

Namespace CloudyWing.SpreadsheetExporter

Classes

[Cell](#)

The spreadsheet cell.

[Constants](#)

The constants.

[DataValidation](#)

Represents data validation rules that can be applied to spreadsheet cells. Data validation restricts the type of data or values that users can enter into a cell.

[ExporterBase](#)

The exporter base.

[PageSettings](#)

The page settings.

[PaperSize](#)

The paper size.

[SheetCreatedEventArgs](#)

Event arguments after sheet create.

[Sheeter](#)

The sheeter.

[SheeterContext](#)

The sheeter context.

[SpreadsheetExportedEventArgs](#)

Event arguments after spreadsheet export.

[SpreadsheetExportingEventArgs](#)

Event arguments before spreadsheet export.

[SpreadsheetManager](#)

The spreadsheet manager.

Structs

[CellFont](#)

The cell font.

[CellStyle](#)

The cell style.

Interfaces

[ISpreadsheetExporter](#)

Interface for spreadsheet exporters.

Enums

[DataValidationOperator](#)

Defines the comparison operators used in data validation.

[DataValidationType](#)

Defines the types of data validation that can be applied to cells.

[FontStyles](#)

The font styles.

[HorizontalAlignment](#)

The horizontal text alignment.

[PageOrientation](#)

The page orientation.

[Spreadsheet FileMode](#)

The file mode of spreadsheet

[VerticalAlignment](#)

The vertical text alignment.

Class Cell

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The spreadsheet cell.

```
public class Cell
```

Inheritance

[object](#) ← Cell

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

CellStyleGenerator

Gets or sets the cell style generator. Pass the cell index and row index to the generator. The index start at 0.

```
public Func<int, int, CellStyle> CellStyleGenerator { get; set; }
```

Property Value

[Func](#) <[int](#), [int](#), [CellStyle](#)>

The cell style.

DataValidationGenerator

Gets or sets the data validation generator. Pass the cell index and row index to the generator. The index start at 0.

```
public Func<int, int, DataValidation> DataValidationGenerator { get; set; }
```

Property Value

[Func<int, int, DataValidation>](#)

The data validation generator.

FormulaGenerator

Gets or sets the formula generator. Pass the cell index and row index to the generator. The index start at 0.

```
public Func<int, int, string> FormulaGenerator { get; set; }
```

Property Value

[Func<int, int, string>](#)

The formula generator.

Point

Gets or sets the point.

```
public Point Point { get; set; }
```

Property Value

[Point](#)

The point.

Size

Gets or sets the cell size.

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

Property Value

[Size](#)

The cell size.

ValueGenerator

Gets or sets the cell content value generator. Pass the cell index and row index to the generator. The index start at 0.

```
public Func<int, int, object> ValueGenerator { get; set; }
```

Property Value

[Func](#) <[int](#), [int](#), [object](#)>

The cell content value.

Methods

GetCellStyle()

Gets the cell style.

```
public CellStyle GetCellStyle()
```

Returns

[CellStyle](#)

The cell style.

GetDataValidation()

Gets the data validation.

```
public DataValidation GetDataValidation()
```

Returns

[DataValidation](#)

The data validation, or `null` if no validation is specified.

GetFormula()

Gets the formula. If formula starts with `=`, automatically removed.

```
public string GetFormula()
```

Returns

[string](#)

The formula.

GetValue()

Gets the content value.

```
public object GetValue()
```

Returns

[object](#)

The content value.

ShallowCopy()

Shallows the copy.

```
public Cell ShallowCopy()
```

Returns

[Cell](#)

The spreadsheet cell.

Struct CellFont

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The cell font.

```
public record struct CellFont : IEquatable<CellFont>
```

Implements

[IEquatable](#) <[CellFont](#)>

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

CellFont(string, short, Color, FontStyles)

The cell font.

```
public CellFont(string Name = null, short Size = 0, Color Color = default, FontStyles Style = FontStyles.None)
```

Parameters

Name [string](#)

The font name.

Size [short](#)

The size.

Color [Color](#)

The color.

Style [FontStyles](#)

The style.

Properties

Color

The color.

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

Property Value

[Color](#)

Empty

The cell font equals to `new CellFont()`.

```
public static CellFont Empty { get; }
```

Property Value

[CellFont](#)

Name

The font name.

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

Property Value

[string](#)

Size

The size.

```
public short Size { readonly get; set; }
```

Property Value

[short ↗](#)

Style

The style.

```
public FontStyles Style { readonly get; set; }
```

Property Value

[FontStyles](#)

Struct CellStyle

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The cell style.

```
public record struct CellStyle : IEquatable<CellStyle>
```

Implements

[IEquatable](#) <[CellStyle](#)>

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

CellStyle(HorizontalAlignment, VerticalAlignment, bool, bool, Color, CellFont, string, bool)

The cell style.

```
public CellStyle(HorizontalAlignment HorizontalAlignment = HorizontalAlignment.General,  
VerticalAlignment VerticalAlignment = VerticalAlignment.Top, bool HasBorder = false, bool  
WrapText = false, Color BackgroundColor = default, CellFont Font = default, string  
DataFormat = null, bool IsLocked = false)
```

Parameters

HorizontalAlignment [HorizontalAlignment](#)

Gets the horizontal alignment.

VerticalAlignment [VerticalAlignment](#)

Gets the vertical alignment.

HasBorder [bool](#)

Gets a value indicating whether this instance has border.

WrapText [bool](#)

Gets a value indicating whether [wrap text].

BackgroundColor [Color](#)

Gets the color of the background.

Font [CellFont](#)

Gets the font.

DataFormat [string](#)

Gets the data format.

IsLocked [bool](#)

Gets a value indicating whether this instance is locked.

Properties

BackgroundColor

Gets the color of the background.

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

Property Value

[Color](#)

DataFormat

Gets the data format.

```
public string DataFormat { readonly get; set; }
```

Property Value

[string](#) ↗

Empty

The cell style equals to `new CellStyle()`.

```
public static CellStyle Empty { get; }
```

Property Value

[CellStyle](#)

Font

Gets the font.

```
public CellFont Font { readonly get; set; }
```

Property Value

[CellFont](#)

HasBorder

Gets a value indicating whether this instance has border.

```
public bool HasBorder { readonly get; set; }
```

Property Value

[bool](#) ↗

HorizontalAlignment

Gets the horizontal alignment.

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

Property Value

[HorizontalAlignment](#)

IsLocked

Gets a value indicating whether this instance is locked.

```
public bool IsLocked { readonly get; set; }
```

Property Value

[bool](#)

VerticalAlignment

Gets the vertical alignment.

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

Property Value

[VerticalAlignment](#)

WrapText

Gets a value indicating whether [wrap text].

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

Property Value

bool ↗

Class Constants

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The constants.

```
public static class Constants
```

Inheritance

[object](#) ← Constants

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

AutoFitColumnWidth

The automatic fit column width.

```
public const int AutoFitColumnWidth = -1
```

Field Value

[int](#)

AutoFitRowHeight

The automatic fit row height.

```
public const int AutoFitRowHeight = -1
```

Field Value

[int↗](#)

AutoFitRowHeight

The automatic fit row height.

```
[Obsolete("Use AutoFitRowHeight instead. This property will be removed in a  
future version.")]  
public const int AutoFitRowHeight = -1
```

Field Value

[int↗](#)

HiddenColumn

The hidden column.

```
public const int HiddenColumn = 0
```

Field Value

[int↗](#)

HiddenRow

The hidden row.

```
public const int HiddenRow = 0
```

Field Value

[int↗](#)

Class DataValidation

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Represents data validation rules that can be applied to spreadsheet cells. Data validation restricts the type of data or values that users can enter into a cell.

```
public class DataValidation
```

Inheritance

[object](#) ← DataValidation

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ErrorMessage

Gets or sets the error message shown when invalid data is entered.

