

 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PPJoy Namespace

Classes

	Class	Description
	AxisMapping	An AxisMapping object represents an axis control that is defined on a PPJoy joystick Device . An AxisMapping describes the axis' number (index), the type of axis control defined, and the PPJoy AxisDataSource from which the AxisMapping receives the values which it reports to Windows.
	ButtonMapping	A ButtonMapping object represents a button control that is defined on a PPJoy joystick Device . A ButtonMapping defines the button's number (index) and the ButtonDataSource that will provide the ButtonMapping 's state values that it will report to Windows.
	ContinuousPovMapping	A ContinuousPovMapping object represents a specific type of Point-of-View (Pov) control that can be defined on a PPJoy joystick Device . A ContinuousPovMapping defines the Pov's number (index) and the ContinuousPovDataSource that will provide the ContinuousPovMapping 's state values that it will report to Windows.
	Device	A Device object (an <i>instance</i> of the Device class) represents a single PPJoy joystick Device .
	DeviceAlreadyExistsException	Represents an error that occurs when trying to create a PPJoy Device that already exists.

	DeviceManager	Provides methods for creating, retrieving, deleting, and managing details of PPJoy Device objects.
	DeviceNotFoundException	Represents an error that occurs when trying to obtain a reference to a PPJoy Device that does not exist.
	DirectionalPovMapping	A DirectionalPovMapping object represents a specific type of Point-of-View (Pov) control that can be defined on a PPJoy joystick Device . A DirectionalPovMapping defines the Pov's number (index) and the DirectionalPovDataSources that will provide the DirectionalPovMapping 's state values that it will report to Windows.
	Mapping	Mapping is the base class for all Mapping types. A Mapping represents a control on a PPJoy Virtual Joystick Device such as a Point-of-View hat, an axis, or a button. Mappings declare the presence of a specific control, and its position (index) among other controls of the same type on the same Device . Mappings also define the data sources that feed these virtual controls state information, which, in turn, is reported to Windows and is accessible via DirectInput.
	MappingCollection	A MappingCollection is a specialized collection that can store related Mapping objects together. This provides for ease of handling when many different Mappings must be manipulated as a group. A MappingCollection exposes several sub-collections, from which all Mappings of a particular Type can be

retrieved (for instance, all [ButtonMappings](#) in the [MappingCollection](#) can be retrieved from the [ButtonMappings](#) property).

- ◆ [OperationFailedException](#)
Represents an error that occurs when performing a PPJoy IOCTL operation.
- ◆ [PovMapping](#)
[PovMapping](#) is the base class for all [PovMapping](#) types. A [PovMapping](#) is a type of [Mapping](#) that declares and defines a Point-of-View control on a PPJoy Virtual Joystick [Device](#).
- ◆ [PPJoyException](#)
[PPJoyException](#) is the base class Exception for all custom Exceptions that occur within the PPJoy wrapper.
- ◆ [VirtualJoystick](#)
A [VirtualJoystick](#) provides an easy-to-use interface for setting the PPJoy data source states for a single PPJoy Virtual Joystick [Device](#).

Enumerations

Enumeration	Description
 AxisDataSources	<p>AxisDataSources defines an enumeration of PPJoy data sources that can be used with AxisMapping objects.</p>
 AxisTypes	<p>AxisTypes defines an enumeration of Windows axis types that can be reported to Windows by an AxisMapping defined on a PPJoy device. The AxisType associated with an AxisMapping defines how the axis will be labelled by Windows.</p>
 ButtonDataSources	<p>ButtonDataSources defines an enumeration of PPJoy data sources that can be used with ButtonMapping objects.</p>
 ContinuousPovDataSources	<p>ContinuousPovDataSources defines an enumeration of PPJoy data sources that can be used with ContinuousPovMapping objects.</p>
 DirectionalPovDataSources	<p>DirectionalPovDataSources defines an enumeration of PPJoy data sources that can be used with DirectionalPovMapping objects.</p>
 JoystickMapScope	<p>JoystickMapScope defines an enumeration of scopes to which a MappingCollection can be applied.</p>
 JoystickSubTypes	<p>JoystickSubTypes defines an enumeration of all possible PPJoy joystick sub-types.</p>
 JoystickTypes	<p>JoystickTypes defines an enumeration of all possible PPJoy joystick types. A joystick type is a combination of controller type and interface type, as shown in the PPJoy Control Panel.</p>

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisDataSources Enumeration

[See Also](#)

AxisDataSources defines an enumeration of PPJoy data sources that can be used with [AxisMapping](#) objects.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Enumeration AxisDataSources
```

C#

```
public enum AxisDataSources
```

Visual C++

```
public enum class AxisDataSources
```

Members

Member name Description

Digital0

Digital1

Digital2

Digital3

Digital4

Digital5

Digital6

Digital7

Digital8

Digital9

Digital10

Digital11

Digital12

Digital13

Digital14

Digital15

Digital16

Digital17

Digital18

Digital19

Digital20

Digital21

Digital22

Digital23

Digital24

Digital25

Digital26

Digital27

Digital28
Digital29
Digital30
Digital31
Digital32
Digital33
Digital34
Digital35
Digital36
Digital37
Digital38
Digital39
Digital40
Digital41
Digital42
Digital43
Digital44
Digital45
Digital46
Digital47
Digital48
Digital49
Digital50
Digital51
Digital52
Digital53
Digital54
Digital55
Digital56
Digital57
Digital58
Digital59
Digital60

Digital61
Digital62
Digital63
Digital64
Digital65
Digital66
Digital67
Digital68
Digital69
Digital70
Digital71
Digital72
Digital73
Digital74
Digital75
Digital76
Digital77
Digital78
Digital79
Digital80
Digital81
Digital82
Digital83
Digital84
Digital85
Digital86
Digital87
Digital88
Digital89
Digital90
Digital91
Digital92
Digital93

Digital94
Digital95
Digital96
Digital97
Digital98
Digital99
Digital100
Digital101
Digital102
Digital103
Digital104
Digital105
Digital106
Digital107
Digital108
Digital109
Digital110
Digital111
Digital112
Digital113
Digital114
Digital115
Digital116
Digital117
Digital118
Digital119
Digital120
Digital121
Digital122
Digital123
Digital124
Digital125
Digital126

Digital127

Analog0

Analog1

Analog2

Analog3

Analog4

Analog5

Analog6

Analog7

Analog8

Analog9

Analog10

Analog11

Analog12

Analog13

Analog14

Analog15

Analog16

Analog17

Analog18

Analog19

Analog20

Analog21

Analog22

Analog23

Analog24

Analog25

Analog26

Analog27

Analog28

Analog29

Analog30

Analog31

Analog32
Analog33
Analog34
Analog35
Analog36
Analog37
Analog38
Analog39
Analog40
Analog41
Analog42
Analog43
Analog44
Analog45
Analog46
Analog47
Analog48
Analog49
Analog50
Analog51
Analog52
Analog53
Analog54
Analog55
Analog56
Analog57
Analog58
Analog59
Analog60
Analog61
Analog62
Analog63
Reversed0

Reversed1
Reversed2
Reversed3
Reversed4
Reversed5
Reversed6
Reversed7
Reversed8
Reversed9
Reversed10
Reversed11
Reversed12
Reversed13
Reversed14
Reversed15
Reversed16
Reversed17
Reversed18
Reversed19
Reversed20
Reversed21
Reversed22
Reversed23
Reversed24
Reversed25
Reversed26
Reversed27
Reversed28
Reversed29
Reversed30
Reversed31
Reversed32
Reversed33

Reversed34
Reversed35
Reversed36
Reversed37
Reversed38
Reversed39
Reversed40
Reversed41
Reversed42
Reversed43
Reversed44
Reversed45
Reversed46
Reversed47
Reversed48
Reversed49
Reversed50
Reversed51
Reversed52
Reversed53
Reversed54
Reversed55
Reversed56
Reversed57
Reversed58
Reversed59
Reversed60
Reversed61
Reversed62
None

▪ See Also

[PPJoy Namespace](#)

[PPJoy...:::AxisMapping](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisMapping Class

[Members](#) [See Also](#)

An AxisMapping object represents an axis control that is defined on a PPJoy joystick [Device](#). An AxisMapping describes the axis' number (index), the type of axis control defined, and the PPJoy [AxisDataSource](#) from which the AxisMapping receives the values which it reports to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class AxisMapping _
    Inherits Mapping
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class AxisMapping : Mapping
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class AxisMapping sealed : public Mapping
```

▀ Inheritance Hierarchy

System...:::Object

[PPJoy...:::Mapping](#)

PPJoy...:::AxisMapping

See Also

[AxisMapping Members](#)

[PPJoy Namespace](#)

[PPJoy...::Device](#)

  

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

AxisMapping Constructor

[AxisMapping Class](#) [See Also](#)

Overload List

Name	Description
AxisMapping()	Creates a new AxisMapping object.
AxisMapping(Int32)	Creates a new AxisMapping object.

See Also

[AxisMapping Class](#)

[AxisMapping Members](#)

[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

AxisMapping Constructor

[AxisMapping Class](#) [See Also](#)

Creates a new [AxisMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public AxisMapping()
```

Visual C++

```
public:  
AxisMapping()
```

See Also

[AxisMapping Class](#)

[AxisMapping Overload](#)

[PPJoy Namespace](#)

[PPJoy...::Device](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisMapping Constructor (Int32)

[AxisMapping Class](#) [See Also](#)

Creates a new [AxisMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    controlNumber As Integer _  
)
```

C#

```
public AxisMapping(  
    int controlNumber  
)
```

Visual C++

```
public:  
AxisMapping(  
    int controlNumber  
)
```

Parameters

controlNumber

Type: System..:::Int32

The zero-based index of this [AxisMapping](#) in the collection of [AxisMappings](#) defined on the same PPJoy [Device](#). For example, the first [AxisMapping](#) in the collection will have a controlNumber of 0; the second [AxisMapping](#) will have a controlNumber of 1; and so forth.

See Also

[AxisMapping Class](#)
[AxisMapping Overload](#)
[PPJoy Namespace](#)
[Mapping...:::ControlNumber](#)
[PPJoy...:::Device](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

AxisMapping Members

[AxisMapping Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [AxisMapping](#) type exposes the following members.

Constructors

Name	Description
= ♡ AxisMapping	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	AxisType	Gets/sets the type of Windows axis that this AxisMapping will report itself as.
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	MaxDataSource	Gets/sets the PPJoy AxisDataSource that this AxisMapping will use as the source of the values that it will report to Windows.
	MinDataSource	Gets/sets the PPJoy AxisDataSource that this AxisMapping will use as the source of the values that it will report to Windows.

See Also

[AxisMapping Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

AxisMapping Properties

[AxisMapping Class](#) [See Also](#)

The [AxisMapping](#) type exposes the following properties.

Properties

	Name	Description
	AxisType	Gets/sets the type of Windows axis that this AxisMapping will report itself as.
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	MaxDataSource	Gets/sets the PPJoy AxisDataSource that this AxisMapping will use as the source of the values that it will report to Windows.
	MinDataSource	Gets/sets the PPJoy AxisDataSource that this AxisMapping will use as the source of the values that it will report to Windows.

See Also

[AxisMapping Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisMapping...::AxisType Property

[AxisMapping Class](#) [See Also](#)

Gets/sets the type of Windows axis that this [AxisMapping](#) will report itself as.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property AxisType As AxisTypes
```

C#

```
public AxisTypes AxisType { get; set; }
```

Visual C++

```
public:  
property AxisTypes AxisType {  
    AxisTypes get ();  
    void set (AxisTypes value);  
}
```

▪ See Also

[AxisMapping Class](#)

[PPJoy Namespace](#)

[PPJoy...:::AxisTypes](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisMapping...:::MaxDataSource Property

[AxisMapping Class](#) [See Also](#)

Gets/sets the PPJoy [AxisDataSource](#) that this [AxisMapping](#) will use as the source of the values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property MaxDataSource As AxisDataSources
```

C#

```
public AxisDataSources MaxDataSource { get; set; }
```

Visual C++

```
public:  
property AxisDataSources MaxDataSource {  
    AxisDataSources get ();  
    void set (AxisDataSources value);  
}
```

Remarks

When this property is set to a Digital [AxisDataSource](#), then this [AxisMapping](#) will report its **maximum** value to Windows whenever the Digital [AxisDataSource](#)'s value is trueTruetrue (True in Visual Basic).

To use the MaxDataSource property, you must first set the [MinDataSource](#) property to a Digital (boolean) [AxisDataSource](#).

Note: If the [MinDataSource](#) property is not set to a Digital (boolean) [AxisDataSource](#), then setting the MaxDataSource property has no effect.

See Also

[AxisMapping Class](#)

[PPJoy Namespace](#)

[AxisMapping...:::MinDataSource](#)

[PPJoy...:::AxisDataSources](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisMapping...:::MinDataSource Property

[AxisMapping Class](#) [See Also](#)

Gets/sets the PPJoy [AxisDataSource](#) that this [AxisMapping](#) will use as the source of the values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property MinDataSource As AxisDataSources
```

C#

```
public AxisDataSources MinDataSource { get; set; }
```

Visual C++

```
public:  
property AxisDataSources MinDataSource {  
    AxisDataSources get ();  
    void set (AxisDataSources value);  
}
```

Remarks

When the MinDataSource property is set to a Digital [AxisDataSource](#), then this [AxisMapping](#) will report its **minimum** value to Windows whenever the Digital [AxisDataSource](#)'s value is trueTruetrue (True in Visual Basic).

When this property is set to an Analog or Reversed [AxisDataSource](#), then this [AxisMapping](#) will report the value provided by the Analog or Reversed [AxisDataSource](#) assigned to the MinDataSource property, and will ignore any [AxisDataSource](#) set in the [MaxDataSource](#) property.

Details: PPJoy [AxisMappings](#) can be driven from either Digital or Analog [AxisDataSources](#).

Digital [AxisDataSources](#):

If an [AxisMapping](#) is driven by a (pair of) **Digital**[AxisDataSources](#), then the [AxisMapping](#) can only report one of two possible values -- **Minimum** and **Maximum**. How this works is as follows:

If the value of the **Digital** data source assigned to the [AxisMapping](#)'s MinDataSource property is trueTruetrue (True in Visual Basic), then the [AxisMapping](#) will report its value as being the **minimum** axis value.

Alternatively, if the value of the **Digital** data source assigned to the [AxisMapping](#)'s [MaxDataSource](#) property is trueTruetrue (True in Visual Basic), then the [AxisMapping](#) will report its value as being the **maximum** axis value. If **both** the MinDataSource and the [MaxDataSource](#)'s values read the same value (either trueTruetrue (True in Visual Basic) or falseFalsefalsefalse (False in Visual Basic), the behavior is undefined.

Note: If the MinDataSource property is set to a **Digital**[AxisDataSource](#), then the [MaxDataSource](#) property should also be set to a **Digital**[AxisDataSource](#). You cannot set one data source to Digital and the other to Analog, nor should you set the MinDataSource property without also setting the [MaxDataSource](#) property.

Analog/Reversed [AxisDataSource](#):

If an [AxisMapping](#) is driven by an [Analog](#) or [ReversedAxisDataSource](#), then the [AxisMapping](#) will report its value to Windows, based on the value of the underlying Analog or Reversed [AxisDataSource](#) which is assigned to the [AxisMapping](#)'s MinDataSource property.

For Analog [AxisDataSources](#), when the value of the underlying [AxisDataSource](#) increases, the value reported by the [AxisMapping](#) to Windows will increase proportionately.

For [ReversedAxisDataSources](#), when the value of the underlying [AxisDataSource](#) decreases, the value reported by the [AxisMapping](#) to Windows will *increase* proportionately.

Note: If the MinDataSource property is set to an [AnalogAxisDataSource](#) or a [ReversedAxisDataSource](#), then the value of the [MaxDataSource](#) property will be ignored.

See Also

[AxisMapping Class](#)

[PPJoy Namespace](#)

[AxisMapping...::MaxDataSource](#)

[PPJoy...::AxisDataSources](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

AxisTypes Enumeration

[See Also](#)

AxisTypes defines an enumeration of Windows axis types that can be reported to Windows by an [AxisMapping](#) defined on a PPJoy device. The AxisType associated with an [AxisMapping](#) defines how the axis will be labelled by Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Enumeration AxisTypes
```

