 Device.SetTotalPressureState Method

Sets the total pressure state.

C#

```
public void SetTotalPressureState(
    bool totalPressureState
);
```

Visual Basic

```
Public Function SetTotalPressureState(
    totalPressureState As bool
) As void
```



Parameters

Parameters	Description
totalPressureState	Boolean (True to switch on, false to switch off)

1.8.2.56 Start Method

Starts measurement with the device.

Overload List

	Name	Description
	Device.Start () (see page 43)	Starts measurement with the device.
	Device.Start (DeviceStartMode) (see page 43)	Starts measurement with the device using the specified mode.

1.8.2.56.1 Device.Start Method ()

Starts measurement with the device.

C#

```
public void Start();
```

Visual Basic

```
Public Function Start() As void
```

1.8.2.56.2 Device.Start Method (DeviceStartMode)

Starts measurement with the device using the specified mode.

C#

```
public void Start(
    DeviceStartMode startMode
);
```

Visual Basic

```
Public Function Start(
    startMode As DeviceStartMode
) As void
```

Parameters

Parameters	Description
startMode	Measurement mode of the device (standard or single cycle).

1.8.2.57 Device.Stop Method

Stops measurement with the device.

C#

```
public void Stop();
```

















Visual Basic

```
Public Function Stop() As void
```

1.8.3 Device Properties

The properties of the Device class are listed here.

Public Properties

	Name	Description
	AnalogInputs (see page 44)	Gets the AnalogInputs collection object which contains the analog inputs for this device.
	AnalogOutputs (see page 44)	Gets the AnalogOutputs collection object which contains the analog outputs for this device.
	Application (see page 44)	Gets the QUADERA Application object.
	AttachedState (see page 45)	Gets the attached state of the device.
	ConnectedState (see page 45)	Gets the connected state of the device.
	DeviceIP (see page 45)	Property returns the IP of the device.
	DeviceMacAddress (see page 45)	Property returns the MAC of the device.
	DevicePath (see page 46)	This is DevicePath, a member of class Device.
	DigitalInputs (see page 46)	Gets the DigitalInputs collection object which contains the digital inputs for this device.
	DigitalOutputs (see page 46)	Gets the DigitalOutputs collection object which contains the digital outputs for this device.
	IsSimulator (see page 46)	Property returned indicates that a device is a Simulator.
	Name (see page 47)	Gets the name of the device.
	Parent (see page 47)	Gets the parent MeasureProject (see page 83) object.
	PartialPressureAvailable (see page 47)	Gets state for availability of Pressure.
	RunningState (see page 47)	Gets the running state of the device.
	Type (see page 47)	Gets the device type.

1.8.3.1 Device.AnalogInputs Property

Gets the AnalogInputs collection object which contains the analog inputs for this device.

C#

```
public AnalogInputs AnalogInputs;
```

Visual Basic

```
Public Property AnalogInputs() As AnalogInputs
```

Remarks

This method returns the analog inputs for either the first IO700 or IO220 component depending on the device type.

1.8.3.2 Device.AnalogOutputs Property

Gets the AnalogOutputs collection object which contains the analog outputs for this device.

C#

```
public AnalogOutputs AnalogOutputs;
```

Visual Basic

```
Public Property AnalogOutputs() As AnalogOutputs
```

Remarks

This method returns the analog outputs for either the first IO700 or IO220 component depending on the device type.

1.8.3.3 Device.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.8.3.4 Device.AttachedState Property

Gets the attached state of the device.

C#

```
public DeviceAttachedState AttachedState;
```

Visual Basic

```
Public Property AttachedState() As DeviceAttachedState
```

1.8.3.5 Device.ConnectedState Property

Gets the connected state of the device.

C#

```
public DeviceConnectedState ConnectedState;
```

Visual Basic

```
Public Property ConnectedState() As DeviceConnectedState
```

1.8.3.6 Device.DeviceIP Property

Property returns the IP of the device.

C#

```
public string DeviceIP;
```

Visual Basic

```
Public Property DeviceIP() As string
```

Example

The following example displays a message stating the Device (see page 20) IP.

```
Dim ip as String = device.DeviceIP  
MessageBox.Show("The Device IP is: " + ip)
```

1.8.3.7 Device.DeviceMacAddress Property

Property returns the MAC of the device.

C#

```
public string DeviceMacAddress;
```

Visual Basic

```
Public Property DeviceMacAddress() As string
```

Example

The following example displays a message stating the Device (see page 20) Machine Address.

```
Dim mac as String = device.DeviceMacAddress  
MessageBox.Show("The Device MAC is: " + mac)
```

1.8.3.8 Device.DevicePath Property

C#

```
public string DevicePath;
```

Visual Basic

```
Public Property DevicePath() As string
```

Remarks

This is DevicePath, a member of class Device.

1.8.3.9 Device.DigitalInputs Property

Gets the DigitalInputs collection object which contains the digital inputs for this device.

C#

```
public DigitalInputs DigitalInputs;
```

Visual Basic

```
Public Property DigitalInputs() As DigitalInputs
```

Remarks

This method returns the digital inputs for either the first IO700 or IO220 component depending on the device type.

1.8.3.10 Device.DigitalOutputs Property

Gets the DigitalOutputs collection object which contains the digital outputs for this device.

C#

```
public DigitalOutputs DigitalOutputs;
```

Visual Basic

```
Public Property DigitalOutputs() As DigitalOutputs
```

Remarks

This method returns the digital outputs for either the first IO700 or IO220 component depending on the device type.

1.8.3.11 Device.IsSimulator Property

Property returned indicates that a device is a Simulator.

C#

```
public bool IsSimulator;
```

Visual Basic

```
Public Property IsSimulator() As bool
```

Example

The following example displays a message indicating the presence of a simulator.

```
If device.IsSimulator Then
    MessageBox.Show("The Device is a simulator")
Else
    MessageBox.Show("The Device is NOT a simulator")
End If
```

1.8.3.12 Device.Name Property

Gets the name of the device.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

1.8.3.13 Device.Parent Property

Gets the parent MeasureProject (see page 83) object.

C#

```
public MeasureProject Parent;
```

Visual Basic

```
Public Property Parent() As MeasureProject
```

1.8.3.14 Device.PartialPressureAvailable Property

Gets state for availability of Pressure.

C#

```
public bool PartialPressureAvailable;
```

Visual Basic

```
Public Property PartialPressureAvailable() As bool
```

1.8.3.15 Device.RunningState Property

Gets the running state of the device.

C#

```
public DeviceRunningState RunningState;
```

Visual Basic

```
Public Property RunningState() As DeviceRunningState
```

1.8.3.16 Device.Type Property

Gets the device type.

C#

```
public DeviceType Type;
```













Visual Basic

```
Public Property Type() As DeviceType
```

1.8.4 Device Events

The events of the Device class are listed here.

Public Events

	Name	Description
	AnalogInputChanged (see page 48)	Occurs when an Analog input changes state.
	AnalogOutputChanged (see page 48)	Occurs when an Analog output changes state.
	DeviceAttachedStateChanged (see page 49)	Occurs when a Device (see page 20)'s attached state changes.
	DeviceConnectionStateChanged (see page 49)	Occurs when a Device (see page 20)'s connected state changes.
	DeviceError (see page 50)	Occurs when an error or warning is received from the device.
	DeviceFilamentStateChanged (see page 50)	Occurs when a Device (see page 20)'s attached filament state changes.
	DeviceRecipeChanged (see page 50)	Occurs when a Recipe changes.
	DeviceSemStateChanged (see page 51)	Occurs when a Device (see page 20)'s Sem's state changes.
	DeviceTPValueChanged (see page 51)	Occurs when a Device (see page 20)'s Total Pressure Value changes.
	DigitalInputChanged (see page 52)	Occurs when a digital input changes state.
	DigitalOutputChanged (see page 52)	Occurs when a digital output changes state.
	RunningStateChanged (see page 52)	Occurs when the running state of the device changes.

1.8.4.1 Device.AnalogInputChanged Event

Occurs when an Analog input changes state.

C#

```
public event AnalogInputChangedEventHandler AnalogInputChanged;
```

Visual Basic

```
Public Event AnalogInputChanged As AnalogInputChangedEventHandler
```

Example

To respond to AnalogInputChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles Me.Startup
    AddHandler Me.Device.AnalogInputChanged, AddressOf Device_AnalogInputChanged
End Sub
```

The following example displays a message when the value of the first Analog input changes.

```
Private Sub Device_AnalogInputChanged(ByVal sender As Object, ByVal args As
    AnalogInputChangedEventArgs)
    If args.AnalogInput.Index = 1 Then
        MessageBox.Show( "Analog Input 1 has changed value." )
    End If
End Sub
```

1.8.4.2 Device.AnalogOutputChanged Event

Occurs when an Analog output changes state.

C#

```
public event AnalogOutputChangedEventHandler AnalogOutputChanged;
```

Visual Basic

```
Public Event AnalogOutputChanged As AnalogOutputChangedEventHandler
```

Example

To respond to AnalogOutputChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.AnalogOutputChanged, AddressOf Device_AnalogOutputChanged
End Sub
```

The following example displays a message when the value of the first Analog output changes.

```
Private Sub Device_AnalogOutputChanged(ByVal sender As Object, ByVal args As
AnalogOutputChangedEventArgs)
    If args.AnalogOutput.Index = 1 Then
        MessageBox.Show( "Analog Output 1 has changed value." )
    End If
End Sub
```

1.8.4.3 Device.DeviceAttachedStateChanged Event

Occurs when a Device (see page 20)'s attached state changes.

C#

```
public event DeviceAttachedStateChangedEventHandler DeviceAttachedStateChanged;
```

Visual Basic

```
Public Event DeviceAttachedStateChanged As DeviceAttachedStateChangedEventHandler
```

Example

To respond to DeviceAttachedStateChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceAttachedStateChanged, AddressOf
Device_DeviceAttachedStateChanged
End Sub
```

The following example displays a message when the Device (see page 20) Attached State changes.

```
Private Sub Device_DeviceAttachedStateChanged(ByVal sender As Object, ByVal args As
DeviceAttachedStateChangedEventArgs)

    MessageBox.Show( "Device Attached State has changed." )

End Sub
```

1.8.4.4 Device.DeviceConnectionStateChanged Event

Occurs when a Device (see page 20)'s connected state changes.

C#

```
public event DeviceConnectionStateChangedEventHandler DeviceConnectionStateChanged;
```

Visual Basic

```
Public Event DeviceConnectionStateChanged As DeviceConnectionStateChangedEventHandler
```

Example

To respond to DeviceConnectionStateChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceConnectionStateChanged, AddressOf
Device_DeviceConnectionStateChanged
End Sub
```

The following example displays a message when the Device (see page 20) Connection State changes.

```
Private Sub Device_DeviceConnectionStateChanged(ByVal sender As Object, ByVal args As
DeviceConnectionStateChangedEventArgs)
    MessageBox.Show( "Device Connection State has changed value." )
End Sub
```

1.8.4.5 Device.DeviceError Event

Occurs when an error or warning is received from the device.

C#

```
public event DeviceErrorHandler DeviceError;
```

Visual Basic

```
Public Event DeviceError As DeviceErrorHandler
```

Example

To respond to DeviceError events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceError, AddressOf Device_DeviceError
End Sub
```

The following example displays a message when an error or warning occurs.

```
Private Sub Device_DeviceError(ByVal sender As Object, ByVal args As DeviceErrorHandlerEventArgs)
    MessageBox.Show( args.DeviceError.Description, "Device Error" )
End Sub
```

1.8.4.6 Device.DeviceFilamentStateChanged Event

Occurs when a Device (see page 20)'s attached filament state changes.

C#

```
public event DeviceFilamentStateChangedEventHandler DeviceFilamentStateChanged;
```

Visual Basic

```
Public Event DeviceFilamentStateChanged As DeviceFilamentStateChangedEventHandler
```

Example

To respond to filament events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceFilamentStateChanged, AddressOf
Device_DeviceFilamentStateChanged
End Sub
```

The following example displays a message when the value of the Filament state changes.

```
Private Sub Device_DeviceFilamentStateChanged(ByVal sender As Object, ByVal args As
DeviceFilamentStateChangedEventArgs)
    If args.DeviceFilamentState.On Then
        MessageBox.Show( "Filament is now On." )
    End If
End Sub
```

1.8.4.7 Device.DeviceRecipeChanged Event

Occurs when a Recipe changes.

C#

```
public event DeviceRecipeChangedEventHandler DeviceRecipeChanged;
```

Visual Basic

```
Public Event DeviceRecipeChanged As DeviceRecipeChangedEventHandler
```

Example

To respond to DeviceRecipeChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceRecipeChanged, AddressOf Device_DeviceRecipeChanged
End Sub
```

The following example displays a message when the Recipe changes.

