Visual Basic  C#
Visual C++
PPJoy Wrapper Library reference library
PPJoy Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AxisMapping</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>ButtonMapping</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>ContinuousPovMapping</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Device</strong></td>
<td>A Device object (an instance of the Device class) represents a single PPJoy joystick Device. Represents an error that occurs when trying to create a PPJoy Device that already exists.</td>
</tr>
</tbody>
</table>
**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

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AxisDataSources</strong></td>
<td>Defines an enumeration of PPJoy data sources that can be used with AxisMapping objects.</td>
</tr>
<tr>
<td><strong>AxisTypes</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>ButtonDataSources</strong></td>
<td>Defines an enumeration of PPJoy data sources that can be used with ButtonMapping objects.</td>
</tr>
<tr>
<td><strong>ContinuousPovDataSources</strong></td>
<td>Defines an enumeration of PPJoy data sources that can be used with ContinuousPovMapping objects.</td>
</tr>
<tr>
<td><strong>DirectionalPovDataSources</strong></td>
<td>Defines an enumeration of PPJoy data sources that can be used with DirectionalPovMapping objects.</td>
</tr>
<tr>
<td><strong>JoystickMapScope</strong></td>
<td>Defines an enumeration of scopes to which a MappingCollection can be applied.</td>
</tr>
<tr>
<td><strong>JoystickSubTypes</strong></td>
<td>Defines an enumeration of all possible PPJoy joystick sub-types.</td>
</tr>
<tr>
<td><strong>JoystickTypes</strong></td>
<td>A joystick type is a combination of controller type and interface type, as shown in the PPJoy Control Panel.</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital0</td>
<td></td>
</tr>
<tr>
<td>Digital1</td>
<td></td>
</tr>
<tr>
<td>Digital2</td>
<td></td>
</tr>
<tr>
<td>Digital3</td>
<td></td>
</tr>
<tr>
<td>Digital4</td>
<td></td>
</tr>
<tr>
<td>Digital5</td>
<td></td>
</tr>
<tr>
<td>Digital6</td>
<td></td>
</tr>
<tr>
<td>Digital7</td>
<td></td>
</tr>
<tr>
<td>Digital8</td>
<td></td>
</tr>
<tr>
<td>Digital9</td>
<td></td>
</tr>
<tr>
<td>Digital10</td>
<td></td>
</tr>
<tr>
<td>Digital11</td>
<td></td>
</tr>
<tr>
<td>Digital12</td>
<td></td>
</tr>
<tr>
<td>Digital13</td>
<td></td>
</tr>
<tr>
<td>Digital14</td>
<td></td>
</tr>
<tr>
<td>Digital15</td>
<td></td>
</tr>
<tr>
<td>Digital16</td>
<td></td>
</tr>
<tr>
<td>Digital17</td>
<td></td>
</tr>
<tr>
<td>Digital18</td>
<td></td>
</tr>
<tr>
<td>Digital19</td>
<td></td>
</tr>
<tr>
<td>Digital20</td>
<td></td>
</tr>
<tr>
<td>Digital21</td>
<td></td>
</tr>
<tr>
<td>Digital22</td>
<td></td>
</tr>
<tr>
<td>Digital23</td>
<td></td>
</tr>
<tr>
<td>Digital24</td>
<td></td>
</tr>
<tr>
<td>Digital25</td>
<td></td>
</tr>
<tr>
<td>Digital26</td>
<td></td>
</tr>
<tr>
<td>Digital27</td>
<td></td>
</tr>
</tbody>
</table>
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
See Also

PPJoy Namespace
PPJoy:::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.

**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
PPJoy Wrapper Library reference library

AxisMapping Constructor

AxisMapping Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AxisMapping()</code></td>
<td>Creates a new <code>AxisMapping</code> object.</td>
</tr>
<tr>
<td><code>AxisMapping(Int32)</code></td>
<td>Creates a new <code>AxisMapping</code> object.</td>
</tr>
</tbody>
</table>
See Also

AxisMapping Class
AxisMapping Members
PPJoy Namespace
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
Creates a new AxisMapping object.

