[This is preliminary documentation and is subject to change.]
### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Calculator</strong></td>
<td>Encapsulates common mathematical functions.</td>
</tr>
<tr>
<td><strong>Guard</strong></td>
<td>Defines helper methods for validating arguments passed into methods.</td>
</tr>
<tr>
<td><strong>ParticleEffect</strong></td>
<td>Defines the root of a particle effect hierarchy.</td>
</tr>
<tr>
<td><strong>RandomHelper</strong></td>
<td>Defines helper methods for choosing random numbers of performing random operations.</td>
</tr>
</tbody>
</table>
## Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Particle</strong></td>
<td>Defines the data structure for a particle.</td>
</tr>
<tr>
<td><strong>VariableFloat</strong></td>
<td>Defines a floating point object which has a definable random variation.</td>
</tr>
<tr>
<td><strong>VariableFloat3</strong></td>
<td>Defines a Vector3 object which has a definable random variation.</td>
</tr>
</tbody>
</table>
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlendMode</td>
<td>Enumerates the possible blending modes for an emitter.</td>
</tr>
</tbody>
</table>

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Enumerates the possible blending modes for an emitter.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public enum BlendMode

Visual Basic (Declaration)
Public Enumeration BlendMode

Visual C++
public enum class BlendMode
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No blending.</td>
</tr>
<tr>
<td>Add</td>
<td>Additive blending.</td>
</tr>
<tr>
<td>Alpha</td>
<td>Alpha blending.</td>
</tr>
<tr>
<td>Subtract</td>
<td>Subtractive blending.</td>
</tr>
<tr>
<td>Multiply</td>
<td>Multiplicative blending.</td>
</tr>
</tbody>
</table>
See Also

ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Encapsulates common mathematical functions.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static class Calculator

Visual Basic (Declaration)

Public NotInheritable Class Calculator

Visual C++

public ref class Calculator abstract sealed
Inheritance Hierarchy

System...:::Object
ProjectMercury...:::Calculator
See Also

Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Calculator type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Acos</td>
<td>Returns the angle whose cosine is the specified value.</td>
</tr>
<tr>
<td>Asin</td>
<td>Returns the angle whose sine is the specified value.</td>
</tr>
<tr>
<td>Atan</td>
<td>Returns the angle whose tangent is the specified number.</td>
</tr>
<tr>
<td>Atan2</td>
<td>Returns the angle whose tangent is the quotient of the two specified numbers.</td>
</tr>
<tr>
<td>Clamp</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Cos</td>
<td>Returns the cosine of the specified angle.</td>
</tr>
<tr>
<td>Cosh</td>
<td>Returns the hyperbolic cosine of the specified angle.</td>
</tr>
<tr>
<td>CubicInterpolate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LinearInterpolate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Log</td>
<td>Returns the natural (base e) logarithm of the specified value.</td>
</tr>
<tr>
<td>Max</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Min</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Pow</td>
<td>Returns the specified value raised to the specified power.</td>
</tr>
<tr>
<td>Sin</td>
<td>Returns the sine of the specified angle.</td>
</tr>
<tr>
<td>Sinh</td>
<td>Returns the hyperbolic sine of the specified angle.</td>
</tr>
<tr>
<td>Sqr</td>
<td>Returns the square root of the specified value.</td>
</tr>
<tr>
<td>Tan</td>
<td>Returns the tangent of the specified angle.</td>
</tr>
<tr>
<td>Tanh</td>
<td>Returns the hyperbolic tangent of the specified angle.</td>
</tr>
<tr>
<td>Wrap</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi</td>
<td>Represents the value of pi.</td>
</tr>
<tr>
<td>PiOver2</td>
<td>Represents the value of pi divided by two.</td>
</tr>
<tr>
<td>PiOver4</td>
<td>Represents the value of pi divided by four.</td>
</tr>
<tr>
<td>TwoPi</td>
<td>Represents the value of pi times two.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **Calculator** type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi</td>
<td>Represents the value of pi.</td>
</tr>
<tr>
<td>PiOver2</td>
<td>Represents the value of pi divided by two.</td>
</tr>
<tr>
<td>PiOver4</td>
<td>Represents the value of pi divided by four.</td>
</tr>
<tr>
<td>TwoPi</td>
<td>Represents the value of pi times two.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Represents the value of pi.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
Version: 3.1.0.0
## Syntax

**C#**

public const float Pi

**Visual Basic (Declaration)**

Public Const Pi As Single

**Visual C++**

public:
    literal float Pi
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Represents the value of pi divided by two.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public const float PiOver2

Visual Basic (Declaration)

Public Const PiOver2 As Single

Visual C++

public:
    literal float PiOver2
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Represents the value of pi divided by four.

**Namespace:** [ProjectMercury](http://www.projectmercury.com)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public const float PiOver4

Visual Basic (Declaration)

Public Const PiOver4 As Single

Visual C++

public:
literal float PiOver4
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Represents the value of pi times two.

**Namespace:**  [ProjectMercury](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public const float TwoPi

Visual Basic (Declaration)

Public Const TwoPi As Single

Visual C++

public:
literal float TwoPi
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Calculator type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Acos</td>
<td>Returns the angle whose cosine is the specified value.</td>
</tr>
<tr>
<td>Asin</td>
<td>Returns the angle whose sine is the specified value.</td>
</tr>
<tr>
<td>Atan</td>
<td>Returns the angle whose tangent is the specified number.</td>
</tr>
<tr>
<td>Atan2</td>
<td>Returns the angle whose tangent is the quotient of the two specified numbers.</td>
</tr>
<tr>
<td>Clamp</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Cos</td>
<td>Returns the cosine of the specified angle.</td>
</tr>
<tr>
<td>Cosh</td>
<td>Returns the hyperbolic cosine of the specified angle.</td>
</tr>
<tr>
<td>CubicInterpolate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LinearInterpolate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Log</td>
<td>Returns the natural (base e) logarithm of the specified value.</td>
</tr>
<tr>
<td>Max</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Min</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Pow</td>
<td>Returns the specified value raised to the specified power.</td>
</tr>
<tr>
<td>Sin</td>
<td>Returns the sine of the specified angle.</td>
</tr>
<tr>
<td>Sinh</td>
<td>Returns the hyperbolic sine of the specified angle.</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Returns the square root of the specified value.</td>
</tr>
<tr>
<td>Tan</td>
<td>Returns the tangent of the specified angle.</td>
</tr>
<tr>
<td>Tanh</td>
<td>Returns the hyperbolic tangent of the specified angle.</td>
</tr>
<tr>
<td>Wrap</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Abs Method

[This is preliminary documentation and is subject to change.]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Abs(Single)</code></td>
<td>Returns the absolute value of a single precision floating point number.</td>
</tr>
<tr>
<td><code>Abs(Single%)</code></td>
<td>Assigns the absolute value of a single precision floating point number.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the absolute value of a single precision floating point number.

**Namespace:** [ProjectMercury](#)
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public static float Abs(
    float value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Abs ( _
    value As Single _
) As Single
```

**Visual C++**

```cpp
public:
static float Abs(
    float value
)
```

### Parameters

**value**

Type: `System::Single`
Source value.

### Return Value

The absolute value of the source value.
See Also

Calculator Class
Abs Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Assigns the absolute value of a single precision floating point number.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
# Syntax

## C#

```csharp
public static void Abs(
ref float value
)
```

## Visual Basic (Declaration)

```vbnet
Public Shared Sub Abs ( _
    ByRef value As Single _
)
```

## Visual C++

```cpp
gpublic:
static void Abs(
    float% value
)
```

## Parameters

value

- Type: `System::Single`%
- Source value.
See Also

Calculator Class
Abs Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the angle whose cosine is the specified value.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public static float Acos(float value)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Acos(_
    value As Single _
) As Single
```

**Visual C++**

```cpp
public:
    static float Acos(
        float value
    )
```

### Parameters

**value**

Type: `System::Single`

A number representing a cosine.

### Return Value

The angle whose cosine is the specified value.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the angle whose sine is the specified value.

**Namespace:** ProjectMercury
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Asin(float value)

Visual Basic (Declaration)

Public Shared Function Asin ( _
    value As Single _
) As Single

Visual C++

public:
static float Asin(
    float value
)

Parameters

value
    Type: System::Single
    A number representing a sine.

Return Value

The angle whose sine is the specified value.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Atan Method

Returns the angle whose tangent is the specified number.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Atan(  
    float value
)

Visual Basic (Declaration)

Public Shared Function Atan ( _  
    value As Single _
) As Single

Visual C++

public:  
static float Atan(  
    float value
)

Parameters

value
    Type: System::Single  
    A number representing a tangent.

Return Value

The angle whose tangent is the specified number.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the angle whose tangent is the quotient of the two specified numbers.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Atan2(
    float y,
    float x
)

Visual Basic (Declaration)

Public Shared Function Atan2 ( _
    y As Single, _
    x As Single _
) As Single

Visual C++

public:
static float Atan2(
    float y,
    float x
)

Parameters

y
Type: System::Single
The y coordinate of a point.

x
Type: System::Single
The x coordinate of a point.

Return Value

The angle whose tangent is the quotient of the two specified numbers.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Clamp Method

Calculator Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clamp(Single, Single, Single)</td>
<td>Restricts a value to be within a specified range.</td>
</tr>
<tr>
<td>Clamp(Single%, Single, Single)</td>
<td>Restricts a value to be within a specified range.</td>
</tr>
<tr>
<td>Clamp&lt;(Of &lt;(T)&gt;)(T, T, T)</td>
<td>Restricts a value to be within a specified range.</td>
</tr>
</tbody>
</table>

```
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Restricts a value to be within a specified range.

**Namespace:**  [ProjectMercury](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Clamp(
    float value,
    float min,
    float max
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Clamp ( _
    value As Single, _
    min As Single, _
    max As Single _
) As Single
```

Visual C++

```cpp
public:
static float Clamp(
    float value,
    float min,
    float max
)
```

Parameters

value
Type: `System::Single`
The value to clamp.

min
Type: `System::Single`
The minimum value. If value is less than min, min will be returned.

max
Type: `System::Single`
The maximum value. If value is greater than max, max will be returned.

**Return Value**

The clamped value.
See Also

Calculator Class
Clamp Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Restricts a value to be within a specified range.

**Namespace:** [ProjectMercury](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void Clamp(
    ref float value,
    float min,
    float max
)

Visual Basic (Declaration)

Public Shared Sub Clamp ( _
    ByRef value As Single, _
    min As Single, _
    max As Single _
)

Visual C++

public:
static void Clamp(
    float% value,
    float min,
    float max
)

Parameters

value
    Type: System:::Single %
The value to clamp.

min
    Type: System:::Single
    The minimum value. If value is less than min, value will be assigned min.

max
    Type: System:::Single
The maximum value. If value is greater than max, value will be assigned max.
See Also

Calculator Class
Clamp Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Restricts a value to be within a specified range.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static T Clamp<T>(
    T value,
    T min,
    T max
)
where T : IComparable<T>

Visual Basic (Declaration)

Public Shared Function Clamp(Of T As IComparable(Of T)) ( _
    value As T, _
    min As T, _
    max As T _
) As T

Visual C++

public:
    generic<typename T>
where T : IComparable<T>
static T Clamp(
    T value,
    T min,
    T max
)

Parameters

value
    Type: T
    The value to clamp.

min
    Type: T
    The minimum value. If value is less than min, min will be returned.
max

Type: T
The maximum value. If value is greater than max, max will be returned.
Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:ProjectMercury.Calculator.Clamp`1(``0,``0,``0)""]

Return Value

The clamped value.
See Also

Calculator Class
Clamp Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the cosine of the specified angle.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Cos(
    float value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Cos ( _
    value As Single _
) As Single
```

**Visual C++**

```cpp
public:
static float Cos(
    float value
)
```

**Parameters**

`value`

Type: `System::::Single`
An angle specified in radians.

**Return Value**

The cosine of the specified angle.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the hyperbolic cosine of the specified angle.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

class Cosh{
   public static float Cosh( float value )
}

Visual Basic (Declaration)

Public Shared Function Cosh ( _
   value As Single _
) As Single

Visual C++

public:
static float Cosh( float value )

Parameters

value
   Type: System::::Single
   An angle specified in radians.

Return Value

The hyperbolic cosine of the specified angle.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CubicInterpolate(Single, Single, Single)</code></td>
<td>Interpolates between two values using a cubic equation.</td>
</tr>
<tr>
<td><code>CubicInterpolate(Single%, Single, Single)</code></td>
<td>Interpolates between two values using a cubic equation.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Interpolates between two values using a cubic equation.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float CubicInterpolate(
    float value1,
    float value2,
    float amount
)

Visual Basic (Declaration)

Public Shared Function CubicInterpolate ( _
    value1 As Single, _
    value2 As Single, _
    amount As Single _
) As Single

Visual C++

public:
static float CubicInterpolate(
    float value1,
    float value2,
    float amount
)

Parameters

value1
    Type: System::Single
    Source value.

value2
    Type: System::Single
    Source value.

amount
    Type: System::Single
Weighting value.

**Return Value**

Interpolated value.
See Also

Calculator Class
CubicInterpolate Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Interpolates between two values using a cubic equation.

**Namespace:** ProjectMercury
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static void CubicInterpolate(
    ref float value,
    float value1,
    float value2,
    float amount
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub CubicInterpolate (_
    ByRef value As Single, _
    value1 As Single, _
    value2 As Single, _
    amount As Single _
)
```

Visual C++

```cpp
public:
static void CubicInterpolate(
    float% value,
    float value1,
    float value2,
    float amount
)
```

Parameters

value
Type: `System::Single` %
The output value.

value1
Type: `System::Single`
Source value.
value2
  Type: System::Single
  Source value.

amount
  Type: System::Single
  Weighting value.
See Also

Calculator Class
CubicInterpolate Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members
Project Mercury API Reference
Calculator....:::LinearInterpolate Method
Calculator Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LinearInterpolate(Single, Single, Single)</code></td>
<td>Linearly interpolates between two values.</td>
</tr>
<tr>
<td><code>LinearInterpolate(Single%, Single, Single, Single)</code></td>
<td>Linearly interpolates between two values.</td>
</tr>
<tr>
<td><code>LinearInterpolate(Single, Single, Single, Single)</code></td>
<td>Linearly interpolates between three values, where the position of the middle value is variable.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Linearly interpolates between two values.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float LinearInterpolate(
    float value1,
    float value2,
    float amount
)"
```

Visual Basic (Declaration)

```vbnet
Public Shared Function LinearInterpolate ( _
    value1 As Single, _
    value2 As Single, _
    amount As Single _
) As Single
```

Visual C++

```cpp
public:
static float LinearInterpolate(
    float value1,
    float value2,
    float amount
)"
```

Parameters

value1
  Type: System:::Single
  Source value.

value2
  Type: System:::Single
  Source value.

amount
  Type: System:::Single
Value between 0 and 1 indicating the weight of value2.

Return Value

Interpolated value.
See Also

Calculator Class
LinearInterpolate Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Linearly interpolates between two values.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void LinearInterpolate(
    ref float value,
    float value1,
    float value2,
    float amount
)

Visual Basic (Declaration)

Public Shared Sub LinearInterpolate (_
    ByRef value As Single, _
    value1 As Single, _
    value2 As Single, _
    amount As Single _
)

Visual C++

public:
static void LinearInterpolate(
    float% value,
    float value1,
    float value2,
    float amount
)

Parameters

value
    Type: System::Single %
    The output value.

value1
    Type: System::Single
    Source value.
value2
   Type: System.. Single
   Source value.

amount
   Type: System.. Single
   Value between 0 and 1 indicating the weight of value2.
See Also

Calculator Class
LinearInterpolate Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Linearly interpolates between three values, where the position of the middle value is variable.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float LinearInterpolate(
    float value1,
    float value2,
    float value2Position,
    float value3,
    float amount
)

Visual Basic (Declaration)

Public Shared Function LinearInterpolate ( _
    value1 As Single, _
    value2 As Single, _
    value2Position As Single, _
    value3 As Single, _
    amount As Single _
) As Single

Visual C++

public:
static float LinearInterpolate(
    float value1,
    float value2,
    float value2Position,
    float value3,
    float amount
)

Parameters

value1
    Type: System::::Single
    Source value.

value2
Type: **System:: Single**
Source value.

value2Position
Type: **System:: Single**
The position of the second source value between 0 and 1.

value3
Type: **System:: Single**
Source value.

amount
Type: **System:: Single**
Value between 0 and 1 indicating the position in the curve to evaluate.

**Return Value**

Interpolated value.
See Also

Calculator Class
LinearInterpolate Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the natural (base e) logarithm of the specified value.

**Namespace:**   ProjectMercury
**Assembly:**   ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Log(
    float value
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Log ( _
    value As Single _
) As Single
```

Visual C++

```cpp
public:
static float Log(
    float value
)
```

Parameters

value

Type: `System::Single`
A number whose logarithm is to be found.

Return Value

The natural (base e) logarithm of the specified value.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
This is preliminary documentation and is subject to change.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Max(Single, Single)</code></td>
<td>Returns the greater of two values.</td>
</tr>
<tr>
<td><code>Max&lt;(Of &lt;(T)&gt;)(T, T)</code></td>
<td>Returns the greater of two values.</td>
</tr>
<tr>
<td><code>Max(Single, Single, Single)</code></td>
<td>Returns the greater of three values.</td>
</tr>
<tr>
<td><code>Max(Single%, Single, Single)</code></td>
<td>Sets value to be the greater of two values.</td>
</tr>
<tr>
<td><code>Max&lt;(Of &lt;(T)&gt;)(T, T)</code></td>
<td>Returns the greater of three values.</td>
</tr>
<tr>
<td><code>Max(Single%, Single, Single, Single)</code></td>
<td>Sets value to the the greater of three values.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the greater of two values.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Max(
    float value1,
    float value2
)

Visual Basic (Declaration)

Public Shared Function Max ( _
    value1 As Single, _
    value2 As Single _
) As Single

Visual C++

public:
static float Max(
    float value1,
    float value2
)

Parameters

value1
    Type: System::Single
    Source value.

value2
    Type: System::Single
    Source value.

Return Value

The greater value.
See Also

Calculator Class
Max Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the greater of two values.

**Namespace:** [ProjectMercury](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
public static T Max<T>(
    T value1,
    T value2
)
where T : IComparable<T>
```

#### Visual Basic (Declaration)

```vbnet
Public Shared Function Max(Of T As IComparable(Of T)) ( _
    value1 As T, _
    value2 As T _
) As T
```

#### Visual C++

```cpp
public:
    generic<typename T>
    where T : IComparable<T>
    static T Max( 
        T value1, 
        T value2 
    )
```

### Parameters

- **value1**
  - Type: T
  - Source value.

- **value2**
  - Type: T
  - Source value.
Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:ProjectMercury.Calculator.Max`1(0,0)""]

Return Value

The greater value, or value1 if the values are equal.
See Also

Calculator Class
Max Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the greater of three values.

**Namespace:** [ProjectMercury](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Max(
    float value1,
    float value2,
    float value3
)

Visual Basic (Declaration)

Public Shared Function Max ( _
    value1 As Single, _
    value2 As Single, _
    value3 As Single _
) As Single

Visual C++

public:
static float Max(
    float value1,
    float value2,
    float value3
)

Parameters

value1
Type: System::Single
Source value.

value2
Type: System::Single
Source value.

value3
Type: System::Single
Source value.

**Return Value**

The greater value.
See Also

Calculator Class
Max Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Sets value to be the greater of two values.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void Max(
    ref float value,
    float value1,
    float value2
)

Visual Basic (Declaration)

Public Shared Sub Max ( _
    ByRef value As Single, _
    value1 As Single, _
    value2 As Single _
)

Visual C++

public:
static void Max(
    float% value,
    float value1,
    float value2
)

Parameters

value
    Type: System::::Single %
    The output value.

value1
    Type: System::::Single
    Source value.

value2
    Type: System::::Single
Source value.
See Also

Calculator Class
Max Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the greater of three values.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static T Max<T>(
    T value1,
    T value2,
    T value3
)
where T : IComparable<T>

Visual Basic (Declaration)

Public Shared Function Max(Of T As IComparable(Of T)) ( _
    value1 As T, _
    value2 As T, _
    value3 As T _
) As T

Visual C++

public:
    generic<typename T>
    where T : IComparable<T>
    static T Max(
        T value1,
        T value2,
        T value3
    )

Parameters

value1
    Type: T
    Source value.

value2
    Type: T
    Source value.
value3
    Type: T
    Source value.
Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:ProjectMercury.Calculator.Max`1(0,0,0)"

Return Value

The greater value, or value1 if the values are equal.
See Also

Calculator Class
Max Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Sets value to the the greater of three values.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public static void Max(
    ref float value,
    float value1,
    float value2,
    float value3
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Sub Max (
    ByRef value As Single, _
    value1 As Single, _
    value2 As Single, _
    value3 As Single _
)
```

### Visual C++

```cpp
public:
    static void Max(
        float% value,
        float value1,
        float value2,
        float value3
    )
```

## Parameters

**value**
- **Type:** `System::Single %`
- The output value.

**value1**
- **Type:** `System::Single`
- Source value.
value2
Type: System::Single
Source value.

value3
Type: System::Single
Source value.
See Also

Calculator Class
Max Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Calculator...::Min Method

Calculation Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Min(Single, Single)</code></td>
<td>Returns the lesser of two values.</td>
</tr>
<tr>
<td><code>Min&lt;(Of &lt;(T)&gt;))(T, T)</code></td>
<td>Returns the lesser of two values.</td>
</tr>
<tr>
<td><code>Min(Single, Single, Single)</code></td>
<td>Returns the lesser of three values.</td>
</tr>
<tr>
<td><code>Min(Single%, Single, Single)</code></td>
<td>Sets value to be the lesser of two values.</td>
</tr>
<tr>
<td><code>Min&lt;(Of &lt;(T)&gt;))(T, T)</code></td>
<td>Returns the lesser of three values.</td>
</tr>
<tr>
<td><code>Min(Single%, Single, Single, Single)</code></td>
<td>Sets value to be the lesser of three values.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the lesser of two values.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Min(
    float value1,
    float value2
)

Visual Basic (Declaration)

Public Shared Function Min ( _
    value1 As Single, _
    value2 As Single _
) As Single

Visual C++

public:
static float Min(
    float value1,
    float value2
)

Parameters

value1
    Type: System::Single
    Source value.

value2
    Type: System::Single
    Source value.

Return Value

The lesser value.
See Also

Calculator Class
Min Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the lesser of two values.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static T Min<T>(
    T value1,
    T value2
)
where T : IComparable<T>
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Min(Of T As IComparable(Of T)) ( _
    value1 As T, _
    value2 As T _
) As T
```

Visual C++

```cpp
public:
    generic<typename T>
where T : IComparable<T>
static T Min(
    T value1,
    T value2
)
```

Parameters

value1
   Type: T
   Source value.

value2
   Type: T
   Source value.
Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:ProjectMercury.Calculator.Min`1(``0,``0)""]

Return Value

The lesser value, or value1 if the values are equal.
See Also

Calculator Class
Min Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the lesser of three values.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Min(
    float value1,
    float value2,
    float value3
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Min ( _
    value1 As Single, _
    value2 As Single, _
    value3 As Single _
) As Single
```

Visual C++

```c++
public:
    static float Min(
        float value1,
        float value2,
        float value3
    )
```

Parameters

value1
Type: `System::::Single`
Source value.

value2
Type: `System::::Single`
Source value.

value3
Type: `System::::Single`
Source value.

Return Value

The lesser value.
See Also

Calculator Class
Min Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Sets value to be the lesser of two values.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static void Min(
    ref float value,
    float value1,
    float value2
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub Min ( _
    ByRef value As Single, _
    value1 As Single, _
    value2 As Single _
)
```

Visual C++

```cpp
public:
static void Min( 
    float% value, 
    float value1, 
    float value2
)
```

Parameters

value
Type: System::Single %
The output value.

value1
Type: System::Single
Source value.

value2
Type: System::Single
Source value.
See Also

Calculator Class
Min Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the lesser of three values.

**Namespace:**  [ProjectMercury](https://example.com/ProjectMercury)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static T Min<T>(
    T value1,
    T value2,
    T value3
)
where T : IComparable<T>
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Min(Of T As IComparable(Of T)) ( _
    value1 As T, _
    value2 As T, _
    value3 As T _
) As T
```

Visual C++

```cpp
public:
    template<typename T>
    where T : IComparable<T>
    static T Min( 
        T value1, 
        T value2, 
        T value3
    )
```

Parameters

value1
Type: T
Source value.

value2
Type: T
Source value.
value3
  Type: T
  Source value.
Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:ProjectMercury.Calculator.Min`1(0,0,0)"

Return Value

The lesser value, or value1 if the values are equal.
See Also

Calculator Class
Min Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Sets value to be the lesser of three values.

**Namespace:**  [ProjectMercury](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static void Min(
    ref float value,
    float value1,
    float value2,
    float value3
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub Min ( _
    ByRef value As Single, _
    value1 As Single, _
    value2 As Single, _
    value3 As Single _
)
```

Visual C++

```cpp
public:
    static void Min( (
        float% value,
        float value1,
        float value2,
        float value3
    )
```

Parameters

value
    Type: `System::Single` %
    The output value.

value1
    Type: `System::Single`
    Source value.
value2
   Type: System::Single
   Source value.

value3
   Type: System::Single
   Source value.
See Also

Calculator Class
Min Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the specified value raised to the specified power.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Pow(
    float value,
    float power
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Pow ( _
    value As Single, _
    power As Single _
) As Single
```

Visual C++

```cpp
public:
static float Pow(
    float value,
    float power
)
```

Parameters

value
Type: System::Single
Source value.

power
Type: System::Single
A single precision floating point number that specifies a power.

Return Value

The specified value raised to the specified power.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the sine of the specified angle.

**Namespace:** [ProjectMercury](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Sin(
    float value
)

Visual Basic (Declaration)

Public Shared Function Sin ( _
    value As Single _
) As Single

Visual C++

public:
static float Sin(
    float value
)

Parameters

value
    Type: System::Single
    An angle specified in radians.

Return Value

The sine of the specified angle.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the hyperbolic sine of the specified angle.

**Namespace:**  [ProjectMercury](https://example.com)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Sinh(
    float value
)

Visual Basic (Declaration)

Public Shared Function Sinh ( _
    value As Single _
) As Single

Visual C++

public:
    static float Sinh(
        float value
    )

Parameters

value

    Type: System:: Single
    An angle specified in radians.

Return Value

The hyperbolic sine of the specified angle.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Calculates the square root of the specified value.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public static float Sqrt(
    float value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Sqrt ( _
    value As Single _
) As Single
```

**Visual C++**

```cpp
public:
static float Sqrt(
    float value
)
```

**Parameters**

value

Type: System::Single

Source value.

**Return Value**

The square root of the specified value.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the tangent of the specified angle.

**Namespace:**  [ProjectMercury](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

**C#**

```csharp
public static float Tan(
    float value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Tan ( _
    value As Single _
) As Single
```

**Visual C++**

```cpp
public:
static float Tan(
    float value
)
```

**Parameters**

value
Type: `System::Single`
An angle specified in radians.

**Return Value**

The tangent of the specified angle.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the hyperbolic tangent of the specified angle.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float Tanh(  
    float value
)

Visual Basic (Declaration)

Public Shared Function Tanh ( _  
    value As Single _
) As Single

Visual C++

public:
static float Tanh(  
    float value
)

Parameters

value
    Type: System:::Single
    An angle specified in radians.

Return Value

The hyperbolic tangent of the specified angle.
See Also

Calculator Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C# □ Visual Basic □ Visual C++ □ Include Protected Members □ Include Inherited Members
Project Mercury API Reference
Calculator:::Wrap Method

Calculator Class See Also Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrap(Single, Single, Single)</td>
<td>Wraps the specified value to be within the specified range.</td>
</tr>
<tr>
<td>Wrap(Single%, Single, Single)</td>
<td>Wraps the specified value to be within the specified range.</td>
</tr>
</tbody>
</table>
See Also

Calculator Class
Calculator Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Widget Description

Wraps the specified value to be within the specified range.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Wrap(
    float value,
    float min,
    float max
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Wrap ( _
    value As Single, _
    min As Single, _
    max As Single _
) As Single
```

Visual C++

```cpp
public:
static float Wrap(
    float value,
    float min,
    float max
)
```

Parameters

value
Type: `System::::Single`
The value to wrap.

min
Type: `System::::Single`
The minimum value.

max
Type: `System::::Single`
The maximum value.

**Return Value**

The wrapped value.
See Also

Calculator Class
Wrap Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Wraps the specified value to be within the specified range.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public static void Wrap(
    ref float value,
    float min,
    float max
)

Visual Basic (Declaration)

Public Shared Sub Wrap (
    ByRef value As Single, 
    min As Single, 
    max As Single 
)

Visual C++

public:
    static void Wrap(
        float% value,
        float min,
        float max
    )

Parameters

value
    Type: System::::Single %
    The value to wrap.

min
    Type: System::::Single
    The minimum value.

max
    Type: System::::Single
The maximum value.
See Also

Calculator Class
Wrap Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines helper methods for validating arguments passed into methods.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public static class Guard

Visual Basic (Declaration)
Public NotInheritable Class Guard

Visual C++
public ref class Guard abstract sealed
Inheritance Hierarchy

System...:::Object
   ProjectMercury...:::Guard
See Also

Guard Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **Guard** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentGreaterThan&lt;((\text{T}))&gt;</td>
<td>Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is greater than the specified threshold.</td>
</tr>
<tr>
<td>ArgumentLessThan&lt;((\text{T}))&gt;</td>
<td>Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is less than the specified threshold.</td>
</tr>
<tr>
<td>ArgumentNotFinite</td>
<td>Performs a check against an argument, and throws a NotFiniteNumberException if it is not a finite number eg NaN, PositiveInfinity or NegativeInfinity.</td>
</tr>
<tr>
<td>ArgumentNull</td>
<td>Performs a check against a string argument, and throws a ArgumentNullException if it is null.</td>
</tr>
<tr>
<td>ArgumentNullOrEmpty</td>
<td>Performs a check against a method argument, and throws an ArgumentNullException if it is null or empty.</td>
</tr>
<tr>
<td>ArgumentOutOfRange&lt;((\text{T}))&gt;</td>
<td>Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is greater than the specified maximum value, or less than the specified minimum value.</td>
</tr>
<tr>
<td>IsFalse</td>
<td>Throws an InvalidOperationException if the specified expression evaluates to false.</td>
</tr>
<tr>
<td>IsTrue</td>
<td>Throws an InvalidOperationException if the specified expression evaluates to true.</td>
</tr>
</tbody>
</table>
See Also

- Guard Class
- ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Guard type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentGreaterThan(&lt;(\text{T}))&gt;</td>
<td>Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is greater than the specified threshold.</td>
</tr>
<tr>
<td>ArgumentLessThan(&lt;(\text{T}))&gt;</td>
<td>Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is less than the specified threshold.</td>
</tr>
<tr>
<td>ArgumentNotFinite</td>
<td>Performs a check against an argument, and throws a NotFiniteNumberException if it is not a finite number eg NaN, PositiveInfinity or NegativeInfinity.</td>
</tr>
<tr>
<td>ArgumentNull</td>
<td>Performs a check against a string argument, and throws a ArgumentNullException if it is null.</td>
</tr>
<tr>
<td>ArgumentNullOrEmpty</td>
<td>Performs a check against a method argument, and throws an ArgumentNullException if it is null or empty.</td>
</tr>
<tr>
<td>ArgumentOutOfRange(&lt;(\text{T}))&gt;</td>
<td>Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is greater than the specified maximum value, or less than the specified minimum value.</td>
</tr>
<tr>
<td>IsFalse</td>
<td>Throws an InvalidOperationException if the specified expression evaluates to false.</td>
</tr>
<tr>
<td>IsTrue</td>
<td>Throws an InvalidOperationException if the specified expression evaluates to true.</td>
</tr>
</tbody>
</table>
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Guard..::..ArgumentGreaterThan<Of <(T)>>> Method

Perform a check against a method argument, and throws an ArgumentOutOfRangeException if it is greater than the specified threshold.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public static void ArgumentGreaterThan<T>(
    string parameter,
    T argument,
    T threshold
)
where T : IComparable<T>
```

### Visual Basic (Declaration)

```vbnet
Public Shared Sub ArgumentGreaterThan(Of T As IComparable(Of T)) (
    _
    parameter As String, _
    argument As T, _
    threshold As T_
)
```

### Visual C++

```cpp
public:
    generic<typename T>
    where T : IComparable<T>
    static void ArgumentGreaterThan(
        String^ parameter,
        T argument,
        T threshold
    )
```

## Parameters

**parameter**

Type: `System::String`

The name of the method parameter.

**argument**

Type: `T`

The value being passed as an argument.
threshold
    Type: T
    The threshold value that the argument must be equal to or less than to pass the test.
Type Parameters

T
The type of argument being checked.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is less than the specified threshold.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void ArgumentLessThan<T>(
    string parameter,
    T argument,
    T threshold
)
where T : Object, IComparable<T>

Visual Basic (Declaration)

Public Shared Sub ArgumentLessThan(Of T As {Object, IComparable(Of 1
    parameter As String, _
    argument As T, _
    threshold As T _
)

Visual C++

public:
    generic<typename T>
    where T : Object, IComparable<T>
    static void ArgumentLessThan(
        String^ parameter,
        T argument,
        T threshold
    )

Parameters

parameter
    Type: System::String
    The name of the method parameter.

argument
    Type: T
    The value being passed as an argument.
threshold

Type: T
The threshold value that the argument must be equal to or greater than to pass the test.
Type Parameters

T

The type of argument being checked.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Perform a check against a method argument, and throws a NotFiniteNumberException if it is not a finite number eg NaN, PositiveInfinity or NegativeInfinity.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void ArgumentNotFinite(
    string parameter,
    float argument
)

Visual Basic (Declaration)

Public Shared Sub ArgumentNotFinite ( _
    parameter As String, _
    argument As Single _
)

Visual C++

public:
static void ArgumentNotFinite(
    String^ parameter,
    float argument
)

Parameters

parameter
  Type: System::::String
  The name of the method parameter.

argument
  Type: System::::Single
  The value being passed as an argument.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Performs a check against an argument, and throws an ArgumentNullException if it is null.

Namespace:  ProjectMercury
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void ArgumentNull(
    string parameter,
    Object argument
)

Visual Basic (Declaration)

Public Shared Sub ArgumentNull ( _
    parameter As String, _
    argument As Object _
)

Visual C++

public:
static void ArgumentNull(
    String^ parameter,
    Object^ argument
)

Parameters

parameter
    Type: System::String
    The name of the method parameter.

argument
    Type: System::Object
    The value being passed as an argument.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Performs a check against a string argument, and throws an ArgumentNullException if it is null or empty.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public static void ArgumentNullOrEmpty(
    string parameter,
    string argument
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Sub ArgumentNullOrEmpty ( _
    parameter As String, _
    argument As String _
)
```

### Visual C++

```cpp
public:
static void ArgumentNullOrEmpty(
    String^ parameter,
    String^ argument
)
```

## Parameters

**parameter**
- **Type:** `System::String`
- The name of the method parameter.

**argument**
- **Type:** `System::String`
- The value being passed as an argument.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Performs a check against a method argument, and throws an ArgumentOutOfRangeException if it is greater than the specified maximum value, or less than the specified minimum value.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void ArgumentOutOfRange<T>(
    string parameter,
    T argument,
    T min,
    T max
)
where T : IComparable<T>

Visual Basic (Declaration)

Public Shared Sub ArgumentOutOfRange(Of T As IComparable(Of T)) ( _
    parameter As String, _
    argument As T, _
    min As T, _
    max As T _
)

Visual C++

public:
    generic<typename T>
where T : IComparable<T>
    static void ArgumentOutOfRange(
        String^ parameter,
        T argument,
        T min,
        T max
    )

Parameters

parameter
  Type: System::String
  The name of the method parameter.

argument
Type: T
The value being passed as an argument.

min
Type: T
The minimum allowed value (inclusive).

max
Type: T
The maximum allowed value (inclusive).
Type Parameters

T

The type of argument being checked.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Throws an InvalidOperationException if the specified expression evaluates to false.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static void IsFalse(
    bool expression,
    string message
)

Visual Basic (Declaration)

Public Shared Sub IsFalse ( _
    expression As Boolean, _
    message As String _
)

Visual C++

public:
    static void IsFalse(
        bool expression,
        String^ message
    )

Parameters

expression
    Type: System::Boolean
    The expression to evaluate.

message
    Type: System::String
    The error message.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Throws an InvalidOperationException if the specified expression evaluates to true.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public static void IsTrue(
    bool expression,
    string message
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Sub IsTrue ( _
    expression As Boolean, _
    message As String _
)
```

### Visual C++

```cpp
public:
static void IsTrue(
    bool expression,
    String^ message
)
```

## Parameters

**expression**
- Type: `System::Boolean`
- The expression to evaluate.

**message**
- Type: `System::String`
- The error message.
See Also

Guard Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the data structure for a particle.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public struct Particle

Visual Basic (Declaration)

Public Structure Particle

Visual C++

public value class Particle
See Also

Particle Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `Particle` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ApplyForce</strong></td>
<td>Applies a force to the particle.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Rotate</strong></td>
<td>Applies a rotation to the Particle.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns the fully qualified type name of this instance. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets or sets the age of the particle in the range 0-1.</td>
</tr>
<tr>
<td>Colour</td>
<td>Gets or sets the colour of the particle. The W component is opacity.</td>
</tr>
<tr>
<td>Inception</td>
<td>Gets or sets the time at which the particle was released.</td>
</tr>
<tr>
<td>Momentum</td>
<td>Gets or sets the current momentum of the particle.</td>
</tr>
<tr>
<td>Position</td>
<td>Gets or sets the position of the particle.</td>
</tr>
<tr>
<td>Rotation</td>
<td>Gets or sets the rotation of the particle in radians.</td>
</tr>
<tr>
<td>Scale</td>
<td>Gets or sets the scale of the particle.</td>
</tr>
<tr>
<td>Velocity</td>
<td>Gets or sets the sum of the forces which are currently acting on the particle.</td>
</tr>
<tr>
<td>VertexElements</td>
<td>Contains the vertex element data for a Particle.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![s] SizeInBytes</td>
<td>Gets the size of a Particle structure in bytes.</td>
</tr>
</tbody>
</table>
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `Particle` type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Age</strong></td>
<td>Gets or sets the age of the particle in the range 0-1.</td>
</tr>
<tr>
<td><strong>Colour</strong></td>
<td>Gets or sets the colour of the particle. The W component is opacity.</td>
</tr>
<tr>
<td><strong>Inception</strong></td>
<td>Gets or sets the time at which the particle was released.</td>
</tr>
<tr>
<td><strong>Momentum</strong></td>
<td>Gets or sets the current momentum of the particle.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Gets or sets the position of the particle.</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td>Gets or sets the rotation of the particle in radians.</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Gets or sets the scale of the particle.</td>
</tr>
<tr>
<td><strong>Velocity</strong></td>
<td>Gets or sets the sum of the forces which are currently acting on the particle.</td>
</tr>
<tr>
<td><strong>VertexElements</strong></td>
<td>Contains the vertex element data for a Particle.</td>
</tr>
</tbody>
</table>
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the age of the particle in the range 0-1.

**Namespace:** ProjectMercury
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Age

Visual Basic (Declaration)

Public Age As Single

Visual C++

public:
float Age
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the colour of the particle. The W component is opacity.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector4 Colour

Visual Basic (Declaration)

Public Colour As Vector4

Visual C++

public:
Vector4 Colour
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the time at which the particle was released.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Inception

Visual Basic (Declaration)

Public Inception As Single

Visual C++

public:
float Inception
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the current momentum of the particle.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 Momentum

Visual Basic (Declaration)

Public Momentum As Vector2

Visual C++

public:
    Vector2 Momentum
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the position of the particle.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 Position

Visual Basic (Declaration)

Public Position As Vector2

Visual C++

public:
Vector2 Position
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
[This is preliminary documentation and is subject to change.]

Gets or sets the rotation of the particle in radians.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Rotation

Visual Basic (Declaration)

Public Rotation As Single

Visual C++

public:
float Rotation
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the scale of the particle.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

public float Scale

### Visual Basic (Declaration)

Public Scale As Single

### Visual C++

public:
float Scale
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the sum of the forces which are currently acting on the particle.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 Velocity

Visual Basic (Declaration)

Public Velocity As Vector2

Visual C++

public:
Vector2 Velocity
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Contains the vertex element data for a Particle.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static readonly VertexElement[] VertexElements

Visual Basic (Declaration)

Public Shared ReadOnly VertexElements As VertexElement()

Visual C++

public:
static initonly array<VertexElement>^ VertexElements
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Particle type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ApplyForce</strong></td>
<td>Applies a force to the particle.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td>(Inherited from <strong>ValueType</strong>)</td>
<td></td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>)</td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td>(Inherited from <strong>ValueType</strong>)</td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>)</td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>)</td>
<td></td>
</tr>
<tr>
<td><strong>Rotate</strong></td>
<td>Applies a rotation to the Particle.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns the fully qualified type name of this instance.</td>
</tr>
<tr>
<td>(Inherited from <strong>ValueType</strong>)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Applies a force to the particle.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public void ApplyForce(
    ref Vector2 force
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub ApplyForce ( 
    ByRef force As Vector2 
)
```

**Visual C++**

```cpp
public:
    void ApplyForce(
        Vector2% force
    )
```

### Parameters

- **force**
  - Type: `Vector2`%
  - A vector describing the force and direction.
See Also

- Particle Structure
- ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Applies a rotation to the Particle.

