

Namespace

Namespaces ► Yogesh.ExcelXml

C#

[Missing namespace summary documentation for N:Yogesh.ExcelXml]

Declaration Syntax

C#

Visual Basic

Visual C++

```
namespace Yogesh.ExcelXml
```

```
Namespace Yogesh.ExcelXml
```

```
namespace Yogesh.ExcelXml
```

Types

All Types

Classes

Enumerations

Icon	Type	Description
	AlignmentOptions	Gets or sets cell's alignment options
	AlignmentOptionsBase	Gets or sets cell's alignment options
	Borderline	Different style of border lines
	BorderOptions	Gets or sets the border options
	BorderOptionsBase	Gets or sets the border options

	BorderSides	Different type of border sides.
	Cell	Cell class represents a single cell in a worksheet
	CellCollection	Represents a strongly typed list of cells that can be accessed by index.
	CellIndexInfo	Cell Index Information
	CellSettingsApplier	Gets and sets various cell and range properties
	Column	Column class represents a column properties of a single column in a worksheet
	ContentType	The cell content type
	DisplayFormatType	The default display format of the cell in excel
	DocumentProperties	Gets or sets document properties
	ExcelXmlWorkbook	This class represents a excel

		workbook
	FontOptions	Gets or sets cell's font options
	FontOptionsBase	Gets or sets cell's font options
	Formula	Formula is a formula builder class which can be stored directly in a cell
	HorizontalAlignment	Cell's horizontal alignment values
	InteriorOptions	Gets or sets cell's interior options
	InteriorOptionsBase	Gets or sets cell's interior options
	PageLayout	Page layout
	PageOrientation	Orientation mode
	Parameter	Parameter denotes a single parameter in a formula
	ParameterType	Formula parameter types

	Pattern	Different types of cell background patterns
	PrintOptions	Gets or sets various sheet printing options
	Range	Defines a range of cells
	Row	Row class represents a single row in a worksheet
	Styles	Style class for cells, rows and worksheets
	VerticalAlignment	Cell's vertical alignment values
	Worksheet	Worksheet class represents a single sheet in a workbook
	XmlStyle	Style class for cells, rows and worksheets

A Sandcastle Documented Class Library **AlignmentOptions Class**

Namespaces ► [Yogesh.ExcelXml](#) ► AlignmentOptions C#

Gets or sets cell's alignment options

Declaration Syntax

C# Visual Basic Visual C++

```
public class AlignmentOptions
```

```
Public Class AlignmentOptions
```

```
public ref class AlignmentOptions
```

Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	AlignmentOptions()	Creates a new instance
	AlignmentOptions(AlignmentOptions)	Creates a new instance based on another instance
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an Object to

		<p>attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.</p> <p>(Inherited from Object.)</p>
	GetHashCode()	<p>Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table</p> <p>(Inherited from Object.)</p>
	GetType()	<p>Gets the Type of the current instance.</p> <p>(Inherited from Object.)</p>
	Horizontal	Gets or sets horizontal alignment of the cell
	Indent	Gets or sets the indent
	MemberwiseClone()	<p>Creates a shallow copy of the current Object.</p> <p>(Inherited from Object.)</p>
	Rotate	Gets or sets the text rotation
	ShrinkToFit	Gets or sets cell's shrink to cell setting

	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Vertical	Gets or sets vertical alignment of the cell
	WrapText	Gets or sets text wrap setting

Inheritance Hierarchy

Object

└─ AlignmentOptions

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Constructor

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [AlignmentOptions](#) ►

[AlignmentOptions\(\)](#)

C# ▼

Members

Icon	Member	Description
	AlignmentOptions()	Creates a new instance
	AlignmentOptions(AlignmentOptions)	Creates a new instance based on another instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Constructor

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [AlignmentOptions](#) ►

AlignmentOptions()

C# ▼

Creates a new instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public AlignmentOptions()
```

```
Public Sub New
```

```
public:  
AlignmentOptions()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Constructor (ao)

Namespaces ► [Yogesh.ExcelXml](#) ► [AlignmentOptions](#) ►

AlignmentOptions(AlignmentOptions)

C# ▼

Creates a new instance based on another instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public AlignmentOptions(  
    AlignmentOptions ao  
)
```

```
Public Sub New ( _  
    ao As AlignmentOptions _  
)
```

```
public:  
AlignmentOptions(  
    AlignmentOptions^ ao  
)
```

▣ Parameters

ao (AlignmentOptions)

Instance to copy

Gets or sets horizontal alignment of the cell

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public HorizontalAlignment Horizontal { get; set; }</pre>		
<pre>Public Property Horizontal As HorizontalAlignment</pre>		
<pre>public: virtual property HorizontalAlignment Horizontal { HorizontalAlignment get () sealed; void set (HorizontalAlignment value) sealed; }</pre>		

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the indent

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public int Indent { get; set; }</pre>		
<pre>Public Property Indent As Integer</pre>		
<pre>public: virtual property int Indent { int get () sealed; void set (int value) sealed; }</pre>		

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the text rotation

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Rotate As Integer
```

```
public:  
virtual property int Rotate {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets cell's shrink to cell setting

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool ShrinkToFit { get; set; }
```

```
Public Property ShrinkToFit As Boolean
```

```
public:  
virtual property bool ShrinkToFit {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets vertical alignment of the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public VerticalAlignment Vertical { get; set; }
```

```
Public Property Vertical As VerticalAlignment
```

```
public:  
virtual property VerticalAlignment Vertical {  
    VerticalAlignment get () sealed;  
    void set (VerticalAlignment value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets text wrap setting

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool WrapText { get; set; }
```

```
Public Property WrapText As Boolean
```

```
public:  
virtual property bool WrapText {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Class

Namespaces ► [Yogesh.ExcelXml](#) ► AlignmentOptionsBase

Gets or sets cell's alignment options

Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class AlignmentOptionsBase : CellSettingsAppl</code>		

```
Public Class AlignmentOptionsBase _
    Inherits CellSettingsApplier
```

```
public ref class AlignmentOptionsBase : public Cells
```

Members

All Members	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static <input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a

		particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	Horizontal	Gets or sets horizontal alignment of the cell
	Indent	Gets or sets the indent
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Rotate	Gets or sets the text rotation
	ShrinkToFit	Gets or sets cell's shrink to cell setting
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Vertical	Gets or sets vertical alignment of the cell
	WrapText	Gets or sets text wrap setting

▣ Inheritance Hierarchy

Object

└─ CellSettingsApplier

└─ AlignmentOptionsBase

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



Gets or sets horizontal alignment of the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public HorizontalAlignment Horizontal { get; set; }
```

```
Public Property Horizontal As HorizontalAlignment
```

```
public:  
virtual property HorizontalAlignment Horizontal {  
    HorizontalAlignment get () sealed;  
    void set (HorizontalAlignment value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the indent

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Indent As Integer
```

```
public:  
virtual property int Indent {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the text rotation

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Rotate As Integer
```

```
public:  
virtual property int Rotate {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

ShrinkToFit

Gets or sets cell's shrink to cell setting

▣ Declaration Syntax

C# Visual Basic Visual C++

```
public bool ShrinkToFit { get; set; }
```

```
Public Property ShrinkToFit As Boolean
```

```
public:  
virtual property bool ShrinkToFit {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets vertical alignment of the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public VerticalAlignment Vertical { get; set; }
```

```
Public Property Vertical As VerticalAlignment
```

```
public:  
virtual property VerticalAlignment Vertical {  
    VerticalAlignment get () sealed;  
    void set (VerticalAlignment value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

WrapText

Gets or sets text wrap setting

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool WrapText { get; set; }
```

```
Public Property WrapText As Boolean
```

```
public:  
virtual property bool WrapText {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Different style of border lines

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum Borderline
```

```
Public Enumeration Borderline
```

```
public enum class Borderline
```

▣ Members

Member	Description
Continuous	Continuous line
Dash	Dash line
DashDot	DashDot line
DashDotDot	DashDotDot line
Double	Double line
Dot	Dot line

SlantDashDot

SlantDashDot line

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the border options

[-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public class BorderOptions
```

```
Public Class BorderOptions
```

```
public ref class BorderOptions
```

[-] Members

All Members

Constructors

Methods

Properties

 Public
 Protected

 Instance
 Static

 Declare
 Inherited

Icon	Member	Description
	BorderOptions()	Creates a new instance
	BorderOptions(BorderOptions)	Creates a new instance based on another instance
	Color	Gets or sets border color
	Equals(Object)	Determines whether the specified Object is equal to the current Object .

		(Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	LineStyle	Gets or sets the border line style
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Sides	Gets or sets the border side flags
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Weight	Gets or sets the width of the border

▣ Inheritance Hierarchy

Object

└─ BorderOptions

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Members

Icon	Member	Description
	BorderOptions()	Creates a new instance
	BorderOptions(BorderOptions)	Creates a new instance based on another instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Creates a new instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BorderOptions()
```

```
Public Sub New
```

```
public:  
BorderOptions()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **BorderOptions Constructor**
(borderOptions)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [BorderOptions](#) ►
BorderOptions(BorderOptions)

C# ▼

Creates a new instance based on another instance

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public BorderOptions(  
    BorderOptions borderOptions  
)
```

```
Public Sub New ( _  
    borderOptions As BorderOptions _  
)
```

```
public:  
BorderOptions(  
    BorderOptions^ borderOptions  
)
```

▣ **Parameters**

borderOptions (BorderOptions)

Instance to copy

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets border color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color Color { get; set; }
```

```
Public Property Color As Color
```

```
public:  
virtual property Color Color {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the border line style

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Borderline LineStyle { get; set; }
```

```
Public Property LineStyle As Borderline
```

```
public:  
virtual property Borderline LineStyle {  
    Borderline get () sealed;  
    void set (Borderline value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the border side flags

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public BorderSides Sides { get; set; }</pre>		
<pre>Public Property Sides As BorderSides</pre>		
<pre>public: virtual property BorderSides Sides { BorderSides get () sealed; void set (BorderSides <i>value</i>) sealed; }</pre>		

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the width of the border

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Weight As Integer
```

```
public:  
virtual property int Weight {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Namespaces ► [Yogesh.ExcelXml](#) ► [BorderOptionsBase](#) C#

Gets or sets the border options

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class BorderOptionsBase : CellSettingsApplier
```

```
Public Class BorderOptionsBase _
    Inherits CellSettingsApplier
```

```
public ref class BorderOptionsBase : public CellSett
```

Members

All Members

Methods

Properties

 Public
 Protected

 Instance
 Static

 Declare
 Inherited

Icon	Member	Description
	Color	Gets or sets border color
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)

	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	LineStyle	Gets or sets the border line style
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Sides	Gets or sets the border side flags
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Weight	Gets or sets the width of the border

Inheritance Hierarchy

Object

└─ [CellSettingsApplier](#)

└─ BorderOptionsBase



Gets or sets border color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color Color { get; set; }
```

```
Public Property Color As Color
```

```
public:  
virtual property Color Color {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the border line style

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Borderline LineStyle { get; set; }
```

```
Public Property LineStyle As Borderline
```

```
public:  
virtual property Borderline LineStyle {  
    Borderline get () sealed;  
    void set (Borderline value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the border side flags

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BorderSides Sides { get; set; }
```

```
Public Property Sides As BorderSides
```

```
public:  
virtual property BorderSides Sides {  
    BorderSides get () sealed;  
    void set (BorderSides value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the width of the border

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Weight As Integer
```

```
public:  
virtual property int Weight {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Different type of border sides.

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
[FlagsAttribute]  
public enum BorderSides
```

```
<FlagsAttribute> _  
Public Enumeration BorderSides
```

```
[FlagsAttribute]  
public enum class BorderSides
```

▣ Members

Member	Description
None	No border
Top	Cell has a top border
Left	Cell has a left border
Bottom	Cell has a botom border
Right	Cell has a right border

All

Cell has full border on all sides

▣ **Remarks**

Multiple values can be combined by an or (i.e. "|") operation.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Cell class represents a single cell in a worksheet

Declaration Syntax

C# Visual Basic Visual C++

```
public class Cell : Styles
```

```
Public Class Cell _
    Inherits Styles
```

```
public ref class Cell : public Styles
```

Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	Alignment	Gets or sets cell alignment options (Inherited from Styles .)
	Border	Gets or sets border settings (Inherited from Styles .)
	Comment	Gets or sets the comment for the cell
	ContentType	Returns the cell content type

	CustomFormatString	Gets or sets custom display format string (Inherited from Styles .)
	Delete()	Deletes a cell from the parent row
	DisplayFormat	Gets or sets the cell display format (Inherited from Styles .)
	Empty()	Empties the content of a cell
	Equals(Object)	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 .)
	Font	Gets or sets the font options (Inherited from Styles .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)

	GetValue<T>()	Gets the value of a cell converted to system type
	HRef	Gets or sets the a external reference as a link
	Index	Index information of the cell
	Interior	Gets or sets interior options (Inherited from Styles.)
	IsEmpty()	Checks whether the cell has no content and no comment
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object.)
	Style	Returns the XmlStyle reference of the cell (Inherited from Styles.)
	ToString()	Returns a String that represents the current Object . (Inherited from Object.)
	Unmerge()	Unmerges a cell
	Value	Gets or sets the value of the cell

▣ **Remarks**

Cell class represents a single cell in a worksheet.

You cannot directly declare a instance of a cell from your code by using **new** keyword. The only way to access a cell is to retrieve it from a worksheet or a row.

▣ **Inheritance Hierarchy**

Object

└─ **CellSettingsApplier**

└─ **Styles**

└─ **Cell**

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the comment for the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Comment { get; set; }
```

```
Public Property Comment As String
```

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

▣ Remarks

Comment is in raw html format which means you can insert bold and italics markers just like regular html

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Returns the cell content type

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ContentType ContentType { get; }
```

```
Public ReadOnly Property ContentType As ContentType
```

```
public:  
property ContentType ContentType {  
    ContentType get ();  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Deletes a cell from the parent row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Delete()
```

```
Public Sub Delete
```

```
public:  
void Delete()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Empties the content of a cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Empty()
```

```
Public Sub Empty
```

```
public:  
void Empty()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets the value of a cell converted to a system type

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public T GetValue<T>()
```

```
Public Function GetValue(Of T) As T
```

```
public:  
generic<typename T>  
T GetValue()
```

▣ Generic Template Parameters

T

Type to convert to

▣ Return Value

Cell value converted to system type

Gets or sets the a external reference as a link

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string HRef { get; set; }
```

```
Public Property HRef As String
```

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

▣ Remarks

The value of HRef is not verified.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Index information of the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public CellIndexInfo Index { get; }
```

```
Public ReadOnly Property Index As CellIndexInfo
```

```
public:  
property CellIndexInfo^ Index {  
    CellIndexInfo^ get ();  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Checks whether the cell has no content and no comment

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool IsEmpty()
```

```
Public Function IsEmpty As Boolean
```

```
public:  
bool IsEmpty()
```

▣ Return Value

true if empty, false otherwise

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Unmerges a cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Unmerge()
```

```
Public Sub Unmerge
```

```
public:  
void Unmerge()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the value of the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object Value { get; set; }
```

```
Public Property Value As Object
```

```
public:  
property Object^ Value {  
    Object^ get ();  
    void set (Object^ value);  
}
```

▣ Remarks

Value returns a boxed **String** value of the cell or sets the value of the cell to...