**Namespace:**  PPJoy  
**Assembly:**  PPJoyWrapper (in PPJoyWrapper.dll)
## Syntax

### Visual Basic (Declaration)

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

### C#

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

### Visual C++

```cpp
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
The **AxisMapping** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AxisMapping</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>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.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AxisType</strong></td>
<td>Gets/sets the type of Windows axis that this <strong>AxisMapping</strong> will report itself as. 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 <strong>Mapping</strong>.) Gets/sets the PPJoy <strong>AxisDataSource</strong> that this <strong>AxisMapping</strong> will use as the source of the values that it will report to Windows.</td>
</tr>
<tr>
<td><strong>ControlNumber</strong></td>
<td>Gets/sets the PPJoy <strong>AxisDataSource</strong> that this <strong>AxisMapping</strong> will use as the source of the values that it will report to Windows.</td>
</tr>
<tr>
<td><strong>MaxDataSource</strong></td>
<td>Gets/sets the PPJoy <strong>AxisDataSource</strong> that this <strong>AxisMapping</strong> will use as the source of the values that it will report to Windows.</td>
</tr>
<tr>
<td><strong>MinDataSource</strong></td>
<td>Gets/sets the PPJoy <strong>AxisDataSource</strong> that this <strong>AxisMapping</strong> will use as the source of the values that it will report to Windows.</td>
</tr>
</tbody>
</table>
See Also

AxisMapping Class
PPJoy Namespace
The **AxisMapping** type exposes the following properties.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AxisType</strong></td>
<td>Gets/sets the type of Windows axis that this <strong>AxisMapping</strong> will report itself as. 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 <strong>Mapping</strong>.)</td>
</tr>
<tr>
<td><strong>ControlNumber</strong></td>
<td>Gets/sets the PPJoy <strong>AxisDataSource</strong> that this <strong>AxisMapping</strong> will use as the source of the values that it will report to Windows. Gets/sets the PPJoy <strong>AxisDataSource</strong> that this <strong>AxisMapping</strong> will use as the source of the values that it will report to Windows.</td>
</tr>
</tbody>
</table>
See Also

AxisMapping Class
PPJoy Namespace
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
AxisMapping..::.MaxDataSource Property

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 true. (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
AxisMapping.MinDataSource Property

AxesMapping 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 true (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 true (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 true (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 true or false (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 **AxisDataSources:**

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 `AxisDataSource`, when the value of the underlying `AxisDataSource` increases, the value reported by the `AxisMapping` to Windows will increase proportionately.

For `ReversedAxisDataSource`, 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
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
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>XRotation</td>
<td></td>
</tr>
<tr>
<td>YRotation</td>
<td></td>
</tr>
<tr>
<td>ZRotation</td>
<td></td>
</tr>
<tr>
<td>Slider</td>
<td></td>
</tr>
<tr>
<td>Pov</td>
<td></td>
</tr>
<tr>
<td>Dial</td>
<td></td>
</tr>
<tr>
<td>Wheel</td>
<td></td>
</tr>
<tr>
<td>VX</td>
<td></td>
</tr>
<tr>
<td>VY</td>
<td></td>
</tr>
<tr>
<td>VZ</td>
<td></td>
</tr>
<tr>
<td>VBRX</td>
<td></td>
</tr>
<tr>
<td>VBRY</td>
<td></td>
</tr>
<tr>
<td>VBRZ</td>
<td></td>
</tr>
<tr>
<td>Rudder</td>
<td></td>
</tr>
<tr>
<td>Throttle</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PPJoy Namespace
PPJoy::AxisMapping
PPJoy::PovMapping
PPJoy::ContinuousPovMapping
PPJoy::DirectionalPovMapping
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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital0</td>
<td></td>
</tr>
<tr>
<td>Digital1</td>
<td></td>
</tr>
<tr>
<td>Digital2</td>
<td></td>
</tr>
<tr>
<td>Digital3</td>
<td></td>
</tr>
<tr>
<td>Digital4</td>
<td></td>
</tr>
<tr>
<td>Digital5</td>
<td></td>
</tr>
<tr>
<td>Digital6</td>
<td></td>
</tr>
<tr>
<td>Digital7</td>
<td></td>
</tr>
<tr>
<td>Digital8</td>
<td></td>
</tr>
<tr>
<td>Digital9</td>
<td></td>
</tr>
<tr>
<td>Digital10</td>
<td></td>
</tr>
<tr>
<td>Digital11</td>
<td></td>
</tr>
<tr>
<td>Digital12</td>
<td></td>
</tr>
<tr>
<td>Digital13</td>
<td></td>
</tr>
<tr>
<td>Digital14</td>
<td></td>
</tr>
<tr>
<td>Digital15</td>
<td></td>
</tr>
<tr>
<td>Digital16</td>
<td></td>
</tr>
<tr>
<td>Digital17</td>
<td></td>
</tr>
<tr>
<td>Digital18</td>
<td></td>
</tr>
<tr>
<td>Digital19</td>
<td></td>
</tr>
<tr>
<td>Digital20</td>
<td></td>
</tr>
<tr>
<td>Digital21</td>
<td></td>
</tr>
<tr>
<td>Digital22</td>
<td></td>
</tr>
<tr>
<td>Digital23</td>
<td></td>
</tr>
<tr>
<td>Digital24</td>
<td></td>
</tr>
<tr>
<td>Digital25</td>
<td></td>
</tr>
<tr>
<td>Digital26</td>
<td></td>
</tr>
<tr>
<td>Digital27</td>
<td></td>
</tr>
</tbody>
</table>
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
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
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)

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

#### C#

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

#### Visual C++

```cpp
[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
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ButtonMapping()()</td>
<td>Creates a new ButtonMapping object.</td>
</tr>
<tr>
<td>ButtonMapping(Int32)</td>
<td>Creates a new ButtonMapping object.</td>
</tr>
</tbody>
</table>
See Also

ButtonMapping Class
ButtonMapping Members
PPJoy Namespace
PPJoy Wrapper Library reference library

ButtonDownMapping Constructor

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
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

colorNumber
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ButtonMapping</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>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.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ControlNumber</strong></td>
<td>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 <a href="#">Mapping</a>).</td>
</tr>
<tr>
<td><strong>DataSource</strong></td>
<td>Gets/sets the PPJoy <a href="#">ButtonDataSource</a> that this <a href="#">ButtonMapping</a> will use as the source of the values that it will report to Windows.</td>
</tr>
</tbody>
</table>
See Also

ButtonMapping Class
PPJoy Namespace
The **ButtonMapping** type exposes the following properties.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlNumber</td>
<td>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.)</td>
</tr>
<tr>
<td>DataSource</td>
<td>Gets/sets the PPJoy ButtonDataSource that this ButtonMapping will use as the source of the values that it will report to Windows.</td>
</tr>
</tbody>
</table>
See Also

ButtonMapping Class
PPJoy Namespace
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#**

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

**Visual C++**

```cpp
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 true (True in Visual Basic). Similarly, a value of unpressed will be reported, whenever the Digital ButtonDataSource's value is false (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
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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog0</td>
<td></td>
</tr>
<tr>
<td>Analog1</td>
<td></td>
</tr>
<tr>
<td>Analog2</td>
<td></td>
</tr>
<tr>
<td>Analog3</td>
<td></td>
</tr>
<tr>
<td>Analog4</td>
<td></td>
</tr>
<tr>
<td>Analog5</td>
<td></td>
</tr>
<tr>
<td>Analog6</td>
<td></td>
</tr>
<tr>
<td>Analog7</td>
<td></td>
</tr>
<tr>
<td>Analog8</td>
<td></td>
</tr>
<tr>
<td>Analog9</td>
<td></td>
</tr>
<tr>
<td>Analog10</td>
<td></td>
</tr>
<tr>
<td>Analog11</td>
<td></td>
</tr>
<tr>
<td>Analog12</td>
<td></td>
</tr>
<tr>
<td>Analog13</td>
<td></td>
</tr>
<tr>
<td>Analog14</td>
<td></td>
</tr>
<tr>
<td>Analog15</td>
<td></td>
</tr>
<tr>
<td>Analog16</td>
<td></td>
</tr>
<tr>
<td>Analog17</td>
<td></td>
</tr>
<tr>
<td>Analog18</td>
<td></td>
</tr>
<tr>
<td>Analog19</td>
<td></td>
</tr>
<tr>
<td>Analog20</td>
<td></td>
</tr>
<tr>
<td>Analog21</td>
<td></td>
</tr>
<tr>
<td>Analog22</td>
<td></td>
</tr>
<tr>
<td>Analog23</td>
<td></td>
</tr>
<tr>
<td>Analog24</td>
<td></td>
</tr>
<tr>
<td>Analog25</td>
<td></td>
</tr>
<tr>
<td>Analog26</td>
<td></td>
</tr>
<tr>
<td>Analog27</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PPJoy Namespace
PPJoy::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.

**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
PPJoy Wrapper Library reference library
ContinuousPovMapping Constructor

ContinuousPovMapping Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ContinuousPovMapping()</strong></td>
<td>Creates a new <strong>ContinuousPovMapping</strong> object.</td>
</tr>
<tr>
<td><strong>ContinuousPovMapping(Int32)</strong></td>
<td>Creates a new <strong>ContinuousPovMapping</strong> object.</td>
</tr>
</tbody>
</table>
See Also

ContinuousPovMapping Class
ContinuousPovMapping Members
PPJoy Namespace
Creates a new `ContinuousPovMapping` object.

**Namespace:** [PPJoy](https://example.com/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
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

tcontrolNumber
    Type: System::Int32
    The zero-based index of this PovMapping in the collection of PovMappings defined on the same PJPoy 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
The ContinuousPovMapping type exposes the following members.
# Constructors

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ControlNumber</strong></td>
<td>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 <a href="#">Mapping</a>).</td>
</tr>
<tr>
<td><strong>DataSource</strong></td>
<td>Gets/sets the PPJoy <a href="#">ContinuousPovDataSource</a> that this <a href="#">PovMapping</a> will use as the source of the values that it will report to Windows.</td>
</tr>
</tbody>
</table>
See Also

ContinuousPovMapping Class
PPJoy Namespace
The **ContinuousPovMapping** type exposes the following properties.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlNumber</td>
<td>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.)</td>
</tr>
<tr>
<td>DataSource</td>
<td>Gets/sets the PPJoy ContinuousPovDataSource that this PovMapping will use as the source of the values that it will report to Windows.</td>
</tr>
</tbody>
</table>
See Also

ContinuousPovMapping Class
PPJoy Namespace
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
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)**

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

**C#**

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

**Visual C++**