**Namespace:**  [ProjectMercury](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public void Rotate(
    float radians
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub Rotate ()
    radians As Single
)```

### Visual C++

```cpp
public:
void Rotate(
    float radians
)
```

## Parameters

- **radians**
  - Type: `System::Single`
  - The angle to rotate in radians.
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Particle type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="SizeInBytes" /></td>
<td>Gets the size of a Particle structure in bytes.</td>
</tr>
</tbody>
</table>
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the size of a Particle structure in bytes.

**Namespace:** ProjectMercury
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static int SizeInBytes { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property SizeInBytes As Integer

Visual C++

public:
static property int SizeInBytes {
    int get ();
}

}
See Also

Particle Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the root of a particle effect hierarchy.

**Namespace:**  [ProjectMercury](#)
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class ParticleEffect : EmitterCollection

Visual Basic (Declaration)

Public Class ParticleEffect
    Inherits EmitterCollection

Visual C++

public ref class ParticleEffect : public EmitterCollection
Inheritance Hierarchy

System..:::Object
System.Collections.Generic..:::List<Of <Emitter>>
ProjectMercury.Emitters..:::EmitterCollection
ProjectMercury..:::ParticleEffect
See Also

ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ParticleEffect` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticleEffect</td>
<td>Initializes a new instance of the ParticleEffect class.</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an object to the end of the <code>List(Of &lt;T&gt;</code>) &lt;br&gt; (Inherited from <code>List(Of &lt;Emitter&gt;</code>) &lt;br&gt; Adds the elements of the specified collection to the end of the <code>List(Of &lt;T&gt;)</code> &lt;br&gt; (Inherited from <code>List(Of &lt;Emitter&gt;</code>) &lt;br&gt; Returns a read-only <code>IList(Of &lt;T&gt;)</code> wrapper for the current collection. &lt;br&gt; (Inherited from <code>List(Of &lt;Emitter&gt;</code>) &lt;br&gt; Converts the elements in the current <code>List(Of &lt;T&gt;)</code> to another type, and returns a list containing the converted elements. &lt;br&gt; (Inherited from <code>List(Of &lt;Emitter&gt;</code>) &lt;br&gt; Returns a deep copy of the ParticleEffect. &lt;br&gt; Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. &lt;br&gt; Determines whether the <code>List(Of &lt;T&gt;)</code> contains elements that match the conditions defined by the specified predicate. &lt;br&gt; Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. &lt;br&gt; (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>AddRange</td>
<td>(Inherited from <code>List(Of &lt;Emitter&gt;</code>)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>(Inherited from <code>List(Of &lt;Emitter&gt;</code>)</td>
</tr>
<tr>
<td>BinarySearch</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Clear</td>
<td>(Inherited from <code>List(Of &lt;Emitter&gt;</code>) &lt;br&gt; Removes all elements from the <code>List(Of &lt;T&gt;)</code> &lt;br&gt; (Inherited from <code>List(Of &lt;Emitter&gt;</code>) &lt;br&gt; Determines whether an element is in the <code>List(Of &lt;T&gt;)</code>. &lt;br&gt; (Inherited from <code>List(Of &lt;Emitter&gt;</code>)</td>
</tr>
<tr>
<td>Contains</td>
<td>(Inherited from <code>List(Of &lt;Emitter&gt;</code>)</td>
</tr>
<tr>
<td>ConvertAll(Of &lt;TOutput&gt;)</td>
<td>to another type, and returns a list</td>
</tr>
<tr>
<td>CopyTo</td>
<td>(Inherited from <code>List(Of &lt;Emitter&gt;</code>)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td></td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Exists</td>
<td>(Inherited from <code>List(Of &lt;Emitter&gt;</code>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Find**               | Searches for an element that matches the conditions defined by the specified predicate, and returns the first occurrence within the entire `List<Of <(T)>>`.
| (Inherited from `List<Of <(Emitter)>`).) |
| **FindAll**            | Retrieves all the elements that match the conditions defined by the specified predicate.
| (Inherited from `List<Of <(Emitter)>`).) |
| **FindIndex**          | Overloaded.                                                                 |
| **FindLast**           | Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire `List<Of <(T)>>`.
| (Inherited from `List<Of <(Emitter)>`).) |
| **FindLastIndex**      | Overloaded.                                                                 |
| **ForEach**            | Performs the specified action on each element of the `List<Of <(T)>>`.
| (Inherited from `List<Of <(Emitter)>`).) |
| **GetEnumerator**      | Returns an enumerator that iterates through the `List<Of <(T)>>`.
| (Inherited from `List<Of <(Emitter)>`).) |
| **GetHashCode**        | Serves as a hash function for a particular type.
| (Inherited from `Object`.) |
| **GetRange**           | Creates a shallow copy of a range of elements in the source `List<Of <(T)>>`.
| (Inherited from `List<Of <(Emitter)>`).) |
| **GetType**            | Gets the `Type` of the current instance.
| (Inherited from `Object`.) |
| **IndexOf**            | Overloaded.                                                                 |
| **Initialise**         | Initialises all Emitters within the ParticleEffect.                        |
| **Insert**             | Inserts an element into the `List<Of <(T)>>` at the specified index.
| (Inherited from `List<Of <(Emitter)>`).) |
| **InsertRange**        | Inserts the elements of a collection into the `List<Of <(T)>>` at the specified index.
| (Inherited from `List<Of <(Emitter)>`).) |
| **LastIndexOf**        | Overloaded.                                                                 |
| **LoadContent**        | Loads content required by Emitters within the                             |
MemberwiseClone

Creates a shallow copy of the current Object.
(Inherited from Object.)

OnNameChanged

 Raises the [E:NameChanged] event.

Remove

Removes the first occurrence of a specific object from the List(Of T). (Inherited from List(Of Emitter).)

RemoveAll

Removes all the elements that match the conditions defined by the specified predicate.
(Inherited from List(Of Emitter).)

RemoveAt

Removes the element at the specified index of the List(Of T). (Inherited from List(Of Emitter).)

RemoveRange

Removes a range of elements from the List(Of T). (Inherited from List(Of Emitter).)

Reverse

Overloaded.

Sort

Overloaded.

Terminate

Terminates all Emitters within the ParticleEffect with immediate effect.

ToArray

Copies the elements of the List(Of T) to a new array.
(Inherited from List(Of Emitter).)

ToString

Returns a String that represents the current Object. (Inherited from Object.)

Trigger

Overloaded.

Sets the capacity to the actual number of elements in the List(Of T), if that number is less than a threshold value. (Inherited from List(Of Emitter).)

Determines whether every element in the List(Of T) matches the conditions defined by the specified predicate. (Inherited from List(Of Emitter).)

Update

Overloaded.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Gets or sets the author of the ParticleEffect.</td>
</tr>
<tr>
<td>Description</td>
<td>Gets or sets the description of the ParticleEffect.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the total number of active Particles in the ParticleEffect.</td>
</tr>
<tr>
<td>Capacity</td>
<td>Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from List&lt;(Of (Emitter)&gt;).)</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements actually contained in the List&lt;(Of (T)&gt;). (Inherited from List&lt;(Of (Emitter)&gt;).)</td>
</tr>
<tr>
<td>Item</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the ParticleEffect.</td>
</tr>
</tbody>
</table>
**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Occurs when name of the ParticleEffect has been changed.</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ParticleEffect` class.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public ParticleEffect()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ParticleEffect()
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ParticleEffect type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author</strong></td>
<td>Gets or sets the author of the ParticleEffect.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Gets or sets the description of the ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the author of the ParticleEffect.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public string Author

Visual Basic (Declaration)

Public Author As String

Visual C++

public:
String^ Author
See Also

ParticleEffect Class  
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the description of the ParticleEffect.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public string Description

Visual Basic (Declaration)

Public Description As String

Visual C++

public:
    String^ Description
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ParticleEffect` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an object to the end of the List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).) Adds the elements of the specified collection to the end of the List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td>AddRange</td>
<td>Returns a read-only IList(Of(Of(T)&gt;)) wrapper for the current collection. (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>(Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td>BinarySearch</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all elements from the List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).) Determines whether an element is in the List(Of(Of(T)&gt;)).</td>
</tr>
<tr>
<td>Contains</td>
<td>(Inherited from List(Of(Of(Emitter)&gt;)).) Converts the elements in the current List(Of(Of(T)&gt;)).</td>
</tr>
<tr>
<td>ConvertAll</td>
<td>Converts the elements in the current List(Of(Of(T)&gt;)) to another type, and returns a list containing the converted elements. (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the ParticleEffect.</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>Exists</td>
<td>Determines whether the List(Of(Of(T)&gt;)) contains elements that match the conditions defined by the specified predicate. (Inherited from List(Of(Of(Emitter)&gt;)).)</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>
</tbody>
</table>
- **Find**: Searches for an element that matches the conditions defined by the specified predicate, and returns the first occurrence within the entire `List<(Of <T>)>.
(Inherited from `List<(Of <Emitter>)>.)
- **FindAll**: Retrieves all the elements that match the conditions defined by the specified predicate.
(Inherited from `List<(Of <Emitter>)>.)
- **FindIndex**: Overloaded.
- **FindLast**: Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire `List<(Of <T>)>.
(Inherited from `List<(Of <Emitter>)>.)
- **FindLastIndex**: Overloaded.
- **ForEach**: Performs the specified action on each element of the `List<(Of <T>)>.
(Inherited from `List<(Of <Emitter>)>.)
- **GetEnumerator**: Returns an enumerator that iterates through the `List<(Of <T>)>.
(Inherited from `List<(Of <Emitter>)>.)
- **GetHashCode**: Serves as a hash function for a particular type.
(Inherited from `Object`.)
- **GetRange**: Creates a shallow copy of a range of elements in the source `List<(Of <T>)>.
(Inherited from `List<(Of <Emitter>)>.)
- **GetType**: Gets the `Type` of the current instance.
(Inherited from `Object`.)
- **IndexOf**: Overloaded.
- **Initialise**: Initialises all Emitters within the ParticleEffect.
- **Insert**: Inserts an element into the `List<(Of <T>)>` at the specified index.
(Inherited from `List<(Of <Emitter>)>))
- **InsertRange**: Inserts the elements of a collection into the `List<(Of <T>)>` at the specified index.
(Inherited from `List<(Of <Emitter>)>))
- **LastIndexOf**: Overloaded.
- **LoadContent**: Loads content required by Emitters within the
ParticleEffect.

**MemberwiseClone**
Creates a shallow copy of the current *Object*.  
(Inherited from *Object*.)

**OnNameChanged**
Raises the [E:NameChanged] event.

**Remove**
Removes the first occurrence of a specific object from the *List*<(Of <(T)>)>.
(Inherited from *List*<(Of <(Emitter)>)>.)

**RemoveAll**
Removes all the elements that match the conditions defined by the specified predicate.  
(Inherited from *List*<(Of <(Emitter)>)>.)

**RemoveAt**
Removes the element at the specified index of the *List*<(Of <(T)>)>.
(Inherited from *List*<(Of <(Emitter)>)>.)

**RemoveRange**
Removes a range of elements from the *List*<(Of <(T)>)>.
(Inherited from *List*<(Of <(Emitter)>)>.)

**Reverse**
Overloaded.

**Sort**
Overloaded.

**Terminate**
Terminates all Emitters within the ParticleEffect with immediate effect.

**ToArray**
Copies the elements of the *List*<(Of <(T)>)> to a new array.  
(Inherited from *List*<(Of <(Emitter)>)>.)

**ToString**
Returns a *String* that represents the current *Object*.  
(Inherited from *Object*.)

**Trigger**
Overloaded.

**TrimExcess**
Sets the capacity to the actual number of elements in the *List*<(Of <(T)>)> if that number is less than a threshold value.  
(Inherited from *List*<(Of <(Emitter)>)>.)

**TrueForAll**
Determines whether every element in the *List*<(Of <(T)>)> matches the conditions defined by the specified predicate.  
(Inherited from *List*<(Of <(Emitter)>)>.)

**Update**
Overloaded.
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect Class

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinarySearch(T)</td>
<td>Searches the entire sorted <code>List(Of </code>(T)<code>)</code>) for an element using the default comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;(Of </code>&lt;(Emitter<code>)&gt;)</code>).</td>
</tr>
<tr>
<td>BinarySearch(T, IComparer&lt;(Of <code>(T)</code>)&gt;)</td>
<td>Searches the entire sorted <code>List&lt;(Of </code>&lt;(T)<code>)&gt;)</code> for an element using the specified comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;(Of </code>&lt;(Emitter<code>)&gt;)</code>).</td>
</tr>
<tr>
<td>BinarySearch(Int32, Int32, T, IComparer&lt;(Of <code>(T)</code>)&gt;)</td>
<td>Searches a range of elements in the sorted <code>List&lt;(Of </code>&lt;(T)<code>)&gt;)</code> for an element using the specified comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;(Of </code>&lt;(Emitter<code>)&gt;)</code>).</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members

Project Mercury API Reference

ParticleEffect...:::CopyTo Method

ParticleEffect Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyTo(array&lt;T&gt;[[],[]])</td>
<td>Copies the entire List&lt;Of &lt;(T)&gt;&gt;) to a compatible one-dimensional array, starting at the beginning of the target array. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
<tr>
<td>CopyTo(array&lt;T&gt;[[], Int32])</td>
<td>Copies the entire List&lt;Of &lt;(T)&gt;&gt;) to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
<tr>
<td>CopyTo(Int32, array&lt;T&gt;[[], Int32, Int32])</td>
<td>Copies a range of elements from the List&lt;Of &lt;(T)&gt;&gt;) to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the ParticleEffect.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public virtual ParticleEffect DeepCopy()
```

### Visual Basic (Declaration)

```vbnet
Public Overridable Function DeepCopy As ParticleEffect
```

### Visual C++

```cpp
public:
    virtual ParticleEffect^ DeepCopy()
```

## Return Value

See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++
Include Protected Members
Include Inherited Members
Project Mercury API Reference
ParticleEffect..::.FindIndex Method

ParticleEffect Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **FindIndex(Predicate<Of (T)>)** | Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the entire List<Of (T)>.
(Inherited from List<Of (Emitter)>.) |
| **FindIndex(Int32, Predicate<Of (T)>)** | Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List<Of (T)> that extends from the specified index to the last element.
(Inherited from List<Of (Emitter)>.) |
| **FindIndex(Int32, Int32, Predicate<Of (T)>)** | Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List<Of (T)> that starts at the specified index and contains the specified number of elements.
(Inherited from List<Of (Emitter)>.) |
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect..::.FindLastIndex Method

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **FindLastIndex**<br>
**Predicate**<br><br>
(T>)>)) | Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the entire List<Of (T)>).<br>(Inherited from List<Of (Emitter)>). |
| **FindLastIndex**<br>
**Int32,**<br>
**Predicate**<br><br>
(T>)>)) | Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the List<Of (T)> that extends from the first element to the specified index.<br>(Inherited from List<Of (Emitter)>). |
| **FindLastIndex**<br>
**Int32,**<br>
**Int32,**<br>
**Predicate**<br><br>
(T>)>)) | Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the List<Of (T)> that contains the specified number of elements and ends at the specified index.<br>(Inherited from List<Of (Emitter)>). |
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect::IndexOf Method

ParticleEffect Class   See Also   Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| `IndexOf(T)` | Searches for the specified object and returns the zero-based index of the first occurrence within the entire `List<T>`.  
(Inherited from `List<(Of <(Emitter)>)>`). |
| `IndexOf(T, Int32)` | Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the `List<T>` that extends from the specified index to the last element.  
(Inherited from `List<(Of <(Emitter)>)>`). |
| `IndexOf(T, Int32, Int32)` | Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the `List<T>` that starts at the specified index and contains the specified number of elements.  
(Inherited from `List<(Of <(Emitter)>)>`). |
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initialises all Emitters within the ParticleEffect.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public virtual void Initialise()

Visual Basic (Declaration)
Public Overridable Sub Initialise

Visual C++
public:
virtual void Initialise()
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members  Project Mercury API Reference  ParticleEffect::LastIndexOf Method

 ParticleEffect Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| `LastIndexOf(T)`           | Searches for the specified object and returns the zero-based index of the last occurrence within the entire `List<Of <(T)>>`.
|                             | (Inherited from `List<Of <(Emitter)>>`.)                                   |
| `LastIndexOf(T, Int32)`    | Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the `List<Of <(T)>>` that extends from the first element to the specified index.
|                             | (Inherited from `List<Of <(Emitter)>>`.)                                   |
| `LastIndexOf(T, Int32, Int32)` | Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the `List<Of <(T)>>` that contains the specified number of elements and ends at the specified index.
|                             | (Inherited from `List<Of <(Emitter)>>`.)                                   |
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Loads content required by Emitters within the ParticleEffect.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public virtual void LoadContent(
    ContentManager content
)  

Visual Basic (Declaration)

Public Overridable Sub LoadContent ( _
    content As ContentManager _
)  

Visual C++

public:
    virtual void LoadContent(
        ContentManager^ content
    )  

Parameters

content
    Type: ContentManager

[Missing <param name="content"/> documentation for
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect.::.OnNameChanged Method

Raises the [E:NameChanged] event.

**Namespace:**  [ProjectMercury]

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

protected virtual void OnNameChanged(
    EventArgs e
)

Visual Basic (Declaration)

Protected Overridable Sub OnNameChanged ( _
    e As EventArgs _
)

Visual C++

protected:
virtual void OnNameChanged(
    EventArgs^ e
)

Parameters

e
Type: System::::EventArgs
The EventArgs instance containing the event data.
See Also

[ParticleEffect Class](#)
[ProjectMercury Namespace](#)

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse()()</td>
<td>Reverses the order of the elements in the entire List&lt;(Of &lt;(T)&gt;&gt;). (Inherited from List&lt;(Of &lt;(Emitter)&gt;)).</td>
</tr>
<tr>
<td>Reverse(Int32, Int32)</td>
<td>Reverses the order of the elements in the specified range. (Inherited from List&lt;(Of &lt;(Emitter)&gt;)).</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect API Reference
ParticleEffect::Sort Method

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Sort()()</code></td>
<td>Sorts the elements in the entire <code>List(Of &lt;T&gt;[])</code> using the default comparer. (Inherited from <code>List(Of &lt;Emitter&gt;[])</code>).</td>
</tr>
<tr>
<td><code>Sort(IComparer(Of &lt;T&gt;))</code></td>
<td>Sorts the elements in the entire <code>List(Of &lt;T&gt;[])</code> using the specified comparer. (Inherited from <code>List(Of &lt;Emitter&gt;[])</code>).</td>
</tr>
<tr>
<td><code>Sort(Comparison(Of &lt;T&gt;))</code></td>
<td>Sorts the elements in the entire <code>List(Of &lt;T&gt;[])</code> using the specified <code>Comparison(Of &lt;T&gt;[])</code>. (Inherited from <code>List(Of &lt;Emitter&gt;[])</code>).</td>
</tr>
<tr>
<td><code>Sort(Int32, Int32, IComparer(Of &lt;T&gt;))</code></td>
<td>Sorts the elements in a range of elements in <code>List(Of &lt;T&gt;[])</code> using the specified comparer. (Inherited from <code>List(Of &lt;Emitter&gt;[])</code>).</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Terminates all Emitters within the ParticleEffect with immediate effect.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public virtual void Terminate()

Visual Basic (Declaration)

Public Overridable Sub Terminate

Visual C++

public:
virtual void Terminate()
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Project Mercury API Reference
ParticleEffect::Trigger Method

ParticleEffect Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the ParticleEffect at the specified position.</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the ParticleEffect at the specified position.</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Triggers the ParticleEffect at the specified position.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
public virtual void Trigger(
    Vector2 position
)
```

#### Visual Basic (Declaration)

```vbnet
Public Overridable Sub Trigger (_
    position As Vector2 _
)
```

#### Visual C++

```cpp
public:
virtual void Trigger(
    Vector2 position
)
```

### Parameters

**position**

Type: Vector2

See Also

ParticleEffect Class
Trigger Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Triggers the ParticleEffect at the specified position.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
public virtual void Trigger(
    ref Vector2 position
)
```

#### Visual Basic (Declaration)

```vbnet
Public Overridable Sub Trigger ( _
    ByRef position As Vector2 _
)
```

#### Visual C++

```cpp
public:
    virtual void Trigger(
        Vector2% position
    )
```

### Parameters

**position**

Type: Vector2 %

See Also

ParticleEffect Class
Trigger Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect Class

Project Mercury API Reference

See Also, Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Update(Single)</code></td>
<td>Updates all Emitters within the ParticleEffect.</td>
</tr>
<tr>
<td><code>Update(Single, Single)</code></td>
<td><strong>Obsolete.</strong> Updates all Emitters within the ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect..::.Update Method (Single)

Updates all Emitters within the ParticleEffect.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public virtual void Update(
    float deltaSeconds
)

Visual Basic (Declaration)

Public Overridable Sub Update (_
    deltaSeconds As Single _
)

Visual C++

public:
    virtual void Update(
        float deltaSeconds
    )

Parameters

deltaSeconds
    Type: System::Single
    Elapsed frame time in whole and fractional seconds.
See Also

ParticleEffect Class
Update Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Updates all Emitters within the ParticleEffect.

**Namespace:** [ProjectMercury](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll)  
**Version:** 3.1.0.0
Syntax

C#

[ObsoleteAttribute("Use Update(deltaSeconds) instead.", false)]
public virtual void Update(
    float totalSeconds,
    float deltaSeconds
)

Visual Basic (Declaration)

<ObsoleteAttribute("Use Update(deltaSeconds) instead.", False)> _
Public Overridable Sub Update ( _
    totalSeconds As Single, _
    deltaSeconds As Single _
)

Visual C++

[ObsoleteAttribute(L"Use Update(deltaSeconds) instead.", false)]
public:
virtual void Update(
    float totalSeconds,
    float deltaSeconds
)

Parameters

totalSeconds
    Type: System::Single
    Total game time in whole and fractional seconds.

deltaSeconds
    Type: System::Single
    Elapsed frame time in whole and fractional seconds.
See Also

ParticleEffect Class
Update Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ParticleEffect` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the total number of active Particles in the ParticleEffect. Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from <code>List&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Capacity</td>
<td>Gets the number of elements actually contained in the <code>List&lt;Emitter&gt;</code>. (Inherited from <code>List&lt;Emitter&gt;</code>.)</td>
</tr>
<tr>
<td>Count</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Item</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the total number of active Particles in the ParticleEffect.

**Namespace:**  [ProjectMercury](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public int ActiveParticlesCount { get; }

Visual Basic (Declaration)

Public Readonly Property ActiveParticlesCount As Integer

Visual C++

public:
property int ActiveParticlesCount {
    int get ();
}


See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ParticleEffect Item Property

ParticleEffect Class   See Also   Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item[Int32]</td>
<td>Gets or sets the element at the specified index. (Inherited from List&lt;Emitter&gt;.)</td>
</tr>
<tr>
<td>Item[String]</td>
<td>Gets the element with the specified name. (Inherited from EmitterCollection.)</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ParticleEffect Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the name of the ParticleEffect.

**Namespace:**  [ProjectMercury](#)
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public string Name { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property Name As String
```

### Visual C++

```cpp
public:
    property String^ Name {
        String^ get ();
        void set (String^ value);
    }
```

## Field Value

The name.
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ParticleEffect` type exposes the following members.
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Occurs when name of the ParticleEffect has been changed.</td>
</tr>
</tbody>
</table>
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Occurs when name of the ParticleEffect has been changed.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  Version: 3.1.0.0
## Syntax

### C#

```csharp
public event EventHandler NameChanged
```

### Visual Basic (Declaration)

```vbnet
Public Event NameChanged As EventHandler
```

### Visual C++

```cpp
public:
    event EventHandler^ NameChanged {
        void add (EventHandler^ value);
        void remove (EventHandler^ value);
    }
```
See Also

ParticleEffect Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper Class

Defines helper methods for choosing random numbers of performing random operations.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static class RandomHelper

Visual Basic (Declaration)

Public NotInheritable Class RandomHelper

Visual C++

public ref class RandomHelper abstract sealed
Inheritance Hierarchy

System::Object
ProjectMercury::RandomHelper
See Also

RandomHelper Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper Members

[This is preliminary documentation and is subject to change.]

The RandomHelper type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChooseOne</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>NextBool</td>
<td>Returns a random boolean value.</td>
</tr>
<tr>
<td>NextFloat</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>NextInt</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>NextUnitVector</td>
<td>Returns a random two dimensional unit vector.</td>
</tr>
<tr>
<td>Variation</td>
<td>Returns a random variation of the specified value.</td>
</tr>
</tbody>
</table>
See Also

RandomHelper Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RandomHelper` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChooseOne</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>NextBool</td>
<td>Returns a random boolean value.</td>
</tr>
<tr>
<td>NextFloat</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>NextInt</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>NextUnitVector</td>
<td>Returns a random two dimensional unit vector.</td>
</tr>
<tr>
<td>Variation</td>
<td>Returns a random variation of the specified value.</td>
</tr>
</tbody>
</table>
See Also

RandomHelper Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper...:::ChooseOne Method

RandomHelper Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChooseOne(array&lt;Int32&gt;[][])</td>
<td>Chooses a random item from the specified parameters and returns it.</td>
</tr>
<tr>
<td>ChooseOne(array&lt;Single&gt;[][])</td>
<td>Chooses a random item from the specified parameters and returns it.</td>
</tr>
<tr>
<td>ChooseOne&lt;(Of &lt;(T)&gt;&gt;) (array&lt;T&gt;[][])</td>
<td>Chooses a random item from the specified parameters and returns it.</td>
</tr>
</tbody>
</table>
See Also

RandomHelper Class
RandomHelper Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Chooses a random item from the specified parameters and returns it.

**Namespace:**  [ProjectMercury](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static int ChooseOne(
    params int[] values
)

Visual Basic (Declaration)

Public Shared Function ChooseOne ( _
    ParamArray values As Integer() _
) As Integer

Visual C++

public:
    static int ChooseOne(
        ... array<int>^ values
    )

Parameters

values
    Type: array< System::<::Int32 >[]>()

[Missing <param name="values"/> documentation for
]

Return Value

[Missing <returns> documentation for
]
See Also

RandomHelper Class
ChooseOne Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Choses a random item from the specified parameters and returns it.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public static float ChooseOne(
    params float[] values
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function ChooseOne (_
    ParamArray values As Single() _
) As Single
```

### Visual C++

```cpp
public:
    static float ChooseOne(
        ... array<float>^ values
    )
```

## Parameters

values

Type: array&lt; System::Single >&[](0)[]

[Missing &lt;param name="values"/&gt; documentation for

## Return Value

[Missing &lt;returns&gt; documentation for
See Also

RandomHelper Class
ChooseOne Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper.ChooseOne(Of <T>) Method (array<T>[][])
Syntax

C#

public static T ChooseOne<T>(
    params T[] values
)

Visual Basic (Declaration)

Public Shared Function ChooseOne(Of T) ( _
    ParamArray values As T() _
) As T

Visual C++

public:
    generic<typename T>
    static T ChooseOne(
        ... array<T>^ values
    )

Parameters

values
    Type: array< T >[][]

[Missing <param name="values"/> documentation for "M:ProjectMercury.RandomHelper.ChooseOne`1(\`0[])"]
<table>
<thead>
<tr>
<th>Type Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
</tr>
</tbody>
</table>

[Missing <typeparam name="T"/> documentation for "M:ProjectMercury.RandomHelper.ChooseOne`1(0[])"]

<table>
<thead>
<tr>
<th>Return Value</th>
</tr>
</thead>
</table>

[Missing <returns> documentation for "M:ProjectMercury.RandomHelper.ChooseOne`1(0[])"]
See Also

RandomHelper Class
ChooseOne Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper Class

NextBool Method

Namespace: ProjectMercury

Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

Returns a random boolean value.
Syntax

C#

public static bool NextBool()

Visual Basic (Declaration)

Public Shared Function NextBool As Boolean

Visual C++

public:
static bool NextBool()

Return Value

See Also

RandomHelper Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper...::.NextFloat Method

RandomHelper Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>NextFloat()</code></td>
<td>Returns a random float between 0.0 and 1.0.</td>
</tr>
<tr>
<td><code>NextFloat(Single)</code></td>
<td>Returns a random float between 0.0 and the specified upper bound.</td>
</tr>
<tr>
<td><code>NextFloat(Single, Single)</code></td>
<td>Returns a random float within the specified range.</td>
</tr>
</tbody>
</table>
See Also

RandomHelper Class
RandomHelper Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a random float between 0.0 and 1.0.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

**C#**

```csharp
public static float NextFloat()
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function NextFloat As Single
```

**Visual C++**

```cpp
public:
static float NextFloat()
```

## Return Value

See Also

RandomHelper Class
NextFloat Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a random float between 0.0 and the specified upper bound.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float NextFloat(
    float max
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function NextFloat ( 
    max As Single 
) As Single
```

Visual C++

```cpp
public:
static float NextFloat( 
    float max
)
```

Parameters

max

Type: `System::Single`

The inclusive upper bound of the random number returned.

Return Value

[Missing <returns> documentation for
See Also

RandomHelper Class
NextFloat Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper...:..NextFloat Method (Single, Single)

RandomHelper Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Returns a random float within the specified range.

Namespace:  ProjectMercury
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static float NextFloat(
    float min,
    float max
)

Visual Basic (Declaration)

Public Shared Function NextFloat ( _
    min As Single, _
    max As Single _
) As Single

Visual C++

public:
static float NextFloat(
    float min,
    float max
)

Parameters

min
    Type: System::Single
    The inclusive lower bound of the random number returned.

max
    Type: System::Single
    The inclusive upper bound of the random number returned.

Return Value

See Also

RandomHelper Class
NextFloat Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RandomHelper...:::NextInt Method

RandomHelper Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>NextInt()</code></td>
<td>Returns a non-negative random whole number.</td>
</tr>
<tr>
<td><code>NextInt(Int32)</code></td>
<td>Returns a non-negative random whole number less than the specified maximum.</td>
</tr>
<tr>
<td><code>NextInt(Int32, Int32)</code></td>
<td>Returns a random number within a specified range.</td>
</tr>
</tbody>
</table>
See Also

RandomHelper Class
RandomHelper Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a non-negative random whole number.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```
public static int NextInt()
```

**Visual Basic (Declaration)**

```
Public Shared Function NextInt As Integer
```

**Visual C++**

```
public:
static int NextInt()
```

**Return Value**

See Also

RandomHelper Class
NextInt Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a non-negative random whole number less than the specified maximum.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public static int NextInt(
    int max
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function NextInt (_
    max As Integer _
) As Integer
```

**Visual C++**

```cpp
public:
static int NextInt(
    int max
)
```

### Parameters

`max`

Type: `System::::Int32`

The exclusive upper bound the random number to be generated.

### Return Value

See Also

RandomHelper Class
NextInt Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a random number within a specified range.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static int NextInt(
    int min,
    int max
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function NextInt (_
    min As Integer, _
    max As Integer _
) As Integer
```

Visual C++

```csharp
public:
static int NextInt( 
    int min,
    int max
)
```

Parameters

min
- Type: `System::::Int32`
- The inclusive lower bound of the random number returned.

max
- Type: `System::::Int32`
- The exclusive upper bound of the random number returned.

Return Value

See Also

RandomHelper Class
NextInt Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a random two dimensional unit vector.

**Namespace:** [ProjectMercury](https://example.com)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static Vector2 NextUnitVector()

Visual Basic (Declaration)

Public Shared Function NextUnitVector As Vector2

Visual C++

public:
static Vector2 NextUnitVector()

Return Value

A random two dimensional unit vector.
See Also

RandomHelper Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a random variation of the specified value.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public static float Variation(
    float value,
    float variation
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Variation ( _
    value As Single, _
    variation As Single _
) As Single
```

Visual C++

```cpp
public:
static float Variation(
    float value,
    float variation
)
```

Parameters

value
- Type: System::Single
- The value.

variation
- Type: System::Single
- The variation multiple of the value.

Return Value

[Missing <returns> documentation for
Examples

A value of 10 with a variation of 0.5 will result in a random number between 5.0 and 15.
See Also

RandomHelper Class
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a floating point object which has a definable random variation.

**Namespace:**  [ProjectMercury](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public struct VariableFloat : IEquatable<VariableFloat>

Visual Basic (Declaration)

Public Structure VariableFloat _
    Implements IEquatable(Of VariableFloat)

Visual C++

public value class VariableFloat : IEquatable<VariableFloat>
See Also

VariableFloat Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VariableFloat` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <a href="#">Object</a> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <a href="#">ValueType</a>::.GetHashCode().)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Samples the VariableFloat.</td>
</tr>
<tr>
<td><strong>Sample</strong></td>
<td>Returns a <a href="#">String</a> representation of the object.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Overrides <a href="#">ValueType</a>::.ToString().)</td>
</tr>
</tbody>
</table>
### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Overloaded</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>The base value for the VariableFloat.</td>
</tr>
<tr>
<td>Variation</td>
<td>The range of the random variation around the base value.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VariableFloat` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>The base value for the VariableFloat.</td>
</tr>
<tr>
<td>Variation</td>
<td>The range of the random variation around the base value.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
VariableFloat..::.Value Field

VariableFloat Structure  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

The base value for the VariableFloat.

Namespace:  ProjectMercury
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Value

Visual Basic (Declaration)

Public Value As Single

Visual C++

public:
    float Value
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The range of the random variation around the base value.

Namespace:  ProjectMercury
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Variation

Visual Basic (Declaration)

Public Variation As Single

Visual C++

public:
float Variation
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VariableFloat` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>Sample</strong></td>
<td>Samples the VariableFloat.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> representation of the object.</td>
</tr>
</tbody>
</table>

(Overrides [ValueType](#)::[GetHashCode()](#).)

(Inherited from [Object](#).)