C#

```
public enum AxisTypes
```

Visual C++

```
public enum class AxisTypes
```

Members

Member name Description

Unknown

X

Y

Z

XRotation

YRotation

ZRotation

Slider

Pov

Dial

Wheel

VX

VY

VZ

VBRX

VBRY

VBRZ

Rudder

Throttle

See Also

[PPJoy Namespace](#)

[PPJoy...::AxisMapping](#)

[PPJoy...::PovMapping](#)

[PPJoy...::ContinuousPovMapping](#)

[PPJoy...::DirectionalPovMapping](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ButtonDataSources Enumeration

[See Also](#)

ButtonDataSources defines an enumeration of PPJoy data sources that can be used with [ButtonMapping](#) objects.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Enumeration ButtonDataSources
```

C#

```
public enum ButtonDataSources
```

Visual C++

```
public enum class ButtonDataSources
```

Members

Member name Description

Digital0

Digital1

Digital2

Digital3

Digital4

Digital5

Digital6

Digital7

Digital8

Digital9

Digital10

Digital11

Digital12

Digital13

Digital14

Digital15

Digital16

Digital17

Digital18

Digital19

Digital20

Digital21

Digital22

Digital23

Digital24

Digital25

Digital26

Digital27

Digital28
Digital29
Digital30
Digital31
Digital32
Digital33
Digital34
Digital35
Digital36
Digital37
Digital38
Digital39
Digital40
Digital41
Digital42
Digital43
Digital44
Digital45
Digital46
Digital47
Digital48
Digital49
Digital50
Digital51
Digital52
Digital53
Digital54
Digital55
Digital56
Digital57
Digital58
Digital59
Digital60

Digital61
Digital62
Digital63
Digital64
Digital65
Digital66
Digital67
Digital68
Digital69
Digital70
Digital71
Digital72
Digital73
Digital74
Digital75
Digital76
Digital77
Digital78
Digital79
Digital80
Digital81
Digital82
Digital83
Digital84
Digital85
Digital86
Digital87
Digital88
Digital89
Digital90
Digital91
Digital92
Digital93

Digital94
Digital95
Digital96
Digital97
Digital98
Digital99
Digital100
Digital101
Digital102
Digital103
Digital104
Digital105
Digital106
Digital107
Digital108
Digital109
Digital110
Digital111
Digital112
Digital113
Digital114
Digital115
Digital116
Digital117
Digital118
Digital119
Digital120
Digital121
Digital122
Digital123
Digital124
Digital125
Digital126

Digital127
Analog0Min
Analog1Min
Analog2Min
Analog3Min
Analog4Min
Analog5Min
Analog6Min
Analog7Min
Analog8Min
Analog9Min
Analog10Min
Analog11Min
Analog12Min
Analog13Min
Analog14Min
Analog15Min
Analog16Min
Analog17Min
Analog18Min
Analog19Min
Analog20Min
Analog21Min
Analog22Min
Analog23Min
Analog24Min
Analog25Min
Analog26Min
Analog27Min
Analog28Min
Analog29Min
Analog30Min
Analog31Min

Analog32Min
Analog33Min
Analog34Min
Analog35Min
Analog36Min
Analog37Min
Analog38Min
Analog39Min
Analog40Min
Analog41Min
Analog42Min
Analog43Min
Analog44Min
Analog45Min
Analog46Min
Analog47Min
Analog48Min
Analog49Min
Analog50Min
Analog51Min
Analog52Min
Analog53Min
Analog54Min
Analog55Min
Analog56Min
Analog57Min
Analog58Min
Analog59Min
Analog60Min
Analog61Min
Analog62Min
Analog63Min
Analog0Max

Analog1Max
Analog2Max
Analog3Max
Analog4Max
Analog5Max
Analog6Max
Analog7Max
Analog8Max
Analog9Max
Analog10Max
Analog11Max
Analog12Max
Analog13Max
Analog14Max
Analog15Max
Analog16Max
Analog17Max
Analog18Max
Analog19Max
Analog20Max
Analog21Max
Analog22Max
Analog23Max
Analog24Max
Analog25Max
Analog26Max
Analog27Max
Analog28Max
Analog29Max
Analog30Max
Analog31Max
Analog32Max
Analog33Max

Analog34Max
Analog35Max
Analog36Max
Analog37Max
Analog38Max
Analog39Max
Analog40Max
Analog41Max
Analog42Max
Analog43Max
Analog44Max
Analog45Max
Analog46Max
Analog47Max
Analog48Max
Analog49Max
Analog50Max
Analog51Max
Analog52Max
Analog53Max
Analog54Max
Analog55Max
Analog56Max
Analog57Max
Analog58Max
Analog59Max
Analog60Max
Analog61Max
Analog62Max
None

▪ See Also

[PPJoy Namespace](#)

[PPJoy...:::ButtonMapping](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ButtonMapping Class

[Members](#) [See Also](#)

A ButtonMapping object represents a button control that is defined on a PPJoy joystick [Device](#). A ButtonMapping defines the button's number (index) and the [ButtonDataSource](#) that will provide the ButtonMapping's state values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class ButtonMapping _
    Inherits Mapping
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class ButtonMapping : Mapping
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class ButtonMapping sealed : public Mapping
```

▀ Inheritance Hierarchy

System..:::Object

[PPJoy..:::Mapping](#)

PPJoy..:::ButtonMapping

See Also

[ButtonMapping Members](#)

[PPJoy Namespace](#)

[PPJoy...::Device](#)

[PPJoy...::Mapping](#)

[PPJoy...::ButtonDataSources](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

ButtonMapping Constructor

[ButtonMapping Class](#) [See Also](#)

Overload List

	Name	Description
•	ButtonMapping(0)	Creates a new ButtonMapping object.
•	ButtonMapping(Int32)	Creates a new ButtonMapping object.

See Also

[ButtonMapping Class](#)

[ButtonMapping Members](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ButtonMapping Constructor

[ButtonMapping Class](#) [See Also](#)

Creates a new [ButtonMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public ButtonMapping()
```

Visual C++

```
public:  
ButtonMapping()
```

See Also

[ButtonMapping Class](#)
[ButtonMapping Overload](#)
[PPJoy Namespace](#)
[PPJoy...::Device](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ButtonMapping Constructor (Int32)

[ButtonMapping Class](#) [See Also](#)

Creates a new [ButtonMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    controlNumber As Integer _  
)
```

C#

```
public ButtonMapping(  
    int controlNumber  
)
```

Visual C++

```
public:  
ButtonMapping(  
    int controlNumber  
)
```

Parameters

controlNumber

Type: System..:::Int32

The zero-based index of this [ButtonMapping](#) in the collection of [ButtonMappings](#) defined on the same PPJoy [Device](#). For example, the first [ButtonMapping](#) in the collection will have a controlNumber of 0, the second [ButtonMapping](#) will have a controlNumber of 1, and so forth.

See Also

[ButtonMapping Class](#)
[ButtonMapping Overload](#)
[PPJoy Namespace](#)
[Mapping...:::ControlNumber](#)
[PPJoy...:::Device](#)

≡ ⌂ ⌂

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

ButtonMapping Members

[ButtonMapping Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [ButtonMapping](#) type exposes the following members.

Constructors

Name	Description
= ♡ ButtonMapping	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	DataSource	Gets/sets the PPJoy ButtonDataSource that this ButtonMapping will use as the source of the values that it will report to Windows.

▪ See Also

[ButtonMapping Class](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

ButtonMapping Properties

[ButtonMapping Class](#) [See Also](#)

The [ButtonMapping](#) type exposes the following properties.

Properties

	Name	Description
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	DataSource	Gets/sets the PPJoy ButtonDataSource that this ButtonMapping will use as the source of the values that it will report to Windows.

▪ See Also

[ButtonMapping Class](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ButtonMapping....:::DataSource Property

[ButtonMapping Class](#) [See Also](#)

Gets/sets the PPJoy [ButtonDataSource](#) that this [ButtonMapping](#) will use as the source of the values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property DataSource As ButtonDataSources
```

C#

```
public ButtonDataSources DataSource { get; set; }
```

Visual C++

```
public:  
property ButtonDataSources DataSource {  
    ButtonDataSources get ();  
    void set (ButtonDataSources value);  
}
```

Remarks

When this property is set to a **Digital** [ButtonDataSource](#), then this [ButtonMapping](#) will report a value of **pressed** to Windows whenever the Digital [ButtonDataSource](#)'s value is trueTruetrue (True in Visual Basic). Similarly, a value of **unpressed** will be reported, whenever the Digital [ButtonDataSource](#)'s value is falseFalsefalsefalse (False in Visual Basic).

When this property is set to an **Analog-Max** [ButtonDataSource](#), then this [ButtonMapping](#) will report a value of **pressed** to Windows whenever the Analog [ButtonDataSource](#)'s value is at its **maximum** value. Similarly, this [ButtonMapping](#) will report a value of **unpressed** to Windows whenever the Analog [ButtonDataSource](#)'s value is at any other value other than its **maximum**.

When this property is set to an **Analog-Min** [ButtonDataSource](#), then this [ButtonMapping](#) will report a value of **pressed** to Windows whenever the Analog [ButtonDataSource](#)'s value is at its **minimum**. Similarly, this [ButtonMapping](#) will report a value of **unpressed** to Windows whenever the Analog [ButtonDataSource](#)'s value is at any other value other than its **minimum**.

See Also

[ButtonMapping Class](#)
[PPJoy Namespace](#)
[PPJoy...::ButtonDataSources](#)
[PPJoy...::ButtonMapping](#)
[PPJoy...::Mapping](#)
[PPJoy...::Device](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ContinuousPovDataSources Enumeration

[See Also](#)

ContinuousPovDataSources defines an enumeration of PPJoy data sources that can be used with [ContinuousPovMapping](#) objects.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Enumeration ContinuousPovDataSources
```

C#

```
public enum ContinuousPovDataSources
```

Visual C++

```
public enum class ContinuousPovDataSources
```

Members

Member name	Description
Analog0	
Analog1	
Analog2	
Analog3	
Analog4	
Analog5	
Analog6	
Analog7	
Analog8	
Analog9	
Analog10	
Analog11	
Analog12	
Analog13	
Analog14	
Analog15	
Analog16	
Analog17	
Analog18	
Analog19	
Analog20	
Analog21	
Analog22	
Analog23	
Analog24	
Analog25	
Analog26	
Analog27	

Analog28
Analog29
Analog30
Analog31
Analog32
Analog33
Analog34
Analog35
Analog36
Analog37
Analog38
Analog39
Analog40
Analog41
Analog42
Analog43
Analog44
Analog45
Analog46
Analog47
Analog48
Analog49
Analog50
Analog51
Analog52
Analog53
Analog54
Analog55
Analog56
Analog57
Analog58
Analog59
Analog60

Analog61
Analog62
Analog63
Reversed0
Reversed1
Reversed2
Reversed3
Reversed4
Reversed5
Reversed6
Reversed7
Reversed8
Reversed9
Reversed10
Reversed11
Reversed12
Reversed13
Reversed14
Reversed15
Reversed16
Reversed17
Reversed18
Reversed19
Reversed20
Reversed21
Reversed22
Reversed23
Reversed24
Reversed25
Reversed26
Reversed27
Reversed28
Reversed29

Reversed30
Reversed31
Reversed32
Reversed33
Reversed34
Reversed35
Reversed36
Reversed37
Reversed38
Reversed39
Reversed40
Reversed41
Reversed42
Reversed43
Reversed44
Reversed45
Reversed46
Reversed47
Reversed48
Reversed49
Reversed50
Reversed51
Reversed52
Reversed53
Reversed54
Reversed55
Reversed56
Reversed57
Reversed58
Reversed59
Reversed60
Reversed61
Reversed62

| None

See Also

[PPJoy Namespace](#)

[PPJoy...:::ContinuousPovMapping](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ContinuousPovMapping Class

[Members](#) [See Also](#)

A ContinuousPovMapping object represents a specific type of Point-of-View (Pov) control that can be defined on a PPJoy joystick [Device](#). A ContinuousPovMapping defines the Pov's number (index) and the [ContinuousPovDataSource](#) that will provide the ContinuousPovMapping's state values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class ContinuousPovMapping _
    Inherits PovMapping
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class ContinuousPovMapping : PovMapping
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class ContinuousPovMapping sealed : public PovMapping
```

■ Remarks

A ContinuousPovMapping sources its values from a single Analog or Reversed [ContinuousPovDataSource](#).

Contrast this behavior with that of a [DirectionalPovMapping](#) control, which sources its values from a set of Digital [DirectionalPovDataSources](#).

▀ Inheritance Hierarchy

System...:::Object

[PPJoy...:::Mapping](#)

[PPJoy...:::PovMapping](#)

PPJoy....:::ContinuousPovMapping

See Also

[ContinuousPovMapping Members](#)
[PPJoy Namespace](#)
[PPJoy...:::ContinuousPovDataSources](#)
[PPJoy...:::DirectionalPovMapping](#)
[PPJoy...:::DirectionalPovDataSources](#)
[PPJoy...:::PovMapping](#)
[PPJoy...:::Device](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

ContinuousPovMapping Constructor

[ContinuousPovMapping Class](#) [See Also](#)

Overload List

	Name	Description
•	ContinuousPovMapping()	Creates a new ContinuousPovMapping object.
•	ContinuousPovMapping(Int32)	Creates a new ContinuousPovMapping object.

See Also

[ContinuousPovMapping Class](#)
[ContinuousPovMapping Members](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ContinuousPovMapping Constructor

[ContinuousPovMapping Class](#) [See Also](#)

Creates a new [ContinuousPovMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public ContinuousPovMapping()
```

Visual C++

```
public:  
ContinuousPovMapping()
```

See Also

[ContinuousPovMapping Class](#)
[ContinuousPovMapping Overload](#)
[PPJoy Namespace](#)
[PPJoy...::PovMapping](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ContinuousPovMapping Constructor (Int32)

[ContinuousPovMapping Class](#) [See Also](#)

Creates a new [ContinuousPovMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    controlNumber As Integer _  
)
```

C#

```
public ContinuousPovMapping(  
    int controlNumber  
)
```

Visual C++

```
public:  
ContinuousPovMapping(  
    int controlNumber  
)
```

Parameters

controlNumber

Type: System..:::Int32

The zero-based index of this [PovMapping](#) in the collection of [PovMappings](#) defined on the same PPJoy [Device](#). For example, the first [PovMapping](#) in the collection will have a controlNumber of 0, the second [PovMapping](#) will have a controlNumber of 1, and so forth.

See Also

[ContinuousPovMapping Class](#)
[ContinuousPovMapping Overload](#)
[PPJoy Namespace](#)
[Mapping...:::ControlNumber](#)
[PPJoy...:::PovMapping](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

ContinuousPovMapping Members

[ContinuousPovMapping Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [ContinuousPovMapping](#) type exposes the following members.

▪ Constructors

	Name	Description
≡ ♡	ContinuousPovMapping	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	DataSource	Gets/sets the PPJoy ContinuousPovDataSource that this PovMapping will use as the source of the values that it will report to Windows.

See Also

[ContinuousPovMapping Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

ContinuousPovMapping Properties

[ContinuousPovMapping Class](#) [See Also](#)

The [ContinuousPovMapping](#) type exposes the following properties.

Properties

	Name	Description
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	DataSource	Gets/sets the PPJoy ContinuousPovDataSource that this PovMapping will use as the source of the values that it will report to Windows.

See Also

[ContinuousPovMapping Class](#)
[PPJoy Namespace](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

ContinuousPovMapping..:::DataSource Property

[ContinuousPovMapping Class](#) [See Also](#)

Gets/sets the PPJoy [ContinuousPovDataSource](#) that this [PovMapping](#) will use as the source of the values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property DataSource As ContinuousPovDataSources
```

C#

```
public ContinuousPovDataSources DataSource { get; set; }
```

Visual C++

```
public:  
property ContinuousPovDataSources DataSource {  
    ContinuousPovDataSources get ();  
    void set (ContinuousPovDataSources value);  
}
```

Remarks

When an **Analog** [ContinuousPovDataSource](#) is assigned, then, as the [ContinuousPovDataSource](#)'s value increases, the values reported to Windows by this [PovMapping](#) will increase proportionately, proceeding clock-wise from **North**.