1. **String**
2. **Boolean**
3. **Byte**
4. **Int16**
5. **Int32**
6. **Int64**
7. **Double**
8. **Decimal**
9. **DateTime**
10. [Cell](#)
11. [Formula](#)

If the type is not any of the above, cell value is set to null.

2.89.501.2158

A Sandcastle Documented Class Library **CellCollection Class**

Namespaces ► [Yogesh.ExcelXml](#) ► CellCollection C#

Represents a strongly typed list of cells that can be accessed by index.

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class CellCollection : List<Cell>
```

```
Public Class CellCollection _  
    Inherits List(Of Cell)
```

```
public ref class CellCollection : public List<Cell^>
```

Members

All Members

Constructors

Methods

Properties

Public
 Protected

Instance
 Static

Declare
 Inherited

Icon	Member	Description
	CellCollection()	
	Add(Range)	Adds a range to the collection
	Add(Worksheet)	Adds a worksheet to the collection
	Add(Row)	Adds a row to the collection

	<code>Add(Cell, Predicate<Cell>)</code>	Adds a single cell to the collection if it matches the filter condition
	<code>Add(Range, Predicate<Cell>)</code>	Adds a range to the collection if it matches the filter condition
	<code>Add(Worksheet, Predicate<Cell>)</code>	Adds a worksheet to the collection if it matches the filter condition
	<code>Add(Row, Predicate<Cell>)</code>	Adds a row to the collection if it matches the filter condition
	<code>Add(T)</code>	Adds an object to the end of the List<T> . (Inherited from List<Cell> .)
	<code>AddRange(IEnumerable<T>)</code>	Adds the elements of the specified collection to the end of the List<T> . (Inherited from List<Cell> .)
	<code>AsReadOnly()</code>	Returns a read-only IList<T> wrapper for the current collection. (Inherited from List<Cell> .)
	<code>BinarySearch(Int32, Int32, T, IComparer<T>)</code>	Searches a range of elements in the sorted

		<p>List<T> for an element using the specified comparer and returns the zero-based index of the element.</p> <p>(Inherited from List<Cell>.)</p>
	BinarySearch(T)	<p>Searches the entire sorted List<T> for an element using the default comparer and returns the zero-based index of the element.</p> <p>(Inherited from List<Cell>.)</p>
	BinarySearch(T, IComparer<T>)	<p>Searches the entire sorted List<T> for an element using the specified comparer and returns the zero-based index of the element.</p> <p>(Inherited from List<Cell>.)</p>
	Capacity	<p>Gets or sets the total number of elements the internal data structure can hold without resizing.</p> <p>(Inherited from List<Cell>.)</p>
	Clear()	<p>Removes all elements from the List<T>.</p> <p>(Inherited from List<Cell>.)</p>
	Contains(T)	<p>Determines whether an element is in the List<T>.</p> <p>(Inherited from List<Cell>.)</p>

	ConvertAll<TOutput>(Converter<T, TOutput>)	<p>Converts the elements in the current List<T> to another type, and returns a list containing the converted elements.</p> <p>(Inherited from List<Cell>.)</p>
	CopyTo(T[])	<p>Copies the entire List<T> to a compatible one-dimensional array, starting at the beginning of the target array.</p> <p>(Inherited from List<Cell>.)</p>
	CopyTo(Int32, T[], Int32, Int32)	<p>Copies a range of elements from the List<T> to a compatible one-dimensional array, starting at the specified index of the target array.</p> <p>(Inherited from List<Cell>.)</p>
	CopyTo(T[], Int32)	<p>Copies the entire List<T> to a compatible one-dimensional array, starting at the specified index of the target array.</p> <p>(Inherited from List<Cell>.)</p>
	Count	<p>Gets the number of elements actually contained in the List<T>.</p> <p>(Inherited from List<Cell>.)</p>
	Equals(Object)	<p>Determines whether the specified Object is equal to the current Object.</p>

		(Inherited from Object .)
	Exists(Predicate<T>)	<p>Determines whether the List<T> contains elements that match the conditions defined by the specified predicate.</p> <p>(Inherited from List<Cell>.)</p>
	Finalize()	<p>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.</p> <p>(Inherited from Object.)</p>
	Find(Predicate<T>)	<p>Searches for an element that matches the condition defined by the specified predicate, and returns the first occurrence within the entire List<T>.</p> <p>(Inherited from List<Cell>.)</p>
	FindAll(Predicate<T>)	<p>Retrieves the all the elements that match the conditions defined by the specified predicate.</p> <p>(Inherited from List<Cell>.)</p>
	FindIndex(Predicate<T>)	<p>Searches for an element that matches the condition defined by the specified predicate, and returns the zero-based index of the first occurrence within the</p>

		entire List<T> . (Inherited from List<Cell>).
	FindIndex(Int32, Predicate<T>)	Searches for an element that matches the condition defined by the specified predicate, 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<Cell>).
	FindIndex(Int32, Int32, Predicate<T>)	Searches for an element that matches the condition defined by the specified predicate, 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<Cell>).
	FindLast(Predicate<T>)	Searches for an element that matches the condition defined by the specified predicate, and returns the last occurrence within the entire List<T> . (Inherited from List<Cell>).
	FindLastIndex(Predicate<T>)	Searches for an element

		<p>that matches the condition defined by the specified predicate, and returns the zero-based index of the last occurrence within the entire List<T>.</p> <p>(Inherited from List<Cell>.)</p>
	FindLastIndex(Int32, Predicate<T>)	<p>Searches for an element that matches the condition defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the List<T> that extends from the first element to the specified index.</p> <p>(Inherited from List<Cell>.)</p>
	FindLastIndex(Int32, Int32, Predicate<T>)	<p>Searches for an element that matches the condition defined by the specified predicate, and returns the zero-based index of the last occurrence within the range of elements in the List<T> that contains the specified number of elements and ends at the specified index.</p> <p>(Inherited from List<Cell>.)</p>
	ForEach(Action<T>)	<p>Performs the specified action on each element of the List<T>.</p> <p>(Inherited from List<Cell>.)</p>

	GetEnumerator()	Returns an enumerator that iterates through the List<T> . (Inherited from List<Cell> .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetRange(Int32, Int32)	Creates a shallow copy of range of elements in the source List<T> . (Inherited from List<Cell> .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	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<Cell> .)
	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<Cell>).
	IndexOf(T, Int32, Int32)	<p>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.</p> <p>(Inherited from List<Cell>).</p>
	Insert(Int32, T)	<p>Inserts an element into the List<T> at the specified index.</p> <p>(Inherited from List<Cell>).</p>
	InsertRange(Int32, IEnumerable<T>)	<p>Inserts the elements of a collection into the List<T> at the specified index.</p> <p>(Inherited from List<Cell>).</p>
	Item[Int32]	<p>Gets or sets the element at the specified index.</p> <p>(Inherited from List<Cell>).</p>
	LastIndexOf(T)	<p>Searches for the specified object and returns the zero based index of the last occurrence within the entire List<T>.</p> <p>(Inherited from List<Cell>).</p>
	LastIndexOf(T, Int32)	<p>Searches for the specified object and returns the zero</p>

		<p>based index of the last occurrence within the range of elements in the List<T> that extends from the first element to the specified index.</p> <p>(Inherited from List<Cell>.)</p>
☰	LastIndexOf(T, Int32, Int32)	<p>Searches for the specified object and returns the zero based index of the last occurrence within the range of elements in the List<T> that contains the specified number of elements and ends at the specified index.</p> <p>(Inherited from List<Cell>.)</p>
💡	MemberwiseClone()	<p>Creates a shallow copy of the current Object.</p> <p>(Inherited from Object.)</p>
☰	Remove(T)	<p>Removes the first occurrence of a specific object from the List<T>.</p> <p>(Inherited from List<Cell>.)</p>
☰	RemoveAll(Predicate<T>)	<p>Removes all the elements that match the conditions defined by the specified predicate.</p> <p>(Inherited from List<Cell>.)</p>
☰	RemoveAt(Int32)	<p>Removes the element at the specified index of the List<T>.</p>

		(Inherited from List<Cell>).
	RemoveRange(Int32, Int32)	Removes a range of elements from the List<T> (Inherited from List<Cell>).
	Reverse()	Reverses the order of the elements in the entire List<T> . (Inherited from List<Cell>).
	Reverse(Int32, Int32)	Reverses the order of the elements in the specified range. (Inherited from List<Cell>).
	Sort()	Sorts the elements in the entire List<T> using the default comparer. (Inherited from List<Cell>).
	Sort(IComparer<T>)	Sorts the elements in the entire List<T> using the specified comparer. (Inherited from List<Cell>).
	Sort(Int32, Int32, IComparer<T>)	Sorts the elements in a range of elements in List<T> using the specific comparer. (Inherited from List<Cell>).
	Sort(Comparison<T>)	Sorts the elements in the entire List<T> using the specified Comparison<T> (Inherited from List<Cell>).

	ToArray()	Copies the elements of the List<T> to a new array. (Inherited from List<Cell> .)
	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<Cell> .)
	TrueForAll(Predicate<T>)	Determines whether every element in the List<T> matches the conditions defined by the specified predicate. (Inherited from List<Cell> .)

Inheritance Hierarchy

Object

└─ **List<Cell>**

└─ CellCollection

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

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

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public CellCollection()
```

```
Public Sub New
```

```
public:  
CellCollection()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Members

Icon	Member	Description
	Add(Range)	Adds a range to the collection
	Add(Worksheet)	Adds a worksheet to the collection
	Add(Row)	Adds a row to the collection
	Add(Cell, Predicate<Cell>)	Adds a single cell to the collection if it matches the filter condition
	Add(Range, Predicate<Cell>)	Adds a range to the collection if it matches the filter condition
	Add(Worksheet, Predicate<Cell>)	Adds a worksheet to the collection if it matches the filter condition
	Add(Row, Predicate<Cell>)	Adds a row to the collection if it matches the filter condition
	Add(T)	Adds an object to the end of the List<T> .

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **Add Method (cell, filterCondition)**

Namespaces ► [Yogesh.ExcelXml](#) ► [CellCollection](#) ► [Add\(Cell, Predicate<Cell>\)](#)

Adds a single cell to the collection if it matches the filter condition

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void Add(  
    Cell cell,  
    Predicate<Cell> filterCondition  
)
```

```
Public Sub Add ( _  
    cell As Cell, _  
    filterCondition As Predicate(Of Cell) _  
)
```

```
public:  
void Add(  
    Cell^ cell,  
    Predicate<Cell^>^ filterCondition  
)
```

▣ **Parameters**

cell (Cell)

Cell to add

filterCondition (Predicate<Cell>)

Filter predicate

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **Add Method (range)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [CellCollection](#) ► [Add\(Range\)](#)

Adds a range to the collection

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public void Add(Range range)</pre>		
	<pre>Public Sub Add (_ range As Range _)</pre>	
<pre>public: void Add(Range^ range)</pre>		

▣ **Parameters**

range (Range)
Range to add

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **Add Method (range, filterCondition)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [CellCollection](#) ► [Add\(Range, Predicate<Cell>\)](#)

Adds a range to the collection if it matches the filter condition

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void Add(  
    Range range,  
    Predicate<Cell> filterCondition  
)
```

```
Public Sub Add ( _  
    range As Range, _  
    filterCondition As Predicate(Of Cell) _  
)
```

```
public:  
void Add(  
    Range^ range,  
    Predicate<Cell>^ filterCondition  
)
```

▣ **Parameters**

range (Range)

Range to add

filterCondition (Predicate<Cell>)

Filter predicate

A Sandcastle Documented Class Library **Add Method (row)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [CellCollection](#) ► [Add\(Row\)](#)

C# ▼

Adds a row to the collection

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void Add(  
    Row row  
)
```

```
Public Sub Add ( _  
    row As Row _  
)
```

```
public:  
void Add(  
    Row^ row  
)
```

▣ **Parameters**

row (Row)

Row to add

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **Add Method (row, filterCondition)**

Namespaces ► [Yogesh.ExcelXml](#) ► [CellCollection](#) ► [Add\(Row, Predicate<Cell>\)](#)

Adds a row to the collection if it matches the filter condition

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public void Add(Row row, Predicate<Cell> filterCondition)</pre>		

```
Public Sub Add ( _  
    row As Row, _  
    filterCondition As Predicate(Of Cell) _  
)
```

```
public:  
void Add(  
    Row^ row,  
    Predicate<Cell^>^ filterCondition  
)
```

▣ **Parameters**

row (Row)

Row to add

filterCondition (Predicate<Cell>)

Filter predicate

C# ▼

Adds a worksheet to the collection

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Add(  
    Worksheet ws  
)
```

```
Public Sub Add ( _  
    ws As Worksheet _  
)
```

```
public:  
void Add(  
    Worksheet^ ws  
)
```

▣ Parameters

ws (Worksheet)
Worksheet to add

A Sandcastle Documented Class Library **Add Method (ws, filterCondition)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [CellCollection](#) ► [Add\(Worksheet, Predicate<Cell>\)](#)

Adds a worksheet to the collection if it matches the filter condition

C#

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void Add(  
    Worksheet ws,  
    Predicate<Cell> filterCondition  
)
```

```
Public Sub Add ( _  
    ws As Worksheet, _  
    filterCondition As Predicate(Of Cell) _  
)
```

```
public:  
void Add(  
    Worksheet^ ws,  
    Predicate<Cell>^ filterCondition  
)
```

▣ **Parameters**

ws (Worksheet)

Worksheet to add

filterCondition (Predicate<Cell>)

Filter predicate

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Cell Index Information

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class CellIndexInfo
```

```
Public Class CellIndexInfo
```

```
public ref class CellIndexInfo
```

Members

All Members

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	ColumnIndex	Column index starting from 0
	Equals(Object)	Determines whether the specified Object is equal to the current Object (Inherited from Object .)
	ExcelColumnIndex	Index in excel format, eg. A1
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is

		reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	RowIndex	Row index starting from 0
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)

▣ Inheritance Hierarchy

Object

└─ CellIndexInfo

Column index starting from 0

Declaration Syntax

C#

Visual Basic

Visual C++

```
public int ColumnIndex { get; private set; }
```

```
Public Property ColumnIndex As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Index in excel format, eg. A1

Declaration Syntax

C#

Visual Basic

Visual C++

```
public string ExcelColumnIndex { get; private set; }
```

```
Public Property ExcelColumnIndex As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Row index starting from 0

[-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public int RowIndex { get; private set; }
```

```
Public Property RowIndex As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets and sets various cell and range properties

[-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract class CellSettingsApplier
```

```
Public MustInherit Class CellSettingsApplier
```

```
public ref class CellSettingsApplier abstract
```

[-] Members

All Members

Methods

 Public Protected Instance Static Declared Inherited

Icon	Member	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table.

		(Inherited from Object .)
🔍	GetType()	Gets the Type of the current instance (Inherited from Object .)
💡	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
🔍	ToString()	Returns a String that represents the current Object . (Inherited from Object .)