(Overrides [ValueType](#)::[ToString()](#).)
### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
VariableFloat::Equals Method

VariableFloat Structure  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(VariableFloat)</code></td>
<td>Indicates whether the current object is equal to another object of the same type.</td>
</tr>
<tr>
<td><code>Equals(Object)</code></td>
<td>Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>ValueType::.Equals(Object)</code>.)</td>
</tr>
</tbody>
</table>
See Also

VariableFloat Structure
VariableFloat Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Indicates whether the current object is equal to another object of the same type.

**Namespace:** ProjectMercury

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public bool Equals(
    VariableFloat other
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Equals ( _
    other As VariableFloat _
) As Boolean
```

**Visual C++**

```cpp
public:
    virtual bool Equals(
        VariableFloat other
    ) sealed
```

**Parameters**

other

Type: `ProjectMercury::VariableFloat`

An object to compare with this object.

**Return Value**

true if the current object is equal to the other parameter; otherwise, false.

**Implements**

`IEquatable<T>`
See Also

VariableFloat Structure
Equals Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Indicates whether this instance and a specified object are equal.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public override bool Equals(
    Object obj
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function Equals (_
    obj As Object _
) As Boolean
```

Visual C++

```cpp
public:
    virtual bool Equals(
        Object^ obj
    ) override
```

Parameters

```text
obj
Type: System::Object
Another object to compare to.
```

Return Value

```text
true if obj and this instance are the same type and represent the same value; otherwise, false.
```
See Also

VariableFloat Structure
Equals Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the hash code for this instance.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override int GetHashCode()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetHashCode As Integer
```

**Visual C++**

```cpp
public:
virtual int GetHashCode() override
```

**Return Value**

A 32-bit signed integer that is the hash code for this instance.
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit(VariableFloat)</td>
<td>Implicit cast operation from VariableFloat to float.</td>
</tr>
<tr>
<td>Implicit(Single)</td>
<td>Implicit cast operator from float to VariableFloat.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat Structure
VariableFloat Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Implicit cast operation from VariableFloat to float.

**Namespace:**  ProjectMercury

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public static implicit operator float (VariableFloat value)

Visual Basic (Declaration)

Public Shared Widening Operator CType (_
    value As VariableFloat _
) As Single

Visual C++

static implicit operator float (VariableFloat value)

Parameters

value

Type: ProjectMercury::VariableFloat

[Missing <param name="value"/> documentation for

Return Value

[Missing <returns> documentation for
See Also

VariableFloat Structure
Implicit Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Implicit cast operator from float to VariableFloat.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
public static implicit operator VariableFloat (float value)
```

#### Visual Basic (Declaration)

```vbnet
Public Shared Widening Operator CType ( _
    value As Single _
) As VariableFloat
```

#### Visual C++

```cpp
static implicit operator VariableFloat (float value)
```

### Parameters

value

Type: `System::Single`

[Missing `<param name="value"/>` documentation for 

### Return Value

[Missing `<returns>` documentation for 
See Also

VariableFloat Structure
Implicit Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
VariableFloat Structure  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Samples the VariableFloat.

Namespace:  ProjectMercury
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Sample()

Visual Basic (Declaration)

Public Function Sample As Single

Visual C++

public:
float Sample()

Return Value

A randomised float value.
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a String representation of the object.

Namespace: ProjectMercury
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public override string ToString()

**Visual Basic (Declaration)**

Public Overrides Function ToString As String

**Visual C++**

public:
virtual String^ ToString() override

**Return Value**

A String representation of the object.
See Also

VariableFloat Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Vector3 object which has a definable random variation.

Namespace:  ProjectMercury  
Assembly:  ProjectMercury (in ProjectMercury.dll)  Version: 3.1.0.0
Syntax

**C#**

```csharp
public struct VariableFloat3 : IEquatable<VariableFloat3>
```

**Visual Basic (Declaration)**

```vbnet
Public Structure VariableFloat3
    Implements IEquatable(Of VariableFloat3)
```

**Visual C++**

```cpp
public value class VariableFloat3 : IEquatable<VariableFloat3>
```
See Also

VariableFloat3 Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VariableFloat3` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance. (Overrides <a href="#">ValueType::GetHashCode()</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>Sample</strong></td>
<td>Samples the VariableFloat3.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns the fully qualified type name of this instance. (Overrides <a href="#">ValueType::ToString()</a>)</td>
</tr>
</tbody>
</table>
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>The base value of the VariableFloat3.</td>
</tr>
<tr>
<td>Variation</td>
<td>The range of the random variation around the base value.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VariableFloat3` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
<td>The base value of the VariableFloat3.</td>
</tr>
<tr>
<td><strong>Variation</strong></td>
<td>The range of the random variation around the base value.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The base value of the VariableFloat3.

**Namespace:**  ProjectMercury  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector3 Value

Visual Basic (Declaration)

Public Value As Vector3

Visual C++

public:
Vector3 Value
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The range of the random variation around the base value.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector3 Variation

Visual Basic (Declaration)

Public Variation As Vector3

Visual C++

public:
Vector3 Variation
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VariableFloat3` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>Sample</strong></td>
<td>Samples the VariableFloat3.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns the fully qualified type name of this instance.</td>
</tr>
</tbody>
</table>

(Overrides [ValueType](#)::[GetHashCode](#).)

(Overrides [ValueType](#)::[ToString](#).)
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Project Mercury API Reference
VariableFloat3:::Equals Method

VariableFloat3 Structure  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(VariableFloat3)</td>
<td>Indicates whether the current object is equal to another object of the same type.</td>
</tr>
<tr>
<td>Equals(Object)</td>
<td>Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td></td>
<td>(Overrides ValueType..::.Equals(Object).)</td>
</tr>
</tbody>
</table>
See Also

VariableFloat3 Structure
VariableFloat3 Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Indicates whether the current object is equal to another object of the same type.

**Namespace:**  [ProjectMercury](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public bool Equals(
    VariableFloat3 other
)
```

Visual Basic (Declaration)

```vbnet
Public Function Equals (_
    other As VariableFloat3 _
) As Boolean
```

Visual C++

```cpp
public:
virtual bool Equals(
    VariableFloat3 other
) sealed
```

Parameters

other

Type: `ProjectMercury::VariableFloat3`
An object to compare with this object.

Return Value

true if the current object is equal to the other parameter; otherwise, false.

Implements

`IEquatable<(Of <(T)>).:.:.Equals(T)`
See Also

VariableFloat3 Structure
Equals Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Indicates whether this instance and a specified object are equal.

**Namespace:** [ProjectMercury](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public override bool Equals(
    Object obj
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function Equals ( _
    obj As Object _
) As Boolean
```

Visual C++

```cpp
public:
    virtual bool Equals(
        Object^ obj
    ) override
```

Parameters

**obj**

Type: System::Object

Another object to compare to.

Return Value

true if obj and this instance are the same type and represent the same value; otherwise, false.
See Also

VariableFloat3 Structure
Equals Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the hash code for this instance.

**Namespace:**  [ProjectMercury](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#  
public override int GetHashCode()  

Visual Basic (Declaration)  
Public Overrides Function GetHashCode As Integer  

Visual C++  
public: virtual int GetHashCode() override  

Return Value  

A 32-bit signed integer that is the hash code for this instance.
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
VariableFloat3::Implicit Method

VariableFloat3 Structure  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit(Vector3)</td>
<td>Performs an implicit conversion from Vector3 to VariableFloat3.</td>
</tr>
<tr>
<td>Implicit(VariableFloat3)</td>
<td>Performs an implicit conversion from VariableFloat3 to Vector3.</td>
</tr>
</tbody>
</table>
See Also

VariableFloat3 Structure
VariableFloat3 Members
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Performs an implicit conversion from Vector3 to `VariableFloat3`.

**Namespace:**  ProjectMercury

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```
public static implicit operator VariableFloat3 (Vector3 value)
```

**Visual Basic (Declaration)**

```
Public Shared Widening Operator CType ( _
    value As Vector3 _
) As VariableFloat3
```

**Visual C++**

```
static implicit operator VariableFloat3 (Vector3 value)
```

### Parameters

**value**

Type: Vector3  
The value.

### Return Value

The result of the conversion.
See Also

VariableFloat3 Structure
Implicit Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Performs an implicit conversion from `VariableFloat3` to `Vector3`.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public static implicit operator Vector3 (VariableFloat3 value)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Widening Operator CType (_
    value As VariableFloat3 _
) As Vector3
```

**Visual C++**

```cpp
static implicit operator Vector3 (VariableFloat3 value)
```

**Parameters**

`value`

Type: `ProjectMercury::<VariableFloat3`

The value.

**Return Value**

The result of the conversion.
See Also

VariableFloat3 Structure
Implicit Overload
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Samples the VariableFloat3.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public Vector3 Sample()

Visual Basic (Declaration)

Public Function Sample As Vector3

Visual C++

public:
Vector3 Sample()

Return Value

A randomised Vector3 value.
See Also

VariableFloat3 Structure
ProjectMercury Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns the fully qualified type name of this instance.

**Namespace:** ProjectMercury  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override string ToString()

Visual Basic (Declaration)

Public Overrides Function ToString As String

Visual C++

public:
virtual String^ ToString() override

Return Value

A String containing a fully qualified type name.
See Also

VariableFloat3 Structure
ProjectMercury_NameSpace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th><strong>Class</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CircleEmitter</td>
<td>Defines an Emitter which releases Particles in a circle or ring shape.</td>
</tr>
<tr>
<td>ConeEmitter</td>
<td>Defines an Emitter which releases particles in a beam which gradually becomes wider.</td>
</tr>
<tr>
<td>Emitter</td>
<td>Defines the base class for a Particle Emitter. The basic implementation releases Particles from a single point.</td>
</tr>
<tr>
<td>EmitterCollection</td>
<td>Defines a collection of Emitter objects. Contains extension methods for the Emitter class which aid backwards compatibility with previous releases of the engine.</td>
</tr>
<tr>
<td>EmitterCompatibilityExtensions</td>
<td>Defines an Emitter which releases Particles at a random point along a line.</td>
</tr>
<tr>
<td>LineEmitter</td>
<td>Defines an Emitter which releases Particles based on a mask array, typically from an image.</td>
</tr>
<tr>
<td>MaskEmitter</td>
<td>Emits particles in the shape of a polygon defined with the Points property.</td>
</tr>
<tr>
<td>PolygonEmitter</td>
<td>Collection of points to generate a polygon.</td>
</tr>
<tr>
<td>PolygonPointCollection</td>
<td>Defines an Emitter which releases particles in a rectangle shape.</td>
</tr>
<tr>
<td>RectEmitter</td>
<td><strong>Obsolete.</strong></td>
</tr>
<tr>
<td>TextureEmitter</td>
<td><strong>Obsolete.</strong></td>
</tr>
<tr>
<td>Delegate</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ParticleEventHandler</strong></td>
<td>Defines an event handler for a Particle related event.</td>
</tr>
</tbody>
</table>
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolygonOrigin</td>
<td>Enumerates the origin options for a polygon shape.</td>
</tr>
</tbody>
</table>

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines an Emitter which releases Particles in a circle or ring shape.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class CircleEmitter : Emitter

Visual Basic (Declaration)

Public Class CircleEmitter _
   Inherits Emitter

Visual C++

public ref class CircleEmitter : public Emitter
Inheritance Hierarchy

System..::.Object
ProjectMercury.Emitters..::.Emitter
ProjectMercury.Emitters..::.CircleEmitter
See Also

CircleEmitter Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The CircleEmitter type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CircleEmitter</td>
<td>Initializes a new instance of the CircleEmitter class</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an uninitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <a href="#">Emitter::DeepCopy()</a>.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from <a href="#">Emitter</a>.</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overides <a href="#">Emitter::GenerateOffsetAndForce(Vector2%, Vector2%)</a>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>OnNameChanged</strong></td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>OnParticleReleased</strong></td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>OnParticleRetired</strong></td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Terminate</strong></td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Initialize** | Initializes the Emitter.  
(Defined by [EmitterCompatibilityExtensions](#).) |
| **Update** | Updates the Emitter.  
(Defined by [EmitterCompatibilityExtensions](#).) |
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Radiate</strong></td>
<td>True if particles should radiate away from the center of the circle, else false.</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td><strong>Ring</strong></td>
<td>True if particles should be spawned only on the edge of the circle, else false.</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Initialised</td>
<td>True if the Emitter object has been initialised, else false. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Radius</td>
<td>Defines the radius of the circle.</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
CircleEmitter Constructor

Initializes a new instance of the CircleEmitter class

**Namespace:** [ProjectMercury.Emitters](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#
public CircleEmitter()

Visual Basic (Declaration)
Public Sub New

Visual C++
public:
CircleEmitter()
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `CircleEmitter` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>Radiate</strong></td>
<td>True if particles should radiate away from the center of the circle, else false.</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>Ring</strong></td>
<td>True if particles should be spawned only on the edge of the circle, else false.</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
True if particles should radiate away from the center of the circle, else false.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public bool Radiate

Visual Basic (Declaration)

Public Radiate As Boolean

Visual C++

public:
    bool Radiate
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
True if particles should be spawned only on the edge of the circle, else false.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public bool Ring

Visual Basic (Declaration)

Public Ring As Boolean

Visual C++

public:
  bool Ring
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `CircleEmitter` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an unitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter:: DeepCopy().)</td>
</tr>
<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>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a</td>
</tr>
<tr>
<td></td>
<td>minimum trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is</td>
</tr>
<tr>
<td></td>
<td>released.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter:: GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</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>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
</tbody>
</table>
- **MemberwiseClone** (Inherited from **Object**.)
- **OnNameChanged**
  Raises the NameChanged event.
  (Inherited from **Emitter**.)
- **OnParticleReleased**
  Raises the ParticleReleased event.
  (Inherited from **Emitter**.)
- **OnParticleRetired**
  Raises the ParticleRetired event.
  (Inherited from **Emitter**.)
- **Terminate**
  Terminates the emitter immediately.
  (Inherited from **Emitter**.)
- **ToString**
  Returns a **String** that represents the current
  **Object**.
  (Inherited from **Object**.)
- **Trigger**
  Overloaded.
- **Update**
  Updates the Emitter and all Particles within.
  (Inherited from **Emitter**.)
### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter.</td>
<td>Defined by <a href="https://example.com">EmitterCompatibilityExtensions</a></td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter.</td>
<td>Defined by <a href="https://example.com">EmitterCompatibilityExtensions</a></td>
</tr>
</tbody>
</table>
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an initialised deep copy of the Emitter.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
# Syntax

**C#**

```csharp
public override Emitter DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Emitter
```

**Visual C++**

```cpp
public:
virtual Emitter^ DeepCopy() override
```

## Return Value

A deep copy of the Emitter.
See Also

CircleEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected override void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)

Visual Basic (Declaration)

Protected Overrides Sub GenerateOffsetAndForce(
    <OutAttribute> ByRef offset As Vector2,
    <OutAttribute> ByRef force As Vector2
)

Visual C++

protected:
virtual void GenerateOffsetAndForce(
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
) override

Parameters

offset
    Type: Vector2 %
    The offset of the Particle from the trigger location.

force
    Type: Vector2 %
    A unit vector defining the initial force of the Particle.
See Also

CircleEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
CircleEmitter:::Initialise Method

CircleEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialise()()</td>
<td>Initialises the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Initialise(Int32, Single)</td>
<td>Initialises the Emitter. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

CircleEmitter Class
CircleEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
CircleEmitter Class

See Also

Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position...</td>
<td>Emitter</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position...</td>
<td>Emitter</td>
</tr>
</tbody>
</table>
See Also

CircleEmitter Class
CircleEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
CircleEmitter type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **ActiveParticlesCount** | Gets the number of Particles which are currently active.  
(Inherited from Emitter.)                           |
| **Budget**       | Gets or sets the number of Particles which are available to the Emitter.  
(Inherited from Emitter.)                           |
| **Initialised**  | True if the Emitter object has been initialised, else false.  
(Inherited from Emitter.)                           |
| **Name**         | Gets or sets the name of the Emitter.  
(Inherited from Emitter.)                           |
| **Radius**       | Defines the radius of the circle.                                           |
| **ReleaseQuantity** | Gets or sets the number of Particles which will be released on each trigger.  
(Inherited from Emitter.)                           |
| **Term**         | Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.  
(Inherited from Emitter.)                           |
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the radius of the circle.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Radius { get; set; }

Visual Basic (Declaration)

Public Property Radius As Single

Visual C++

public:
property float Radius {
    float get ();
    void set (float value);
}
See Also

CircleEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The CircleEmitter type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

CircleEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines an Emitter which releases particles in a beam which gradually becomes wider.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class ConeEmitter : Emitter

Visual Basic (Declaration)

Public Class ConeEmitter
    Inherits Emitter

Visual C++

public ref class ConeEmitter : public Emitter
Inheritance Hierarchy

System:::Object
  ProjectMercury.Emitters:::Emitter
    ProjectMercury.Emitters:::ConeEmitter
See Also

ConeEmitter Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ConeEmitter type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConeEmitter</td>
<td>Initializes a new instance of the <em>ConeEmitter</em> class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an unitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <a href="#">Emitter.:::DeepCopy()</a>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is released.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <a href="#">Emitter.:::GenerateOffsetAndForce(Vector2%, Vector2%)</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
</tbody>
</table>
MemberwiseClone (Inherited from Object.)

OnNameChanged Raises the NameChanged event. (Inherited from Emitter.)

OnParticleReleased Raises the ParticleReleased event. (Inherited from Emitter.)

OnParticleRetired Raises the ParticleRetired event. (Inherited from Emitter.)

Terminate Terminates the emitter immediately. (Inherited from Emitter.)

ToString Returns a String that represents the current Object. (Inherited from Object.)

Trigger Overloaded.

Update Updates the Emitter and all Particles within. (Inherited from Emitter.)
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
</tbody>
</table>
**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can</td>
</tr>
<tr>
<td></td>
<td>be triggered).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter,</td>
</tr>
<tr>
<td></td>
<td>expressed in whole and fractional seconds. Triggers which occur during this</td>
</tr>
<tr>
<td></td>
<td>period will be ignored.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
</tbody>
</table>
- **ReleaseRotation**
  Gets or sets the rotation of released Particles.
  (Inherited from [Emitter](#).)

- **ReleaseScale**
  Gets or sets the scale of released particles.
  (Inherited from [Emitter](#).)

- **ReleaseSpeed**
  Gets or sets the speed at which Particles travel when they are released.
  (Inherited from [Emitter](#).)

- **TriggerOffset**
  The Emitters trigger offset in relation to the ParticleEffect.
  (Inherited from [Emitter](#).)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActiveParticlesCount</strong></td>
<td>Gets the number of Particles which are currently active. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td>Gets or sets the number of Particles which are available to the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ConeAngle</strong></td>
<td>The angle (in radians) from edge to edge of the ConeEmitters beam.</td>
</tr>
<tr>
<td><strong>Direction</strong></td>
<td>The angle (in radians) that the ConeEmitters beam is facing.</td>
</tr>
<tr>
<td><strong>Initialised</strong></td>
<td>True if the Emitter object has been initialised, else false. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets or sets the name of the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseQuantity</strong></td>
<td>Gets or sets the number of Particles which will be released on each trigger. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Term</strong></td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NameChanged</strong></td>
<td>Raised when the name of the Emitter has been changed. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ParticleReleased</strong></td>
<td>Raised when a Particle is released by the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ParticleRetired</strong></td>
<td>Raised when a Particle expires. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the ConeEmitter class

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public ConeEmitter()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
ConeEmitter()
```
See Also

ConeEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ConeEmitter` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReleaseRotation</td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ReleaseScale</td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ReleaseSpeed</td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>TriggerOffset</td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class  
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ConeEmitter type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter:::DeepCopy().)</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>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum</td>
</tr>
<tr>
<td></td>
<td>trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is released.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter:::GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</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>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>OnNameChanged</td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>OnParticleReleased</td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>OnParticleRetired</td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Terminate</td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</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>
<tr>
<td>Trigger</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an uninitialised deep copy of the Emitter.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public override Emitter DeepCopy()

**Visual Basic (Declaration)**

Public Overrides Function DeepCopy As Emitter

**Visual C++**

public:
virtual Emitter^ DeepCopy() override

**Return Value**

A deep copy of the Emitter.
See Also

ConeEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected override void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)

Visual Basic (Declaration)

Protected Overrides Sub GenerateOffsetAndForce (_
    <OutAttribute> ByRef offset As Vector2, _
    <OutAttribute> ByRef force As Vector2 _
)

Visual C++

protected:
virtual void GenerateOffsetAndForce( 
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
) override

Parameters

offset
    Type: Vector2 %
    The offset of the Particle from the trigger location.

force
    Type: Vector2 %
    A unit vector defining the initial force of the Particle.
See Also

ConeEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialise()()</td>
<td>Initialises the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Initialise(Int32, Single)</td>
<td>Initialises the Emitter. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class
ConeEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C# Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Project Mercury API Reference
ConeEmitter:::Trigger Method

ConeEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class
ConeEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ConeEmitter type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ConeAngle</td>
<td>The angle (in radians) from edge to edge of the ConeEmitters beam.</td>
</tr>
<tr>
<td>Direction</td>
<td>The angle (in radians) that the ConeEmitters beam is facing.</td>
</tr>
<tr>
<td>Initialised</td>
<td>True if the Emitter object has been initialised, else false. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The angle (in radians) from edge to edge of the ConeEmitters beam.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float ConeAngle { get; set; }

Visual Basic (Declaration)

Public Property ConeAngle As Single

Visual C++

public:
 property float ConeAngle {
    float get ();
    void set (float value);
}
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System::.ArgumentOutOfRangeException</code></td>
<td>Thrown if the specified value is either too small or too large.</td>
</tr>
</tbody>
</table>
See Also

ConeEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The angle (in radians) that the ConeEmitters beam is facing.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Direction { get; set; }

Visual Basic (Declaration)

Public Property Direction As Single

Visual C++

public:
property float Direction {
    float get ();
    void set (float value);
}
See Also

ConeEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ConeEmitter type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
</tbody>
</table>

*Inherited from [Emitter](#).*
See Also

ConeEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the base class for a Particle Emitter. The basic implementation releases Particles from a single point.

**Namespace:**  ProjectMercury.Emitters

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public class Emitter
```

**Visual Basic (Declaration)**

```vbnet
Public Class Emitter
```

**Visual C++**

```cpp
public ref class Emitter
```
Inheritance Hierarchy

System:::Object
ProjectMercury.Emitters:::Emitter
  ProjectMercury.Emitters:::CircleEmitter
  ProjectMercury.Emitters:::ConeEmitter
  ProjectMercury.Emitters:::LineEmitter
  ProjectMercury.Emitters:::MaskEmitter
  ProjectMercury.Emitters:::PolygonEmitter
  ProjectMercury.Emitters:::RectEmitter
  ProjectMercury.Emitters:::TextureEmitter
See Also

Emitter Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Emitter type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emitter</td>
<td>Instantiates a new instance of the Emitter class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</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>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum</td>
</tr>
<tr>
<td></td>
<td>trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is</td>
</tr>
<tr>
<td></td>
<td>released.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</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>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</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>OnNameChanged</td>
<td>Raises the NameChangedEventArgs event.</td>
</tr>
<tr>
<td>OnParticleReleased</td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td>OnParticleRetired</td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td>Terminate</td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td></td>
<td>Returns a String that represents the current state of the Emitter.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the Emitter. (Defined by <code>EmitterCompatibilityExtensions</code>.)</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter. (Defined by <code>EmitterCompatibilityExtensions</code>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered).</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored.</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter.</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter.</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles.</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method.</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released.</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td>initialised</td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Instantiates a new instance of the Emitter class.

**Namespace:**  [ProjectMercury.Emitters](https://example.com)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Emitter()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
Emitter()
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Emitter type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered).</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored.</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter.</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter.</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles.</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method.</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released.</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitter's trigger offset in relation to the ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The blending mode to be used by Renderers when rendering this Emitter.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:** 3.1.0.0
Syntax

C#

public BlendMode BlendMode

Visual Basic (Declaration)

Public BlendMode As BlendMode

Visual C++

public:
BlendMode BlendMode
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered).

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0

[This is preliminary documentation and is subject to change.]
## Syntax

### C#

```csharp
public bool Enabled
```

### Visual Basic (Declaration)

```vbnet
Public Enabled As Boolean
```

### Visual C++

```cpp
public:
    bool Enabled
```
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored.

_namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public float MinimumTriggerPeriod
```

**Visual Basic (Declaration)**

```vbnet
Public MinimumTriggerPeriod As Single
```

**Visual C++**

```cpp
public:
float MinimumTriggerPeriod
```
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the collection of Modifiers which are acting upon the Emitter.

**Namespace:** [ProjectMercury.Emitters](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public ModifierCollection Modifiers

Visual Basic (Declaration)

Public Modifiers As ModifierCollection

Visual C++

public:
ModifierCollection^ Modifiers
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter Field

[This is preliminary documentation and is subject to change.]

Gets or sets the array of particles managed by the emitter.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public Particle[] Particles
```

**Visual Basic (Declaration)**

```vbnet
Public Particles As Particle()
```

**Visual C++**

```cpp
public:
array<Particle>^ Particles
```
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the Texture2D used to display the Particles.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Texture2D ParticleTexture

Visual Basic (Declaration)

Public ParticleTexture As Texture2D

Visual C++

public:
Texture2D^ ParticleTexture
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the asset name of a texture to load in the LoadContent method.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public string ParticleTextureAssetName
```

**Visual Basic (Declaration)**

```vbnet
Public ParticleTextureAssetName As String
```

**Visual C++**

```cpp
public:
String^ ParticleTextureAssetName
```
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter...::ReleaseColour Field

Get or set the colour of released Particles.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public  VariableFloat3  ReleaseColour

Visual Basic (Declaration)
Public ReleaseColour  As  VariableFloat3

Visual C++
public:
VariableFloat3  ReleaseColour
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the initial impulse applied to Particles as they are released.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 ReleaseImpulse

Visual Basic (Declaration)

Public ReleaseImpulse As Vector2

Visual C++

public:
Vector2 ReleaseImpulse
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the opacity of released Particles.

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public VariableFloat ReleaseOpacity

Visual Basic (Declaration)

Public ReleaseOpacity As VariableFloat

Visual C++

public:
VariableFloat ReleaseOpacity
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the rotation of released Particles.

**Namespace:**  ProjectMercury.Emitters

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public VariableFloat ReleaseRotation

Visual Basic (Declaration)

Public ReleaseRotation As VariableFloat

Visual C++

public:
VariableFloat ReleaseRotation
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the scale of released particles.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public `VariableFloat` ReleaseScale

**Visual Basic (Declaration)**

Public ReleaseScale As `VariableFloat`

**Visual C++**

public:
`VariableFloat` ReleaseScale
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the speed at which Particles travel when they are released.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public VariableFloat ReleaseSpeed

Visual Basic (Declaration)

Public ReleaseSpeed As VariableFloat

Visual C++

public:
VariableFloat ReleaseSpeed
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Emitters trigger offset in relation to the ParticleEffect.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 TriggerOffset

Visual Basic (Declaration)

Public TriggerOffset As Vector2

Visual C++

public:
Vector2 TriggerOffset
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Emitter type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</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. (Inherited from Object.)</td>
</tr>
<tr>
<td>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is released.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>OnNameChanged</td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td>OnParticleReleased</td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td>OnParticleRetired</td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td>Terminate</td>
<td>Terminates the emitter immediately. Returns a String that represents the current</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td><code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
</tbody>
</table>
# Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by <a href="https://www.example.com">EmitterCompatibilityExtensions</a>).</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by <a href="https://www.example.com">EmitterCompatibilityExtensions</a>).</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Copies the fields of the Emitter base class into the specified Emitter.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected void CopyBaseFields(
    Emitter emitter
)

Visual Basic (Declaration)

Protected Sub CopyBaseFields (_
    emitter As Emitter _
)

Visual C++

protected:
void CopyBaseFields(
    Emitter^ emitter
)

Parameters

emitter
  Type: ProjectMercury.Emitters:::Emitter
  The Emitter which will be copied into.
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an uninitialised deep copy of the Emitter.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public virtual Emitter DeepCopy()

Visual Basic (Declaration)

Public Overridable Function DeepCopy As Emitter

Visual C++

public:
virtual Emitter^ DeepCopy()

Return Value

A deep copy of the Emitter.
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public void ForceNextTrigger();

Visual Basic (Declaration)

Public Sub ForceNextTrigger

Visual C++

public:
void ForceNextTrigger();
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected virtual void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)

Visual Basic (Declaration)

Protected Overridable Sub GenerateOffsetAndForce (_
    <OutAttribute> ByRef offset As Vector2, _
    <OutAttribute> ByRef force As Vector2 _
)

Visual C++

protected:
virtual void GenerateOffsetAndForce(
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
)

Parameters

offset
    Type: Vector2 %
    The offset of the Particle from the trigger location.

force
    Type: Vector2 %
    A unit vector defining the initial force of the Particle.
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter Initialise Method

Emitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Initialise()</code></td>
<td>Initialises the Emitter.</td>
</tr>
<tr>
<td><code>Initialise(Int32, Single)</code></td>
<td>Initialises the Emitter.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
Emitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initialises the Emitter.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public virtual void Initialise()

Visual Basic (Declaration)

Public Overridable Sub Initialise

Visual C++

public:
virtual void Initialise()
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System.Exception.InvalidOperationException</code></td>
<td>Thrown if the Term and/or Budget properties have not been set.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
Initialise Overload
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initialise Method (Int32, Single)

Initialises the Emitter.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public void Initialise(
    int budget,
    float term
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Initialise (_
    budget As Integer, _
    term As Single _
)
```

Visual C++

```cpp
public:
void Initialise(
    int budget,
    float term
)
```

Parameters

budget
Type: `System::::Int32`
The number of Particles which are available to the Emitter.

term
Type: `System::::Single`
The length of time that released Particles will remain active, in whole and fractional seconds.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System..:..ArgumentException</code></td>
<td>Thrown if the budget parameter is less than one, or if the term parameter is not a positive value.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
Initialise Overload
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Loads resources required by the Emitter via a ContentManager.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public virtual void LoadContent(FileDownload content)

Visual Basic (Declaration)

Public Overridable Sub LoadContent(content As ContentManager)

Visual C++

public:
virtual void LoadContent(
           ContentManager^ content
)

Parameters

content
Type: ContentManager
The ContentManager used to load resources.
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentLoadException</td>
<td>Thrown if the asset defined in the ParticleTextureAssetName property could not be loaded.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Raises the NameChanged event.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected virtual void OnNameChanged(
    EventArgs e
)

Visual Basic (Declaration)

Protected Overridable Sub OnNameChanged ( _
    e As EventArgs _
)

Visual C++

protected:
virtual void OnNameChanged(
    EventArgs^ e
)

Parameters

e

Type: System::::EventArgs

[Missing <param name="e"/> documentation for "M:ProjectMercury.Emitters.Emitter.OnNameChanged(System.EventArgs)" ]
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Raises the ParticleReleased event.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  Version: 3.1.0.0
## Syntax

### C#

```csharp
protected virtual void OnParticleReleased(
    ref Particle particle
)
```

### Visual Basic (Declaration)

```vbnet
Protected Overridable Sub OnParticleReleased ( _
    ByVal particle As Particle _
)
```

### Visual C++

```cpp
protected:
virtual void OnParticleReleased(
    Particle% particle
)
```

## Parameters

**particle**

Type: `ProjectMercury::__Particle`

The particle which has been released.
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Raises the ParticleRetired event.

**Namespace:** `ProjectMercury.Emitters`  
**Assembly:** `ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0`
**Syntax**

**C#**

```csharp
protected virtual void OnParticleRetired(
    ref Particle particle
)
```

**Visual Basic (Declaration)**

```vbnet
Protected Overridable Sub OnParticleRetired ( _
    ByVal particle As Particle _
) 
```

**Visual C++**

```cpp
protected:
virtual void OnParticleRetired(
    Particle* particle
) 
```

**Parameters**

`particle`  
Type: `ProjectMercury::Particle`  
The particle which has been retired.
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Terminates the emitter immediately.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public void Terminate()

Visual Basic (Declaration)

Public Sub Terminate

Visual C++

public:
void Terminate()
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter Class

See Also

Send Feedback

[This is preliminary documentation and is subject to change.]
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position...</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position...</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
Emitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Triggers the Emitter at the specified position...

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public void Trigger(
    Vector2 position
)

Visual Basic (Declaration)

Public Sub Trigger ( _
    position As Vector2 _
)

Visual C++

public:
void Trigger(
    Vector2 position
)

Parameters

position
    Type: Vector2

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System::InvalidOperationException</strong></td>
<td>Thrown if the Emitter has not been initialised.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
Trigger Overload
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Triggers the Emitter at the specified position...

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public void Trigger(
    ref Vector2 triggerPosition
)

Visual Basic (Declaration)

Public Sub Trigger (_
    ByRef triggerPosition As Vector2 _
)

Visual C++

public:
void Trigger(
    Vector2% triggerPosition
)

Parameters

triggerPosition
    Type: Vector2 %

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System:::InvalidOperationException</td>
<td>Thrown if the Emitter has not been initialised.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
Trigger Overload
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter...:.:Update Method

Emitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Updates the Emitter and all Particles within.

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public void Update(
    float deltaSeconds
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Update ( _
    deltaSeconds As Single _
)
```

**Visual C++**

```cpp
public:
void Update(
    float deltaSeconds
)
```

### Parameters

`deltaSeconds`

Type: `System::Single`

Elapsed frame time in whole and fractional seconds.
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter Properties

Emitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

The Emitter type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td>initialised</td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the number of Particles which are currently active.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public int ActiveParticlesCount { get; }

Visual Basic (Declaration)

Public ReadOnly Property ActiveParticlesCount As Integer

Visual C++

public:
property int ActiveParticlesCount {
    int get ();
}

See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitter...:::Budget Property

Gets or sets the number of Particles which are available to the Emitter.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public int Budget { get; set; }

Visual Basic (Declaration)

Public Property Budget As Integer

Visual C++

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

## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System:::InvalidOperationException</code></td>
<td>Thrown if trying to set this property after the Emitter has been initialised.</td>
</tr>
<tr>
<td><code>System:::ArgumentOutOfRangeException</code></td>
<td>Thrown if the specified value is less than 1.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
True if the Emitter object has been initialised, else false.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public bool Initialised { get; private set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Initialised As Boolean
```

**Visual C++**

```cpp
public:
property bool Initialised {
    bool get ();
    void set (bool value);
}
```
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitters Name Property

Gets or sets the name of the Emitter.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

**C#**

public `string` Name { get; set; }

**Visual Basic (Declaration)**

Public Property Name As `String`

**Visual C++**

public:
property `String^` Name {
    `String^` get ();
    void set (`String^` value);
}
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the number of Particles which will be released on each trigger.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public int ReleaseQuantity { get; set; }

Visual Basic (Declaration)

Public Property ReleaseQuantity As Integer

Visual C++

public:
property int ReleaseQuantity {
    int get ();
    void set (int value);
}
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System:::ArgumentOutOfRangeException</code></td>
<td>Thrown if the specified value is less than 1.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emitted	Term Property

Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public float Term { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property Term As Single
```

### Visual C++

```cpp
public:
property float Term {
    float get ();
    void set (float value);
}
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System:::InvalidOperationException</code></td>
<td>Thrown if trying to set this property after the Emitter has been initialised.</td>
</tr>
<tr>
<td><code>System:::ArgumentOutOfRangeException</code></td>
<td>Thrown if the supplied value is less than or equal to 0.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `Emitter` type exposes the following members.
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NameChanged</strong></td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td><strong>ParticleReleased</strong></td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td><strong>ParticleRetired</strong></td>
<td>Raised when a Particle expires.</td>
</tr>
</tbody>
</table>
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Raised when the name of the Emitter has been changed.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

global event EventHandler NameChanged

Visual Basic (Declaration)

Public Event NameChanged As EventHandler

Visual C++

public:
    event EventHandler^ NameChanged {
        void add (EventHandler^ value);
        void remove (EventHandler^ value);
    }
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Raised when a Particle is released by the Emitter.