```
Private Sub Device_DeviceRecipeChanged(ByVal sender As Object, ByVal args As
DeviceRecipeChangedEventArgs)
    MessageBox.Show( "The Recipe has changed" )
End Sub
```

1.8.4.8 Device.DeviceSemStateChanged Event

Occurs when a Device (see page 20)'s Sem's state changes.

C#

```
public event DeviceSemStateChangedEventHandler DeviceSemStateChanged;
```

Visual Basic

```
Public Event DeviceSemStateChanged As DeviceSemStateChangedEventHandler
```

Example

To respond to DeviceSemStateChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceSemStateChanged, AddressOf Device_DeviceSemStateChanged
End Sub
```

The following example displays a message when the Sem changes.

```
Private Sub Device_DeviceSemStateChanged(ByVal sender As Object, ByVal args As
DeviceSemStateChangedEventArgs)
    If args.State.value = DeviceSemState.On Then
        MessageBox.Show( "The Sem is now On" )
    End If
End Sub
```

1.8.4.9 Device.DeviceTPValueChanged Event

Occurs when a Device (see page 20)'s Total Pressure Value changes.

C#

```
public event DeviceTPValueChangedEventHandler DeviceTPValueChanged;
```

Visual Basic

```
Public Event DeviceTPValueChanged As DeviceTPValueChangedEventHandler
```

Example

To respond to DeviceTPValueChanged events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DeviceTPValueChanged, AddressOf Device_DeviceTPValueChanged
End Sub
```

The following example displays a message when the Total Pressure Value changes.

```
Private Sub Device_DeviceTPValueChanged(ByVal sender As Object, ByVal args As
```

```

DeviceTPValueChangedEventArgs)
    MessageBox.Show( "Total Pressure Value changed." )
End Sub

```

1.8.4.10 Device.DigitalInputChanged Event

Occurs when a digital input changes state.

C#

```
public event DigitalInputChangedEventHandler DigitalInputChanged;
```

Visual Basic

```
Public Event DigitalInputChanged As DigitalInputChangedEventHandler
```

Example

To respond to DigitalInputChanged events, an event handler must first be registered.

```

Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DigitalInputChanged, AddressOf Device_DigitalInputChanged
End Sub

```

The following example displays a message when the value of the first digital input changes.

```

Private Sub Device_DigitalInputChanged(ByVal sender As Object, ByVal args As
DigitalInputChangedEventArgs)
    If args.DigitalInput.Index = 1 Then
        MessageBox.Show( "Digital Input 1 has changed value." )
    End If
End Sub

```

1.8.4.11 Device.DigitalOutputChanged Event

Occurs when a digital output changes state.

C#

```
public event DigitalOutputChangedEventHandler DigitalOutputChanged;
```

Visual Basic

```
Public Event DigitalOutputChanged As DigitalOutputChangedEventHandler
```

Example

To respond to DigitalOutputChanged events, an event handler must first be registered.

```

Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Startup
    AddHandler Me.Device.DigitalOutputChanged, AddressOf Device_DigitalOutputChanged
End Sub

```

The following example displays a message when the value of the first digital output changes.

```

Private Sub Device_DigitalOutputChanged(ByVal sender As Object, ByVal args As
DigitalOutputChangedEventArgs)
    If args.DigitalOutput.Index = 1 Then
        MessageBox.Show( "Digital Output 1 has changed value." )
    End If
End Sub

```

1.8.4.12 Device.RunningStateChanged Event

Occurs when the running state of the device changes.

C#

```
public event RunningStateChangedEventHandler RunningStateChanged;
```

Visual Basic

```
Public Event RunningStateChanged As RunningStateChangedEventHandler
```

Example

To respond to `RunningStateChanged` events, an event handler must first be registered.

```
Private Sub ThisMeasureProject_Startup(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles Me.Startup
    AddHandler Me.Device.RunningStateChanged, AddressOf Device_RunningStateChanged
End Sub
```

The following example displays a message when the device starts measurement.

```
Private Sub Device_RunningStateChanged(ByVal sender As Object, ByVal args As
    RunningStateChangedEventArgs)
    If args.RunningState = DeviceRunningState.Started Then
        MessageBox.Show( "Device has started measurement." )
    End If
End Sub
```

1.9 DeviceError Class

Represents error or warning information from the device.

File

DeviceError.cs

C#

```
public class DeviceError;
```

Visual Basic

```
Public Class DeviceError
```




Inheritance Hierarchy

DeviceError

1.9.1 DeviceError Members

The following tables list the members exposed by `DeviceError`.




Public Properties

	Name	Description
	Code (see page 54)	Gets the error code.
	Description (see page 54)	Gets a detailed description of the error (eg: "filament 1 defect")
	Type (see page 54)	Gets the error type.

1.9.2 DeviceError Properties

The properties of the DeviceError class are listed here.

Public Properties

	Name	Description
	Code (🔗 see page 54)	Gets the error code.
	Description (🔗 see page 54)	Gets a detailed description of the error (eg: "filament 1 defect")
	Type (🔗 see page 54)	Gets the error type.

1.9.2.1 DeviceError.Code Property

Gets the error code.

C#

```
public String Code;
```

Visual Basic

```
Public Property Code() As String
```

Remarks

This is a three digit number of the form E001 for errors and W001 for warnings. See the device documentation for a detailed list of the error codes.

1.9.2.2 DeviceError.Description Property

Gets a detailed description of the error (eg: "filament 1 defect")

C#

```
public String Description;
```

Visual Basic

```
Public Property Description() As String
```

1.9.2.3 DeviceError.Type Property

Gets the error type.

C#

```
public DeviceErrorType Type;
```

Visual Basic

```
Public Property Type() As DeviceErrorType
```

1.10 DeviceEventArgs Class

Provides data for the DeviceError (🔗 see page 55) event.

File

DeviceEventArgs.cs

C#

```
public class DeviceEventArgs;
```

Visual Basic

```
Public Class DeviceEventArgs
```

Inheritance Hierarchy

DeviceEventArgs

1.10.1 DeviceEventArgs Members

The following tables list the members exposed by DeviceEventArgs.

Public Properties

	Name	Description
	DeviceError (see page 55)	Gets the device error information.

1.10.2 DeviceEventArgs Properties

The properties of the DeviceEventArgs class are listed here.

Public Properties

	Name	Description
	DeviceError (see page 55)	Gets the device error information.

1.10.2.1 DeviceEventArgs.DeviceError Property

Gets the device error information.

C#

```
public DeviceError DeviceError;
```

Visual Basic

```
Public Property DeviceError() As DeviceError
```

1.11 Devices Class

Represents the devices which are currently available in QUADERA.

File

Devices.cs

C#

```
public class Devices : IEnumerable;
```

Visual Basic

```
Public Class Devices
    Inherits IEnumerable
```

Inheritance Hierarchy

Devices

1.11.1 Devices Members

The following tables list the members exposed by Devices.

Public Methods

	Name	Description
	this (see page 56)	Gets a Device (see page 20) object from the collection.

Public Properties

	Name	Description
	Application (see page 57)	Gets the QUADERA Application object.
	Count (see page 57)	Gets the number of Device (see page 20) objects in the collection.
	Parent (see page 58)	Gets the parent Application (see page 57) object.

1.11.2 Devices Methods

The methods of the Devices class are listed here.

Public Methods

	Name	Description
	this (see page 56)	Gets a Device (see page 20) object from the collection.

1.11.2.1 this Indexer

Gets a Device (see page 20) object from the collection.

Overload List

	Name	Description
	Devices.this (int) (see page 56)	Gets the Device (see page 20) object at the specified index in the collection.
	Devices.this (string) (see page 57)	Gets the Device (see page 20) object with the specified name in the collection.

1.11.2.1.1 Devices.this Indexer (int)

Gets the Device (see page 20) object at the specified index in the collection.

C#

```
public Device this[
    int index
```

```
];
```

Visual Basic

```
Public Property Item(index as Integer) as Device
```

Parameters

Parameters	Description
index	The index of the device.

Remarks

The index of the first device is 0.

1.11.2.1.2 Devices.this Indexer (string)

Gets the Device (see page 20) object with the specified name in the collection.

C#

```
public Device this[
    string name
];
```

Visual Basic

```
Public Property Item(name as String) as Device
```



Parameters

Parameters	Description
name	The name of the device.

1.11.3 Devices Properties

The properties of the Devices class are listed here.

Public Properties

	Name	Description
	Application (see page 57)	Gets the QUADERA Application object.
	Count (see page 57)	Gets the number of Device (see page 20) objects in the collection.
	Parent (see page 58)	Gets the parent Application (see page 57) object.

1.11.3.1 Devices.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.11.3.2 Devices.Count Property

Gets the number of Device (see page 20) objects in the collection.

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.11.3.3 Devices.Parent Property

Gets the parent Application (see page 57) object.

C#

```
public Application Parent;
```

Visual Basic

```
Public Property Parent() As Application
```

1.12 DeviceAttachedState Enumeration

Represents the device attached state.

File

Device.cs

C#

```
public enum DeviceAttachedState {
    Attaching = 1,
    Attached = 2,
    Detaching = 3,
    Detached = 4
}
```

Visual Basic

```
Public Enum DeviceAttachedState
    Attaching = 1
    Attached = 2
    Detaching = 3
    Detached = 4
End Enum
```

Members

Members	Description
Attaching	Device (see page 20) is attaching
Attached	Device (see page 20) is attached
Detaching	Device (see page 20) is Detaching
Detached	Device (see page 20) is Detached

1.13 DeviceAttachedStateChangedEventArgs Class

Provides data for the DeviceAttachedStateChanged event.

File

DeviceAttachedStateChangedEventArgs.cs

C#

```
public class DeviceAttachedStateChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DeviceAttachedStateChangedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

DeviceAttachedStateChangedEventArgs

1.13.1 DeviceAttachedStateChangedEventArgs Members

The following tables list the members exposed by DeviceAttachedStateChangedEventArgs.

Public Properties

	Name	Description
	AttachedState (see page 59)	Gets the new AttachedState of the device.

1.13.2 DeviceAttachedStateChangedEventArgs Properties

The properties of the DeviceAttachedStateChangedEventArgs class are listed here.

Public Properties

	Name	Description
	AttachedState (see page 59)	Gets the new AttachedState of the device.

1.13.2.1 DeviceAttachedStateChangedEventArgs.AttachedState Property

Gets the new AttachedState of the device.

C#

```
public DeviceAttachedState AttachedState;
```

Visual Basic

```
Public Property AttachedState() As DeviceAttachedState
```

1.14 DeviceConnectedState Enumeration

Represents the connected state of the device.

File

Device.cs

C#

```
public enum DeviceConnectedState {
    Connected = 1,
    Disconnected = 2,
    Connecting = 3,
    Disconnecting = 4
}
```

Visual Basic

```
Public Enum DeviceConnectedState
    Connected = 1
    Disconnected = 2
    Connecting = 3
    Disconnecting = 4
End Enum
```

Members

Members	Description
Connected	The device is connected.
Disconnected	The device is disconnected.
Connecting	The device is executing the Connect command.
Disconnecting	The device is executing the Disconnect command.

1.15 DeviceConnectionStateChangedEventArgs Class

Provides data for the DeviceConnectionStateChanged event.

File

DeviceConnectionStateChangedEventArgs.cs

C#

```
public class DeviceConnectionStateChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DeviceConnectionStateChangedEventArgs
    Inherits EventArgs
```


Inheritance Hierarchy

DeviceConnectionStateChangedEventArgs

1.15.1 DeviceConnectionStateChangedEventArgs Members

The following tables list the members exposed by DeviceConnectionStateChangedEventArgs.


Public Properties

	Name	Description
	ConnectedState (see page 61)	Gets the new ConnectionState of the device.

1.15.2 DeviceConnectionStateChangedEventArgs Properties

The properties of the DeviceConnectionStateChangedEventArgs class are listed here.

Public Properties

	Name	Description
	ConnectedState (see page 61)	Gets the new ConnectionState of the device.

1.15.2.1

DeviceConnectionStateChangedEventArgs.ConnectedState Property

Gets the new ConnectionState of the device.

C#

```
public DeviceConnectedState ConnectedState;
```

Visual Basic

```
Public Property ConnectedState() As DeviceConnectedState
```

1.16 DeviceErrorType Enumeration

Represents the device error type.