When a **Reversed** [ContinuousPovDataSource](#) is assigned, the value of the Reversed [ContinuousPovDataSource](#) itself will *decrease* as the value of the corresponding Analog [ContinuousPovDataSource](#) *increases*. This, in turn, means that the value reported to Windows by this [PovMapping](#) will *decrease*, as the value of the corresponding Analog [ContinuousPovDataSource](#) *increases*.

Note: When the assigned [ContinuousPovDataSource](#)'s value is set to -1, this [PovMapping](#) will report its position as **centered**.

See Also

[ContinuousPovMapping Class](#)
[PPJoy Namespace](#)
[PPJoy...::ContinuousPovDataSources](#)
[PPJoy...::PovMapping](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device Class

[Members](#) [See Also](#)

A Device **object** (an *instance* of the Device **class**) represents a single PPJoy joystick Device.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class Device
```

C#

```
[Serializable]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class Device
```

Visual C++

```
[Serializable]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class Device sealed
```

▀ Inheritance Hierarchy

System...:::Object

PPJoy...:::Device

See Also

[Device Members](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Device Members

[Device Class](#) [Methods](#) [Properties](#) [See Also](#)

The [Device](#) type exposes the following members.

Methods

	Name	Description
≡	Delete	Deletes this Device from PPJoy.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetMappings	Overloaded.
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	RemoveMappings	Overloaded.
≡	SetMappings	Overloaded.
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

Name	Description
 DeviceType	Gets the JoystickType of this Device .
 LptNum	Gets the LPT number of this Device .
 ProductId	Gets the Product ID associated with this Device .
 SubType	Gets the JoystickSubType of this Device .
 UnitNum	Gets the unit number of this Device .
 VendorId	Gets the Vendor ID associated with this Device .

▪ See Also

[Device Class](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Device Methods

[Device Class](#) [See Also](#)

The [Device](#) type exposes the following methods.

Methods

	Name	Description
≡	Delete	Deletes this Device from PPJoy.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetMappings	Overloaded.
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	RemoveMappings	Overloaded.
≡	SetMappings	Overloaded.
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

▪ See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device..:::Delete Method

[Device Class](#) [See Also](#)

Deletes this [Device](#) from PPJoy.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub Delete ( _  
    removeDirectInput As Boolean, _  
    removeDriver As Boolean _  
)
```

C#

```
public void Delete(  
    bool removeDirectInput,  
    bool removeDriver  
)
```

Visual C++

```
public:  
void Delete(  
    bool removeDirectInput,  
    bool removeDriver  
)
```

Parameters

removeDirectInput

Type: System..:::Boolean

If trueTruetrue (True in Visual Basic), this [Device](#)'s registration will be removed from DirectInput. If falseFalsefalse (False in Visual Basic), the [Device](#)'s DirectInput registration will not be removed.

removeDriver

Type: System..:::Boolean

If trueTruetrue (True in Visual Basic), the [Device](#)'s drivers will be unregistered from the system. If falseFalsefalse (False in Visual Basic), the [Device](#)'s drivers will not be unregistered from the system.

See Also

[Device Class](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Device...:::GetMappings Method

[Device Class](#) [See Also](#)

Overload List

	Name	Description
•	GetMappings()	Gets a MappingCollection representing the controls defined on this Device .
•	GetMappings(JoystickMapScope)	Gets a MappingCollection representing the controls defined on this Device , or representing the controls defined in this Device 's interface.

See Also

[Device Class](#)

[Device Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::GetMappings Method

[Device Class](#) [See Also](#)

Gets a [MappingCollection](#) representing the controls defined on this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function GetMappings As MappingCollection
```

C#

```
public MappingCollection GetMappings()
```

Visual C++

```
public:  
MappingCollection^ GetMappings()
```

Return Value

A [MappingCollection](#) object representing the controls defined directly on this [Device](#).

See Also

[Device Class](#)

[GetMappings Overload](#)

[PPJoy Namespace](#)

[PPJoy...:::Mapping](#)

[PPJoy...:::MappingCollection](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::GetMappings Method (JoystickMapScope)

[Device Class](#) [See Also](#)

Gets a [MappingCollection](#) representing the controls defined on this [Device](#), or representing the controls defined in this [Device](#)'s interface.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function GetMappings ( _  
    scope As JoystickMapScope _  
) As MappingCollection
```

C#

```
public MappingCollection GetMappings(  
    JoystickMapScope scope  
)
```

Visual C++

```
public:  
MappingCollection^ GetMappings(  
    JoystickMapScope scope  
)
```

Parameters

scope

Type: [PPJoy..:::JoystickMapScope](#)

Scope from which to retrieve the [MappingCollection](#).

Return Value

If the scope argument is set to [Interface](#), then this method returns a [MappingCollection](#) object representing the controls defined in this [Device](#)'s interface.

If the scope argument is set to [Device](#), then this method returns a [MappingCollection](#) object representing the controls defined directly on this [Device](#) itself.

See Also

[Device Class](#)

[GetMappings Overload](#)

[PPJoy Namespace](#)

[PPJoy...:::Mapping](#)

[PPJoy...:::MappingCollection](#)

[PPJoy...:::JoystickMapScope](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Device...:::RemoveMappings Method

[Device Class](#) [See Also](#)

Overload List

	Name	Description
≡ ♡	RemoveMappings()	Removes the custom-defined Mappings from this Device , without affecting the Mappings defined in the Device 's interface.
≡ ♡	RemoveMappings(JoystickMapScope)	Removes the custom-defined Mappings from this Device OR from its interface.

See Also

[Device Class](#)

[Device Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::RemoveMappings Method

[Device Class](#) [See Also](#)

Removes the custom-defined [Mappings](#) from this [Device](#), without affecting the [Mappings](#) defined in the [Device](#)'s interface.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub RemoveMappings
```

C#

```
public void RemoveMappings()
```

Visual C++

```
public:  
void RemoveMappings()
```

See Also

[Device Class](#)

[RemoveMappings Overload](#)

[PPJoy Namespace](#)

[PPJoy...::JoystickMapScope](#)

[PPJoy...::MappingCollection](#)

[PPJoy...::Mapping](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::RemoveMappings Method (JoystickMapScope)

[Device Class](#) [See Also](#)

Removes the custom-defined [Mappings](#) from this [Device](#) OR from its interface.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub RemoveMappings ( _  
    scope As JoystickMapScope _  
)
```

C#

```
public void RemoveMappings(  
    JoystickMapScope scope  
)
```

Visual C++

```
public:  
void RemoveMappings(  
    JoystickMapScope scope  
)
```

Parameters

scope

Type: [PPJoy..:::JoystickMapScope](#)

The [JoystickMapScope](#) from which to remove all custom-defined [Mappings](#).

See Also

[Device Class](#)

[RemoveMappings Overload](#)

[PPJoy Namespace](#)

[PPJoy...:::MappingCollection](#)

[PPJoy...:::Mapping](#)

[PPJoy...:::JoystickMapScope](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Device...:::SetMappings Method

[Device Class](#) [See Also](#)

Overload List

	Name	Description
•	SetMappings(MappingCollection)	Associates a set of Mappings (a MappingCollection) with a specific PPJoy Device .
•	SetMappings(JoystickMapScope, MappingCollection)	Associates a set of Mappings (a MappingCollection) with a specific PPJoy Device in a specific JoystickMapScope .

See Also

[Device Class](#)

[Device Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::SetMappings Method (JoystickMapScope, MappingCollection)

[Device Class](#) [See Also](#)

Associates a set of [Mappings](#) (a [MappingCollection](#)) with a specific PPJoy [Device](#) in a specific [JoystickMapScope](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub SetMappings ( _  
    scope As JoystickMapScope, _  
    newMappings As MappingCollection _  
)
```

C#

```
public void SetMappings(  
    JoystickMapScope scope,  
    MappingCollection newMappings  
)
```

Visual C++

```
public:  
void SetMappings(  
    JoystickMapScope scope,  
    MappingCollection^ newMappings  
)
```

Parameters

scope

Type: [PPJoy..:::JoystickMapScope](#)

The [JoystickMapScope](#) in which to store the new custom [Mappings](#).

newMappings

Type: [PPJoy..:::MappingCollection](#)

A [MappingCollection](#) object containing the new [Mappings](#) to associate with the specified scope.

See Also

[Device Class](#)

[SetMappings Overload](#)

[PPJoy Namespace](#)

[PPJoy...::MappingCollection](#)

[PPJoy...::Mapping](#)

[PPJoy...::JoystickMapScope](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::SetMappings Method (MappingCollection)

[Device Class](#) [See Also](#)

Associates a set of [Mappings](#) (a [MappingCollection](#)) with a specific PPJoy [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub SetMappings ( _  
    newMappings As MappingCollection _  
)
```

C#

```
public void SetMappings(  
    MappingCollection newMappings  
)
```

Visual C++

```
public:  
void SetMappings(  
    MappingCollection^ newMappings  
)
```

Parameters

newMappings

Type: [PPJoy..:::MappingCollection](#)

A [MappingCollection](#) object containing the new [Mappings](#) to associate with the [Device](#).

See Also

[Device Class](#)

[SetMappings Overload](#)

[PPJoy Namespace](#)

[PPJoy...:::MappingCollection](#)

[PPJoy...:::Mapping](#)

[PPJoy...:::JoystickMapScope](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Device Properties

[Device Class](#) [See Also](#)

The [Device](#) type exposes the following properties.

Properties

Name	Description
 DeviceType	Gets the JoystickType of this Device .
 LptNum	Gets the LPT number of this Device .
 ProductId	Gets the Product ID associated with this Device .
 SubType	Gets the JoystickSubType of this Device .
 UnitNum	Gets the unit number of this Device .
 VendorId	Gets the Vendor ID associated with this Device .

▪ See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device..:::DeviceType Property

[Device Class](#) [See Also](#)

Gets the [JoystickType](#) of this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property DeviceType As JoystickTypes
```

C#

```
public JoystickTypes DeviceType { get; }
```

Visual C++

```
public:  
property JoystickTypes DeviceType {  
    JoystickTypes get ();  
}
```

See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device..:::LptNum Property

[Device Class](#) [See Also](#)

Gets the LPT number of this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property LptNum As Integer
```

C#

```
public int LptNum { get; }
```

Visual C++

```
public:  
property int LptNum {  
    int get ();  
}
```

Remarks

Virtual joystick [Device](#)s will have LptNum = 0.

See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::ProductId Property

[Device Class](#) [See Also](#)

Gets the Product ID associated with this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property ProductId As Integer
```

C#

```
public int ProductId { get; }
```

Visual C++

```
public:  
property int ProductId {  
    int get ();  
}
```

See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::SubType Property

[Device Class](#) [See Also](#)

Gets the [JoystickSubType](#) of this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property SubType As JoystickSubTypes
```

C#

```
public JoystickSubTypes SubType { get; }
```

Visual C++

```
public:  
property JoystickSubTypes SubType {  
    JoystickSubTypes get ();  
}
```

See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::UnitNum Property

[Device Class](#) [See Also](#)

Gets the unit number of this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property UnitNum As Integer
```

C#

```
public int UnitNum { get; }
```

Visual C++

```
public:  
property int UnitNum {  
    int get ();  
}
```

See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Device...:::VendorId Property

[Device Class](#) [See Also](#)

Gets the Vendor ID associated with this [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property VendorId As Integer
```

C#

```
public int VendorId { get; }
```

Visual C++

```
public:  
property int VendorId {  
    int get ();  
}
```

See Also

[Device Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Class

[Members](#) [See Also](#)

Represents an error that occurs when trying to create a PPJoy [Device](#) that already exists.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class DeviceAlreadyExistsException _
    Inherits PPJoyException
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class DeviceAlreadyExistsException : PPJoyException
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class DeviceAlreadyExistsException : public PPJoyException
```

▀ Inheritance Hierarchy

System...:::Object

System...:::Exception

System...:::ApplicationException

[PPJoy...:::PPJoyException](#)

PPJoy...:::DeviceAlreadyExistsException

See Also

[DeviceAlreadyExistsException Members](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Constructor

[DeviceAlreadyExistsException Class](#) [See Also](#)

Overload List

Name	Description
 DeviceAlreadyExistsException()	Initializes a new instance of the DeviceAlreadyExistsExcept class.
 DeviceAlreadyExistsException(String)	Initializes a new instance of the DeviceAlreadyExistsExcept class.
 DeviceAlreadyExistsException(SerializationInfo, StreamingContext)	Initializes a new instance of the DeviceAlreadyExistsExcept class.
 DeviceAlreadyExistsException(String, Exception)	Initializes a new instance of the DeviceAlreadyExistsExcept class.

See Also

[DeviceAlreadyExistsException Class](#)

[DeviceAlreadyExistsException Members](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Constructor

[DeviceAlreadyExistsException Class](#) [See Also](#)

Initializes a new instance of the [DeviceAlreadyExistsException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public DeviceAlreadyExistsException()
```

Visual C++

```
public:  
DeviceAlreadyExistsException()
```

See Also

[DeviceAlreadyExistsException Class](#)

[DeviceAlreadyExistsException Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Constructor (SerializationInfo,
StreamingContext)

[DeviceAlreadyExistsException Class](#) [See Also](#)

Initializes a new instance of the [DeviceAlreadyExistsException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected DeviceAlreadyExistsException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

Visual C++

```
protected:  
DeviceAlreadyExistsException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

Parameters

info

Type: System.Runtime.Serialization..:::SerializationInfo

The SerializationInfo that holds the serialized object data about the Exception being thrown.

context

Type: System.Runtime.Serialization..:::StreamingContext

The StreamingContext that contains contextual information about the source or destination.

See Also

[DeviceAlreadyExistsException Class](#)

[DeviceAlreadyExistsException Overload](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Constructor (String)

[DeviceAlreadyExistsException Class](#) [See Also](#)

Initializes a new instance of the [DeviceAlreadyExistsException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String _  
)
```

C#

```
public DeviceAlreadyExistsException(  
    string message  
)
```

Visual C++

```
public:  
DeviceAlreadyExistsException(  
    String^ message  
)
```

Parameters

message

Type: System..:::String

The message that describes the error.