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```
public event ParticleEventHandler ParticleReleased
```

**Visual Basic (Declaration)**

```
Public Event ParticleReleased As ParticleEventHandler
```

**Visual C++**

```
public:
    event ParticleEventHandler^ ParticleReleased {
        void add (ParticleEventHandler^ value);
        void remove (ParticleEventHandler^ value);
    }
```
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Raised when a Particle expires.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public event ParticleEventHandler ParticleRetired

Visual Basic (Declaration)

Public Event ParticleRetired As ParticleEventHandler

Visual C++

public:
    event ParticleEventHandler^ ParticleRetired {
        void add (ParticleEventHandler^ value);
        void remove (ParticleEventHandler^ value);
    }
See Also

Emitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection Class

Defines a collection of Emitter objects.

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public class EmitterCollection : List<Emitter>

Visual Basic (Declaration)

Public Class EmitterCollection
    Inherits List(Of Emitter)

Visual C++

public ref class EmitterCollection : public List<Emitter>
Inheritance Hierarchy

System:::Object
System.Collections.Generic:::List(Of Emitter)
ProjectMercury.Emitters:::EmitterCollection
ProjectMercury:::ParticleEffect
See Also

EmitterCollection Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **EmitterCollection** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmitterCollection</td>
<td>Initializes a new instance of the EmitterCollection class</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Add**         | Adds an object to the end of the `List<Of <(T)>>`.
|                 | (Inherited from `List<Of <(Emitter)>`).) |
| **AddRange**    | Adds the elements of the specified collection to the end of the `List<Of <(T)>>`.
|                 | (Inherited from `List<Of <(Emitter)>`).) |
| **AsReadOnly**  | Returns a read-only `IList<Of <(T)>>>` wrapper for the current collection. |
|                 | (Inherited from `List<Of <(Emitter)>`).) |
| **BinarySearch**| Overloaded. |
| **Clear**       | Removes all elements from the `List<Of <(T)>>`.
|                 | (Inherited from `List<Of <(Emitter)>`).) |
| **Contains**    | Determines whether an element is in the `List<Of <(T)>>>`.
|                 | (Inherited from `List<Of <(Emitter)>`).) |
| **ConvertAll<Of <(TOutput)>>** | Converts the elements in the current `List<Of <(T)>>` to another type, and returns a list containing the converted elements. |
|                 | (Inherited from `List<Of <(Emitter)>`).) |
| **CopyTo**      | Overloaded. |
| **Equals**      | Determines whether the specified `Object` is equal to the current `Object`. |
|                 | (Inherited from `Object`.) |
| **Exists**      | Determines whether the `List<Of <(T)>>` contains elements that match the conditions defined by the specified predicate. |
|                 | (Inherited from `List<Of <(Emitter)>>`).) |
| **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`.) |
| **Searches**    | Searches for an element that matches the conditions |
Find defined by the specified predicate, and returns the first occurrence within the entire List<Of (T)>.
(Inherited from List<Of (Emitter)>.)

FindAll Retrieves all the elements that match the conditions defined by the specified predicate.
(Inherited from List<Of (Emitter)>.)

FindIndex Overloaded.

FindLast Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire List<Of (T)>.
(Inherited from List<Of (Emitter)>.)

FindLastIndex Overloaded.

ForEach List<Of (T)>.
(Inherited from List<Of (Emitter)>.)

GetEnumerator Returns a shallow copy of a range of elements in the source List<Of (T)>.
(Inherited from List<Of (Emitter)>.)

GetHashCode Serves as a hash function for a particular type.
(Inherited from Object.)

GetRange Gets the Type of the current instance.
(Inherited from Object.)

IndexOOf Overloaded.

Insert Inserts an element into the List<Of (T)> at the specified index.
(Inherited from List<Of (Emitter)>.)

InsertRange Inserts the elements of a collection into the List<Of (T)> at the specified index.
(Inherited from List<Of (Emitter)>.)

LastIndexOf Overloaded.

MemberwiseClone Creates a shallow copy of the current Object.
(Inherited from Object.)

Removes the first occurrence of a specific object
- **Remove** from the `List<T>`.  
  (Inherited from `List<Emitter>`.)
- **RemoveAll** removes all the elements that match the conditions defined by the specified predicate.  
  (Inherited from `List<Emitter>`.)
- **RemoveAt** removes the element at the specified index of the `List<T>`.  
  (Inherited from `List<Emitter>`.)
- **RemoveRange** removes a range of elements from the `List<T>`.  
  (Inherited from `List<Emitter>`.)
- **Reverse** is overloaded.
- **Sort** is overloaded.
- **ToArray** copies the elements of the `List<T>` to a new array.  
  (Inherited from `List<Emitter>`.)
- **ToString** returns a `String` that represents the current `Object`.  
  (Inherited from `Object`.)
- **TrimExcess** sets the capacity to the actual number of elements in the `List<T>`, if that number is less than a threshold value.  
  (Inherited from `List<Emitter>`.)
- **TrueForAll** determines whether every element in the `List<T>` matches the conditions defined by the specified predicate.  
  (Inherited from `List<Emitter>`.)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Capacity | Gets or sets the total number of elements the internal data structure can hold without resizing.  
  (Inherited from List<T>). |
| Count   | Gets the number of elements actually contained in the List<T>.  
  (Inherited from List<T>). |
| Item    | Overloaded. |
See Also

EmitterCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection Constructor

Initializes a new instance of the `EmitterCollection` class

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public EmitterCollection()

Visual Basic (Declaration)
Public Sub New

Visual C++
public:
EmitterCollection()
See Also

EmitterCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection type exposes the following members.

[This is preliminary documentation and is subject to change.]
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Add        | Adds an object to the end of the List<Of (T)>.
   (Inherited from List<Of (Emitter)>.)  |
| AddRange   | Adds the elements of the specified collection to the end of the List<Of (T>).  |
   (Inherited from List<Of (Emitter)>.)  |
| AsReadOnly | Returns a read-only IList<Of (T)> wrapper for the current collection.  |
   (Inherited from List<Of (Emitter)>.)  |
| BinarySearch | Overloaded.                  |
| Clear      | Removes all elements from the List<Of (T)>.
   (Inherited from List<Of (Emitter)>.)  |
| Contains   | Determines whether an element is in the List<Of (T>).  |
   (Inherited from List<Of (Emitter)>.)  |
| ConvertAll<Of (TOutput)> | Converts the elements in the current List<Of (T)><Of (TOutput)> to another type, and returns a list containing the converted elements.  |
   (Inherited from List<Of (Emitter)>.)  |
| CopyTo     | Overloaded.                  |
| Equals     | Determines whether the specified Object is equal to the current Object.  |
   (Inherited from Object.)  |
| Exists     | Determines whether the List<Of (T)> contains elements that match the conditions defined by the specified predicate.
   (Inherited from List<Of (Emitter)>.)  |
| 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.)  |
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find</strong></td>
<td>defined by the specified predicate, and returns the first occurrence within the entire List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>FindAll</strong></td>
<td>Retrieves all the elements that match the conditions defined by the specified predicate. (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>FindIndex</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>FindLast</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>FindLastIndex</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ForEach</strong></td>
<td>Performs the specified action on each element of the List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetRange</strong></td>
<td>Creates a shallow copy of a range of elements in the source List(Of(Of(T)&gt;)). (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>IndexOf</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Inserts an element into the List(Of(Of(T)&gt;)) at the specified index. (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>InsertRange</strong></td>
<td>Inserts the elements of a collection into the List(Of(Of(T)&gt;)) at the specified index. (Inherited from List(Of(Of(Emitter)&gt;)).)</td>
</tr>
<tr>
<td><strong>LastIndexOf</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>
</tbody>
</table>

Removes the first occurrence of a specific object
Remove from the `List(Of T)`.  
(Inherited from `List(Of Emitter)`).

Removes the all the elements that match the 
conditions defined by the specified predicate.  
(Inherited from `List(Of Emitter)`).

Removes the element at the specified index of the

`RemoveAt` `List(Of T)`.  
(Inherited from `List(Of Emitter)`).

Removes a range of elements from the `List(Of T)`.  
(Inherited from `List(Of Emitter)`).

Reverse Overloaded.

Sort Overloaded.

`ToArray` Copies the elements of the `List(Of T)` to a new array.  
(Inherited from `List(Of Emitter)`).

`ToString` Returns a `String` that represents the current `Object`.  
(Inherited from `Object`).

Sets the capacity to the actual number of elements 
in the `List(Of T)`, if that number is less than a threshold value.  
(Inherited from `List(Of Emitter)`).

Determines whether every element in the `List(Of T)` matches the conditions defined by the specified predicate.  
(Inherited from `List(Of Emitter)`).
See Also

EmitterCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinarySearch(T)</td>
<td>Searches the entire sorted List&lt;(Of &lt;(T)&gt;)&gt; for an element using the default comparer and returns the zero-based index of the element. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
<tr>
<td>BinarySearch(T, IComparer&lt;Of &lt;(T)&gt;))</td>
<td>Searches the entire sorted List&lt;(Of &lt;(T)&gt;)&gt; for an element using the specified comparer and returns the zero-based index of the element. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
<tr>
<td>BinarySearch(Int32, Int32, T, IComparer&lt;Of &lt;(T)&gt;))</td>
<td>Searches a range of elements in the sorted List&lt;(Of &lt;(T)&gt;)&gt; for an element using the specified comparer and returns the zero-based index of the element. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection.:::CopyTo Method

EmitterCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CopyTo(array&lt;T&gt;[][])</code></td>
<td>Copies the entire <code>List&lt;Of &lt;(T)&gt;&gt;</code> to a compatible one-dimensional array, starting at the beginning of the target array. (Inherited from <code>List&lt;Of &lt;(Emitter)&gt;</code>).)</td>
</tr>
<tr>
<td><code>CopyTo(array&lt;T&gt;[][], Int32)</code></td>
<td>Copies the entire <code>List&lt;Of &lt;(T)&gt;&gt;</code> to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from <code>List&lt;Of &lt;(Emitter)&gt;</code>).)</td>
</tr>
<tr>
<td><code>CopyTo(Int32, array&lt;T&gt;[][], Int32, Int32)</code></td>
<td>Copies a range of elements from the <code>List&lt;Of &lt;(T)&gt;&gt;</code> to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from <code>List&lt;Of &lt;(Emitter)&gt;</code>).)</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FindIndex(Predicate&lt;Of (T)&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the entire List&lt;Of (T)&gt;. (Inherited from List&lt;Of (Emitter)&gt;.)</td>
</tr>
<tr>
<td><strong>FindIndex(Int32, Predicate&lt;Of (T)&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List&lt;Of (T)&gt; that extends from the specified index to the last element. (Inherited from List&lt;Of (Emitter)&gt;.)</td>
</tr>
<tr>
<td><strong>FindIndex(Int32, Int32, Predicate&lt;Of (T)&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List&lt;Of (T)&gt; that starts at the specified index and contains the specified number of elements. (Inherited from List&lt;Of (Emitter)&gt;.)</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection...::FindLastIndex Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindLastIndex(Predicate(Of (T)&gt;)())</td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the entire List(Of (T)&gt;). (Inherited from List(Of (Emitter)&gt;).)</td>
</tr>
<tr>
<td>FindLastIndex(Int32, Predicate(Of (T)&gt;)())</td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the List(Of (T)&gt;). that extends from the first element to the specified index. (Inherited from List(Of (Emitter)&gt;).)</td>
</tr>
<tr>
<td>FindLastIndex(Int32, Int32, Predicate(Of (T)&gt;)())</td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the List(Of (T)&gt; that contains the specified number of elements and ends at the specified index. (Inherited from List(Of (Emitter)&gt;).)</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection..::.IndexOf Method

EmitterCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexOf(T)</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire List&lt;Of T&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List&lt;Of Emitter&gt;).</td>
</tr>
<tr>
<td>IndexOf(T, Int32)</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the List&lt;Of T&gt; that extends from the specified index to the last element.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List&lt;Of Emitter&gt;).</td>
</tr>
<tr>
<td>IndexOf(T, Int32, Int32)</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the List&lt;Of T&gt; that starts at the specified index and contains the specified number of elements.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List&lt;Of Emitter&gt;).</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection..:::LastIndexOf Method

EmitterCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| LastIndexOf(T) | Searches for the specified object and returns the zero-based index of the last occurrence within the entire List(Of (T)>).  
(Inherited from List(Of (Emitter)>).)|
| LastIndexOf(T, Int32) | Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the List(Of (T)>) that extends from the first element to the specified index.  
(Inherited from List(Of (Emitter)>).) |
| LastIndexOf(T, Int32, Int32) | Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the List(Of (T)>) that contains the specified number of elements and ends at the specified index.  
(Inherited from List(Of (Emitter)>).) |
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection::Reverse Method

EmitterCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse()()</td>
<td>Reverses the order of the elements in the entire List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
<tr>
<td>Reverse(Int32, Int32)</td>
<td>Reverses the order of the elements in the specified range. (Inherited from List&lt;Of &lt;(Emitter)&gt;&gt;.)</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection...::Sort Method

EmitterCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sort()</strong></td>
<td>Sorts the elements in the entire <code>List(Of (T)&gt;)</code> using the default comparer. (Inherited from <code>List(Of (Emitter)&gt;</code>).)</td>
</tr>
<tr>
<td><strong>Sort(IComparer(Of (T)&gt;])</strong></td>
<td>Sorts the elements in the entire <code>List(Of (T)&gt;)</code> using the specified comparer. (Inherited from <code>List(Of (Emitter)&gt;</code>).)</td>
</tr>
<tr>
<td><strong>Sort(Comparison(Of (T)&gt;)</strong></td>
<td>Sorts the elements in the entire <code>List(Of (T)&gt;)</code> using the specified <code>Comparison(Of (T)&gt;</code>). (Inherited from <code>List(Of (Emitter)&gt;</code>).)</td>
</tr>
<tr>
<td><strong>Sort(Int32, Int32, IComparer(Of (T)&gt;))</strong></td>
<td>Sorts the elements in a range of elements in <code>List(Of (T)&gt;)</code> using the specified comparer. (Inherited from <code>List(Of (Emitter)&gt;</code>).)</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `EmitterCollection` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from <code>List&lt;(Of </code>&lt;Emitter<code>)&gt;</code>).</td>
</tr>
<tr>
<td>Count</td>
<td><code>List&lt;(Of </code>&lt;T<code>)&gt;</code>. (Inherited from <code>List&lt;(Of </code>&lt;Emitter<code>)&gt;</code>).</td>
</tr>
<tr>
<td>Item</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCollection..::.Item Property

EmitterCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Int32" alt="Item" />)</td>
<td>Gets or sets the element at the specified index. (Inherited from <code>List&lt;Emitter&gt;</code>).</td>
</tr>
<tr>
<td><img src="String" alt="Item" />)</td>
<td>Gets the element with the specified name.</td>
</tr>
</tbody>
</table>
See Also

EmitterCollection Class
EmitterCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the element with the specified name.

**Namespace:** [ProjectMercury.Emitters](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Emitter Item[
    string name
] { get; }

Visual Basic (Declaration)

Public ReadOnly Property Item ( _
    name As String _
) As Emitter

Visual C++

public:
    property Emitter^ Item[String^ name] { 
        Emitter^ get (String^ name);
    }

Parameters

name

    Type: System:::String
    The name of the Emitter to fetch.

Return Value

The first Emitter whose name matches the specified name.
See Also

EmitterCollection Class
Item Overload
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
[This is preliminary documentation and is subject to change.]

Contains extension methods for the Emitter class which aid backwards compatibility with previous releases of the engine.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

**C#**

```csharp
public static class EmitterCompatibilityExtensions
```

**Visual Basic (Declaration)**

```vbnet
<ExtensionAttribute> _
Public NotInheritable Class EmitterCompatibilityExtensions
```

**Visual C++**

```cpp
[ExtensionAttribute]
public ref class EmitterCompatibilityExtensions abstract sealed
```
Inheritance Hierarchy

System::Object
ProjectMercury.Emitters::EmitterCompatibilityExtensions
See Also

EmitterCompatibilityExtensions Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
EmitterCompatibilityExtensions Members

The **EmitterCompatibilityExtensions** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Obsolete. Initializes the Emitter.</td>
</tr>
<tr>
<td>Update</td>
<td>Obsolete. Updates the Emitter.</td>
</tr>
</tbody>
</table>
See Also

EmitterCompatibilityExtensions Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `EmitterCompatibilityExtensions` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Obsolete.</td>
</tr>
<tr>
<td></td>
<td>Initializes the Emitter.</td>
</tr>
<tr>
<td>Update</td>
<td>Obsolete.</td>
</tr>
<tr>
<td></td>
<td>Updates the Emitter.</td>
</tr>
</tbody>
</table>


See Also

EmitterCompatibilityExtensions Class  
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes the Emitter.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
[ClergetAttributes("Use 'Initialise' method instead.", false)]
public static void Initialize(
    this Emitter emitter)
```

### Visual Basic (Declaration)

```vbnet
<ObsoleteAttribute("Use 'Initialise' method instead.", False)>
<ExtensionAttribute>
Public Shared Sub Initialize (_
    emitter As Emitter _
)
```

### Visual C++

```cpp
[ClergetAttributes(L"Use 'Initialise' method instead.", false)]
[ExtensionAttribute]
public:
static void Initialize(_
    Emitter^ emitter)
```

### Parameters

**emitter**

Type: `ProjectMercury.Emitters..::.Emitter`

[Missing `<param name="emitter"/>` documentation for
Remarks

The 'Initialize' method was renamed to 'Initialise' in revision 49307 (21st April 2009).
See Also

EmitterCompatibilityExtensions Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Updates the Emitter.

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

[ObsoleteAttribute("Use Update(deltaSeconds) method instead.", false]
public static void Update(
    this Emitter emitter,
    float totalSeconds,
    float deltaSeconds
)

Visual Basic (Declaration)

<ObsoleteAttribute("Use Update(deltaSeconds) method instead.", False
<ExtensionAttribute>
Public Shared Sub Update ( _
    emitter As Emitter, _
    totalSeconds As Single, _
    deltaSeconds As Single _
)

Visual C++

[ObsoleteAttribute(L"Use Update(deltaSeconds) method instead.", false
[ExtensionAttribute]
public:
    static void Update(
        Emitter^ emitter,
        float totalSeconds,
        float deltaSeconds
    )

Parameters

emitter
    Type: ProjectMercury.Emitters..::.Emitter
    The extended instance.

totalSeconds
    Type: System..::.Single
Total game time in whole and fractional seconds.

deltaSeconds
  Type: System::Single
  Elapsed game time in whole and fractional seconds.
See Also

EmitterCompatibilityExtensions Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines an Emitter which releases Particles at a random point along a line.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class LineEmitter : Emitter

Visual Basic (Declaration)

Public Class LineEmitter
    Inherits Emitter

Visual C++

public ref class LineEmitter : public Emitter
Inheritance Hierarchy

System:::Object

ProjectMercury.Emitters:::Emitter

ProjectMercury.Emitters:::LineEmitter
See Also

- LineEmitter Members
- ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `LineEmitter` type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LineEmitter</td>
<td>Initializes a new instance of the <a href="#">LineEmitter</a> class</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter. (Overrides Emitter:::DeepCopy()().)</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>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overrides Emitter:::GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>

Creates a shallow copy of the current Object.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MemberwiseClone</code></td>
<td>(Inherited from <code>Object</code>.)</td>
<td></td>
</tr>
<tr>
<td><code>OnNameChanged</code></td>
<td>Raises the NameChanged event.</td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><code>OnParticleReleased</code></td>
<td>Raises the ParticleReleased event.</td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><code>OnParticleRetired</code></td>
<td>Raises the ParticleRetired event.</td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><code>Terminate</code></td>
<td>Terminates the emitter immediately.</td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Trigger</code></td>
<td>Overloaded.</td>
<td></td>
</tr>
<tr>
<td><code>Update</code></td>
<td>Updates the Emitter and all Particles within.</td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlendMode</td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>EmitBothWays</td>
<td>If true, will emit particles both ways. Only work when Rectilinear is enabled.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from Emitter.)</td>
</tr>
<tr>
<td>MinimumTriggerPeriod</td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored.  (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Modifiers</td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Particles</td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleTexture</td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleTextureAssetName</td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Rectilinear</td>
<td>If true, will emit particles perpendicular to the angle of the line.</td>
</tr>
<tr>
<td>ReleaseColour</td>
<td>Gets or sets the colour of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Gets or sets the initial impulse applied to                                                                ystals to load in the LoadContent method. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
- **ReleaseImpulse**:Particles as they are released. (Inherited from Emitter.)
- **ReleaseOpacity**:Gets or sets the opacity of released Particles. (Inherited from Emitter.)
- **ReleaseRotation**:Gets or sets the rotation of released Particles. (Inherited from Emitter.)
- **ReleaseScale**:Gets or sets the scale of released particles. (Inherited from Emitter.)
- **ReleaseSpeed**:Gets or sets the speed at which Particles travel when they are released. (Inherited from Emitter.)
- **TriggerOffset**:The Emitters trigger offset in relation to the ParticleEffect. (Inherited from Emitter.)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ActiveParticlesCount | Gets the number of Particles which are currently active.  
(Inherited from Emitter.) |
| Angle             | Gets or sets the rotation of the line around its middle point.  
Gets or sets the number of Particles which are available to the Emitter.  
(Inherited from Emitter.) |
| Budget            | Gets or sets the number of Particles which are available to the Emitter.  
(Inherited from Emitter.) |
| initialised       | True if the Emitter object has been initialised, else false.  
(Inherited from Emitter.) |
| Length            | Gets or sets the length of the line.                                           |
| Name              | Gets or sets the name of the Emitter.  
(Inherited from Emitter.) |
| ReleaseQuantity   | Gets or sets the number of Particles which will be released on each trigger.  
(Inherited from Emitter.) |
| Term              | Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.  
(Inherited from Emitter.) |
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
LineEmitter Constructor

Initializes a new instance of the LineEmitter class

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public LineEmitter()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
LineEmitter()
See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `LineEmitter` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>EmitBothWays</strong></td>
<td>If true, will emit particles both ways. Only work when Rectilinear is enabled.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Rectilinear</strong></td>
<td>If true, will emit particles perpendicular to the angle of the line.</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td></td>
<td>Gets or sets the initial impulse applied to</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReleaseImpulse</td>
<td>Particles as they are released.</td>
</tr>
<tr>
<td>ReleaseOpacity</td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td>ReleaseRotation</td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td>ReleaseScale</td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td>ReleaseSpeed</td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td>TriggerOffset</td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
If true, will emit particles both ways. Only work when Rectilinear is enabled.

**Namespace:** ProjectMercury.Emitters
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public bool EmitBothWays

Visual Basic (Declaration)

Public EmitBothWays As Boolean

Visual C++

public:
    bool EmitBothWays
See Also

LineEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
If true, will emit particles perpendicular to the angle of the line.

**Namespace:**  ProjectMercury.Emitters

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public bool Rectilinear

Visual Basic (Declaration)

Public Rectilinear As Boolean

Visual C++

public:
bool Rectilinear
See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `LineEmitter` type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Methods</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter::&lt;tt&gt; DeepCopy()()&lt;/tt&gt;.)</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</td>
</tr>
<tr>
<td></td>
<td>the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and</td>
</tr>
<tr>
<td></td>
<td>is currently 'cooling down'.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is released.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter::&lt;tt&gt; GenerateOffsetAndForce(Vector2%, Vector2%)&lt;/tt&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</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>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Inherited from <strong>Object</strong>.</td>
</tr>
<tr>
<td>OnNameChanged</td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td>OnParticleReleased</td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td>OnParticleRetired</td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td>Terminate</td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <strong>String</strong> that represents the current</td>
</tr>
<tr>
<td>Trigger</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by <a href="#">EmitterCompatibilityExtensions</a>).</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by <a href="#">EmitterCompatibilityExtensions</a>).</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an uninitialised deep copy of the Emitter.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Emitter DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Emitter

Visual C++

public:
virtual Emitter^ DeepCopy() override

Return Value

A deep copy of the Emitter.
See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
protected override void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)
```

#### Visual Basic (Declaration)

```vbnet
Protected Overrides Sub GenerateOffsetAndForce (_
    <OutAttribute> ByRef offset As Vector2, _
    <OutAttribute> ByRef force As Vector2 _
)
```

#### Visual C++

```cpp
protected:
virtual void GenerateOffsetAndForce(
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
) override
```

### Parameters

**offset**
- Type: Vector2 %
- The offset of the Particle from the trigger location.

**force**
- Type: Vector2 %
- A unit vector defining the initial force of the Particle.
See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference
LineEmitter...::Initialise Method

LineEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialise()()</td>
<td>Initialises the Emitter.</td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Initialise(Int32, Single)</td>
<td>Initialises the Emitter.</td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
LineEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
LineEmitter....: Trigger Method

LineEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
LineEmitter Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `LineEmitter` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Angle</td>
<td>Gets or sets the rotation of the line around its middle point.</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Initialised</td>
<td>True if the Emitter object has been initialised, else false. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Length</td>
<td>Gets or sets the length of the line.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the rotation of the line around its middle point.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Angle { get; set; }

Visual Basic (Declaration)

Public Property Angle As Single

Visual C++

public:
property float Angle {
    float get ();
    void set (float value);
}

See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the length of the line.

**Namespace:** [ProjectMercury.Emitters](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

public float Length { get; set; }

**Visual Basic (Declaration)**

Public Property Length As Single

**Visual C++**

public:
property float Length {
    float get ();
    void set (float value);
}

See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `LineEmitter` type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

LineEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines an Emitter which releases Particles based on a mask array, typically from an image.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class MaskEmitter : Emitter

Visual Basic (Declaration)

Public NotInheritable Class MaskEmitter _
    Inherits Emitter

Visual C++

public ref class MaskEmitter sealed : public Emitter
Inheritance Hierarchy

System:::Object
ProjectMercury.Emitters:::Emitter
    ProjectMercury.Emitters:::MaskEmitter
See Also

MaskEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `MaskEmitter` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaskEmitter</strong></td>
<td>Initializes a new instance of the MaskEmitter class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplyMaskTexture</td>
<td>Applies a mask texture to the MaskEmitter.</td>
</tr>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>Emitter::DeepCopy()</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum</td>
</tr>
<tr>
<td></td>
<td>trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is</td>
</tr>
<tr>
<td></td>
<td>released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>Emitter::LoadContent(ContentManager)</code>.)</td>
</tr>
</tbody>
</table>
- **MemberwiseClone**
  Creates a shallow copy of the current [Object](#).
  (Inherited from [Object](#).)

- **OnNameChanged**
  Raises the NameChanged event.
  (Inherited from [Emitter](#).)

- **OnParticleReleased**
  Raises the ParticleReleased event.
  (Inherited from [Emitter](#).)

- **OnParticleRetired**
  Raises the ParticleRetired event.
  (Inherited from [Emitter](#).)

- **Terminate**
  Terminates the emitter immediately.
  (Inherited from [Emitter](#).)

- **ToString**
  Returns a [String](#) that represents the current [Object](#).
  (Inherited from [Object](#).)

- **Trigger**
  Overloaded.

- **Update**
  Updates the Emitter and all Particles within.
  (Inherited from [Emitter](#).)
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the Emitter. (Defined by <a href="#">EmitterCompatibilityExtensions</a>.)</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter. (Defined by <a href="#">EmitterCompatibilityExtensions</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActiveParticlesCount</strong></td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td>Gets or sets the height.</td>
</tr>
<tr>
<td><strong>Initialised</strong></td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Mask</strong></td>
<td>Gets or sets the mask array.</td>
</tr>
<tr>
<td><strong>MaskTextureContentPath</strong></td>
<td>Gets or sets the content path to the mask texture.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseQuantity</strong></td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Term</strong></td>
<td>Gets or sets the length of time that released Particles will remain active,</td>
</tr>
<tr>
<td></td>
<td>in whole and fractional seconds.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Threshold</strong></td>
<td>Gets or sets the threshold value above which samples in the mask will be</td>
</tr>
<tr>
<td></td>
<td>used as release points.</td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td>Gets or sets the width.</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NameChanged</strong></td>
<td>Raised when the name of the Emitter has been changed. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleReleased</strong></td>
<td>Raised when a Particle is released by the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleRetired</strong></td>
<td>Raised when a Particle expires. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the **MaskEmitter** class

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public MaskEmitter()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
MaskEmitter()
```
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **MaskEmitter** type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from Emitters)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from Emitters)</td>
</tr>
</tbody>
</table>
**ReleaseRotation**

Gets or sets the rotation of released Particles.
(Inherited from [Emitter](#).)

**ReleaseScale**

Gets or sets the scale of released particles.
(Inherited from [Emitter](#).)

**ReleaseSpeed**

Gets or sets the speed at which Particles travel when they are released.
(Inherited from [Emitter](#).)

**TriggerOffset**

The Emitters trigger offset in relation to the ParticleEffect.
(Inherited from [Emitter](#).)
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `MaskEmitter` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ApplyMaskTexture</strong></td>
<td>Applies a mask texture to the MaskEmitter.</td>
</tr>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an uninitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter::: DeepCopy().)</td>
</tr>
<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 operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</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>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter::: LoadContent(ContentManager).)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>OnNameChanged</strong></td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td><strong>OnParticleReleased</strong></td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td><strong>OnParticleRetired</strong></td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td><strong>Terminate</strong></td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <strong>String</strong> that represents the current</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Applies a mask texture to the MaskEmitter.

**Namespace:** `ProjectMercury.Emitters`  
**Assembly:** `ProjectMercury` (in `ProjectMercury.dll`) Version: 3.1.0.0
Syntax

C#

public void ApplyMaskTexture(
    Texture2D maskTexture
)

Visual Basic (Declaration)

Public Sub ApplyMaskTexture ( _
    maskTexture As Texture2D _
)

Visual C++

public:
void ApplyMaskTexture(
    Texture2D^ maskTexture
)

Parameters

maskTexture
    Type: Texture2D
    A texture reference representing the mask.
See Also

MaskEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an uninitialised deep copy of the Emitter.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override Emitter DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Emitter
```

**Visual C++**

```cpp
public:
virtual Emitter^ DeepCopy() override
```

**Return Value**

A deep copy of the Emitter.
See Also

MaskEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
MaskEmitter::<.Initialise Method

MaskEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Initialise()</code></td>
<td>Initialises the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><code>Initialise(Int32, Single)</code></td>
<td>Initialises the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Emitter</code>.)</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
MaskEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Load resources required by the Emitter via a ContentManager.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

```csharp
public override void LoadContent(
    ContentManager content
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub LoadContent (_
    content As ContentManager _
)
```

Visual C++

```cpp
public:
virtual void LoadContent(
    ContentManager^ content
) override
```

Parameters

content
  Type: ContentManager
  The ContentManager used to load resources.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentLoadException</td>
<td>Thrown if the asset defined in the ParticleTextureAssetName property could not be loaded.</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
MaskEmitter...:::Trigger Method

MaskEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
MaskEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `MaskEmitter` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td>Height</td>
<td>Gets or sets the height.</td>
</tr>
<tr>
<td>Initialised</td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td>Mask</td>
<td>Gets or sets the mask array.</td>
</tr>
<tr>
<td>MaskTextureContentPath</td>
<td>Gets or sets the content path to the mask texture.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.</td>
</tr>
<tr>
<td>Threshold</td>
<td>Gets or sets the threshold value above which samples in the mask will be used as release points.</td>
</tr>
<tr>
<td>Width</td>
<td>Gets or sets the width.</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the height.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Height { get; set; }

Visual Basic (Declaration)

Public Property Height As Single

Visual C++

public:
property float Height {
    float get ();
    void set (float value);
}

Field Value

The height.
See Also

- MaskEmitter Class
- ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the mask array.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public byte[][] Mask { get; set; }

Visual Basic (Declaration)

Public Property Mask As Byte()()

Visual C++

public:
property array<array<unsigned char>^>^ Mask {
    array<array<unsigned char>^>^ get ();
    void set (array<array<unsigned char>^>^ value);
}

Field Value

The mask array.
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the content path to the mask texture.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public string MaskTextureContentPath { get; set; }

Visual Basic (Declaration)

Public Property MaskTextureContentPath As String

Visual C++

public:
property String^ MaskTextureContentPath {
String^ get ();
void set (String^ value);
}

Field Value

The mask texture content path.
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the threshold value above which samples in the mask will be used as release points.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Threshold { get; set; }

Visual Basic (Declaration)

Public Property Threshold As Single

Visual C++

public:
property float Threshold {
    float get ();
    void set (float value);
}

Field Value

The threshold value.
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the width.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
**Syntax**

**C#**

```csharp
public float Width { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Width As Single
```

**Visual C++**

```cpp
public:
property float Width {
float get ();
void set (float value);
}
```

**Field Value**

The width.
See Also

MaskEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The MaskEmitter type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>).</td>
</tr>
</tbody>
</table>
See Also

MaskEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines an event handler for a Particle related event.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

```csharp
public delegate void ParticleEventHandler(
    Emitter emitter,
    ref Particle particle
)
```

Visual Basic (Declaration)

```vbnet
Public Delegate Sub ParticleEventHandler ( _
    emitter As Emitter, _
    ByRef particle As Particle _
)
```

Visual C++

```cpp
public delegate void ParticleEventHandler(
    Emitter^ emitter,
    Particle% particle
)
```

Parameters

emitter
  Type: `ProjectMercury.Emitters..::.Emitter`
  The Emitter which raised the event.

particle
  Type: `ProjectMercury..::.Particle`
  The Particle which is the subject of the event.
See Also

ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Emits particles in the shape of a polygon defined with the Points property.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public class PolygonEmitter : Emitter
```

**Visual Basic (Declaration)**

```vbnet
Public Class PolygonEmitter
    Inherits Emitter
```

**Visual C++**

```cpp
public ref class PolygonEmitter : public Emitter
```
Inheritance Hierarchy

System::Object
   ProjectMercury.Emitters::Emitter
      ProjectMercury.Emitters::PolygonEmitter
See Also

PolygonEmitter Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `PolygonEmitter` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolygonEmitter</td>
<td>Initializes a new instance of the PolygonEmitter class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an uninitialised deep copy of the Emitter. (Overrides Emitter:::DeepCopy().)</td>
</tr>
<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>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overrides Emitter:::GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (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>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager. (Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>OnNameChanged</td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>OnParticleReleased</td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>OnParticleRetired</td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Terminate</td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</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>
<tr>
<td>Trigger</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by <a href="https://example.com/emittercompatibilityextensions">EmitterCompatibilityExtensions</a>.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by <a href="https://example.com/emittercompatibilityextensions">EmitterCompatibilityExtensions</a>.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Gets or sets a value indicating whether or not the polygon should be closed.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActiveParticlesCount</strong></td>
<td>Gets the number of Particles which are currently active.</td>
<td>Emitter</td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
<td>Emitter</td>
</tr>
<tr>
<td><strong>Initialised</strong></td>
<td>True if the Emitter object has been initialised, else false.</td>
<td>Emitter</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets or sets the name of the Emitter.</td>
<td>Emitter</td>
</tr>
<tr>
<td><strong>Origin</strong></td>
<td>Gets or sets the origin of the point collection.</td>
<td></td>
</tr>
<tr>
<td><strong>Points</strong></td>
<td>Polygon points.</td>
<td></td>
</tr>
<tr>
<td><strong>ReleaseQuantity</strong></td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
<td>Emitter</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td>Polygon rotation.</td>
<td></td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Polygon scale.</td>
<td></td>
</tr>
<tr>
<td><strong>Term</strong></td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.</td>
<td>Emitter</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>NameChanged</code></td>
<td>Raised when the name of the Emitter has been</td>
<td></td>
</tr>
<tr>
<td></td>
<td>changed. (Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><code>ParticleReleased</code></td>
<td>Raised when a Particle is released by the Emitter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><code>ParticleRetired</code></td>
<td>Raised when a Particle expires.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `PolygonEmitter` class.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public PolygonEmitter()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
PolygonEmitter()
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The PolygonEmitter type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Gets or sets a value indicating wether or not the polygon should be closed.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating wether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are relased. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.    (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets a value indicating whether or not the polygon should be closed.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public bool Close

**Visual Basic (Declaration)**

Public Close As Boolean

**Visual C++**

public:
bool Close
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `PolygonEmitter` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter. (Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an uninitialised deep copy of the Emitter. (Overrides <code>Emitter::DeepCopy()()</code>. )</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from <code>Emitter</code>.)</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overrides <code>Emitter::::GenerateOffsetAndForce(Vector2%, Vector2%)</code>. )</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager. (Inherited from <code>Emitter</code>.)</td>
</tr>
</tbody>
</table>

Creates a shallow copy of the current `Object`.
MemberwiseClone  (Inherited from Object.)

OnNameChanged  Raises the NameChanged event.  (Inherited from Emitter.)

OnParticleReleased  Raises the ParticleReleased event.  (Inherited from Emitter.)

OnParticleRetired  Raises the ParticleRetired event.  (Inherited from Emitter.)

Terminate  Terminates the emitter immediately.  (Inherited from Emitter.)

ToString  Returns a String that represents the current Object.  (Inherited from Object.)

Trigger  Overloaded.

Update  Updates the Emitter and all Particles within.  (Inherited from Emitter.)
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by <code>EmitterCompatibilityExtensions</code>.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by <code>EmitterCompatibilityExtensions</code>.)</td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an uninitialised deep copy of the Emitter.