File

DeviceError.cs

C#

```
public enum DeviceErrorType {
    DeviceError = 1,
    DeviceWarning = 2
}
```

Visual Basic

```
Public Enum DeviceErrorType
    DeviceError = 1
    DeviceWarning = 2
End Enum
```

Members

Members	Description
DeviceError	An error condition raised by the device.
DeviceWarning	A warning condition raised by the device.

1.17 DeviceFilamentState Enumeration

Represents the filament state.

File

Device.cs

C#

```
public enum DeviceFilamentState {
    Off = 1,
    On = 2,
    NotReady = 3
}
```

Visual Basic

```
Public Enum DeviceFilamentState
    Off = 1
    On = 2
    NotReady = 3
End Enum
```

Members

Members	Description
Off	The filament is off.
On	The filament is on.

1.18 DeviceFilamentStateChangedEventArgs Class

Provides data for the DeviceFilamentStateChanged event.

File

DeviceFilamentStateChangedEventArgs.cs

C#

```
public class DeviceFilamentStateChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DeviceFilamentStateChangedEventArgs
    Inherits EventArgs
```


Inheritance Hierarchy

DeviceFilamentStateChangedEventArgs

1.18.1 DeviceFilamentStateChangedEventArgs Members

The following tables list the members exposed by DeviceFilamentStateChangedEventArgs.


Public Properties

	Name	Description
	FilamentState (see page 63)	Gets the new FilamentState of the device.

1.18.2 DeviceFilamentStateChangedEventArgs Properties

The properties of the DeviceFilamentStateChangedEventArgs class are listed here.

Public Properties

	Name	Description
	FilamentState (see page 63)	Gets the new FilamentState of the device.

1.18.2.1 DeviceFilamentStateChangedEventArgs.FilamentState Property

Gets the new FilamentState of the device.

C#

```
public DeviceFilamentState FilamentState;
```

Visual Basic

```
Public Property FilamentState() As DeviceFilamentState
```

1.19 DeviceRunningState Enumeration

Represents the running state of the device.

File

Device.cs

C#

```
public enum DeviceRunningState {
    Running = 1,
    Stopped = 2,
    Starting = 3,
    Stopping = 4
}
```

Visual Basic

```
Public Enum DeviceRunningState
    Running = 1
    Stopped = 2
    Starting = 3
    Stopping = 4
End Enum
```

Members

Members	Description
Running	The device is running.
Stopped	The device is stopped.
Starting	The device is executing the start command.

Stopping	The device is executing the stop command.
----------	---

1.20 DeviceSemState Enumeration

Represents the SEM state.

File

Device.cs

C#

```
public enum DeviceSemState {
    Off = 1,
    On = 2,
    NotReady = 3
}
```

Visual Basic

```
Public Enum DeviceSemState
    Off = 1
    On = 2
    NotReady = 3
End Enum
```

Members

Members	Description
Off	The SEM is off.
On	The SEM is on.

1.21 DeviceSemStateChangedEventArgs Class

Provides data for the DeviceSemStateChanged event.

File

DeviceSemStateChangedEventArgs.cs

C#

```
public class DeviceSemStateChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DeviceSemStateChangedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

DeviceSemStateChangedEventArgs

1.21.1 DeviceSemStateChangedEventArgs Members

The following tables list the members exposed by DeviceSemStateChangedEventArgs.


Public Properties

	Name	Description
	SemState (🔗 see page 65)	Gets the new SemState of the device.

1.21.2 DeviceSemStateChangedEventArgs Properties

The properties of the DeviceSemStateChangedEventArgs class are listed here.

Public Properties

	Name	Description
	SemState (🔗 see page 65)	Gets the new SemState of the device.

1.21.2.1 DeviceSemStateChangedEventArgs.SemState Property

Gets the new SemState of the device.

C#

```
public DeviceSemState SemState;
```

Visual Basic

```
Public Property SemState() As DeviceSemState
```

1.22 DeviceStartMode Enumeration

Specifies the measurement mode of the device.

File

Device.cs

C#

```
public enum DeviceStartMode {
    Default = 0,
    SingleCycle = 1
}
```

Visual Basic

```
Public Enum DeviceStartMode
    Default = 0
    SingleCycle = 1
End Enum
```

Members

Members	Description
Default	Start the device in normal mode.
SingleCycle	Start the device in single cycle mode. After the device has completed a measurement cycle it will automatically stop.

1.23 DeviceTotalPressureState Enumeration

Represents the total pressure state.

File

Device.cs

C#

```
public enum DeviceTotalPressureState {  
    On = 0,  
    Off = 1,  
    SensorError = 2,  
    UnderRange = 3,  
    OverRange = 4  
}
```

Visual Basic

```
Public Enum DeviceTotalPressureState  
    On = 0  
    Off = 1  
    SensorError = 2  
    UnderRange = 3  
    OverRange = 4  
End Enum
```

Members

Members	Description
On	The total pressure is on.
Off	The total pressure is off.
SensorError	Sensor Error
UnderRange	Under Range
OverRange	Over Range

1.24 DeviceTPValueChangedEventArgs Class

Provides data for the DeviceSemStateChanged event.

File

DeviceTPValueChangedEventArgs.cs

C#

```
public class DeviceTPValueChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DeviceTPValueChangedEventArgs  
    Inherits EventArgs
```

Inheritance Hierarchy

DeviceTPValueChangedEventArgs

1.24.1 DeviceTPValueChangedEventArgs Members

The following tables list the members exposed by DeviceTPValueChangedEventArgs.

Public Properties

	Name	Description
	Value (see page 67)	Gets the new SemState of the device.

1.24.2 DeviceTPValueChangedEventArgs Properties

The properties of the DeviceTPValueChangedEventArgs class are listed here.

Public Properties

	Name	Description
	Value (see page 67)	Gets the new SemState of the device.

1.24.2.1 DeviceTPValueChangedEventArgs.Value Property

Gets the new SemState of the device.

C#

```
public float Value;
```

Visual Basic

```
Public Property Value() As float
```

1.25 DeviceType Enumeration

Represents the device type.

File

Device.cs

C#

```
public enum DeviceType {
    QMG700 = 1,
    QMG220 = 2
}
```

Visual Basic

```
Public Enum DeviceType
    QMG700 = 1
    QMG220 = 2
End Enum
```

Members

Members	Description
QMG700	The device is a QMG 700.
QMG220	The device is a QMG 220.

1.26 DigitalInput Class

Represents a single digital input controlled by the device.

File

DigitalInput.cs

C#

```
public class DigitalInput;
```

Visual Basic

```
Public Class DigitalInput
```







Inheritance Hierarchy

DigitalInput

1.26.1 DigitalInput Members

The following tables list the members exposed by DigitalInput.







Public Properties

	Name	Description
	Application (see page 69)	Gets the QUADERA Application object.
	Index (see page 69)	Gets the index of this digital input.
	IOType (see page 69)	Gets the IO (see page 80) component type which this digital input belongs to.
	Name (see page 69)	Gets the name of this digital input.
	Parent (see page 69)	Gets the parent DigitalInputs (see page 71) collection object.
	Value (see page 70)	Gets a boolean value representing the state of a digital input.

1.26.2 DigitalInput Properties

The properties of the DigitalInput class are listed here.

Public Properties

	Name	Description
	Application (see page 69)	Gets the QUADERA Application object.
	Index (see page 69)	Gets the index of this digital input.
	IOType (see page 69)	Gets the IO (see page 80) component type which this digital input belongs to.
	Name (see page 69)	Gets the name of this digital input.
	Parent (see page 69)	Gets the parent DigitalInputs (see page 71) collection object.
	Value (see page 70)	Gets a boolean value representing the state of a digital input.

1.26.2.1 DigitalInput.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.26.2.2 DigitalInput.Index Property

Gets the index of this digital input.

C#

```
public int Index;
```

Visual Basic

```
Public Property Index() As Integer
```

Remarks

The index of the first digital input is 1.

1.26.2.3 DigitalInput.IOType Property

Gets the IO ([see page 80](#)) component type which this digital input belongs to.

C#

```
public IOType IOType;
```

Visual Basic

```
Public Property IOType() As IOType
```

1.26.2.4 DigitalInput.Name Property

Gets the name of this digital input.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

1.26.2.5 DigitalInput.Parent Property

Gets the parent DigitalInputs ([see page 71](#)) collection object.

C#

```
public DigitalInputs Parent;
```

Visual Basic

```
Public Property Parent() As DigitalInputs
```

1.26.2.6 DigitalInput.Value Property

Gets a boolean value representing the state of a digital input.

C#

```
public bool Value;
```

Visual Basic

```
Public Property Value() As bool
```

Remarks

This property can only be used when the device is connected.

Example

The following example code displays a message depending on the state of the first digital input.

```
Sub ReadDigitalInputState()
    If Me.Device.DigitalInputs(1).Value = True Then
        MsgBox "State is on"
    Else
        MsgBox "State is off"
    End If
End Sub
```

1.27 DigitalInputChangedEventArgs Class

Provides data for the DigitalInputChanged event.

File

DigitalInputChangedEventArgs.cs

C#

```
public class DigitalInputChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DigitalInputChangedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

DigitalInputChangedEventArgs

1.27.1 DigitalInputChangedEventArgs Members

The following tables list the members exposed by DigitalInputChangedEventArgs.

Public Properties

	Name	Description
	DigitalInput (see page 71)	Gets the DigitalInput object which has changed.

1.27.2 DigitalInputChangedEventArgs Properties

The properties of the DigitalInputChangedEventArgs class are listed here.

Public Properties

	Name	Description
	DigitalInput (see page 71)	Gets the DigitalInput object which has changed.

1.27.2.1 DigitalInputChangedEventArgs.DigitalInput Property

Gets the DigitalInput object which has changed.

C#

```
public DigitalInput DigitalInput;
```

Visual Basic

```
Public Property DigitalInput() As DigitalInput
```

1.28 DigitalInputs Class

Represents the collection of DigitalInput ([see page 68](#)) objects controlled by the device. If Device ([see page 20](#)) is not connected or hardware is not installed, the count will be zero

File

DigitalInputs.cs

C#

```
public class DigitalInputs : IEnumerable;
```

Visual Basic

```
Public Class DigitalInputs
Inherits IEnumerable
```

Inheritance Hierarchy

DigitalInputs


1.28.1 DigitalInputs Members




The following tables list the members exposed by DigitalInputs.

Public Methods

	Name	Description
	this (see page 72)	This is the overview for the this indexer overload.

Public Properties


	Name	Description
	Application (see page 73)	Gets the QUADERA Application object.

	Count (see page 73)	Gets the number of DigitalInput (see page 68) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 73)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 74)	Gets the parent Device (see page 20) object.

1.28.2 DigitalInputs Methods

The methods of the DigitalInputs class are listed here.



Public Methods

	Name	Description
	this (see page 72)	This is the overview for the this indexer overload.

1.28.2.1 this Indexer

This is the overview for the this indexer overload.

Overload List

	Name	Description
	DigitalInputs.this (int) (see page 72)	Gets the DigitalInput (see page 68) object at the specified index in the collection.
	DigitalInputs.this (String) (see page 72)	Gets the DigitalInput (see page 68) object with the specified name in the collection.

1.28.2.1.1 DigitalInputs.this Indexer (String)

Gets the DigitalInput ([see page 68](#)) object with the specified name in the collection.

C#

```
public DigitalInput this[
    String name
];
```

Visual Basic

```
Public Property Item(name as String) as DigitalInput
```

Parameters

Parameters	Description
name	The name of the digital input.

1.28.2.1.2 DigitalInputs.this Indexer (int)

Gets the DigitalInput ([see page 68](#)) object at the specified index in the collection.

C#

```
public DigitalInput this[
    int index
];
```

Visual Basic

```
Public Property Item(index as Integer) as DigitalInput
```

Parameters

Parameters	Description
index	The index of the digital input.





Remarks

The index of the first digital input is 1.

1.28.3 DigitalInputs Properties

The properties of the DigitalInputs class are listed here.

Public Properties

	Name	Description
	Application (↗ see page 73)	Gets the QUADERA Application object.
	Count (↗ see page 73)	Gets the number of DigitalInput (↗ see page 68) objects in the collection. If Device (↗ see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (↗ see page 73)	Indicates whether IO (↗ see page 80) hardware is installed
	Parent (↗ see page 74)	Gets the parent Device (↗ see page 20) object.

1.28.3.1 DigitalInputs.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.28.3.2 DigitalInputs.Count Property

Gets the number of DigitalInput ([↗](#) see page 68) objects in the collection. If Device ([↗](#) see page 20) is not connected or hardware is not installed, the count will be zero

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.28.3.3 DigitalInputs.HardwareInstalled Property

Indicates whether IO ([↗](#) see page 80) hardware is installed

C#

```
public bool HardwareInstalled;
```

Visual Basic

```
Public Property HardwareInstalled() As bool
```

1.28.3.4 DigitalInputs.Parent Property

Gets the parent Device (see page 20) object.