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

Property Value

[string](#)

The error message.

ErrorTitle

Gets or sets the title of the error message shown when invalid data is entered.

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

Property Value

[string](#) ↗

The error title.

Formula

Gets or sets the formula for validation. Used with Custom validation type (required), and can also be used with Date, Time, Integer, Decimal, and TextLength validation types as an alternative to Value1/Value2. When using formulas with Date/Time/Numeric validations, the formula will be automatically prefixed with '=' if not already present.

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

Property Value

[string](#) ↗

The validation formula (e.g., "TODAY()", "=A1+10").

IsBlankAllowed

Gets or sets a value indicating whether blank/empty values are allowed.

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

Property Value

[bool](#) ↗

true if blank values are allowed; otherwise, **false**. Default is **true**.

IsDropdownShown

Gets or sets a value indicating whether the dropdown button is shown for list validation. Only applicable when ValidationType is List.

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

Property Value

[bool](#) ↗

`true` if dropdown is shown; otherwise, `false`. Default is `true`.

IsErrorAlertShown

Gets or sets a value indicating whether the error alert is shown when invalid data is entered.

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

Property Value

[bool](#) ↗

`true` if error alert is shown; otherwise, `false`. Default is `true`.

IsInputPromptShown

Gets or sets a value indicating whether the input prompt is shown when the cell is selected.

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

Property Value

[bool](#) ↗

`true` if input prompt is shown; otherwise, `false`. Default is `false`.

ListItems

Gets or sets the list of valid items for List validation type. Only used when ValidationType is List.

```
public IEnumerable<string> ListItems { get; set; }
```

Property Value

[IEnumerable](#)<[string](#)>

The list items.

Operator

Gets or sets the comparison operator for the validation. Not required for List or Custom validation types.

```
public DataValidationOperator? Operator { get; set; }
```

Property Value

[DataValidationOperator?](#)

The operator, or [null](#) if not applicable.

PromptMessage

Gets or sets the input prompt message shown when the cell is selected.

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

Property Value

[string](#)

The prompt message.

PromptTitle

Gets or sets the title of the input prompt message.

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

Property Value

[string](#)

The prompt title.

ValidationType

Gets or sets the type of validation to apply.

```
public DataValidationType ValidationType { get; set; }
```

Property Value

[DataValidationType](#)

The validation type.

Value1

Gets or sets the first value for comparison. For Between/NotBetween operators, this is the minimum value. For other operators, this is the value to compare against.

```
public object Value1 { get; set; }
```

Property Value

[object](#)

The first comparison value.

Value2

Gets or sets the second value for comparison. Only used with Between and NotBetween operators (represents the maximum value).

```
public object Value2 { get; set; }
```

Property Value

object ↗

The second comparison value, or `null` if not applicable.

Enum DataValidationOperator

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Defines the comparison operators used in data validation.

```
public enum DataValidationOperator
```

Fields

Between = 0

Value must be between two values (inclusive).

Equal = 2

Value must be equal to a specified value.

GreaterThan = 4

Value must be greater than a specified value.

GreaterThanOrEqual = 6

Value must be greater than or equal to a specified value.

LessThan = 5

Value must be less than a specified value.

LessThanOrEqual = 7

Value must be less than or equal to a specified value.

NotBetween = 1

Value must not be between two values.

NotEqual = 3

Value must not be equal to a specified value.

Enum DataValidationType

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Defines the types of data validation that can be applied to cells.

```
public enum DataValidationType
```

Fields

Custom = 6

Custom validation - allows custom validation formulas.

Date = 3

Date validation - restricts input to dates.

Decimal = 2

Decimal validation - restricts input to decimal numbers.

Integer = 1

Integer validation - restricts input to whole numbers.

List = 0

List validation - restricts input to values from a predefined list.

TextLength = 5

Text length validation - restricts the length of text input.

Time = 4

Time validation - restricts input to time values.

Class ExporterBase

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The exporter base.

```
public abstract class ExporterBase : ISpreadsheetExporter
```

Inheritance

[object](#) ← ExporterBase

Implements

[ISpreadsheetExporter](#)

Derived

[ExcelExporter](#), [ExcelExporter](#)

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Properties

ContentType

Gets the content-type.

```
public abstract string ContentType { get; }
```

Property Value

[string](#)

The content-type.

DefaultBasicSheetName

Gets or sets the default basic name of the sheet.

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

Property Value

[string](#)

The default basic name of the sheet.

DefaultFont

Gets or sets the default font.

```
public CellFont? DefaultFont { get; set; }
```

Property Value

[CellFont?](#)

The default font.

FileNameExtension

Gets the file name extension.

```
public abstract string FileNameExtension { get; }
```

Property Value

[string](#)

The file name extension.

HasPassword

Gets a value indicating whether this instance has password.

```
public bool HasPassword { get; }
```

Property Value

[bool](#)

`true` if this instance has password; otherwise, `false`.

LastSheeter

Gets the last sheeter. When there is no sheeter, create a sheeter.

```
public Sheeter LastSheeter { get; }
```

Property Value

[Sheeter](#)

The last sheeter.

Password

Gets or sets the workbook protection password.

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

Property Value

[string](#)

The password.

Methods

CreateSheeter(string, double?)

Creates the sheeter.

```
public Sheeter CreateSheeter(string sheetName = null, double? defaultRowHeight = null)
```

Parameters

`sheetName` [string](#)?

Name of the sheet.

`defaultRowHeight` [double](#)?

Default height of the row.

Returns

[Sheeter](#)

The sheeter.

ExecuteExport(IEnumerable<SheeterContext>)

Executes the export.

```
protected abstract byte[] ExecuteExport(IEnumerable<SheeterContext> contexts)
```

Parameters

`contexts` [IEnumerable](#)<[SheeterContext](#)>

The contexts.

Returns

[byte](#)[]

The exported bytes of the file.

Export()

Exports bytes of spreadsheet.

```
public byte[] Export()
```

Returns

[byte](#) []

The bytes of spreadsheet.

Exceptions

[SheeterNotFoundException](#)

Sheeter[0] is null.

ExportFile(string, Spreadsheet FileMode)

Exports the spreadsheet file.

```
public void ExportFile(string path, Spreadsheet FileMode fileMode  
= Spreadsheet FileMode.Create)
```

Parameters

[path](#) [string](#)

The path.

[fileMode](#) [Spreadsheet FileMode](#)

The file mode.

Exceptions

[SheeterNotFoundException](#)

No sheeters have been created.

[IOException](#)

An I/O error occurred while creating the file.

[UnauthorizedAccessException](#)

The caller does not have the required permission.

GetSheeter(int)

Gets the sheeter.

```
public Sheeter GetSheeter(int index)
```

Parameters

index [int](#)

The index.