▣ Inheritance Hierarchy

Object

- └─ CellSettingsApplier
 - └─ Styles
 - └─ FontOptionsBase
 - └─ InteriorOptionsBase
 - └─ AlignmentOptionsBase
 - └─ BorderOptionsBase

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Column class represents a column properties of a single column in a worksheet

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class Column
```

```
Public Class Column
```

```
public ref class Column
```

Members

All Members

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms

		and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	Hidden	Gets or sets the hidden status of the column
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Style	Gets or sets the XmlStyle reference of the column. Setting this option only affects cells which are added after this value is set. The cells which are added in the same column retain their original style settings.
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Width	Gets or sets the default width of the column

Remarks

Column class represents a column properties of a single column in a worksheet.

You cannot directly declare a instance of a column class from your code by using **new** keyword. The only way to access a column is to retrieve it from a worksheet by using the **Columns(Int32)** method of the **Worksheet** class.

▣ Inheritance Hierarchy

Object

└─ Column

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the hidden status of the column

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Hidden { get; set; }
```

```
Public Property Hidden As Boolean
```

```
public:  
property bool Hidden {  
    bool get ();  
    void set (bool value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the [XmlStyle](#) reference of the column.

Setting this option only affects cells which are added after this value is set. The cells which are added in the same column retain their original style settings.

Declaration Syntax

C#

Visual Basic

Visual C++

```
public XmlStyle Style { get; set; }
```

```
Public Property Style As XmlStyle
```

```
public:  
property XmlStyle^ Style {  
    XmlStyle^ get ();  
    void set (XmlStyle^ value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the default width of the column

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public double Width { get; set; }
```

```
Public Property Width As Double
```

```
public:  
property double Width {  
    double get ();  
    void set (double value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

The cell content type

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum ContentType
```

```
Public Enumeration ContentType
```

```
public enum class ContentType
```

▣ Members

Member	Description
None	Cell does not contain anything
String	Cell contains a string
Number	Cell contains a number
DateTime	Cell contains a DateTime value
Boolean	Cell contains a bool value
Formula	Cell contains a formula

UnresolvedValue

Cell contains a formula which cannot be resolved

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Enumeration

Namespaces ► [Yogesh.ExcelXml](#) ► [DisplayFormatType](#)

The default display format of the cell in excel

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum DisplayFormatType
```

```
Public Enumeration DisplayFormatType
```

```
public enum class DisplayFormatType
```

▣ Members

Member	Description
None	General format
Text	Displays anything as text (i.e. Left aligned without formatting)
Fixed	Displays numeric values with two fixed decimals
Standard	Displays numeric values with two fixed decimals and digit grouping
Percent	Displays numeric values as percentage values

Scientific	Displays numeric values in scientific notation
GeneralDate	Displays numeric or date values as formatted date values
ShortDate	Displays numeric or date values as short date format
LongDate	Displays numeric or date values as long date format
Time	Displays numeric or date values in time format
Custom	Custom defined format

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets document properties

Declaration Syntax

C# Visual Basic Visual C++

```
public class DocumentProperties
```

```
Public Class DocumentProperties
```

```
public ref class DocumentProperties
```

Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	DocumentProperties()	Creates an instance with empty document properties
	Author	Gets or sets the author of the workbook
	Company	Gets or sets the company of the workbook
	Equals(Object)	Determines whether the specified

		Object is equal to the current Object (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	LastAuthor	Gets or sets the last author of the workbook
	Manager	Gets or sets the manager of the workbook
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Subject	Gets or sets the subject of the workbook
	Title	Gets or sets the title of the workbook

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

▣ Inheritance Hierarchy

Object

└─ DocumentProperties

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Constructor

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [DocumentProperties](#) ►

DocumentProperties()

C# ▼

Creates an instance with empty document properties

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DocumentProperties()
```

```
Public Sub New
```

```
public:  
DocumentProperties()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the author of the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Author { get; set; }
```

```
Public Property Author As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the company of the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Company { get; set; }
```

```
Public Property Company As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the last author of the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string LastAuthor { get; set; }
```

```
Public Property LastAuthor As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the manager of the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Manager { get; set; }
```

```
Public Property Manager As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the subject of the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Subject { get; set; }
```

```
Public Property Subject As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the title of the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Title { get; set; }
```

```
Public Property Title As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

This class represents a excel workbook

Declaration Syntax

C# Visual Basic Visual C++

```
public class ExcelXmlWorkbook
```

```
Public Class ExcelXmlWorkbook
```

```
public ref class ExcelXmlWorkbook
```

Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	ExcelXmlWorkbook()	Creates a new instance
	Add()	Adds a new sheet at the end of a sheets
	Add(String)	Adds a new sheet at the end of a sheets
	AddNamedRange(Range, String)	Add a named range to the book with a book scope

	DataSetToWorkbook(DataSet)	Converts a dataset to a work boo
	DefaultStyle	Gets or sets default font options of the sheet
	DeleteSheet(Int32)	Delete a sheet by index
	DeleteSheet(String)	Delete a sheet by name
	DeleteSheet(Worksheet)	Delete a sheet by instance
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Export(String)	Export the workbook to a file
	Export(Stream)	Export the workbook to a stream
	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()	<p>Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table.</p> <p>(Inherited from Object.)</p>
	GetSheetByName(String)	<p>Gets a Worksheet reference matching a particular name</p>
	GetType()	<p>Gets the Type of the current instance.</p> <p>(Inherited from Object.)</p>
	Import(String)	<p>Imports an excel xml workbook into an ExcelXmlWorkbook instance</p>
	Import(Stream)	<p>Imports an excel xml workbook into an ExcelXmlWorkbook instance</p>
	InsertAfter(Int32)	<p>Insert a new sheet after another sheet</p>
	InsertAfter(String)	<p>Insert a new sheet after another sheet</p>
	InsertAfter(Worksheet)	<p>Insert a new sheet after another sheet</p>
		

	InsertBefore(Int32)	Insert a new sheet before another sheet
	InsertBefore(String)	Insert a new sheet before another sheet
	InsertBefore(Worksheet)	Insert a new sheet before another sheet
	Item[Int32]	Returns the sheet at a given position
	Item[String]	Returns the sheet by sheet name
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Properties	Gets or sets document properties
	SheetCount	The number of sheets in this book
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)

▣ Inheritance Hierarchy

Object

└─ ExcelXmlWorkbook

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Constructor

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

ExcelXmlWorkbook()

C# ▼

Creates a new instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ExcelXmlWorkbook()
```

```
Public Sub New
```

```
public:  
ExcelXmlWorkbook()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Members

Icon	Member	Description
	Add()	Adds a new sheet at the end of all sheets
	Add(String)	Adds a new sheet at the end of all sheets

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Adds a new sheet at the end of all sheets

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet Add()
```

```
Public Function Add As Worksheet
```

```
public:  
Worksheet^ Add()
```

▣ Return Value

New Worksheet instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **Add Method (sheetName)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

Add(String)

Adds a new sheet at the end of all sheets

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public Worksheet Add(  
    string sheetName  
)
```

```
Public Function Add ( _  
    sheetName As String _  
) As Worksheet
```

```
public:  
Worksheet^ Add(  
    String^ sheetName  
)
```

▣ **Parameters**

sheetName (String)
Sheet name

▣ **Return Value**

New Worksheet instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **AddNamedRange Method**
(range, name)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

AddNamedRange(Range, String)

C# ▼

Add a named range to the book with a book scope

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void AddNamedRange(  
    Range range,  
    string name  
)
```

```
Public Sub AddNamedRange ( _  
    range As Range, _  
    name As String _  
)
```

```
public:  
void AddNamedRange(  
    Range^ range,  
    String^ name  
)
```

▣ **Parameters**

range (Range)

Range to be named

name (String)

Name of the range

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DataSetToWorkbook Method (source)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ► [DataSetToWorkbook\(DataSet\)](#)

Converts a dataset to a work book

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static ExcelXmlWorkbook DataSetToWorkbook(  
    DataSet source  
)
```

```
Public Shared Function DataSetToWorkbook ( _  
    source As DataSet _  
) As ExcelXmlWorkbook
```

```
public:  
static ExcelXmlWorkbook^ DataSetToWorkbook(  
    DataSet^ source  
)
```

▣ Parameters

source (DataSet)

The source dataset to convert to a work book

▣ Return Value

Returns the [ExcelXmlWorkbook](#) for the dataset.

▣ Remarks

All the tables are converted into sheets with sheet names as table + tab number, eg. "Table0" "Table1" etc. Supported types which can be successfully converted to cells are the same as described in [Cell](#) except [Cell](#) and [Formula](#)

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets default font options of the sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public XmlStyle DefaultStyle { get; set; }
```

```
Public Property DefaultStyle As XmlStyle
```

```
public:  
property XmlStyle^ DefaultStyle {  
    XmlStyle^ get ();  
    void set (XmlStyle^ value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

DeleteSheet()

▣ **Members**

Icon	Member	Description
	DeleteSheet(Int32)	Delete a sheet by index
	DeleteSheet(String)	Delete a sheet by name
	DeleteSheet(Worksheet)	Delete a sheet by instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

DeleteSheet(Int32)

Delete a sheet by index

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteSheet(  
    int index  
)
```

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

```
public:  
void DeleteSheet(  
    int index  
)
```

▣ Parameters

index (Int32)

Index number of sheet to delete

(sheetName)

Namespaces ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

DeleteSheet(String)

Delete a sheet by name

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteSheet(  
    string sheetName  
)
```

```
Public Sub DeleteSheet ( _  
    sheetName As String _  
)
```

```
public:  
void DeleteSheet(  
    String^ sheetName  
)
```

▣ Parameters

sheetName (String)

Name of sheet to delete

DeleteSheet(Worksheet)

Delete a sheet by instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteSheet(  
    Worksheet ws  
)
```

```
Public Sub DeleteSheet ( _  
    ws As Worksheet _  
)
```

```
public:  
void DeleteSheet(  
    Worksheet^ ws  
)
```

▣ Parameters

ws (Worksheet)

Instance of sheet to delete

Members

Icon	Member	Description
	Export(String)	Export the workbook to a file
	Export(Stream)	Export the workbook to a stream

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

C# ▼

Export the workbook to a stream

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Export(  
    Stream stream  
)
```

```
Public Function Export ( _  
    stream As Stream _  
) As Boolean
```

```
public:  
bool Export(  
    Stream^ stream  
)
```

▣ Parameters

stream (Stream)

Output stream

▣ Return Value

true if the export was successful, false otherwise

Export the workbook to a file

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Export(  
    string fileName  
)
```

```
Public Function Export ( _  
    fileName As String _  
) As Boolean
```

```
public:  
bool Export(  
    String^ fileName  
)
```

▣ Parameters

fileName (String)

Output file name

▣ Return Value

true if the export was successful, false otherwise

A Sandcastle Documented Class Library **GetSheetByName Method**
(sheetName)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►
GetSheetByName(String)

Gets a Worksheet reference matching a particular name

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public Worksheet GetSheetByName(  
    string sheetName  
)
```

```
Public Function GetSheetByName ( _  
    sheetName As String _  
) As Worksheet
```

```
public:  
Worksheet^ GetSheetByName(  
    String^ sheetName  
)
```

▣ **Parameters**

sheetName (String)
Name to find

▣ **Return Value**

returns instance of matching sheet, null otherwise

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

C# ▼

[-] **Members**

Icon	Member	Description
	Import(String)	Imports a excel xml workbook into a ExcelXmlWorkbook instance
	Import(Stream)	Imports a excel xml workbook into a ExcelXmlWorkbook instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

Import(Stream)

Imports an excel xml workbook into an ExcelXmlWorkbook instance

C#

Declaration Syntax

C#

Visual Basic

Visual C++

```
public static ExcelXmlWorkbook Import(  
    Stream stream  
)
```

```
Public Shared Function Import ( _  
    stream As Stream _  
) As ExcelXmlWorkbook
```

```
public:  
static ExcelXmlWorkbook^ Import(  
    Stream^ stream  
)
```

Parameters

stream (Stream)

Stream to import

Return Value

If import was successful, the ExcelXmlWorkbook instance, null otherwise

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

Import(String)

Imports an excel xml workbook into an ExcelXmlWorkbook instance

C#

Declaration Syntax

C#

Visual Basic

Visual C++

```
public static ExcelXmlWorkbook Import(  
    string importFile  
)
```

```
Public Shared Function Import ( _  
    importFile As String _  
) As ExcelXmlWorkbook
```

```
public:  
static ExcelXmlWorkbook^ Import(  
    String^ importFile  
)
```

Parameters

importFile (String)

File to import

Return Value

If import was successful, the ExcelXmlWorkbook instance, null otherwise

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

InsertAfter()

▣ **Members**

Icon	Member	Description
	InsertAfter(Int32)	Insert a new sheet after another sheet
	InsertAfter(String)	Insert a new sheet after another sheet
	InsertAfter(Worksheet)	Insert a new sheet after another sheet

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Insert a new sheet after another sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet InsertAfter(  
    int index  
)
```

```
Public Function InsertAfter ( _  
    index As Integer _  
) As Worksheet
```

```
public:  
Worksheet^ InsertAfter(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of sheet after which new sheet will be added

▣ Return Value

New worksheet instance

▣ Remarks

If index is not in bounds, the new sheet is added to the end of all sheets

InsertAfter Method (sheetName)

Namespaces ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

InsertAfter(String)

C#

Insert a new sheet after another sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet InsertAfter(  
    string sheetName  
)
```

```
Public Function InsertAfter ( _  
    sheetName As String _  
) As Worksheet
```

```
public:  
Worksheet^ InsertAfter(  
    String^ sheetName  
)
```

▣ Parameters

sheetName (String)

Name of sheet after which new sheet will be added

▣ Return Value

New worksheet instance

▣ Remarks

If sheet is not found, the new sheet is added to the end of all sheets

Insert a new sheet after another sheet

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public Worksheet InsertAfter(Worksheet ws)</pre>		
	<pre>Public Function InsertAfter (_ ws As Worksheet _) As Worksheet</pre>	
		<pre>public: Worksheet^ InsertAfter(Worksheet^ ws)</pre>

▣ Parameters

ws (Worksheet)

Instance of sheet after which new sheet will be added

▣ Return Value

New worksheet instance

▣ Remarks

If sheet is not found, the new sheet is added to the end of all sheets

InsertBefore()

Members

Icon	Member	Description
	InsertBefore(Int32)	Insert a new sheet before another sheet
	InsertBefore(String)	Insert a new sheet before another sheet
	InsertBefore(Worksheet)	Insert a new sheet before another sheet

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Insert a new sheet before another sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet InsertBefore(  
    int index  
)
```

```
Public Function InsertBefore ( _  
    index As Integer _  
) As Worksheet
```

```
public:  
Worksheet^ InsertBefore(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of sheet before which new sheet will be added

▣ Return Value

New worksheet instance

▣ Remarks

If index is less than 0, the new sheet is added to the end of all sheets

InsertBefore Method (sheetName)

Namespaces ► [Yogesh.ExcelXml](#) ► [ExcelXmlWorkbook](#) ►

InsertBefore(String)

C#

Insert a new sheet before another sheet

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet InsertBefore(  
    string sheetName  
)
```

```
Public Function InsertBefore ( _  
    sheetName As String _  
) As Worksheet
```

```
public:  
Worksheet^ InsertBefore(  
    String^ sheetName  
)
```