Namespace:  ProjectMercury.Emitters
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Emitter DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Emitter

Visual C++

public:
virtual Emitter^ DeepCopy() override

Return Value

A deep copy of the Emitter.
See Also

PolygonEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
protected override void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)
```

**Visual Basic (Declaration)**

```vbnet
Protected Overrides Sub GenerateOffsetAndForce (_
    <OutAttribute> ByRef offset As Vector2, _
    <OutAttribute> ByRef force As Vector2 _
)
```

**Visual C++**

```c++
protected:
virtual void GenerateOffsetAndForce(
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
) override
```

**Parameters**

**offset**

Type: Vector2 %

The offset of the Particle from the trigger location.

**force**

Type: Vector2 %

A unit vector defining the initial force of the Particle.
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference
PolygonEmitter..::.Initialise Method

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Initialise()</code></td>
<td>Initialises the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><code>Initialise(Int32, Single)</code></td>
<td>Initialises the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
PolygonEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference
PolygonEmitter::Trigger Method

PolygonEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Vector2" alt="Trigger" /></td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
<tr>
<td><img src="Vector2%25" alt="Trigger" /></td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
PolygonEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PolygonEmitter** type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActiveParticlesCount</strong></td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><strong>Initialised</strong></td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><strong>Origin</strong></td>
<td>Gets or sets the origin of the point collection.</td>
</tr>
<tr>
<td><strong>Points</strong></td>
<td>Polygon points.</td>
</tr>
<tr>
<td><strong>ReleaseQuantity</strong></td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td>Polygon rotation.</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Polygon scale.</td>
</tr>
<tr>
<td><strong>Term</strong></td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds.</td>
</tr>
<tr>
<td>(Inherited from Emitter.)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
PolygonEmitter..::.Origin Property

Gets or sets the origin of the point collection.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public PolygonOrigin Origin { get; set; }

Visual Basic (Declaration)

Public Property Origin As PolygonOrigin

Visual C++

public:
property PolygonOrigin Origin {
    PolygonOrigin get ();
    void set (PolygonOrigin value);
}

Field Value

The origin.
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Polygon points.

**Namespace:**  [ProjectMercury.Emitters](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll)  Version: 3.1.0.0
Syntax

C#

public PolygonPointCollection Points { get; set; }

Visual Basic (Declaration)

Public Property Points As PolygonPointCollection

Visual C++

public:
property PolygonPointCollection^ Points {
    PolygonPointCollection^ get ();
    void set (PolygonPointCollection^ value);
}
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Polygon rotation.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Rotation { get; set; }

Visual Basic (Declaration)

Public Property Rotation As Single

Visual C++

public:
property float Rotation {
    float get ();
    void set (float value);
}
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Polygon scale.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Scale { get; set; }

Visual Basic (Declaration)

Public Property Scale As Single

Visual C++

public:
property float Scale {
    float get ();
    void set (float value);
}
See Also

PolygonEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The PolygonEmitter type exposes the following members.
### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
See Also

PolygonEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Enumerates the origin options for a polygon shape.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public enum PolygonOrigin
```

**Visual Basic (Declaration)**

```vbnet
Public Enumeration PolygonOrigin
```

**Visual C++**

```cpp
public enum class PolygonOrigin
```
### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>No origin is specified, the translation vector will not be set.</td>
</tr>
<tr>
<td>Center</td>
<td>The translation vector will be set to move the origin into the centre of the shape.</td>
</tr>
<tr>
<td>Origin</td>
<td>The translation vector will be set to move the origin to the first point in this shape.</td>
</tr>
</tbody>
</table>
See Also

ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Collection of points to generate a polygon.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public class PolygonPointCollection : List<Vector2>, IList, ICollection, IEnumerable
```

**Visual Basic (Declaration)**

```vbnet
Public Class PolygonPointCollection
    Inherits List(Of Vector2)
    Implements IList, ICollection, IEnumerable
```

**Visual C++**

```cpp
public ref class PolygonPointCollection : public List<Vector2>, IList, ICollection, IEnumerable
```
Remarks

By implementing the IList interface explicitly, we can effectively override certain methods of the base class without them being declared as virtual.
Inheritance Hierarchy

System..:::Object
System.Collections.Generic..:::List<(Of <(Vector2)>)>)
ProjectMercury.Emitters..:::PolygonPointCollection
See Also

PolygonPointCollection Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PolygonPointCollection** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolygonPointCollection</td>
<td>Initializes a new instance of the PolygonPointCollection class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Add        | Adds an object to the end of the `List<Of <(T)>>`.
| AddRange   | Adds the elements of the specified collection to the end of the `List<Of <(T)>>`.
| AsReadOnly | Returns a read-only `IList<Of <(T)>>` wrapper for the current collection. |
| BinarySearch | Overloaded. |
| Clear      | Removes all elements from the `List<Of <(T)>>`.
| Contains   | Determines whether an element is in the `List<Of <(T)>>`.
<p>| ConvertAll&lt;Of&lt;(T)&gt;&gt;(&lt;TOutput&gt;)&gt; | Converts the elements in the current <code>List&lt;Of &lt;(T)&gt;&gt;</code> to another type, and returns a list containing the converted elements. |
| CopyTo     | Overloaded. |
| Equals     | Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. |
| Exists     | Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. |
| Finalize   | Searches for an element that matches the conditions |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find</strong></td>
<td>defined by the specified predicate, and returns the first occurrence within the entire List(Of &lt;T&gt;).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>FindAll</strong></td>
<td>Retrieves all the elements that match the conditions defined by the specified predicate.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>FindIndex</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>FindLast</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the</td>
</tr>
<tr>
<td></td>
<td>last occurrence within the entire List(Of &lt;T&gt;).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>FindLastIndex</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ForEach</strong></td>
<td>Performs the specified action on each element of the List(Of &lt;T&gt;).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetRange</strong></td>
<td>Creates a shallow copy of a range of elements in the source List(Of &lt;T&gt;).</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</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>IndexOf</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Inserts an element into the List(Of &lt;T&gt;) at the specified index.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>InsertRange</strong></td>
<td>inserts the elements of a collection into the List(Of &lt;T&gt;) at the specified index.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List(Of (Vector2))).</td>
</tr>
<tr>
<td><strong>LastIndexOf</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></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td></td>
<td>Removes the first occurrence of a specific object</td>
</tr>
</tbody>
</table>
- **Remove** from the `List<Of <(T)>>`.
  (Inherited from `List<Of <(Vector2)>>`.)
  Removes the all the elements that match the conditions defined by the specified predicate.
  (Inherited from `List<Of <(Vector2)>>`.)

- **RemoveAll**
  (Inherited from `List<Of <(Vector2)>>`.)
  Removes the element at the specified index of the `List<Of <(T)>>`.
  (Inherited from `List<Of <(Vector2)>>`.)

- **RemoveAt**
  (Inherited from `List<Of <(Vector2)>>`.)
  Removes a range of elements from the `List<Of <(T)>>`.
  (Inherited from `List<Of <(Vector2)>>`.)

- **RemoveRange**
  (Inherited from `List<Of <(Vector2)>>`.)
  Overloaded.

- **Reverse**
  (Inherited from `List<Of <(Vector2)>>`.)
  Overloaded.

- **Sort**
  (Inherited from `List<Of <(Vector2)>>`.)
  Overloaded.

- **ToArray**
  (Inherited from `List<Of <(Vector2)>>`.)
  Copies the elements of the `List<Of <(T)>>` to a new array.

- **ToString**
  (Inherited from `List<Of <(Vector2)>>`.)
  Returns a `String` that represents the current `Object`.
  (Inherited from `Object`.)

- **TrimExcess**
  (Inherited from `List<Of <(Vector2)>>`.)
  Sets the capacity to the actual number of elements in the `List<Of <(T)>>`, if that number is less than a threshold value.
  (Inherited from `List<Of <(Vector2)>>`.)

- **TrueForAll**
  (Inherited from `List<Of <(Vector2)>>`.)
  Determines whether every element in the `List<Of <(T)>>` matches the conditions defined by the specified predicate.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TranslationOffset</strong></td>
<td>Gets or sets an offset vector for the shape.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from <code>List&lt;(Of (Vector2))&gt;</code>.)</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements actually contained in the <code>List&lt;(Of (T))&gt;</code>. (Inherited from <code>List&lt;(Of (Vector2))&gt;</code>.)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from <code>List&lt;(Of (Vector2))&gt;</code>.)</td>
</tr>
<tr>
<td>Origin</td>
<td>Gets or sets the origin of the shape defined by the points in the collection.</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `PolygonPointCollection` class

**Namespace:**  `ProjectMercury.Emitters`

**Assembly:**  `ProjectMercury` (in `ProjectMercury.dll`) Version: 3.1.0.0
Syntax

C#

public PolygonPointCollection()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
PolygonPointCollection()
See Also

PolygonPointCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PolygonPointCollection** type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TranslationOffset</td>
<td>Gets or sets an offset vector for the shape.</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets an offset vector for the shape.

**Namespace:**  [ProjectMercury.Emitters](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 TranslationOffset

Visual Basic (Declaration)

Public TranslationOffset As Vector2

Visual C++

public:
Vector2 TranslationOffset
See Also

PolygonPointCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PolygonPointCollection** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Adds an object to the end of the List(Of (T)&gt;). (Inherited from List(Of (Vector2)&gt;).) Adds the elements of the specified collection to the end of the List(Of (T)&gt;).</td>
</tr>
<tr>
<td><strong>AddRange</strong></td>
<td>Returns a read-only IList(Of (T)&gt;) wrapper for the current collection. (Inherited from List(Of (Vector2)&gt;).)</td>
</tr>
<tr>
<td><strong>AsReadOnly</strong></td>
<td>Returns a read-only IList(Of (T)&gt;) wrapper for the current collection. (Inherited from List(Of (Vector2)&gt;).)</td>
</tr>
<tr>
<td><strong>BinarySearch</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Clear</strong></td>
<td>Removes all elements from the List(Of (T)&gt;). (Inherited from List(Of (Vector2)&gt;).) Determines whether an element is in the List(Of (T)&gt;).</td>
</tr>
<tr>
<td><strong>Contains</strong></td>
<td>Converting the elements in the current List(Of (T)&gt;). (Inherited from List(Of (Vector2)&gt;).)</td>
</tr>
<tr>
<td><strong>ConvertAll(Of &lt;TOutput&gt;)</strong></td>
<td>Converts the elements in the current List(Of (T)&gt; to another type, and returns a list containing the converted elements. (Inherited from List(Of (Vector2)&gt;).)</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. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Determines whether the List(Of (T)&gt;) contains elements that match the conditions defined by the specified predicate. (Inherited from List(Of (Vector2)&gt;).)</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>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Find</td>
<td>defined by the specified predicate, and returns the first occurrence within the entire List&lt;Of &lt;T&gt;&gt;. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>FindAll</td>
<td>Retrieves all the elements that match the conditions defined by the specified predicate. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>FindIndex</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>FindLast</td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>FindLastIndex</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ForEach</td>
<td>Performs the specified action on each element of the List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetRange</td>
<td>Creates a shallow copy of a range of elements in the source List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an element into the List&lt;Of &lt;(T)&gt;&gt; at the specified index. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>InsertRange</td>
<td>Inserts the elements of a collection into the List&lt;Of &lt;(T)&gt;&gt; at the specified index. (Inherited from List&lt;Of &lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td>LastIndexOf</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td></td>
<td>Removes the first occurrence of a specific object</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes all the elements that match the conditions defined by the specified predicate.</td>
</tr>
<tr>
<td>RemoveAll</td>
<td>Removes the all the elements that match the conditions defined by the specified predicate.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the element at the specified index of the List.</td>
</tr>
<tr>
<td>RemoveRange</td>
<td>Removes a range of elements from the List.</td>
</tr>
<tr>
<td>Reverse</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Sort</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ToArray</td>
<td>Copies the elements of the List to a new array.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td>TrimExcess</td>
<td>Sets the capacity to the actual number of elements in the List, if that number is less than a threshold value.</td>
</tr>
<tr>
<td>TrueForAll</td>
<td>Determines whether every element in the List matches the conditions defined by the specified predicate.</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members

Project Mercury API Reference
PolygonPointCollection::BinarySearch Method

PolygonPointCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BinarySearch(T)</strong></td>
<td>Searches the entire sorted <code>List&lt;Of &lt;(T)&gt;&gt;</code> for an element using the default comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;Of &lt;(Vector2)&gt;&gt;</code>).</td>
</tr>
<tr>
<td><strong>BinarySearch(T, IComparer&lt;Of &lt;(T)&gt;&gt;)</strong></td>
<td>Searches the entire sorted <code>List&lt;Of &lt;(T)&gt;&gt;</code> for an element using the specified comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;Of &lt;(Vector2)&gt;&gt;</code>).</td>
</tr>
<tr>
<td><strong>BinarySearch(Int32, Int32, T, IComparer&lt;Of &lt;(T)&gt;&gt;)</strong></td>
<td>Searches a range of elements in the sorted <code>List&lt;Of &lt;(T)&gt;&gt;</code> for an element using the specified comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;Of &lt;(Vector2)&gt;&gt;</code>).</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
PolygonPointCollection::CopyTo Method

PolygonPointCollection Class   See Also   Send Feedback

[This is preliminary documentation and is subject to change.]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyTo(array(&lt;T&gt;)[[]])</td>
<td>Copies the entire (\text{List&lt;(Of&lt;(&lt;T&gt;)&gt;)}) to a compatible one-dimensional array, starting at the beginning of the target array. (Inherited from (\text{List&lt;(Of&lt;(\langle\text{Vector2}\rangle)&gt;)}).)</td>
</tr>
<tr>
<td>CopyTo(array(&lt;T&gt;)[[]], Int32)</td>
<td>Copies the entire (\text{List&lt;(Of&lt;(&lt;T&gt;)&gt;)}) to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from (\text{List&lt;(Of&lt;(\langle\text{Vector2}\rangle)&gt;)}).)</td>
</tr>
<tr>
<td>CopyTo(Int32, array(&lt;T&gt;)[[]], Int32, Int32)</td>
<td>Copies a range of elements from the (\text{List&lt;(Of&lt;(&lt;T&gt;)&gt;)}) to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from (\text{List&lt;(Of&lt;(\langle\text{Vector2}\rangle)&gt;)}).)</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
PolygonPointCollection::FindIndex Method

PolygonPointCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FindIndex(Predicate&lt;Of&lt;T&gt;&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the entire List&lt;Of&lt;T&gt;&gt;. (Inherited from List&lt;Of&lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td><strong>FindIndex(Int32,Predicate&lt;Of&lt;T&gt;&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List&lt;Of&lt;T&gt;&gt; that extends from the specified index to the last element. (Inherited from List&lt;Of&lt;(Vector2)&gt;&gt;.)</td>
</tr>
<tr>
<td><strong>FindIndex(Int32,Int32,Predicate&lt;Of&lt;T&gt;&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List&lt;Of&lt;T&gt;&gt; that starts at the specified index and contains the specified number of elements. (Inherited from List&lt;Of&lt;(Vector2)&gt;&gt;.)</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>FindLastIndex(Predicate&lt;Of (T)&gt;)</code></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the entire <code>List&lt;Of (T)&gt;)</code>. (Inherited from <code>List&lt;Of (Vector2)&gt;)</code>).</td>
</tr>
<tr>
<td><code>FindLastIndex(Int32, Predicate&lt;Of (T)&gt;)</code></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the <code>List&lt;Of (T)&gt;)</code> that extends from the first element to the specified index. (Inherited from <code>List&lt;Of (Vector2)&gt;)</code>).</td>
</tr>
<tr>
<td><code>FindLastIndex(Int32, Int32, Predicate&lt;Of (T)&gt;)</code></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the <code>List&lt;Of (T)&gt;)</code> that contains the specified number of elements and ends at the specified index. (Inherited from <code>List&lt;Of (Vector2)&gt;)</code>).</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IndexOf(T)</code></td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire `List(Of &lt;T&gt;).'</td>
</tr>
<tr>
<td></td>
<td>(Inherited from `List&lt;(Of &lt;(Vector2)&gt;)&gt;).)</td>
</tr>
<tr>
<td><code>IndexOf(T, Int32)</code></td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the <code>List&lt;(Of &lt;(T)&gt;)&gt; that extends from the specified index to the last element. (Inherited from </code>List&lt;(Of &lt;(Vector2)&gt;)&gt;).)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from `List&lt;(Of &lt;(Vector2)&gt;)&gt;).)</td>
</tr>
<tr>
<td><code>IndexOf(T, Int32, Int32)</code></td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the <code>List&lt;(Of &lt;(T)&gt;)&gt; that starts at the specified index and contains the specified number of elements. (Inherited from </code>List&lt;(Of &lt;(Vector2)&gt;)&gt;).)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from `List&lt;(Of &lt;(Vector2)&gt;)&gt;).)</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
PolygonPointCollection::LastIndexOf Method

PolygonPointCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LastIndexOf(T)</strong></td>
<td>Searches for the specified object and returns the zero-based index of the last occurrence within the entire List&lt;(Of &lt;(T)&gt;&gt;). (Inherited from List&lt;(Of &lt;(Vector2)&gt;)).</td>
</tr>
<tr>
<td><strong>LastIndexOf(T, Int32)</strong></td>
<td>Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the List&lt;(Of &lt;(T)&gt;)&gt; that extends from the first element to the specified index. (Inherited from List&lt;(Of &lt;(Vector2)&gt;)).</td>
</tr>
<tr>
<td><strong>LastIndexOf(T, Int32, Int32)</strong></td>
<td>Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the List&lt;(Of &lt;(T)&gt;)&gt; that contains the specified number of elements and ends at the specified index. (Inherited from List&lt;(Of &lt;(Vector2)&gt;)).</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class  
PolygonPointCollection Members  
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members

Project Mercury API Reference

PolygonPointCollection...::Reverse Method

PolygonPointCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse()</td>
<td>Reverses the order of the elements in the entire List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Vector2)&gt;).)</td>
</tr>
<tr>
<td>Reverse(Int32, Int32)</td>
<td>Reverses the order of the elements in the specified range.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from List&lt;Of &lt;(Vector2)&gt;).)</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
PolygonPointCollection::Sort Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort()()</td>
<td>Sorts the elements in the entire <code>List(Of &lt;(T)&gt;)</code> using the default comparer. (Inherited from <code>List&lt;(Of &lt;(Vector2)&gt;)&gt;.</code>)</td>
</tr>
<tr>
<td>Sort(IComparer&lt;(Of &lt;(T)&gt;)&gt;)</td>
<td>Sorts the elements in the entire <code>List&lt;(Of &lt;(T)&gt;)&gt;</code> using the specified comparer. (Inherited from <code>List&lt;(Of &lt;(Vector2)&gt;)&gt;.</code>)</td>
</tr>
<tr>
<td>Sort(Comparison&lt;(Of &lt;(T)&gt;)&gt;)</td>
<td>Sorts the elements in the entire <code>List&lt;(Of &lt;(T)&gt;)&gt;</code> using the specified <code>Comparison&lt;(Of &lt;(T)&gt;)&gt;.</code> (Inherited from <code>List&lt;(Of &lt;(Vector2)&gt;)&gt;.</code>)</td>
</tr>
<tr>
<td>Sort(Int32, Int32, IComparer&lt;(Of &lt;(T)&gt;)&gt;)</td>
<td>Sorts the elements in a range of elements in <code>List&lt;(Of &lt;(T)&gt;)&gt;</code> using the specified comparer. (Inherited from <code>List&lt;(Of &lt;(Vector2)&gt;)&gt;.</code>)</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
PolygonPointCollection Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PolygonPointCollection** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from List(Of (Vector2)&gt;).)</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements actually contained in the List(Of (T)&gt;). (Inherited from List(Of (Vector2)&gt;).)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from List(Of (Vector2)&gt;).)</td>
</tr>
<tr>
<td>Origin</td>
<td>Gets or sets the origin of the shape defined by the points in the collection.</td>
</tr>
</tbody>
</table>
See Also

PolygonPointCollection Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the origin of the shape defined by the points in the collection.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public PolygonOrigin Origin { get; set; }
```

Visual Basic (Declaration)

```
Public Property Origin As PolygonOrigin
```

Visual C++

```cpp
public:
property PolygonOrigin Origin {
PolygonOrigin get ();
void set (PolygonOrigin value);
}
```
See Also

PolygonPointCollection Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines an Emitter which releases particles in a rectangle shape.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class RectEmitter : Emitter

Visual Basic (Declaration)

Public Class RectEmitter
    Inherits Emitter

Visual C++

public ref class RectEmitter : public Emitter
Inheritance Hierarchy

System..:::Object
  ProjectMercury.Emitters..:::Emitter
    ProjectMercury.Emitters..:::RectEmitter
See Also

RectEmitter Members
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RectEmitter` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RectEmitter</td>
<td>Initializes a new instance of the RectEmitter class</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an uninitialised deep copy of the Emitter. (Overrides Emitter::: DeepCopy().)</td>
</tr>
<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>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overrides Emitter::: GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (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>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager. (Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
</tbody>
</table>
**MemberwiseClone**  
(Inherited from **Object**.)

**OnNameChanged**  
Raises the NameChanged event.  
(Inherited from **Emitter**.)

**OnParticleReleased**  
 Raises the ParticleReleased event.  
(Inherited from **Emitter**.)

**OnParticleRetired**  
 Raises the ParticleRetired event.  
(Inherited from **Emitter**.)

**Terminate**  
Terminates the emitter immediately.  
(Inherited from **Emitter**.)

**ToString**  
Returns a **String** that represents the current **Object**.  
(Inherited from **Object**.)

**Trigger**  
Overloaded.

**Update**  
Updates the Emitter and all Particles within.  
(Inherited from **Emitter**.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by <em>EmitterCompatibilityExtensions</em>.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by <em>EmitterCompatibilityExtensions</em>.)</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>Frame</strong></td>
<td>True if the Particles should be released only from the edge of the rectangle, else false.</td>
</tr>
<tr>
<td></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from <a href="#">Emitter</a>).</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ActiveParticlesCount</strong></td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td>Gets or sets the height of the rectangle..</td>
</tr>
<tr>
<td></td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Initialised</strong></td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseQuantity</strong></td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td>Gets or sets the rotation of the rectangle.</td>
</tr>
<tr>
<td></td>
<td>Gets or sets the length of time that released Particles will remain active, in</td>
</tr>
<tr>
<td></td>
<td>whole and fractional seconds.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Term</strong></td>
<td>Gets or sets the width of the rectangle.</td>
</tr>
<tr>
<td></td>
<td>Gets or sets the number of Particles which are currently active.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>

- [Emitter](#): Class that emits particles.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NameChanged</strong></td>
<td>Raised when the name of the Emitter has been changed. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ParticleReleased</strong></td>
<td>Raised when a Particle is released by the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>ParticleRetired</strong></td>
<td>Raised when a Particle expires. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `RectEmitter` class

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RectEmitter()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RectEmitter()
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RectEmitter Fields

RectEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

The RectEmitter type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled</td>
</tr>
<tr>
<td></td>
<td>(can be triggered). (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Frame</strong></td>
<td>True if the Particles should be released only from the edge of the</td>
</tr>
<tr>
<td></td>
<td>rectangle, else false. Defines the minimum amount of time between triggers</td>
</tr>
<tr>
<td></td>
<td>for the Emitter, expressed in whole and fractional seconds. Triggers which</td>
</tr>
<tr>
<td></td>
<td>occur during this period will be ignored.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter,</td>
</tr>
<tr>
<td></td>
<td>expressed in whole and fractional seconds. Triggers which occur during</td>
</tr>
<tr>
<td></td>
<td>this period will be ignored.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td><strong>TriggerOffset</strong></td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
True if the Particles should be released only from the edge of the rectangle, else false.

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public bool Frame

Visual Basic (Declaration)

Public Frame As Boolean

Visual C++

public:
bool Frame
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RectEmitter** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</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</td>
</tr>
<tr>
<td></td>
<td>operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a</td>
</tr>
<tr>
<td></td>
<td>minimum trigger period and is currently 'cooling down'.</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is</td>
</tr>
<tr>
<td></td>
<td>released.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
</tbody>
</table>
MemberwiseClone  (Inherited from Object.)

OnNameChanged  Raises the NameChanged event.  (Inherited from Emitter.)

OnParticleReleased  Raises the ParticleReleased event.  (Inherited from Emitter.)

OnParticleRetired  Raises the ParticleRetired event.  (Inherited from Emitter.)

Terminate  Terminates the emitter immediately.  (Inherited from Emitter.)

ToString  Returns a String that represents the current Object.  (Inherited from Object.)

Trigger  Overloaded.

Update  Updates the Emitter and all Particles within.  (Inherited from Emitter.)
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter. (Defined by EmitterCompatibilityExtensions.)</td>
</tr>
</tbody>
</table>
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RectEmitter..::.DeepCopy Method

Returns an uninitialised deep copy of the Emitter.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Emitter DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Emitter

Visual C++

public:
virtual Emitter^ DeepCopy() override

Return Value

A deep copy of the Emitter.
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected override void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)

Visual Basic (Declaration)

Protected Overrides Sub GenerateOffsetAndForce (_
    <OutAttribute> ByRef offset As Vector2, _
    <OutAttribute> ByRef force As Vector2 _
)

Visual C++

protected:
virtual void GenerateOffsetAndForce(
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
) override

Parameters

offset
    Type: Vector2 %
    The offset of the Particle from the trigger location.

force
    Type: Vector2 %
    A unit vector defining the initial force of the Particle.
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RectEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialise()()</td>
<td>Initialises the Emitter.</td>
</tr>
<tr>
<td>Initialise(Int32, Single)</td>
<td>Initialises the Emitter.</td>
</tr>
</tbody>
</table>

(Inherited from Emitter.)
See Also

RectEmitter Class
RectEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
RectEmitter Class

See Also

Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

RectEmitter Class
RectEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RectEmitter` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Height</td>
<td>Gets or sets the height of the rectangle.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Initialised</td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Rotation</td>
<td>Gets or sets the rotation of the rectangle.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active,</td>
</tr>
<tr>
<td></td>
<td>in whole and fractional seconds.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>Width</td>
<td>Gets or sets the width of the rectangle.</td>
</tr>
</tbody>
</table>
See Also

RectEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(ftp://mpe.codeplex.com/People/ProjectPeople.aspx)
RectEmitter.Height Property

Gets or sets the height of the rectangle.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Height { get; set; }

Visual Basic (Declaration)

Public Property Height As Single

Visual C++

public:
    property float Height {
        float get ();
        void set (float value);
    }

Field Value

The height of the rectangle.
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the rotation of the rectangle.

**Namespace:**  [ProjectMercury.Emitters](#)
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public float Rotation { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Rotation As Single
```

**Visual C++**

```c++
public:
property float Rotation {
    float get ();
    void set (float value);
}
```

**Field Value**

The rotation of the rectangle measured in radians.
See Also

RectEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the width of the rectangle.

**Namespace:** ProjectMercury.Emitters

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public float Width { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Width As Single
```

**Visual C++**

```csharp
public:
property float Width {
    float get ();
    void set (float value);
}
```

Field Value

The width of the rectangle.
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RectEmitter` type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
<td>(Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

RectEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
TextureEmitter Class

Obsolete.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

[ObsoleteAttribute("Replaced by MaskEmitter")]
public class TextureEmitter : Emitter

Visual Basic (Declaration)

<ObsoleteAttribute("Replaced by MaskEmitter")>
Public Class TextureEmitter _
    Inherits Emitter

Visual C++

[ObsoleteAttribute(L"Replaced by MaskEmitter")]
public ref class TextureEmitter : public Emitter
Inheritance Hierarchy

System..:::Object
ProjectMercury.Emitters..:::Emitter
   ProjectMercury.Emitters..:::TextureEmitter
See Also

TextureEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TextureEmitter` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextureEmitter</td>
<td>Initializes a new instance of the TextureEmitter class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyBaseFields</td>
<td>Copies the fields of the Emitter base class into the specified Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns an uninitialised deep copy of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Emitter:: DeepCopy().)</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. (Inherited from Object.)</td>
</tr>
<tr>
<td>ForceNextTrigger</td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>GenerateOffsetAndForce</td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overides Emitter:: GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</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>Initialise</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads resources required by the Emitter via a ContentManager.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>OnNameChanged</strong></td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td><strong>OnParticleReleased</strong></td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td><strong>OnParticleRetired</strong></td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td><strong>Terminate</strong></td>
<td>Terminates the emitter immediately.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
</tbody>
</table>
**Extension Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the Emitter. (Defined by <code>EmitterCompatibilityExtensions</code>.)</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Updates the Emitter. (Defined by <code>EmitterCompatibilityExtensions</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from <a href="#">Emitter</a>)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReleaseRotation</strong></td>
<td>Gets or sets the rotation of released Particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseScale</strong></td>
<td>Gets or sets the scale of released particles.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseSpeed</strong></td>
<td>Gets or sets the speed at which Particles travel when they are released.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Threshold</strong></td>
<td>Gets or sets the threshold over which pixels will trigger the release of particles.</td>
</tr>
<tr>
<td></td>
<td>The Emitters trigger offset in relation to the ParticleEffect.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveParticlesCount</td>
<td>Gets the number of Particles which are currently active.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ApplyPixelColours</td>
<td>Gets or sets a value indicating whether particles should assume the colour</td>
</tr>
<tr>
<td></td>
<td>of the underlying pixel in the texture.</td>
</tr>
<tr>
<td>Budget</td>
<td>Gets or sets the number of Particles which are available to the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Initialised</td>
<td>True if the Emitter object has been initialised, else false.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets or sets the name of the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>ReleaseQuantity</td>
<td>Gets or sets the number of Particles which will be released on each trigger.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Scale</td>
<td>Gets or sets the scale factor of the texture (in screen space).</td>
</tr>
<tr>
<td>Term</td>
<td>Gets or sets the length of time that released Particles will remain active,</td>
</tr>
<tr>
<td></td>
<td>in whole and fractional seconds.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td>Texture</td>
<td>Gets or sets the texture used to lookup particle release offsets.</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the \texttt{TextureEmitter} class.

\textbf{Namespace:} \texttt{ProjectMercury.Emitters}
\textbf{Assembly}: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public TextureEmitter()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
TextureEmitter()
```
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TextureEmitter` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BlendMode</strong></td>
<td>The blending mode to be used by Renderers when rendering this Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>Gets or sets a value indicating whether or not the Emitter is enabled (can be triggered). (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>MinimumTriggerPeriod</strong></td>
<td>Defines the minimum amount of time between triggers for the Emitter, expressed in whole and fractional seconds. Triggers which occur during this period will be ignored. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Modifiers</strong></td>
<td>Gets the collection of Modifiers which are acting upon the Emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td>Gets or sets the array of particles managed by the emitter. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTexture</strong></td>
<td>Gets or sets the Texture2D used to display the Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ParticleTextureAssetName</strong></td>
<td>Gets the asset name of a texture to load in the LoadContent method. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseColour</strong></td>
<td>Gets or sets the colour of released Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseImpulse</strong></td>
<td>Gets or sets the initial impulse applied to Particles as they are released. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
<tr>
<td><strong>ReleaseOpacity</strong></td>
<td>Gets or sets the opacity of released Particles. (Inherited from <a href="#">Emitter</a>.)</td>
</tr>
</tbody>
</table>
- **ReleaseRotation**: Gets or sets the rotation of released particles. (Inherited from *Emitter*.)
- **ReleaseScale**: Gets or sets the scale of released particles. (Inherited from *Emitter*.)
- **ReleaseSpeed**: Gets or sets the speed at which Particles travel when they are released. (Inherited from *Emitter*.)
- **Threshold**: Gets or sets the threshold over which pixels will trigger the release of particles. The Emitters trigger offset in relation to the ParticleEffect. (Inherited from *Emitter*.)
- **TriggerOffset**
See Also

(TextureEmitter Class
 ProjectMercury.Emitters Namespace)

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
TextureEmitter..::.Threshold Field

Gets or sets the threshold over which pixels will trigger the release of particles.

Namespace: ProjectMercury.Emitters
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public float Threshold

Visual Basic (Declaration)
Public Threshold As Single

Visual C++
public:
float Threshold
See Also

TextureEmitter Class  
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TextureEmitter` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyBaseFields</strong></td>
<td>Copies the fields of the Emitter base class into the specified Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns an uninitialised deep copy of the Emitter. (Overrides Emitter::: DeepCopy()().)</td>
</tr>
<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>ForceNextTrigger</strong></td>
<td>Forces the Emitter to execute its next trigger, even if it has a minimum trigger period and is currently 'cooling down'. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>GenerateOffsetAndForce</strong></td>
<td>Generates an offset vector and force vector for a Particle when it is released. (Overides Emitter::: GenerateOffsetAndForce(Vector2%, Vector2%).)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (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>Initialise</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>LoadContent</strong></td>
<td>Loads resources required by the Emitter via a ContentManager. (Inherited from Emitter.)</td>
</tr>
<tr>
<td><strong>Creates a shallow copy of the current Object.</strong></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Raises the NameChanged event.</td>
</tr>
<tr>
<td><strong>OnNameChanged</strong></td>
<td>Raises the ParticleReleased event.</td>
</tr>
<tr>
<td><strong>OnParticleReleased</strong></td>
<td>Raises the ParticleRetired event.</td>
</tr>
<tr>
<td><strong>OnParticleRetired</strong></td>
<td>Terminate the emitter immediately.</td>
</tr>
<tr>
<td><strong>Terminate</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>Updates the Emitter and all Particles within.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Initializes the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Defined by <a href="#">EmitterCompatibilityExtensions</a>.)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates the Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Defined by <a href="#">EmitterCompatibilityExtensions</a>.)</td>
</tr>
</tbody>
</table>
See Also

TextureEmitter Class
ProjectMercury.Emiters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns an uninitialised deep copy of the Emitter.

**Namespace:** [ProjectMercury.Emitters](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Emitter DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Emitter

Visual C++

public:
virtual Emitter^ DeepCopy() override

Return Value

A deep copy of the Emitter.
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Generates an offset vector and force vector for a Particle when it is released.

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  Version: 3.1.0.0
Syntax

C#

protected override void GenerateOffsetAndForce(
    out Vector2 offset,
    out Vector2 force
)

Visual Basic (Declaration)

Protected Overrides Sub GenerateOffsetAndForce ( _
    <OutAttribute> ByRef offset As Vector2, _
    <OutAttribute> ByRef force As Vector2 _
)

Visual C++

protected:
virtual void GenerateOffsetAndForce(
    [OutAttribute] Vector2% offset,
    [OutAttribute] Vector2% force
) override

Parameters

offset

    Type: Vector2 %
    The offset of the Particle from the trigger location.

force

    Type: Vector2 %
    A unit vector defining the initial force of the Particle.
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Include Protected Members
Include Inherited Members

Project Mercury API Reference
TextureEmitter...:::Initialise Method

TextureEmitter Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialise()()</td>
<td>Initialises the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Initialise(Int32, Single)</td>
<td>Initialises the Emitter. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

TextureEmitter Class
TextureEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger(Vector2)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
<tr>
<td>Trigger(Vector2%)</td>
<td>Triggers the Emitter at the specified position... (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

TextureEmitter Class
TextureEmitter Members
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TextureEmitter` type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActiveParticlesCount</strong></td>
<td>Gets the number of Particles which are currently active. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ApplyPixelColours</strong></td>
<td>Gets or sets a value indicating whether particles should assume the colour of the underlying pixel in the texture.</td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td>Gets or sets the number of Particles which are available to the Emitter. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Initialised</strong></td>
<td>True if the Emitter object has been initialised, else false. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets or sets the name of the Emitter. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>ReleaseQuantity</strong></td>
<td>Gets or sets the number of Particles which will be released on each trigger. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Gets or sets the scale factor of the texture (in screen space).</td>
</tr>
<tr>
<td><strong>Term</strong></td>
<td>Gets or sets the length of time that released Particles will remain active, in whole and fractional seconds. (Inherited from <strong>Emitter</strong>.)</td>
</tr>
<tr>
<td><strong>Texture</strong></td>
<td>Gets or sets the texture used to lookup particle release offsets.</td>
</tr>
</tbody>
</table>
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets a value indicating whether particles should assume the colour of the underlying pixel in the texture

**Namespace:**  ProjectMercury.Emitters  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  Version: 3.1.0.0
Syntax

C#

public bool ApplyPixelColours { get; set; }

Visual Basic (Declaration)

Public Property ApplyPixelColours As Boolean

Visual C++

public:
property bool ApplyPixelColours {
    bool get ();
    void set (bool value);
}
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the scale factor of the texture (in screen space).

**Namespace:** ProjectMercury.Emitters  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Scale { get; set; }

Visual Basic (Declaration)

Public Property Scale As Single

Visual C++

public:
property float Scale {
    float get ();
    void set (float value);
}
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the texture used to lookup particle release offsets.

**Namespace:** [ProjectMercury.Emitters](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Texture2D Texture { get; set; }

Visual Basic (Declaration)

Public Property Texture As Texture2D

Visual C++

public:
property Texture2D^ Texture {
    Texture2D^ get ();
    void set (Texture2D^ value);
}

Field Value

The texture.
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TextureEmitter` type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameChanged</td>
<td>Raised when the name of the Emitter has been changed. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleReleased</td>
<td>Raised when a Particle is released by the Emitter. (Inherited from Emitter.)</td>
</tr>
<tr>
<td>ParticleRetired</td>
<td>Raised when a Particle expires. (Inherited from Emitter.)</td>
</tr>
</tbody>
</table>
See Also

TextureEmitter Class
ProjectMercury.Emitters Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference
ProjectMercury.Modifiers Namespace

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColourInterpolatorModifier</td>
<td>Defines a modifier which changes the colour of particles based on a linear interpolation over three values.</td>
</tr>
<tr>
<td>ColourMergeModifier</td>
<td>Defines a Modifier which merges the colour of particles towards a single colour over their lifetime. Works best when Particles are being released with random colours, where you require the particles to have a uniform colour at the end of their lifetime.</td>
</tr>
<tr>
<td>ColourModifier</td>
<td>Defines a Modifier which gradually changes the colour of a Particle over the course of its lifetime.</td>
</tr>
<tr>
<td>DampingModifier</td>
<td>Defines a Modifier which applies a damping force to a Particle over its lifetime.</td>
</tr>
<tr>
<td>HueShiftModifier</td>
<td>Defines a Modifier which adjusts the hue of a Particles colour over time.</td>
</tr>
<tr>
<td>LinearGravityModifier</td>
<td>Defines a Modifier that applies a constant force vector to Particles over their lifetime.</td>
</tr>
<tr>
<td>Modifier</td>
<td>Defines the base class for an object which modifies Particle values.</td>
</tr>
<tr>
<td>ModifierCollection</td>
<td>Defines a collection of Modifiers.</td>
</tr>
<tr>
<td>OpacityInterpolatorModifier</td>
<td>Defines a modifier which changes the opacity of particles based on a linear interpolation over three values.</td>
</tr>
<tr>
<td>OpacityModifier</td>
<td>Defines a Modifier which gradually changes the opacity of a Particle over its lifetime.</td>
</tr>
<tr>
<td>OpacityOscillator</td>
<td>Defines a Modifier which adjusts the opacity of a Particle over its lifetime.</td>
</tr>
<tr>
<td>Modifier</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PlatformModifier</td>
<td>Defines a Modifier which freezes a particle when it comes into contact with a bounding box.</td>
</tr>
<tr>
<td>RadialForceModifier</td>
<td>Defines a Modifier which applies a force to a Particle when it enters a circular area.</td>
</tr>
<tr>
<td>RadialGravityModifier</td>
<td>Defines a Modifier which pulls Particles towards it.</td>
</tr>
<tr>
<td>RectangleConstraintDeflector</td>
<td>Defines a Modifier which constrains &amp; deflects particles inside a rectangle.</td>
</tr>
<tr>
<td>RectangleForceModifier</td>
<td>Defines a Modifier which applies a force to a Particle when it enters a rectangular area.</td>
</tr>
<tr>
<td>RotationModifier</td>
<td>Defines a Modifier which alters the rotation of a Particle over its lifetime.</td>
</tr>
<tr>
<td>RotationRateModifier</td>
<td>Defines a modifier which changes the rotation rate of particles over their lifetime.</td>
</tr>
<tr>
<td>ScaleInterpolatorModifier</td>
<td>Defines a modifier which changes the scale of particles based on a linear interpolation over three values.</td>
</tr>
<tr>
<td>ScaleMergeModifier</td>
<td>Defines a Modifier which merges the scale of particles towards a single scale over their lifetime. Works best when Particles are being released with random scale, where you require the particles to have a uniform scale at the end of their lifetime.</td>
</tr>
<tr>
<td>ScaleModifier</td>
<td>Defines a Modifier which adjusts the scale of a Particle over its lifetime.</td>
</tr>
<tr>
<td>ScaleOscillator</td>
<td>Defines a Modifier which adjusts the scale of Particles based on a sine wave.</td>
</tr>
<tr>
<td>SineForceModifier</td>
<td>Defines a Modifier which applies a sine wave force to a Particle over the course of its lifetime.</td>
</tr>
</tbody>
</table>
**TrajectoryRotationModifier**  Defines a Modifier which adjusts the rotation of a Particle to follow its trajectory.

**VelocityClampModifier**  Defines a Modifier which limits the velocity of Particles to a specified value.

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a modifier which changes the colour of particles based on a linear interpolation over three values.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class ColourInterpolatorModifier : Modifier

Visual Basic (Declaration)

Public Class ColourInterpolatorModifier _
Inherits Modifier

Visual C++

public ref class ColourInterpolatorModifier : public Modifier
Inheritance Hierarchy

System..:::Object

ProjectMercury.Modifiers..:::Modifier

ProjectMercury.Modifiers..:::ColourInterpolatorModifier
See Also

ColourInterpolatorModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ColourInterpolatorModifier type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColourInterpolatorModifier</td>
<td>Initializes a new instance of the <a href="#">ColourInterpolatorModifier</a> class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified &lt;code&gt;Object&lt;/code&gt; is equal to the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an &lt;code&gt;Object&lt;/code&gt; to attempt to free resources and perform other cleanup operations before the &lt;code&gt;Object&lt;/code&gt; is reclaimed by garbage collection. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the &lt;code&gt;Type&lt;/code&gt; of the current instance. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a &lt;code&gt;String&lt;/code&gt; that represents the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinalColour</td>
<td>Gets or sets the final colour.</td>
</tr>
<tr>
<td>InitialColour</td>
<td>Gets or sets the initial colour.</td>
</tr>
<tr>
<td>MiddleColour</td>
<td>Gets or sets the middle colour.</td>
</tr>
<tr>
<td>MiddlePosition</td>
<td>Gets or sets the middle colour position.</td>
</tr>
</tbody>
</table>
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the [ColourInterpolatorModifier](#) class

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public ColourInterpolatorModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ColourInterpolatorModifier()
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ColourInterpolatorModifier` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

### Return Value

See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

ColourInterpolatorModifier Class  
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
 ColourInterpolatorModifier Properties

[This is preliminary documentation and is subject to change.]

The ColourInterpolatorModifier type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinalColour</td>
<td>Gets or sets the final colour.</td>
</tr>
<tr>
<td>InitialColour</td>
<td>Gets or sets the initial colour.</td>
</tr>
<tr>
<td>MiddleColour</td>
<td>Gets or sets the middle colour.</td>
</tr>
<tr>
<td>MiddlePosition</td>
<td>Gets or sets the middle colour position.</td>
</tr>
</tbody>
</table>
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the final colour.

**Namespace:**  ProjectMercury.Modifiers

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector3 FinalColour { get; set; }

Visual Basic (Declaration)

Public Property FinalColour As Vector3

Visual C++

public:
property Vector3 FinalColour {
    Vector3 get ();
    void set (Vector3 value);
}

Field Value

The final colour.
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
InitialColour Property

Gets or sets the initial colour.

**Namespace:** [ProjectMercury.Modifiers](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector3 InitialColour { get; set; }

Visual Basic (Declaration)

Public Property InitialColour As Vector3

Visual C++

public:
    property Vector3 InitialColour {
        Vector3 get ();
        void set (Vector3 value);
    }

Field Value

The initial colour.
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the middle colour.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public Vector3 MiddleColour { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property MiddleColour As Vector3
```