C#

```
public Device Parent;
```

Visual Basic

```
Public Property Parent() As Device
```

1.29 DigitalOutput Class

Represents a single digital output controlled by the device.

File

DigitalOutput.cs

C#

```
public class DigitalOutput;
```

Visual Basic

```
Public Class DigitalOutput
```







Inheritance Hierarchy

DigitalOutput

1.29.1 DigitalOutput Members

The following tables list the members exposed by DigitalOutput.







Public Properties

	Name	Description
	Application (see page 75)	Gets the QUADERA Application object.
	Index (see page 75)	Gets the index of this digital output.
	IOType (see page 75)	Gets the IO (see page 80) component type which this digital output belongs to.
	Name (see page 75)	Gets the name of this digital output.
	Parent (see page 76)	Gets the parent DigitalOutputs (see page 77) collection object.
	Value (see page 76)	Gets or sets a boolean value representing the state of a digital output.

1.29.2 DigitalOutput Properties

The properties of the DigitalOutput class are listed here.

Public Properties

	Name	Description
	Application (see page 75)	Gets the QUADERA Application object.
	Index (see page 75)	Gets the index of this digital output.
	IOType (see page 75)	Gets the IO (see page 80) component type which this digital output belongs to.
	Name (see page 75)	Gets the name of this digital output.
	Parent (see page 76)	Gets the parent DigitalOutputs (see page 77) collection object.
	Value (see page 76)	Gets or sets a boolean value representing the state of a digital output.

1.29.2.1 DigitalOutput.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.29.2.2 DigitalOutput.Index Property

Gets the index of this digital output.

C#

```
public int Index;
```

Visual Basic

```
Public Property Index() As Integer
```

Remarks

The index of the first digital output is 1.

1.29.2.3 DigitalOutput.IOType Property

Gets the IO ([see page 80](#)) component type which this digital output belongs to.

C#

```
public IOType IOType;
```

Visual Basic

```
Public Property IOType() As IOType
```

1.29.2.4 DigitalOutput.Name Property

Gets the name of this digital output.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

1.29.2.5 DigitalOutput.Parent Property

Gets the parent DigitalOutputs (see page 77) collection object.

C#

```
public DigitalOutputs Parent;
```

Visual Basic

```
Public Property Parent() As DigitalOutputs
```

1.29.2.6 DigitalOutput.Value Property

Gets or sets a boolean value representing the state of a digital output.

C#

```
public bool Value;
```

Visual Basic

```
Public Property Value() As bool
```

Remarks

This property can only be used when the device is connected.

Example

The following example code sets the state of the first digital output to on.

```
Sub SetDigitalOutputState()  
    Me.Device.DigitalOutputs(1).Value = True  
End Sub
```

1.30 DigitalOutputChangedEventArgs Class

Provides data for the DigitalOutputChanged event.

File

DigitalOutputChangedEventArgs.cs

C#

```
public class DigitalOutputChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class DigitalOutputChangedEventArgs  
    Inherits EventArgs
```


Inheritance Hierarchy

```
DigitalOutputChangedEventArgs
```

1.30.1 DigitalOutputChangedEventArgs Members

The following tables list the members exposed by DigitalOutputChangedEventArgs.

Public Properties

	Name	Description
	DigitalOutput (see page 77)	Gets the DigitalOutput object which has changed.

1.30.2 DigitalOutputChangedEventArgs Properties

The properties of the DigitalOutputChangedEventArgs class are listed here.

Public Properties

	Name	Description
	DigitalOutput (see page 77)	Gets the DigitalOutput object which has changed.

1.30.2.1 DigitalOutputChangedEventArgs.DigitalOutput Property

Gets the DigitalOutput object which has changed.

C#

```
public DigitalOutput DigitalOutput;
```

Visual Basic

```
Public Property DigitalOutput() As DigitalOutput
```

1.31 DigitalOutputs Class

Represents the collection of DigitalOutput ([see page 74](#)) objects controlled by the device. If Device ([see page 20](#)) is not connected or hardware is not installed, the count will be zero

File

DigitalOutputs.cs

C#

```
public class DigitalOutputs : IEnumerable;
```

Visual Basic

```
Public Class DigitalOutputs
    Inherits IEnumerable
```


Inheritance Hierarchy

DigitalOutputs






1.31.1 DigitalOutputs Members

The following tables list the members exposed by DigitalOutputs.

Public Methods

	Name	Description
	this (see page 78)	This is the overview for the this indexer overload.


Public Properties

	Name	Description
	Application (see page 79)	Gets the QUADERA Application object.
	Count (see page 79)	Gets the number of DigitalOutput (see page 74) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 79)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 80)	Gets the parent Device (see page 20) object.
	Values (see page 80)	Gets the Array of bits containing Digital output values.

1.31.2 DigitalOutputs Methods

The methods of the DigitalOutputs class are listed here.



Public Methods

	Name	Description
	this (see page 78)	This is the overview for the this indexer overload.

1.31.2.1 this Indexer

This is the overview for the this indexer overload.

Overload List

	Name	Description
	DigitalOutputs.this (int) (see page 78)	Gets the DigitalOutput (see page 74) object at the specified index in the collection.
	DigitalOutputs.this (String) (see page 78)	Gets the DigitalOutput (see page 74) object with the specified name in the collection.

1.31.2.1.1 DigitalOutputs.this Indexer (String)

Gets the DigitalOutput ([see page 74](#)) object with the specified name in the collection.

C#

```
public DigitalOutput this[
    String name
];
```

Visual Basic

```
Public Property Item(name as String) as DigitalOutput
```

Parameters

Parameters	Description
name	The name of the digital output.

1.31.2.1.2 DigitalOutputs.this Indexer (int)

Gets the DigitalOutput ([see page 74](#)) object at the specified index in the collection.

C#

```
public DigitalOutput this[
```

```

    int index
};

```

Visual Basic

```
Public Property Item(index as Integer) as DigitalOutput
```

Parameters

Parameters	Description
index	The index of the digital output.






Remarks

The index of the first digital output is 1.

1.31.3 DigitalOutputs Properties

The properties of the DigitalOutputs class are listed here.

Public Properties

	Name	Description
	Application (see page 79)	Gets the QUADERA Application object.
	Count (see page 79)	Gets the number of DigitalOutput (see page 74) objects in the collection. If Device (see page 20) is not connected or hardware is not installed, the count will be zero
	HardwareInstalled (see page 79)	Indicates whether IO (see page 80) hardware is installed
	Parent (see page 80)	Gets the parent Device (see page 20) object.
	Values (see page 80)	Gets the Array of bits containing Digital output values.

1.31.3.1 DigitalOutputs.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.31.3.2 DigitalOutputs.Count Property

Gets the number of DigitalOutput ([see page 74](#)) objects in the collection. If Device ([see page 20](#)) is not connected or hardware is not installed, the count will be zero

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.31.3.3 DigitalOutputs.HardwareInstalled Property

Indicates whether IO ([see page 80](#)) hardware is installed

C#

```
public bool HardwareInstalled;
```

Visual Basic

```
Public Property HardwareInstalled() As bool
```

1.31.3.4 DigitalOutputs.Parent Property

Gets the parent Device (see page 20) object.

C#

```
public Device Parent;
```

Visual Basic

```
Public Property Parent() As Device
```

1.31.3.5 DigitalOutputs.Values Property

Gets the Array of bits containing Digital output values.

C#

```
public UInt32 Values;
```

Visual Basic

```
Public Property Values() As UInt32
```

1.32 IO Class

Represents an IO component such as the IO700 or IO220.

File

IO.cs

C#

```
public class IO;
```

Visual Basic

```
Public Class IO
```

Remarks

With this class the user can access the external input / output values for a device.







Inheritance Hierarchy

IO

1.32.1 IO Members

The following tables list the members exposed by IO.







Public Properties

	Name	Description
	AnalogInputs (see page 81)	Gets the AnalogInputs collection object which contains the analog inputs for this IO (see page 80) component.
	AnalogOutputs (see page 81)	Gets the AnalogOutputs collection object which contains the analog outputs for this IO (see page 80) component.
	DigitalInputs (see page 82)	Gets the DigitalInputs collection object which contains the digital inputs for this IO (see page 80) component.
	DigitalOutputs (see page 82)	Gets the DigitalOutputs collection object which contains the digital outputs for this IO (see page 80) component.
	IsInstalled (see page 82)	Gets a boolean value indicating if the IO (see page 80) component is installed.
	Type (see page 82)	Gets the IO (see page 80) component type.

1.32.2 IO Properties

The properties of the IO class are listed here.

Public Properties

	Name	Description
	AnalogInputs (see page 81)	Gets the AnalogInputs collection object which contains the analog inputs for this IO (see page 80) component.
	AnalogOutputs (see page 81)	Gets the AnalogOutputs collection object which contains the analog outputs for this IO (see page 80) component.
	DigitalInputs (see page 82)	Gets the DigitalInputs collection object which contains the digital inputs for this IO (see page 80) component.
	DigitalOutputs (see page 82)	Gets the DigitalOutputs collection object which contains the digital outputs for this IO (see page 80) component.
	IsInstalled (see page 82)	Gets a boolean value indicating if the IO (see page 80) component is installed.
	Type (see page 82)	Gets the IO (see page 80) component type.

1.32.2.1 IO.AnalogInputs Property

Gets the AnalogInputs collection object which contains the analog inputs for this IO ([see page 80](#)) component.

C#

```
public AnalogInputs AnalogInputs;
```

Visual Basic

```
Public Property AnalogInputs() As AnalogInputs
```

1.32.2.2 IO.AnalogOutputs Property

Gets the AnalogOutputs collection object which contains the analog outputs for this IO ([see page 80](#)) component.

C#

```
public AnalogOutputs AnalogOutputs;
```

Visual Basic

```
Public Property AnalogOutputs() As AnalogOutputs
```

1.32.2.3 IO.DigitalInputs Property

Gets the DigitalInputs collection object which contains the digital inputs for this IO (see page 80) component.

C#

```
public DigitalInputs DigitalInputs;
```

Visual Basic

```
Public Property DigitalInputs() As DigitalInputs
```

1.32.2.4 IO.DigitalOutputs Property

Gets the DigitalOutputs collection object which contains the digital outputs for this IO (see page 80) component.

C#

```
public DigitalOutputs DigitalOutputs;
```

Visual Basic

```
Public Property DigitalOutputs() As DigitalOutputs
```

1.32.2.5 IO.IsInstalled Property

Gets a boolean value indicating if the IO (see page 80) component is installed.

C#

```
public bool IsInstalled;
```

Visual Basic

```
Public Property IsInstalled() As bool
```

1.32.2.6 IO.Type Property

Gets the IO (see page 80) component type.

C#

```
public IOType Type;
```

Visual Basic

```
Public Property Type() As IOType
```

1.33 IOType Enumeration

Represents the type of the IO (see page 80) component.

File

IO.cs

C#

```
public enum IOType {  
    IO700_1 = 1,  
    IO700_2 = 2,  
    IO700_3 = 3,  
    IO720 = 4,  
    IO220 = 5  
}
```

```
}

```

Visual Basic

```
Public Enum IOType
    IO700_1 = 1
    IO700_2 = 2
    IO700_3 = 3
    IO720 = 4
    IO220 = 5
End Enum

```

Members

Members	Description
IO700_1	First IO700 component
IO700_2	Second IO700 component
IO700_3	Third IO700 component
IO720	IO720 component
IO220	IO220 component

1.34 MeasureProject Class

Represents a measurement project.

File

MeasureProject.cs

C#

```
public class MeasureProject;

```

Visual Basic

```
Public Class MeasureProject

```

Inheritance Hierarchy

MeasureProject

1.34.1 MeasureProject Members











The following tables list the members exposed by MeasureProject.

Public Methods

	Name	Description
	Activate (see page 84)	Activates the measurement project form and gives it focus.
	Backup (see page 84)	Creates a backup of the measure project with the given file name.
	Close (see page 85)	Closes the measurement project.
	Export (see page 85)	Exports cycle data as text.
	SetBounds (see page 86)	Sets the location and size of the measurement project window.

Public Properties






	Name	Description
	Application (see page 87)	Gets the QUADERA Application object.