Parameters

sheetName (String)

Name of sheet before which new sheet will be added

Return Value

New worksheet instance

Remarks

If sheet is not found, the new sheet is added to the end of all sheets

Insert a new sheet before another sheet

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public Worksheet InsertBefore(Worksheet ws)</pre>		
	<pre>Public Function InsertBefore (_ ws As Worksheet _) As Worksheet</pre>	
		<pre>public: Worksheet^ InsertBefore(Worksheet^ ws)</pre>

▣ Parameters

ws (Worksheet)

Instance of sheet before which new sheet will be added

▣ Return Value

New worksheet instance

▣ Remarks

If sheet is not found, the new sheet is added to the end of all sheets

Members

Icon	Member	Description
	Item[Int32]	Returns the sheet at a given position
	Item[String]	Returns the sheet by sheet name

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Returns the sheet at a given position

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet this[
    int index
] { get; }
```

```
Public ReadOnly Default Property Item ( _
    index As Integer _
) As Worksheet
```

```
public:
property Worksheet^ default[int index] {
    Worksheet^ get (int index);
}
```

▣ Parameters

index (Int32)

Index of the [Worksheet](#) starting from 0

▣ Return Value

[Worksheet](#) reference to the requested sheet

Returns the sheet by sheet name

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Worksheet this[
    string sheetName
] { get; }
```

```
Public ReadOnly Default Property Item ( _
    sheetName As String _
) As Worksheet
```

```
public:
property Worksheet^ default[String^ sheetName] {
    Worksheet^ get (String^ sheetName);
}
```

▣ Parameters

sheetName (String)

Name of [Worksheet](#)

▣ Return Value

[Worksheet](#) reference to the requested sheet

Gets or sets document properties

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DocumentProperties Properties { get; set; }
```

```
Public Property Properties As DocumentProperties
```

```
public:  
property DocumentProperties^ Properties {  
    DocumentProperties^ get ();  
    void set (DocumentProperties^ value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

The number of sheets in this book

▣ Declaration Syntax

[C#](#)[Visual Basic](#)[Visual C++](#)

```
public int SheetCount { get; }
```

```
Public ReadOnly Property SheetCount As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets cell's font options

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class FontOptions
```

```
Public Class FontOptions
```

```
public ref class FontOptions
```

Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	FontOptions()	Creates a new instance
	FontOptions(FontOptions)	Creates a new instance based on another instance
	Bold	Gets or sets font's bold property
	Color	Gets or sets font's color

	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithm and data structures like a hash table (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	Italic	Gets or sets font's italic property
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Name	Gets or sets the name of the font
	Size	Gets or sets the size of the font
	Strikeout	Gets or sets font's strike-through property

	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Underline	Gets or sets font's underline property.

Inheritance Hierarchy

Object

└─ FontOptions

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Members

Icon	Member	Description
	FontOptions()	Creates a new instance
	FontOptions(FontOptions)	Creates a new instance based on another instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Creates a new instance

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public FontOptions()
```

```
Public Sub New
```

```
public:  
FontOptions()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **FontOptions Constructor**
(fo)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [FontOptions](#) ►
FontOptions(FontOptions)

C#

Creates a new instance based on another instance

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public FontOptions(  
    FontOptions fo  
)
```

```
Public Sub New ( _  
    fo As FontOptions _  
)
```

```
public:  
FontOptions(  
    FontOptions^ fo  
)
```

▣ **Parameters**

fo (FontOptions)
Instance to copy

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets font's bold property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Bold { get; set; }
```

```
Public Property Bold As Boolean
```

```
public:  
virtual property bool Bold {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets font's color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color Color { get; set; }
```

```
Public Property Color As Color
```

```
public:  
virtual property Color Color {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets font's italic property

Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Italic { get; set; }
```

```
Public Property Italic As Boolean
```

```
public:  
virtual property bool Italic {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the name of the font

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Name As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the size of the font

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Size As Integer
```

```
public:  
virtual property int Size {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets font's strike-through property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Strikeout { get; set; }
```

```
Public Property Strikeout As Boolean
```

```
public:  
virtual property bool Strikeout {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets font's underline property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Underline { get; set; }
```

```
Public Property Underline As Boolean
```

```
public:  
virtual property bool Underline {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets cell's font options

Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class FontOptionsBase : CellSettingsApplier
```

```
Public Class FontOptionsBase _
    Inherits CellSettingsApplier
```

```
public ref class FontOptionsBase : public CellSetting
```

Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	Bold	Gets or sets font's bold property
	Color	Gets or sets font's color
	Equals(Object)	Determines whether the specified Object is equal to the current Object (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup

		operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	Italic	Gets or sets font's italic property
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Name	Gets or sets the name of the font
	Size	Gets or sets the size of the font
	Strikeout	Gets or sets font's strike-through property
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Underline	Gets or sets font's underline property

▣ Inheritance Hierarchy

Object

- └─ CellSettingsApplier
 - └─ FontOptionsBase

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



Gets or sets font's bold property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Bold { get; set; }
```

```
Public Property Bold As Boolean
```

```
public:  
virtual property bool Bold {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets font's color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color Color { get; set; }
```

```
Public Property Color As Color
```

```
public:  
virtual property Color Color {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets font's italic property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Italic { get; set; }
```

```
Public Property Italic As Boolean
```

```
public:  
virtual property bool Italic {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the name of the font

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Name As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the size of the font

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Size As Integer
```

```
public:  
virtual property int Size {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets font's strike-through property

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public bool Strikeout { get; set; }</pre>		
<pre>Public Property Strikeout As Boolean</pre>		
<pre>public: virtual property bool Strikeout { bool get () sealed; void set (bool <i>value</i>) sealed; }</pre>		

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets font's underline property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Underline { get; set; }
```

```
Public Property Underline As Boolean
```

```
public:  
virtual property bool Underline {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Formula is a formula builder class which can be stored directly in a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class Formula
```

```
Public Class Formula
```

```
public ref class Formula
```

Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	Formula(String)	Constructs a formula without any parameters
	Formula(String, Range)	Constructs a formula and adds a range as the first parameter
	Formula(String, String)	Constructs a formula and adds a string as the first parameter
	Formula(String, Formula)	Constructs a formula and adds

		another formula as the first parameter
	<code>Formula(String, Range, Predicate<Cell>)</code>	Constructs a formula and adds a filtered range as the first parameter
	<code>Add(Range)</code>	Adds a range as a parameter in a formula
	<code>Add(String)</code>	Adds a string as a parameter in a formula
	<code>Add(Formula)</code>	Adds another formula as a parameter in a formula
	<code>Add(Range, Predicate<Cell>)</code>	Adds a filtered range as a parameter
	<code>Equals(Object)</code>	Determines whether the specified Object is equal to the current Object (Inherited from Object .)
	<code>Finalize()</code>	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	<code>GetHashCode()</code>	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table.

		(Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	MustHaveParameters	Check to force parameters in function
	Parameters	Readonly list of formula parameters
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)

▣ Inheritance Hierarchy

Object

└─ Formula

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Members

Icon	Member	Description
	Formula(String)	Constructs a formula without any parameters
	Formula(String, Range)	Constructs a formula and adds a range as the first parameter
	Formula(String, String)	Constructs a formula and adds a string as the first parameter
	Formula(String, Formula)	Constructs a formula and adds another formula as the first parameter
	Formula(String, Range, Predicate<Cell>)	Constructs a formula and adds a filtered range as the first parameter

(function)

Namespaces ► [Yogesh.ExcelXml](#) ► [Formula](#) ► Formula(String)

C#

Constructs a formula without any parameters

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public Formula(string <i>function</i>)</pre>		

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

```
public:  
Formula(  
    String^ function  
)
```

▣ Parameters

function (String)
Function name

▣ Examples

C# Cop

```
Formula formula = new Formula("sum");
```

A Sandcastle Documented Class Library

Formula Constructor

(function, parameter)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Formula\(String, String\)](#)

Constructs a formula and adds a string as the first parameter

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Formula(  
    string function,  
    string parameter  
)
```

```
Public Sub New ( _  
    function As String, _  
    parameter As String _  
)
```

```
public:  
Formula(  
    String^ function,  
    String^ parameter  
)
```

Parameters

function (String)

Function name

parameter (String)

String to add as parameter

Examples

C#

Cop

```
Formula formula = new Formula("sum", "0,1");
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **Formula Constructor**
(function, formula)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Formula\(String, Formula\)](#)

Constructs a formula and adds another formula as the first parameter C# ▼

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Formula(  
    string function,  
    Formula formula  
)
```

```
Public Sub New ( _  
    function As String, _  
    formula As Formula _  
)
```

```
public:  
Formula(  
    String^ function,  
    Formula^ formula  
)
```

▣ Parameters

function (String)

Function name

formula (Formula)

Another formula to add to this formula's parameter list

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

(function, range)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Formula\(String, Range\)](#)

C#

Constructs a formula and adds a range as the first parameter

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Formula(  
    string function,  
    Range range  
)
```

```
Public Sub New ( _  
    function As String, _  
    range As Range _  
)
```

```
public:  
Formula(  
    String^ function,  
    Range^ range  
)
```

▣ Parameters

function (String)

Function name

range (Range)

Range to add as parameter

▣ Examples

C#

Cop

```
Formula formula = new Formula("sum", new Range(cell1
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



A Sandcastle Documented Class Library

Formula Constructor (function, range, cellCompare)

Namespaces ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Formula\(String, Range, Predicate<Cell>\)](#)

Constructs a formula and adds a filtered range as the first parameter

C#

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Formula(  
    string function,  
    Range range,  
    Predicate<Cell> cellCompare  
)
```

```
Public Sub New ( _  
    function As String, _  
    range As Range, _  
    cellCompare As Predicate(Of Cell) _  
)
```

```
public:  
Formula(  
    String^ function,  
    Range^ range,  
    Predicate<Cell>^ cellCompare  
)
```

Parameters

function (String)

Function name

range (Range)

Range to add as parameter

cellCompare (Predicate<Cell>)

A custom defined to compare the values of the range

▣ Remarks

Custom delegates can filter all cells and auto add them to the parameter list of a formula by passing a `System.Predicate>Cell<`, i.e. a delegate which accepts `Cell` as its value and returns `bool` to both `Formula` constructor or `Add`. All the values accessors (i.e. `Value`, `NumericValue` etc.) and cell style can be checked.

▣ Examples

Lets assume column 1,2,3,6 and 7 are bold...

C#

Cop

```
XmlStyle style = new XmlStyle();
style.Font.Bold = true;

// VS2008 style
sheet[7, 3].Value = new Formula("sum", new Range(sheet,
    cell => cell.Style == style));

// or VS2005 style
sheet[7, 3].Value = new Formula("sum", new Range(sheet,
    delegate (Cell cell) { return cell.Style ==
```

In the first example of style, the value of the cell will be =SUM(A4:C4, F4:G4).

Continuous ranges matching to true will be joined as one parameter, i.e. A4:C4 and not as separate parameters, i.e. A4,B4,C4

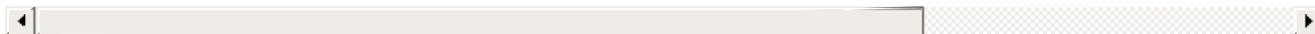
Using value accessors...

C#

Cop

```
sheet[7, 3].Value = new Formula("sum", new Range(sheet,
    cell => cell.NumericValue > 10000 & cell
```

2.89.501.2158



Members

Icon	Member	Description
	Add(Range)	Adds a range as a parameter in a formula
	Add(String)	Adds a string as a parameter in a formula
	Add(Formula)	Adds another formula as a parameter in a formula
	Add(Range, Predicate<Cell>)	Adds a filtered range as a parameter

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **Add Method (parameter)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Add\(String\)](#)

Adds a string as a parameter in a formula

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public void Add(  
    string parameter  
)
```

```
Public Sub Add ( _  
    parameter As String _  
)
```

```
public:  
void Add(  
    String^ parameter  
)
```

▣ **Parameters**

parameter (String)
String to add as parameter

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **Add Method (formula)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Add\(Formula\)](#)

Adds another formula as a parameter in a formula

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public void Add(  
    Formula formula  
)
```

```
Public Sub Add ( _  
    formula As Formula _  
)
```

```
public:  
void Add(  
    Formula^ formula  
)
```

▣ **Parameters**

formula (Formula)

Another formula to add to this formula's parameter list

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Adds a range as a parameter in a formula

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Add(  
    Range range  
)
```

```
Public Sub Add ( _  
    range As Range _  
)
```

```
public:  
void Add(  
    Range^ range  
)
```

▣ Parameters

range (Range)

Range to add as parameter

A Sandcastle Documented Class Library **Add Method (range, cellCompare)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Formula](#) ► [Add\(Range, Predicate<Cell>\)](#)

Adds a filtered range as a parameter

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void Add(  
    Range range,  
    Predicate<Cell> cellCompare  
)
```

```
Public Sub Add ( _  
    range As Range, _  
    cellCompare As Predicate(Of Cell) _  
)
```

```
public:  
void Add(  
    Range^ range,  
    Predicate<Cell^>^ cellCompare  
)
```

▣ **Parameters**

range (Range)

Range to add as parameter

cellCompare (Predicate<Cell>)

A custom defined cell to compare the values of the range

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Property

Namespaces ► [Yogesh.ExcelXml](#) ► [Formula](#) ► MustHaveParameters

C# ▼

Check to force parameters in function

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool MustHaveParameters { get; set; }
```

```
Public Property MustHaveParameters As Boolean
```

```
public:  
property bool MustHaveParameters {  
    bool get ();  
    void set (bool value);  
}
```

▣ Remarks

In case if this flag is set and formula does not contain one or more parameters then when the formula is assigned to a cell, the cell is left empty.