Visual C++

```cpp
public:
    property Vector3 MiddleColour {
        Vector3 get ();
        void set (Vector3 value);
    }
```

Field Value

The middle colour.
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the middle colour position.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MiddlePosition { get; set; }

Visual Basic (Declaration)

Public Property MiddlePosition As Single

Visual C++

public:
property float MiddlePosition {
    float get ();
    void set (float value);
}

Field Value

The middle position.
See Also

ColourInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which merges the colour of particles towards a single colour over their lifetime. Works best when Particles are being released with random colours, where you require the particles to have a uniform colour at the end of their lifetime.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class ColourMergeModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class ColourMergeModifier _
    Inherits Modifier

Visual C++

public ref class ColourMergeModifier sealed : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
  ProjectMercury.Modifiers::ColourMergeModifier
See Also

ColourMergeModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ColourMergeModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColourMergeModifier</td>
<td>Initializes a new instance of the ColourMergeModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::DeepCopy().)</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::Process(Single, Particle*, Int32).)</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>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>MergeColour</td>
<td>The final colour of Particles when they are retired.</td>
</tr>
</tbody>
</table>
See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ColourMergeModifier` class

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public ColourMergeModifier()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
ColourMergeModifier()
```
See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ColourMergeModifier type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MergeColour</td>
<td>The final colour of Particles when they are retired.</td>
</tr>
</tbody>
</table>
See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The final colour of Particles when they are retired.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector3 MergeColour

Visual Basic (Declaration)

Public MergeColour As Vector3

Visual C++

public:
Vector3 MergeColour
See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ColourMergeModifier type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..::.DeepCopy().()().)</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. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (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>Process</td>
<td>Processes the particles. (Overides Modifier..::.Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

*Methods*
See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

### Return Value

See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** [ProjectMercury.Modifiers](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

Visual C++

```cpp
public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override
```

Parameters

dt
  Type: System:::Single
  Elapsed time in whole and fractional seconds.

particleArray
  Type: ProjectMercury:::Particle *
  A pointer to an array of particles.

count
  Type: System:::Int32
The number of particles which need to be processed.
See Also

ColourMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which gradually changes the colour of a Particle over the course of its lifetime.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public sealed class ColourModifier : Modifier
```

**Visual Basic (Declaration)**

```vbnet
Public NotInheritable Class ColourModifier_
    Inherits Modifier
```

**Visual C++**

```cpp
public ref class ColourModifier sealed : public Modifier
```
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
  ProjectMercury.Modifiers::ColourModifier
See Also

ColourModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ColourModifier type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColourModifier</td>
<td>Initializes a new instance of the <em>ColourModifier</em> class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier:::DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier:::Process(Single, Particle*, Int32).)</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>InitialColour</td>
<td>The initial colour of Particles when they are released.</td>
</tr>
<tr>
<td>UltimateColour</td>
<td>The ultimate colour of Particles when they are retired.</td>
</tr>
</tbody>
</table>
See Also

**ColourModifier Class**  
**ProjectMercury.Modifiers Namespace**

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ColourModifier` class

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public ColourModifier()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
ColourModifier()
```
See Also

[ColourModifier Class](#)
[ProjectMercury.Modifiers Namespace](#)

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ColourModifier type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitialColour</td>
<td>The initial colour of Particles when they are released.</td>
</tr>
<tr>
<td>UltimateColour</td>
<td>The ultimate colour of Particles when they are retired.</td>
</tr>
</tbody>
</table>
See Also

ColourModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The initial colour of Particles when they are released.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
**Syntax**

**C#**

```csharp
public Vector3 InitialColour
```

**Visual Basic (Declaration)**

```vbnet
Public InitialColour As Vector3
```

**Visual C++**

```cpp
public:
Vector3 InitialColour
```
See Also

ColourModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ultimate colour of Particles when they are retired.

**Namespace:** ProjectMercury.Modifiers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public Vector3 UltimateColour
```

**Visual Basic (Declaration)**

```vbnet
Public UltimateColour As Vector3
```

**Visual C++**

```cpp
public:
Vector3 UltimateColour
```
See Also

ColourModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **ColourModifier** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier.***: DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier.***: Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ColourModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public override Modifier DeepCopy()

**Visual Basic (Declaration)**

Public Overrides Function DeepCopy As Modifier

**Visual C++**

public:
virtual Modifier^ DeepCopy() override

**Return Value**

See Also

ColourModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

[ColourModifier Class](#)
[ProjectMercury.Modifiers Namespace](#)

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which applies a damping force to a Particle over its lifetime.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class DampingModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class DampingModifier _
   Inherits Modifier

Visual C++

public ref class DampingModifier sealed : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
   ProjectMercury.Modifiers::DampingModifier
See Also

DampingModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **DampingModifier** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DampingModifier</td>
<td>Initializes a new instance of the <strong>DampingModifier</strong> class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..::. DeepCopy()().)</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. (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>Process</td>
<td>Processes the particles. (Overides Modifier..::. Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DampingCoefficient</td>
<td>The damping coefficient.</td>
</tr>
</tbody>
</table>
See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `DampingModifier` class

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public DampingModifier()  

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
DampingModifier()
See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `DampingModifier` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DampingCoefficient</td>
<td>The damping coefficient.</td>
</tr>
</tbody>
</table>
See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The damping coefficient.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public float DampingCoefficient

Visual Basic (Declaration)
Public DampingCoefficient As Single

Visual C++
public:
float DampingCoefficient
See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `DampingModifier` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified &lt;code&gt;Object&lt;/code&gt; is equal to the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an &lt;code&gt;Object&lt;/code&gt; to attempt to free resources and perform other cleanup operations before the &lt;code&gt;Object&lt;/code&gt; is reclaimed by garbage collection. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the &lt;code&gt;Type&lt;/code&gt; of the current instance. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a &lt;code&gt;String&lt;/code&gt; that represents the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
</tbody>
</table>
See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public override Modifier DeepCopy()
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function DeepCopy As Modifier
```

### Visual C++

```cpp
public:
virtual Modifier^ DeepCopy() override
```

## Return Value

See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

**Parameters**

`dt`

Type: `System::Single`
Elapsed time in whole and fractional seconds.

`particleArray`

Type: `ProjectMercury::Particle` *
A pointer to an array of particles.

`count`

Type: `System::Int32`
The number of particles which need to be processed.
See Also

DampingModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which adjusts the hue of a Particles colour over time.

**Namespace:** [ProjectMercury.Modifiers](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class HueShiftModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class HueShiftModifier _
    Inherits Modifier

Visual C++

public ref class HueShiftModifier sealed : public Modifier
Inheritance Hierarchy

System..:::Object
  ProjectMercury.Modifiers..:::Modifier
    ProjectMercury.Modifiers..:::HueShiftModifier
See Also

HueShiftModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **HueShiftModifier** type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HueShiftModifier</td>
<td>Initializes a new instance of the HueShiftModifier class</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides <code>Modifier::DeepCopy()</code>)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles. (Overrides <code>Modifier::Process(Single, Particle*, Int32)</code>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>HueShift</td>
<td>The amount to adjust the hue in degrees per second.</td>
</tr>
</tbody>
</table>
See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
HueShiftModifier Constructor

Initializes a new instance of the HueShiftModifier class

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public HueShiftModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
HueShiftModifier()
See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **HueShiftModifier** type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HueShift</td>
<td>The amount to adjust the hue in degrees per second.</td>
</tr>
</tbody>
</table>
See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
HueShiftModifier.HueShift Field

The amount to adjust the hue in degrees per second.

**Namespace:** ProjectMercury.Modifiers
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float HueShift

Visual Basic (Declaration)

Public HueShift As Single

Visual C++

public:
float HueShift
See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **HueShiftModifier** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;DeepCopy&gt;().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** [ProjectMercury.Modifiers](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process (_,
    dt As Single, _,
    particleArray As Particle*, _,
    count As Integer _
)

Visual C++

public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override

Parameters

dt
    Type: System::::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::::Particle *
    A pointer to an array of particles.

count
    Type: System::::Int32
The number of particles which need to be processed.
See Also

HueShiftModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier that applies a constant force vector to Particles over their lifetime.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class LinearGravityModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class LinearGravityModifier _
    Inherits Modifier

Visual C++

public ref class LinearGravityModifier sealed : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
   ProjectMercury.Modifiers::LinearGravityModifier
See Also

LinearGravityModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `LinearGravityModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearGravityModifier</td>
<td>Initializes a new instance of the LinearGravityModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::DeepCopy().)</td>
</tr>
<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. (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>Process</strong></td>
<td>Processes the particles. (Overides Modifier::Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravity</td>
<td>Gets or sets the gravity vector.</td>
</tr>
</tbody>
</table>
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the LinearGravityModifier class

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

class LinearGravityModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:

LinearGravityModifier();
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The LinearGravityModifier type exposes the following members.
Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravity</td>
<td>Gets or sets the gravity vector.</td>
</tr>
</tbody>
</table>
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the gravity vector.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public Vector2 Gravity
```

**Visual Basic (Declaration)**

```vbnet
Public Gravity As Vector2
```

**Visual C++**

```cpp
public:
    Vector2 Gravity
```
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The LinearGravityModifier type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt;DeepCopy()&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified &lt;code&gt;Object&lt;/code&gt; is equal to the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an &lt;code&gt;Object&lt;/code&gt; to attempt to free resources and perform other cleanup operations before the &lt;code&gt;Object&lt;/code&gt; is reclaimed by garbage collection. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the &lt;code&gt;Type&lt;/code&gt; of the current instance. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Processes the particles. (Overrides Modifier::&lt;code&gt;Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a &lt;code&gt;String&lt;/code&gt; that represents the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
</tbody>
</table>
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

[Missing <returns> documentation for
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

**Parameters**

**dt**

Type: **System::Single**

Elapsed time in whole and fractional seconds.

**particleArray**

Type: **ProjectMercury::Particle** *

A pointer to an array of particles.

**count**

Type: **System::Int32**
The number of particles which need to be processed.
See Also

LinearGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Modifies the base class for an object which modifies Particle values.

**Namespace:** ProjectMercury.Modifiers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public abstract class Modifier

Visual Basic (Declaration)
Public MustInherit Class Modifier

Visual C++
public ref class Modifier abstract
Inheritance Hierarchy

System..::.Object
ProjectMercury.Modifiers..::.Modifier
  ProjectMercury.Modifiers..::.ColourInterpolatorModifier
  ProjectMercury.Modifiers..::.ColourMergeModifier
  ProjectMercury.Modifiers..::.ColourModifier
  ProjectMercury.Modifiers..::.DampingModifier
  ProjectMercury.Modifiers..::.HueShiftModifier
  ProjectMercury.Modifiers..::.LinearGravityModifier
  ProjectMercury.Modifiers..::.OpacityInterpolatorModifier
  ProjectMercury.Modifiers..::.OpacityModifier
  ProjectMercury.Modifiers..::.OpacityOscillator
  ProjectMercury.Modifiers..::.PlatformModifier
  ProjectMercury.Modifiers..::.RadialForceModifier
  ProjectMercury.Modifiers..::.RadialGravityModifier
  ProjectMercury.Modifiers..::.RectangleConstraintDeflector
  ProjectMercury.Modifiers..::.RectangleForceModifier
  ProjectMercury.Modifiers..::.RotationModifier
  ProjectMercury.Modifiers..::.RotationRateModifier
  ProjectMercury.Modifiers..::.ScaleInterpolatorModifier
  ProjectMercury.Modifiers..::.ScaleMergeModifier
  ProjectMercury.Modifiers..::.ScaleModifier
  ProjectMercury.Modifiers..::.ScaleOscillator
  ProjectMercury.Modifiers..::.SineForceModifier
  ProjectMercury.Modifiers..::.TrajectoryRotationModifier
  ProjectMercury.Modifiers..::.VelocityClampModifier
See Also

Modifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **Modifier** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modifier</td>
<td>Initializes a new instance of the Modifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.                           (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance.                               (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.                         (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.            (Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

Modifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the Modifier class

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected Modifier()

Visual Basic (Declaration)

Protected Sub New

Visual C++

protected:
Modifier()
See Also

Modifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `Modifier` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</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>GetHashCode</td>
<td>Serves as a hash function for a particular type.</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>Process</td>
<td>Processes the particles.</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>
See Also

Modifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public abstract Modifier DeepCopy()

Visual Basic (Declaration)

Public MustOverride Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() abstract

Return Value

See Also

Modifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public abstract void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

### Visual Basic (Declaration)

```vbnet
Public MustOverride Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

### Visual C++

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

## Parameters

### dt
- Type: `System::::Single`
- Elapsed time in whole and fractional seconds.

### particleArray
- Type: `ProjectMercury::::Particle *`
- A pointer to an array of particles.

### count
- Type: `System::::Int32`
The number of particles which need to be processed.
See Also

Modifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a collection of Modifiers.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class ModifierCollection : List<Modifier>

Visual Basic (Declaration)

Public Class ModifierCollection
    Inherits List(Of Modifier)

Visual C++

public ref class ModifierCollection : public List<Modifier^>
Inheritance Hierarchy

System..:::Object
System.Collections.Generic..:::List<Of <(Modifier)>>
ProjectMercury.Modifiers..:::ModifierCollection
See Also

ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ModifierCollection` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModifierCollection</td>
<td>Initializes a new instance of the ModifierCollection class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an object to the end of the <code>List&lt;Of &lt;(T)&gt; &gt;</code>. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>). Adds the elements of the specified collection to the end of the <code>List&lt;Of &lt;(T)&gt; &gt;</code>. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>).</td>
</tr>
<tr>
<td>AddRange</td>
<td>Returns a read-only <code>IList&lt;Of &lt;(T)&gt; &gt;</code> wrapper for the current collection. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>).</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Returns a <code>read-only </code>IList&lt;Of &lt;(T)&gt; &gt;<code>wrapper for the current collection. (Inherited from</code>List&lt;Of &lt;(Modifier)&gt; &gt;`).</td>
</tr>
<tr>
<td>BinarySearch</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all elements from the <code>List&lt;Of &lt;(T)&gt; &gt;</code>. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>). Determines whether an element is in the <code>List&lt;Of &lt;(T)&gt; &gt;</code>. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>).</td>
</tr>
<tr>
<td>Contains</td>
<td>Converts the elements in the current <code>List&lt;Of &lt;(T)&gt; &gt;</code> to another type, and returns a list containing the converted elements. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>).</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the <code>ModifierCollection</code>.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ConvertAll&lt;Of &lt;(TOutput)&gt; &gt;</td>
<td>Converts the elements in the current <code>List&lt;Of &lt;(T)&gt; &gt;</code> to another type, and returns a list containing the converted elements. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>).</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td>Exists</td>
<td>Determines whether the <code>List&lt;Of &lt;(T)&gt; &gt;</code> contains elements that match the conditions defined by the specified predicate. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt; &gt;</code>).</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>).</td>
</tr>
</tbody>
</table>
Find

Searches for an element that matches the conditions defined by the specified predicate, and returns the first occurrence within the entire `List(Of (T)>).` (Inherited from `List(Of (Modifier)>).`)

FindAll

Retrieves all the elements that match the conditions defined by the specified predicate. (Inherited from `List(Of (Modifier)>).`)

FindIndex

Overloaded.

FindLast

Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire `List(Of (T)>).` (Inherited from `List(Of (Modifier)>).`)

FindLastIndex

Overloaded.

ForEach

Performs the specified action on each element of the `List(Of (T)>).` (Inherited from `List(Of (Modifier)>).`)

GetEnumerator

Returns an enumerator that iterates through the `List(Of (T)>).` (Inherited from `List(Of (Modifier)>).`)

GetHashCode

Serves as a hash function for a particular type. (Inherited from `Object`.)

GetRange

Creates a shallow copy of a range of elements in the source `List(Of (T)>).` (Inherited from `List(Of (Modifier)>).`)

GetType

Gets the `Type` of the current instance. (Inherited from `Object`.)

IndexOf

Overloaded.

Insert

Inserts an element into the `List(Of (T)>)` at the specified index. (Inherited from `List(Of (Modifier)>).`)

InsertRange

Inserts the elements of a collection into the `List(Of (T)>)` at the specified index. (Inherited from `List(Of (Modifier)>).`)

LastIndexOf

Overloaded.

MemberwiseClone

Creates a shallow copy of the current `Object`. (Inherited from `Object`.)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>List&lt;Of&lt; T&gt;&gt;</code>. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
<tr>
<td>RemoveAll</td>
<td>Removes all the elements that match the conditions defined by the specified predicate. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the element at the specified index of the <code>List&lt;Of&lt; T&gt;&gt;</code>. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
<tr>
<td>RemoveRange</td>
<td>Removes a range of elements from the <code>List&lt;Of&lt; T&gt;&gt;</code>. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
<tr>
<td>Reverse</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Sort</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ToArray</td>
<td>Copies the elements of the <code>List&lt;Of&lt; T&gt;&gt;</code> to a new array. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TrimExcess</td>
<td>Sets the capacity to the actual number of elements in the <code>List&lt;Of&lt; T&gt;&gt;</code>, if that number is less than a threshold value. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
<tr>
<td>TrueForAll</td>
<td>Determines whether every element in the <code>List&lt;Of&lt; T&gt;&gt;</code> matches the conditions defined by the specified predicate. (Inherited from <code>List&lt;Of&lt;(Modifier)&gt;&gt;</code>).</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from List(Of (Modifier)&gt;).)</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements actually contained in the List(Of (T&gt;).) (Inherited from List(Of (Modifier)&gt;).)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from List(Of (Modifier)&gt;).)</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ModifierCollection` class

**Namespace:**  `ProjectMercury.Modifiers`

**Assembly:**  `ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0`
**Syntax**

**C#**

```csharp
public ModifierCollection()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
ModifierCollection()
```
See Also

ModifierCollection Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **ModifierCollection** type exposes the following members.
<table>
<thead>
<tr>
<th>Methods</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add</td>
<td>Adds an object to the end of the <code>List(Of &lt;T&gt;)</code>. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>). Adds the elements of the specified collection to the end of the <code>List(Of &lt;T&gt;)</code>. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>AddRange</td>
<td>Returns a read-only <code>IList(Of &lt;T&gt;)</code> wrapper for the current collection. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>AsReadOnly</td>
<td>Returns a read-only <code>IList(Of &lt;T&gt;)</code> wrapper for the current collection. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>BinarySearch</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Clear</td>
<td>Removes all elements from the <code>List(Of &lt;T&gt;)</code>. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>). Determines whether an element is in the <code>List(Of &lt;T&gt;)</code>. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>Contains</td>
<td>Converts the elements in the current <code>List(Of &lt;T&gt;)</code> to another type, and returns a list containing the converted elements. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>ConvertAll(Of &lt;TOutput&gt;)</td>
<td>Converts the elements in the current <code>List(Of &lt;T&gt;)</code> to another type, and returns a list containing the converted elements. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>CopyTo</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>DeepCopy</td>
<td>Returns a deep copy of the ModifierCollection. Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td></td>
<td>Equals</td>
<td>Determines whether the <code>List(Of &lt;T&gt;)</code> contains elements that match the conditions defined by the specified predicate. (Inherited from <code>List(Of &lt;Modifier&gt;)</code>).</td>
</tr>
<tr>
<td></td>
<td>Exists</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td></td>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>).</td>
</tr>
</tbody>
</table>
Find

Searches for an element that matches the conditions defined by the specified predicate, and returns the first occurrence within the entire `List<Of <(T)>`. (Inherited from `List<Of <(Modifier)>`).)

FindAll

Retrieves all the elements that match the conditions defined by the specified predicate. (Inherited from `List<Of <(Modifier)>`).)

FindIndex

Overloaded.

FindLast

Searches for an element that matches the conditions defined by the specified predicate, and returns the last occurrence within the entire `List<Of <(T)>`. (Inherited from `List<Of <(Modifier)>`).)

FindLastIndex

Overloaded.

ForEach

Performs the specified action on each element of the `List<Of <(T)>`. (Inherited from `List<Of <(Modifier)>`).)

GetEnumerator

Returns an enumerator that iterates through the `List<Of <(T)>`. (Inherited from `List<Of <(Modifier)>`).)

GetHashCode

Serves as a hash function for a particular type. (Inherited from `Object`.)

GetRange

Creates a shallow copy of a range of elements in the source `List<Of <(T)>`. (Inherited from `List<Of <(Modifier)>`).)

GetType

Gets the `Type` of the current instance. (Inherited from `Object`.)

IndexOf

Overloaded.

Insert

Inserts an element into the `List<Of <(T)>>` at the specified index. (Inherited from `List<Of <(Modifier)>`).)

InsertRange

Inserts the elements of a collection into the `List<Of <(T)>>` at the specified index. (Inherited from `List<Of <(Modifier)>`).)

LastIndexOf

Overloaded.

MemberwiseClone

Creates a shallow copy of the current `Object`. (Inherited from `Object`.)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remove</strong></td>
<td>Removes the first occurrence of a specific object from the <code>List(Of </code>&lt;T&gt;<code>&gt;). (Inherited from </code>List(Of <code>&lt;Modifier&gt;</code>&gt;`)).</td>
</tr>
<tr>
<td><strong>RemoveAll</strong></td>
<td>Removes all the elements that match the conditions defined by the specified predicate. (Inherited from <code>List(Of </code>&lt;Modifier&gt;<code>&gt;</code>)).</td>
</tr>
<tr>
<td><strong>RemoveAt</strong></td>
<td>Removes the element at the specified index of the <code>List(Of </code>&lt;T&gt;<code>)</code>. (Inherited from <code>List(Of </code>&lt;Modifier&gt;<code>&gt;</code>)).</td>
</tr>
<tr>
<td><strong>RemoveRange</strong></td>
<td>Removes a range of elements from the <code>List(Of </code>&lt;T&gt;<code>)</code>. (Inherited from <code>List(Of </code>&lt;Modifier&gt;<code>&gt;</code>)).</td>
</tr>
<tr>
<td><strong>Reverse</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Sort</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToArray</strong></td>
<td>Copies the elements of the <code>List(Of </code>&lt;T&gt;<code>)</code> to a new array. (Inherited from <code>List(Of </code>&lt;Modifier&gt;<code>&gt;</code>)).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>TrimExcess</strong></td>
<td>Sets the capacity to the actual number of elements in the <code>List(Of </code>&lt;T&gt;<code>)</code>, if that number is less than a threshold value. (Inherited from <code>List(Of </code>&lt;Modifier&gt;<code>&gt;</code>)).</td>
</tr>
<tr>
<td><strong>TrueForAll</strong></td>
<td>Determines whether every element in the <code>List(Of </code>&lt;T&gt;<code>)</code> matches the conditions defined by the specified predicate. (Inherited from <code>List(Of </code>&lt;Modifier&gt;<code>&gt;</code>)).</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinarySearch(T)</td>
<td>Searches the entire sorted <code>List&lt;Of (T)&gt;</code> for an element using the default comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
<tr>
<td>BinarySearch(T, IComparer&lt;Of (T)&gt;))</td>
<td>Searches the entire sorted <code>List&lt;Of (T)&gt;</code> for an element using the specified comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
<tr>
<td>BinarySearch(Int32, Int32, T, IComparer&lt;Of (T)&gt;))</td>
<td>Searches a range of elements in the sorted <code>List&lt;Of (T)&gt;</code> for an element using the specified comparer and returns the zero-based index of the element. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ModifierCollection::CopyTo Method

ModifierCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CopyTo(array&lt;T&gt;[], Int32)</strong></td>
<td>Copies the entire <code>List&lt;Of &lt;(T)&gt;&gt;</code> to a compatible one-dimensional array, starting at the beginning of the target array. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt;&gt;</code>.)</td>
</tr>
<tr>
<td><strong>CopyTo(array&lt;T&gt;[][], Int32)</strong></td>
<td>Copies the entire <code>List&lt;Of &lt;(T)&gt;&gt;</code> to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt;&gt;</code>.)</td>
</tr>
<tr>
<td><strong>CopyTo(Int32, array&lt;T&gt;[][], Int32, Int32)</strong></td>
<td>Copies a range of elements from the <code>List&lt;Of &lt;(T)&gt;&gt;</code> to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from <code>List&lt;Of &lt;(Modifier)&gt;&gt;</code>.)</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the ModifierCollection.

Namespace: ProjectMercury.Modifiers  
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public ModifierCollection DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Function DeepCopy As ModifierCollection
```

**Visual C++**

```cpp
public:
 ModifierCollection^ DeepCopy()
```

**Return Value**

A deep copy of the ModifierCollection.
See Also

ModifierCollection Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference

ModifierCollection:::FindIndex Method

ModifierCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>FindIndex(Predicate&lt;Of (T)&gt;))</code></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the entire <code>List&lt;Of (T)&gt;</code>. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
<tr>
<td><code>FindIndex(Int32, Predicate&lt;Of (T)&gt;))</code></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the <code>List&lt;Of (T)&gt;</code> that extends from the specified index to the last element. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
<tr>
<td><code>FindIndex(Int32, Int32, Predicate&lt;Of (T)&gt;))</code></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the <code>List&lt;Of (T)&gt;</code> that starts at the specified index and contains the specified number of elements. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
ModifierCollection..::.FindLastIndex Method

ModifierCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FindLastIndex(Predicate&lt;Of (T)&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the entire <code>List&lt;Of (T)&gt;</code>. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
<tr>
<td><strong>FindLastIndex(Int32, Predicate&lt;Of (T)&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the <code>List&lt;Of (T)&gt;</code> that extends from the first element to the specified index. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
<tr>
<td><strong>FindLastIndex(Int32, Int32, Predicate&lt;Of (T)&gt;)</strong></td>
<td>Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the <code>List&lt;Of (T)&gt;</code> that contains the specified number of elements and ends at the specified index. (Inherited from <code>List&lt;Of (Modifier)&gt;</code>).</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **IndexOf(T)** | Searches for the specified object and returns the zero-based index of the first occurrence within the entire List<T>.  
(Inherited from List<T>.) |
| **IndexOf(T, Int32)** | Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the List<T> that extends from the specified index to the last element.  
(Inherited from List<T>.) |
| **IndexOf(T, Int32, Int32)** | Searches for the specified object and returns the zero-based index of the first occurrence within the range of elements in the List<T> that starts at the specified index and contains the specified number of elements.  
(Inherited from List<T>.) |
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference

ModifierCollection..::.LastIndexOf Method

ModifierCollection Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| LastIndexOf(T) | Searches for the specified object and returns the zero-based index of the last occurrence within the entire List(Of (T)).  
(Inherited from List(Of (Modifier)).) |
| LastIndexOf(T, Int32) | Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the List(Of (T)) that extends from the first element to the specified index.  
(Inherited from List(Of (Modifier)).) |
| LastIndexOf(T, Int32, Int32) | Searches for the specified object and returns the zero-based index of the last occurrence within the range of elements in the List(Of (T)) that contains the specified number of elements and ends at the specified index.  
(Inherited from List(Of (Modifier)).) |
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse()()</td>
<td>Reverses the order of the elements in the entire List&lt;Of &lt;T&gt;&gt;.</td>
</tr>
<tr>
<td>(Inherited from List&lt;Of &lt;(Modifier)&gt;&gt;.)</td>
<td>Reverses the order of the elements in the specified range.</td>
</tr>
<tr>
<td>Reverse(Int32, Int32)</td>
<td>(Inherited from List&lt;Of &lt;(Modifier)&gt;&gt;.).</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sort()</strong></td>
<td>Sorts the elements in the entire <code>List&lt;Of&lt;T&gt;&gt;</code> using the default comparer. (Inherited from <code>List&lt;Of&lt;Modifier&gt;&gt;</code>).</td>
</tr>
<tr>
<td><strong>Sort(IComparer&lt;Of&lt;T&gt;&gt;)</strong>*</td>
<td>Sorts the elements in the entire <code>List&lt;Of&lt;T&gt;&gt;</code> using the specified comparer. (Inherited from <code>List&lt;Of&lt;Modifier&gt;&gt;</code>).</td>
</tr>
<tr>
<td><strong>Sort(Comparison&lt;Of&lt;T&gt;&gt;)</strong>*</td>
<td>Sorts the elements in the entire <code>List&lt;Of&lt;T&gt;&gt;</code> using the specified <code>Comparison&lt;Of&lt;T&gt;&gt;</code>. (Inherited from <code>List&lt;Of&lt;Modifier&gt;&gt;</code>).</td>
</tr>
<tr>
<td><strong>Sort(Int32, Int32, IComparer&lt;Of&lt;T&gt;&gt;)</strong>*</td>
<td>Sorts the elements in a range of elements in <code>List&lt;Of&lt;T&gt;&gt;</code> using the specified comparer. (Inherited from <code>List&lt;Of&lt;Modifier&gt;&gt;</code>).</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ModifierCollection Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ModifierCollection` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>Gets or sets the total number of elements the internal data structure can hold without resizing. (Inherited from List&lt;Of &lt;(Modifier)&gt;).)</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements actually contained in the List&lt;Of &lt;(T)&gt;&gt;. (Inherited from List&lt;Of &lt;(Modifier)&gt;).)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from List&lt;Of &lt;(Modifier)&gt;).)</td>
</tr>
</tbody>
</table>
See Also

ModifierCollection Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a modifier which changes the opacity of particles based on a linear interpolation over three values.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public class OpacityInterpolatorModifier : Modifier
```

Visual Basic (Declaration)

```vbnet
Public Class OpacityInterpolatorModifier
    Inherits Modifier
```

Visual C++

```cpp
public ref class OpacityInterpolatorModifier : public Modifier
```
Inheritance Hierarchy

System..:::Object
ProjectMercury.Modifiers..:::Modifier
   ProjectMercury.Modifiers..:::OpacityInterpolatorModifier
See Also

OpacityInterpolatorModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `OpacityInterpolatorModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpacityInterpolatorModifier</td>
<td>Initializes a new instance of the <a href="OpacityInterpolatorModifier">OpacityInterpolatorModifier</a> class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..::.DeepCopy().)</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. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (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>Process</td>
<td>Processes the particles. (Overides Modifier..::.Process(Single, Particle*, Int32).)</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>FinalOpacity</td>
<td>Gets or sets the final opacity.</td>
</tr>
<tr>
<td>InitialOpacity</td>
<td>Gets or sets the initial opacity.</td>
</tr>
<tr>
<td>MiddleOpacity</td>
<td>Gets or sets the middle opacity.</td>
</tr>
<tr>
<td>MiddlePosition</td>
<td>Gets or sets the middle opacity position.</td>
</tr>
</tbody>
</table>
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `OpacityInterpolatorModifier` class

**Namespace:**  `ProjectMercury.Modifiers`

**Assembly:**  `ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0`
Syntax

C#

public OpacityInterpolatorModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
OpacityInterpolatorModifier()
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **OpacityInterpolatorModifier** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;DeepCopy()()&gt;.)</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;Process(Single, Particle*, Int32&gt;).)</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>
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityInterpolatorModifier..::.DeepCopy Method

OpacityInterpolatorModifier Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Returns a deep copy of the Modifier implementation.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** [ProjectMercury.Modifiers](https://example.com)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
**Syntax**

**C#**

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override
```

**Parameters**

**dt**
- Type: System::Single
- Elapsed time in whole and fractional seconds.

**particleArray**
- Type: ProjectMercury::Particle *
- A pointer to an array of particles.

**count**
- Type: System::Int32
The number of particles which need to be processed.
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **OpacityInterpolatorModifier** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FinalOpacity</strong></td>
<td>Gets or sets the final opacity.</td>
</tr>
<tr>
<td><strong>InitialOpacity</strong></td>
<td>Gets or sets the initial opacity.</td>
</tr>
<tr>
<td><strong>MiddleOpacity</strong></td>
<td>Gets or sets the middle opacity.</td>
</tr>
<tr>
<td><strong>MiddlePosition</strong></td>
<td>Gets or sets the middle opacity position.</td>
</tr>
</tbody>
</table>
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the final opacity.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public float FinalOpacity { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property FinalOpacity As Single
```

Visual C++

```c++
public:
property float FinalOpacity {
    float get ();
    void set (float value);
}
```

Field Value

The final opacity.
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the initial opacity.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public float InitialOpacity { get; set; }
```

Visual Basic (Declaration)

Public Property InitialOpacity As Single

Visual C++

```cpp
public:
property float InitialOpacity {
    float get ();
    void set (float value);
}
```

Field Value

The initial opacity.
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the middle opacity.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  [ProjectMercury](#) (in [ProjectMercury.dll](#)) Version: 3.1.0.0
Syntax

C#

public float MiddleOpacity { get; set; }

Visual Basic (Declaration)

Public Property MiddleOpacity As Single

Visual C++

public:
    property float MiddleOpacity {
        float get ();
        void set (float value);
    }

Field Value

The middle opacity.
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the middle opacity position.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MiddlePosition { get; set; }

Visual Basic (Declaration)

Public Property MiddlePosition As Single

Visual C++

public:
property float MiddlePosition {
    float get () ;
    void set (float value); 
}

Field Value

The middle position.
See Also

OpacityInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which gradually changes the opacity of a Particle over its lifetime.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class OpacityModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class OpacityModifier _
    Inherits Modifier

Visual C++

public ref class OpacityModifier sealed : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
  ProjectMercury.Modifiers::OpacityModifier
See Also

OpacityModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityModifier Members

OpacityModifier Class  Constructors  Methods  Properties  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

The OpacityModifier type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpacityModifier</td>
<td>Initializes a new instance of the OpacityModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier..::. DeepCopy().)</td>
</tr>
<tr>
<td></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>Equals</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>Finalize</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>Processes the particles.</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier..::. Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>MemberwiseClone</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>Initial</td>
<td>Gets or sets the initial opacity of Particles as they are released.</td>
</tr>
<tr>
<td>Ultimate</td>
<td>Gets or sets the ultimate opacity of Particles as they are retired.</td>
</tr>
</tbody>
</table>
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `OpacityModifier` class

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public OpacityModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
OpacityModifier()
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **OpacityModifier** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;ref&gt; DeepCopy&lt;/ref&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified &lt;ref&gt; Object &lt;/ref&gt; is equal to the current &lt;ref&gt; Object &lt;/ref&gt;.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an &lt;ref&gt; Object &lt;/ref&gt; to attempt to free resources and perform other cleanup operations before the &lt;ref&gt; Object &lt;/ref&gt; is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the &lt;ref&gt; Type &lt;/ref&gt; of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;ref&gt; Object &lt;/ref&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current &lt;ref&gt; Object &lt;/ref&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;ref&gt; Object &lt;/ref&gt;.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;ref&gt; Process&lt;/ref&gt;(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a &lt;ref&gt; String &lt;/ref&gt; that represents the current &lt;ref&gt; Object &lt;/ref&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;ref&gt; Object &lt;/ref&gt;.)</td>
</tr>
</tbody>
</table>
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityModifier DeepCopy Method

OpacityModifier Class

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]

Returns a deep copy of the Modifier implementation.
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override void Process(  
    float dt,  
    Particle* particleArray,  
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
virtual void Process(   
    float dt,  
    Particle* particleArray,  
    int count
) override
```

### Parameters

**dt**
- Type: **System::Single**
- Elapsed time in whole and fractional seconds.

**particleArray**
- Type: **ProjectMercury::Particle** *
- A pointer to an array of particles.

**count**
- Type: **System::Int32**
The number of particles which need to be processed.
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `OpacityModifier` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Gets or sets the initial opacity of Particles as they are released.</td>
</tr>
<tr>
<td>Ultimate</td>
<td>Gets or sets the ultimate opacity of Particles as they are retired.</td>
</tr>
</tbody>
</table>
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityModifier.Initial Property

Gets or sets the initial opacity of Particles as they are released.

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public float Initial { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property Initial As Single
```

### Visual C++

```cpp
public:
    property float Initial {
        float get ();
        void set (float value);
    }
```
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the ultimate opacity of Particles as they are retired.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public float Ultimate { get; set; }

Visual Basic (Declaration)

Public Property Ultimate As Single

Visual C++

public:
property float Ultimate {
    float get ();
    void set (float value);
}
See Also

OpacityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which adjusts the opacity of Particles based on a sine wave.