Returns

[Sheeter](#)

The sheeter.

Exceptions

[ArgumentOutOfRangeException](#)

index is less than 0 or greater than or equal to the number of sheeters.

OnSheetCreated(SheetCreatedEventArgs)

Raises the SheetCreated event.

```
protected virtual void OnSheetCreated(SheetCreatedEventArgs args)
```

Parameters

args [SheetCreatedEventArgs](#)

The [SheetCreatedEventArgs](#) instance containing the event data.

OnSpreadsheetExported(SpreadsheetEventArgs)

Raises the SpreadsheetExported event.

```
protected virtual void OnSpreadsheetExported(SpreadsheetEventArgs args)
```

Parameters

args [SpreadsheetEventArgs](#)

The [SpreadsheetEventArgs](#) instance containing the event data.

OnSpreadsheetExporting(SpreadsheetEventArgs)

Raises the SpreadsheetExporting event.

```
protected virtual void OnSpreadsheetExporting(SpreadsheetEventArgs args)
```

Parameters

args [SpreadsheetEventArgs](#)

The [SpreadsheetEventArgs](#) instance containing the event data.

TryGetSheeter(int, out Sheeter)

Tries to get the sheeter at the specified index.

```
public bool TryGetSheeter(int index, out Sheeter sheeter)
```

Parameters

index [int](#)

The zero-based index of the sheeter to get.

sheeter [Sheeter](#)

When this method returns, contains the sheeter at the specified index if the index is valid; otherwise, `null`. This parameter is passed uninitialized.

Returns

[bool](#)

`true` if the index is valid and the sheeter was retrieved; otherwise, `false`.

Events

SheetCreatedEvent

Occurs when [sheet created event].

```
public event EventHandler<SheetCreatedEventArgs> SheetCreatedEvent
```

Event Type

[EventHandler](#) <[SheetCreatedEventArgs](#)>

SpreadsheetExportedEvent

Occurs when [spreadsheet exported event].

```
public event EventHandler<SpreadsheetExportedEventArgs> SpreadsheetExportedEvent
```

Event Type

[EventHandler](#) <[SpreadsheetExportedEventArgs](#)>

SpreadsheetExportingEvent

Occurs when [spreadsheet exporting event].

```
public event EventHandler<SpreadsheetExportingEventArgs> SpreadsheetExportingEvent
```

Event Type

[EventHandler](#) <[SpreadsheetExportingEventArgs](#)>

Enum FontStyles

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The font styles.

```
[Flags]
public enum FontStyles
```

Fields

HasUnderline = 4

The has underline.

IsBold = 1

The is bold.

IsItalic = 2

The is italic.

IsStrikeout = 8

The is strikeout.

None = 0

The none.

Enum HorizontalAlignment

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The horizontal text alignment.

```
public enum HorizontalAlignment
```

Fields

Center = 2

The center.

General = 0

The general.

Justify = 4

The justify.

Left = 1

The left.

Right = 3

The right.

Interface ISpreadsheetExporter

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Interface for spreadsheet exporters.

```
public interface ISpreadsheetExporter
```

Properties

ContentType

Gets the content-type.

```
string ContentType { get; }
```

Property Value

[string](#) ↗

The content-type.

DefaultBasicSheetName

Gets or sets the default basic name of the sheet.

```
string DefaultBasicSheetName { get; set; }
```

Property Value

[string](#) ↗

The default basic name of the sheet.

DefaultFont

Gets or sets the default font.

```
CellFont? DefaultFont { get; set; }
```

Property Value

[CellFont?](#)

The default font.

FileNameExtension

Gets the file name extension.

```
string FileNameExtension { get; }
```

Property Value

[string](#)

The file name extension.

HasPassword

Gets a value indicating whether this instance has password.

```
bool HasPassword { get; }
```

Property Value

[bool](#)

true if this instance has password; otherwise, **false**.

LastSheeter

Gets the last sheeter. When there is no sheeter, create a sheeter.

```
Sheeter LastSheeter { get; }
```

Property Value

[Sheeter](#)

The last sheeter.

Password

Gets or sets the workbook protection password.

```
string Password { get; set; }
```

Property Value

[string](#)

The password.

Methods

CreateSheeter(string, double?)

Creates the sheeter.

```
Sheeter CreateSheeter(string sheetName = null, double? defaultRowHeight = null)
```

Parameters

[sheetName](#) [string](#)

Name of the sheet.

[defaultRowHeight](#) [double](#)?

Default height of the row.

Returns

[Sheeter](#)

The sheeter.

Export()

Exports bytes of spreadsheet.

```
byte[] Export()
```

Returns

[byte](#)[]

The bytes of spreadsheet.

Exceptions

[SheeterNotFoundException](#)

No sheeters have been created.

ExportFile(string, Spreadsheet FileMode)

Exports the spreadsheet file.

```
void ExportFile(string path, Spreadsheet FileMode fileMode = Spreadsheet FileMode.Create)
```

Parameters

[path](#) [string](#)[]

The path.

[fileMode](#) [Spreadsheet FileMode](#)

The file mode.

GetSheeter(int)

Gets the sheeter.

```
Sheeter GetSheeter(int index)
```

Parameters

`index` [int](#)

The index.

Returns

[Sheeter](#)

The sheeter.

Exceptions

[ArgumentOutOfRangeException](#)

`index` is less than 0 or greater than or equal to the number of sheeters.

Events

SheetCreatedEvent

Occurs when [sheet created event].

```
event EventHandler<SheetCreatedEventArgs> SheetCreatedEvent
```

Event Type

[EventHandler](#) <[SheetCreatedEventArgs](#)>

SpreadsheetExportedEvent

Occurs when [spreadsheet exported event].

event EventHandler<SpreadsheetExportedEventArgs> SpreadsheetExportedEvent

Event Type

[EventHandler](#) <SpreadsheetExportedEventArgs>

SpreadsheetExportingEvent

Occurs when [spreadsheet exporting event].

event EventHandler<SpreadsheetExportingEventArgs> SpreadsheetExportingEvent

Event Type

[EventHandler](#) <SpreadsheetExportingEventArgs>

Enum PageOrientation

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The page orientation.

```
public enum PageOrientation
```

Fields

Landscape = 1

The landscape orientation.

Portrait = 0

The portrait orientation.

Class PageSettings

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The page settings.

```
public class PageSettings
```

Inheritance

[object](#) ← PageSettings

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

PageOrientation

Gets or sets the page orientation. The defaults is [Portrait](#).

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

Property Value

[PageOrientation](#)

The page orientation.

PaperSize

Gets or sets the size of the paper.

```
public PaperSize PaperSize { get; set; }
```

Property Value

PaperSize

The size of the pager. The defaults is **A4**.