▣ Examples

C#

Cop

```
sheet[0, 0].Value = 2;  
sheet[1, 0].Value = 12;  
sheet[2, 0].Value = 9;  
sheet[3, 0].Value = 7;
```

```
Formula formula1 = new Formula("Sum", new Range(sheet  
    delegate (Cell cell) { retur
```

```
formula1.MustHaveParameters = false; // default value
sheet[4, 0].Value = formula; // cell value will be '

Formula formula2 = new Formula("Sum", new Range(sheet
    delegate (Cell cell) { return

formula2.MustHaveParameters = true;
sheet[5, 0].Value = formula; // cell will be empty
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



Readonly list of formula paramters

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<Parameter> Parameters { get; }
```

```
Public ReadOnly Property Parameters As IList(Of Para
```

```
public:  
property IList<Parameter^>^ Parameters {  
    IList<Parameter^>^ get ();  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



Enumeration

Namespaces ► [Yogesh.ExcelXml](#) ► HorizontalAlignment

Cell's horizontal alignment values

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum HorizontalAlignment
```

```
Public Enumeration HorizontalAlignment
```

```
public enum class HorizontalAlignment
```

▣ Members

Member	Description
None	None
Left	Left aligned
Center	Centered
Right	Right aligned
Fill	Stretched to fill
Justify	Justified

Distributed	Distributed

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets cell's interior options

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class InteriorOptions
```

```
Public Class InteriorOptions
```

```
public ref class InteriorOptions
```

Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	InteriorOptions()	Creates a new instance
	InteriorOptions(InteriorOptions)	Creates a new instance based on another instance
	Color	Gets or sets cell background color
	Equals(Object)	Determines whether the specified Object is equal to the current

		<p>Object.</p> <p>(Inherited from Object.)</p>
	Finalize()	<p>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.</p> <p>(Inherited from Object.)</p>
	GetHashCode()	<p>Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table.</p> <p>(Inherited from Object.)</p>
	GetType()	<p>Gets the Type of the current instance.</p> <p>(Inherited from Object.)</p>
	MemberwiseClone()	<p>Creates a shallow copy of the current Object.</p> <p>(Inherited from Object.)</p>
	Pattern	Gets or sets cell pattern
	PatternColor	Gets or sets cell pattern color
	ToString()	<p>Returns a String that represents the current Object.</p> <p>(Inherited from Object.)</p>

▣ Inheritance Hierarchy

Object

└─ InteriorOptions

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

▣ **Members**

Icon	Member	Description
	InteriorOptions()	Creates a new instance
	InteriorOptions(InteriorOptions)	Creates a new instance based on another instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InteriorOptions()

Creates a new instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public InteriorOptions()
```

```
Public Sub New
```

```
public:  
InteriorOptions()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InteriorOptions Constructor (io)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [InteriorOptions](#) ►

InteriorOptions(InteriorOptions)

C# ▼

Creates a new instance based on another instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public InteriorOptions(  
    InteriorOptions io  
)
```

```
Public Sub New ( _  
    io As InteriorOptions _  
)
```

```
public:  
InteriorOptions(  
    InteriorOptions^ io  
)
```

▣ Parameters

io (InteriorOptions)
Instance to copy

Gets or sets cell background color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color Color { get; set; }
```

```
Public Property Color As Color
```

```
public:  
virtual property Color Color {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets cell pattern

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Pattern Pattern { get; set; }
```

```
Public Property Pattern As Pattern
```

```
public:  
virtual property Pattern Pattern {  
    Pattern get () sealed;  
    void set (Pattern value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets cell pattern color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color PatternColor { get; set; }
```

```
Public Property PatternColor As Color
```

```
public:  
virtual property Color PatternColor {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

C# ▼

Gets or sets cell's interior options

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class InteriorOptionsBase : CellSettingsAppli
```

```
Public Class InteriorOptionsBase _
    Inherits CellSettingsApplier
```

```
public ref class InteriorOptionsBase : public CellSe
```

Members

All Members

Methods

Properties

 Public Protected Instance Static

Declare

Inherited

Icon	Member	Description
	Color	Gets or sets cell background color
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)

	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Pattern	Gets or sets cell background color
	PatternColor	Gets or sets cell pattern color
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)

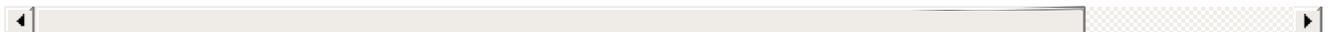
▣ Inheritance Hierarchy

Object

└─ [CellSettingsApplier](#)

└─ InteriorOptionsBase

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158



Gets or sets cell background color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color Color { get; set; }
```

```
Public Property Color As Color
```

```
public:  
virtual property Color Color {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets cell background color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Pattern Pattern { get; set; }
```

```
Public Property Pattern As Pattern
```

```
public:  
virtual property Pattern Pattern {  
    Pattern get () sealed;  
    void set (Pattern value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets cell pattern color

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Color PatternColor { get; set; }
```

```
Public Property PatternColor As Color
```

```
public:  
virtual property Color PatternColor {  
    Color get () sealed;  
    void set (Color value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Page layout

[-] **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public enum PageLayout
```

```
Public Enumeration PageLayout
```

```
public enum class PageLayout
```

[-] **Members**

Member	Description
None	None
CenterHorizontal	Centers the page horizontally
CenterVertical	Centers the page vertically
CenterVerticalAndHorizontal	Centers the page vertically and horizontally

Enumeration

Namespaces ► [Yogesh.ExcelXml](#) ► PageOrientation

C#

Orientation mode

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum PageOrientation
```

```
Public Enumeration PageOrientation
```

```
public enum class PageOrientation
```

▣ Members

Member	Description
None	None.
Landscape	Landscape orientation
Portrait	Portrait orientation

Parameter denotes a single parameter in a formula

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class Parameter
```

```
Public Class Parameter
```

```
public ref class Parameter
```

Members

All Members

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table.

		(Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	ParameterType	Parameter type
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Value	Value of the parameter

Inheritance Hierarchy

Object

└─ Parameter

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Parameter type

[-] **Declaration Syntax**

C#

Visual Basic

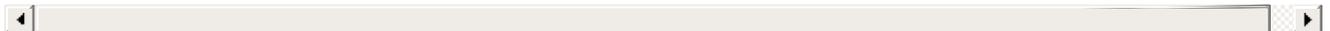
Visual C++

```
public ParameterType ParameterType { get; private se
```

```
Public Property ParameterType As ParameterType
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



Value of the parameter

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object Value { get; private set; }
```

```
Public Property Value As Object
```

```
public:  
property Object^ Value {  
    Object^ get ();  
    void set (Object^ value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Formula parameter types

[-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum ParameterType
```

```
Public Enumeration ParameterType
```

```
public enum class ParameterType
```

[-] Members

Member	Description
String	A unknown string parameter
Range	A Range
Formula	A Formula

Different types of cell background patterns

[-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum Pattern
```

```
Public Enumeration Pattern
```

```
public enum class Pattern
```

[-] Members

Member	Description
Solid	Solid
Gray25	Gray25
Gray50	Gray50
Gray75	Gray75
Gray125	Gray125
Gray0625	Gray0625

HorzStripe	HorzStripe
VertStripe	VertStripe
ReverseDiagStripe	ReverseDiagStripe
DiagStripe	DiagStripe
DiagCross	DiagCross
ThickDiagCross	ThickDiagCross
ThinHorzStripe	ThinHorzStripe
ThinVertStripe	ThinVertStripe
ThinReverseDiagStripe	ThinReverseDiagStripe
ThinDiagStripe	ThinDiagStripe
ThinHorzCross	ThinHorzCross
ThinDiagCross	ThinDiagCross

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets various sheet printing options

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class PrintOptions
```

```
Public Class PrintOptions
```

```
public ref class PrintOptions
```

Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	PrintOptions()	
	Equals(Object)	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 particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	Layout	Gets or sets page layout
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Orientation	Gets or sets page orientation
	ResetHeaders()	Resets header rows/columns.
	ResetMargins()	Resets print margins
	SetFitToPage(Int32, Int32)	Sets excel's fit to page property
	SetHeaderFooterMargins(Double, Double)	Sets print header and footer margins

	<code>SetMargins(Double, Double, Double, Double)</code>	Sets print margins
	<code>SetScaleToSize(Int32)</code>	Sets excel's scale or zoom property
	<code>SetTitleColumns(Int32, Int32)</code>	Sets print header columns which are repeated at left on every page
	<code>SetTitleRows(Int32, Int32)</code>	Sets print header rows which are repeated at top on every page
	<code>ToString()</code>	Returns a String that represents the current Object (Inherited from Object .)

Inheritance Hierarchy

Object

└─ PrintOptions

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

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

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public PrintOptions()
```

```
Public Sub New
```

```
public:  
PrintOptions()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets page layout

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public PageLayout Layout { get; set; }
```

```
Public Property Layout As PageLayout
```

```
public:  
property PageLayout Layout {  
    PageLayout get ();  
    void set (PageLayout value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets page orientation

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public PageOrientation Orientation { get; set; }
```

```
Public Property Orientation As PageOrientation
```

```
public:  
property PageOrientation Orientation {  
    PageOrientation get ();  
    void set (PageOrientation value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Resets header rows/columns.

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void ResetHeaders()
```

```
Public Sub ResetHeaders
```

```
public:  
void ResetHeaders()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Resets print margins

[-] **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void ResetMargins()
```

```
Public Sub ResetMargins
```

```
public:  
void ResetMargins()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **SetFitToPage Method (width, height)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [PrintOptions](#) ►

SetFitToPage(Int32, Int32)

C#

Sets excel's fit to page property

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetFitToPage(  
    int width,  
    int height  
)
```

```
Public Sub SetFitToPage ( _  
    width As Integer, _  
    height As Integer _  
)
```

```
public:  
void SetFitToPage(  
    int width,  
    int height  
)
```

▣ Parameters

width (Int32)

Number of pages to fit the page horizontally

height (Int32)

Number of pages to fit the page vertically

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Method (header, footer)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [PrintOptions](#) ►

SetHeaderFooterMargins(Double, Double)

C# ▼

Sets print header and footer margins

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetHeaderFooterMargins(  
    double header,  
    double footer  
)
```

```
Public Sub SetHeaderFooterMargins ( _  
    header As Double, _  
    footer As Double _  
)
```

```
public:  
void SetHeaderFooterMargins(  
    double header,  
    double footer  
)
```

▣ Parameters

header (Double)

Header margin

footer (Double)

Footer margin

A Sandcastle Documented Class Library **SetMargins Method (left, top, right, bottom)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [PrintOptions](#) ►

SetMargins(Double, Double, Double, Double)

Sets print margins

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void SetMargins(  
    double left,  
    double top,  
    double right,  
    double bottom  
)
```

```
Public Sub SetMargins ( _  
    left As Double, _  
    top As Double, _  
    right As Double, _  
    bottom As Double _  
)
```

```
public:  
void SetMargins(  
    double left,  
    double top,  
    double right,  
    double bottom  
)
```

▣ **Parameters**

left (Double)

Left margin

top (Double)
Top margin

right (Double)
Right margin

bottom (Double)
Bottom margin

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library

SetScaleToSize Method (scale)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [PrintOptions](#) ► [SetScaleToSize\(Int32\)](#)

Sets excel's scale or zoom property

C# ▼

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetScaleToSize(  
    int scale  
)
```

```
Public Sub SetScaleToSize ( _  
    scale As Integer _  
)
```

```
public:  
void SetScaleToSize(  
    int scale  
)
```

▣ Parameters

scale (Int32)
Scale to size

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library

SetTitleColumns Method (left, right)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [PrintOptions](#) ►

SetTitleColumns(Int32, Int32)

Sets print header columns which are repeated at left on every page

C#

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetTitleColumns(  
    int left,  
    int right  
)
```

```
Public Sub SetTitleColumns ( _  
    left As Integer, _  
    right As Integer _  
)
```

```
public:  
void SetTitleColumns(  
    int left,  
    int right  
)
```

Parameters

left (Int32)

Left print column

right (Int32)

Right print column

Remarks

Important Note: Left and right column parameters are **NOT** zero based like row and column indexes

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **SetTitleRows Method (top, bottom)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [PrintOptions](#) ►

SetTitleRows(Int32, Int32)

C#

Sets print header rows which are repeated at top on every page

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetTitleRows(  
    int top,  
    int bottom  
)
```

```
Public Sub SetTitleRows ( _  
    top As Integer, _  
    bottom As Integer _  
)
```

```
public:  
void SetTitleRows(  
    int top,  
    int bottom  
)
```

▣ Parameters

top (Int32)

Top print row

bottom (Int32)

Bottom print row

▣ Remarks

Important Note: Top and bottom row parameters are **NOT** zero based lil row and column indexers

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Defines a range of cells

Declaration Syntax

C#

Visual Basic

Visual C++

```
public class Range : Styles, IEnumerable<Cell>,
    IEnumerable
```

```
Public Class Range _
    Inherits Styles _
    Implements IEnumerable(Of Cell), IEnumerable
```

```
public ref class Range : public Styles,
    IEnumerable<Cell^>, IEnumerable
```

Members

All Members

Constructors

Methods

Properties

<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	Range(Cell)	Defines a range
	Range(Cell, Cell)	Defines a range
	Absolute	Gets or sets the range's flag to return a absolute reference or otherwise

	Alignment	Gets or sets cell alignment options (Inherited from Styles .)
	AutoFilter()	Sets this range as a auto-filter range in the sheet
	Border	Gets or sets border settings (Inherited from Styles .)
	ColumnCount	Gets the number of columns in a range
	Contains(Cell)	Checks if a particular cell is present in a range or not
	CustomFormatString	Gets or sets custom display format string (Inherited from Styles .)
	DisplayFormat	Gets or sets the cell display format (Inherited from Styles .)
	Equals(Object)	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 .)

	Font	Gets or sets the font options (Inherited from Styles .)
	GetEnumerator()	Get a cell enumerator
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	Interior	Gets or sets interior options (Inherited from Styles .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Merge()	Merges a range into one cell
	Name	Gets or sets the name of the range
	RowCount	Gets the number of rows in a range
	SetAsPrintArea()	Sets this range as the current print area in the sheet

	Style	Returns the XmlStyle reference of the cell (Inherited from Styles .)
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)
	Unmerge()	Unmerges a merged range

▣ Inheritance Hierarchy

Object

- └─ [CellSettingsApplier](#)
- └─ [Styles](#)
- └─ Range

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Members

Icon	Member	Description
	Range(Cell)	Defines a range
	Range(Cell, Cell)	Defines a range

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Defines a range

▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public Range(  
    Cell cell  
)
```

```
Public Sub New ( _  
    cell As Cell _  
)
```

```
public:  
Range(  
    Cell^ cell  
)
```

▣ Parameters

cell (Cell)

A single cell as a range

(cellFrom, cellTo)

Namespaces ► [Yogesh.ExcelXml](#) ► [Range](#) ► Range(Cell, Cell)

C# ▼

Defines a range

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public Range(Cell cellFrom, Cell cellTo)</pre>		

```
Public Sub New ( _  
    cellFrom As Cell, _  
    cellTo As Cell _  
)
```

```
public:  
Range(  
    Cell^ cellFrom,  
    Cell^ cellTo  
)
```

▣ Parameters

cellFrom (Cell)
Starting cell

cellTo (Cell)
Ending cell

▣ Remarks

Defines a rectangular area of a sheet with a starting cell and a ending c

2.89.501.2158

Gets or sets the range's flag to return a absolute reference or otherwise

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Absolute { get; set; }
```

```
Public Property Absolute As Boolean
```

```
public:  
property bool Absolute {  
    bool get ();  
    void set (bool value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Sets this range as a auto-filter range in the sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void AutoFilter()
```

```
Public Sub AutoFilter
```

```
public:  
void AutoFilter()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets the number of columns in a range

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int ColumnCount { get; }
```

```
Public ReadOnly Property ColumnCount As Integer
```

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

▣ Return Value

Number of columns in a range

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

C# ▼

Checks if a particular cell is present in a range or not

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Contains(  
    Cell cell  
)
```

```
Public Function Contains ( _  
    cell As Cell _  
) As Boolean
```

```
public:  
bool Contains(  
    Cell^ cell  
)
```

▣ Parameters

cell (Cell)

Cell to check

▣ Return Value

true if cell is present, false otherwise

Get a cell enumerator

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerator<Cell> GetEnumerator()
```

```
Public Function GetEnumerator As IEnumerator(Of Cell
```

```
public:  
virtual IEnumerator<Cell^> GetEnumerator() sealed
```

▣ Return Value

returns IEnumerator<Cell>

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Merges a range into one cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Merge()
```