	Device (see page 87)	Gets the Device object that is used by this measurement project.
	Height (see page 87)	Gets the height of the measurement project window.
	Left (see page 87)	Gets the left position of the measurement project window.
	MeasurementTasks (see page 87)	Gets the collection of measurement tasks in this project.
	MeasureProjectRecipes (see page 88)	Gets the collection of Recipes.
	Name (see page 88)	Gets the name of this measurement project.
	Parent (see page 88)	Gets the parent MeasureProjects (see page 89) collection object.
	Top (see page 88)	Gets the top position of the measurement project window.
	Type (see page 88)	Gets the measurement project type.
	Width (see page 89)	Gets the width of the measurement project window.

1.34.2 MeasureProject Methods

The methods of the MeasureProject class are listed here.

Public Methods

	Name	Description
	Activate (see page 84)	Activates the measurement project form and gives it focus.
	Backup (see page 84)	Creates a backup of the measure project with the given file name.
	Close (see page 85)	Closes the measurement project.
	Export (see page 85)	Exports cycle data as text.
	SetBounds (see page 86)	Sets the location and size of the measurement project window.

1.34.2.1 MeasureProject.Activate Method

Activates the measurement project form and gives it focus.

C#

```
public void Activate();
```

Visual Basic

```
Public Function Activate() As void
```

1.34.2.2 MeasureProject.Backup Method

Creates a backup of the measure project with the given file name.

C#

```
public void Backup(
    String fileName
);
```

Visual Basic

```
Public Function Backup(
    fileName As String
) As void
```

Parameters

Parameters	Description
fileName	The backup filename.

Remarks

The filename can be absolute or relative. If the filename is relative then the standard data directory will be used. This command cannot be used while the device is running.

1.34.2.3 MeasureProject.Close Method

Closes the measurement project.

C#

```
public void Close();
```



Visual Basic

```
Public Function Close() As void
```

1.34.2.4 Export Method

Exports cycle data as text.

Overload List

	Name	Description
	MeasureProject.Export (String) (see page 85)	Exports the most recent cycle as text to the named file.
	MeasureProject.Export (String, int, int) (see page 85)	Exports the specified cycles as text to the named file.

1.34.2.4.1 MeasureProject.Export Method (String)

Exports the most recent cycle as text to the named file.

C#

```
public void Export(
    String fileName
);
```

Visual Basic

```
Public Function Export(
    fileName As String
) As void
```

Parameters

Parameters	Description
fileName	The filename of the exported data.

Remarks

The filename can be absolute or relative. If the filename is relative then the standard data directory will be used.

This overloaded method exports data for the final cycle.

1.34.2.4.2 MeasureProject.Export Method (String, int, int)

Exports the specified cycles as text to the named file.

C#

```
public void Export(
    String fileName,
    int startCycle,
    int endCycle
);
```

Visual Basic

```
Public Function Export(
    fileName As String,
    startCycle As Integer,
    endCycle As Integer
) As void
```

Parameters

Parameters	Description
fileName	The filename of the exported data.
startCycle	The start of the export data range.
endCycle	The end of the export data range.

Remarks

The filename can be absolute or relative. If the filename is relative then the standard data directory will be used.

1.34.2.5 MeasureProject.SetBounds Method

Sets the location and size of the measurement project window.

C#

```
public void SetBounds(
    int x,
    int y,
    int width,
    int height
);
```

Visual Basic

```
Public Function SetBounds(
    x As Integer,
    y As Integer,
    width As Integer,
    height As Integer
) As void
```



Parameters










Parameters	Description
x	The left position of the window. The top position of the window. The width of the window. The height of the window.

1.34.3 MeasureProject Properties

The properties of the MeasureProject class are listed here.

Public Properties

	Name	Description
	Application (see page 87)	Gets the QUADERA Application object.
	Device (see page 87)	Gets the Device object that is used by this measurement project.

	Height (see page 87)	Gets the height of the measurement project window.
	Left (see page 87)	Gets the left position of the measurement project window.
	MeasurementTasks (see page 87)	Gets the collection of measurement tasks in this project.
	MeasureProjectRecipes (see page 88)	Gets the collection of Recipes.
	Name (see page 88)	Gets the name of this measurement project.
	Parent (see page 88)	Gets the parent MeasureProjects (see page 89) collection object.
	Top (see page 88)	Gets the top position of the measurement project window.
	Type (see page 88)	Gets the measurement project type.
	Width (see page 89)	Gets the width of the measurement project window.

1.34.3.1 MeasureProject.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.34.3.2 MeasureProject.Device Property

Gets the Device object that is used by this measurement project.

C#

```
public Device Device;
```

Visual Basic

```
Public Property Device() As Device
```

1.34.3.3 MeasureProject.Height Property

Gets the height of the measurement project window.

C#

```
public int Height;
```

Visual Basic

```
Public Property Height() As Integer
```

1.34.3.4 MeasureProject.Left Property

Gets the left position of the measurement project window.

C#

```
public int Left;
```

Visual Basic

```
Public Property Left() As Integer
```

1.34.3.5 MeasureProject.MeasurementTasks Property

Gets the collection of measurement tasks in this project.

C#

```
public MeasurementTasks MeasurementTasks;
```

Visual Basic

```
Public Property MeasurementTasks() As MeasurementTasks
```

1.34.3.6 MeasureProject.MeasureProjectRecipes Property

Gets the collection of Recipes.

C#

```
public Recipes MeasureProjectRecipes;
```

Visual Basic

```
Public Property MeasureProjectRecipes() As Recipes
```

1.34.3.7 MeasureProject.Name Property

Gets the name of this measurement project.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

1.34.3.8 MeasureProject.Parent Property

Gets the parent MeasureProjects ([see page 89](#)) collection object.

C#

```
public MeasureProjects Parent;
```

Visual Basic

```
Public Property Parent() As MeasureProjects
```

1.34.3.9 MeasureProject.Top Property

Gets the top position of the measurement project window.

C#

```
public int Top;
```

Visual Basic

```
Public Property Top() As Integer
```

1.34.3.10 MeasureProject.Type Property

Gets the measurement project type.

C#

```
public INFICON.InSitu.QUADERA.VSTA.MeasureProjectType Type;
```

Visual Basic

```
Public Property Type() As INFICON.InSitu.QUADERA.VSTA.MeasureProjectType
```

Remarks

Device (see page 87) operations such as Connect and Start are not possible with a DataFile measure project type.

1.34.3.11 MeasureProject.Width Property

Gets the width of the measurement project window.

C#

```
public int Width;
```

Visual Basic

```
Public Property Width() As Integer
```

1.35 MeasureProjects Class

Represents the collection of open measurement projects in QUADERA.

File

MeasureProjects.cs

C#

```
public class MeasureProjects : IEnumerable;
```

Visual Basic

```
Public Class MeasureProjects
Inherits IEnumerable
```

Inheritance Hierarchy

MeasureProjects




1.35.1 MeasureProjects Members

The following tables list the members exposed by MeasureProjects.

Public Methods

	Name	Description
	Open (see page 90)	Opens the measure project with the given name.
	this (see page 90)	Gets a MeasureProject (see page 83) object from the collection.


Public Properties

	Name	Description
	Application (see page 91)	Gets the QUADERA Application object.
	Count (see page 91)	Gets the number of MeasureProject (see page 83) objects in the collection.
	Parent (see page 92)	Gets the parent Application (see page 91) object.

1.35.2 MeasureProjects Methods

The methods of the MeasureProjects class are listed here.

Public Methods

	Name	Description
	Open (see page 90)	Opens the mesasure project with the given name.
	this (see page 90)	Gets a MeasureProject (see page 83) object from the collection.

1.35.2.1 MeasureProjects.Open Method

Opens the mesasure project with the given name.

C#

```
public MeasureProject Open(
    String fileName
);
```

Visual Basic

```
Public Function Open(
    fileName As String
) As MeasureProject
```

Parameters

Parameters	Description
fileName	The filename of the measurement project.



Remarks

The filename can be absolute or relative. If the filename is relative then the standard data directory will be used.

1.35.2.2 this Indexer

Gets a MeasureProject ([see page 83](#)) object from the collection.

Overload List

	Name	Description
	MeasureProjects.this (int) (see page 90)	Gets the MeasureProject (see page 83) object at the specified index in the collection.
	MeasureProjects.this (string) (see page 91)	Gets the MeasureProject (see page 83) object with the specified name in the collection.

1.35.2.2.1 MeasureProjects.this Indexer (int)

Gets the MeasureProject ([see page 83](#)) object at the specified index in the collection.

C#

```
public MeasureProject this[
    int index
];
```

Visual Basic

```
Public Property Item(index as Integer) as MeasureProject
```

Parameters

Parameters	Description
index	The index of the measurement project.

Remarks

The index of the first measurement project is 0.

1.35.2.2.2 MeasureProjects.this Indexer (string)

Gets the MeasureProject (see page 83) object with the specified name in the collection.

C#

```
public MeasureProject this[
    string name
];
```

Visual Basic

```
Public Property Item(name as String) as MeasureProject
```




Parameters

Parameters	Description
name	The name of the measurement project.

1.35.3 MeasureProjects Properties

The properties of the MeasureProjects class are listed here.

Public Properties

	Name	Description
	Application (see page 91)	Gets the QUADERA Application object.
	Count (see page 91)	Gets the number of MeasureProject (see page 83) objects in the collection.
	Parent (see page 92)	Gets the parent Application (see page 91) object.

1.35.3.1 MeasureProjects.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.35.3.2 MeasureProjects.Count Property

Gets the number of MeasureProject (see page 83) objects in the collection.

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.35.3.3 MeasureProjects.Parent Property

Gets the parent Application (see page 91) object.

C#

```
public Application Parent;
```

Visual Basic

```
Public Property Parent() As Application
```

1.36 MeasureProjectType Enumeration

Represents the type of a measurement project.

File

MeasureProject.cs

C#

```
public enum MeasureProjectType {
    Template = 1,
    DataFile = 2
}
```

Visual Basic

```
Public Enum MeasureProjectType
    Template = 1
    DataFile = 2
End Enum
```

Members

Members	Description
Template	The measure project was created from a template.
DataFile	The measure project is a saved data file.

1.37 MeasurementTask Class

Represents a measurement task in a measurement project.

File

MeasurementTask.cs

C#

```
public class MeasurementTask;
```

Visual Basic

```
Public Class MeasurementTask
```




Inheritance Hierarchy

MeasurementTask












1.37.1 MeasurementTask Members

The following tables list the members exposed by MeasurementTask.

Public Methods

	Name	Description
	GetCycleData (see page 93)	Gets the start time, end time, 3 dimensions of Cycle data and corresponding units.
	GetLastMeasureValue (see page 94)	Gets the Last Measure Value of Measurement Task. Remarks
	GetMeasureTaskStatistics (see page 94)	This is the overview for the GetMeasureTaskStatistics method overload.




Public Properties

	Name	Description
	Application (see page 97)	Gets the QUADERA Application object.
	ApplyBackgroundSubtraction (see page 97)	This is ApplyBackgroundSubtraction, a member of class MeasurementTask.
	ApplyCharacteristic (see page 98)	Used to have characteristics applied to applicable Tasks.
	Cycles (see page 98)	Gets the number of cycles of data stored in task
	Index (see page 98)	Gets the index of this measurement task.
	Name (see page 98)	Gets the Name of Measurement Task.
	Parent (see page 98)	Gets the parent MeasurementTasks (see page 99) collection object.
	TaskID (see page 99)	Gets the Task ID of Measurement Task.
	TaskType (see page 99)	Gets the Type of Measurement Task.
	TaskUnit (see page 99)	Gets the Unit of Measurement Task.
	UsePartialPressure (see page 99)	Used to have Ion Currents converted to Partial Pressure in Mass Tasks.

1.37.2 MeasurementTask Methods

The methods of the MeasurementTask class are listed here.

Public Methods

	Name	Description
	GetCycleData (see page 93)	Gets the start time, end time, 3 dimensions of Cycle data and corresponding units.
	GetLastMeasureValue (see page 94)	Gets the Last Measure Value of Measurement Task. Remarks
	GetMeasureTaskStatistics (see page 94)	This is the overview for the GetMeasureTaskStatistics method overload.

1.37.2.1 MeasurementTask.GetCycleData Method

Gets the start time, end time, 3 dimensions of Cycle data and corresponding units.

C#

```
public bool GetCycleData(
```

```

    int cycle,
    ref DateTime startTime,
    ref DateTime endTime,
    ref string xUnit,
    ref string yUnit,
    ref string zUnit,
    ref long[] xData,
    ref double[] yData,
    ref double[] zData
);

```

Visual Basic

```

Public Function GetCycleData(
    cycle As Integer,
    ByRef startTime As DateTime,
    ByRef endTime As DateTime,
    ByRef xUnit As string,
    ByRef yUnit As string,
    ByRef zUnit As string,
    ByRef xData As long[],
    ByRef yData As double[],
    ByRef zData As double[]
) As bool

```

Parameters

Parameters	Description
cycle	cycle value.
startTime	Start time of cycle.
endTime	End time of cycle.
xUnit	Unit for X Data.
yUnit	Unit for Y Data.
zUnit	Unit for Z Data.
xData	X Data.
yData	Y Data.
zData	Z Data.