See Also

[DeviceAlreadyExistsException Class](#)

[DeviceAlreadyExistsException Overload](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Constructor (String, Exception)

[DeviceAlreadyExistsException Class](#) [See Also](#)

Initializes a new instance of the [DeviceAlreadyExistsException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public DeviceAlreadyExistsException(  
    string message,  
    Exception inner  
)
```

Visual C++

```
public:  
DeviceAlreadyExistsException(  
    String^ message,  
    Exception^ inner  
)
```

Parameters

message

Type: System..:::String

The error message that explains the reason for the exception.

inner

Type: System..:::Exception

The exception that is the cause of the current exception, or nullNothingnullptr a null reference (Nothing in Visual Basic) if no inner exception is specified.

See Also

[DeviceAlreadyExistsException Class](#)

[DeviceAlreadyExistsException Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceAlreadyExistsException Members

[DeviceAlreadyExistsException Class](#) [Constructors](#) [Methods](#) [Properties](#)

[See Also](#)

The [DeviceAlreadyExistsException](#) type exposes the following members.

Constructors

	Name	Description
≡	DeviceAlreadyExistsException	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
≡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
≡	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
≡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional, user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

See Also

[DeviceAlreadyExistsException Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager Class

[Members](#) [See Also](#)

Provides methods for creating, retrieving, deleting, and managing details of PPJoy [Device](#) objects.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class DeviceManager
```

C#

```
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class DeviceManager
```

Visual C++

```
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class DeviceManager
```

▀ Inheritance Hierarchy

System...:::Object
PPJoy...:::DeviceManager

See Also

[DeviceManager Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager Constructor

[DeviceManager Class](#) [See Also](#)

Creates an instance of the [DeviceManager](#) class which can manage PPJoy

[Devices](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public DeviceManager()
```

Visual C++

```
public:  
DeviceManager()
```

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceManager Members

[DeviceManager Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [DeviceManager](#) type exposes the following members.

Constructors

	Name	Description
•	DeviceManager	Creates an instance of the DeviceManager class which can manage PPJoy Devices .

Methods

	Name	Description
≡	CreateDevice	Creates and registers a new joystick Device with PPJoy.
≡	DeleteAllDevices	Deletes all registered PPJoy joystick Devices .
≡	DeleteDevice	Deletes a Device from PPJoy.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetAllDevices	Enumerates all defined PPJoy Devices .
≡	GetDevice	Retrieves a Device object that matches the specified query parameters.
≡	GetDeviceByProductId	Gets a Device object representing the PPJoy device whose product ID matches the supplied Product ID.
≡	GetDeviceMappings	Gets a MappingCollection representing the controls mappings associated with a defined PPJoy device.
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
≡	IsVirtualDevice	Checks whether a given integer containing a VendorID/ProductID combination (typically obtained from DirectInput) refers to a virtual Device or a physical Device .

≡	MaxValidUnitNumber	Gets the maximum valid unit number for a given joystick type.
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	RemoveDeviceMappings	Removes the mappings from a specific PPJoy device or the default mappings from its interface.
≡	SetDeviceMappings	Sets the mappings for a specific PPJoy device.
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	IdealMappings	Gets a custom MappingCollection that defines the broadest possible set of controls that can be assigned to a PPJoy Device . The controls are pre-set to expose the maximum capabilities that a virtual joystick Device can express.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceManager Methods

[DeviceManager Class](#) [See Also](#)

The [DeviceManager](#) type exposes the following methods.

Methods

	Name	Description
≡	CreateDevice	Creates and registers a new joystick Device with PPJoy.
≡	DeleteAllDevices	Deletes all registered PPJoy joystick Devices .
≡	DeleteDevice	Deletes a Device from PPJoy.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetAllDevices	Enumerates all defined PPJoy Devices .
≡	GetDevice	Retrieves a Device object that matches the specified query parameters.
≡	GetDeviceByProductId	Gets a Device object representing the PPJoy device whose product ID matches the supplied Product ID.
≡	GetDeviceMappings	Gets a MappingCollection representing the controls mappings associated with a defined PPJoy device.
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
≡	IsVirtualDevice	Checks whether a given integer containing a VendorID/ProductID combination (typically obtained from DirectInput) refers to a virtual Device or a physical Device .

≡	MaxValidUnitNumber	Gets the maximum valid unit number for a given joystick type.
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	RemoveDeviceMappings	Removes the mappings from a specific PPJoy device or the default mappings from its interface.
≡	SetDeviceMappings	Sets the mappings for a specific PPJoy device.
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::CreateDevice Method

[DeviceManager Class](#) [See Also](#)

Creates and registers a new joystick [Device](#) with PPJoy.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub CreateDevice ( _  
    lptNum As Integer, _  
    joystickType As JoystickTypes, _  
    subType As JoystickSubTypes, _  
    unitNum As Integer _  
)
```

C#

```
public void CreateDevice(  
    int lptNum,  
    JoystickTypes joystickType,  
    JoystickSubTypes subType,  
    int unitNum  
)
```

Visual C++

```
public:  
void CreateDevice(  
    int lptNum,  
    JoystickTypes joystickType,  
    JoystickSubTypes subType,  
    int unitNum  
)
```

Parameters

lptNum

Type: System..:::Int32

LPT number of the [Device](#) to create/register.

joystickType

Type: [PPJoy..:::JoystickTypes](#)

[JoystickType](#) of the [Device](#) to create/register.

subType

Type: [PPJoy..:::JoystickSubTypes](#)

[JoystickSubType](#) of the [Device](#) to create/register.

unitNum

Type: System..:::Int32

Unit number of the [Device](#) to create/register.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::DeleteAllDevices Method

[DeviceManager Class](#) [See Also](#)

Deletes all registered PPJoy joystick [Devices](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub DeleteAllDevices ( _  
    removeDirectInput As Boolean, _  
    removeDriver As Boolean _  
)
```

C#

```
public void DeleteAllDevices(  
    bool removeDirectInput,  
    bool removeDriver  
)
```

Visual C++

```
public:  
void DeleteAllDevices(  
    bool removeDirectInput,  
    bool removeDriver  
)
```

Parameters

removeDirectInput

Type: System..:::Boolean

If trueTruetrue (True in Visual Basic), each [Device](#)'s registration will be removed from DirectInput. If falseFalsefalse (False in Visual Basic), no [Device](#)'s DirectInput registration will be removed.

removeDriver

Type: System..:::Boolean

If trueTruetrue (True in Visual Basic), each [Device](#)'s drivers will be unregistered from the system. If falseFalsefalse (False in Visual Basic), no [Device](#)'s drivers will be unregistered from the system.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::DeleteDevice Method

[DeviceManager Class](#) [See Also](#)

Deletes a [Device](#) from PPJoy.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub DeleteDevice ( _  
    device As Device, _  
    removeDirectInput As Boolean, _  
    removeDriver As Boolean _  
)
```

C#

```
public void DeleteDevice(  
    Device device,  
    bool removeDirectInput,  
    bool removeDriver  
)
```

Visual C++

```
public:  
void DeleteDevice(  
    Device^ device,  
    bool removeDirectInput,  
    bool removeDriver  
)
```

Parameters

device

Type: [PPJoy..::Device](#)

a [Device](#) to delete from PPJoy.

removeDirectInput

Type: System..::Boolean

If trueTruetrue (True in Visual Basic), the [Device](#)'s registration will be removed from DirectInput. If falseFalsefalse (False in Visual Basic), the [Device](#)'s DirectInput registration will not be removed.

`removeDriver`

Type: System..:::Boolean

If trueTruetrue (True in Visual Basic), the [Device](#)'s drivers will be unregistered from the system. If falseFalsefalse (False in Visual Basic), the [Device](#)'s drivers will not be unregistered from the system.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...:: GetAllDevices Method

[DeviceManager Class](#) [See Also](#)

Enumerates all defined PPJoy [Devices](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function GetAllDevices As Device\(\)
```

C#

```
public Device[] GetAllDevices()
```

Visual C++

```
public:  
array<Device>^ GetAllDevices()
```

Return Value

An array of [Device](#) objects, where each element in the array represents a single defined PPJoy [Device](#).

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::GetDevice Method

[DeviceManager Class](#) [See Also](#)

Retrieves a [Device](#) object that matches the specified query parameters.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function GetDevice ( _  
    lptNum As Integer, _  
    unitNum As Integer _  
) As Device
```

C#

```
public Device GetDevice(  
    int lptNum,  
    int unitNum  
)
```

Visual C++

```
public:  
Device^ GetDevice(  
    int lptNum,  
    int unitNum  
)
```

Parameters

lptNum

Type: System..:::Int32

LPT Port number of the [Device](#) to retrieve; set to 0 for Virtual Joystick [Devices](#).

unitNum

Type: System..:::Int32

Unit number of the [Device](#) to retrieve.

Return Value

A [Device](#) object matching the search criteria, or nullNothingnullptr a null

reference (Nothing in Visual Basic), if no matching [Device](#) is found.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::GetDeviceByProductId Method

[DeviceManager Class](#) [See Also](#)

Gets a [Device](#) object representing the PPJoy device whose product ID matches the supplied Product ID.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function GetDeviceByProductId ( _  
    productId As Integer _  
) As Device
```

C#

```
public Device GetDeviceByProductId(  
    int productId  
)
```

Visual C++

```
public:  
Device^ GetDeviceByProductId(  
    int productId  
)
```

Parameters

productId

Type: System..:::Int32

The Product ID of the [Device](#) to return.

Return Value

a [Device](#) object representing the PPJoy device whose Product ID matches the value supplied in the productId argument.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::GetDeviceMappings Method

[DeviceManager Class](#) [See Also](#)

Gets a MappingCollection representing the controls mappings associated with a defined PPJoy device.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function GetDeviceMappings ( _  
    lptNum As Integer, _  
    joystickType As JoystickTypes, _  
    unitNum As Integer, _  
    scope As JoystickMapScope _  
) As MappingCollection
```

C#

```
public MappingCollection GetDeviceMappings(  
    int lptNum,  
    JoystickTypes joystickType,  
    int unitNum,  
    JoystickMapScope scope  
)
```

Visual C++

```
public:  
MappingCollection^ GetDeviceMappings(  
    int lptNum,  
    JoystickTypes joystickType,  
    int unitNum,  
    JoystickMapScope scope  
)
```

Parameters

lptNum

Type: System..:::Int32

LPT number of the joystick whose mappings will be returned.

joystickType

Type: [PPJoy..:::JoystickTypes](#)

Type of joystick whose mappings will be returned.

unitNum

Type: System..:::Int32

Unit number of the device of the given type whose mappings will be returned.

scope

Type: [PPJoy..:::JoystickMapScope](#)

Scope to return mappings from.

Return Value

a MappingCollection object, where each element in the collection represents a single control mapping.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...:::IsVirtualDevice Method

[DeviceManager Class](#) [See Also](#)

Checks whether a given integer containing a VendorID/ProductID combination (typically obtained from DirectInput) refers to a virtual [Device](#) or a physical [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function IsVirtualDevice ( _  
    vendorIdentityProductId As Integer _  
) As Boolean
```

C#

```
public bool IsVirtualDevice(  
    int vendorIdentityProductId  
)
```

Visual C++

```
public:  
bool IsVirtualDevice(  
    int vendorIdentityProductId  
)
```

Parameters

vendorIdentityProductId

Type: System..:::Int32

A 32-bit integer containing a VendorID (in the high 16 bits) and a Product ID (in the low 16 bits), indicating a particular [Device](#) on the system.

Return Value

trueTrue (True in Visual Basic), if the [Device](#) matching the specified vendorIdentityProductId is a PPJoy virtual [Device](#), or falseFalse (False in Visual Basic) if it is a physical [Device](#).

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 🔍

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...:::MaxValidUnitNumber Method

[DeviceManager Class](#) [See Also](#)

Gets the maximum valid unit number for a given joystick type.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function MaxValidUnitNumber ( _  
    joystickType As JoystickTypes _  
) As Integer
```

C#

```
public int MaxValidUnitNumber(  
    JoystickTypes joystickType  
)
```

Visual C++

```
public:  
int MaxValidUnitNumber(  
    JoystickTypes joystickType  
)
```

Parameters

joystickType

Type: [PPJoy..:::JoystickTypes](#)

Joystick type to determine the maximum valid unit number for.

Return Value

The maximum valid unit number for the specified joystick type.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...:::RemoveDeviceMappings Method

[DeviceManager Class](#) [See Also](#)

Removes the mappings from a specific PPJoy device or the default mappings from its interface.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub RemoveDeviceMappings ( _  
    lptNum As Integer, _  
    joystickType As JoystickTypes, _  
    unitNum As Integer, _  
    scope As JoystickMapScope _  
)
```

C#

```
public void RemoveDeviceMappings(  
    int lptNum,  
    JoystickTypes joystickType,  
    int unitNum,  
    JoystickMapScope scope  
)
```

Visual C++

```
public:  
void RemoveDeviceMappings(  
    int lptNum,  
    JoystickTypes joystickType,  
    int unitNum,  
    JoystickMapScope scope  
)
```

Parameters

lptNum

Type: System..:::Int32

The LPT number of the device whose mappings or whose interface's mappings will be removed.

joystickType

Type: [PPJoy..:::JoystickTypes](#)

The type of the device whose mappings or whose interface's mappings will

be removed.

unitNum

Type: System..:::Int32

The unit number of the device whose mappings or whose interface's mappings will be removed.

scope

Type: [PPJoy..:::JoystickMapScope](#)

the scope of the mappings to remove (the device's, or the device's interface's default mappings.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...::SetDeviceMappings Method

[DeviceManager Class](#) [See Also](#)

Sets the mappings for a specific PPJoy device.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub SetDeviceMappings ( _  
    lptNum As Integer, _  
    joystickType As JoystickTypes, _  
    unitNum As Integer, _  
    scope As JoystickMapScope, _  
    newMappings As MappingCollection _  
)
```

C#

```
public void SetDeviceMappings(  
    int lptNum,  
    JoystickTypes joystickType,  
    int unitNum,  
    JoystickMapScope scope,  
    MappingCollection newMappings  
)
```

Visual C++

```
public:  
void SetDeviceMappings(  
    int lptNum,  
    JoystickTypes joystickType,  
    int unitNum,  
    JoystickMapScope scope,  
    MappingCollection^ newMappings  
)
```

Parameters

lptNum

Type: System..:::Int32

The LPT number of the device whose mappings will be set to the newly-supplied mappings.

`joystickType`

Type: [PPJoy..:::JoystickTypes](#)

The type of device whose mappings will be set to the newly-supplied mappings.

`unitNum`

Type: System..:::Int32

The unit number of the device whose mappings will be set.

`scope`

Type: [PPJoy..:::JoystickMapScope](#)

The scope in which to set the new mappings -- either for the device instance itself, or for the device's interface defaults for all devices of the same type that do not override those defaults.

`newMappings`

Type: [PPJoy..:::MappingCollection](#)

A MappingCollection object containing the new mappings to associate with this device or interface.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceManager Properties

[DeviceManager Class](#) [See Also](#)

The [DeviceManager](#) type exposes the following properties.

Properties

	Name	Description
	IdealMappings	Gets a custom MappingCollection that defines the broadest possible set of controls that can be assigned to a PPJoy Device . The controls are pre-set to expose the maximum capabilities that a virtual joystick Device can express.

See Also

[DeviceManager Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceManager...:::IdealMappings Property

[DeviceManager Class](#) [See Also](#)

Gets a custom [MappingCollection](#) that defines the broadest possible set of controls that can be assigned to a PPJoy [Device](#). The controls are pre-set to expose the maximum capabilities that a virtual joystick [Device](#) can express.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property IdealMappings As MappingCollection
```

C#

```
public MappingCollection IdealMappings { get; }
```

Visual C++

```
public:  
property MappingCollection^ IdealMappings {  
    MappingCollection^ get ();  
}
```

Return Value

A fully-loaded [MappingCollection](#) object that can be assigned to a [Device](#) using the [SetMappings\(MappingCollection\)](#) method.

Remarks

The [MappingCollection](#) that will be returned will define a control set that includes 8 axes, 32 buttons, and 2 POVs.

Each [ButtonMapping](#) will have its [DataSource](#) property pre-set to a [ButtonDataSource](#) that corresponds with the [ButtonMapping](#)'s [ControlNumber](#) property value, such that the #1 button in the collection will source its data from [Digital0](#); the #2 button will source its data from [Digital1](#); and so on.

Each [PovMapping](#) will be a [ContinuousPovMapping](#), and will have its [DataSource](#) property set to [Analog8](#) for Pov #1, and [Analog9](#) for Pov #2.

Each [AxisMapping](#) will have its [MinDataSource](#) property set to an [AxisDataSource](#) that corresponds with the [AxisMapping](#)'s [ControlNumber](#) property value, such that the #1 axis will source its data from [Analog0](#); the #2 button will source its data from [Analog1](#); and so on. Additionally, each [AxisMapping](#) will have its [AxisType](#) property set to an [AxisType](#) in such a way as to ensure that the defined [AxisMappings](#) will include a member of each of the following [AxisTypes](#):

- [X](#)
- [Y](#)
- [Z](#)
- [XRotation](#)
- [YRotation](#)
- [ZRotation](#)
- [Slider](#) - #1
- [Slider](#) - #2

See Also

[DeviceManager Class](#)

[PPJoy Namespace](#)

[PPJoy...::Device](#)

[PPJoy...::MappingCollection](#)

[PPJoy...::Mapping](#)

[PPJoy...::AxisTypes](#)

[PPJoy...::AxisDataSources](#)

[PPJoy...::AxisMapping](#)

[PPJoy...::ButtonMapping](#)

[PPJoy...::ButtonDataSources](#)

[PPJoy...::PovMapping](#)

[PPJoy...::ContinuousPovMapping](#)

[PPJoy...::ContinuousPovDataSources](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceNotFoundException Class

[Members](#) [See Also](#)

Represents an error that occurs when trying to obtain a reference to a PPJoy [Device](#) that does not exist.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class DeviceNotFoundException _
    Inherits PPJoyException
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class DeviceNotFoundException : PPJoyException
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class DeviceNotFoundException : public PPJoyException
```

■ Inheritance Hierarchy

System...:::Object

 System...:::Exception

 System...:::ApplicationException

[PPJoy...:::PPJoyException](#)

 PPJoy...:::DeviceNotFoundException

See Also

[DeviceNotFoundException Members](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceNotFoundException Constructor

[DeviceNotFoundException Class](#) [See Also](#)

Overload List

	Name	Description
•	DeviceNotFoundException()	Initializes a new instance of the DeviceNotFoundException class.
•	DeviceNotFoundException(String)	Initializes a new instance of the DeviceNotFoundException class.
•	DeviceNotFoundException(SerializationInfo, StreamingContext)	Initializes a new instance of the DeviceNotFoundException class.
•	DeviceNotFoundException(String, Exception)	Initializes a new instance of the DeviceNotFoundException class.