```
Public Function Merge As Boolean
```

```
public:  
bool Merge()
```

▣ Return Value

true if merge was successful, false otherwise

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the name of the range

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Name As String
```

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

▣ Remarks

This property always adds global (i.e. Workbook level) named ranges. To add sheet limited ranges, use [AddNamedRange\(Range, String\)](#) method of [Worksheet](#) class.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets the number of rows in a range

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int RowCount { get; }
```

```
Public ReadOnly Property RowCount As Integer
```

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

▣ Return Value

Number of rows in a range

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Sets this range as the current print area in the sheet

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void SetAsPrintArea()
```

```
Public Sub SetAsPrintArea
```

```
public:  
void SetAsPrintArea()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Unmerges a merged range

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Unmerge()
```

```
Public Sub Unmerge
```

```
public:  
void Unmerge()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Row class represents a single row in a worksheet

[-] **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public class Row : Styles, IEnumerable<Cell>, IEnumerable</pre>		

```
Public Class Row _
    Inherits Styles _
    Implements IEnumerable(Of Cell), IEnumerable
```

```
public ref class Row : public Styles,
    IEnumerable<Cell^>, IEnumerable
```

[-] **Members**

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	AddCell()	Adds a cells to the end of the row
	Alignment	Gets or sets cell alignment options (Inherited from Styles .)
	Border	Gets or sets border settings (Inherited from Styles .)

	CellCount	Returns the number of cell in a row
	CustomFormatString	Gets or sets custom display format string (Inherited from Styles .)
	Delete()	Deletes the row from the parent sheet
	DeleteCell(Int32)	Deletes a cell
	DeleteCell(Cell)	Deletes a cell
	DeleteCell(Int32, Boolean)	Deletes a cell
	DeleteCell(Cell, Boolean)	Deletes a cell
	DeleteCells(Int32, Int32, Boolean)	Delete a specific number of cells starting from a cell index
	DeleteCells(Cell, Int32, Boolean)	Delete a specific number of cells starting from a cell instance
	DeleteCells(Int32, Int32)	Delete a specific number of cells starting from a cell index
	DeleteCells(Cell, Int32)	Delete a specific number of cells

		starting from a cell instance
	DisplayFormat	Gets or sets the cell display format (Inherited from Styles.)
	Equals(Object)	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.)
	Font	Gets or sets the font options (Inherited from Styles.)
	GetEnumerator()	Get a cell enumerator
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table (Inherited from Object.)
	GetType()	Gets the Type of the current instance (Inherited from Object.)
	Height	Gets or sets the row height
	Hidden	Row is hidden?

	InsertCellAfter(Int32)	Inserts a cell after another cell
	InsertCellAfter(Cell)	Inserts a cell after another cell
	InsertCellBefore(Int32)	Inserts a cell before another cell
	InsertCellBefore(Cell)	Inserts a cell before another cell
	InsertCellsAfter(Int32, Int32)	Inserts a specific number of cells after a cell
	InsertCellsAfter(Cell, Int32)	Inserts a specific number of cells after a cell
	InsertCellsBefore(Int32, Int32)	Inserts a specific number of cells before a cell
	InsertCellsBefore(Cell, Int32)	Inserts a specific number of cells before a cell
	Interior	Gets or sets interior options (Inherited from Styles.)
	Item[Int32]	Returns the cell at a given position

	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Style	Returns the XmlStyle reference of the cell (Inherited from Styles .)
	ToString()	Returns a String that represents the current Object . (Inherited from Object .)

▣ **Remarks**

Row class represents a single row in a worksheet.

You cannot directly declare an instance of a row from your code by using the **new** keyword. The only way to access a row is to retrieve it from a worksheet.

▣ **Inheritance Hierarchy**

Object

└─ **CellSettingsApplier**

└─ **Styles**

└─ **Row**

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Adds a cells to the end of the row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Cell AddCell()
```

```
Public Function AddCell As Cell
```

```
public:  
Cell^ AddCell()
```

▣ Return Value

Instance of the newly created cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Returns the number of cell in a row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int CellCount { get; }
```

```
Public ReadOnly Property CellCount As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Deletes the row from the parent sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Delete()
```

```
Public Sub Delete
```

```
public:  
void Delete()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Members

Icon	Member	Description
	DeleteCell(Int32)	Deletes a cell
	DeleteCell(Cell)	Deletes a cell
	DeleteCell(Int32, Boolean)	Deletes a cell
	DeleteCell(Cell, Boolean)	Deletes a cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Deletes a cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteCell(  
    int index  
)
```

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

```
public:  
void DeleteCell(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of cell to delete

A Sandcastle Documented Class Library **DeleteCell Method (index, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Row](#) ► [DeleteCell\(Int32, Boolean\)](#)

Deletes a cell

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void DeleteCell(  
    int index,  
    bool cascade  
)
```

```
Public Sub DeleteCell ( _  
    index As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteCell(  
    int index,  
    bool cascade  
)
```

▣ **Parameters**

index (Int32)

Index of cell to delete

cascade (Boolean)

if true, the cell is removed and cells to the right are cascaded leftwards. if false, the cell is only emptied

Deletes a cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteCell(  
    Cell cell  
)
```

```
Public Sub DeleteCell ( _  
    cell As Cell _  
)
```

```
public:  
void DeleteCell(  
    Cell^ cell  
)
```

▣ Parameters

cell (Cell)

Instance of cell to delete

A Sandcastle Documented Class Library **DeleteCell Method (cell, cascade)**

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► DeleteCell(Cell, Boolean)

Deletes a cell

C# ▼

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public void DeleteCell(Cell cell, bool cascade)</pre>		

```
Public Sub DeleteCell ( _  
    cell As Cell, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteCell(  
    Cell^ cell,  
    bool cascade  
)
```

▣ **Parameters**

cell (Cell)

Instance of cell to delete

cascade (Boolean)

if true, the cell is removed and cells to the right are cascaded leftwards. if false, the cell is only emptied

Members

Icon	Member	Description
	DeleteCells(Int32, Int32, Boolean)	Delete a specific number of cells starting from a cell index
	DeleteCells(Cell, Int32, Boolean)	Delete a specific number of cells starting from a cell instance
	DeleteCells(Int32, Int32)	Delete a specific number of cells starting from a cell index
	DeleteCells(Cell, Int32)	Delete a specific number of cells starting from a cell instance

A Sandcastle Documented Class Library **DeleteCells Method (index, numberOfCells)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Row](#) ► [DeleteCells\(Int32, Int32\)](#)

Delete a specific number of cells starting from a cell index

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public void DeleteCells(int index, int numberOfCells)</pre>		

```
Public Sub DeleteCells ( _  
    index As Integer, _  
    numberOfCells As Integer _  
)
```

```
public:  
void DeleteCells(  
    int index,  
    int numberOfCells  
)
```

▣ **Parameters**

index (Int32)

Index of cell from which the cells are deleted

numberOfCells (Int32)

Number of cells to delete

▣ **Remarks**

The cells are removed and cells to the right are cascaded leftwards.

2.89.501.2158

A Sandcastle Documented Class Library **DeleteCells Method (index, numberOfCells, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Row](#) ► DeleteCells(Int32, Int32, Boolean)

Delete a specific number of cells starting from a cell index

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteCells(  
    int index,  
    int numberOfCells,  
    bool cascade  
)
```

```
Public Sub DeleteCells ( _  
    index As Integer, _  
    numberOfCells As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteCells(  
    int index,  
    int numberOfCells,  
    bool cascade  
)
```

▣ Parameters

index (Int32)

Index of cell from which the cells are deleted

numberOfCells (Int32)

Number of cells to delete

cascade (Boolean)

if true, the cells are removed and cells to the right are cascaded leftwards. if false, the cells are only emptied

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DeleteCells Method (cell, numberOfCells)**

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [DeleteCells\(Cell, Int32\)](#)

Delete a specific number of cells starting from a cell instance

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void DeleteCells(  
    Cell cell,  
    int numberOfCells  
)
```

```
Public Sub DeleteCells ( _  
    cell As Cell, _  
    numberOfCells As Integer _  
)
```

```
public:  
void DeleteCells(  
    Cell^ cell,  
    int numberOfCells  
)
```

▣ **Parameters**

cell (Cell)

Instance of cell from which the cells are deleted

numberOfCells (Int32)

Number of cells to delete

▣ **Remarks**

The cells are removed and cells to the right are cascaded leftwards.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:

2.89.501.2158

A Sandcastle Documented Class Library **DeleteCells Method (cell, numberOfCells, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Row](#) ► DeleteCells(Cell, Int32, Boolean)

Delete a specific number of cells starting from a cell instance

C# ▼

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void DeleteCells(  
    Cell cell,  
    int numberOfCells,  
    bool cascade  
)
```

```
Public Sub DeleteCells ( _  
    cell As Cell, _  
    numberOfCells As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteCells(  
    Cell^ cell,  
    int numberOfCells,  
    bool cascade  
)
```

▣ **Parameters**

cell (Cell)

Instance of cell from which the cells are deleted

numberOfCells (Int32)

Number of cells to delete

cascade (Boolean)

if true, the cells are removed and cells to the right are cascaded leftwards. if false, the cells are only emptied

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Get a cell enumerator

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerator<Cell> GetEnumerator()
```

```
Public Function GetEnumerator As IEnumerator(Of Cell
```

```
public:  
virtual IEnumerator<Cell^> GetEnumerator() sealed
```

▣ Return Value

returns IEnumerator<Cell>

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the row height

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public double Height { get; set; }
```

```
Public Property Height As Double
```

```
public:  
property double Height {  
    double get ();  
    void set (double value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Row is hidden?

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Hidden { get; set; }
```

```
Public Property Hidden As Boolean
```

```
public:  
property bool Hidden {  
    bool get ();  
    void set (bool value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

InsertCellAfter Method

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellAfter\(\)](#)

Members

Icon	Member	Description
	InsertCellAfter(Int32)	Inserts a cell after another cell
	InsertCellAfter(Cell)	Inserts a cell after another cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

(index)Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellAfter\(Int32\)](#)

C# ▼

Inserts a cell after another cell

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public Cell InsertCellAfter(  
    int index  
)
```

```
Public Function InsertCellAfter ( _  
    index As Integer _  
) As Cell
```

```
public:  
Cell^ InsertCellAfter(  
    int index  
)
```

▣ **Parameters****index (Int32)**

Index of cell after which the cell is to be inserted

▣ **Return Value**

[Missing <returns> documentation for M:Yogesh.ExcelXml.Row.InsertCellAfter(System.Int32)]

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Inserts a cell after another cell

▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public Cell InsertCellAfter(  
    Cell cell  
)
```

```
Public Function InsertCellAfter ( _  
    cell As Cell _  
) As Cell
```

```
public:  
Cell^ InsertCellAfter(  
    Cell^ cell  
)
```

▣ Parameters

cell (Cell)

Instance of cell after which the cell is to be inserted

▣ Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Row.InsertCellAfter(Yogesh.ExcelXml.Cell)]

[-] **Members**

Icon	Member	Description
	InsertCellBefore(Int32)	Inserts a cell before another cell
	InsertCellBefore(Cell)	Inserts a cell before another cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertCellBefore Method

(index)

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellBefore\(Int32\)](#)

C# ▼

Inserts a cell before another cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Cell InsertCellBefore(  
    int index  
)
```

```
Public Function InsertCellBefore ( _  
    index As Integer _  
) As Cell
```

```
public:  
Cell^ InsertCellBefore(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of cell before which the cell is to be inserted

▣ Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Row.InsertCellBefore(System.Int32)]

InsertCellBefore Method (cell)

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellBefore\(Cell\)](#)

C# ▼

Inserts a cell before another cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Cell InsertCellBefore(  
    Cell cell  
)
```

```
Public Function InsertCellBefore ( _  
    cell As Cell _  
) As Cell
```

```
public:  
Cell^ InsertCellBefore(  
    Cell^ cell  
)
```

Parameters

cell (Cell)

Instance of cell before which the cell is to be inserted

Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Row.InsertCellBefore(Yogesh.ExcelXml.Cell)]

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Members

Icon	Member	Description
	InsertCellsAfter(Int32, Int32)	Inserts a specific number of cells after a cell
	InsertCellsAfter(Cell, Int32)	Inserts a specific number of cells after a cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertCellsAfter Method (index, cells)

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellsAfter\(Int32, Int32\)](#)

C#

Inserts a specific number of cells after a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertCellsAfter(  
    int index,  
    int cells  
)
```

```
Public Sub InsertCellsAfter ( _  
    index As Integer, _  
    cells As Integer _  
)
```

```
public:  
void InsertCellsAfter(  
    int index,  
    int cells  
)
```

Parameters

index (Int32)

Index of cell after which the cells are to be inserted

cells (Int32)

Number of cells to insert

InsertCellsAfter Method (cell, cells)

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellsAfter\(Cell, Int32\)](#)

Inserts a specific number of cells after a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertCellsAfter(  
    Cell cell,  
    int cells  
)
```

```
Public Sub InsertCellsAfter ( _  
    cell As Cell, _  
    cells As Integer _  
)
```

```
public:  
void InsertCellsAfter(  
    Cell^ cell,  
    int cells  
)
```

Parameters

cell ([Cell](#))

Instance of cell after which the cells are to be inserted

cells ([Int32](#))

Number of cells to insert

Members

Icon	Member	Description
	InsertCellsBefore(Int32, Int32)	Inserts a specific number of cells before a cell
	InsertCellsBefore(Cell, Int32)	Inserts a specific number of cells before a cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertCellsBefore Method (index, cells)

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellsBefore\(Int32, Int32\)](#)

Inserts a specific number of cells before a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertCellsBefore(  
    int index,  
    int cells  
)
```

```
Public Sub InsertCellsBefore ( _  
    index As Integer, _  
    cells As Integer _  
)
```

```
public:  
void InsertCellsBefore(  
    int index,  
    int cells  
)
```

Parameters

index (Int32)

Index of cell before which the cells are to be inserted

cells (Int32)

Number of cells to insert

A Sandcastle Documented Class Library

InsertCellsBefore Method (cell, cells)

Namespaces ► [Yogesh.ExcelXml](#) ► [Row](#) ► [InsertCellsBefore\(Cell, Int32\)](#)

C#

Inserts a specific number of cells before a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertCellsBefore(  
    Cell cell,  
    int cells  
)
```

```
Public Sub InsertCellsBefore ( _  
    cell As Cell, _  
    cells As Integer _  
)
```

```
public:  
void InsertCellsBefore(  
    Cell^ cell,  
    int cells  
)
```

Parameters

cell (Cell)

Instance of cell before which the cells are to be inserted

cells (Int32)

Number of cells to insert

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Returns the cell at a given position

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Cell this[
    int colIndex
] { get; }
```

```
Public ReadOnly Default Property Item ( _
    colIndex As Integer _
) As Cell
```

```
public:
property Cell^ default[int colIndex] {
    Cell^ get (int colIndex);
}
```

▣ Parameters

colIndex (Int32)

Index of the [Cell](#) starting from 0

▣ Return Value

[Cell](#) reference to the requested cell

Style class for cells, rows and worksheets

Declaration Syntax

C# Visual Basic Visual C++

```
public abstract class Styles : CellSettingsApplier
```

```
Public MustInherit Class Styles _
    Inherits CellSettingsApplier
```

```
public ref class Styles abstract : public CellSettin
```

Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	Alignment	Gets or sets cell alignment options
	Border	Gets or sets border settings
	CustomFormatString	Gets or sets custom display format string
	DisplayFormat	Gets or sets the cell display format

	Equals(Object)	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 .)
	Font	Gets or sets the font options
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance (Inherited from Object .)
	Interior	Gets or sets interior options
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Style	Returns the XmlStyle reference of the cell
	ToString()	Returns a String that represents the

current **Object**.