1.37.2.2 MeasurementTask.GetLastMeasureValue Method

Gets the Last Measure Value of Measurement Task.

Remarks

C#

```

public double GetLastMeasureValue();

```

Visual Basic

```


Public Function GetLastMeasureValue() As double




```

1.37.2.3 GetMeasureTaskStatistics Method

This is the overview for the GetMeasureTaskStatistics method overload.

Overload List

	Name	Description
	MeasurementTask.GetMeasureTaskStatistics (double, double, double, double) (🔗 see page 95)	Gets the Minimum, Maximum, Mean and Standard Deviation of the Measurement Task.

	MeasurementTask.GetMeasureTaskStatistics (double, double, double, double, DateTime) (see page 95)	Gets the Minamin, Maxamin, Mean and Standard Deviation of the Measurement Task.
	MeasurementTask.GetMeasureTaskStatistics (double, double, double, double, DateTime, int) (see page 96)	This is GetMeasureTaskStatistics, a member of class MeasurementTask.
	MeasurementTask.GetMeasureTaskStatistics (double, double, double, double, int) (see page 96)	Gets the Minimum, Maximum, Mean and Standard Deviation of the Measurement Task.

1.37.2.3.1 MeasurementTask.GetMeasureTaskStatistics Method (double, double, double, double)

Gets the Minimum, Maximum, Mean and Standard Deviation of the Measurement Task.

C#

```
public bool GetMeasureTaskStatistics(
    ref double min,
    ref double max,
    ref double mean,
    ref double stdDeviation
);
```

Visual Basic

```
Public Function GetMeasureTaskStatistics(
    ByRef min As double,
    ByRef max As double,
    ByRef mean As double,
    ByRef stdDeviation As double
) As bool
```

Parameters

Parameters	Description
min	Min value.
max	Max value.
mean	Mean value.
stdDeviation	Standard Deviation.

1.37.2.3.2 MeasurementTask.GetMeasureTaskStatistics Method (double, double, double, double, DateTime, DateTime)

Gets the Minamin, Maxamin, Mean and Standard Deviation of the Measurement Task.

C#

```
public bool GetMeasureTaskStatistics(
    ref double min,
    ref double max,
    ref double mean,
    ref double stdDeviation,
    DateTime start,
    DateTime end
);
```

Visual Basic

```
Public Function GetMeasureTaskStatistics(
    ByRef min As double,
    ByRef max As double,
    ByRef mean As double,
    ByRef stdDeviation As double,
```

```

    start As DateTime,
    end As DateTime
) As bool

```

Parameters

Parameters	Description
min	Min value.
max	Max value.
mean	Mean value.
stdDeviation	Standard Deviation.
start	Start of range value.
end	End of range value. Remarks

1.37.2.3.3 MeasurementTask.GetMeasureTaskStatistics Method (double, double, double, double, DateTime, DateTime, int)

C#

```

public bool GetMeasureTaskStatistics(
    ref double min,
    ref double max,
    ref double mean,
    ref double stdDeviation,
    DateTime start,
    DateTime end,
    int skipFirstValues
);

```

Visual Basic

```

Public Function GetMeasureTaskStatistics(
    ByRef min As double,
    ByRef max As double,
    ByRef mean As double,
    ByRef stdDeviation As double,
    start As DateTime,
    end As DateTime,
    skipFirstValues As Integer
) As bool

```

Remarks

This is GetMeasureTaskStatistics, a member of class MeasurementTask.

1.37.2.3.4 MeasurementTask.GetMeasureTaskStatistics Method (double, double, double, double, int)

Gets the Minimum, Maximum, Mean and Standard Deviation of the Measurement Task.

C#

```

public bool GetMeasureTaskStatistics(
    ref double min,
    ref double max,
    ref double mean,
    ref double stdDeviation,
    int skipFirstValues
);

```

Visual Basic

```

Public Function GetMeasureTaskStatistics(
    ByRef min As double,
    ByRef max As double,
    ByRef mean As double,

```

```

    ByRef stdDeviation As double,
    skipFirstValues As Integer
) As bool

```












Parameters

Parameters	Description
min	Min value.
max	Max value.
mean	Mean value.
stdDeviation	Standard Deviation.

1.37.3 MeasurementTask Properties

The properties of the MeasurementTask class are listed here.

Public Properties

	Name	Description
	Application (see page 97)	Gets the QUADERA Application object.
	ApplyBackgroundSubtraction (see page 97)	This is ApplyBackgroundSubtraction, a member of class MeasurementTask.
	ApplyCharacteristic (see page 98)	Used to have characteristics applied to applicable Tasks.
	Cycles (see page 98)	Gets the number of cycles of data stored in task
	Index (see page 98)	Gets the index of this measurement task.
	Name (see page 98)	Gets the Name of Measurement Task.
	Parent (see page 98)	Gets the parent MeasurementTasks (see page 99) collection object.
	TaskID (see page 99)	Gets the Task ID of Measurement Task.
	TaskType (see page 99)	Gets the Type of Measurement Task.
	TaskUnit (see page 99)	Gets the Unit of Measurement Task.
	UsePartialPressure (see page 99)	Used to have Ion Currents converted to Partial Pressure in Mass Tasks.

1.37.3.1 MeasurementTask.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.37.3.2 MeasurementTask.ApplyBackgroundSubtraction Property

C#

```
public bool ApplyBackgroundSubtraction;
```

Visual Basic

```
Public Property ApplyBackgroundSubtraction() As bool
```

Remarks

This is ApplyBackgroundSubtraction, a member of class MeasurementTask.

1.37.3.3 MeasurementTask.ApplyCharacteristic Property

Used to have characteristics applied to applicable Tasks.

C#

```
public bool ApplyCharacteristic;
```

Visual Basic

```
Public Property ApplyCharacteristic() As bool
```

1.37.3.4 MeasurementTask.Cycles Property

Gets the number of cycles of data stored in task

C#

```
public int Cycles;
```

Visual Basic

```
Public Property Cycles() As Integer
```

1.37.3.5 MeasurementTask.Index Property

Gets the index of this measurement task.

C#

```
public int Index;
```

Visual Basic

```
Public Property Index() As Integer
```

1.37.3.6 MeasurementTask.Name Property

Gets the Name of Measurement Task.

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

Remarks

The Name of the Measurement Task.

1.37.3.7 MeasurementTask.Parent Property

Gets the parent MeasurementTasks ([see page 99](#)) collection object.

C#

```
public MeasurementTasks Parent;
```

Visual Basic

```
Public Property Parent() As MeasurementTasks
```

1.37.3.8 MeasurementTask.TaskID Property

Gets the Task ID of Measurement Task.

C#

```
public long TaskID;
```

Visual Basic

```
Public Property TaskID() As long
```

1.37.3.9 MeasurementTask.TaskType Property

Gets the Type of Measurement Task.

C#

```
public string TaskType;
```

Visual Basic

```
Public Property TaskType() As string
```

Remarks

The Type of the Measurement Task.

1.37.3.10 MeasurementTask.TaskUnit Property

Gets the Unit of Measurement Task.

C#

```
public string TaskUnit;
```

Visual Basic

```
Public Property TaskUnit() As string
```

Remarks

The Unit may change depending on Characteristics that have been applied.

1.37.3.11 MeasurementTask.UsePartialPressure Property

Used to have Ion Currents converted to Partial Pressure in Mass Tasks.

C#

```
public bool UsePartialPressure;
```

Visual Basic

```
Public Property UsePartialPressure() As bool
```

1.38 MeasurementTasks Class

Represents the collection of MeasurementTask (see page 92) objects controlled by the device.

File

MeasurementTasks.cs

C#

```
public class MeasurementTasks : IEnumerable;
```

Visual Basic

```
Public Class MeasurementTasks
    Inherits IEnumerable
```

Inheritance Hierarchy

MeasurementTasks

1.38.1 MeasurementTasks Members

The following tables list the members exposed by MeasurementTasks.

Public Methods

	Name	Description
	this (see page 100)	Gets the MeasurementTask (see page 92) object at the specified index in the collection.

Public Properties

	Name	Description
	Application (see page 101)	Gets the QUADERA Application object.
	Count (see page 101)	Gets the number of MeasurementTask (see page 92) objects in the collection.
	Parent (see page 101)	Gets the parent Device (see page 20) object.

1.38.2 MeasurementTasks Methods

The methods of the MeasurementTasks class are listed here.

Public Methods

	Name	Description
	this (see page 100)	Gets the MeasurementTask (see page 92) object at the specified index in the collection.

1.38.2.1 MeasurementTasks.this Indexer

Gets the MeasurementTask (see page 92) object at the specified index in the collection.

C#

```
public MeasurementTask this[
    int index
];
```

Visual Basic

```
Public Property Item(index as Integer) as MeasurementTask
```

Parameters

Parameters	Description
index	The index of the analog output.




Remarks

The index of the first MeasurementTask (see page 92) is 0.

1.38.3 MeasurementTasks Properties

The properties of the MeasurementTasks class are listed here.

Public Properties

	Name	Description
	Application (see page 101)	Gets the QUADERA Application object.
	Count (see page 101)	Gets the number of MeasurementTask (see page 92) objects in the collection.
	Parent (see page 101)	Gets the parent Device (see page 20) object.

1.38.3.1 MeasurementTasks.Application Property

Gets the QUADERA Application object.

C#

```
public Application Application;
```

Visual Basic

```
Public Property Application() As Application
```

1.38.3.2 MeasurementTasks.Count Property

Gets the number of MeasurementTask (see page 92) objects in the collection.

C#

```
public int Count;
```

Visual Basic

```
Public Property Count() As Integer
```

1.38.3.3 MeasurementTasks.Parent Property

Gets the parent Device (see page 20) object.

C#

```
public MeasureProject Parent;
```

Visual Basic

```
Public Property Parent() As MeasureProject
```

1.39 RunningStateChangedEventArgs Class

Provides data for the RunningStateChanged event.

File

RunningStateChangedEventArgs.cs

C#

```
public class RunningStateChangedEventArgs : EventArgs;
```

Visual Basic

```
Public Class RunningStateChangedEventArgs
    Inherits EventArgs
```

Inheritance Hierarchy

```
RunningStateChangedEventArgs
```

1.39.1 RunningStateChangedEventArgs Members

The following tables list the members exposed by RunningStateChangedEventArgs.

Public Properties

	Name	Description
	RunningState (see page 102)	Gets the new RunningState of the device.

1.39.2 RunningStateChangedEventArgs Properties

The properties of the RunningStateChangedEventArgs class are listed here.

Public Properties

	Name	Description
	RunningState (see page 102)	Gets the new RunningState of the device.

1.39.2.1 RunningStateChangedEventArgs.RunningState Property

Gets the new RunningState of the device.