**Namespace:** ProjectMercury.Modifiers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class OpacityOscillator : Modifier

Visual Basic (Declaration)

Public Class OpacityOscillator _
   Inherits Modifier

Visual C++

public ref class OpacityOscillator : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
ProjectMercury.Modifiers::OpacityOscillator
See Also

OpacityOscillator Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **OpacityOscillator** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OpacityOscillator</strong></td>
<td>Initializes a new instance of the <strong>OpacityOscillator</strong> class</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..::. DeepCopy()().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier..::. Process(Single, Particle*, Int32).)</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>Frequency</strong></td>
<td>Gets or sets the oscillator frequency (the number of cycles per second).</td>
</tr>
<tr>
<td><strong>MaximumOpacity</strong></td>
<td>Gets or sets the maximum opacity (the opacity of Particles at the positive peak of the sine wave).</td>
</tr>
<tr>
<td><strong>MinimumOpacity</strong></td>
<td>Gets or sets the minimum opacity (the opacity of Particles at the negative peak of the sine wave).</td>
</tr>
</tbody>
</table>
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityOscillator Constructor

Initializes a new instance of the `OpacityOscillator` class

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public OpacityOscillator()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
OpacityOscillator()
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **OpacityOscillator** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..::. DeepCopy().).</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier..::. Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityOscillator DeepCopy Method

OpacityOscillator Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Returns a deep copy of the Modifier implementation.

Namespace:  ProjectMercury.Modifiers  
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

**Return Value**

See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

    public override void Process(
        float dt,
        Particle* particleArray,
        int count
    )

Visual Basic (Declaration)

    Public Overrides Sub Process ( _
        dt As Single, _
        particleArray As Particle*, _
        count As Integer _
    )

Visual C++

    public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override

Parameters

dt
    Type: System:::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury:::Particle *
    A pointer to an array of particles.

count
    Type: System:::Int32
The number of particles which need to be processed.
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityOscillator Properties

OpacityOscillator Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

The **OpacityOscillator** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency</strong></td>
<td>Gets or sets the oscillator frequency (the number of cycles per second).</td>
</tr>
<tr>
<td><strong>MaximumOpacity</strong></td>
<td>Gets or sets the maximum opacity (the opacity of Particles at the positive peak of the sine wave).</td>
</tr>
<tr>
<td><strong>MinimumOpacity</strong></td>
<td>Gets or sets the minimum opacity (the opacity of Particles at the negative peak of the sine wave).</td>
</tr>
</tbody>
</table>
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityOscillator.Frequency Property

Gets or sets the oscillator frequency (the number of cycles per second).

**Namespace:** ProjectMercury.Modifiers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Frequency { get; set; }

Visual Basic (Declaration)

Public Property Frequency As Single

Visual C++

public:
property float Frequency {
    float get ();
    void set (float value);
}
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
OpacityOscillator....:::MaximumOpacity Property

OpacityOscillator Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Gets or sets the maximum opacity (the opacity of Particles at the positive peak of the sine wave).

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MaximumOpacity { get; set; }

Visual Basic (Declaration)

Public Property MaximumOpacity As Single

Visual C++

public:
property float MaximumOpacity {
    float get ();
    void set (float value);
}

Field Value

The maximum opacity.
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the minimum opacity (the opacity of Particles at the negative peak of the sine wave).

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public float MinimumOpacity { get; set; }
```

**Visual Basic (Declaration)**

Public Property MinimumOpacity As Single

**Visual C++**

```cpp
public:
property float MinimumOpacity {
       float get ();
     void set (float value);
}
```

**Field Value**

The minimum opacity.
See Also

OpacityOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which freezes a particle when it comes into contact with a bounding box.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class PlatformModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class PlatformModifier _
   Inherits Modifier

Visual C++

public ref class PlatformModifier sealed : public Modifier
Inheritance Hierarchy

System:::Object
ProjectMercury.Modifiers:::Modifier
  ProjectMercury.Modifiers:::PlatformModifier
See Also

PlatformModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `PlatformModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlatformModifier</td>
<td>Initializes a new instance of the PlatformModifier class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..:::DeepCopy()()().)</td>
</tr>
<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. (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>Process</strong></td>
<td>Processes the particles. (Overrides Modifier..:::Process(Single, Particle*, Int32).)</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>Platforms</td>
<td>The list of platforms.</td>
</tr>
</tbody>
</table>
See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `PlatformModifier` class.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public PlatformModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
PlatformModifier()
See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `PlatformModifier` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier:: DeepCopy()().)</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier:: Process(Single, Particle*, Int32).)</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>
See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public override Modifier DeepCopy()
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function DeepCopy As Modifier
```

### Visual C++

```cpp
public:
virtual Modifier^ DeepCopy() override
```

## Return Value

See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

Visual C++

```cpp
public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override
```

Parameters

dt
    Type: `System::.::Single`
    Elapsed time in whole and fractional seconds.

particleArray
    Type: `ProjectMercury::.::Particle` *
    A pointer to an array of particles.

count
    Type: `System::.::Int32`
The number of particles which need to be processed.
See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The PlatformModifier type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platforms</td>
<td>The list of platforms.</td>
</tr>
</tbody>
</table>
See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The list of platforms.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public List<BoundingBox> Platforms { get; set; }

Visual Basic (Declaration)

Public Property Platforms As List(Of BoundingBox)

Visual C++

public:
property List<BoundingBox>^ Platforms {
    List<BoundingBox>^ get ();
    void set (List<BoundingBox>^ value);
}
See Also

PlatformModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which applies a force to a Particle when it enters a circular area.

**Namespace:** [ProjectMercury.Modifiers](#)
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public class RadialForceModifier : Modifier
```

**Visual Basic (Declaration)**

```vbnet
Public Class RadialForceModifier
    Inherits Modifier
```

**Visual C++**

```cpp
public ref class RadialForceModifier : public Modifier
```
Inheritance Hierarchy

System..:::Object
    ProjectMercury.Modifiers..:::Modifier
        ProjectMercury.Modifiers..:::RadialForceModifier
See Also

RadialForceModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RadialForceModifier type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RadialForceModifier</td>
<td>Initializes a new instance of the RadialForceModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;i&gt; DeepCopy&lt;/i&gt;().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;i&gt; Process&lt;/i&gt;(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force</td>
<td>Gets or sets the force vector.</td>
</tr>
<tr>
<td>Position</td>
<td>Gets or sets the position of the force.</td>
</tr>
<tr>
<td>Strength</td>
<td>Gets or sets the strength of the force.</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>Gets or sets the radius of the force.</td>
</tr>
</tbody>
</table>
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `RadialForceModifier` class

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RadialForceModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RadialForceModifier()
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RadialForceModifier** type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Force</strong></td>
<td>Gets or sets the force vector.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Gets or sets the position of the force.</td>
</tr>
<tr>
<td><strong>Strength</strong></td>
<td>Gets or sets the strength of the force.</td>
</tr>
</tbody>
</table>
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the force vector.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public Vector2 Force

Visual Basic (Declaration)

Public Force As Vector2

Visual C++

public:
Vector2 Force
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the position of the force.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public Vector2 Position

**Visual Basic (Declaration)**

Public Position As Vector2

**Visual C++**

public:
Vector2 Position
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the strength of the force.

**Namespace:**  ProjectMercury.Modifiers

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Strength

Visual Basic (Declaration)

Public Strength As Single

Visual C++

public:
float Strength
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RadialForceModifier type exposes the following members.

[This is preliminary documentation and is subject to change.]
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;).</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified &lt;code&gt;Object&lt;/code&gt; is equal to the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an &lt;code&gt;Object&lt;/code&gt; to attempt to free resources and perform other cleanup operations before the &lt;code&gt;Object&lt;/code&gt; is reclaimed by garbage collection. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the &lt;code&gt;Type&lt;/code&gt; of the current instance. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a &lt;code&gt;String&lt;/code&gt; that represents the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
</tbody>
</table>
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
## Syntax

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

## Return Value

See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

generic override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RadialForceModifier** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>Gets or sets the radius of the force.</td>
</tr>
</tbody>
</table>
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the radius of the force.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public float Radius { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property Radius As Single
```

Visual C++

```cpp
public:
property float Radius {
    float get();
    void set(float value);
}
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System:::ArgumentOutOfRangeException</code></td>
<td>Thrown if the specified value is negative or zero</td>
</tr>
</tbody>
</table>
See Also

RadialForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which pulls Particles towards it.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class RadialGravityModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class RadialGravityModifier _
Inherits Modifier

Visual C++

public ref class RadialGravityModifier sealed : public Modifier
Inheritance Hierarchy

System::Object
   ProjectMercury.Modifiers::Modifier
      ProjectMercury.Modifiers::RadialGravityModifier
See Also

RadialGravityModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RadialGravityModifier** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RadialGravityModifier</td>
<td>Initializes a new instance of the RadialGravityModifier class</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::: DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::: Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position</strong></td>
<td>The position of the gravity well.</td>
</tr>
<tr>
<td><strong>Strength</strong></td>
<td>The strength of the gravity well.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerRadius</td>
<td>Gets or sets the inner radius of the gravity well, within which Particles will not be attracted.</td>
</tr>
<tr>
<td>Radius</td>
<td>Gets or sets the radius of the gravity well.</td>
</tr>
</tbody>
</table>
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the RadialGravityModifier class

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RadialGravityModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RadialGravityModifier()
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RadialGravityModifier` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>The position of the gravity well.</td>
</tr>
<tr>
<td>Strength</td>
<td>The strength of the gravity well.</td>
</tr>
</tbody>
</table>
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The position of the gravity well.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public Vector2 Position
```

**Visual Basic (Declaration)**

```vbnet
Public Position As Vector2
```

**Visual C++**

```cpp
public:
Vector2 Position
```
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The strength of the gravity well.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Strength

Visual Basic (Declaration)

Public Strength As Single

Visual C++

public:
float Strength
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RadialGravityModifier` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt; Override:: DeepCopy();).</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>GetHashCode</td>
<td>Serves as a hash function for a particular type.</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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt; Override:: Process(Single, Particle*, Int32);).</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>
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override

Parameters

dt
    Type: System::::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::::Particle *
    A pointer to an array of particles.

count
    Type: System::::Int32
The number of particles which need to be processed.
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RadialGravityModifier type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerRadius</td>
<td>Gets or sets the inner radius of the gravity well, within which Particles will not be attracted.</td>
</tr>
<tr>
<td>Radius</td>
<td>Gets or sets the radius of the gravity well.</td>
</tr>
</tbody>
</table>
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RadialGravityModifier Class

Gets or sets the inner radius of the gravity well, within which Particles will not be attracted.

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float InnerRadius { get; set; }

Visual Basic (Declaration)

Public Property InnerRadius As Single

Visual C++

public:
property float InnerRadius {
    float get ();
    void set (float value);
}
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the radius of the gravity well.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public float Radius { get; set; }
```

**Visual Basic (Declaration)**

Public Property Radius As Single

**Visual C++**

```cpp
public:
    property float Radius {
        float get ();
        void set (float value);
    }
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System:::ArgumentOutOfRangeException</td>
<td>Thrown if the specified value is negative or zero.</td>
</tr>
</tbody>
</table>
See Also

RadialGravityModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which constrains & deflects particles inside a rectangle.

**Namespace**: ProjectMercury.Modifiers  
**Assembly**: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
C#

public sealed class RectangleConstraintDeflector : Modifier

Visual Basic (Declaration)

Public NotInheritable Class RectangleConstraintDeflector _
Inherits Modifier

Visual C++

public ref class RectangleConstraintDeflector sealed : public Modifier
Inheritance Hierarchy

System..:::Object
   ProjectMercury.Modifiers..:::Modifier
      ProjectMercury.Modifiers..:::RectangleConstraintDeflector
See Also

RectangleConstraintDeflector Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RectangleConstraintDeflector` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RectangleConstraintDeflector</td>
<td>Initializes a new instance of the RectangleConstraintDeflector class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;).</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified &lt;code&gt;Object&lt;/code&gt; is equal to the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an &lt;code&gt;Object&lt;/code&gt; to attempt to free resources and perform other cleanup operations before the &lt;code&gt;Object&lt;/code&gt; is reclaimed by garbage collection. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the &lt;code&gt;Type&lt;/code&gt; of the current instance. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.) Processes the particles. (Overrides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;).</td>
</tr>
<tr>
<td>Process</td>
<td>Returns a &lt;code&gt;String&lt;/code&gt; that represents the current &lt;code&gt;Object&lt;/code&gt;. (Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Defines the position of the rectangle boundary constraint.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Gets or sets the height of the rectangle.</td>
</tr>
<tr>
<td>RestitutionCoefficient</td>
<td>Gets or sets the restitution coefficient (bounce factor) of Particles when the hit the deflector.</td>
</tr>
<tr>
<td>Width</td>
<td>Gets or sets the width of the rectangle deflector.</td>
</tr>
</tbody>
</table>
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the RectangleConstraintDeflector class

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RectangleConstraintDeflector()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RectangleConstraintDeflector()
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RectangleConstraintDeflector type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Defines the position of the rectangle boundary constraint.</td>
</tr>
</tbody>
</table>
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the position of the rectangle boundary constraint.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 Position

Visual Basic (Declaration)

Public Position As Vector2

Visual C++

public:
Vector2 Position
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RectangleConstraintDeflector type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</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>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
# Syntax

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
    virtual Modifier^ DeepCopy() override
```

## Return Value

[Missing <returns> documentation for
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RectangleConstraintDeflector type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Height</strong></td>
<td>Gets or sets the height of the rectangle.</td>
</tr>
<tr>
<td><strong>RestitutionCoefficient</strong></td>
<td>Gets or sets the restitution coefficient (bounce factor) of Particles when the hit the deflector.</td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td>Gets or sets the width of the rectangle deflector.</td>
</tr>
</tbody>
</table>
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the height of the rectangle.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
### Syntax

**C#**

```csharp
public float Height { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Height As Single
```

**Visual C++**

```cpp
public:
property float Height {
    float get ();
    void set (float value);
}
```

### Field Value

The height of the rectangle.
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the restitution coefficient (bounce factor) of Particles when the hit the deflector.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public VariableFloat RestitutionCoefficient { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property RestitutionCoefficient As VariableFloat
```

Visual C++

```cpp
public:
property VariableFloat RestitutionCoefficient {
    VariableFloat get ();
    void set (VariableFloat value);
}
```
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the width of the rectangle deflector.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#
public float Width { get; set; }

Visual Basic (Declaration)
Public Property Width As Single

Visual C++
public:
property float Width {
    float get ();
    void set (float value);
}

Field Value

The width of the rectangle deflector.
See Also

RectangleConstraintDeflector Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which applies a force to a Particle when it enters a rectangular area.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class RectangleForceModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class RectangleForceModifier __
    Inherits Modifier

Visual C++

public ref class RectangleForceModifier sealed : public Modifier
Inheritance Hierarchy

System..:::Object
   ProjectMercury.Modifiers..:::Modifier
      ProjectMercury.Modifiers..:::RectangleForceModifier
See Also

RectangleForceModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RectangleForceModifier type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RectangleForceModifier</td>
<td>Initializes a new instance of the RectangleForceModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier:: DeepCopy().)</td>
</tr>
<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. (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>Process</strong></td>
<td>Processes the particles. (Overrides Modifier:: Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force</td>
<td>Gets or sets the force vector.</td>
</tr>
<tr>
<td>Position</td>
<td>Gets or sets the position of the centre of the rectangular force area.</td>
</tr>
<tr>
<td>Strength</td>
<td>Gets or sets the strength of the force.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom</td>
<td>Gets the position of the bottom edge of the rectangle.</td>
</tr>
<tr>
<td>Height</td>
<td>Gets or sets the height of the rectangular force area.</td>
</tr>
<tr>
<td>Left</td>
<td>Gets the position of the left edge of the rectangle.</td>
</tr>
<tr>
<td>Right</td>
<td>Gets the position of the right edge of the rectangle.</td>
</tr>
<tr>
<td>Top</td>
<td>Gets the position of the top edge of the rectangle.</td>
</tr>
<tr>
<td>Width</td>
<td>Gets or sets the width of the rectangular force area.</td>
</tr>
</tbody>
</table>
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the RectangleForceModifier class

**Namespace:**  ProjectMercury.Modifiers

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RectangleForceModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RectangleForceModifier()
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RectangleForceModifier type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force</td>
<td>Gets or sets the force vector.</td>
</tr>
<tr>
<td>Position</td>
<td>Gets or sets the position of the centre of the rectangular force area.</td>
</tr>
<tr>
<td>Strength</td>
<td>Gets or sets the strength of the force.</td>
</tr>
</tbody>
</table>
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++
Project Mercury API Reference
RectangleForceModifier...:.Force Field

RectangleForceModifier Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Gets or sets the force vector.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 Force

Visual Basic (Declaration)

Public Force As Vector2

Visual C++

public:
Vector2 Force
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the position of the centre of the rectangular force area.

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Vector2 Position

Visual Basic (Declaration)

Public Position As Vector2

Visual C++

public:
Vector2 Position
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the strength of the force.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Strength

Visual Basic (Declaration)

Public Strength As Single

Visual C++

public:
float Strength
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RectangleForceModifier type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier:: DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier:: Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RectangleForceModifier DeepCopy Method

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

Returns a deep copy of the Modifier implementation.
## Syntax

### C#

```csharp
public override Modifier DeepCopy()
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function DeepCopy As Modifier
```

### Visual C++

```cpp
public:
virtual Modifier^ DeepCopy() override
```

## Return Value

See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RectangleForceModifier` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom</td>
<td>Gets the position of the bottom edge of the rectangle.</td>
</tr>
<tr>
<td>Height</td>
<td>Gets or sets the height of the rectangular force area.</td>
</tr>
<tr>
<td>Left</td>
<td>Gets the position of the left edge of the rectangle.</td>
</tr>
<tr>
<td>Right</td>
<td>Gets the position of the right edge of the rectangle.</td>
</tr>
<tr>
<td>Top</td>
<td>Gets the position of the top edge of the rectangle.</td>
</tr>
<tr>
<td>Width</td>
<td>Gets or sets the width of the rectangular force area.</td>
</tr>
</tbody>
</table>
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the position of the bottom edge of the rectangle.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Bottom { get; }

Visual Basic (Declaration)

Public ReadOnly Property Bottom As Single

Visual C++

public:
property float Bottom {
    float get ();
}
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the height of the rectangular force area.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll)  
Version: 3.1.0.0
Syntax

C#

public float Height { get; set; }

Visual Basic (Declaration)

Public Property Height As Single

Visual C++

public:
property float Height {
float get ();
void set (float value);
}
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the position of the left edge of the rectangle.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public float Left { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property Left As Single
```

Visual C++

```cpp
public:
property float Left {
    float get ();
}
```
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the position of the right edge of the rectangle.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Right { get; }

Visual Basic (Declaration)

Public ReadOnly Property Right As Single

Visual C++

public:
    property float Right {
        float get();
    }
}
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the position of the top edge of the rectangle.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Top { get; }

Visual Basic (Declaration)

Public ReadOnly Property Top As Single

Visual C++

public:
property float Top {
    float get ();
}

See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the width of the rectangular force area.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Width { get; set; }

Visual Basic (Declaration)

Public Property Width As Single

Visual C++

public:
    property float Width {
        float get();
        void set (float value);
    }
See Also

RectangleForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which alters the rotation of a Particle over its lifetime.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public sealed class RotationModifier : Modifier
```

**Visual Basic (Declaration)**

```vbnet
Public NotInheritable Class RotationModifier _
    Inherits Modifier
```

**Visual C++**

```cpp
public ref class RotationModifier sealed : public Modifier
```
Inheritance Hierarchy

System:::Object
ProjectMercury.Modifiers:::Modifier
  ProjectMercury.Modifiers:::RotationModifier
See Also

RotationModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RotationModifier type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationModifier</td>
<td>Initializes a new instance of the RotationModifier class</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationRate</td>
<td>The rate of rotation in radians per second.</td>
</tr>
</tbody>
</table>
See Also

RotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RotationModifier Constructor

Initializes a new instance of the RotationModifier class

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public RotationModifier()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
RotationModifier()
```
See Also

RotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RotationModifier type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationRate</td>
<td>The rate of rotation in radians per second.</td>
</tr>
</tbody>
</table>
See Also

RotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RotationRate Field

RotationRate Field

[This is preliminary documentation and is subject to change.]

The rate of rotation in radians per second.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float RotationRate

Visual Basic (Declaration)

Public RotationRate As Single

Visual C++

public:
float RotationRate
See Also

RotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RotationModifier type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>Modifier::DeepCopy()</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overides <code>Modifier::Process(Single, Particle*, Int32)</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

RotationModifier Class  
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RotationModifier::DeepCopy Method

RotationModifier Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Returns a deep copy of the Modifier implementation.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

**Return Value**

See Also

RotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

### Parameters

**dt**

Type: `System:::Single`

Elapsed time in whole and fractional seconds.

**particleArray**

Type: `ProjectMercury:::Particle` *

A pointer to an array of particles.

**count**

Type: `System:::Int32`
The number of particles which need to be processed.
See Also

RotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a modifier which changes the rotation rate of particles over their lifetime.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** [ProjectMercury](#) (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class RotationRateModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class RotationRateModifier _
    Inherits Modifier

Visual C++

public ref class RotationRateModifier sealed : public Modifier
Inheritance Hierarchy

System..::..Object
ProjectMercury.Modifiers..::..Modifier
  ProjectMercury.Modifiers..::..RotationRateModifier
See Also

RotationRateModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RotationRateModifier** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationRateModifier</td>
<td>Initializes a new instance of the RotationRateModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt;DeepCopy()&lt;/code&gt;.)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;code&gt;Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinalRate</td>
<td>Gets or sets the final rotation rate in radians per second.</td>
</tr>
<tr>
<td>InitialRate</td>
<td>Gets or sets the initial rotation rate in radians per second.</td>
</tr>
</tbody>
</table>
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the RotationRateModifier class

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RotationRateModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RotationRateModifier()
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The RotationRateModifier type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinalRate</td>
<td>Gets or sets the final rotation rate in radians per second.</td>
</tr>
<tr>
<td>InitialRate</td>
<td>Gets or sets the initial rotation rate in radians per second.</td>
</tr>
</tbody>
</table>
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the final rotation rate in radians per second.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float FinalRate

Visual Basic (Declaration)

Public FinalRate As Single

Visual C++

public:
float FinalRate
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the initial rotation rate in radians per second.

**Namespace:** ProjectMercury.Modifiers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public `float` InitialRate

**Visual Basic (Declaration)**

Public InitialRate As `Single`

**Visual C++**

public:
`float` InitialRate
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RotationRateModifier** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;.)</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;.)</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>
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

#### Visual Basic (Declaration)

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

#### Visual C++

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

### Parameters

**dt**

Type: `System::::Single`

Elapsed time in whole and fractional seconds.

**particleArray**

Type: `ProjectMercury::::Particle` *

A pointer to an array of particles.

**count**

Type: `System::::Int32`
The number of particles which need to be processed.
See Also

RotationRateModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a modifier which changes the scale of particles based on a linear interpolation over three values.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public class ScaleInterpolatorModifier : Modifier
```

**Visual Basic (Declaration)**

```vbnet
Public Class ScaleInterpolatorModifier
    Inherits Modifier
```

**Visual C++**

```cpp
public ref class ScaleInterpolatorModifier : public Modifier
```
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
    ProjectMercury.Modifiers::ScaleInterpolatorModifier
See Also

ScaleInterpolatorModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **ScaleInterpolatorModifier** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScaleInterpolatorModifier</td>
<td>Initializes a new instance of the ScaleInterpolatorModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;br/&gt;DeepCopy().)</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::&lt;br/&gt;Process(Single, Particle*, Int32).)</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>FinalScale</td>
<td>Gets or sets the final scale.</td>
</tr>
<tr>
<td>InitialScale</td>
<td>Gets or sets the initial scale.</td>
</tr>
<tr>
<td>MiddlePosition</td>
<td>Gets or sets the middle scale position.</td>
</tr>
<tr>
<td>MiddleScale</td>
<td>Gets or sets the middle scale.</td>
</tr>
</tbody>
</table>
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ScaleInterpolatorModifier` class

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public ScaleInterpolatorModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ScaleInterpolatorModifier()
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleInterpolatorModifier` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;).</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrrides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;).</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>
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

### Return Value

See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

#### C#

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

#### Visual Basic (Declaration)

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

#### Visual C++

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

### Parameters

**dt**

Type: `System::::Single`

Elapsed time in whole and fractional seconds.

**particleArray**

Type: `ProjectMercury::::Particle` *

A pointer to an array of particles.

**count**

Type: `System::::Int32`
The number of particles which need to be processed.
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleInterpolatorModifier` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FinalScale</strong></td>
<td>Gets or sets the final scale.</td>
</tr>
<tr>
<td><strong>InitialScale</strong></td>
<td>Gets or sets the initial scale.</td>
</tr>
<tr>
<td><strong>MiddlePosition</strong></td>
<td>Gets or sets the middle scale position.</td>
</tr>
<tr>
<td><strong>MiddleScale</strong></td>
<td>Gets or sets the middle scale.</td>
</tr>
</tbody>
</table>
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the final scale.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public float FinalScale { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property FinalScale As Single
```

### Visual C++

```cpp
public:
    property float FinalScale {
        float get ();
        void set (float value);
    }
```
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the initial scale.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float InitialScale { get; set; }

Visual Basic (Declaration)

Public Property InitialScale As Single

Visual C++

public:
property float InitialScale {
    float get ();
    void set (float value);
}
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the middle scale position.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MiddlePosition { get; set; }

Visual Basic (Declaration)

Public Property MiddlePosition As Single

Visual C++

public:
property float MiddlePosition {
    float get ();
    void set (float value);
}
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the middle scale.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public float MiddleScale { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property MiddleScale As Single
```

**Visual C++**

```cpp
public:
property float MiddleScale {
    float get ();
    void set (float value);
}
```
See Also

ScaleInterpolatorModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which merges the scale of particles towards a single scale over their lifetime. Works best when Particles are being released with random scale, where you require the particles to have a uniform scale at the end of their lifetime.

**Namespace:**  [ProjectMercury.Modifiers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class ScaleMergeModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class ScaleMergeModifier _
    Inherits Modifier

Visual C++

public ref class ScaleMergeModifier sealed : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
   ProjectMercury.Modifiers::ScaleMergeModifier
See Also

ScaleMergeModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleMergeModifier` type exposes the following members.
Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScaleMergeModifier</td>
<td>Initializes a new instance of the ScaleMergeModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier..::.DeepCopy().()().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier..::.Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>MergeScale</td>
<td>Gets or sets the final scale of Particles when they are retired.</td>
</tr>
</tbody>
</table>
See Also

ScaleMergeModifier Class  
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the ScaleMergeModifier class

**Namespace:** ProjectMercury.Modifiers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
 Syntax

**C#**

```csharp
public ScaleMergeModifier()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
ScaleMergeModifier()
```
See Also

ScaleMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ScaleMergeModifier type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier:: DeepCopy().)</td>
</tr>
<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. (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>Process</strong></td>
<td>Processes the particles. (Overrides Modifier:: Process(Single, Particle*, Int32).)</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

ScaleMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

### Return Value

See Also

ScaleMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

### Parameters

**dt**
- Type: **System::::Single**
- Elapsed time in whole and fractional seconds.

**particleArray**
- Type: **ProjectMercury::::Particle** *
- A pointer to an array of particles.

**count**
- Type: **System::::Int32**
The number of particles which need to be processed.
See Also

ScaleMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleMergeModifier` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MergeScale</td>
<td>Gets or sets the final scale of Particles when they are retired.</td>
</tr>
</tbody>
</table>
See Also

ScaleMergeModifier Class  
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the final scale of Particles when they are retired.

**Namespace:** [ProjectMercury.Modifiers](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MergeScale { get; set; }

Visual Basic (Declaration)

Public Property MergeScale As Single

Visual C++

public:
property float MergeScale {
    float get ();
    void set (float value);
}

Field Value

The merge scale.
See Also

ScaleMergeModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which adjusts the scale of a Particle over its lifetime.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public class ScaleModifier : Modifier

**Visual Basic (Declaration)**

Public Class ScaleModifier _
    Inherits Modifier

**Visual C++**

public ref class ScaleModifier : public Modifier
Inheritance Hierarchy

System:::Object
ProjectMercury.Modifiers:::Modifier
ProjectMercury.Modifiers:::ScaleModifier
See Also

ScaleModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScaleModifier</td>
<td>Initializes a new instance of the ScaleModifier class</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier:: DeepCopy())</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier:: Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InitialScale</code></td>
<td>The initial scale of the Particle in pixels.</td>
</tr>
<tr>
<td><code>UltimateScale</code></td>
<td>The ultimate scale of the Particle in pixels.</td>
</tr>
</tbody>
</table>
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ScaleModifier` class

**Namespace:**  `ProjectMercury.Modifiers`

**Assembly:**  `ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0`
Syntax

**C#**

```csharp
public ScaleModifier()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
ScaleModifier()
```
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleModifier` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitialScale</td>
<td>The initial scale of the Particle in pixels.</td>
</tr>
<tr>
<td>UltimateScale</td>
<td>The ultimate scale of the Particle in pixels.</td>
</tr>
</tbody>
</table>
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The initial scale of the Particle in pixels.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0

[This is preliminary documentation and is subject to change.]
Syntax

C#

public float InitialScale

Visual Basic (Declaration)

Public InitialScale As Single

Visual C++

public:
float InitialScale
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The ultimate scale of the Particle in pixels.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public float UltimateScale

Visual Basic (Declaration)

Public UltimateScale As Single

Visual C++

public:
float UltimateScale
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleModifier` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrrides Modifier::&lt;code&gt;Data.DeepCopy()&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified &lt;code&gt;Object&lt;/code&gt; is equal to the current &lt;code&gt;Object&lt;/code&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an &lt;code&gt;Object&lt;/code&gt; to attempt to free resources and perform other cleanup operations before the &lt;code&gt;Object&lt;/code&gt; is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the &lt;code&gt;Type&lt;/code&gt; of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current &lt;code&gt;Object&lt;/code&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrrides Modifier::&lt;code&gt;Data.Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a &lt;code&gt;String&lt;/code&gt; that represents the current &lt;code&gt;Object&lt;/code&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from &lt;code&gt;Object&lt;/code&gt;.)</td>
</tr>
</tbody>
</table>
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
    virtual Modifier^ DeepCopy() override
```

**Return Value**

See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
    virtual void Process(
        float dt,
        Particle* particleArray,
        int count
    ) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

ScaleModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which adjusts the scale of Particles based on a sine wave.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class ScaleOscillator : Modifier

Visual Basic (Declaration)

Public Class ScaleOscillator _
  Inherits Modifier

Visual C++

public ref class ScaleOscillator : public Modifier
Inheritance Hierarchy

System::Object
ProjectMercury.Modifiers::Modifier
  ProjectMercury.Modifiers::::ScaleOscillator
See Also

ScaleOscillator Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **ScaleOscillator** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScaleOscillator</td>
<td>Initializes a new instance of the ScaleOscillator class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;DeepCopy&gt;().)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;Process(Single, Particle*, Int32).)</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>Frequency</strong></td>
<td>Gets or sets the oscillator frequency (the number of cycles per second).</td>
</tr>
<tr>
<td><strong>MaximumScale</strong></td>
<td>Gets or sets the maximum scale (the scale of Particles at the positive peak of the sine wave).</td>
</tr>
<tr>
<td><strong>MinimumScale</strong></td>
<td>Gets or sets the minimum scale (the scale of Particles at the negative peak of the sine wave).</td>
</tr>
</tbody>
</table>
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `ScaleOscillator` class

**Namespace:**  `ProjectMercury.Modifiers`

**Assembly:**  `ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0`
Syntax

C#

public ScaleOscillator()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ScaleOscillator()
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `ScaleOscillator` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::&lt;code&gt; DeepCopy()&lt;/code&gt;.)</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. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::&lt;code&gt; Process(Single, Particle*, Int32)&lt;/code&gt;.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** [ProjectMercury.Modifiers](#)

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

**Return Value**

See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override

Parameters

dt
    Type: System::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury::Particle *
    A pointer to an array of particles.

count
    Type: System::Int32
The number of particles which need to be processed.
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **ScaleOscillator** type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency</strong></td>
<td>Gets or sets the oscillator frequency (the number of cycles per second).</td>
</tr>
<tr>
<td><strong>MaximumScale</strong></td>
<td>Gets or sets the maximum scale (the scale of Particles at the positive peak of the sine wave).</td>
</tr>
<tr>
<td><strong>MinimumScale</strong></td>
<td>Gets or sets the minimum scale (the scale of Particles at the negative peak of the sine wave).</td>
</tr>
</tbody>
</table>
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the oscillator frequency (the number of cycles per second).

**Namespace:**  [ProjectMercury.Modifiers](#)

**Assembly:**  [ProjectMercury.dll](#) Version: 3.1.0.0
Syntax

C#

class MyClass
{
    public float Frequency { get; set; }
}

Visual Basic (Declaration)

Public Property Frequency As Single

Visual C++

public:
    property float Frequency
    {
        float get();
        void set(float value);
    }
}
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the maximum scale (the scale of Particles at the positive peak of the sine wave).

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MaximumScale { get; set; }

Visual Basic (Declaration)

Public Property MaximumScale As Single

Visual C++

public:
property float MaximumScale {
    float get ();
    void set (float value);
}

Field Value

The maximum scale.
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the minimum scale (the scale of Particles at the negative peak of the sine wave).

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public float MinimumScale { get; set; }

Visual Basic (Declaration)

Public Property MinimumScale As Single

Visual C++

public:
property float MinimumScale {
    float get ();
    void set (float value);
}

Field Value

The minimum scale.
See Also

ScaleOscillator Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which applies a sine wave force to a Particle over the course of its lifetime.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class SineForceModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class SineForceModifier _
   Inherits Modifier

Visual C++

public ref class SineForceModifier sealed : public Modifier
Inheritance Hierarchy

System:::Object
ProjectMercury.Modifiers:::Modifier
    ProjectMercury.Modifiers:::.SineForceModifier
See Also

SineForceModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `SineForceModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SineForceModifier</td>
<td>Initializes a new instance of the <strong>SineForceModifier</strong> class</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::DeepCopy())</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. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (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>Process</td>
<td>Processes the particles. (Overrides Modifier::Process(Single, Particle*, Int32).)</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>Amplitude</td>
<td>Gets or sets the amplitude of the sine wave.</td>
</tr>
<tr>
<td>Frequency</td>
<td>Gets or sets the frequency of the sine wave.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation</td>
<td>Gets or sets the rotation of the sine force.</td>
</tr>
</tbody>
</table>
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `SineForceModifier` class.

**Namespace:**  [ProjectMercury.Modifiers](https://example.com/ProjectMercury.Modifiers)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

public SineForceModifier()

**Visual Basic (Declaration)**

Public Sub New

**Visual C++**

public:
SineForceModifier()
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **SineForceModifier** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplitude</td>
<td>Gets or sets the amplitude of the sine wave.</td>
</tr>
<tr>
<td>Frequency</td>
<td>Gets or sets the frequency of the sine wave.</td>
</tr>
</tbody>
</table>
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the amplitude of the sine wave.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public float Amplitude
```

**Visual Basic (Declaration)**

```vbnet
Public Amplitude As Single
```

**Visual C++**

```cpp
public:
float Amplitude
```
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
SineForceModifier.Frequency Field

Gets or sets the frequency of the sine wave.