```cpp
[SerializableAttribute]
[ClassInterfaceAttribute(ClassInterfaceType::AutoDual)]
public ref class Device sealed
```
Inheritance Hierarchy

System...Object
PPJoy...Device
See Also

Device Members
PPJoy Namespace
The `Device` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td>Deletes this <a href="#">Device</a> from PPJoy.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>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.)</td>
</tr>
<tr>
<td><strong>GetMappings</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>RemoveMappings</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>SetMappings</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeviceType</strong></td>
<td>Gets the <strong>JoystickType</strong> of this <strong>Device</strong>.</td>
</tr>
<tr>
<td><strong>LptNum</strong></td>
<td>Gets the LPT number of this <strong>Device</strong>.</td>
</tr>
<tr>
<td><strong>ProductId</strong></td>
<td>Gets the Product ID associated with this <strong>Device</strong>.</td>
</tr>
<tr>
<td><strong>SubType</strong></td>
<td>Gets the <strong>JoystickSubType</strong> of this <strong>Device</strong>.</td>
</tr>
<tr>
<td><strong>UnitNum</strong></td>
<td>Gets the unit number of this <strong>Device</strong>.</td>
</tr>
<tr>
<td><strong>VendorId</strong></td>
<td>Gets the Vendor ID associated with this <strong>Device</strong>.</td>
</tr>
</tbody>
</table>
See Also

Device Class
PPJoy Namespace
PPJoy Wrapper Library reference library

Device Methods

See Also

The **Device** type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td>Deletes this <a href="#">Device</a> from PPJoy.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>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.)</td>
</tr>
<tr>
<td><strong>GetMappings</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>RemoveMappings</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>SetMappings</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

Device Class
PPJoy Namespace
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 true (True in Visual Basic), this Device's registration will be removed from DirectInput. If false (False in Visual Basic), the Device's DirectInput registration will not be removed.

removeDriver
Type: System:::Boolean
If true (True in Visual Basic), the Device's drivers will be unregistered from the system. If false (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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetMappings()</code></td>
<td>Gets a <code>MappingCollection</code> representing the controls defined on this <code>Device</code>.</td>
</tr>
<tr>
<td><code>GetMappings(JoystickMapScope)</code></td>
<td>Gets a <code>MappingCollection</code> representing the controls defined on this <code>Device</code>, or representing the controls defined in this <code>Device</code>'s interface.</td>
</tr>
</tbody>
</table>
See Also

Device Class
Device Members
PPJoy Namespace
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
GetMappings Method (JoystickMapScope)

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
PPJoy Wrapper Library reference library
Device...::RemoveMappings Method

Device Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveMappings()()()</td>
<td>Removes the custom-defined Mappings from this Device, without affecting the Mappings defined in the Device's interface.</td>
</tr>
<tr>
<td>RemoveMappings(JoystickMapScope) Mappings</td>
<td>Removes the custom-defined Mappings from this Device OR from its interface.</td>
</tr>
</tbody>
</table>
See Also

Device Class
Device Members
PPJoy Namespace
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
PPJoy Wrapper Library reference library
Device:::RemoveMappings Method (JoystickMapScope)

Remove 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
PPJoy Wrapper Library reference library
Device..SetMappings Method

Device Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetMappings(MappingCollection) MappingCollection</td>
<td>Associates a set of Mappings (a PPJoy Device).</td>
</tr>
<tr>
<td>SetMappings(JoystickMapScope, MappingCollection)</td>
<td>Associates a set of Mappings (a PPJoy Device in a specific JoystickMapScope).</td>
</tr>
</tbody>
</table>
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
 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
The **Device** type exposes the following properties.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceType</td>
<td>Gets the JoystickType of this Device.</td>
</tr>
<tr>
<td>LptNum</td>
<td>Gets the LPT number of this Device.</td>
</tr>
<tr>
<td>ProductId</td>
<td>Gets the Product ID associated with this Device.</td>
</tr>
<tr>
<td>SubType</td>
<td>Gets the JoystickSubType of this Device.</td>
</tr>
<tr>
<td>UnitNum</td>
<td>Gets the unit number of this Device.</td>
</tr>
<tr>
<td>VendorId</td>
<td>Gets the Vendor ID associated with this Device.</td>
</tr>
</tbody>
</table>
See Also

Device Class
PPJoy Namespace
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
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 Devices will have LptNum = 0.
See Also

Device Class
PPJoy Namespace
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
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
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
PPJoy Wrapper Library reference library

Device.

VendorId Property

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
DeviceAlreadyExistsException Class

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceAlreadyExistsException()()()</td>
<td>Initializes a new instance of the <code>DeviceAlreadyExistsException</code> class.</td>
</tr>
<tr>
<td>DeviceAlreadyExistsException(String)</td>
<td>Initializes a new instance of the <code>DeviceAlreadyExistsException</code> class.</td>
</tr>
<tr>
<td>DeviceAlreadyExistsException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the <code>DeviceAlreadyExistsException</code> class.</td>
</tr>
<tr>
<td>DeviceAlreadyExistsException(String, Exception)</td>
<td>Initializes a new instance of the <code>DeviceAlreadyExistsException</code> class.</td>
</tr>
</tbody>
</table>
See Also

DeviceAlreadyExistsException Class
DeviceAlreadyExistsException Members
PPJoy Namespace
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
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)

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

### C#

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

### Visual C++

```cpp
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
Initializes a new instance of the `DeviceAlreadyExistsException` class.

**Namespace:** [PPJoy](https://github.com/PPJoy/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
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 null
  Nothingnullptr a null reference (Nothing in Visual Basic) if no inner
  exception is specified.
See Also

DeviceAlreadyExistsException Class
DeviceAlreadyExistsException Overload
PPJoy Namespace
PPJoy Wrapper Library reference library
DeviceAlreadyExistsException Members

The **DeviceAlreadyExistsException** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceAlreadyExistsException</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

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

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

DeviceAlreadyExistsException Class
PPJoy Namespace
PPJoy Wrapper Library reference library
DeviceManager Class

Provides methods for creating, retrieving, deleting, and managing details of PPJoy Device objects.