(Inherited from **Object**.)

▣ Inheritance Hierarchy

Object

└─ CellSettingsApplier

└─ Styles

└─ Range

└─ Worksheet

└─ Row

└─ Cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158



Gets or sets cell alignment options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IAlignmentOptions Alignment { get; set; }
```

```
Public Property Alignment As IAlignmentOptions
```

```
public:  
virtual property IAlignmentOptions^ Alignment {  
    IAlignmentOptions^ get () sealed;  
    void set (IAlignmentOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets border settings

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IBorderOptions Border { get; set; }
```

```
Public Property Border As IBorderOptions
```

```
public:  
virtual property IBorderOptions^ Border {  
    IBorderOptions^ get () sealed;  
    void set (IBorderOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Property

Namespaces ► [Yogesh.ExcelXml](#) ► [Styles](#) ► CustomFormatString

Gets or sets custom display format string

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string CustomFormatString { get; set; }
```

```
Public Property CustomFormatString As String
```

```
public:  
virtual property String^ CustomFormatString {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets the cell display format

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DisplayFormatType DisplayFormat { get; set; }
```

```
Public Property DisplayFormat As DisplayFormatType
```

```
public:  
virtual property DisplayFormatType DisplayFormat {  
    DisplayFormatType get () sealed;  
    void set (DisplayFormatType value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the font options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IFontOptions Font { get; set; }
```

```
Public Property Font As IFontOptions
```

```
public:  
virtual property IFontOptions^ Font {  
    IFontOptions^ get () sealed;  
    void set (IFontOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets interior options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IInteriorOptions Interior { get; set; }
```

```
Public Property Interior As IInteriorOptions
```

```
public:  
virtual property IInteriorOptions^ Interior {  
    IInteriorOptions^ get () sealed;  
    void set (IInteriorOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Returns the [XmlStyle](#) reference of the cell

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public XmlStyle Style { get; set; }
```

```
Public Property Style As XmlStyle
```

```
public:  
property XmlStyle^ Style {  
    XmlStyle^ get ();  
    void set (XmlStyle^ value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Enumeration

Namespaces ► [Yogesh.ExcelXml](#) ► VerticalAlignment

Cell's vertical alignment values

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum VerticalAlignment
```

```
Public Enumeration VerticalAlignment
```

```
public enum class VerticalAlignment
```

▣ Members

Member	Description
None	None
Top	Top aligned
Center	Centered
Bottom	Bottom aligned
Justify	Justified
Distributed	Distributed

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Worksheet class represents a single sheet in a workbook

[-] **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public class Worksheet : Styles, IEnumerable<Cell>, IEnumerable</pre>		

```
Public Class Worksheet _
    Inherits Styles _
    Implements IEnumerable(Of Cell), IEnumerable
```

```
public ref class Worksheet : public Styles,
    IEnumerable<Cell^>, IEnumerable
```

[-] **Members**

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	AddNamedRange(Range, String)	Add a named range to the book with limited scope with this sheet
	AddRow()	Adds a row at the end of the sheet
	Alignment	Gets or sets cell alignment options (Inherited from Styles .)

	Border	Gets or sets border settings (Inherited from Styles .)
	ColumnCount	Number of columns in this worksheet
	Columns(Int32)	Returns the column at a given position
	CustomFormatString	Gets or sets custom display format string (Inherited from Styles .)
	Delete()	Delete this sheet from the workbook
	DeleteColumn(Int32, Boolean)	Completely removes a column at a given index
	DeleteColumn(Int32)	Completely removes a column at a given index
	DeleteColumns(Int32, Int32, Boolean)	Completely removes a specified number of columns from a given index
	DeleteColumns(Int32, Int32)	Completely removes a specified number of columns from a given index
		

	DeleteRow(Int32)	Deletes a row
	DeleteRow(Row)	Deletes a row
	DeleteRow(Int32, Boolean)	Deletes a row
	DeleteRow(Row, Boolean)	Deletes a row
	DeleteRows(Int32, Int32)	Delete a specific number of rows starting from a row index
	DeleteRows(Row, Int32)	Delete a specific number of rows starting from a row instance
	DeleteRows(Int32, Int32, Boolean)	Delete a specific number of rows starting from a row index
	DeleteRows(Row, Int32, Boolean)	Delete a specific number of rows starting from a row instance
	DisplayFormat	Gets or sets the cell display format (Inherited from Styles .)
	Equals(Object)	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 .)
	Font	Gets or sets the font options (Inherited from Styles .)
	FreezeLeftColumns	Gets or sets left frozen column setting
	FreezeTopRows	Gets or sets top frozen row setting
	GetEnumerator()	Get a cell enumerator
	GetHashCode()	Serves as a hash function for a particular type. GetHashCode() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object .)
	GetType()	Gets the Type of the current instance. (Inherited from Object .)
	InsertColumnAfter(Int32)	Inserts a column after a given column index
	InsertColumnBefore(Int32)	Inserts a column before a given column index

	<code>InsertColumnsAfter(Int32, Int32)</code>	Inserts a specified number of columns after a given column index
	<code>InsertColumnsBefore(Int32, Int32)</code>	Inserts a specified number of columns before a given column index
	<code>InsertRowAfter(Int32)</code>	Inserts a row after another row
	<code>InsertRowAfter(Row)</code>	Inserts a row after another row
	<code>InsertRowBefore(Int32)</code>	Inserts a row before another row
	<code>InsertRowBefore(Row)</code>	Inserts a row before another row
	<code>InsertRowsAfter(Int32, Int32)</code>	Inserts a specific number of rows after a cell
	<code>InsertRowsAfter(Row, Int32)</code>	Inserts a specific number of rows after a cell
	<code>InsertRowsBefore(Int32, Int32)</code>	Inserts a specific number of rows before a row
		

	InsertRowsBefore(Row, Int32)	Inserts a specific number of rows before a row
	Interior	Gets or sets interior options (Inherited from Styles.)
	Item[Int32, Int32]	Returns the cell at a given position
	Item[Int32]	Returns the row at a given position
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object.)
	Name	Gets or sets the sheet name
	PrintOptions	Gets or sets various sheet printing options
	RowCount	Returns the number of rows present in the sheet
	Style	Returns the XmlStyle reference of the cell (Inherited from Styles.)
	TabColor	Gets or sets the tab color

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

▣ **Remarks**

Worksheet class represents a single sheet in a workbook.

You cannot directly declare a instance of a sheet from your code by using **new** keyword. The only way to access a sheet is to retrieve it from a workbook.

▣ **Inheritance Hierarchy**

Object

└─ [CellSettingsApplier](#)

└─ [Styles](#)

└─ [Worksheet](#)

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **AddNamedRange Method**
(range, name)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

AddNamedRange(Range, String)

Add a named range to the book with limited scope with this sheet

C#

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void AddNamedRange(  
    Range range,  
    string name  
)
```

```
Public Sub AddNamedRange ( _  
    range As Range, _  
    name As String _  
)
```

```
public:  
void AddNamedRange(  
    Range^ range,  
    String^ name  
)
```

▣ **Parameters**

range (Range)

Range to be named

name (String)

Name of the range

▣ **Remarks**

This property always adds sheet level named ranges. To add globally valid ranges, use [Name](#) property in [Range](#).

▣ **Remarks**

Range may not necessarily reside in this sheet

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Adds a row at the end of the sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Row AddRow()
```

```
Public Function AddRow As Row
```

```
public:  
Row^ AddRow()
```

▣ Return Value

The new row instance which is added

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Number of columns in this worksheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int ColumnCount { get; }
```

```
Public ReadOnly Property ColumnCount As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

C# ▼

Returns the column at a given position

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Column Columns(  
    int colIndex  
)
```

```
Public Function Columns ( _  
    colIndex As Integer _  
) As Column
```

```
public:  
Column^ Columns(  
    int colIndex  
)
```

▣ Parameters

colIndex (Int32)

Index of the [Column](#) starting from 0

▣ Return Value

[Column](#) reference to the requested column

Delete this sheet from the workbook

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Delete()
```

```
Public Sub Delete
```

```
public:  
void Delete()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

C# ▼

[-] **Members**

Icon	Member	Description
	DeleteColumn(Int32, Boolean)	Completely removes a column at a given index
	DeleteColumn(Int32)	Completely removes a column at a given index

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

(index)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

DeleteColumn(Int32)

C# ▼

Completely removes a column at a given index

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteColumn(  
    int index  
)
```

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

```
public:  
void DeleteColumn(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of column to delete columns from

▣ Remarks

The column is removed and columns to the right are cascaded leftward:

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DeleteColumn Method**
(index, cascade)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►
DeleteColumn(Int32, Boolean)

C# ▼

Completely removes a column at a given index

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteColumn(  
    int index,  
    bool cascade  
)
```

```
Public Sub DeleteColumn ( _  
    index As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteColumn(  
    int index,  
    bool cascade  
)
```

▣ Parameters

index (Int32)

Index of column to delete columns from

cascade (Boolean)

if true, the columns are removed and columns to the right are cascaded leftwards. if false, the columns are only emptied

C# ▼

▣ **Members**

Icon	Member	Description
	DeleteColumns(Int32, Int32, Boolean)	Completely removes a specified a number of columns from a given index
	DeleteColumns(Int32, Int32)	Completely removes a specified a number of columns from a given index

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **DeleteColumns Method**
(index, numberOfColumns)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►
DeleteColumns(Int32, Int32)

Completely removes a specified a number of columns from a given index

C#

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteColumns(  
    int index,  
    int numberOfColumns  
)
```

```
Public Sub DeleteColumns ( _  
    index As Integer, _  
    numberOfColumns As Integer _  
)
```

```
public:  
void DeleteColumns(  
    int index,  
    int numberOfColumns  
)
```

▣ Parameters

index (Int32)

Index of column to delete columns from

numberOfColumns (Int32)

Number of columns to delete

▣ Remarks

The columns are removed and columns to the right are cascaded leftwards

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DeleteColumns Method**
(index, numberOfColumns, cascade)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

DeleteColumns(Int32, Int32, Boolean)

Completely removes a specified a number of columns from a given index

C#

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteColumns(  
    int index,  
    int numberOfColumns,  
    bool cascade  
)
```

```
Public Sub DeleteColumns ( _  
    index As Integer, _  
    numberOfColumns As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteColumns(  
    int index,  
    int numberOfColumns,  
    bool cascade  
)
```

▣ Parameters

index (Int32)

Index of column to delete columns from

numberOfColumns (Int32)

Number of columns to delete

cascade (Boolean)

if true, the columns are removed and columns to the right are cascaded leftwards. if false, the columns are only emptied

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Members

Icon	Member	Description
	DeleteRow(Int32)	Deletes a row
	DeleteRow(Row)	Deletes a row
	DeleteRow(Int32, Boolean)	Deletes a row
	DeleteRow(Row, Boolean)	Deletes a row

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

C# ▼

Deletes a row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteRow(  
    int index  
)
```

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

```
public:  
void DeleteRow(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of row to delete

▣ Remarks

The row is removed and rows after the row specified are cascaded upwards.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **DeleteRow Method (index, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [DeleteRow\(Int32, Boolean\)](#)

Deletes a row

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void DeleteRow(  
    int index,  
    bool cascade  
)
```

```
Public Sub DeleteRow ( _  
    index As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteRow(  
    int index,  
    bool cascade  
)
```

▣ **Parameters**

index (Int32)

Index of row to delete

cascade (Boolean)

if true, the row is removed and rows after the row specified are cascaded upwards. if false, the rows are only emptied

C# ▼

Deletes a row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteRow(  
    Row row  
)
```

```
Public Sub DeleteRow ( _  
    row As Row _  
)
```

```
public:  
void DeleteRow(  
    Row^ row  
)
```

▣ Parameters

row (Row)

Instance of row to delete

▣ Remarks

The row is removed and rows after the row specified are cascaded upwards.

A Sandcastle Documented Class Library **DeleteRow Method (row, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► DeleteRow(Row, Boolean)

Deletes a row

C# ▼

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void DeleteRow(  
    Row row,  
    bool cascade  
)
```

```
Public Sub DeleteRow ( _  
    row As Row, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteRow(  
    Row^ row,  
    bool cascade  
)
```

▣ **Parameters**

row (Row)

Instance of row to delete

cascade (Boolean)

if true, the row is removed and rows after the row specified are cascaded upwards. if false, the rows are only emptied

C# ▼

[-] **Members**

Icon	Member	Description
	DeleteRows(Int32, Int32)	Delete a specific number of rows starting from a row index
	DeleteRows(Row, Int32)	Delete a specific number of rows starting from a row instance
	DeleteRows(Int32, Int32, Boolean)	Delete a specific number of rows starting from a row index
	DeleteRows(Row, Int32, Boolean)	Delete a specific number of rows starting from a row instance

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library **DeleteRows Method (index, numberOfRows)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [DeleteRows\(Int32, Int32\)](#)

Delete a specific number of rows starting from a row index

C#

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void DeleteRows(  
    int index,  
    int numberOfRows  
)
```

```
Public Sub DeleteRows ( _  
    index As Integer, _  
    numberOfRows As Integer _  
)
```

```
public:  
void DeleteRows(  
    int index,  
    int numberOfRows  
)
```

▣ **Parameters**

index (Int32)

Index of row from which the rows are deleted

numberOfRows (Int32)

Number of rows to delete

▣ **Remarks**

The rows are removed and rows after the row specified are cascaded upwards.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DeleteRows Method (index, numberOfRows, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [DeleteRows\(Int32, Int32, Boolean\)](#)

Delete a specific number of rows starting from a row index

C#

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteRows(  
    int index,  
    int numberOfRows,  
    bool cascade  
)
```

```
Public Sub DeleteRows ( _  
    index As Integer, _  
    numberOfRows As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteRows(  
    int index,  
    int numberOfRows,  
    bool cascade  
)
```

▣ Parameters

index (Int32)

Index of row from which the rows are deleted

numberOfRows (Int32)

Number of rows to delete

cascade (Boolean)

if true, the rows are removed and rows after the row specified are cascaded upwards. if false, the rows are only emptied

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DeleteRows Method (row, numberOfRows)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [DeleteRows\(Row, Int32\)](#)

C#

Delete a specific number of rows starting from a row instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteRows(  
    Row row,  
    int numberOfRows  
)
```

```
Public Sub DeleteRows ( _  
    row As Row, _  
    numberOfRows As Integer _  
)
```