See Also

- [DeviceNotFoundException Class](#)
 - [DeviceNotFoundException Members](#)
 - [PPJoy Namespace](#)
-

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceNotFoundException Constructor

[DeviceNotFoundException Class](#) [See Also](#)

Initializes a new instance of the [DeviceNotFoundException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public DeviceNotFoundException()
```

Visual C++

```
public:  
DeviceNotFoundException()
```

See Also

[DeviceNotFoundException Class](#)
[DeviceNotFoundException Overload](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceNotFoundException Constructor (SerializationInfo, StreamingContext)

[DeviceNotFoundException Class](#) [See Also](#)

Initializes a new instance of the [DeviceNotFoundException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected DeviceNotFoundException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

Visual C++

```
protected:  
DeviceNotFoundException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

Parameters

info

Type: System.Runtime.Serialization..:::SerializationInfo

The SerializationInfo that holds the serialized object data about the Exception being thrown.

context

Type: System.Runtime.Serialization..:::StreamingContext

The StreamingContext that contains contextual information about the source or destination.

See Also

[DeviceNotFoundException Class](#)
[DeviceNotFoundException Overload](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceNotFoundException Constructor (String)

[DeviceNotFoundException Class](#) [See Also](#)

Initializes a new instance of the [DeviceNotFoundException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String _  
)
```

C#

```
public DeviceNotFoundException(  
    string message  
)
```

Visual C++

```
public:  
DeviceNotFoundException(  
    String^ message  
)
```

Parameters

message

Type: System..:::String

The message that describes the error.

See Also

[DeviceNotFoundException Class](#)
[DeviceNotFoundException Overload](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DeviceNotFoundException Constructor (String, Exception)

[DeviceNotFoundException Class](#) [See Also](#)

Initializes a new instance of the [DeviceNotFoundException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public DeviceNotFoundException(  
    string message,  
    Exception inner  
)
```

Visual C++

```
public:  
DeviceNotFoundException(  
    String^ message,  
    Exception^ inner  
)
```

Parameters

message

Type: System..:::String

The error message that explains the reason for the exception.

inner

Type: System..:::Exception

The exception that is the cause of the current exception, or nullNothingnullptr a null reference (Nothing in Visual Basic) if no inner exception is specified.

See Also

[DeviceNotFoundException Class](#)
[DeviceNotFoundException Overload](#)
[PPJoy Namespace](#)

≡ ⌂ ⌂

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DeviceNotFoundException Members

[DeviceNotFoundException Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [DeviceNotFoundException](#) type exposes the following members.

Constructors

	Name	Description
≡	DeviceNotFoundException	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
≡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
≡	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
≡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional, user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

See Also

[DeviceNotFoundException Class](#)

[PPJoy Namespace](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovDataSources Enumeration

[See Also](#)

DirectionalPovDataSources defines an enumeration of PPJoy data sources that can be used with [DirectionalPovMapping](#) objects.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Enumeration DirectionalPovDataSources
```

C#

```
public enum DirectionalPovDataSources
```

Visual C++

```
public enum class DirectionalPovDataSources
```

Members

Member name Description

Digital0

Digital1

Digital2

Digital3

Digital4

Digital5

Digital6

Digital7

Digital8

Digital9

Digital10

Digital11

Digital12

Digital13

Digital14

Digital15

Digital16

Digital17

Digital18

Digital19

Digital20

Digital21

Digital22

Digital23

Digital24

Digital25

Digital26

Digital27

Digital28
Digital29
Digital30
Digital31
Digital32
Digital33
Digital34
Digital35
Digital36
Digital37
Digital38
Digital39
Digital40
Digital41
Digital42
Digital43
Digital44
Digital45
Digital46
Digital47
Digital48
Digital49
Digital50
Digital51
Digital52
Digital53
Digital54
Digital55
Digital56
Digital57
Digital58
Digital59
Digital60

Digital61
Digital62
Digital63
Digital64
Digital65
Digital66
Digital67
Digital68
Digital69
Digital70
Digital71
Digital72
Digital73
Digital74
Digital75
Digital76
Digital77
Digital78
Digital79
Digital80
Digital81
Digital82
Digital83
Digital84
Digital85
Digital86
Digital87
Digital88
Digital89
Digital90
Digital91
Digital92
Digital93

Digital94
Digital95
Digital96
Digital97
Digital98
Digital99
Digital100
Digital101
Digital102
Digital103
Digital104
Digital105
Digital106
Digital107
Digital108
Digital109
Digital110
Digital111
Digital112
Digital113
Digital114
Digital115
Digital116
Digital117
Digital118
Digital119
Digital120
Digital121
Digital122
Digital123
Digital124
Digital125
Digital126

Digital127
Analog0Min
Analog1Min
Analog2Min
Analog3Min
Analog4Min
Analog5Min
Analog6Min
Analog7Min
Analog8Min
Analog9Min
Analog10Min
Analog11Min
Analog12Min
Analog13Min
Analog14Min
Analog15Min
Analog16Min
Analog17Min
Analog18Min
Analog19Min
Analog20Min
Analog21Min
Analog22Min
Analog23Min
Analog24Min
Analog25Min
Analog26Min
Analog27Min
Analog28Min
Analog29Min
Analog30Min
Analog31Min

Analog32Min
Analog33Min
Analog34Min
Analog35Min
Analog36Min
Analog37Min
Analog38Min
Analog39Min
Analog40Min
Analog41Min
Analog42Min
Analog43Min
Analog44Min
Analog45Min
Analog46Min
Analog47Min
Analog48Min
Analog49Min
Analog50Min
Analog51Min
Analog52Min
Analog53Min
Analog54Min
Analog55Min
Analog56Min
Analog57Min
Analog58Min
Analog59Min
Analog60Min
Analog61Min
Analog62Min
Analog63Min
Analog0Max

Analog1Max
Analog2Max
Analog3Max
Analog4Max
Analog5Max
Analog6Max
Analog7Max
Analog8Max
Analog9Max
Analog10Max
Analog11Max
Analog12Max
Analog13Max
Analog14Max
Analog15Max
Analog16Max
Analog17Max
Analog18Max
Analog19Max
Analog20Max
Analog21Max
Analog22Max
Analog23Max
Analog24Max
Analog25Max
Analog26Max
Analog27Max
Analog28Max
Analog29Max
Analog30Max
Analog31Max
Analog32Max
Analog33Max

Analog34Max
Analog35Max
Analog36Max
Analog37Max
Analog38Max
Analog39Max
Analog40Max
Analog41Max
Analog42Max
Analog43Max
Analog44Max
Analog45Max
Analog46Max
Analog47Max
Analog48Max
Analog49Max
Analog50Max
Analog51Max
Analog52Max
Analog53Max
Analog54Max
Analog55Max
Analog56Max
Analog57Max
Analog58Max
Analog59Max
Analog60Max
Analog61Max
Analog62Max
None

▪ See Also

[PPJoy Namespace](#)

[PPJoy...:::DirectionalPovMapping](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping Class

[Members](#) [See Also](#)

A DirectionalPovMapping object represents a specific type of Point-of-View (Pov) control that can be defined on a PPJoy joystick [Device](#). A DirectionalPovMapping defines the Pov's number (index) and the [DirectionalPovDataSources](#) that will provide the DirectionalPovMapping's state values that it will report to Windows.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class DirectionalPovMapping _
    Inherits PovMapping
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class DirectionalPovMapping : PovMapping
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class DirectionalPovMapping sealed : public PovMapping
```

■ Remarks

A DirectionalPovMapping sources its values from one or more Digital (or quasi-Digital) [DirectionalPovDataSources](#).

Contrast this behavior with that of a [ContinuousPovMapping](#), which sources its values from a single Analog or Reversed [ContinuousPovDataSource](#).

▀ Inheritance Hierarchy

System...:::Object

[PPJoy...:::Mapping](#)

[PPJoy...:::PovMapping](#)

PPJoy....:::DirectionalPovMapping

See Also

[DirectionalPovMapping Members](#)
[PPJoy Namespace](#)
[PPJoy...::ContinuousPovDataSources](#)
[PPJoy...::ContinuousPovMapping](#)
[PPJoy...::DirectionalPovDataSources](#)
[PPJoy...::PovMapping](#)
[PPJoy...::Device](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DirectionalPovMapping Constructor

[DirectionalPovMapping Class](#) [See Also](#)

Overload List

	Name	Description
•	DirectionalPovMapping()	Creates a new DirectionalPovMapping object.
•	DirectionalPovMapping(Int32)	Creates a new DirectionalPovMapping object.

See Also

[DirectionalPovMapping Class](#)
[DirectionalPovMapping Members](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping Constructor

[DirectionalPovMapping Class](#) [See Also](#)

Creates a new [DirectionalPovMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public DirectionalPovMapping()
```

Visual C++

```
public:  
DirectionalPovMapping()
```

See Also

[DirectionalPovMapping Class](#)
[DirectionalPovMapping Overload](#)
[PPJoy Namespace](#)
[PPJoy...::PovMapping](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping Constructor (Int32)

[DirectionalPovMapping Class](#) [See Also](#)

Creates a new [DirectionalPovMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    controlNumber As Integer _  
)
```

C#

```
public DirectionalPovMapping(  
    int controlNumber  
)
```

Visual C++

```
public:  
DirectionalPovMapping(  
    int controlNumber  
)
```

Parameters

controlNumber

Type: System..:::Int32

The zero-based index of this [PovMapping](#) in the collection of [PovMappings](#) defined on the same PPJoy [Device](#). For example, the first [PovMapping](#) in the collection will have a controlNumber of 0, the second [PovMapping](#) will have a controlNumber of 1, and so forth.

See Also

[DirectionalPovMapping Class](#)
[DirectionalPovMapping Overload](#)
[PPJoy Namespace](#)
[Mapping...:::ControlNumber](#)
[PPJoy...:::PovMapping](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DirectionalPovMapping Members

[DirectionalPovMapping Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [DirectionalPovMapping](#) type exposes the following members.

▪ Constructors

	Name	Description
≡ ♀	DirectionalPovMapping	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
	EastDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the East direction.
	NorthDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the North direction.
	SouthDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the South direction.
	WestDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the West direction.

▪ See Also

[DirectionalPovMapping Class](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

DirectionalPovMapping Properties

[DirectionalPovMapping Class](#) [See Also](#)

The [DirectionalPovMapping](#) type exposes the following properties.

Properties

Name	Description
 ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)
 EastDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the East direction.
 NorthDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the North direction.
 SouthDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the South direction.
 WestDataSource	Gets/sets the DirectionalPovDataSource that this DirectionalPovMapping will use to determine that it should report to Windows that it is being pressed in the West direction.

▪ See Also

[DirectionalPovMapping Class](#)

[PPJoy Namespace](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping...:::EastDataSource Property

[DirectionalPovMapping Class](#) [See Also](#)

Gets/sets the [DirectionalPovDataSource](#) that this [DirectionalPovMapping](#) will use to determine that it should report to Windows that it is being pressed in the **East** direction.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property EastDataSource As DirectionalPovDataSources
```

C#

```
public DirectionalPovDataSources EastDataSource { get; set; }
```

Visual C++

```
public:  
property DirectionalPovDataSources EastDataSource {  
    DirectionalPovDataSources get ();  
    void set (DirectionalPovDataSources value);  
}
```

Remarks

When this property is set to a **Digital** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **East** direction to Windows whenever the Digital [DirectionalPovSources](#)'s value is trueTrue (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital [DirectionalPovDataSource](#)'s value is falseFalse (False in Visual Basic) (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Max** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **East** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **maximum** value. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **maximum** (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Min** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **East** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **minimum**. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **minimum** (and no other direction is reporting **pressed**).

See Also

[DirectionalPovMapping Class](#)

[PPJoy Namespace](#)

[PPJoy...:::DirectionalPovDataSources](#)

[DirectionalPovMapping...:::NorthDataSource](#)

[DirectionalPovMapping...:::SouthDataSource](#)

[DirectionalPovMapping...:::WestDataSource](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping...:::NorthDataSource Property

[DirectionalPovMapping Class](#) [See Also](#)

Gets/sets the [DirectionalPovDataSource](#) that this [DirectionalPovMapping](#) will use to determine that it should report to Windows that it is being pressed in the **North** direction.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property NorthDataSource As DirectionalPovDataSources
```

C#

```
public DirectionalPovDataSources NorthDataSource { get; set; }
```

Visual C++

```
public:  
property DirectionalPovDataSources NorthDataSource {  
    DirectionalPovDataSources get ();  
    void set (DirectionalPovDataSources value);  
}
```

Remarks

When this property is set to a **Digital** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **North** direction to Windows whenever the Digital [DirectionalPovSources](#)'s value is trueTrue (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital [DirectionalPovDataSource](#)'s value is falseFalse (False in Visual Basic) (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Max** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **North** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **maximum** value. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **maximum** (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Min** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **North** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **minimum**. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **minimum** (and no other direction is reporting **pressed**).

See Also

[DirectionalPovMapping Class](#)

[PPJoy Namespace](#)

[PPJoy...:::DirectionalPovDataSources](#)

[DirectionalPovMapping...:::NorthDataSource](#)

[DirectionalPovMapping...:::SouthDataSource](#)

[DirectionalPovMapping...:::EastDataSource](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping...:::SouthDataSource Property

[DirectionalPovMapping Class](#) [See Also](#)

Gets/sets the [DirectionalPovDataSource](#) that this [DirectionalPovMapping](#) will use to determine that it should report to Windows that it is being pressed in the **South** direction.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property SouthDataSource As DirectionalPovDataSources
```

C#

```
public DirectionalPovDataSources SouthDataSource { get; set; }
```

Visual C++

```
public:  
property DirectionalPovDataSources SouthDataSource {  
    DirectionalPovDataSources get ();  
    void set (DirectionalPovDataSources value);  
}
```

Remarks

When this property is set to a **Digital** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **South** direction to Windows whenever the Digital [DirectionalPovSources](#)'s value is trueTrue (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital [DirectionalPovDataSource](#)'s value is falseFalse (False in Visual Basic) (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Max** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **South** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **maximum** value. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **maximum** (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Min** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **South** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **minimum**. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **minimum** (and no other direction is reporting **pressed**).

See Also

[DirectionalPovMapping Class](#)

[PPJoy Namespace](#)

[PPJoy...:::DirectionalPovDataSources](#)

[DirectionalPovMapping...:::NorthDataSource](#)

[DirectionalPovMapping...:::WestDataSource](#)

[DirectionalPovMapping...:::EastDataSource](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

DirectionalPovMapping...:::WestDataSource Property

[DirectionalPovMapping Class](#) [See Also](#)

Gets/sets the [DirectionalPovDataSource](#) that this [DirectionalPovMapping](#) will use to determine that it should report to Windows that it is being pressed in the **West** direction.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property WestDataSource As DirectionalPovDataSources
```

C#

```
public DirectionalPovDataSources WestDataSource { get; set; }
```

Visual C++

```
public:  
property DirectionalPovDataSources WestDataSource {  
    DirectionalPovDataSources get ();  
    void set (DirectionalPovDataSources value);  
}
```

Remarks

When this property is set to a **Digital** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **West** direction to Windows whenever the Digital [DirectionalPovDataSource](#)'s value is trueTrue (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital [DirectionalPovDataSource](#)'s value is falseFalse (False in Visual Basic) (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Max** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **West** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **maximum** value. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **maximum** (and no other direction is reporting **pressed**).

When this property is set to an **Analog-Min** [DirectionalPovDataSource](#), then this [DirectionalPovMapping](#) will report a value of **pressed** in the **West** direction to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at its **minimum**. Similarly, this [DirectionalPovMapping](#) will report a value of **centered** to Windows whenever the Analog [DirectionalPovDataSource](#)'s value is at any other value other than its **minimum** (and no other direction is reporting **pressed**).