Namespace: PPJoy
Assembly: PPJoyWrapper (in PPJoyWrapper.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<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
Create 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
The **DeviceManager** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceManager</td>
<td>Creates an instance of the DeviceManager class which can manage PPJoy Devices.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateDevice</td>
<td>Creates and registers a new joystick Device with PPJoy.</td>
</tr>
<tr>
<td>DeleteAllDevices</td>
<td>Deletes all registered PPJoy joystick Devices.</td>
</tr>
<tr>
<td>DeleteDevice</td>
<td>Deletes a Device from PPJoy.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetAllDevices</td>
<td>Enumerates all defined PPJoy Devices.</td>
</tr>
<tr>
<td>GetDevice</td>
<td>Retrieves a Device object that matches the specified query parameters.</td>
</tr>
<tr>
<td>GetDeviceByProductId</td>
<td>Gets a Device object representing the PPJoy device whose product ID matches the supplied Product ID.</td>
</tr>
<tr>
<td>GetDeviceMappings</td>
<td>Gets a MappingCollection representing the controls mappings associated with a defined PPJoy device.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>IsVirtualDevice</td>
<td>Checks whether a given integer containing a VendorID/ProductID combination (typically obtained from DirectInput) refers to a virtual Device or a physical Device.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MaxValidUnitNumber</td>
<td>Gets the maximum valid unit number for a given joystick type.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>RemoveDeviceMappings</td>
<td>Removes the mappings from a specific PPJoy device or the default mappings</td>
</tr>
<tr>
<td></td>
<td>from its interface.</td>
</tr>
<tr>
<td>SetDeviceMappings</td>
<td>Sets the mappings for a specific PPJoy device.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdealMappings</td>
<td>Gets a custom <a href="#">MappingCollection</a> that defines the broadest possible set of controls that can be assigned to a PPJoy <a href="#">Device</a>. The controls are pre-set to expose the maximum capabilities that a virtual joystick <a href="#">Device</a> can express.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateDevice</td>
<td>Creates and registers a new joystick Device with PPJoy.</td>
</tr>
<tr>
<td>DeleteAllDevices</td>
<td>Deletes all registered PPJoy joystick Devices.</td>
</tr>
<tr>
<td>DeleteDevice</td>
<td>Deletes a Device from PPJoy.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetAllDevices</td>
<td>Enumerates all defined PPJoy Devices.</td>
</tr>
<tr>
<td>GetDevice</td>
<td>Retrieves a Device object that matches the specified query parameters.</td>
</tr>
<tr>
<td>GetDeviceByProductId</td>
<td>Gets a Device object representing the PPJoy device whose product ID matches the supplied Product ID.</td>
</tr>
<tr>
<td>GetDeviceMappings</td>
<td>Gets a MappingCollection representing the controls mappings associated with a defined PPJoy device.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. GetHashCode()()() is suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>IsVirtualDevice</td>
<td>Checks whether a given integer containing a VendorID/ProductID combination (typically obtained from DirectInput) refers to a virtual Device or a physical Device.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>MaxValidUnitNumber</td>
<td>Gets the maximum valid unit number for a given joystick type.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>RemoveDeviceMappings</td>
<td>Removes the mappings from a specific PPJoy device or the default mappings from its interface.</td>
</tr>
<tr>
<td>SetDeviceMappings</td>
<td>Sets the mappings for a specific PPJoy device.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

DeviceManager Class
PPJoy Namespace
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
 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 trueTrueTrueTrue (True in Visual Basic), each **Device**'s registration will be removed from DirectInput. If falseFalseFalseFalse (False in Visual Basic), no **Device**'s DirectInput registration will be removed.

removeDriver
   Type: System::Boolean
   If trueTrueTrueTrue (True in Visual Basic), each **Device**'s drivers will be unregistered from the system. If falseFalseFalseFalse (False in Visual Basic), no **Device**'s drivers will be unregistered from the system.
See Also

DeviceManager Class
PPJoy Namespace
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 true (True in Visual Basic), the Device's registration will be removed from DirectInput. If false (False in Visual Basic), the Device's DirectInput registration will not be removed.
removeDriver

Type: System::Boolean

If true (True in Visual Basic), the Device's drivers will be unregistered from the system. If false (False in Visual Basic), the Device's drivers will not be unregistered from the system.
See Also

DeviceManager Class
PPJoy Namespace
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
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 null
reference (Nothing in Visual Basic), if no matching Device is found.
See Also

DeviceManager Class
PPJoy Namespace
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.d11)
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
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
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

true (True in Visual Basic), if the Device matching the specified vendorIdentityProductId is a PPJoy virtual Device, or false (False in Visual Basic) if it is a physical Device.
See Also

DeviceManager Class
PPJoy Namespace
PPJoy Wrapper Library reference library
DeviceManager::MaxValidUnitNumber Method

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: PJPJoy:::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
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
PPJoy Wrapper Library reference library

DeviceManager...::SetDeviceMappings Method

See Also

DeviceManager Class

Sets the mappings for a specific PPJoy device.

Namespace: PPJoy
Assembly: PPJoyWrapper (in PPJoyWrapper.dll)
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdealMappings</td>
<td>Gets a custom <a href="#">MappingCollection</a> that defines the broadest possible set of controls that can be assigned to a PPJoy <a href="#">Device</a>. The controls are pre-set to expose the maximum capabilities that a virtual joystick <a href="#">Device</a> can express.</td>
</tr>
</tbody>
</table>
See Also

DeviceManager Class
PPJoy Namespace
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
DeviceNotFoundException Class

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
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DeviceNotFoundException()</code></td>
<td>Initializes a new instance of the <code>DeviceNotFoundException</code> class.</td>
</tr>
<tr>
<td><code>DeviceNotFoundException(String)</code></td>
<td>Initializes a new instance of the <code>DeviceNotFoundException</code> class.</td>
</tr>
<tr>
<td><code>DeviceNotFoundException(SerializationInfo, StreamingContext)</code></td>
<td>Initializes a new instance of the <code>DeviceNotFoundException</code> class.</td>
</tr>
<tr>
<td><code>DeviceNotFoundException(String, Exception)</code></td>
<td>Initializes a new instance of the <code>DeviceNotFoundException</code> class.</td>
</tr>
</tbody>
</table>
See Also

DeviceNotFoundException Class
DeviceNotFoundException Members
PPJoy Namespace
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
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
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
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)**