```
public:  
void DeleteRows(  
    Row^ row,  
    int numberOfRows  
)
```

▣ Parameters

row (Row)

Instance of row from which the rows are deleted

numberOfRows (Int32)

Number of rows to delete

▣ Remarks

The rows are removed and rows after the row specified are cascaded upwards.

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library **DeleteRows Method (row, numberOfRows, cascade)**

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► DeleteRows(Row, Int32, Boolean)

C#

Delete a specific number of rows starting from a row instance

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void DeleteRows(  
    Row row,  
    int numberOfRows,  
    bool cascade  
)
```

```
Public Sub DeleteRows ( _  
    row As Row, _  
    numberOfRows As Integer, _  
    cascade As Boolean _  
)
```

```
public:  
void DeleteRows(  
    Row^ row,  
    int numberOfRows,  
    bool cascade  
)
```

▣ Parameters

row (**Row**)

Instance of row from which the rows are deleted

numberOfRows (**Int32**)

Number of rows to delete

cascade (**Boolean**)

if true, the rows are removed and rows after the row specified are cascaded upwards. if false, the rows are only emptied

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Property

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► FreezeLeftColumns

Gets or sets left freezed column setting

▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>public int FreezeLeftColumns { get; set; }</code>		

```
Public Property FreezeLeftColumns As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Gets or sets top frozen row setting

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property FreezeTopRows As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Get a cell enumerator

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerator<Cell> GetEnumerator()
```

```
Public Function GetEnumerator As IEnumerator(Of Cell
```

```
public:  
virtual IEnumerator<Cell^> GetEnumerator() sealed
```

▣ Return Value

returns IEnumerator<Cell>

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library

InsertColumnAfter Method (index)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [InsertColumnAfter\(Int32\)](#)

C# ▼

Inserts a column after a given column index

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertColumnAfter(  
    int index  
)
```

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

```
public:  
void InsertColumnAfter(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of column after which new column should be inserted

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertColumnBefore Method (index)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertColumnBefore(Int32)

C# ▼

Inserts a column before a given column index

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertColumnBefore(  
    int index  
)
```

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

```
public:  
void InsertColumnBefore(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of column before which new column should be inserted

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library

InsertColumnsAfter Method (index, numberOfColumns)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertColumnsAfter(Int32, Int32)

Inserts a specified number of columns after a given column index

C#

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertColumnsAfter(  
    int index,  
    int numberOfColumns  
)
```

```
Public Sub InsertColumnsAfter ( _  
    index As Integer, _  
    numberOfColumns As Integer _  
)
```

```
public:  
void InsertColumnsAfter(  
    int index,  
    int numberOfColumns  
)
```

Parameters

index (Int32)

Index of column after which columns should be inserted

numberOfColumns (Int32)

Number of columns to insert

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

A Sandcastle Documented Class Library

InsertColumnsBefore Method (index, numberOfColumns)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertColumnsBefore(Int32, Int32)

Inserts a specified number of columns before a given column index

C#

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertColumnsBefore(  
    int index,  
    int numberOfColumns  
)
```

```
Public Sub InsertColumnsBefore ( _  
    index As Integer, _  
    numberOfColumns As Integer _  
)
```

```
public:  
void InsertColumnsBefore(  
    int index,  
    int numberOfColumns  
)
```

Parameters

index (Int32)

Index of column before which columns should be inserted

numberOfColumns (Int32)

Number of columns to insert

C#

Members

Icon	Member	Description
	InsertRowAfter(Int32)	Inserts a row after another row
	InsertRowAfter(Row)	Inserts a row after another row

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertRowAfter Method

(index)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowAfter(Int32)

C#

Inserts a row after another row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Row InsertRowAfter(  
    int index  
)
```

```
Public Function InsertRowAfter ( _  
    index As Integer _  
) As Row
```

```
public:  
Row^ InsertRowAfter(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of row after which the new row is to be inserted

▣ Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Worksheet.InsertRowAfter(System.Int32)]

InsertRowAfter Method (row)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowAfter(Row)

C#

Inserts a row after another row

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Row InsertRowAfter(  
    Row row  
)
```

```
Public Function InsertRowAfter ( _  
    row As Row _  
) As Row
```

```
public:  
Row^ InsertRowAfter(  
    Row^ row  
)
```

Parameters

row (Row)

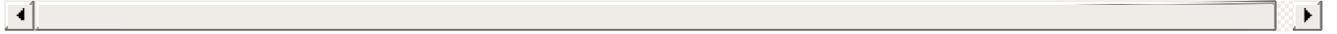
Instance of row after which the new row is to be inserted

Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Worksheet.InsertRowAfter(Yogesh.ExcelXml.F

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158





C# ▼

▣ **Members**

Icon	Member	Description
	InsertRowBefore(Int32)	Inserts a row before another row
	InsertRowBefore(Row)	Inserts a row before another row

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

A Sandcastle Documented Class Library

InsertRowBefore Method (index)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [InsertRowBefore\(Int32\)](#)

C# ▼

Inserts a row before another row

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Row InsertRowBefore(  
    int index  
)
```

```
Public Function InsertRowBefore ( _  
    index As Integer _  
) As Row
```

```
public:  
Row^ InsertRowBefore(  
    int index  
)
```

▣ Parameters

index (Int32)

Index of row before which the new row is to be inserted

▣ Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Worksheet.InsertRowBefore(System.Int32)]

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertRowBefore Method (row)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowBefore(Row)

C#

Inserts a row before another row

Declaration Syntax

C#

Visual Basic

Visual C++

```
public Row InsertRowBefore(  
    Row row  
)
```

```
Public Function InsertRowBefore ( _  
    row As Row _  
) As Row
```

```
public:  
Row^ InsertRowBefore(  
    Row^ row  
)
```

Parameters

row (Row)

Instance of row before which the new row is to be inserted

Return Value

[Missing <returns> documentation for M:Yogesh.ExcelXml.Worksheet.InsertRowBefore(Yogesh.ExcelXml

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158



C# ▼

[-] **Members**

Icon	Member	Description
	InsertRowsAfter(Int32, Int32)	Inserts a specific number of rows after a cell
	InsertRowsAfter(Row, Int32)	Inserts a specific number of rows after a cell

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertRowsAfter Method (index, rows)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowsAfter(Int32, Int32)

C# ▼

Inserts a specific number of rows after a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertRowsAfter(  
    int index,  
    int rows  
)
```

```
Public Sub InsertRowsAfter ( _  
    index As Integer, _  
    rows As Integer _  
)
```

```
public:  
void InsertRowsAfter(  
    int index,  
    int rows  
)
```

Parameters

index (Int32)

Index of row after which the new rows are inserted

rows (Int32)

Number of rows to insert

(row, rows)[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowsAfter(Row, Int32)

Inserts a specific number of rows after a cell

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertRowsAfter(  
    Row row,  
    int rows  
)
```

```
Public Sub InsertRowsAfter ( _  
    row As Row, _  
    rows As Integer _  
)
```

```
public:  
void InsertRowsAfter(  
    Row^ row,  
    int rows  
)
```

Parameters**row (Row)**

Instance of row after which the new rows are inserted

rows (Int32)

Number of rows to insert

C#

[-] **Members**

Icon	Member	Description
	InsertRowsBefore(Int32, Int32)	Inserts a specific number of rows before a row
	InsertRowsBefore(Row, Int32)	Inserts a specific number of rows before a row

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

InsertRowsBefore Method (index, rows)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowsBefore(Int32, Int32)

C#

Inserts a specific number of rows before a row

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertRowsBefore(  
    int index,  
    int rows  
)
```

```
Public Sub InsertRowsBefore ( _  
    index As Integer, _  
    rows As Integer _  
)
```

```
public:  
void InsertRowsBefore(  
    int index,  
    int rows  
)
```

Parameters

index (Int32)

Index of row before which the new rows are inserted

rows (Int32)

Number of rows to insert

InsertRowsBefore Method (row, rows)

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ►

InsertRowsBefore(Row, Int32)

Inserts a specific number of rows before a row

Declaration Syntax

C#

Visual Basic

Visual C++

```
public void InsertRowsBefore(  
    Row row,  
    int rows  
)
```

```
Public Sub InsertRowsBefore ( _  
    row As Row, _  
    rows As Integer _  
)
```

```
public:  
void InsertRowsBefore(  
    Row^ row,  
    int rows  
)
```

Parameters

row (Row)

Instance of row before which the new rows are inserted

rows (Int32)

Number of rows to insert

Members

Icon	Member	Description
	Item[Int32, Int32]	Returns the cell at a given position
	Item[Int32]	Returns the row at a given position

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Returns the row at a given position

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Row this[
    int rowIndex
] { get; }
```

```
Public ReadOnly Default Property Item ( _
    rowIndex As Integer _
) As Row
```

```
public:
property Row^ default[int rowIndex] {
    Row^ get (int rowIndex);
}
```

▣ Parameters

rowIndex (Int32)

Index of the [Row](#) starting from 0

▣ Return Value

[Row](#) reference to the requested row

Item Property (colIndex, rowIndex)

Namespaces ► [Yogesh.ExcelXml](#) ► [Worksheet](#) ► [Item\[Int32, Int32\]](#)

C#

Returns the cell at a given position

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Cell this[
    int colIndex,
    int rowIndex
] { get; }
```

```
Public ReadOnly Default Property Item ( _
    colIndex As Integer, _
    rowIndex As Integer _
) As Cell
```

```
public:
property Cell^ default[int colIndex, int rowIndex] {
    Cell^ get (int colIndex, int rowIndex);
}
```

▣ Parameters

colIndex (Int32)

Index of the [Cell](#) starting from 0

rowIndex (Int32)

Index of the [Row](#) starting from 0

▣ Return Value

[Cell](#) reference to the requested cell

Gets or sets the sheet name

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property Name As String
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets various sheet printing options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public PrintOptions PrintOptions { get; set; }
```

```
Public Property PrintOptions As PrintOptions
```

```
public:  
property PrintOptions^ PrintOptions {  
    PrintOptions^ get ();  
    void set (PrintOptions^ value);  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Returns the number of rows present in the sheet

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int RowCount { get; }
```

```
Public ReadOnly Property RowCount As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the tab color

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Property TabColor As Integer
```

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

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Style class for cells, rows and worksheets

[-] **Declaration Syntax**

C# Visual Basic Visual C++

```
public class XmlStyle
```

```
Public Class XmlStyle
```

```
public ref class XmlStyle
```

[-] **Members**

All Members Constructors Methods Properties

<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	XmlStyle()	Creates a new instance
	XmlStyle(XmlStyle)	Creates a new instance from another instance of XmlStyle
	Alignment	Gets or sets cell alignment options
	Border	Gets or sets border settings

	CustomFormatString	Gets or sets a custom display string
	DisplayFormat	Gets or sets the cell display format
	Equality(XmlStyle, XmlStyle)	Equality operator
	Equals(Object)	Equality operator (Overrides Object.Equals(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.)
	Font	Gets or sets the font options
	GetHashCode()	Returns the hash code of the class (Overrides Object.GetHashCode().)
	GetType()	Gets the Type of the current instance (Inherited from Object.)
	Inequality(XmlStyle, XmlStyle)	Inequality operator
	Interior	Gets or sets interior options
	MemberwiseClone()	Creates a shallow copy of the current Object.

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

▣ Inheritance Hierarchy

Object

└─ XmlStyle

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Members

Icon	Member	Description
	XmlStyle()	Creates a new instance
	XmlStyle(XmlStyle)	Creates a new instance from another instance of XmlStyle

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Creates a new instance

[-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public XmlStyle()
```

```
Public Sub New
```

```
public:  
XmlStyle()
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Creates a new instance from another instance of XmlStyle

[-] **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public XmlStyle(  
    XmlStyle style  
)
```

```
Public Sub New ( _  
    style As XmlStyle _  
)
```

```
public:  
XmlStyle(  
    XmlStyle^ style  
)
```

[-] **Parameters**

style (XmlStyle)
Instance to copy

Gets or sets cell alignment options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IAlignmentOptions Alignment { get; set; }
```

```
Public Property Alignment As IAlignmentOptions
```

```
public:  
virtual property IAlignmentOptions^ Alignment {  
    IAlignmentOptions^ get () sealed;  
    void set (IAlignmentOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets border settings

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IBorderOptions Border { get; set; }
```

```
Public Property Border As IBorderOptions
```

```
public:  
virtual property IBorderOptions^ Border {  
    IBorderOptions^ get () sealed;  
    void set (IBorderOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

Property

Namespaces ► [Yogesh.ExcelXml](#) ► [XmlStyle](#) ► CustomFormatString

C# ▼

Gets or sets a custom display string

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string CustomFormatString { get; set; }
```

```
Public Property CustomFormatString As String
```

```
public:  
virtual property String^ CustomFormatString {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Gets or sets the cell display format

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DisplayFormatType DisplayFormat { get; set; }
```

```
Public Property DisplayFormat As DisplayFormatType
```

```
public:  
virtual property DisplayFormatType DisplayFormat {  
    DisplayFormatType get () sealed;  
    void set (DisplayFormatType value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [XmlStyle](#) ► Equality(XmlStyle, XmlStyle)

Equality operator

Declaration Syntax

C#

Visual Basic

Visual C++

```
public static bool operator ==(
    XmlStyle cellOne,
    XmlStyle cellTwo
)
```

```
Public Shared Operator = ( _
    cellOne As XmlStyle, _
    cellTwo As XmlStyle _
) As Boolean
```

```
public:
static bool operator ==(
    XmlStyle^ cellOne,
    XmlStyle^ cellTwo
)
```

Parameters

cellOne (XmlStyle)

Instance one to compare

cellTwo (XmlStyle)

Instance two to compare

Return Value

true if all the values of the styles match, false otherwise

2.89.501.2158

Equality operator

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

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

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

▣ Parameters

obj (Object)

Instance to compare

▣ Return Value

true if all the values of the styles match, false otherwise

Gets or sets the font options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IFontOptions Font { get; set; }
```

```
Public Property Font As IFontOptions
```

```
public:  
virtual property IFontOptions^ Font {  
    IFontOptions^ get () sealed;  
    void set (IFontOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

Returns the hash code of the class

▣ Declaration Syntax

C#

Visual Basic

Visual C++

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

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

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

▣ Return Value

Hash code of the class

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version: 2.89.501.2158

[Namespaces](#) ► [Yogesh.ExcelXml](#) ► [XmlStyle](#) ► [Inequality\(XmlStyle, XmlStyle\)](#)

Inequality operator

Declaration Syntax

C#

Visual Basic

Visual C++

```
public static bool operator !=(  
    XmlStyle cellOne,  
    XmlStyle cellTwo  
)
```

```
Public Shared Operator <> ( _  
    cellOne As XmlStyle, _  
    cellTwo As XmlStyle _  
) As Boolean
```

```
public:  
static bool operator !=(  
    XmlStyle^ cellOne,  
    XmlStyle^ cellTwo  
)
```

Parameters

cellOne (XmlStyle)

Instance one to compare

cellTwo (XmlStyle)

Instance two to compare

Return Value

true if the values of the styles dont match, false otherwise

2.89.501.2158

Gets or sets interior options

▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IInteriorOptions Interior { get; set; }
```

```
Public Property Interior As IInteriorOptions
```

```
public:  
virtual property IInteriorOptions^ Interior {  
    IInteriorOptions^ get () sealed;  
    void set (IInteriorOptions^ value) sealed;  
}
```

Assembly: Yogesh.ExcelXml (Module: Yogesh.ExcelXml) Version:
2.89.501.2158