Class PaperSize

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The paper size.

```
public class PaperSize : IEquatable<PaperSize>, IEquatable<int>, IComparable<PaperSize>,
IComparable<int>, IComparable
```

Inheritance

[object](#) ← PaperSize

Implements

[IEquatable](#)<[PaperSize](#)>, [IEquatable](#)<[int](#)>, [IComparable](#)<[PaperSize](#)>, [IComparable](#)<[int](#)>,
[IComparable](#)

Inherited Members

[object.Equals\(object, object\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),
[object.ReferenceEquals\(object, object\)](#)

Properties

A2

Gets the A2 paper (420 mm by 594 mm).

```
public static PaperSize A2 { get; }
```

Property Value

[PaperSize](#)

66

A3

Gets the A3 paper (297 mm by 420 mm).

```
public static PaperSize A3 { get; }
```

Property Value

[PaperSize](#)

8

A3Extra

Gets the A3 extra paper (322 mm by 445 mm).

```
public static PaperSize A3Extra { get; }
```

Property Value

[PaperSize](#)

63

A3ExtraTransverse

Gets the A3 extra transverse paper (322 mm by 445 mm).

```
public static PaperSize A3ExtraTransverse { get; }
```

Property Value

[PaperSize](#)

68

A3Transverse

Gets the A3 transverse paper (297 mm by 420 mm).

```
public static PaperSize A3Transverse { get; }
```

Property Value

[PaperSize](#)

67

A4

Gets the A4 paper (210 mm by 297 mm).

```
public static PaperSize A4 { get; }
```

Property Value

[PaperSize](#)

9

A4Extra

Gets the a4 extra paper (236 mm by 322 mm).

```
public static PaperSize A4Extra { get; }
```

Property Value

[PaperSize](#)

53

A4Plus

Gets the A4 plus paper (210 mm by 330 mm).

```
public static PaperSize A4Plus { get; }
```

Property Value

[PaperSize](#)

60

A4Small

Gets the A4 small paper (210 mm by 297 mm).

```
public static PaperSize A4Small { get; }
```

Property Value

[PaperSize](#)

10

A4Transverse

Gets the a4 transverse (210 mm by 297 mm).

```
public static PaperSize A4Transverse { get; }
```

Property Value

[PaperSize](#)

55

A5

Gets the A5 paper (148 mm by 210 mm).

```
public static PaperSize A5 { get; }
```

Property Value

[PaperSize](#)

11

A5Extra

Gets the A5 extra paper (174 mm by 235 mm).

```
public static PaperSize A5Extra { get; }
```

Property Value

[PaperSize](#)

64

A5Transverse

Gets the A5 transverse paper (148 mm by 210 mm).

```
public static PaperSize A5Transverse { get; }
```

Property Value

[PaperSize](#)

61

APlus

Gets a plus paper (227 mm by 356 mm).

```
public static PaperSize APlus { get; }
```

Property Value

[PaperSize](#)

57

B4

Gets the B4 paper (250 mm by 353 mm).

```
public static PaperSize B4 { get; }
```

Property Value

[PaperSize](#)

12

B4Envelope

Gets the B4 envelope (250 mm by 353 mm).

```
public static PaperSize B4Envelope { get; }
```

Property Value

[PaperSize](#)

33.

B5

Gets the B5 paper (176 mm by 250 mm).

```
public static PaperSize B5 { get; }
```

Property Value

[PaperSize](#)

13

B5Envelope

Gets the B5 envelope (176 mm by 250 mm).

```
public static PaperSize B5Envelope { get; }
```

Property Value

[PaperSize](#)

34

B5Extra

Gets the ISO B5 extra paper (201 mm by 276 mm).

```
public static PaperSize B5Extra { get; }
```

Property Value

[PaperSize](#)

65

B5Transverse

Gets the JIS B5 transverse paper (182 mm by 257 mm).

```
public static PaperSize B5Transverse { get; }
```

Property Value

[PaperSize](#)

62

B6Envelope

Gets the B6 envelope (176 mm by 125 mm).

```
public static PaperSize B6Envelope { get; }
```

Property Value

[PaperSize](#)

35

BPlus

Gets the b plus paper (305 mm by 487 mm).

```
public static PaperSize BPlus { get; }
```

Property Value

[PaperSize](#)

58

C3Envelope

Gets the C3 envelope (324 mm by 458 mm).

```
public static PaperSize C3Envelope { get; }
```

Property Value

[PaperSize](#)

29

C4Envelope

Gets the C4 envelope (229 mm by 324 mm).

```
public static PaperSize C4Envelope { get; }
```

Property Value

[PaperSize](#)

30

C5Envelope

Gets the C5 envelope (162 mm by 229 mm).

```
public static PaperSize C5Envelope { get; }
```

Property Value

[PaperSize](#)

28

C65Envelope

Gets the C65 envelope (114 mm by 229 mm).

```
public static PaperSize C6Envelope { get; }
```

Property Value

[PaperSize](#)

32

C6Envelope

Gets the C6 envelope (114 mm by 162 mm).

```
public static PaperSize C6Envelope { get; }
```

Property Value

[PaperSize](#)

31

CSheet

Gets the C sheet paper (17 in. by 22 in.).

```
public static PaperSize CSheet { get; }
```

Property Value

[PaperSize](#)

24

DLEnvelope

Gets the DL envelope (110 mm by 220 mm).

```
public static PaperSize DLEnvelope { get; }
```

Property Value

[PaperSize](#)

27

DSheet

Gets the D sheet paper (22 in. by 34 in.).

```
public static PaperSize DSheet { get; }
```

Property Value

[PaperSize](#)

25

ESheet

Gets the E sheet paper (34 in. by 44 in.).

```
public static PaperSize ESheet { get; }
```

Property Value

[PaperSize](#)

26

Executive

Gets the executive paper (7.25 in. by 10.5 in.).

```
public static PaperSize Executive { get; }
```

Property Value

[PaperSize](#)

7

Folio

Gets the folio paper (8.5 in. by 13 in.).

```
public static PaperSize Folio { get; }
```

Property Value

[PaperSize](#)

14

GermanLegalFanfold

Gets the german legal fanfold (8.5 in. by 13 in.).

```
public static PaperSize GermanLegalFanfold { get; }
```

Property Value

[PaperSize](#)

41

GermanStandardFanfold

Gets the german standard fanfold (8.5 in. by 12 in.).

```
public static PaperSize GermanStandardFanfold { get; }
```

Property Value

[PaperSize](#)

40

Height

Gets the height.

```
public int Height { get; }
```

Property Value

[int↗](#)

The height.

InviteEnvelope

Gets the invite envelope (220 mm by 220 mm).

```
public static PaperSize InviteEnvelope { get; }
```

Property Value

[PaperSize](#)

47

IsoB4

Gets the ISO B4 (250 mm by 353 mm).

```
public static PaperSize IsoB4 { get; }
```

Property Value

[PaperSize](#)

42

ItalyEnvelope

Gets the italy envelope (110 mm by 230 mm).

```
public static PaperSize ItalyEnvelope { get; }
```

Property Value

[PaperSize](#)

36

JapanesePostcard

Gets the japanese postcard (200 mm by 148 mm).

```
public static PaperSize JapanesePostcard { get; }
```

Property Value

[PaperSize](#)

43

Ledger

Gets the ledger paper (17 in. by 11 in.).

```
public static PaperSize Ledger { get; }
```

Property Value

[PaperSize](#)

4

Legal

Gets the legal paper (8.5 in. by 14 in.).

```
public static PaperSize Legal { get; }
```

Property Value

[PaperSize](#)

5

LegalExtra

Gets the legal extra paper (9.275 in. by 15 in.).

```
public static PaperSize LegalExtra { get; }
```

Property Value

[PaperSize](#)

51

Letter

Gets the letter paper (8.5 in. by 11 in.).

```
public static PaperSize Letter { get; }
```