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

**C#**

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

**Visual C++**

```cpp
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 null if no inner exception is specified.
See Also

DeviceNotFoundException Class
DeviceNotFoundException Overload
PPJoy Namespace
The `DeviceNotFoundException` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceNotFoundException</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

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

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

DeviceNotFoundException Class
PPJoy Namespace
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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital0</td>
<td></td>
</tr>
<tr>
<td>Digital1</td>
<td></td>
</tr>
<tr>
<td>Digital2</td>
<td></td>
</tr>
<tr>
<td>Digital3</td>
<td></td>
</tr>
<tr>
<td>Digital4</td>
<td></td>
</tr>
<tr>
<td>Digital5</td>
<td></td>
</tr>
<tr>
<td>Digital6</td>
<td></td>
</tr>
<tr>
<td>Digital7</td>
<td></td>
</tr>
<tr>
<td>Digital8</td>
<td></td>
</tr>
<tr>
<td>Digital9</td>
<td></td>
</tr>
<tr>
<td>Digital10</td>
<td></td>
</tr>
<tr>
<td>Digital11</td>
<td></td>
</tr>
<tr>
<td>Digital12</td>
<td></td>
</tr>
<tr>
<td>Digital13</td>
<td></td>
</tr>
<tr>
<td>Digital14</td>
<td></td>
</tr>
<tr>
<td>Digital15</td>
<td></td>
</tr>
<tr>
<td>Digital16</td>
<td></td>
</tr>
<tr>
<td>Digital17</td>
<td></td>
</tr>
<tr>
<td>Digital18</td>
<td></td>
</tr>
<tr>
<td>Digital19</td>
<td></td>
</tr>
<tr>
<td>Digital20</td>
<td></td>
</tr>
<tr>
<td>Digital21</td>
<td></td>
</tr>
<tr>
<td>Digital22</td>
<td></td>
</tr>
<tr>
<td>Digital23</td>
<td></td>
</tr>
<tr>
<td>Digital24</td>
<td></td>
</tr>
<tr>
<td>Digital25</td>
<td></td>
</tr>
<tr>
<td>Digital26</td>
<td></td>
</tr>
<tr>
<td>Digital27</td>
<td></td>
</tr>
</tbody>
</table>
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
| 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
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
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirectionalPovMapping()()</td>
<td>Creates a new <code>DirectionalPovMapping</code> object.</td>
</tr>
<tr>
<td>DirectionalPovMapping(Int32)</td>
<td>Creates a new <code>DirectionalPovMapping</code> object.</td>
</tr>
</tbody>
</table>
See Also

DirectionalPovMapping Class
DirectionalPovMapping Members
PPJoy Namespace
PPJoy Wrapper Library reference library
DirectionalPovMapping Constructor

**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
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

countNumber

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 countNumber of 0, the second PovMapping will have a countNumber of 1, and so forth.
See Also

DirectionalPovMapping Class
DirectionalPovMapping Overload
PPJoy Namespace
Mapping...:::ControlNumber
PPJoy...:::PovMapping
DirectionalPovMapping Members

The **DirectionalPovMapping** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirectionalPovMapping</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ControlNumber</strong></td>
<td>Gets/setsthe index to use for this control in the collection of all other controls of the same type on the same device. (Inherited from <a href="#">Mapping</a>.)</td>
</tr>
<tr>
<td><strong>EastDataSource</strong></td>
<td>Gets/sets the <a href="#">DirectionalPovDataSource</a> that this <a href="#">DirectionalPovMapping</a> will use to determine that it should report to Windows that it is being pressed in the <strong>East</strong> direction.</td>
</tr>
<tr>
<td><strong>NorthDataSource</strong></td>
<td>Gets/sets the <a href="#">DirectionalPovDataSource</a> that this <a href="#">DirectionalPovMapping</a> will use to determine that it should report to Windows that it is being pressed in the <strong>North</strong> direction.</td>
</tr>
<tr>
<td><strong>SouthDataSource</strong></td>
<td>Gets/sets the <a href="#">DirectionalPovDataSource</a> that this <a href="#">DirectionalPovMapping</a> will use to determine that it should report to Windows that it is being pressed in the <strong>South</strong> direction.</td>
</tr>
<tr>
<td><strong>WestDataSource</strong></td>
<td>Gets/sets the <a href="#">DirectionalPovDataSource</a> that this <a href="#">DirectionalPovMapping</a> will use to determine that it should report to Windows that it is being pressed in the <strong>West</strong> direction.</td>
</tr>
</tbody>
</table>
See Also

DirectionalPovMapping Class
PPJoy Namespace
The **DirectionalPovMapping** type exposes the following properties.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlNumber</td>
<td>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.)</td>
</tr>
<tr>
<td>EastDataSource</td>
<td>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.</td>
</tr>
<tr>
<td>NorthDataSource</td>
<td>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.</td>
</tr>
<tr>
<td>SouthDataSource</td>
<td>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.</td>
</tr>
<tr>
<td>WestDataSource</td>
<td>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.</td>
</tr>
</tbody>
</table>
See Also

DirectionalPovMapping Class
PPJoy Namespace
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 **DirectionalPovDataSources**'s value is True (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital **DirectionalPovDataSource**'s value is False (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
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 DirectionalPovDataSources's value is true (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital DirectionalPovDataSource's value is false (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
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 **DirectionalPovDataSources**'s value is true (True in Visual Basic). Similarly, a value of **centered** will be reported to Windows, whenever the Digital **DirectionalPovDataSource**'s value is false (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
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 DirectionalPovDataSources's value is true (True in Visual Basic). Similarly, a value of centered will be reported to Windows, whenever the Digital DirectionalPovDataSource's value is false (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
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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>The <a href="#">MappingCollection</a> applies to all devices on the same LPT number (i.e. all virtual devices or all physical parallel-port devices on the same LPT port).</td>
</tr>
<tr>
<td>Device</td>
<td>The <a href="#">MappingCollection</a> 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.</td>
</tr>
</tbody>
</table>
See Also

PPJoy Namespace
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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genesis_Pad_A_B_C_Start</td>
<td></td>
</tr>
<tr>
<td>Genesis_Pad_A_B_C_X_Y_Z_Start_Mode</td>
<td></td>
</tr>
<tr>
<td>SNES_or_Virtual_Gameboy</td>
<td></td>
</tr>
<tr>
<td>NES</td>
<td></td>
</tr>
<tr>
<td>NotApplicable</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PPJoy Namespace
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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joystick_TheMaze</td>
<td></td>
</tr>
<tr>
<td>Joystick_IanHerries</td>
<td></td>
</tr>
<tr>
<td>Joystick_TurboGraFX</td>
<td></td>
</tr>
<tr>
<td>Joystick_Linux_v0802</td>
<td></td>
</tr>
<tr>
<td>Joystick_Linux_DB9c</td>
<td></td>
</tr>
<tr>
<td>Joystick_TorMod</td>
<td></td>
</tr>
<tr>
<td>Joystick_DirectPad_Pro</td>
<td></td>
</tr>
<tr>
<td>Joystick_TurboGraFX_SwappedButtons</td>
<td></td>
</tr>
<tr>
<td>Joystick_LPT_JoyStick</td>
<td></td>
</tr>
<tr>
<td>Joystick_CHAMPgames</td>
<td></td>
</tr>
<tr>
<td>Joystick_STFormat</td>
<td></td>
</tr>
<tr>
<td>Joystick_SNESKey2600</td>
<td></td>
</tr>
<tr>
<td>Joystick_Amiga_4_Player</td>
<td></td>
</tr>
<tr>
<td>Joystick_PCAE</td>
<td></td>
</tr>
<tr>
<td>Genesis_Pad_Linux</td>
<td></td>
</tr>
<tr>
<td>Genesis_Pad_DirectPad_Pro</td>
<td></td>
</tr>
<tr>
<td>Genesis_Pad_NTPad_XP</td>
<td></td>
</tr>
<tr>
<td>SNES_or_NESPad_Linux</td>
<td></td>
</tr>
<tr>
<td>SNES_or_NESPad_DirectPadPro_Or_SNESKey</td>
<td></td>
</tr>
<tr>
<td>Genesis_Pad_ConsoleCable</td>
<td></td>
</tr>
<tr>
<td>Genesis_Pad_SNESKey</td>
<td></td>
</tr>
<tr>
<td>Playstation_Pad_PSXPBLib</td>
<td></td>
</tr>
<tr>
<td>Playstation_Pad_DirectPad_Pro</td>
<td></td>
</tr>
<tr>
<td>Playstation_Pad_Linux</td>
<td></td>
</tr>
<tr>
<td>Playstation_Pad_NTPad_XP</td>
<td></td>
</tr>
<tr>
<td>Playstation_Pad_Megatap</td>
<td></td>
</tr>
<tr>
<td>Virtual_Joystick</td>
<td></td>
</tr>
<tr>
<td>Joystick_Linux_gamecon</td>
<td></td>
</tr>
</tbody>
</table>
Joystick_LPTswitch
Radio_Control_TX
SNES_or_NESPad_PowerPad
Genesis_Pad_DirectPad_Pro_V6
See Also

PPJoy Namespace
PPJoy::::Device
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 accessable 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Mapping()</code></td>
<td>Creates a new Mapping object.</td>
</tr>
<tr>
<td><code>Mapping(Int32)</code></td>
<td>Creates a new Mapping object.</td>
</tr>
</tbody>
</table>
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
PPJoy Wrapper Library reference library
Mapping Constructor (Int32)

Creates a new Mapping object.

**Namespace:** PPJoy

**Assembly:** PPJoyWrapper (in PPJoyWrapper.dll)
### Syntax

**Visual Basic (Declaration)**

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

**C#**

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

**Visual C++**