C#

```
public DeviceRunningState RunningState;
```

Visual Basic

```
Public Property RunningState() As DeviceRunningState
```

Index

A

- AnalogInput class 2
 - about AnalogInput class 2
 - AnalogInput members 2
 - AnalogInput properties 3
- AnalogInput.Application property 3
- AnalogInput.ApplyCharacteristic property 3
- AnalogInput.Index property 3
- AnalogInput.IOType property 4
- AnalogInput.Name property 4
- AnalogInput.Parent property 4
- AnalogInput.Unit property 4
- AnalogInput.Value property 4
- AnalogInputChangedEventArgs class 5
 - about AnalogInputChangedEventArgs class 5
 - AnalogInputChangedEventArgs members 5
 - AnalogInputChangedEventArgs properties 5
- AnalogInputChangedEventArgs.AnalogInput property 5
- AnalogInputs class 6
 - about AnalogInputs class 6
 - AnalogInputs members 6
 - AnalogInputs methods 6
 - AnalogInputs properties 7
 - this indexer 7
- AnalogInputs.Application property 8
- AnalogInputs.Count property 8
- AnalogInputs.HardwareInstalled property 8
- AnalogInputs.Parent property 8
- AnalogInputs.this indexer 7
- AnalogOutput class 9
 - about AnalogOutput class 9
 - AnalogOutput members 9
 - AnalogOutput properties 9
- AnalogOutput.Application property 10
- AnalogOutput.Index property 10
- AnalogOutput.IOType property 10
- AnalogOutput.Name property 10
- AnalogOutput.Parent property 10
- AnalogOutput.Value property 11
- AnalogOutputChangedEventArgs class 11
 - about AnalogOutputChangedEventArgs class 11
 - AnalogOutputChangedEventArgs members 11
 - AnalogOutputChangedEventArgs properties 11
- AnalogOutputChangedEventArgs.AnalogOutput property 12
- AnalogOutputs class 12
 - about AnalogOutputs class 12
 - AnalogOutputs members 12
 - AnalogOutputs methods 13
 - AnalogOutputs properties 14
 - this indexer 13
- AnalogOutputs.Application property 14
- AnalogOutputs.Count property 14
- AnalogOutputs.HardwareInstalled property 14
- AnalogOutputs.Parent property 14
- AnalogOutputs.this indexer 13
- Application class 15
 - about Application class 15
 - Application members 15
 - Application methods 16
 - Application properties 18
 - CompressDevice method 16
- Application.AutoApplyCalibrations property 18
- Application.CompressAllDevices method 16
- Application.CompressDevice method 16, 17
- Application.Delay method 17
- Application.Devices property 18
- Application.MeasureProjects property 19
- Application.Parent property 19
- Application.PopupMessageManager property 19
- Application.SendDeviceMessage method 17
- Application.UnitsConverter property 19
- Application.Version property 19
- Application.WorkspaceHeight property 20
- Application.WorkspaceWidth property 20
- Attached enumeration member 58
- Attaching enumeration member 58

C

- Connected enumeration member 59
- Connecting enumeration member 59

D

- DataFile enumeration member 92
- Default enumeration member 65
- Detached enumeration member 58
- Detaching enumeration member 58
- Device class 20
 - about Device class 20
 - Device events 47
 - Device members 20
 - Device methods 23
 - Device properties 43
 - NewMeasureProject method 39
 - Start method 43
- Device.AnalogInputChanged event 48
- Device.AnalogInputs property 44
- Device.AnalogOutputChanged event 48
- Device.AnalogOutputs property 44
- Device.Application property 44
- Device.AttachedState property 45
- Device.ClearMSAMasses method 25
- Device.ClearTotalPressure method 25
- Device.Connect method 25
- Device.ConnectedState property 45
- Device.DeviceAttachedStateChanged event 49
- Device.DeviceConnectionStateChanged event 49
- Device.DeviceError event 50
- Device.DeviceFilamentStateChanged event 50
- Device.DeviceIP property 45
- Device.DeviceMacAddress property 45
- Device.DevicePath property 46
- Device.DeviceRecipeChanged event 50
- Device.DeviceSemStateChanged event 51
- Device.DeviceTPValueChanged event 51
- Device.DigitalInputChanged event 52
- Device.DigitalInputs property 46
- Device.DigitalOutputChanged event 52
- Device.DigitalOutputs property 46
- Device.Disconnect method 25
- Device.DoCalibrationGSC method 25
- Device.DoCalibrationMSC method 26
- Device.DoCalibrationOffset method 27
- Device.DoCalibrationZeroGas method 27
- Device.DoGetActiveFilament method 28
- Device.DoGetIonSourceActiveParameterSet method 28
- Device.DoGetIonSourceV method 28
- Device.DoGetIonSourceV1 method 28
- Device.DoGetIonSourceV2 method 29
- Device.DoGetIonSourceV3 method 29
- Device.DoGetIonSourceV4 method 29
- Device.DoGetIonSourceV5 method 29
- Device.DoGetIonSourceVoltage method 30
- Device.DoMSA method 30
- Device.DoSetActiveFilament method 31
- Device.DoSetEmissionCurrent method 31
- Device.DoSetIonSourceActiveParameterSet method 31
- Device.DoSetIonSourceV method 32
- Device.DoSetIonSourceV1 method 32
- Device.DoSetIonSourceV2 method 32
- Device.DoSetIonSourceV3 method 33
- Device.DoSetIonSourceV4 method 33
- Device.DoSetIonSourceV5 method 33
- Device.DoSetIonSourceVoltage method 34
- Device.GetCommonSemVoltage method 34
- Device.GetEPIOffsetValueRange05 method 34
- Device.GetEPIOffsetValueRange06 method 34
- Device.GetEPIOffsetValueRange07 method 35
- Device.GetEPIOffsetValueRange08 method 35
- Device.GetEPIOffsetValueRange09 method 35
- Device.GetEPIOffsetValueRange10 method 35
- Device.GetEPIOffsetValueRange11 method 36
- Device.GetEPIOffsetValueRange12 method 36
- Device.GetFilamentState method 36
- Device.GetFineTuneMasses method 36
- Device.GetMSAMeasuredMasses method 37
- Device.GetMSAStatus method 37
- Device.GetOPCItem method 37
- Device.GetSemState method 37
- Device.GetTotalPressureState method 37
- Device.GetTotalPressureValue method 38
- Device.IO method 38
- Device.IsSimulator property 46
- Device.MSAActualMasses method 38
- Device.Name property 47

- Device.NewMeasureProject method 39
- Device.Parent property 47
- Device.PartialPressureAvailable property 47
- Device.ResetFineTuneActualMasses method 40
- Device.RunningState property 47
- Device.RunningStateChanged event 52
- Device.SetCommonSemVoltage method 40
- Device.SetFilamentState method 40
- Device.SetFineTuneActualMasses method 40
- Device.SetSemState method 42
- Device.SetTotalPressure method 42
- Device.SetTotalPressureState method 42
- Device.Start method 43
- Device.Stop method 43
- Device.Type property 47
- DeviceAttachedState enumeration 58
- DeviceAttachedStateChangedEventArgs class 58
 - about DeviceAttachedStateChangedEventArgs class 58
 - DeviceAttachedStateChangedEventArgs members 59
 - DeviceAttachedStateChangedEventArgs properties 59
- DeviceAttachedStateChangedEventArgs.AttachedState property 59
- DeviceConnectedState enumeration 59
- DeviceConnectionStateChangedEventArgs class 60
 - about DeviceConnectionStateChangedEventArgs class 60
 - DeviceConnectionStateChangedEventArgs members 60
 - DeviceConnectionStateChangedEventArgs properties 61
- DeviceConnectionStateChangedEventArgs.ConnectedState property 61
- DeviceError class 53
 - about DeviceError class 53
 - DeviceError members 53
 - DeviceError properties 54
- DeviceError enumeration member 61
- DeviceError.Code property 54
- DeviceError.Description property 54
- DeviceError.Type property 54
- DeviceErrorEventArgs class 54
 - about DeviceErrorEventArgs class 54
 - DeviceErrorEventArgs members 55
 - DeviceErrorEventArgs properties 55
- DeviceErrorEventArgs.DeviceError property 55
- DeviceErrorType enumeration 61
- DeviceFilamentState enumeration 62
- DeviceFilamentStateChangedEventArgs class 62
 - about DeviceFilamentStateChangedEventArgs class 62
 - DeviceFilamentStateChangedEventArgs members 62
 - DeviceFilamentStateChangedEventArgs properties 63
- DeviceFilamentStateChangedEventArgs.FilamentState property 63
- DeviceRunningState enumeration 63
- Devices class 55
 - about Devices class 55
 - Devices members 56
 - Devices methods 56
 - Devices properties 57
 - this indexer 56
- Devices.Application property 57
- Devices.Count property 57
- Devices.Parent property 58
- Devices.this indexer 56, 57
- DeviceSemState enumeration 64
- DeviceSemStateChangedEventArgs class 64
 - about DeviceSemStateChangedEventArgs class 64
 - DeviceSemStateChangedEventArgs members 64
 - DeviceSemStateChangedEventArgs properties 65
- DeviceSemStateChangedEventArgs.SemState property 65
- DeviceStartMode enumeration 65
- DeviceTotalPressureState enumeration 66
- DeviceTPValueChangedEventArgs class 66
 - about DeviceTPValueChangedEventArgs class 66
 - DeviceTPValueChangedEventArgs members 67
 - DeviceTPValueChangedEventArgs properties 67
- DeviceTPValueChangedEventArgs.Value property 67
- DeviceType enumeration 67
- DeviceWarning enumeration member 61
- DigitalInput class 68
 - about DigitalInput class 68
 - DigitalInput members 68
 - DigitalInput properties 68
- DigitalInput.Application property 69
- DigitalInput.Index property 69
- DigitalInput.IOType property 69
- DigitalInput.Name property 69
- DigitalInput.Parent property 69

[DigitalInput.Value property](#) 70
[DigitalInputChangedEventArgs class](#) 70
 about [DigitalInputChangedEventArgs class](#) 70
 [DigitalInputChangedEventArgs members](#) 70
 [DigitalInputChangedEventArgs properties](#) 71
[DigitalInputChangedEventArgs.DigitalInput property](#) 71
[DigitalInputs class](#) 71
 about [DigitalInputs class](#) 71
 [DigitalInputs members](#) 71
 [DigitalInputs methods](#) 72
 [DigitalInputs properties](#) 73
 [this indexer](#) 72
[DigitalInputs.Application property](#) 73
[DigitalInputs.Count property](#) 73
[DigitalInputs.HardwareInstalled property](#) 73
[DigitalInputs.Parent property](#) 74
[DigitalInputs.this indexer](#) 72
[DigitalOutput class](#) 74
 about [DigitalOutput class](#) 74
 [DigitalOutput members](#) 74
 [DigitalOutput properties](#) 74
[DigitalOutput.Application property](#) 75
[DigitalOutput.Index property](#) 75
[DigitalOutput.IOType property](#) 75
[DigitalOutput.Name property](#) 75
[DigitalOutput.Parent property](#) 76
[DigitalOutput.Value property](#) 76
[DigitalOutputChangedEventArgs class](#) 76
 about [DigitalOutputChangedEventArgs class](#) 76
 [DigitalOutputChangedEventArgs members](#) 76
 [DigitalOutputChangedEventArgs properties](#) 77
[DigitalOutputChangedEventArgs.DigitalOutput property](#) 77
[DigitalOutputs class](#) 77
 about [DigitalOutputs class](#) 77
 [DigitalOutputs members](#) 77
 [DigitalOutputs methods](#) 78
 [DigitalOutputs properties](#) 79
 [this indexer](#) 78
[DigitalOutputs.Application property](#) 79
[DigitalOutputs.Count property](#) 79
[DigitalOutputs.HardwareInstalled property](#) 79
[DigitalOutputs.Parent property](#) 80

[DigitalOutputs.this indexer](#) 78
[DigitalOutputs.Values property](#) 80
[Disconnected enumeration member](#) 59
[Disconnecting enumeration member](#) 59

I

[IO class](#) 80
 about [IO class](#) 80
 [IO members](#) 80
 [IO properties](#) 81
[IO.AnalogInputs property](#) 81
[IO.AnalogOutputs property](#) 81
[IO.DigitalInputs property](#) 82
[IO.DigitalOutputs property](#) 82
[IO.IsInstalled property](#) 82
[IO.Type property](#) 82
[IO220 enumeration member](#) 82
[IO700_1 enumeration member](#) 82
[IO700_2 enumeration member](#) 82
[IO700_3 enumeration member](#) 82
[IO720 enumeration member](#) 82
[IOType enumeration](#) 82

M

[MeasurementTask class](#) 92
 about [MeasurementTask class](#) 92
 [GetMeasureTaskStatistics method](#) 94
 [MeasurementTask members](#) 93
 [MeasurementTask methods](#) 93
 [MeasurementTask properties](#) 97
[MeasurementTask.Application property](#) 97
[MeasurementTask.ApplyBackgroundSubtraction property](#) 97
[MeasurementTask.ApplyCharacteristic property](#) 98
[MeasurementTask.Cycles property](#) 98
[MeasurementTask.GetCycleData method](#) 93
[MeasurementTask.GetLastMeasureValue method](#) 94
[MeasurementTask.GetMeasureTaskStatistics method](#) 95, 96
[MeasurementTask.Index property](#) 98
[MeasurementTask.Name property](#) 98
[MeasurementTask.Parent property](#) 98
[MeasurementTask.TaskID property](#) 99
[MeasurementTask.TaskType property](#) 99

MeasurementTask.TaskUnit property 99

MeasurementTask.UsePartialPressure property 99

MeasurementTasks class 99

- about MeasurementTasks class 99
- MeasurementTasks members 100
- MeasurementTasks methods 100
- MeasurementTasks properties 101

MeasurementTasks.Application property 101

MeasurementTasks.Count property 101

MeasurementTasks.Parent property 101

MeasurementTasks.this indexer 100

MeasureProject class 83

- about MeasureProject class 83
- Export method 85
- MeasureProject members 83
- MeasureProject methods 84
- MeasureProject properties 86