Property Value

[PaperSize](#)

1

LetterExtra

Gets the letter extra paper (9.275 in. by 12 in.).

```
public static PaperSize LetterExtra { get; }
```

Property Value

[PaperSize](#)

50

LetterExtraTransverse

Gets the letter extra transverse (9.275 in. by 12 in.).

```
public static PaperSize LetterExtraTransverse { get; }
```

Property Value

[PaperSize](#)

56

LetterPlus

Gets the letter plus paper (8.5 in. by 12.69 in.).

```
public static PaperSize LetterPlus { get; }
```

Property Value

[PaperSize](#)

59

LetterSmall

Gets the letter small paper (8.5 in. by 11 in.).

```
public static PaperSize LetterSmall { get; }
```

Property Value

[PaperSize](#)

2

LetterTransverse

Gets the letter transverse (8.275 in. by 11 in.).

```
public static PaperSize LetterTransverse { get; }
```

Property Value

[PaperSize](#)

54

MonarchEnvelope

Gets the monarch envelope (3.875 in. by 7.5 in.).

```
public static PaperSize MonarchEnvelope { get; }
```

Property Value

[PaperSize](#)

37

Name

Gets the name.

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

Property Value

[string](#) ↗

The display name.

Note

Gets the note paper (8.5 in. by 11 in.).

```
public static PaperSize Note { get; }
```

Property Value

[PaperSize](#)

18

Number10Envelope

Gets the #10 envelope (4.125 in. by 9.5 in.).

```
public static PaperSize Number10Envelope { get; }
```

Property Value

[PaperSize](#)

20

Number11Envelope

Gets the #11 envelope (4.5 in. by 10.375 in.).

```
public static PaperSize Number11Envelope { get; }
```

Property Value

[PaperSize](#)

21

Number12Envelope

Gets the #12 envelope (4.75 in. by 11 in.).

```
public static PaperSize Number12Envelope { get; }
```

Property Value

[PaperSize](#)

22

Number14Envelope

Gets the #14 envelope (5 in. by 11.5 in.).

```
public static PaperSize Number14Envelope { get; }
```

Property Value

[PaperSize](#)

23

Number9Envelope

Gets the #9 envelope (3.875 in. by 8.875 in.).

```
public static PaperSize Number9Envelope { get; }
```

Property Value

[PaperSize](#)

19

PersonalEnvelope

Gets the personal (6 3/4) envelope (3.625 in. by 6.5 in.).

```
public static PaperSize PersonalEnvelope { get; }
```

Property Value

[PaperSize](#)

38

Quarto

Gets the quarto paper (215 mm by 275 mm).

```
public static PaperSize Quarto { get; }
```

Property Value

[PaperSize](#)

15

Standard10x11

Gets the standard paper (10 in. by 11 in.).

```
public static PaperSize Standard10x11 { get; }
```

Property Value

[PaperSize](#)

45

Standard10x14

Gets the standard paper (10 in. by 14 in.)

```
public static PaperSize Standard10x14 { get; }
```

Property Value

[PaperSize](#)

16

Standard11x17

Gets the standard paper (11 in. by 17 in.).

```
public static PaperSize Standard11x17 { get; }
```

Property Value

[PaperSize](#)

17

Standard15x11

Gets the standard paper (15 in. by 11 in.).

```
public static PaperSize Standard15x11 { get; }
```

Property Value

[PaperSize](#)

46

Standard9x11

Gets the standard paper (9 in. by 11 in.).

```
public static PaperSize Standard9x11 { get; }
```

Property Value

[PaperSize](#)

44

Statement

Gets the statement paper (5.5 in. by 8.5 in.).

```
public static PaperSize Statement { get; }
```

Property Value

[PaperSize](#)

6

Tabloid

Gets the tabloid paper (11 in. by 17 in.).

```
public static PaperSize Tabloid { get; }
```

Property Value

[PaperSize](#)

3

USStandardFanfold

Gets the us standard fanfold (14.875 in. by 11 in.).

```
public static PaperSize USStandardFanfold { get; }
```

Property Value

[PaperSize](#)

39

Value

Gets the value.

```
public int Value { get; }
```

Property Value

[int ↗](#)

The value.

Width

Gets the width.

```
public int Width { get; }
```

Property Value

[int ↗](#)

The width.

Methods

CompareTo(PaperSize)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

```
public int CompareTo(PaperSize other)
```

Parameters

other [PaperSize](#)

An object to compare with this instance.

Returns

[int](#)

A value that indicates the relative order of the objects being compared. The return value has these meanings:

Value	Meaning
Less than zero	This instance precedes <code>other</code> in the sort order.
Zero	This instance occurs in the same position in the sort order as <code>other</code> .
Greater than zero	This instance follows <code>other</code> in the sort order.

CompareTo(int)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

```
public int CompareTo(int other)
```

Parameters

`other` [int](#)

An object to compare with this instance.

Returns

[int](#)

A value that indicates the relative order of the objects being compared. The return value has these meanings:

Value	Meaning
Less than zero	This instance precedes <code>other</code> in the sort order.
Zero	This instance occurs in the same position in the sort order as <code>other</code> .

Value	Meaning
Greater than zero	This instance follows <code>other</code> in the sort order.

CompareTo(object)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

```
public int CompareTo(object obj)
```

Parameters

`obj` [object](#)

An object to compare with this instance.

Returns

[int](#)

A value that indicates the relative order of the objects being compared. The return value has these meanings:

Value	Meaning
Less than zero	This instance precedes <code>obj</code> in the sort order.
Zero	This instance occurs in the same position in the sort order as <code>obj</code> .
Greater than zero	This instance follows <code>obj</code> in the sort order.

Exceptions

[ArgumentException](#)

`obj` is not the same type as this instance.

Equals(PaperSize)

Indicates whether the current object is equal to another object of the same type.

```
public bool Equals(PaperSize other)
```

Parameters

other [PaperSize](#)

An object to compare with this object.

Returns

[bool](#)

[true](#) if the current object is equal to the **other** parameter; otherwise, [false](#).

Equals(int)

Indicates whether the current object is equal to another object of the same type.

```
public bool Equals(int other)
```

Parameters

other [int](#)

An object to compare with this object.

Returns

[bool](#)

[true](#) if the current object is equal to the **other** parameter; otherwise, [false](#).

Equals(object)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object obj)
```

Parameters

obj [object](#)

The object to compare with the current object.

Returns

[bool](#)

[true](#) if the specified object is equal to the current object; otherwise, [false](#).

GetAll()

Gets the items.

```
public static IEnumerable<PaperSize> GetAll()
```

Returns

[IEnumerable](#) <[PaperSize](#)>

The all paper sizes.

GetHashCode()

Serves as the default hash function.

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

Returns

[int](#)

A hash code for the current object.

ToString()

Converts to string.

```
public override string ToString()
```

Returns

[string](#)

The name.

Operators

operator ==(PaperSize, PaperSize)

Implements the operator ==.

```
public static bool operator ==(PaperSize left, PaperSize right)
```

Parameters

left [PaperSize](#)

The left.

right [PaperSize](#)

The right.

Returns

[bool](#)

The result of the operator.

operator ==(PaperSize, int)

Implements the operator ==.