```cpp
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
The **Mapping** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>

---
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>GetHashCode()() is suitable for use in hashing algorithms and data</td>
</tr>
<tr>
<td></td>
<td>structures like a hash table.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>ControlNumber</code></td>
<td>Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device.</td>
</tr>
</tbody>
</table>
See Also

Mapping Class
PPJoy Namespace
PPJoy Wrapper Library reference library

Mapping Properties

See Also

The Mapping type exposes the following properties.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlNumber</td>
<td>Gets/sets the index to use for this control in the collection of all other controls of the same type on the same device.</td>
</tr>
</tbody>
</table>
See Also

Mapping Class
PPJoy Namespace
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
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
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
The MappingCollection type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MappingCollection</td>
<td>Creates a new MappingCollection object.</td>
</tr>
</tbody>
</table>
## Methods

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AxisMappings</strong></td>
<td>Gets a <a href="#">MappingCollection</a> containing all the AxisMappings contained in this MappingCollection. Gets a <a href="#">MappingCollection</a> containing all the AxisMappings contained in this MappingCollection.</td>
</tr>
<tr>
<td><strong>ButtonMappings</strong></td>
<td>Gets a <a href="#">MappingCollection</a> containing all the ButtonMapping objects contained in this MappingCollection.</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Gets the number of <a href="#">Mappings</a> in the <a href="#">MappingCollection</a>.</td>
</tr>
<tr>
<td><strong>IsFixedSize</strong></td>
<td>Gets a value indicating whether the <a href="#">MappingCollection</a> has a fixed size.</td>
</tr>
<tr>
<td><strong>IsReadOnly</strong></td>
<td>Gets a value indicating whether the <a href="#">MappingCollection</a> is read-only.</td>
</tr>
<tr>
<td><strong>IsSynchronized</strong></td>
<td>Gets a value indicating whether access to the <a href="#">MappingCollection</a> is synchronized (thread safe).</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Gets an item from the <a href="#">MappingCollection</a>, given its index number.</td>
</tr>
<tr>
<td><strong>PovMappings</strong></td>
<td>Gets a <a href="#">MappingCollection</a> containing all the PovMappings contained in this MappingCollection.</td>
</tr>
<tr>
<td><strong>SyncRoot</strong></td>
<td>Gets an object that can be used to synchronize access to the <a href="#">MappingCollection</a>.</td>
</tr>
</tbody>
</table>
## Explicit Interface Implementations

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IList::Item</code></td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
PPJoy Namespace
See Also

The **MappingCollection** type exposes the following methods.
## Methods

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

MappingCollection Class
PPJoy Namespace
PPJoy Wrapper Library reference library
MappingCollection:::Add Method
MappingCollection Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(Object)</td>
<td>Adds a Object (referencing a Mapping object) to the MappingCollection.</td>
</tr>
<tr>
<td>Add(Mapping)</td>
<td>Adds a Mapping to the MappingCollection.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
MappingCollection Members
PPJoy Namespace
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
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
Removes all elements from the MappingCollection.

Namespace: PPJoy
Assembly: PPJoyWrapper (in PPJoyWrapper.dll)
### Syntax

**Visual Basic (Declaration)**

```vbnet
Public Sub Clear
```

**C#**

```csharp
public void Clear()
```

**Visual C++**

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

**Implements**

```csharp
IList<>::Clear()
```
See Also

MappingCollection Class
PPJoy Namespace
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#**

```csharp
public Object Clone()
```

**Visual C++**

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

**Return Value**

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

**Implements**

ICloneable....::Clone()()
See Also

MappingCollection Class
PPJoy Namespace
PPJoy Wrapper Library reference library
MappingCollection..::.Contains Method
MappingCollection Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(Object)</td>
<td>Determines whether an element is in the MappingCollection.</td>
</tr>
<tr>
<td>Contains(Mapping)</td>
<td>Determines whether a specific Mapping is in the MappingCollection.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
MappingCollection Members
PPJoy Namespace
Determines whether a specific Mapping is in the MappingCollection.