See Also

[DirectionalPovMapping Class](#)

[PPJoy Namespace](#)

[PPJoy...:::DirectionalPovDataSources](#)

[DirectionalPovMapping...:::NorthDataSource](#)

[DirectionalPovMapping...:::SouthDataSource](#)

[DirectionalPovMapping...:::EastDataSource](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

JoystickMapScope Enumeration

[See Also](#)

JoystickMapScope defines an enumeration of scopes to which a

[MappingCollection](#) can be applied.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Enumeration JoystickMapScope
```

C#

```
public enum JoystickMapScope
```

Visual C++

```
public enum class JoystickMapScope
```

Members

Member name	Description
	The MappingCollection applies to all devices on the same Interface LPT number (i.e. all virtual devices or all physical parallel-port devices on the same LPT port).
Device	The MappingCollection applies only to the specific device it is assigned to. All other devices will either inherit their mappings from their interface (parent), or will have their own mappings assigned to them, or will use a default set of mappings.

▪ See Also

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

JoystickSubTypes Enumeration

[See Also](#)

JoystickSubTypes defines an enumeration of all possible PPJoy joystick sub-types.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Enumeration JoystickSubTypes
```

C#

```
public enum JoystickSubTypes
```

Visual C++

```
public enum class JoystickSubTypes
```

Members

Member name	Description
Genesis_Pad_A_B_C_Start	
Genesis_Pad_A_B_C_X_Y_Z_Start_Mode	
SNES_or_Virtual_Gameboy	
NES	
NotApplicable	

▪ See Also

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

JoystickTypes Enumeration

[See Also](#)

JoystickTypes defines an enumeration of all possible PPJoy joystick types. A joystick type is a combination of controller type and interface type, as shown in the PPJoy Control Panel.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Enumeration JoystickTypes
```

C#

```
public enum JoystickTypes
```

Visual C++

```
public enum class JoystickTypes
```

Members

Member name	Description
Joystick_TheMaze	
Joystick_IanHerries	
Joystick_TurboGraFX	
Joystick_Linux_v0802	
Joystick_Linux_DB9c	
Joystick_TorMod	
Joystick_DirectPad_Pro	
Joystick_TurboGraFX_SwappedButtons	
Joystick_LPT_JoyStick	
Joystick_CHAMPgames	
Joystick_STFormat	
Joystick_SNESKey2600	
Joystick_Amiga_4_Player	
Joystick_PCAE	
Genesis_Pad_Linux	
Genesis_Pad_DirectPad_Pro	
Genesis_Pad_NTPad_XP	
SNES_or_NESPad_Linux	
SNES_or_NESPad_DirectPadPro_Or_SNESKey	
Genesis_Pad_ConsoleCable	
Genesis_Pad_SNESKey	
Playstation_Pad_PSXPBLib	
Playstation_Pad_DirectPad_Pro	
Playstation_Pad_Linux	
Playstation_Pad_NTPad_XP	
Playstation_Pad_Megatap	
Virtual_Joystick	
Joystick_Linux_gamecon	

Joystick_LPTswitch
Radio_Control_TX
SNES_or_NESPad_PowerPad
Genesis_Pad_DirectPad_Pro_V6

▪ See Also

[PPJoy Namespace](#)

[PPJoy...:::Device](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Mapping Class

[Members](#) [See Also](#)

Mapping is the base class for all Mapping types. A Mapping represents a control on a PPJoy Virtual Joystick [Device](#) such as a Point-of-View hat, an axis, or a button. Mappings declare the presence of a specific control, and its position (index) among other controls of the same type on the same [Device](#). Mappings also define the data sources that feed these virtual controls state information, which, in turn, is reported to Windows and is accessible via DirectInput.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class Mapping
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class Mapping
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class Mapping
```

▀ Inheritance Hierarchy

System...:::Object

PPJoy...:::Mapping

[PPJoy...:::PovMapping](#)

[PPJoy...:::ButtonMapping](#)

[PPJoy...:::AxisMapping](#)

See Also

[Mapping Members](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Mapping Constructor

[Mapping Class](#) [See Also](#)

Overload List

Name	Description
`1 Mapping()	Creates a new Mapping object.
`2 Mapping(Int32)	Creates a new Mapping object.

See Also

[Mapping Class](#)

[Mapping Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Mapping Constructor

[Mapping Class](#) [See Also](#)

Creates a new Mapping object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public Mapping()
```

Visual C++

```
public:  
Mapping()
```

▪ See Also

[Mapping Class](#)

[Mapping Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Mapping Constructor (Int32)

[Mapping Class](#) [See Also](#)

Creates a new Mapping object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    controlNumber As Integer _  
)
```

C#

```
public Mapping(  
    int controlNumber  
)
```

Visual C++

```
public:  
Mapping(  
    int controlNumber  
)
```

Parameters

controlNumber

Type: System..:::Int32

an index to use for this control in the collection of all other controls of the same type on the same device.

See Also

[Mapping Class](#)

[Mapping Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Mapping Members

[Mapping](#) [Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [Mapping](#) type exposes the following members.

▪ Constructors

Name	Description
= ♡ Mapping	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

Name	Description
 ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device.

See Also

[Mapping Class](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

Mapping Properties

[Mapping Class](#) [See Also](#)

The [Mapping](#) type exposes the following properties.

Properties

Name	Description
 ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device.

See Also

[Mapping Class](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

Mapping...:::ControlNumber Property

[Mapping Class](#) [See Also](#)

Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property ControlNumber As Integer
```

C#

```
public int ControlNumber { get; set; }
```

Visual C++

```
public:  
property int ControlNumber {  
    int get ();  
    void set (int value);  
}
```

See Also

[Mapping Class](#)

[PPJoy Namespace](#)



Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection Class

[Members](#) [See Also](#)

A MappingCollection is a specialized collection that can store related [Mapping](#) objects together. This provides for ease of handling when many different [Mappings](#) must be manipulated as a group. A MappingCollection exposes several sub-collections, from which all [Mappings](#) of a particular Type can be retrieved (for instance, all [ButtonMappings](#) in the MappingCollection can be retrieved from the [ButtonMappings](#) property).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class MappingCollection _
    Implements IList, ICollection, IEnumerable, ICloneable
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class MappingCollection : IList,
    ICollection, IEnumerable, ICloneable
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class MappingCollection sealed : IList,
    ICollection, IEnumerable, ICloneable
```

▀ Inheritance Hierarchy

System...:::Object

PPJoy...:::MappingCollection

See Also

[MappingCollection Members](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection Constructor

[MappingCollection Class](#) [See Also](#)

Creates a new [MappingCollection](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public MappingCollection()
```

Visual C++

```
public:  
MappingCollection()
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection Members

[MappingCollection Class](#) [Constructors](#) [Methods](#) [Properties](#)

[Explicit Interface Implementations](#) [See Also](#)

The [MappingCollection](#) type exposes the following members.

Constructors

	Name	Description
≡	MappingCollection	Creates a new MappingCollection object.

Methods

	Name	Description
≡	Add	Overloaded.
≡	Clear	Removes all elements from the MappingCollection .
≡	Clone	Creates a shallow copy of the MappingCollection .
≡	Contains	Overloaded.
≡	CopyTo	Overloaded.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetEnumerator	Gets an IEnumerator that can be used to iterate over the MappingCollection .
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
≡	IndexOf	Overloaded.
≡	Insert	Overloaded.
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	Remove	Overloaded.
≡	RemoveAt	Removes the element at the specified index of the MappingCollection .
≡	Sort	Sorts the MappingCollection .
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	AxisMappings	Gets a MappingCollection containing all the AxisMappings contained in this MappingCollection .
	ButtonMappings	Gets a MappingCollection containing all the ButtonMapping objects contained in this MappingCollection .
	Count	Gets the number of Mappings in the MappingCollection .
	IsFixedSize	Gets a value indicating whether the MappingCollection has a fixed size.
	IsReadOnly	Gets a value indicating whether the MappingCollection is read-only.
	IsSynchronized	Gets a value indicating whether access to the MappingCollection is synchronized (thread safe).
	Item	Gets an item from the MappingCollection , given its index number.
	PovMappings	Gets a MappingCollection containing all the PovMappings contained in this MappingCollection .
	SyncRoot	Gets an object that can be used to synchronize access to the MappingCollection .

Explicit Interface Implementations

	Name	Description
	IList..:::Item	Gets or sets the element at the specified index.

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection Methods

[MappingCollection Class](#) [See Also](#)

The [MappingCollection](#) type exposes the following methods.

Methods

	Name	Description
≡	Add	Overloaded.
≡	Clear	Removes all elements from the MappingCollection .
≡	Clone	Creates a shallow copy of the MappingCollection .
≡	Contains	Overloaded.
≡	CopyTo	Overloaded.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetEnumerator	Gets an IEnumerator that can be used to iterate over the MappingCollection .
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
≡	IndexOf	Overloaded.
≡	Insert	Overloaded.
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	Remove	Overloaded.
≡	RemoveAt	Removes the element at the specified index of the MappingCollection .
≡	Sort	Sorts the MappingCollection .
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

■ See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection...:::Add Method

[MappingCollection Class](#) [See Also](#)

Overload List

	Name	Description
•	Add(Object)	Adds a Object (referencing a Mapping object) to the MappingCollection .
•	Add(Mapping)	Adds a Mapping to the MappingCollection .

See Also

[MappingCollection Class](#)

[MappingCollection Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::Add Method (Mapping)

[MappingCollection Class](#) [See Also](#)

Adds a [Mapping](#) to the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function Add ( _  
    value As Mapping _  
) As Integer
```

C#

```
public int Add(  
    Mapping value  
)
```

Visual C++

```
public:  
int Add(  
    Mapping^ value  
)
```

Parameters

value

Type: [PPJoy..:::Mapping](#)

A [Mapping](#) to add to the [MappingCollection](#).

Return Value

The index at which the [Mapping](#) was added to the [MappingCollection](#).

See Also

[MappingCollection Class](#)

[Add Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::Add Method (Object)

[MappingCollection Class](#) [See Also](#)

Adds a Object (referencing a [Mapping](#) object) to the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function Add ( _  
    value As Object _  
) As Integer
```

C#

```
public int Add(  
    Object value  
)
```

Visual C++

```
public:  
virtual int Add(  
    Object^ value  
) sealed
```

Parameters

value

Type: System..:::Object
an Object (referencing a [Mapping](#) object) to add to the [MappingCollection](#).

Return Value

The index of the Object in the [MappingCollection](#).

Implements

IList..:::Add(Object)

See Also

[MappingCollection Class](#)

[Add Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::Clear Method

[MappingCollection Class](#) [See Also](#)

Removes all elements from the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub Clear
```

C#

```
public void Clear()
```

Visual C++

```
public:  
virtual void Clear() sealed
```

Implements

```
IList...::Clear()()
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::Clone Method

[MappingCollection Class](#) [See Also](#)

Creates a shallow copy of the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Function Clone As Object
```

C#

```
public Object Clone()
```

Visual C++

```
public:  
virtual Object^ Clone() sealed
```

Return Value

A shallow copy of the [MappingCollection](#).

Implements

ICloneable...::Clone()()

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection....Contains Method

[MappingCollection Class](#) [See Also](#)

Overload List

	Name	Description
•	Contains(Object)	Determines whether an element is in the MappingCollection .
•	Contains(Mapping)	Determines whether a specific Mapping is in the MappingCollection .

See Also

[MappingCollection Class](#)

[MappingCollection Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....Contains Method (Mapping)

[MappingCollection Class](#) [See Also](#)

Determines whether a specific [Mapping](#) is in the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function Contains ( _  
    value As Mapping _  
) As Boolean
```

C#

```
public bool Contains(  
    Mapping value  
)
```

Visual C++

```
public:  
bool Contains(  
    Mapping^ value  
)
```

Parameters

value

Type: [PPJoy..:::Mapping](#)

The [Mapping](#) to locate in the [MappingCollection](#). value can be nullNothingnullptr a null reference (Nothing in Visual Basic).

Return Value

trueTruetrue (True in Visual Basic) if the [MappingCollection](#) contains the specified [Mapping](#), or falseFalsefalsefalse (False in Visual Basic) if it does not.

See Also

[MappingCollection Class](#)

[Contains Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::Contains Method (Object)

[MappingCollection Class](#) [See Also](#)

Determines whether an element is in the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function Contains ( _  
    value As Object _  
) As Boolean
```

C#

```
public bool Contains(  
    Object value  
)
```

Visual C++

```
public:  
virtual bool Contains(  
    Object^ value  
) sealed
```

Parameters

value

Type: System..:::Object

The Object (referencing a [Mapping](#)) to locate in the [MappingCollection](#).
value can be nullNothingnullptr a null reference (Nothing in Visual Basic).

Return Value

trueTruetrue (True in Visual Basic) if the [MappingCollection](#) contains the specified Object, or falseFalsefalse (False in Visual Basic) if it does not.

Implements

IList..:::Contains(Object)

See Also

[MappingCollection Class](#)

[Contains Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection....:::CopyTo Method

[MappingCollection Class](#) [See Also](#)

Overload List

	Name	Description
•	CopyTo(Array, Int32)	Copies the members of the MappingCollection to an array.
•	CopyTo(array<Mapping>[], Int32)	Copies the members of the MappingCollection to an array.

See Also

[MappingCollection Class](#)

[MappingCollection Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::CopyTo Method (Array, Int32)

[MappingCollection Class](#) [See Also](#)

Copies the members of the [MappingCollection](#) to an array.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub CopyTo ( _  
    array As Array, _  
    index As Integer _  
)
```

C#

```
public void CopyTo(  
    Array array,  
    int index  
)
```

Visual C++

```
public:  
virtual void CopyTo(  
    Array^ array,  
    int index  
) sealed
```

Parameters

array

Type: System..:::Array

An Array to copy the [MappingCollection](#)'s members to.

index

Type: System..:::Int32

The zero-based index into the array at which copying should begin at.

Implements

ICollection..:::CopyTo(Array, Int32)

See Also

[MappingCollection Class](#)

[CopyTo Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::CopyTo Method (array<Mapping>[][][], Int32)

[MappingCollection Class](#) [See Also](#)

Copies the members of the [MappingCollection](#) to an array.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub CopyTo ( _  
    array As Mapping\(\), _  
    index As Integer _  
)
```

C#

```
public void CopyTo(  
    Mapping[] array,  
    int index  
)
```

Visual C++

```
public:  
void CopyTo(  
    array<Mapping>^ array,  
    int index  
)
```

Parameters

array

Type: array<[PPJoy..:::Mapping](#)>[]()

A strongly-typed Array (of type [Mapping](#)[]), to which the [MappingCollection](#)'s members will be copied.

index

Type: System..:::Int32

The zero-based index into the array where copying should begin at.

See Also

[MappingCollection Class](#)

[CopyTo Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::GetEnumerator Method

[MappingCollection Class](#) [See Also](#)

Gets an IEnumerator that can be used to iterate over the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Function GetEnumerator As IEnumera
```

C#

```
public IEnumera
```

Visual C++

```
public:  
virtual IEnumera
```

Return Value

Implements

```
IEnumerable...::GetEnumerator()
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection....:::IndexOf Method

[MappingCollection Class](#) [See Also](#)

Overload List

	Name	Description
≡	IndexOf(Object)	Searches for the specified Object and returns the zero-based index of the first occurrence within the entire MappingCollection .
≡	IndexOf(Mapping)	Searches for the specified Mapping and returns the zero-based index of the first occurrence within the entire MappingCollection .

See Also

[MappingCollection Class](#)

[MappingCollection Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::IndexOf Method (Mapping)

[MappingCollection Class](#) [See Also](#)

Searches for the specified [Mapping](#) and returns the zero-based index of the first occurrence within the entire [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function IndexOf ( _  
    value As Mapping _  
) As Integer
```

C#

```
public int IndexOf(  
    Mapping value  
)
```

Visual C++

```
public:  
int IndexOf(  
    Mapping^ value  
)
```

Parameters

value

Type: [PPJoy..:::Mapping](#)

The [Mapping](#) to locate in the [MappingCollection](#). value can be nullNothingnullptr a null reference (Nothing in Visual Basic).

Return Value

The zero-based index of the first occurrence of value within the entire [MappingCollection](#), if found; otherwise, -1.