```
public static bool operator ==(PaperSize left, int right)
```

Parameters

left [PaperSize](#)

The left.

right [int](#)

The right.

Returns

[bool](#)

The result of the operator.

operator ==(int, PaperSize)

Implements the operator ==.

```
public static bool operator ==(int left, PaperSize right)
```

Parameters

left [int](#)

The left.

right [PaperSize](#)

The right.

Returns

[bool](#)

The result of the operator.

explicit operator PaperSize(int)

Performs an explicit conversion from [int](#) to [PaperSize](#).

```
public static explicit operator PaperSize(int value)
```

Parameters

value [int](#)

The value.

Returns

[PaperSize](#)

The result of the conversion.

Exceptions

[InvalidOperationException](#)

value does not correspond to any valid PaperSize.

implicit operator int(PaperSize)

Performs an implicit conversion from [PaperSize](#) to [int](#).

```
public static implicit operator int(PaperSize value)
```

Parameters

value [PaperSize](#)

The value.

Returns

[int](#)

The result of the conversion.

operator !=(PaperSize, PaperSize)

Implements the operator !=.

```
public static bool operator !=(PaperSize left, PaperSize right)
```

Parameters

left [PaperSize](#)

The left.

right [PaperSize](#)

The right.

Returns

[bool](#) ↗

The result of the operator.

operator !=(PaperSize, int)

Implements the operator !=.

```
public static bool operator !=(PaperSize left, int right)
```

Parameters

left [PaperSize](#)

The left.

right [int](#) ↗

The right.

Returns

[bool](#) ↗

The result of the operator.

operator !=(int, PaperSize)

Implements the operator !=.

```
public static bool operator !=(int left, PaperSize right)
```

Parameters

left [int](#)

The left.

right [PaperSize](#)

The right.

Returns

[bool](#)

The result of the operator.

Class SheetCreatedEventArgs

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Event arguments after sheet create.

```
public class SheetCreatedEventArgs
```

Inheritance

[object](#) ← SheetCreatedEventArgs

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

SheetCreatedEventArgs(object, SheeterContext)

Event arguments after sheet create.

```
public SheetCreatedEventArgs(object sheetObject, SheeterContext sheeterContext)
```

Parameters

[sheetObject](#) [object](#)

The sheet object that represents the Excel sheet from the underlying Excel library.

[sheeterContext](#) [SheeterContext](#)

The sheeter contexts.

Exceptions

[ArgumentNullException](#)

sheetObject or sheeterContext

Properties

SheetObject

Gets or sets the sheet object that represents the Excel sheet from the underlying Excel library.

```
public object SheetObject { get; }
```

Property Value

[object](#)

The sheet object.

Exceptions

[ArgumentNullException](#)

sheetObject or sheeterContext

SheeterContext

Gets the sheeter contexts.

```
public SheeterContext SheeterContext { get; }
```

Property Value

[SheeterContext](#)

The sheeter contexts.

Exceptions

[ArgumentNullException](#)

sheetObject or sheeterContext

Class Sheeter

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The sheeter.

```
public class Sheeter
```

Inheritance

[object](#) ← Sheeter

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ColumnWidths

Gets the width of the columns.

```
public IReadOnlyDictionary<int, double> ColumnWidths { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [double](#)>

The width of columns.

DefaultRowHeight

Gets or sets the default height of the row.

```
public double? DefaultRowHeight { get; set; }
```

PropertyValue

[double](#)?

The default height of the row.

PageSettings

Gets the page settings.

```
public PageSettings PageSettings { get; }
```

PropertyValue

[PageSettings](#)

The page settings.

Password

Gets or sets the password.

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

PropertyValue

[string](#)

The password.

SheetName

Gets or sets the name of the sheet.

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

PropertyValue

[string](#)

The name of the sheet.

Templates

Gets the templates.

```
public IReadOnlyList<ITemplate> Templates { get; }
```

Property Value

[IReadOnlyList](#)<ITemplate>

The templates.

Methods

AddTemplate(ITemplate)

Adds the template.

```
public void AddTemplate(ITemplate template)
```

Parameters

[template](#) [ITemplate](#)

The template.

AddTemplates(params ITemplate[])

Adds the templates.

```
public void AddTemplates(params ITemplate[] templates)
```

Parameters

`templates` [ITemplate\[\]](#)

The templates.

AddTemplates(IEnumerable<ITemplate>)

Adds the templates.

```
public void AddTemplates(IEnumerable<ITemplate> templates)
```

Parameters

`templates` [IEnumerable<ITemplate>](#)

The templates.

SetColumnWidth(int, double)

Sets the width of the column.

```
public void SetColumnWidth(int index, double width)
```

Parameters

`index` [int](#)

The zero-based column index.

`width` [double](#)

The column width in characters. Use [HiddenColumn](#) (0) to hide the column. Use [AutoFitColumnWidth](#) (-1) to auto-fit the column width. Positive values specify the exact width in character units.

Class SheeterContext

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The sheeter context.

```
public class SheeterContext
```

Inheritance

[object](#) ← SheeterContext

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

SheeterContext(Sheeter)

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

```
public SheeterContext(Sheeter sheeter)
```

Parameters

sheeter [Sheeter](#)

The sheeter.

Properties

Cells

Gets the cells.

```
public IReadOnlyList<Cell> Cells { get; }
```

Property Value

[IReadOnlyList](#)<[Cell](#)>

The cells.

ColumnWidths

Gets the width of columns.

```
public IReadOnlyDictionary<int, double> ColumnWidths { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [double](#)>

The width of columns.

DefaultRowHeight

Gets or sets the default height of the row.

```
public double? DefaultRowHeight { get; }
```

Property Value

[double](#)?

The default height of the row.

FreezePanes

Gets the freeze panes position. The column and row at this position will be the first unfrozen cell. For example, (1, 1) freezes the first column and first row (A1 becomes the frozen pane).

```
public Point? FreezePanes { get; }
```

Property Value

[Point?](#)

The freeze panes position. Null means no freeze panes.

IsAutoFilterEnabled

Gets a value indicating whether auto filter is enabled for the data range.

```
public bool IsAutoFilterEnabled { get; }
```

Property Value

[bool](#)

true if auto filter is enabled; otherwise, **false**.

IsProtected

Gets a value indicating whether this instance is protected.

```
public bool IsProtected { get; }
```

Property Value

[bool](#)

true if this instance is protected; otherwise, **false**.

PageSettings

Gets the page settings.

```
public PageSettings PageSettings { get; }
```

Property Value

[PageSettings](#)

The page settings.

Password

Gets the password.

```
public string Password { get; }
```

Property Value

[string](#)

The password.

RowHeights

Gets the height of rows.

```
public IReadOnlyDictionary<int, double?> RowHeights { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [double](#)?>

The height of rows.

SheetName

Gets the name of the sheet.

```
public string SheetName { get; }
```

Property Value

[string](#) ↗

The name of the sheet.

Class SpreadsheetExportedEventArgs

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Event arguments after spreadsheet export.

```
public class SpreadsheetExportedEventArgs
```

Inheritance

[object](#) ← SpreadsheetExportedEventArgs

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

SpreadsheetExportedEventArgs(IEnumerable<SheeterContext>)

Event arguments after spreadsheet export.

```
public SpreadsheetExportedEventArgs(IEnumerable<SheeterContext> sheeterContexts)
```

Parameters

sheeterContexts [IEnumerable](#)<[SheeterContext](#)>

The sheeter contexts.