Namespace: ProjectMercury.Modifiers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float Frequency

Visual Basic (Declaration)

Public Frequency As Single

Visual C++

public:
float Frequency
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `SineForceModifier` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</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</td>
</tr>
<tr>
<td></td>
<td>operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public override Modifier DeepCopy()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function DeepCopy As Modifier
```

**Visual C++**

```cpp
public:
virtual Modifier^ DeepCopy() override
```

**Return Value**

See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

Namespace:  ProjectMercury.Modifiers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

**Visual C++**

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

### Parameters

**dt**

Type: `System::::Single`

Elapsed time in whole and fractional seconds.

**particleArray**

Type: `ProjectMercury::::Particle *`

A pointer to an array of particles.

**count**

Type: `System::::Int32`
The number of particles which need to be processed.
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The SineForceModifier type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation</td>
<td>Gets or sets the rotation of the sine force.</td>
</tr>
</tbody>
</table>
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the rotation of the sine force.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public float Rotation { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Rotation As Single
```

**Visual C++**

```csharp
public:
property float Rotation {
    float get ();
    void set (float value);
}
```

### Field Value

The rotation angle in radians.
See Also

SineForceModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which adjusts the rotation of a Particle to follow its trajectory.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class TrajectoryRotationModifier : Modifier

Visual Basic (Declaration)

Public NotInheritable Class TrajectoryRotationModifier
    Inherits Modifier

Visual C++

public ref class TrajectoryRotationModifier sealed : public Modifier
Remarks

Ideally this modifier should be added *after* any other physics modifiers.
Inheritance Hierarchy

System..:::Object
   ProjectMercury.Modifiers..:::Modifier
      ProjectMercury.Modifiers..:::TrajectoryRotationModifier
The `TrajectoryRotationModifier` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrajectoryRotationModifier</td>
<td>Initializes a new instance of the TrajectoryRotationModifier class</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DeepCopy</strong></td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier..::DeepCopy())</td>
</tr>
<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 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>(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>Process</strong></td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier..::Process(Single, Particle*, Int32).)</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>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationOffset</td>
<td>The rotation offset to add to the calculated trajectory rotation.</td>
</tr>
</tbody>
</table>
See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `TrajectoryRotationModifier` class

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public TrajectoryRotationModifier()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```c++
public:
TrajectoryRotationModifier()
```
See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TrajectoryRotationModifier` type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationOffset</td>
<td>The rotation offset to add to the calculated trajectory rotation.</td>
</tr>
</tbody>
</table>
See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The rotation offset to add to the calculated trajectory rotation.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
Version: 3.1.0.0
Syntax

C#

public float RotationOffset

Visual Basic (Declaration)

Public RotationOffset As Single

Visual C++

public:
float RotationOffset
See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `TrajectoryRotationModifier` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation. (Overrides Modifier::::DeepCopy().)</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. (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>Process</td>
<td>Processes the particles. (Overides Modifier::::Process(Single, Particle*, Int32).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:**  [ProjectMercury.Modifiers](#)
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public override void Process(
    float dt,
    Particle* particleArray,
    int count
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub Process ( _
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)
```

Visual C++

```cpp
public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override
```

Parameters

dt
Type: `System::::Single`
Elapsed time in whole and fractional seconds.

particleArray
Type: `ProjectMercury::::Particle` *
A pointer to an array of particles.

count
Type: `System::::Int32`
The number of particles which need to be processed.
See Also

TrajectoryRotationModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Modifier which limits the velocity of Particles to a specified value.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

public class VelocityClampModifier : Modifier

**Visual Basic (Declaration)**

Public Class VelocityClampModifier
    Inherits Modifier

**Visual C++**

public ref class VelocityClampModifier : public Modifier
Remarks

For best results insert this Modifier after any other Modifiers which may alter the velocity of the Particle.
Inheritance Hierarchy

System..::.Object

ProjectMercury.Modifiers..::.Modifier

ProjectMercury.Modifiers..::.VelocityClampModifier
See Also

VelocityClampModifier Members
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The VelocityClampModifier type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VelocityClampModifier</td>
<td>Initializes a new instance of the VelocityClampModifier class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepCopy</td>
<td>Returns a deep copy of the Modifier implementation.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::__::DeepCopy().)</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>(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>Process</td>
<td>Processes the particles.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Modifier::__::Process(Single, Particle*, Int32).)</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>MaximumVelocity</td>
<td>Gets or sets the maximum velocity of Particles.</td>
</tr>
</tbody>
</table>
See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `VelocityClampModifier` class

**Namespace:**  `ProjectMercury.Modifiers`

**Assembly:**  `ProjectMercury` (in `ProjectMercury.dll`) Version: 3.1.0.0
Syntax

C#

public VelocityClampModifier()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
VelocityClampModifier()
See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `VelocityClampModifier` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| DeepCopy        | Returns a deep copy of the Modifier implementation.  
                        (Overrider Modifier::: DeepCopy().) |
| 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.  
                        (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.) |
| Process         | Processes the particles.  
                        (Overrider Modifier::: Process(Single, Particle*, Int32).) |
| ToString        | Returns a String that represents the current Object.  
                        (Inherited from Object.) |
See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Returns a deep copy of the Modifier implementation.

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

public override Modifier DeepCopy()

Visual Basic (Declaration)

Public Overrides Function DeepCopy As Modifier

Visual C++

public:
virtual Modifier^ DeepCopy() override

Return Value

See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Processes the particles.

**Namespace:** ProjectMercury.Modifiers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void Process(
    float dt,
    Particle* particleArray,
    int count
)

Visual Basic (Declaration)

Public Overrides Sub Process (_
    dt As Single, _
    particleArray As Particle*, _
    count As Integer _
)

Visual C++

public:
virtual void Process(
    float dt,
    Particle* particleArray,
    int count
) override

Parameters

dt
    Type: System:::Single
    Elapsed time in whole and fractional seconds.

particleArray
    Type: ProjectMercury:::Particle *
    A pointer to an array of particles.

count
    Type: System:::Int32
The number of particles which need to be processed.
See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **VelocityClampModifier** type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaximumVelocity</td>
<td>Gets or sets the maximum velocity of Particles..</td>
</tr>
</tbody>
</table>
See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the maximum velocity of Particles..

**Namespace:**  ProjectMercury.Modifiers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public float MaximumVelocity { get; set; }

Visual Basic (Declaration)

Public Property MaximumVelocity As Single

Visual C++

public:
property float MaximumVelocity {
    float get ();
    void set (float value);
}

Field Value

The maximum velocity of Particles..
See Also

VelocityClampModifier Class
ProjectMercury.Modifiers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
[This is preliminary documentation and is subject to change.]
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlendModeNotSuportedException</td>
<td>Defines the exception that is thrown when a Renderer does not support the blend mode required by an Emitter.</td>
</tr>
<tr>
<td>PointSpriteRenderer</td>
<td>Defines a Renderer which uses hardware point sprites to render Particles.</td>
</tr>
<tr>
<td>Renderer</td>
<td>Defines the abstract base class for a Renderer.</td>
</tr>
<tr>
<td>RenderTargetRenderer</td>
<td>Defines a renderer which renders to a RenderTarget using a specified renderer.</td>
</tr>
<tr>
<td>SpriteBatchRenderer</td>
<td>Defines a Renderer which uses the standard XNA SpriteBatch class to render Particles.</td>
</tr>
</tbody>
</table>

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the exception that is thrown when a Renderer does not support the blend mode required by an Emitter.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public class BlendModeNotSupportedException : ApplicationException

Visual Basic (Declaration)

Public Class BlendModeNotSupportedException _
   Inherits ApplicationException

Visual C++

public ref class BlendModeNotSupportedException : public ApplicationException
Inheritance Hierarchy

System::Object
System::Exception
System::ApplicationException
ProjectMercury.Renderers::BlendModeNotSupportedException
See Also

BlendModeNotSupportedException Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `BlendModeNotSuportedException` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BlendModeNotSupportedException</code></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 <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception.         (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></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><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></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><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

BlendModeNotSupportedException Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
BlendModeNotSupportedException Constructor

BlendModeNotSupportedException Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![BlendModeNotSuportedException()()()]</td>
<td>Initializes a new instance of the BlendModeNotSuportedException class.</td>
</tr>
<tr>
<td>![BlendModeNotSuportedException(String)]</td>
<td>Initializes a new instance of the BlendModeNotSuportedException class.</td>
</tr>
</tbody>
</table>
See Also

BlendModeNotSupportedException Class
BlendModeNotSupportedException Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `BlendModeNotSupportedException` class.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public BlendModeNotSupportedException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
BlendModeNotSupportedException()
See Also

BlendModeNotSupportedException Class
BlendModeNotSupportedException Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
BlendModeNotSupportedException Constructor (String)

 Initializes a new instance of the BlendModeNotSupportedException class.

 **Namespace:** ProjectMercury.Renderers  
 **Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public BlendModeNotSupportedException(
    string message
)
```

Visual Basic (Declaration)

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

Visual C++

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

Parameters

message
Type: System::String
The exception message.
See Also

BlendModeNotSupportedException Class
BlendModeNotSupportedException Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `BlendModeNotSupportedException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Gets the runtime type of the current instance.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <a href="#">Object</a>). (Inherited from <a href="#">Object</a>). (Inherited from <a href="#">Exception</a>).</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>(Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>(Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>(Inherited from <a href="#">Exception</a>).</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from <a href="#">Exception</a>).</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>(Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <a href="#">Exception</a>).</td>
</tr>
</tbody>
</table>
See Also

BlendModeNotSupportedException Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `BlendModeNotSuportedException` type exposes the following members.
## 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

BlendModeNotSupportedException Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Renderer which uses hardware point sprites to render Particles.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public sealed class PointSpriteRenderer : Renderer

Visual Basic (Declaration)

Public NotInheritable Class PointSpriteRenderer _
Inherits Renderer

Visual C++

public ref class PointSpriteRenderer sealed : public Renderer
Inheritance Hierarchy

System::Object
ProjectMercury.Renderers::Renderer
ProjectMercury.Renderers::PointSpriteRenderer
See Also

PointSpriteRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PointSpriteRenderer** type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointSpriteRenderer</td>
<td>Initializes a new instance of the PointSpriteRenderer class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Dispose</code></td>
<td>Overloaded. Determine whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>Equals</code></td>
<td>(Inherited from <code>Object</code>.) Releasing unmanaged resources and performs other cleanup operations before the <code>Renderer</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>(Inherited from <code>Renderer</code>.) Releases unmanaged resources and performs other cleanup operations before the <code>Renderer</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>(Inherited from <code>Object</code>.) Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>(Inherited from <code>Object</code>.) Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>LoadContent</code></td>
<td>(Overrides <code>Renderer</code>.) Loads any content that is needed by the renderer.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>(Inherited from <code>Object</code>.) Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>RenderEffect</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>RenderEmitter</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>SetRenderState</code></td>
<td>Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>(Inherited from <code>Object</code>.) Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GraphicsDeviceService</code></td>
<td>Hold a reference to the games GraphicsDeviceService. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td><code>ShaderContentPath</code></td>
<td>The path to the PointSprite.fx shader in your content project.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableRotatedPointSprites</td>
<td>Gets or sets a value indicating whether rotated point sprites are enabled.</td>
</tr>
<tr>
<td></td>
<td>Gets or sets the maximum size of point sprites on the GPU. Increase this</td>
</tr>
<tr>
<td></td>
<td>value if you find your particles are not scaling correctly.</td>
</tr>
<tr>
<td>MaximumPointSpriteSize</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `PointSpriteRenderer` class.

**Namespace:**  `ProjectMercury.Renderers`  
**Assembly:**  `ProjectMercury` (in `ProjectMercury.dll`) Version: 3.1.0.0
Syntax

C#

public PointSpriteRenderer()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
PointSpriteRenderer()
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PointSpriteRenderer** type exposes the following members.

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphicsDeviceService</td>
<td>Hold a reference to the games GraphicsDeviceService. (Inherited from Renderer.)</td>
</tr>
<tr>
<td>ShaderContentPath</td>
<td>The path to the PointSprite.fx shader in your content project.</td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The path to the PointSprite.fx shader in your content project.

**Namespace:** ProjectMercury.Renderers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public string ShaderContentPath

Visual Basic (Declaration)

Public ShaderContentPath As String

Visual C++

public:
String^ ShaderContentPath
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The PointSpriteRenderer type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.</td>
</tr>
<tr>
<td>Equals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Releases unmanaged resources and performs other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/xna.graphics.irenderer">Renderer</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/xna.graphics.irenderer">Renderer</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>LoadContent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loads any content that is needed by the renderer.</td>
</tr>
<tr>
<td></td>
<td>(Overides <a href="https://docs.microsoft.com/en-us/dotnet/api/xna.graphics.renderer.loadcontent">Renderer.::.LoadContent(ContentManager)</a> )</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>RenderEffect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/xna.graphics.irenderer">Renderer</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
PointSpriteRenderer...::Dispose Method

PointSpriteRenderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()</td>
<td>Disposes any unmanaged resources being used by this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Renderer</a>.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Renderer</a>.)</td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
PointSpriteRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Loads any content that is needed by the renderer.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public override void LoadContent(
    ContentManager content
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub LoadContent (_
    content As ContentManager _
)
```

**Visual C++**

```cpp
public:
virtual void LoadContent(
    ContentManager^ content
) override
```

Parameters

ccontent
  Type: ContentManager

[Missing <param name="content"/> documentation for
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System..::.InvalidOperationException</code></td>
<td>Thrown if the GraphicsDeviceService has not been set.</td>
</tr>
<tr>
<td><code>ContentLoadException</code></td>
<td>Thrown if the content item defined in the ShaderContentPath property could not be loaded.</td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members

Project Mercury API Reference
PointSpriteRenderer:::RenderEffect Method

PointSpriteRenderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderEffect(ParticleEffect)</td>
<td>Renders the specified ParticleEffect. (Inherited from <a href="#">Renderer</a>.)</td>
</tr>
<tr>
<td>RenderEffect(ParticleEffect, Matrix%)</td>
<td>Renders the specified ParticleEffect, applying the specified transformation offset. (Inherited from <a href="#">Renderer</a>.)</td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
PointSpriteRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderEmitter(Emitter)</td>
<td>Renders the specified Emitter. (Inherited from Renderer.)</td>
</tr>
<tr>
<td>RenderEmitter(Emitter,</td>
<td>...</td>
</tr>
<tr>
<td>Matrix%)</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>(Overrides Renderer::RenderEmitter(Emitter, Matrix%).)</td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
PointSpriteRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified Emitter, applying the specified transformation offset.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public override void RenderEmitter(
    Emitter emitter,
    ref Matrix transform
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub RenderEmitter (_
    emitter As Emitter, _
    ByRef transform As Matrix _
)
```

Visual C++

```cpp
public:
virtual void RenderEmitter(
    Emitter^ emitter,
    Matrix% transform
) override
```

Parameters


Type: Matrix %

```cpp
transform
```

See Also

PointSpriteRenderer Class
RenderEmitter Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **PointSpriteRenderer** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableRotatedPointSprites</td>
<td>Gets or sets a value indicating whether rotated point sprites are enabled.</td>
</tr>
<tr>
<td></td>
<td>Gets or sets the maximum size of point sprites on the GPU. Increase this value if you find your particles are not scaling correctly.</td>
</tr>
<tr>
<td>MaximumPointSpriteSize</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets a value indicating whether rotated point sprites are enabled.

**Namespace:** [ProjectMercury.Renderers](#)  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public bool EnableRotatedPointSprites { get; set; }

Visual Basic (Declaration)

Public Property EnableRotatedPointSprites As Boolean

Visual C++

public:
property bool EnableRotatedPointSprites {
    bool get();
    void set (bool value);
}

Field Value

true if rotated point sprites are enabled; otherwise, false.
Remarks

Enabling rotated point sprites requires shader model 2.0 on the target machine.
See Also

PointSpriteRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the maximum size of point sprites on the GPU. Increase this value if you find your particles are not scaling correctly.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
```csharp
public float MaximumPointSpriteSize { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property MaximumPointSpriteSize As Single
```

**Visual C++**

```cpp
public:
property float MaximumPointSpriteSize {
    float get ();
    void set (float value);
}
```
See Also

**PointSpriteRenderer Class**
**ProjectMercury.Renderers Namespace**

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines the abstract base class for a Renderer.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

class Renderer : IDisposable

Visual Basic (Declaration)

Public MustInherit Class Renderer _
Implements IDisposable

Visual C++

public ref class Renderer abstract : IDisposable
Inheritance Hierarchy

System::Object
ProjectMercury.Renderers::Renderer
  ProjectMercury.Renderers::PointSpriteRenderer
  ProjectMercury.Renderers::RenderTargetRenderer
  ProjectMercury.Renderers::SpriteBatchRenderer
See Also

Renderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **Renderer** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renderer</td>
<td>Initializes a new instance of the <em>Renderer</em> class</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Equals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://learn.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Releases unmanaged resources and performs other cleanup operations before the <a href="https://learn.microsoft.com/en-us/dotnet/api/system.renderer">Renderer</a> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <a href="https://learn.microsoft.com/en-us/dotnet/api/system.object">Object</a>.::Finalize());</td>
</tr>
<tr>
<td>GetHashCode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://learn.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gets the <a href="https://learn.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://learn.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>LoadContent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loads any content needed by the Renderer.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://learn.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>RenderEffect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://learn.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `Renderer` class

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

protected Renderer()

Visual Basic (Declaration)

Protected Sub New

Visual C++

protected:
Renderer()
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Renderer type exposes the following members.

[This is preliminary documentation and is subject to change.]
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
[This is preliminary documentation and is subject to change.]

Hold a reference to the games GraphicsDeviceService.

**Namespace:**  [ProjectMercury.Renderers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public IGraphicsDeviceService GraphicsDeviceService

Visual Basic (Declaration)

Public GraphicsDeviceService As IGraphicsDeviceService

Visual C++

public:
IGraphicsDeviceService^ GraphicsDeviceService
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The Renderer type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td></td>
<td>Releases unmanaged resources and performs other cleanup operations before the <a href="#">Renderer</a> is reclaimed by garbage collection. (Overrides <a href="#">Object::Finalize()</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Loads any content needed by the Renderer.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEffect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td>Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renderer...::..Dispose Method

Renderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()</td>
<td>Disposes any unmanaged resources being used by this instance.</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources</td>
</tr>
</tbody>
</table>
See Also

Renderer Class
Renderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Disposes any unmanaged resources being used by this instance.

**Namespace:**  [ProjectMercury.Renderers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public void Dispose()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Dispose()
```

**Visual C++**

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

**Implements**

`IDisposable::Dispose()`
See Also

Renderer Class
Dispose Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
namespace: ProjectMercury.Renderers
assembly: ProjectMercury (in ProjectMercury.dll) version: 3.1.0.0
Syntax

C#

protected virtual void Dispose(
    bool disposing
)

Visual Basic (Declaration)

Protected Overridable Sub Dispose ( _
    disposing As Boolean _
)

Visual C++

protected:
virtual void Dispose(
    bool disposing
)

Parameters

disposing

Type: System::Boolean
true to release both managed and unmanaged resources; false to release only unmanaged resources.
See Also

Renderer Class
Dispose Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Finalize Method

Renderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Releases unmanaged resources and performs other cleanup operations before the Renderer is reclaimed by garbage collection.

Namespace:  ProjectMercury.Renderers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```
protected override void Finalize()
```

**Visual Basic (Declaration)**

```
Protected Overrides Sub Finalize
```

**Visual C++**

```
protected:
virtual void Finalize() override
```
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Loads any content needed by the Renderer.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public virtual void LoadContent(
    ContentManager content
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overridable Sub LoadContent ( _
    content As ContentManager _
)
```

**Visual C++**

```cpp
public:
    virtual void LoadContent(
        ContentManager^ content
    )
```

**Parameters**

- **content**
  - Type: ContentManager

See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference
Renderer...:..:RenderEffect Method

Renderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderEffect(ParticleEffect)</td>
<td>Renders the specified ParticleEffect.</td>
</tr>
<tr>
<td>RenderEffect(ParticleEffect, Matrix%)</td>
<td>Renders the specified ParticleEffect, applying the specified transformation offset.</td>
</tr>
</tbody>
</table>
See Also

Renderer Class
Renderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified ParticleEffect.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll)  
**Version:** 3.1.0.0
### Syntax

**C#**

```csharp
public virtual void RenderEffect(
    ParticleEffect effect
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overridable Sub RenderEffect ( _
    effect As ParticleEffect _
)
```

**Visual C++**

```cpp
public:
    virtual void RenderEffect(
        ParticleEffect^ effect
    )
```

### Parameters

**effect**

Type: `ProjectMercury::ParticleEffect`

See Also

Renderer Class
RenderEffect Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified ParticleEffect, applying the specified transformation offset.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
Syntax

C#

```csharp
public virtual void RenderEffect(
    ParticleEffect effect,
    ref Matrix transform
)
```

Visual Basic (Declaration)

```vbnet
Public Overridable Sub RenderEffect ( _
    effect As ParticleEffect, _
    ByRef transform As Matrix _
)
```

Visual C++

```cpp
public:
virtual void RenderEffect(
    ParticleEffect^ effect,
    Matrix% transform
)
```

Parameters

effect
Type: ProjectMercury::::ParticleEffect

transform
Type: Matrix %
See Also

Renderer Class
RenderEffect Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference
Renderer...::RenderEmitter Method

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderEmitter(Emitter)</td>
<td>Renders the specified Emitter.</td>
</tr>
<tr>
<td>RenderEmitter(Emitter, Matrix%)</td>
<td>Renders the specified Emitter, applying the specified transformation offset.</td>
</tr>
</tbody>
</table>
See Also

Renderer Class
Renderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified Emitter.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public virtual void RenderEmitter(
    Emitter emitter
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overridable Sub RenderEmitter (_
    emitter As Emitter _
)
```

**Visual C++**

```cpp
public:
virtual void RenderEmitter(
    Emitter^ emitter
)
```

Parameters

**emitter**

Type: `ProjectMercury.Emitters..::.Emitter`

See Also

Renderer Class
RenderEmitter Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified Emitter, applying the specified transformation offset.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
## Syntax

### C#

```csharp
public abstract void RenderEmitter(
    Emitter emitter,
    ref Matrix transform
)
```

### Visual Basic (Declaration)

```vbnet
Public MustOverride Sub RenderEmitter ( _
    emitter As Emitter, _
    ByRef transform As Matrix _
)
```

### Visual C++

```cpp
public:
    virtual void RenderEmitter(
        Emitter^ emitter,
        Matrix% transform
    ) abstract
```

## Parameters

**emitter**
- Type: `ProjectMercury.Emitters..::.Emitter`


**transform**
- Type: `Matrix %`


See Also

Renderer Class
RenderEmitter Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Sets the render state of the graphics device before rendering an Emitter.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

protected virtual void SetRenderState(
    Emitter emitter
)

Visual Basic (Declaration)

Protected Overridable Sub SetRenderState ( _ 
    emitter As Emitter _
)

Visual C++

protected:
virtual void SetRenderState( 
    Emitter^ emitter 
)

Parameters

emitter
    Type: ProjectMercury.Emitters::::Emitter
    The emitter which is about to be rendered.
See Also

Renderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a renderer which renders to a RenderTarget using a specified renderer.

Namespace:  ProjectMercury.Renderers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public sealed class RenderTargetRenderer : Renderer
```

**Visual Basic (Declaration)**

```vbnet
Public NotInheritable Class RenderTargetRenderer
    Inherits Renderer
```

**Visual C++**

```cpp
public ref class RenderTargetRenderer sealed : public Renderer
```
Inheritance Hierarchy

System::Object

ProjectMercury.Renderers::Renderer
  ProjectMercury.Renderers::RenderTargetRenderer
See Also

RenderTargetRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RenderTargetRenderer` type exposes the following members.
## Constructors

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Releases unmanaged resources and performs other cleanup operations before the <code>Renderer</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads any content needed by the Renderer.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td>RenderEffect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td>Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GraphicsDeviceService</strong></td>
<td>Hold a reference to the games GraphicsDeviceService. (Inherited from Renderer.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerRenderer</td>
<td>The renderer which will be used to render to the render target.</td>
</tr>
<tr>
<td>RenderTarget</td>
<td>Gets the render target.</td>
</tr>
<tr>
<td>RenderTargetIndex</td>
<td>Gets or sets the render target index to use, or uses zero if null.</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RenderTargetRenderer Constructor

RenderTargetRenderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RenderTargetRenderer()</code></td>
<td>Initializes a new instance of the <code>RenderTargetRenderer</code> class.</td>
</tr>
<tr>
<td><code>RenderTargetRenderer(Renderer)</code></td>
<td>Initializes a new instance of the <code>RenderTargetRenderer</code> class.</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
RenderTargetRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the `RenderTargetRenderer` class.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public RenderTargetRenderer()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RenderTargetRenderer()
See Also

RenderTargetRenderer Class
RenderTargetRenderer Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the RenderTargetRenderer class.

Namespace: ProjectMercury.Renderers
Assembly: ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public RenderTargetRenderer(
    Renderer innerRenderer
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New ( _
    innerRenderer As Renderer _
)
```

**Visual C++**

```cpp
public:
RenderTargetRenderer(
    Renderer^ innerRenderer
)
```

### Parameters

- `innerRenderer`
  
  Type: `ProjectMercury.Renderers::::Renderer`
  
  The renderer which will be used to render to the render target.
See Also

RenderTargetRenderer Class
RenderTargetRenderer Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RenderTargetRenderer` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphicsDeviceService</td>
<td>Hold a reference to the games GraphicsDeviceService. (Inherited from Renderer.)</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **RenderTargetRenderer** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Overloaded. Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Releases unmanaged resources and performs other cleanup operations before the <code>Renderer</code> is reclaimed by garbage collection. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Loads any content needed by the Renderer. (Overrides <code>Renderer::&lt;&gt;::LoadContent(ContentManager)</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>RenderEffect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td>Sets the render state of the graphics device before rendering an Emitter. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members (http://mpe.codeplex.com/People/ProjectPeople.aspx)
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="Dispose()" /></td>
<td>Disposes any unmanaged resources being used by this instance. (Inherited from Renderer.)</td>
</tr>
<tr>
<td><img src="" alt="Dispose(Boolean)" /></td>
<td>Releases unmanaged and - optionally - managed resources (Inherited from Renderer.)</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
RenderTargetRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RenderTargetRenderer::LoadContent Method

```
RenderTargetRenderer Class  See Also  Send Feedback
```

[This is preliminary documentation and is subject to change.]

Loads any content needed by the Renderer.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override void LoadContent(
    ContentManager content
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub LoadContent (_
    content As ContentManager _
)
```

**Visual C++**

```cpp
public:
virtual void LoadContent(
    ContentManager^ content
) override
```

### Parameters

**content**

Type: ContentManager

See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RenderTargetRenderer::RenderEffect Method

RenderTargetRenderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RenderEffect(ParticleEffect)</code></td>
<td>Renders the specified ParticleEffect. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td><code>RenderEffect(ParticleEffect, Matrix%)</code></td>
<td>Renders the specified ParticleEffect, applying the specified transformation offset. (Inherited from <code>Renderer</code>.)</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
RenderTargetRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
RenderTargetRenderer::RenderEmitter Method

RenderTargetRenderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RenderEmitter(Emitter)</code></td>
<td>Renders the specified Emitter. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Renders the specified Emitter, applying the specified transformation offset.</td>
</tr>
<tr>
<td><code>RenderEmitter(Emitter, Matrix%)</code></td>
<td>(Overrides <code>Renderer...;::RenderEmitter(Emitter, Matrix%).</code>)</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
RenderTargetRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified Emitter, applying the specified transformation offset.

Namespace:  ProjectMercury.Renderers  
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public override void RenderEmitter(
    Emitter emitter,
    ref Matrix transform
)
```

**Visual Basic (Declaration)**

```vbscript
Public Overrides Sub RenderEmitter ( _
    emitter As Emitter, _
    ByRef transform As Matrix _
)
```

**Visual C++**

```c
public:
virtual void RenderEmitter(
    Emitter^ emitter,
    Matrix% transform
) override
```

Parameters

**emitter**

Type: `ProjectMercury.Emitters..::.Emitter`  

[Missing `<param name="emitter"/>` documentation for  

**transform**

Type: `Matrix %`

[Missing `<param name="transform"/>` documentation for  
See Also

RenderTargetRenderer Class
RenderEmitter Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The `RenderTargetRenderer` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerRenderer</td>
<td>The renderer which will be used to render to the render target.</td>
</tr>
<tr>
<td>RenderTarget</td>
<td>Gets the render target.</td>
</tr>
<tr>
<td>RenderTargetIndex</td>
<td>Gets or sets the render target index to use, or uses zero if null.</td>
</tr>
</tbody>
</table>
See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The renderer which will be used to render to the render target.

**Namespace:** ProjectMercury.Renderers

**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

```csharp
public Renderer InnerRenderer { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property InnerRenderer As Renderer
```

Visual C++

```cpp
public:
property Renderer^ InnerRenderer {
    Renderer^ get ();
    void set (Renderer^ value);
}
```
See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets the render target.

**Namespace:**  ProjectMercury.Renderers  
**Assembly:**  ProjectMercury (in ProjectMercury.dll)  
**Version:**  3.1.0.0
**Syntax**

**C#**

```csharp
public RenderTarget2D RenderTarget { get; private set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property RenderTarget As RenderTarget2D
```

**Visual C++**

```cpp
public:
property RenderTarget2D^ RenderTarget {
    RenderTarget2D^ get ();
    void set (RenderTarget2D^ value);
}
```

**Field Value**

The render target.
See Also

RenderTargetRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Gets or sets the render target index to use, or uses zero if null.

**Namespace:**  [ProjectMercury.Renderers](#)  
**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public Nullable<int> RenderTargetIndex { get; set; }

Visual Basic (Declaration)

Public Property RenderTargetIndex As Nullable(Of Integer)

Visual C++

public:
property Nullable<int> RenderTargetIndex {
    Nullable<int> get ();
    void set (Nullable<int> value);
}
See Also

[RenderTargetRenderer Class](#)
[ProjectMercury.Renderers Namespace](#)

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Defines a Renderer which uses the standard XNA SpriteBatch class to render Particles.

**Namespace:**  [ProjectMercury.Renderers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public sealed class SpriteBatchRenderer : Renderer
```

**Visual Basic (Declaration)**

```vbnet
Public NotInheritable Class SpriteBatchRenderer _
    Inherits Renderer
```

**Visual C++**

```cpp
public ref class SpriteBatchRenderer sealed : public Renderer
```
Inheritance Hierarchy

System::Object
ProjectMercury.Renderers::Renderer
  ProjectMercury.Renderers::SpriteBatchRenderer
See Also

SpriteBatchRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The **SpriteBatchRenderer** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpriteBatchRenderer</td>
<td>Initializes a new instance of the SpriteBatchRenderer class</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Dispose</td>
<td>Overloaded. Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from <strong>Object</strong>.) Releases unmanaged resources and performs other cleanup operations before the <strong>Renderer</strong> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <strong>Renderer</strong>.) Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <strong>Object</strong>.) Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <strong>Object</strong>.) Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td>LoadContent</td>
<td>(Overrides <strong>Renderer,</strong>:::<strong>LoadContent</strong>(ContentManager).) Loads any content required by the renderer.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from <strong>Object</strong>.) Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td>RenderEffect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td>Sets the render state of the graphics device before rendering an Emitter. (Inherited from <strong>Renderer</strong>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Inherited from Renderer.)</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Initializes a new instance of the \texttt{SpriteBatchRenderer} class

\textbf{Namespace: } \texttt{ProjectMercury.Renderers}

\textbf{Assembly: } ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
**Syntax**

**C#**

```csharp
public SpriteBatchRenderer()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
SpriteBatchRenderer()
```
See Also

SpriteBatchRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The *SpriteBatchRenderer* type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Inherited from Renderer.)</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
The SpriteBatchRenderer type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>Equals</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Releases unmanaged resources and performs other cleanup operations before the <a href="#">Renderer</a> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Renderer</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>LoadContent</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>RenderEffect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RenderEmitter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetRenderState</td>
<td>Sets the render state of the graphics device before rendering an Emitter.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Renderer</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference

SpriteBatchRenderer...::.Dispose Method

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()()</td>
<td>Disposes any unmanaged resources being used by this instance. (Inherited from Renderer.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources (Inherited from Renderer.)</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
SpriteBatchRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
SpriteBatchRenderer..::.LoadContent Method

**SpriteBatchRenderer Class**  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]

Loads any content required by the renderer.

**Namespace:**  [ProjectMercury.Renderers](#)

**Assembly:**  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public override void LoadContent(
    ContentManager content
)

Visual Basic (Declaration)

Public Overrides Sub LoadContent (_
    content As ContentManager _
)

Visual C++

public:
virtual void LoadContent(
    ContentManager^ content
) override

Parameters

content
    Type: ContentManager

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System::::InvalidOperationException</code></td>
<td>Thrown if the <code>GraphicsDeviceManager</code> has not been set.</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Project Mercury API Reference
SpriteBatchRenderer...:.RenderEffect Method

SpriteBatchRenderer Class  See Also  Send Feedback

[This is preliminary documentation and is subject to change.]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RenderEffect(ParticleEffect)</code></td>
<td>Renders the specified ParticleEffect. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td><code>RenderEffect(ParticleEffect, Matrix%)</code></td>
<td>Renders the specified ParticleEffect, applying the specified transformation offset. (Inherited from <code>Renderer</code>.)</td>
</tr>
<tr>
<td><code>RenderEffect(ParticleEffect, SpriteBatch)</code></td>
<td>Renders the specified ParticleEffect.</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
SpriteBatchRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified ParticleEffect.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

C#

public void RenderEffect(
    ParticleEffect effect,
    SpriteBatch spriteBatch
)

Visual Basic (Declaration)

Public Sub RenderEffect (_
    effect As ParticleEffect, _
    spriteBatch As SpriteBatch _
)

Visual C++

public:
void RenderEffect(
    ParticleEffect^ effect,
    SpriteBatch^ spriteBatch
)

Parameters

effect
Type: ProjectMercury:::ParticleEffect

[sMissing <param name="effect"/> documentation for

spriteBatch
Type: SpriteBatch

[sMissing <param name="spriteBatch"/> documentation for
See Also

SpriteBatchRenderer Class
RenderEffect Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Project Mercury API Reference

SpriteBatchRenderer:::RenderEmitter Method

[This is preliminary documentation and is subject to change.]
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderEmitter(Emitter)</td>
<td>Renders the specified Emitter. (Inherited from <a href="#">Renderer</a>.)</td>
</tr>
<tr>
<td>RenderEmitter(Emitter, SpriteBatch)</td>
<td>Renders the specified Emitter.</td>
</tr>
<tr>
<td>RenderEmitter(Emitter, Matrix%)</td>
<td>Renders the specified Emitter, applying the specified transformation offset. (Overrides <a href="#">Renderer::RenderEmitter(Emitter, Matrix%)</a>.)</td>
</tr>
</tbody>
</table>
See Also

SpriteBatchRenderer Class
SpriteBatchRenderer Members
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified Emitter.

**Namespace:** ProjectMercury.Renderers  
**Assembly:** ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
Syntax

**C#**

```csharp
public void RenderEmitter(
    Emitter emitter,
    SpriteBatch spriteBatch
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub RenderEmitter (_
    emitter As Emitter,_
    spriteBatch As SpriteBatch _
)
```

**Visual C++**

```cpp
public:
void RenderEmitter(
    Emitter^ emitter,
    SpriteBatch^ spriteBatch
)
```

**Parameters**

**emitter**
- Type: `ProjectMercury.Emitters..:::Emitter`

**spriteBatch**
- Type: `SpriteBatch`
See Also

(SpriteBatchRenderer Class)  
(RenderEmitter Overload)  
(ProjectMercury.Renderers Namespace)

Copyright © 2009, 2010 Project Mercury Team Members  
(http://mpe.codeplex.com/People/ProjectPeople.aspx)
Renders the specified Emitter, applying the specified transformation offset.

Namespace:  ProjectMercury.Renderers
Assembly:  ProjectMercury (in ProjectMercury.dll) Version: 3.1.0.0
### Syntax

**C#**

```csharp
public override void RenderEmitter(
    Emitter emitter,
    ref Matrix transform
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub RenderEmitter (_
    emitter As Emitter, _
    ByRef transform As Matrix _
)
```

**Visual C++**

```cpp
public:
    virtual void RenderEmitter(
        Emitter^ emitter,
        Matrix% transform
    ) override
```

### Parameters

**emitter**
- Type: `ProjectMercury.Emitters..::.Emitter`

**transform**
- Type: `Matrix %`
See Also

SpriteBatchRenderer Class
RenderEmitter Overload
ProjectMercury.Renderers Namespace

Copyright © 2009, 2010 Project Mercury Team Members
(http://mpe.codeplex.com/People/ProjectPeople.aspx)