Namespace: PPJoy
Assembly: PPJoyWrapper (in PPJoyWrapper.dll)
Syntax

**Visual Basic (Declaration)**

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

**C#**

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

**Visual C++**

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

**Parameters**

value

*Type:* PPJoy::::Mapping

The Mapping to locate in the MappingCollection. value can be null (Nothing in Visual Basic).

**Return Value**

true (True in Visual Basic) if the MappingCollection contains the specified Mapping, or false (False in Visual Basic) if it does not.
See Also

MappingCollection Class
Contains Overload
PPJoy Namespace
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)**

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

**C#**

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

**Visual C++**

```cpp
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 null (Nothing in Visual Basic).

**Return Value**

true (True in Visual Basic) if the MappingCollection contains the specified Object, or false (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
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyTo(Array, Int32)</td>
<td>Copies the members of the <a href="#">MappingCollection</a> to an array.</td>
</tr>
<tr>
<td>CopyTo(array&lt;Mapping&gt;[J][J], Int32)</td>
<td>Copies the members of the <a href="#">MappingCollection</a> to an array.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
MappingCollection Members
PPJoy Namespace
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
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
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 IEnumerator

**C#**

public IEnumerator GetEnumerator()

**Visual C++**

public:
virtual IEnumerator GetEnumerator() sealed

Return Value

Implements

IEnumerator...GetEnumerator()()
See Also

MappingCollection Class
PPJoy Namespace
PPJoy Wrapper Library reference library
MappingCollection..::.IndexOf Method

MappingCollection Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IndexOf(Object)</code></td>
<td>Searches for the specified Object and returns the zero-based index of the first occurrence within the entire <code>MappingCollection</code>.</td>
</tr>
<tr>
<td><code>IndexOf(Mapping)</code></td>
<td>Searches for the specified <code>Mapping</code> and returns the zero-based index of the first occurrence within the entire <code>MappingCollection</code>.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
MappingCollection Members
PPJoy Namespace
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 null/Nothing/nullptr 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
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 occurence within the entire MappingCollection.

Namespace:  PPJoy
Assembly:  PPJoyWrapper (in PPJoyWrapper.dll)
Syntax

**Visual Basic (Declaration)**

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

**C#**

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

**Visual C++**

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

**Parameters**

`value`

Type: System::Object

The Object to locate in the `MappingCollection`. value can be `null`/`Nothing/nullptra` 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert(Int32, Mapping)</td>
<td>Inserts a Mapping into the MappingCollection at the specified index.</td>
</tr>
<tr>
<td>Insert(Int32, Object)</td>
<td>Inserts an element into the MappingCollection at the specified index.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
MappingCollection Members
PPJoy Namespace
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 nll.
See Also

MappingCollection Class
Insert Overload
PPJoy Namespace
Inserts an element into the MappingCollection at the specified index.

**Namespace:** [PPJoy](https://www.example.com/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 nullNothingnullptra 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Remove(Object)</code></td>
<td>Removes the first occurrence of a specific Object from the <code>MappingCollection</code>.</td>
</tr>
<tr>
<td><code>Remove(Mapping)</code></td>
<td>Removes a <code>Mapping</code> from the <code>MappingCollection</code>.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
MappingCollection Members
PPJoy Namespace
PPJoy Wrapper Library reference library
MappingCollection::Remove Method (Mapping)

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
Removes the first occurrence of a specific Object from the `MappingCollection`.

**Namespace:** PPJoy  
**Assembly:** PPJoyWrapper (in PPJoyWrapper.dll)
**Syntax**

**Visual Basic (Declaration)**

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

**C#**

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

**Visual C++**

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

**Parameters**

value

Type: System::Object

The Object to remove from the MappingCollection. value can be null!Nothing!nullptr null reference (Nothing in Visual Basic).

**Implements**

IList:::Remove(Object)
See Also

MappingCollection Class
Remove Overload
PPJoy Namespace
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
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
The `MappingCollection` type exposes the following properties.
## Properties

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IList::Item</code></td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
PPJoy Namespace
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
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#

color MappingCollection ButtonMappings { get; }

Visual C++

public:

property MappingCollection^ ButtonMappings {
    MappingCollection^ get ();
}

See Also

MappingCollection Class
PPJoy Namespace
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
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)])]
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System::.ArgumentOutOfRangeException</code></td>
<td>index is not a valid index in the <code>IList</code>.</td>
</tr>
<tr>
<td><code>System::.NotSupportedException</code></td>
<td>The property is set and the <code>IList</code> is read-only.</td>
</tr>
</tbody>
</table>
See Also

MappingCollection Class
PPJoy Namespace
PPJoy Wrapper Library reference library
MappingCollection..::.FixedSize 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
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
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

IICollection::IsSynchronized
See Also

MappingCollection Class
PPJoy Namespace
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
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
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
OperationFailedException Class

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
OperationFailedException Constructor

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperationFailedException()()</td>
<td>Initializes a new instance of the OperationFailedException class.</td>
</tr>
<tr>
<td>OperationFailedException(String)</td>
<td>Initializes a new instance of the OperationFailedException class.</td>
</tr>
<tr>
<td>OperationFailedException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the OperationFailedException class.</td>
</tr>
<tr>
<td>OperationFailedException(String, Exception)</td>
<td>Initializes a new instance of the OperationFailedException class.</td>
</tr>
</tbody>
</table>
See Also

OperationFailedException Class
OperationFailedException Members
PPJoy Namespace
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
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](#)
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)**

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

**C#**

```csharp
public OperationFailedException(
    string message
)
```

**Visual C++**

```cpp
public:
OperationFailedException(
    String^ message
)
```

**Parameters**

**message**
- Type: System....String
- The message that describes the error.
See Also

OperationFailedException Class
OperationFailedException Overload
PPJoy Namespace
Initializes a new instance of the `OperationFailedException` class.

**Namespace:** [PPJoy](https://www.ppjoy.com)

**Assembly:** PPJoyWrapper (in PPJoyWrapper.dll)
### Syntax

#### Visual Basic (Declaration)

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

#### C#

```csharp
public OperationFailedException(
    string message,
    Exception inner
)
```

#### Visual C++

```cpp
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 `null`/`Nothing`/`nullptr` null reference (Nothing in Visual Basic) if no inner exception is specified.
See Also

OperationFailedException Class
OperationFailedException Overload
PPJoy Namespace
The **OperationFailedException** type exposes the following members.
## Constructors

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

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

OperationFailedException Class
PPJoy Namespace
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PovMapping()()</td>
<td>Creates a new <code>PovMapping</code> object.</td>
</tr>
<tr>
<td>PovMapping(Int32)</td>
<td>Creates a new <code>PovMapping</code> object.</td>
</tr>
</tbody>
</table>
See Also

PovMapping Class
PovMapping Members
PPJoy Namespace
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
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

ccontrolNumber
    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
The **PovMapping** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PovMapping</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>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.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlNumber</td>
<td>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.)</td>
</tr>
</tbody>
</table>
See Also