See Also

[MappingCollection Class](#)

[IndexOf Overload](#)

[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::IndexOf Method (Object)

[MappingCollection Class](#) [See Also](#)

Searches for the specified Object and returns the zero-based index of the first occurrence within the entire [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Function IndexOf ( _  
    value As Object _  
) As Integer
```

C#

```
public int IndexOf(  
    Object value  
)
```

Visual C++

```
public:  
virtual int IndexOf(  
    Object^ value  
) sealed
```

Parameters

value

Type: System..:::Object

The Object to locate in the [MappingCollection](#). value can be nullNothingnullptr a null reference (Nothing in Visual Basic).

Return Value

The zero-based index of the first occurrence of value within the entire [MappingCollection](#), if found; otherwise, -1.

Implements

IList..:::IndexOf(Object)

See Also

[MappingCollection Class](#)

[IndexOf Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection...:::Insert Method

[MappingCollection Class](#) [See Also](#)

Overload List

	Name	Description
≡	Insert(Int32, Mapping)	Inserts a Mapping into the MappingCollection at the specified index.
≡	Insert(Int32, Object)	Inserts an element into the MappingCollection at the specified index.

See Also

[MappingCollection Class](#)

[MappingCollection Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...::Insert Method (Int32, Mapping)

[MappingCollection Class](#) [See Also](#)

Inserts a [Mapping](#) into the [MappingCollection](#) at the specified index.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub Insert ( _  
    index As Integer, _  
    value As Mapping _  
)
```

C#

```
public void Insert(  
    int index,  
    Mapping value  
)
```

Visual C++

```
public:  
void Insert(  
    int index,  
    Mapping^ value  
)
```

Parameters

index

Type: System..:::Int32

The zero-based index at which value should be inserted.

value

Type: [PPJoy..:::Mapping](#)

The [Mapping](#) to insert. value can be null.

See Also

[MappingCollection Class](#)

[Insert Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...::Insert Method (Int32, Object)

[MappingCollection Class](#) [See Also](#)

Inserts an element into the [MappingCollection](#) at the specified index.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub Insert ( _  
    index As Integer, _  
    value As Object _  
)
```

C#

```
public void Insert(  
    int index,  
    Object value  
)
```

Visual C++

```
public:  
virtual void Insert(  
    int index,  
    Object^ value  
) sealed
```

Parameters

index

Type: System..:::Int32

The zero-based index at which value should be inserted.

value

Type: System..:::Object

The Object to insert. value can be nullNothingnullptr a null reference (Nothing in Visual Basic).

Implements

IList..:::Insert(Int32, Object)

See Also

[MappingCollection Class](#)

[Insert Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection....:::Remove Method

[MappingCollection Class](#) [See Also](#)

▪ Overload List

	Name	Description
•	<u>Remove(Object)</u>	Removes the first occurrence of a specific Object from the <u>MappingCollection</u> .
•	<u>Remove(Mapping)</u>	Removes a <u>Mapping</u> from the <u>MappingCollection</u> .

See Also

[MappingCollection Class](#)

[MappingCollection Members](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::Remove Method (Mapping)

[MappingCollection Class](#) [See Also](#)

Removes a [Mapping](#) from the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub Remove ( _  
                  obj As Mapping _  
)
```

C#

```
public void Remove(  
                  Mapping obj  
)
```

Visual C++

```
public:  
void Remove(  
            Mapping^ obj  
)
```

Parameters

obj

Type: [PPJoy..:::Mapping](#)

A [Mapping](#) to remove from the [MappingCollection](#).

See Also

[MappingCollection Class](#)

[Remove Overload](#)

[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::Remove Method (Object)

[MappingCollection Class](#) [See Also](#)

Removes the first occurrence of a specific Object from the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub Remove ( _  
                  value As Object _  
)
```

C#

```
public void Remove(  
                  Object value  
)
```

Visual C++

```
public:  
virtual void Remove(  
                  Object^ value  
) sealed
```

Parameters

value

Type: System..:::Object

The Object to remove from the [MappingCollection](#). value can be nullNothingnullptr a null reference (Nothing in Visual Basic).

Implements

IList..:::Remove(Object)

See Also

[MappingCollection Class](#)

[Remove Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...::RemoveAt Method

[MappingCollection Class](#) [See Also](#)

Removes the element at the specified index of the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub RemoveAt ( _  
                      index As Integer _  
)
```

C#

```
public void RemoveAt(  
                      int index  
)
```

Visual C++

```
public:  
virtual void RemoveAt(  
                      int index  
) sealed
```

Parameters

index

Type: System..:::Int32

The zero-based index of the element to remove.

Implements

IList..:::RemoveAt(Int32)

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::Sort Method

[MappingCollection Class](#) [See Also](#)

Sorts the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub Sort
```

C#

```
public void Sort()
```

Visual C++

```
public:  
void Sort()
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

MappingCollection Properties

[MappingCollection Class](#) [See Also](#)

The [MappingCollection](#) type exposes the following properties.

Properties

	Name	Description
	AxisMappings	Gets a MappingCollection containing all the AxisMappings contained in this MappingCollection .
	ButtonMappings	Gets a MappingCollection containing all the ButtonMapping objects contained in this MappingCollection .
	Count	Gets the number of Mappings in the MappingCollection .
	IsFixedSize	Gets a value indicating whether the MappingCollection has a fixed size.
	IsReadOnly	Gets a value indicating whether the MappingCollection is read-only.
	IsSynchronized	Gets a value indicating whether access to the MappingCollection is synchronized (thread safe).
	Item	Gets an item from the MappingCollection , given its index number.
	PovMappings	Gets a MappingCollection containing all the PovMappings contained in this MappingCollection .
	SyncRoot	Gets an object that can be used to synchronize access to the MappingCollection .

Explicit Interface Implementations

	Name	Description
	IList..:::Item	Gets or sets the element at the specified index.

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ □

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::AxisMappings Property

[MappingCollection Class](#) [See Also](#)

Gets a [MappingCollection](#) containing all the [AxisMappings](#) contained in this [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property AxisMappings As MappingCollection
```

C#

```
public MappingCollection AxisMappings { get; }
```

Visual C++

```
public:  
property MappingCollection^ AxisMappings {  
    MappingCollection^ get ();  
}
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::ButtonMappings Property

[MappingCollection Class](#) [See Also](#)

Gets a [MappingCollection](#) containing all the [ButtonMapping](#) objects contained in this [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property ButtonMappings As MappingCollection
```

C#

```
public MappingCollection ButtonMappings { get; }
```

Visual C++

```
public:  
property MappingCollection^ ButtonMappings {  
    MappingCollection^ get ();  
}
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::Count Property

[MappingCollection Class](#) [See Also](#)

Gets the number of [Mappings](#) in the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public ReadOnly Property Count As Integer
```

C#

```
public int Count { get; }
```

Visual C++

```
public:  
virtual property int Count {  
    int get () sealed;  
}
```

Implements

```
ICollection...::Count
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...:: IList...:: Item Property

[MappingCollection Class](#) [See Also](#)

Gets or sets the element at the specified index.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Private Property Item ( _  
    index As Integer _  
) As Object Implements IList.Item
```

C#

```
Object IList.Item[  
    int index  
] { get; set; }
```

Visual C++

```
private:  
virtual property Object^ Item[int index] {  
    Object^ get (int index) sealed = IList::Item::get;  
    void set (int index, Object^ value) sealed = IList::Item::set;  
}
```

Parameters

index

Type: System..:::Int32

The zero-based index of the element to get or set.

Return Value

The element at the specified index.

Implements

IList..:::Item[((Int32)])

Exceptions

Exception	Condition
System...:::ArgumentOutOfRangeException	index is not a valid index in the IList.
System...:::NotSupportedException	The property is set and the IList is read-only.

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...:::IsFixedSize Property

[MappingCollection Class](#) [See Also](#)

Gets a value indicating whether the [MappingCollection](#) has a fixed size.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public ReadOnly Property IsFixedSize As Boolean
```

C#

```
public bool IsFixedSize { get; }
```

Visual C++

```
public:  
virtual property bool IsFixedSize {  
    bool get () sealed;  
}
```

Implements

IList..:::IsFixedSize

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...:::IsReadOnly Property

[MappingCollection Class](#) [See Also](#)

Gets a value indicating whether the [MappingCollection](#) is read-only.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public ReadOnly Property IsReadOnly As Boolean
```

C#

```
public bool IsReadOnly { get; }
```

Visual C++

```
public:  
virtual property bool IsReadOnly {  
    bool get () sealed;  
}
```

Implements

IList...::IsReadOnly

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...:::IsSynchronized Property

[MappingCollection Class](#) [See Also](#)

Gets a value indicating whether access to the [MappingCollection](#) is synchronized (thread safe).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public ReadOnly Property IsSynchronized As Boolean
```

C#

```
public bool IsSynchronized { get; }
```

Visual C++

```
public:  
virtual property bool IsSynchronized {  
    bool get () sealed;  
}
```

Implements

ICollection...::IsSynchronized

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection...:::Item Property

[MappingCollection Class](#) [See Also](#)

Gets an item from the [MappingCollection](#), given its index number.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Default Property Item ( _  
    index As Integer _  
) As Mapping
```

C#

```
public Mapping this[  
    int index  
] { get; set; }
```

Visual C++

```
public:  
property Mapping^ default[int index] {  
    Mapping^ get (int index);  
    void set (int index, Mapping^ value);  
}
```

Parameters

index

Type: System..:::Int32

The zero-based index of the item to retrieve.

Return Value

The [Mapping](#) object corresponding to the given index in the [MappingCollection](#).

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

MappingCollection....:::PovMappings Property

[MappingCollection Class](#) [See Also](#)

Gets a [MappingCollection](#) containing all the [PovMappings](#) contained in this [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property PovMappings As MappingCollection
```

C#

```
public MappingCollection PovMappings { get; }
```

Visual C++

```
public:  
property MappingCollection^ PovMappings {  
    MappingCollection^ get ();  
}
```

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

MappingCollection....::SyncRoot Property

[MappingCollection Class](#) [See Also](#)

Gets an object that can be used to synchronize access to the [MappingCollection](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public ReadOnly Property SyncRoot As Object
```

C#

```
public Object SyncRoot { get; }
```

Visual C++

```
public:  
virtual property Object^ SyncRoot {  
    Object^ get () sealed;  
}
```

Implements

ICollection...::SyncRoot

See Also

[MappingCollection Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

OperationFailedException Class

[Members](#) [See Also](#)

Represents an error that occurs when performing a PPJoy IOCTL operation.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class OperationFailedException _
    Inherits PPJoyException
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class OperationFailedException : PPJoyException
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class OperationFailedException : public PPJoyException
```

▀ Inheritance Hierarchy

System...:::Object

 System...:::Exception

 System...:::ApplicationException

PPJoy...:::PPJoyException

 PPJoy...:::OperationFailedException

See Also

[OperationFailedException Members](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

OperationFailedException Constructor

[OperationFailedException Class](#) [See Also](#)

Overload List

	Name	Description
•	<u>OperationFailedException()</u>	Initializes a new instance of the <u>OperationFailedException</u> class.
•	<u>OperationFailedException(String)</u>	Initializes a new instance of the <u>OperationFailedException</u> class.
•	<u>OperationFailedException(SerializationInfo, StreamingContext)</u>	Initializes a new instance of the <u>OperationFailedException</u> class.
•	<u>OperationFailedException(String, Exception)</u>	Initializes a new instance of the <u>OperationFailedException</u> class.

See Also

[OperationFailedException Class](#)
[OperationFailedException Members](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

OperationFailedException Constructor

[OperationFailedException Class](#) [See Also](#)

Initializes a new instance of the [OperationFailedException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public OperationFailedException()
```

Visual C++

```
public:  
OperationFailedException()
```

See Also

[OperationFailedException Class](#)
[OperationFailedException Overload](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

OperationFailedException Constructor (SerializationInfo, StreamingContext)

[OperationFailedException Class](#) [See Also](#)

Initializes a new instance of the [OperationFailedException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected OperationFailedException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

Visual C++

```
protected:  
OperationFailedException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

Parameters

info

Type: System.Runtime.Serialization..:::SerializationInfo

The SerializationInfo that holds the serialized object data about the Exception being thrown.

context

Type: System.Runtime.Serialization..:::StreamingContext

The StreamingContext that contains contextual information about the source or destination.

See Also

[OperationFailedException Class](#)
[OperationFailedException Overload](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

OperationFailedException Constructor (String)

[OperationFailedException Class](#) [See Also](#)

Initializes a new instance of the [OperationFailedException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String _  
)
```

C#

```
public OperationFailedException(  
    string message  
)
```

Visual C++

```
public:  
OperationFailedException(  
    String^ message  
)
```

Parameters

message

Type: System..:::String

The message that describes the error.

See Also

[OperationFailedException Class](#)
[OperationFailedException Overload](#)
[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

OperationFailedException Constructor (String, Exception)

[OperationFailedException Class](#) [See Also](#)

Initializes a new instance of the [OperationFailedException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public OperationFailedException(  
    string message,  
    Exception inner  
)
```

Visual C++

```
public:  
OperationFailedException(  
    String^ message,  
    Exception^ inner  
)
```

Parameters

message

Type: System..:::String

The error message that explains the reason for the exception.

inner

Type: System..:::Exception

The exception that is the cause of the current exception, or nullNothingnullptr a null reference (Nothing in Visual Basic) if no inner exception is specified.

See Also

[OperationFailedException Class](#)
[OperationFailedException Overload](#)
[PPJoy Namespace](#)

≡ ⌂ ⌂

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

OperationFailedException Members

[OperationFailedException Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [OperationFailedException](#) type exposes the following members.

Constructors

	Name	Description
≡	OperationFailedException	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
≡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
≡	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
≡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional, user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

See Also

[OperationFailedException Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PovMapping Class

[Members](#) [See Also](#)

PovMapping is the base class for all PovMapping types. A PovMapping is a type of [Mapping](#) that declares and defines a Point-of-View control on a PPJoy Virtual Joystick [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class PovMapping _
    Inherits Mapping
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class PovMapping : Mapping
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class PovMapping : public Mapping
```

▀ Inheritance Hierarchy

System..:::Object

[PPJoy..:::Mapping](#)

PPJoy..:::PovMapping

[PPJoy..:::ContinuousPovMapping](#)

[PPJoy..:::DirectionalPovMapping](#)

See Also

[PovMapping Members](#)

[PPJoy Namespace](#)

[PPJoy...:::Mapping](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

PovMapping Constructor

[PovMapping Class](#) [See Also](#)

Overload List

Name	Description
<code>PovMapping()</code>	Creates a new PovMapping object.
<code>PovMapping(Int32)</code>	Creates a new PovMapping object.

See Also

[PovMapping Class](#)

[PovMapping Members](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PovMapping Constructor

[PovMapping Class](#) [See Also](#)

Creates a new [PovMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public PovMapping()
```

Visual C++

```
public:  
PovMapping()
```

See Also

[PovMapping Class](#)
[PovMapping Overload](#)
[PPJoy Namespace](#)
[PPJoy...:::Mapping](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PovMapping Constructor (Int32)

[PovMapping Class](#) [See Also](#)

Creates a new [PovMapping](#) object.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    controlNumber As Integer _  
)
```

C#

```
public PovMapping(  
    int controlNumber  
)
```

Visual C++

```
public:  
PovMapping(  
    int controlNumber  
)
```

Parameters

controlNumber

Type: System..:::Int32

The zero-based index of this [PovMapping](#) in the collection of [PovMappings](#) defined on a single PPJoy [Device](#). For example, the first [PovMapping](#) in the collection will have a controlNumber of 0, the second [PovMapping](#) will have a controlNumber of 1, and so forth.

See Also

[PovMapping Class](#)
[PovMapping Overload](#)
[PPJoy Namespace](#)
[PPJoy...:::Mapping](#)
[Mapping...:::ControlNumber](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

PovMapping Members

[PovMapping Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [PovMapping](#) type exposes the following members.

Constructors

Name	Description
= ♡ PovMapping	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Properties

	Name	Description
	ControlNumber	Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from Mapping .)

See Also

[PovMapping Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PPJoyException Class

[Members](#) [See Also](#)

PPJoyException is the base class Exception for all custom Exceptions that occur within the PPJoy wrapper.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<SerializableAttribute> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public Class PPJoyException _
    Inherits ApplicationException
```

C#

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public class PPJoyException : ApplicationException
```

Visual C++

```
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class PPJoyException : public ApplicationException
```

▀ Inheritance Hierarchy

System..:::Object

 System..:::Exception

 System..:::ApplicationException

 PPJoy....:PPJoyException

[PPJoy..:::DeviceAlreadyExistsException](#)

[PPJoy..:::DeviceNotFoundException](#)

[PPJoy....:OperationFailedException](#)

See Also

[PPJoyException Members](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

PPJoyException Constructor

[PPJoyException Class](#) [See Also](#)

Overload List

	Name	Description
•	PPJoyException()	Initializes a new instance of the PPJoyException class.
•	PPJoyException(String)	Initializes a new instance of the PPJoyException class.
•	PPJoyException(SerializationInfo, StreamingContext)	Initializes a new instance of the PPJoyException class.
•	PPJoyException(String, Exception)	Initializes a new instance of the PPJoyException class.

See Also

[PPJoyException Class](#)

[PPJoyException Members](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PPJoyException Constructor

[PPJoyException Class](#) [See Also](#)

Initializes a new instance of the [PPJoyException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public PPJoyException()
```

Visual C++

```
public:  
PPJoyException()
```

See Also

[PPJoyException Class](#)

[PPJoyException Overload](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PPJoyException Constructor (SerializationInfo, StreamingContext)

[PPJoyException Class](#) [See Also](#)

Initializes a new instance of the [PPJoyException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected PPJoyException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

Visual C++

```
protected:  
PPJoyException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

Parameters

info

Type: System.Runtime.Serialization..:::SerializationInfo

The SerializationInfo that holds the serialized object data about the Exception being thrown.

context

Type: System.Runtime.Serialization..:::StreamingContext

The StreamingContext that contains contextual information about the source or destination.

See Also

[PPJoyException Class](#)

[PPJoyException Overload](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PPJoyException Constructor (String)

[PPJoyException Class](#) [See Also](#)

Initializes a new instance of the [PPJoyException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String _  
)
```

C#

```
public PPJoyException(  
    string message  
)
```

Visual C++

```
public:  
PPJoyException(  
    String^ message  
)
```

Parameters

message

Type: System..:::String

The message that describes the error.