MeasureProject.Activate method 84

MeasureProject.Application property 87

MeasureProject.Backup method 84

MeasureProject.Close method 85

MeasureProject.Device property 87

MeasureProject.Export method 85

MeasureProject.Height property 87

MeasureProject.Left property 87

MeasureProject.MeasurementTasks property 87

MeasureProject.MeasureProjectRecipes property 88

MeasureProject.Name property 88

MeasureProject.Parent property 88

MeasureProject.SetBounds method 86

MeasureProject.Top property 88

MeasureProject.Type property 88

MeasureProject.Width property 89

MeasureProjects class 89

- about MeasureProjects class 89
- MeasureProjects members 89
- MeasureProjects methods 90
- MeasureProjects properties 91
- this indexer 90

MeasureProjects.Application property 91

MeasureProjects.Count property 91

MeasureProjects.Open method 90

MeasureProjects.Parent property 92

MeasureProjects.this indexer 90, 91

MeasureProjectType enumeration 92

N

NotReady enumeration member 62, 64

O

Off enumeration member 62, 64, 66

On enumeration member 62, 64, 66

OverRange enumeration member 66

Q

QMG220 enumeration member 67

QMG700 enumeration member 67

Quadera.AnalogInput 2

Quadera.AnalogInput.Application 3

Quadera.AnalogInput.ApplyCharacteristic 3

Quadera.AnalogInput.Index 3

Quadera.AnalogInput.IOType 4

Quadera.AnalogInput.Name 4

Quadera.AnalogInput.Parent 4

Quadera.AnalogInput.Unit 4

Quadera.AnalogInput.Value 4

Quadera.AnalogInputChangedEventArgs 5

Quadera.AnalogInputChangedEventArgs.AnalogInput 5

Quadera.AnalogInputs 6

Quadera.AnalogInputs.Application 8

Quadera.AnalogInputs.Count 8

Quadera.AnalogInputs.HardwareInstalled 8

Quadera.AnalogInputs.Item 7

Quadera.AnalogInputs.Parent 8

Quadera.AnalogOutput 9

Quadera.AnalogOutput.Application 10

Quadera.AnalogOutput.Index 10

Quadera.AnalogOutput.IOType 10

Quadera.AnalogOutput.Name 10

Quadera.AnalogOutput.Parent 10

Quadera.AnalogOutput.Value 11

Quadera.AnalogOutputChangedEventArgs 11

Quadera.AnalogOutputChangedEventArgs.AnalogOutput 12

Quadera.AnalogOutputs 12

Quadera.AnalogOutputs.Application 14
Quadera.AnalogOutputs.Count 14
Quadera.AnalogOutputs.HardwareInstalled 14
Quadera.AnalogOutputs.Item 13
Quadera.AnalogOutputs.Parent 14
Quadera.Application 15
Quadera.Application.AutoApplyCalibrations 18
Quadera.Application.CompressAllDevices 16
Quadera.Application.CompressDevice 16, 17
Quadera.Application.Delay 17
Quadera.Application.Devices 18
Quadera.Application.MeasureProjects 19
Quadera.Application.Parent 19
Quadera.Application.SendDeviceMessage 17
Quadera.Application.Version 19
Quadera.Application.WorkspaceHeight 20
Quadera.Application.WorkspaceWidth 20
Quadera.Device 20
Quadera.Device.AnalogInputChanged 48
Quadera.Device.AnalogInputs 44
Quadera.Device.AnalogOutputChanged 48
Quadera.Device.AnalogOutputs 44
Quadera.Device.Application 44
Quadera.Device.AttachedState 45
Quadera.Device.ClearTotalPressure 25
Quadera.Device.Connect 25
Quadera.Device.ConnectedState 45
Quadera.Device.DeviceAttachedStateChanged 49
Quadera.Device.DeviceConnectionStateChanged 49
Quadera.Device.DeviceError 50
Quadera.Device.DeviceFilamentStateChanged 50
Quadera.Device.DeviceIP 45
Quadera.Device.DeviceMacAddress 45
Quadera.Device.DeviceRecipeChanged 50
Quadera.Device.DeviceSemStateChanged 51
Quadera.Device.DeviceTPValueChanged 51
Quadera.Device.DigitalInputChanged 52
Quadera.Device.DigitalInputs 46
Quadera.Device.DigitalOutputChanged 52
Quadera.Device.DigitalOutputs 46
Quadera.Device.Disconnect 25
Quadera.Device.DoCalibrationGSC 25
Quadera.Device.DoCalibrationMSC 26
Quadera.Device.DoCalibrationOffset 27
Quadera.Device.DoCalibrationZeroGas 27
Quadera.Device.DoGetActiveFilament 28
Quadera.Device.DoGetIonSourceActiveParameterSet 28
Quadera.Device.DoGetIonSourceV 28
Quadera.Device.DoGetIonSourceV1 28
Quadera.Device.DoGetIonSourceV2 29
Quadera.Device.DoGetIonSourceV3 29
Quadera.Device.DoGetIonSourceV4 29
Quadera.Device.DoGetIonSourceV5 29
Quadera.Device.DoGetIonSourceVoltage 30
Quadera.Device.DoMSA 30
Quadera.Device.DoSetActiveFilament 31
Quadera.Device.DoSetIonSourceActiveParameterSet 31
Quadera.Device.DoSetIonSourceV 32
Quadera.Device.DoSetIonSourceV1 32
Quadera.Device.DoSetIonSourceV2 32
Quadera.Device.DoSetIonSourceV3 33
Quadera.Device.DoSetIonSourceV4 33
Quadera.Device.DoSetIonSourceV5 33
Quadera.Device.DoSetIonSourceVoltage 34
Quadera.Device.GetCommonSemVoltage 34
Quadera.Device.GetEPIOffsetValueRange05 34, 35, 36
Quadera.Device.GetFilamentState 36
Quadera.Device.GetFineTuneMasses 36
Quadera.Device.GetMSAMeasuredMasses 37
Quadera.Device.GetMSAStatus 37
Quadera.Device.GetOPCItem 37
Quadera.Device.GetSemState 37
Quadera.Device.GetTotalPressureState 37
Quadera.Device.GetTotalPressureValue 38
Quadera.Device.IO 38
Quadera.Device.IsSimulator 46
Quadera.Device.MSAActualMasses 38
Quadera.Device.Name 47
Quadera.Device.NewMeasureProject 39
Quadera.Device.Parent 47
Quadera.Device.PartialPressureAvailable 47
Quadera.Device.ResetFineTuneActualMasses 40
Quadera.Device.RunningState 47
Quadera.Device.RunningStateChanged 52

Quadera.Device.SetCommonSemVoltage 40	Quadera.DigitalInput.Index 69
Quadera.Device.SetFilamentState 40	Quadera.DigitalInput.IOType 69
Quadera.Device.SetFineTuneActualMasses 40	Quadera.DigitalInput.Name 69
Quadera.Device.SetSemState 42	Quadera.DigitalInput.Parent 69
Quadera.Device.SetTotalPressure 42	Quadera.DigitalInput.Value 70
Quadera.Device.SetTotalPressureState 42	Quadera.DigitalInputChangedEventArgs 70
Quadera.Device.Start 43	Quadera.DigitalInputChangedEventArgs.DigitalInput 71
Quadera.Device.Stop 43	Quadera.DigitalInputs 71
Quadera.Device.Type 47	Quadera.DigitalInputs.Application 73
Quadera.DeviceAttachedState 58	Quadera.DigitalInputs.Count 73
Quadera.DeviceAttachedStateChangedEventArgs 58, 64	Quadera.DigitalInputs.HardwareInstalled 73
Quadera.DeviceAttachedStateChangedEventArgs.AttachedState 59	Quadera.DigitalInputs.Item 72
Quadera.DeviceConnectedState 59	Quadera.DigitalInputs.Parent 74
Quadera.DeviceConnectionStateChangedEventArgs 60	Quadera.DigitalOutput 74
Quadera.DeviceConnectionStateChangedEventArgs.ConnectionState 61	Quadera.DigitalOutput.Application 75
Quadera.DeviceError 53	Quadera.DigitalOutput.Index 75
Quadera.DeviceError.Code 54	Quadera.DigitalOutput.IOType 75
Quadera.DeviceError.Description 54	Quadera.DigitalOutput.Name 75
Quadera.DeviceError.Type 54	Quadera.DigitalOutput.Parent 76
Quadera.DeviceErrorEventArgs 54	Quadera.DigitalOutput.Value 76
Quadera.DeviceErrorEventArgs.DeviceError 55	Quadera.DigitalOutputChangedEventArgs 76
Quadera.DeviceErrorType 61	Quadera.DigitalOutputChangedEventArgs.DigitalOutput 77
Quadera.DeviceFilamentState 62	Quadera.DigitalOutputs 77
Quadera.DeviceFilamentStateChangedEventArgs 62	Quadera.DigitalOutputs.Application 79
Quadera.DeviceFilamentStateChangedEventArgs.FilamentState 63	Quadera.DigitalOutputs.Count 79
Quadera.DeviceRunningState 63	Quadera.DigitalOutputs.HardwareInstalled 79
Quadera.Devices 55	Quadera.DigitalOutputs.Item 78
Quadera.Devices.Application 57	Quadera.DigitalOutputs.Parent 80
Quadera.Devices.Count 57	Quadera.DigitalOutputs.Values 80
Quadera.Devices.Parent 58	Quadera.IO 80
Quadera.DeviceSemState 64	Quadera.IO.AnalogInputs 81
Quadera.DeviceSemStateChangedEventArgs.SemState 65	Quadera.IO.AnalogOutputs 81
Quadera.DeviceStartMode 65	Quadera.IO.DigitalInputs 82
Quadera.DeviceTotalPressureState 66	Quadera.IO.DigitalOutputs 82
Quadera.DeviceTPValueChangedEventArgs 66	Quadera.IO.IsInstalled 82
Quadera.DeviceTPValueChangedEventArgs.SemState 67	Quadera.IO.Type 82
Quadera.DeviceType 67	Quadera.IOType 82
Quadera.DigitalInput 68	Quadera.MeasurementTask 92
Quadera.DigitalInput.Application 69	Quadera.MeasurementTask.Application 97
	Quadera.MeasurementTask.ApplyCharacteristic 98
	Quadera.MeasurementTask.Cycles 98
	Quadera.MeasurementTask.GetCycleData 93

Quadera.MeasurementTask.GetLastMeasureValue 94
Quadera.MeasurementTask.GetMeasureTaskStatistics 95, 96
Quadera.MeasurementTask.Index 98
Quadera.MeasurementTask.Name 98
Quadera.MeasurementTask.Parent 98
Quadera.MeasurementTask.TaskID 99
Quadera.MeasurementTask.TaskUnit 99
Quadera.MeasurementTask.Type 99
Quadera.MeasurementTask.UsePartialPressure 99
Quadera.MeasurementTasks 99
Quadera.MeasurementTasks.Application 101
Quadera.MeasurementTasks.Count 101
Quadera.MeasurementTasks.Item 100
Quadera.MeasurementTasks.Parent 101
Quadera.MeasureProject 83
Quadera.MeasureProject.Activate 84
Quadera.MeasureProject.Application 87
Quadera.MeasureProject.Backup 84
Quadera.MeasureProject.Close 85
Quadera.MeasureProject.Device 87
Quadera.MeasureProject.Export 85
Quadera.MeasureProject.Height 87
Quadera.MeasureProject.Left 87
Quadera.MeasureProject.MeasurementTasks 87
Quadera.MeasureProject.MeasureProjectRecipes 88
Quadera.MeasureProject.Name 88
Quadera.MeasureProject.Parent 88
Quadera.MeasureProject.SetBounds 86
Quadera.MeasureProject.Top 88
Quadera.MeasureProject.Type 88
Quadera.MeasureProject.Width 89
Quadera.MeasureProjects 89
Quadera.MeasureProjects.Application 91
Quadera.MeasureProjects.Count 91
Quadera.MeasureProjects.Item 90, 91
Quadera.MeasureProjects.Open 90
Quadera.MeasureProjects.Parent 92
Quadera.MeasureProjectType 92
Quadera.RunningStateChangedEventArgs 101
Quadera.RunningStateChangedEventArgs.RunningState 102

R

Running enumeration member 63
RunningStateChangedEventArgs class 101
 about RunningStateChangedEventArgs class 101
 RunningStateChangedEventArgs members 102
 RunningStateChangedEventArgs properties 102
RunningStateChangedEventArgs.RunningState property 102

S

SensorError enumeration member 66
SingleCycle enumeration member 65
Starting enumeration member 63
Stopped enumeration member 63
Stopping enumeration member 63
Symbol Reference 1

T

Template enumeration member 92

U

UnderRange enumeration member 66