PovMapping Class
PPJoy Namespace
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPJoyException()()</td>
<td>Initializes a new instance of the PPJoyException class.</td>
</tr>
<tr>
<td>PPJoyException(String)</td>
<td>Initializes a new instance of the PPJoyException class.</td>
</tr>
<tr>
<td>PPJoyException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the PPJoyException class.</td>
</tr>
<tr>
<td>PPJoyException(String, Exception)</td>
<td>Initializes a new instance of the PPJoyException class.</td>
</tr>
</tbody>
</table>
See Also

PPJoyException Class
PPJoyException Members
PPJoy Namespace
PPJoy Exception Constructor

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
PPJoy Exception 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
PPJoy Exception Constructor (String)

PPJoy Exception 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
PPJoy Exception 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 null reference (Nothing in Visual Basic) if no inner
Exception is specified.
See Also

PPJoyException Class
PPJoyException Overload
PPJoy Namespace
The **PPJoyException** type exposes the following members.
## Constructors

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

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

PPJoyException Class
PPJoy Namespac
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualJoystick()()</td>
<td>Creates a new <code>VirtualJoystick</code> instance.</td>
</tr>
<tr>
<td>VirtualJoystick(Int32)</td>
<td>Constructs a new <code>VirtualJoystick</code> instance.</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
VirtualJoystick Members
PPJoy Namespace
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
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
The `VirtualJoystick` type exposes the following fields.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxAnalogDataSources</strong></td>
<td>The maximum number of analog data sources supported by PPJoy on a single Device.</td>
</tr>
<tr>
<td><strong>MaxAnalogDataSourceVal</strong></td>
<td>The maximum value that can be applied to an analog data source. The maximum number of digital data sources supported by PPJoy on a single Device.</td>
</tr>
<tr>
<td><strong>MaxDigitalDataSources</strong></td>
<td>The maximum number of virtual joystick Devices supported by PPJoy.</td>
</tr>
<tr>
<td><strong>MaxVirtualDevices</strong></td>
<td>The maximum number of axes that can be created on a PPJoy virtual joystick Device.</td>
</tr>
<tr>
<td><strong>MaxVisibleAxes</strong></td>
<td>The maximum number of buttons that can be created on a PPJoy virtual joystick Device.</td>
</tr>
<tr>
<td><strong>MaxVisibleButtons</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MaxVisiblePovs</strong></td>
<td>The maximum number of POVs that can be created on a PPJoy virtual joystick Device.</td>
</tr>
<tr>
<td><strong>MinAnalogDataSourceVal</strong></td>
<td>The minimum value that can be applied to an analog data source (except PovCentered).</td>
</tr>
<tr>
<td><strong>PovCentered</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
PPJoy Namespace
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
VirtualJoystick::MaxAnalogDataSourceVal Field

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
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
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
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
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
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
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
The `VirtualJoystick` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualJoystick</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dispose</td>
<td>Public implementation of the Dispose() method.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>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.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.                                                                                              (Inherited from Object.)</td>
</tr>
<tr>
<td>SendUpdates</td>
<td>Sends all pending updates to PPJoy.</td>
</tr>
<tr>
<td>SetAnalogDataSourceValue</td>
<td>Sets an individual Analog data source value.</td>
</tr>
<tr>
<td>SetDigitalDataSourceState</td>
<td>Sets an individual Digital data source state.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.                                                                                       (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MaxAnalogDataSources</td>
<td>The maximum number of analog data sources supported by PPJoy on a single Device.</td>
</tr>
<tr>
<td>MaxAnalogDataSourceVal</td>
<td>The maximum value that can be applied to an analog data source.</td>
</tr>
<tr>
<td>MaxDigitalDataSources</td>
<td>The maximum number of digital data sources supported by PPJoy on a single Device.</td>
</tr>
<tr>
<td>MaxVirtualDevices</td>
<td>The maximum number of virtual joystick Devices supported by PPJoy.</td>
</tr>
<tr>
<td>MaxVisibleAxes</td>
<td>The maximum number of axes that can be created on a PPJoy virtual joystick Device.</td>
</tr>
<tr>
<td>MaxVisibleButtons</td>
<td>The maximum number of buttons that can be created on a PPJoy virtual joystick Device.</td>
</tr>
<tr>
<td>MaxVisiblePovs</td>
<td>The maximum number of POVs that can be created on a PPJoy virtual joystick Device.</td>
</tr>
<tr>
<td>MinAnalogDataSourceVal</td>
<td>The minimum value that can be applied to an analog data source (except PovCentered).</td>
</tr>
<tr>
<td>PovCentered</td>
<td>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.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualStickNumber</td>
<td>Gets/sets the PPJoy virtual Device number that this VirtualJoystick instance is managing.</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
PPJoy Namespace
The VirtualJoystick type exposes the following methods.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Public implementation of the Dispose()()() method.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>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.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Sets an individual Analog data source value.</td>
</tr>
<tr>
<td>SendUpdates</td>
<td>Sends all pending updates to PPJoy.</td>
</tr>
<tr>
<td>SetAnalogDataSourceValue</td>
<td>Sets an individual Digital data source state.</td>
</tr>
<tr>
<td>SetDigitalDataSourceState</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
PPJoy Namespace
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
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
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)**

```vbnet
Public Sub SetAnalogDataSourceValue ( _
    dataSourceNum As Integer, _
    newValue As Integer _
)
```

**C#**

```csharp
public void SetAnalogDataSourceValue(
    int dataSourceNum,
    int newValue
)
```

**Visual C++**

```cpp
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()() method is called. This allows multiple data source value updates to be passed to the PPJoy driver in a single pass.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System..::.ArgumentOutOfRangeException</td>
<td>Thrown if the dataSourceNum param $&lt; 1$ or $&gt; MaxAnalogDataSources$; also</td>
</tr>
<tr>
<td></td>
<td>thrown if the newValue argument is $&lt; \text{MinAnalogDataSourceVal}$ or $&gt;$</td>
</tr>
<tr>
<td></td>
<td>$\text{MaxAnalogDataSourceVal}$</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
PPJoy Namespace
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 `SendUpdates()` method gets called. This allows multiple digital data source state updates to be passed to the PPJoy driver in a single pass.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System..::.ArgumentOutOfRangeException</td>
<td>Thrown if the dataSourceNum argument (&lt; 1 ) or (&gt; ) (\max\text{Digital\text{DataSources}})</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
PPJoy Namespace
The VirtualJoystick type exposes the following properties.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualStickNumber</td>
<td>Gets/sets the PPJoy virtual Device number that this VirtualJoystick instance is managing.</td>
</tr>
</tbody>
</table>
See Also

VirtualJoystick Class
PPJoy Namespace
Virtual Joystick...::.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