See Also

[PPJoyException Class](#)

[PPJoyException Overload](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

PPJoyException Constructor (String, Exception)

[PPJoyException Class](#) [See Also](#)

Initializes a new instance of the [PPJoyException](#) class.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public PPJoyException(  
    string message,  
    Exception inner  
)
```

Visual C++

```
public:  
PPJoyException(  
    String^ message,  
    Exception^ inner  
)
```

Parameters

message

Type: System..:::String

The error message that explains the reason for the Exception.

inner

Type: System..:::Exception

The Exception that is the cause of the current Exception, or nullNothingnullptr a null reference (Nothing in Visual Basic) if no inner Exception is specified.

See Also

[PPJoyException Class](#)

[PPJoyException Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

PPJoyException Members

[PPJoyException Class](#) [Constructors](#) [Methods](#) [Properties](#) [See Also](#)

The [PPJoyException](#) type exposes the following members.

▪ Constructors

Name	Description
= ♡ PPJoyException	Overloaded.

Methods

	Name	Description
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
≡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
≡	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
≡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional, user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

▪ See Also

[PPJoyException Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick Class

[Members](#) [See Also](#)

A VirtualJoystick provides an easy-to-use interface for setting the PPJoy data source states for a single PPJoy Virtual Joystick [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
<ComVisibleAttribute(True)> _
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class VirtualJoystick _
    Implements IDisposable
```

C#

```
[ComVisibleAttribute(true)]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class VirtualJoystick : IDisposable
```

Visual C++

```
[ComVisibleAttribute(true)]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class VirtualJoystick sealed : IDisposable
```

▀ Inheritance Hierarchy

System...:::Object

PPJoy...:::VirtualJoystick

See Also

[VirtualJoystick Members](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

VirtualJoystick Constructor

[VirtualJoystick Class](#) [See Also](#)

Overload List

	Name	Description
•	VirtualJoystick()	Creates a new VirtualJoystick instance.
•	VirtualJoystick(Int32)	Constructs a new VirtualJoystick instance.

See Also

[VirtualJoystick Class](#)

[VirtualJoystick Members](#)

[PPJoy Namespace](#)

≡ 🔍

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick Constructor

[VirtualJoystick Class](#) [See Also](#)

Creates a new [VirtualJoystick](#) instance.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub New
```

C#

```
public VirtualJoystick()
```

Visual C++

```
public:  
VirtualJoystick()
```

Remarks

If you use this "default" constructor, you will need to set the [VirtualStickNumber](#) property manually. The default constructor is supplied so that non-.NET (COM) clients can use this wrapper class, since COM requires classes to have a default constructor. If you don't set the [VirtualStickNumber](#) property manually, the virtual stick number defaults to PPJoy Virtual Joystick #1.

See Also

[VirtualJoystick Class](#)

[VirtualJoystick Overload](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick Constructor (Int32)

[VirtualJoystick Class](#) [See Also](#)

Constructs a new [VirtualJoystick](#) instance.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub New ( _  
    virtualStickNumber As Integer _  
)
```

C#

```
public VirtualJoystick(  
    int virtualStickNumber  
)
```

Visual C++

```
public:  
VirtualJoystick(  
    int virtualStickNumber  
)
```

Parameters

virtualStickNumber

Type: System..:::Int32

The one-based PPJoy virtual number that will be managed by this [VirtualJoystick](#) instance.

See Also

[VirtualJoystick Class](#)

[VirtualJoystick Overload](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

VirtualJoystick Fields

[VirtualJoystick Class](#) [See Also](#)

The [VirtualJoystick](#) type exposes the following fields.

Fields

	Name	Description
• s	MaxAnalogDataSources	The maximum number of analog data sources supported by PPJoy on a single Device .
• s	MaxAnalogDataSourceVal	The maximum value that can be applied to an analog data source.
• s	MaxDigitalDataSources	The maximum number of digital data sources supported by PPJoy on a single Device .
• s	MaxVirtualDevices	The maximum number of virtual joystick Device s supported by PPJoy.
• s	MaxVisibleAxes	The maximum number of axes that can be created on a PPJoy virtual joystick Device .
• s	MaxVisibleButtons	The maximum number of buttons that can be created on a PPJoy virtual joystick Device .
• s	MaxVisiblePovs	The maximum number of POVs that can be created on a PPJoy virtual joystick Device .
• s	MinAnalogDataSourceVal	The minimum value that can be applied to an analog data source (except PovCentered).
• s	PovCentered	The value that should be set on an analog data source when that data source is assigned to a PovMapping and when the PovMapping should be centered .

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MaxAnalogDataSources Field

[VirtualJoystick Class](#) [See Also](#)

The maximum number of analog data sources supported by PPJoy on a single [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxAnalogDataSources As Integer
```

C#

```
public const int MaxAnalogDataSources
```

Visual C++

```
public:  
literal int MaxAnalogDataSources
```

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MaxAnalogDataSourceVal Field

[VirtualJoystick Class](#) [See Also](#)

The maximum value that can be applied to an analog data source.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxAnalogDataSourceVal As Integer
```

C#

```
public const int MaxAnalogDataSourceVal
```

Visual C++

```
public:  
literal int MaxAnalogDataSourceVal
```

See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....:::MaxDigitalDataSources Field

[VirtualJoystick Class](#) [See Also](#)

The maximum number of digital data sources supported by PPJoy on a single [Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxDigitalDataSources As Integer
```

C#

```
public const int MaxDigitalDataSources
```

Visual C++

```
public:  
literal int MaxDigitalDataSources
```

▪ See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MaxVirtualDevices Field

[VirtualJoystick Class](#) [See Also](#)

The maximum number of virtual joystick [Devices](#) supported by PPJoy.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxVirtualDevices As Integer
```

C#

```
public const int MaxVirtualDevices
```

Visual C++

```
public:  
literal int MaxVirtualDevices
```

▪ See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ ▶

□ Visual Basic □ C#

□ Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MaxVisibleAxes Field

[VirtualJoystick Class](#) [See Also](#)

The maximum number of axes that can be created on a PPJoy virtual joystick

[Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxVisibleAxes As Integer
```

C#

```
public const int MaxVisibleAxes
```

Visual C++

```
public:  
literal int MaxVisibleAxes
```

▪ See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MaxVisibleButtons Field

[VirtualJoystick Class](#) [See Also](#)

The maximum number of buttons that can be created on a PPJoy virtual joystick

[Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxVisibleButtons As Integer
```

C#

```
public const int MaxVisibleButtons
```

Visual C++

```
public:  
literal int MaxVisibleButtons
```

▪ See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MaxVisiblePovs Field

[VirtualJoystick Class](#) [See Also](#)

The maximum number of POVs that can be created on a PPJoy virtual joystick

[Device](#).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MaxVisiblePovs As Integer
```

C#

```
public const int MaxVisiblePovs
```

Visual C++

```
public:  
literal int MaxVisiblePovs
```

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::MinAnalogDataSourceVal Field

[VirtualJoystick Class](#) [See Also](#)

The minimum value that can be applied to an analog data source (except [PovCentered](#)).

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Const MinAnalogDataSourceVal As Integer
```

C#

```
public const int MinAnalogDataSourceVal
```

Visual C++

```
public:  
literal int MinAnalogDataSourceVal
```

▪ See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ □

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....:::PovCentered Field

[VirtualJoystick Class](#) [See Also](#)

The value that should be set on an analog data source when that data source is assigned to a [PovMapping](#) and when the [PovMapping](#) should be **centered**.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Const PovCentered As Integer
```

C#

```
public const int PovCentered
```

Visual C++

```
public:  
literal int PovCentered
```

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

VirtualJoystick Members

[VirtualJoystick Class](#) [Constructors](#) [Methods](#) [Fields](#) [Properties](#) [See Also](#)

The [VirtualJoystick](#) type exposes the following members.

Constructors

Name	Description
= ♡ VirtualJoystick	Overloaded.

Methods

	Name	Description
≡	<u>Dispose</u>	Public implementation of the Dispose()() method.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	<u>SendUpdates</u>	Sends all pending updates to PPJoy.
≡	<u>SetAnalogDataSourceValue</u>	Sets an individual Analog data source value.
≡	<u>SetDigitalDataSourceState</u>	Sets an individual Digital data source state.
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

Fields

	Name	Description
• s	MaxAnalogDataSources	The maximum number of analog data sources supported by PPJoy on a single Device .
• s	MaxAnalogDataSourceVal	The maximum value that can be applied to an analog data source.
• s	MaxDigitalDataSources	The maximum number of digital data sources supported by PPJoy on a single Device .
• s	MaxVirtualDevices	The maximum number of virtual joystick Device s supported by PPJoy.
• s	MaxVisibleAxes	The maximum number of axes that can be created on a PPJoy virtual joystick Device .
• s	MaxVisibleButtons	The maximum number of buttons that can be created on a PPJoy virtual joystick Device .
• s	MaxVisiblePovs	The maximum number of POVs that can be created on a PPJoy virtual joystick Device .
• s	MinAnalogDataSourceVal	The minimum value that can be applied to an analog data source (except PovCentered).
• s	PovCentered	The value that should be set on an analog data source when that data source is assigned to a PovMapping and when the PovMapping should be centered .

Properties

	Name	Description
	<u>VirtualStickNumber</u>	Gets/sets the PPJoy virtual <u>Device</u> number that this <u>VirtualJoystick</u> instance is managing.

See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

VirtualJoystick Methods

[VirtualJoystick Class](#) [See Also](#)

The [VirtualJoystick](#) type exposes the following methods.

Methods

	Name	Description
≡	<u>Dispose</u>	Public implementation of the Dispose()() method.
≡	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
💡	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
≡	GetHashCode	Serves as a hash function for a particular type. GetHashCode()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
≡	GetType	Gets the Type of the current instance. (Inherited from Object.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	<u>SendUpdates</u>	Sends all pending updates to PPJoy.
≡	<u>SetAnalogDataSourceValue</u>	Sets an individual Analog data source value.
≡	<u>SetDigitalDataSourceState</u>	Sets an individual Digital data source state.
≡	ToString	Returns a String that represents the current Object. (Inherited from Object.)

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::Dispose Method

[VirtualJoystick Class](#) [See Also](#)

Public implementation of the Dispose()() method.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub Dispose
```

C#

```
public void Dispose()
```

Visual C++

```
public:  
virtual void Dispose() sealed
```

Implements

```
IDisposable...::Dispose()()
```

▪ See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::SendUpdates Method

[VirtualJoystick Class](#) [See Also](#)

Sends all pending updates to PPJoy.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ **Syntax**

Visual Basic (Declaration)

```
Public Sub SendUpdates
```

C#

```
public void SendUpdates()
```

Visual C++

```
public:  
void SendUpdates()
```

See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ ▶

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::SetAnalogDataSourceValue Method

[VirtualJoystick Class](#) [See Also](#)

Sets an individual Analog data source value.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub SetAnalogDataSourceValue ( _
    dataSourceNum As Integer, _
    newValue As Integer _
)
```

C#

```
public void SetAnalogDataSourceValue(
    int dataSourceNum,
    int newValue
)
```

Visual C++

```
public:
void SetAnalogDataSourceValue(
    int dataSourceNum,
    int newValue
)
```

Parameters

dataSourceNum

Type: System..:::Int32

Zero-based index of the Analog data source to update.

newValue

Type: System..:::Int32

A value between [MinAnalogDataSourceVal](#) and [MaxAnalogDataSourceVal](#), that will be assigned to the Analog data source.

Remarks

Axis data source values that are set by calling the SetAnalogDataSourceValue(Int32, Int32) method do not get passed to PPJoy until the [SendUpdates\(000\)](#) method is called. This allows multiple data source value updates to be passed to the PPJoy driver in a single pass.

Exceptions

Exception	Condition
System...:::ArgumentOutOfRangeException	Thrown if the dataSourceNum param < 1 or > <u>MaxAnalogDataSources</u> ; also thrown if the newValue argument is < <u>MinAnalogDataSourceVal</u> or > <u>MaxAnalogDataSourceVal</u>

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....::SetDigitalDataSourceState Method

[VirtualJoystick Class](#) [See Also](#)

Sets an individual Digital data source state.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Sub SetDigitalDataSourceState ( _  
    dataSourceNum As Integer, _  
    newValue As Boolean _  
)
```

C#

```
public void SetDigitalDataSourceState(  
    int dataSourceNum,  
    bool newValue  
)
```

Visual C++

```
public:  
void SetDigitalDataSourceState(  
    int dataSourceNum,  
    bool newValue  
)
```

Parameters

dataSourceNum

Type: System..:::Int32

Zero-based index of the digital data source to update.

newValue

Type: System..:::Boolean

A new value to apply to the specified digital data source.

Remarks

This value does not get passed to PPJoy until the [SendUpdatesQQQ](#) method gets called. This allows multiple digital data source state updates to be passed to the PPJoy driver in a single pass.

Exceptions

Exception	Condition
System...:::ArgumentOutOfRangeException argument < 1 or > <u>MaxDigitalDataSources</u>	Thrown if the dataSourceNum

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)

Visual Basic C#

Visual C++

Include Protected Members

Include Inherited Members

PPJoy Wrapper Library reference library

VirtualJoystick Properties

[VirtualJoystick Class](#) [See Also](#)

The [VirtualJoystick](#) type exposes the following properties.

Properties

	Name	Description
	<u>VirtualStickNumber</u>	Gets/sets the PPJoy virtual <u>Device</u> number that this <u>VirtualJoystick</u> instance is managing.

See Also

[VirtualJoystick Class](#)

[PPJoy Namespace](#)

≡ 

Visual Basic C#

Visual C++

PPJoy Wrapper Library reference library

VirtualJoystick....:::VirtualStickNumber Property

[VirtualJoystick Class](#) [See Also](#)

Gets/sets the PPJoy virtual [Device](#) number that this [VirtualJoystick](#) instance is managing.

Namespace: [PPJoy](#)

Assembly: PPJoyWrapper (in PPJoyWrapper.dll)

▪ Syntax

Visual Basic (Declaration)

```
Public Property VirtualStickNumber As Integer
```

C#

```
public int VirtualStickNumber { get; set; }
```

Visual C++

```
public:  
property int VirtualStickNumber {  
    int get ();  
    void set (int value);  
}
```

See Also

[VirtualJoystick Class](#)
[PPJoy Namespace](#)