Exceptions

[ArgumentNullException](#)

sheeterContexts

Properties

SheeterContexts

Gets the sheeter contexts.

```
public IEnumerable<SheeterContext> SheeterContexts { get; }
```

Property Value

[IEnumerable](#) <[SheeterContext](#)>

The sheeter contexts.

Exceptions

[ArgumentNullException](#)

sheeterContexts

Class SpreadsheetExportingEventArgs

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

Event arguments before spreadsheet export.

```
public class SpreadsheetExportingEventArgs
```

Inheritance

[object](#) ← SpreadsheetExportingEventArgs

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

SpreadsheetExportingEventArgs(IEnumerable<SheeterContext>)

Event arguments before spreadsheet export.

```
public SpreadsheetExportingEventArgs(IEnumerable<SheeterContext> sheeterContexts)
```

Parameters

sheeterContexts [IEnumerable](#)<[SheeterContext](#)>

The sheeter contexts.

Exceptions

[ArgumentNullException](#)

sheeterContexts

Properties

SheeterContexts

Gets the sheeter contexts.

```
public IEnumerable<SheeterContext> SheeterContexts { get; }
```

Property Value

[IEnumerable](#) <[SheeterContext](#)>

The sheeter contexts.

Exceptions

[ArgumentNullException](#)

sheeterContexts

Enum Spreadsheet FileMode

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The file mode of spreadsheet

```
public enum Spreadsheet FileMode
```

Fields

Create = 0

Create a file. If the file already exists, it will be overwritten.

CreateNew = 1

Create a new file. If the file already exists, an IOException exception is thrown.

Class SpreadsheetManager

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The spreadsheet manager.

```
public static class SpreadsheetManager
```

Inheritance

[object](#) ← SpreadsheetManager

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

DefaultCellStyles

Gets or sets the configuration.

```
public static CellStyleConfiguration DefaultCellStyles { get; set; }
```

Property Value

[CellStyleConfiguration](#)

The configuration.

Methods

CreateExporter()

Creates the exporter.

```
public static ISpreadsheetExporter CreateExporter()
```

Returns

[ISpreadsheetExporter](#)

The exporter.

Exceptions

[InvalidOperationException](#)

Exporter factory is not set.

SetExporter(Func<ISpreadsheetExporter>)

Sets the exporter.

```
public static void SetExporter(Func<ISpreadsheetExporter> exporterFactory)
```

Parameters

exporterFactory [Func](#)<[ISpreadsheetExporter](#)>

The exporter factory.

Exceptions

[ArgumentNullException](#)

exporterFactory

[ArgumentException](#)

Factory return value cannot be null. - exporterFactory

Enum VerticalAlignment

Namespace: [CloudyWing.SpreadsheetExporter](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The vertical text alignment.

```
public enum VerticalAlignment
```

Fields

Bottom = 2

The bottom.

Middle = 1

The middle.

Top = 0

The top.

Namespace CloudyWing.SpreadsheetExporter.Config

Classes

[CellStyleConfiguration](#)

The spreadsheet default style.

[CellStyleSetuper](#)

The cell style setuper.

Class CellStyleConfiguration

Namespace: [CloudyWing.SpreadsheetExporter.Config](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The spreadsheet default style.

```
public class CellStyleConfiguration
```

Inheritance

[object](#) ← CellStyleConfiguration

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

CellStyleConfiguration()

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

```
CellStyle: {
    HorizontalAlignment: None,
    VerticalAlignment: Middle,
    HasBorder: false,
    WrapText: false,
    Font: {
        Name: "新細明體",
        Size: 10,
        IsBold: false,
        IsItalic: false,
        HasUnderline: false,
        IsStrikeout: false
    }
}

GridCellStyle: {
    StyleBase: CellStyle
}
```

```
HeaderStyle: {
    StyleBase: CellStyle
    HorizontalAlignment: Center,
    HasBorder: true,
    Font: {
        IsBold: true,
    }
}

FieldStyle: {
    StyleBase: CellStyle
    HasBorder: true
}

public CellStyleConfiguration()
```

CellStyleConfiguration(Action<CellStyleSetuper>)

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

```
public CellStyleConfiguration(Action<CellStyleSetuper> loader)
```

Parameters

loader [Action](#)<[CellStyleSetuper](#)>

The loader.

Properties

CellStyle

Gets the cell style.

```
public virtual CellStyle CellStyle { get; }
```

Property Value

[CellStyle](#)

The cell style.

FieldStyle

Gets the field style.

```
public virtual CellStyle FieldStyle { get; }
```

Property Value

[CellStyle](#)

The field style.

GridCellStyle

Gets the grid cell style.

```
public virtual CellStyle GridCellStyle { get; }
```

Property Value

[CellStyle](#)

The grid cell style.

HeaderStyle

Gets the header style.

```
public virtual CellStyle HeaderStyle { get; }
```

Property Value

[CellStyle](#)

The header style.

Class CellStyleSetuper

Namespace: [CloudyWing.SpreadsheetExporter.Config](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The cell style setuper.

```
public sealed class CellStyleSetuper
```

Inheritance

[object](#) ← CellStyleSetuper

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

CellStyleSetuper(CellStyleConfiguration)

The cell style setuper.

```
public CellStyleSetuper(CellStyleConfiguration configuration)
```

Parameters

configuration [CellStyleConfiguration](#)

The configuration.

Exceptions

[ArgumentNullException](#)

configuration

Properties

CellStyle

Gets or sets the cell style.

```
public CellStyle CellStyle { get; set; }
```

Property Value

[CellStyle](#)

The cell style.

Exceptions

[ArgumentNullException](#)

configuration

FieldStyle

Gets or sets the field style.

```
public CellStyle FieldStyle { get; set; }
```

Property Value

[CellStyle](#)

The field style.

Exceptions

[ArgumentNullException](#)

configuration

GridCellStyle

Gets or sets the grid cell style.

```
public CellStyle GridCellStyle { get; set; }
```

Property Value

[CellStyle](#)

The grid cell style.

Exceptions

[ArgumentNullException](#)

configuration

HeaderStyle

Gets or sets the header style.

```
public CellStyle HeaderStyle { get; set; }
```

Property Value

[CellStyle](#)

The header style.

Exceptions

[ArgumentNullException](#)

configuration

Namespace CloudyWing.SpreadsheetExporter.Excel.EPPlus

Classes

[ExcelExporter](#)

The excel exporter, using epplus.

Class ExcelExporter

Namespace: [CloudyWing.SpreadsheetExporter.Excel.EPPlus](#)

Assembly: CloudyWing.SpreadsheetExporter.Excel.EPPlus.dll

The excel exporter, using epplus.

```
public class ExcelExporter : ExporterBase, ISpreadsheetExporter
```

Inheritance

[object](#) ← [ExporterBase](#) ← ExcelExporter

Implements

[ISpreadsheetExporter](#)

Inherited Members

[ExporterBase.SpreadsheetExportingEvent](#) , [ExporterBase.SpreadsheetExportedEvent](#) ,
[ExporterBase.SheetCreatedEvent](#) , [ExporterBase.Password](#) , [ExporterBase.HasPassword](#) ,
[ExporterBase.DefaultFont](#) , [ExporterBase.DefaultBasicSheetName](#) , [ExporterBase.LastSheeter](#) ,
[ExporterBase.CreateSheeter\(string, double?\)](#) , [ExporterBase.GetSheeter\(int\)](#) ,
[ExporterBase.TryGetSheeter\(int, out Sheeter\)](#) , [ExporterBase.Export\(\)](#) ,
[ExporterBase.OnSpreadsheetExporting\(SpreadsheetExportingEventArgs\)](#) ,
[ExporterBase.OnSpreadsheetExported\(SpreadsheetExportedEventArgs\)](#) ,
[ExporterBase.OnSheetCreated\(SheetCreatedEventArgs\)](#) ,
[ExporterBase.ExportFile\(string, Spreadsheet FileMode\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ContentType

Gets the content-type.

```
public override string ContentType { get; }
```

Property Value

[string](#)

The content-type.

FileNameExtension

Gets the file name extension.

```
public override string FileNameExtension { get; }
```

Property Value

[string](#)

The file name extension.

IsClosedNotImplementedException

Gets or sets a value indicating whether this instance is closed not implemented exception.

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

Property Value

[bool](#)

`true` if this instance is closed not implemented exception; otherwise, `false`.

Methods

ExecuteExport(IEnumerable<SheeterContext>)

Executes the export.

```
protected override byte[] ExecuteExport(IEnumerable<SheeterContext> contexts)
```

Parameters

contexts [IEnumerable](#) <[SheeterContext](#)>

The contexts.

Returns

[byte](#) []

The exported bytes of the file.

See Also

[ExporterBase](#)

Namespace CloudyWing.SpreadsheetExporter.Excel.NPOI

Classes

[ExcelExporter](#)

The excel exporter, using npoi.

Enums

[ExcelFormat](#)

The excel format.

Class ExcelExporter

Namespace: [CloudyWing.SpreadsheetExporter.Excel.NPOI](#)

Assembly: CloudyWing.SpreadsheetExporter.Excel.NPOI.dll

The excel exporter, using npoi.

```
public sealed class ExcelExporter : ExporterBase, ISpreadsheetExporter
```

Inheritance

[object](#) ← [ExporterBase](#) ← ExcelExporter

Implements

[ISpreadsheetExporter](#)

Inherited Members

[ExporterBase.SpreadsheetExportingEvent](#) , [ExporterBase.SpreadsheetExportedEvent](#) ,
[ExporterBase.SheetCreatedEvent](#) , [ExporterBase.Password](#) , [ExporterBase.HasPassword](#) ,
[ExporterBase.DefaultFont](#) , [ExporterBase.DefaultBasicSheetName](#) , [ExporterBase.LastSheeter](#) ,
[ExporterBase.CreateSheeter\(string, double?\)](#) , [ExporterBase.GetSheeter\(int\)](#) ,
[ExporterBase.TryGetSheeter\(int, out Sheeter\)](#) , [ExporterBase.Export\(\)](#) ,
[ExporterBase.ExportFile\(string, Spreadsheet FileMode\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

ExcelExporter(ExcelFormat)

The excel exporter, using npoi.

```
public ExcelExporter(ExcelFormat excelFormat = ExcelFormat.OfficeOpenXmlDocument)
```

Parameters

excelFormat [ExcelFormat](#)

The excel format.

See Also

[ExporterBase](#)

Properties

ContentType

Gets the content-type.

```
public override string ContentType { get; }
```

Property Value

[string](#)

The content-type.

ExcelFormat

Gets or sets the excel format.

```
public ExcelFormat ExcelFormat { get; set; }
```

Property Value

[ExcelFormat](#)

The excel format.

FileNameExtension

Gets the file name extension.

```
public override string FileNameExtension { get; }
```

Property Value

[string](#)

The file name extension.

IsClosedNotImplementedException

Gets or sets a value indicating whether this instance is closed not implemented exception.

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

Property Value

[bool](#)

`true` if this instance is closed not implemented exception; otherwise, `false`.

IsOfficeOpenXmlDocument

Gets a value indicating whether this instance is office open XML document.

```
public bool IsOfficeOpenXmlDocument { get; }
```

Property Value

[bool](#)

`true` if this instance is office open XML document; otherwise, `false`.

Methods

ExecuteExport(IEnumerable<SheeterContext>)

Executes the export.

```
protected override byte[] ExecuteExport(IEnumerable<SheeterContext> contexts)
```

Parameters

contexts [IEnumerable](#) <[SheeterContext](#)>

The contexts.

Returns

[byte](#) []

The exported bytes of the file.

See Also

[ExporterBase](#)

Enum ExcelFormat

Namespace: [CloudyWing.SpreadsheetExporter.Excel.NPOI](#)

Assembly: CloudyWing.SpreadsheetExporter.Excel.NPOI.dll

The excel format.

```
public enum ExcelFormat
```

Fields

ExcelBinaryFileFormat = 0

The filename extension is **.xls**.

OfficeOpenXmlDocument = 1

The filename extension is **.xlsx**.

Namespace CloudyWing.SpreadsheetExporter.Exceptions

Classes

[SheeterNotFoundException](#)

The exception for sheeter not found.

Class SheeterNotFoundException

Namespace: [CloudyWing.SpreadsheetExporter.Exceptions](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The exception for sheeter not found.

```
public class SheeterNotFoundException : Exception, ISerializable
```

Inheritance

[object](#) ← [Exception](#) ← SheeterNotFoundException

Implements

[ISerializable](#)

Inherited Members

[Exception.GetBaseException\(\)](#) , [Exception.GetType\(\)](#) , [Exception.ToString\(\)](#) , [Exception.Data](#) ,
[Exception.HelpLink](#) , [Exception.HResult](#) , [Exception.InnerException](#) , [Exception.Message](#) ,
[Exception.Source](#) , [Exception.StackTrace](#) , [Exception.TargetSite](#) , [Exception.SerializeObjectState](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

SheeterNotFoundException()

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

```
public SheeterNotFoundException()
```

SheeterNotFoundException(SerializationInfo, StreamingContext)

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

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

Parameters

info [SerializationInfo](#)

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

context [StreamingContext](#)

The [StreamingContext](#) that contains contextual information about the source or destination.

SheeterNotFoundException(string)

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

```
public SheeterNotFoundException(string message)
```

Parameters

message [string](#)

The message that describes the error.

SheeterNotFoundException(string, Exception)

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

```
public SheeterNotFoundException(string message, Exception innerException)
```

Parameters

message [string](#)

The error message that explains the reason for the exception.

innerException [Exception](#)

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

See Also

Exception

Namespace CloudyWing.SpreadsheetExporter.Templates

Classes

[MergedTemplate](#)

The merged template. Merge templates into new template.

[TemplateContext](#)

The template context.

Interfaces

[ITemplate](#)

The template interface.

Interface ITemplate

Namespace: [CloudyWing.SpreadsheetExporter.Templates](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The template interface.

```
public interface ITemplate
```

Methods

GetContext()

Gets the context.

```
TemplateContext GetContext()
```

Returns

[TemplateContext](#)

The template context.

Class MergedTemplate

Namespace: [CloudyWing.SpreadsheetExporter.Templates](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The merged template. Merge templates into new template.

```
public class MergedTemplate : ITemplate
```

Inheritance

[object](#) ← MergedTemplate

Implements

[ITemplate](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

MergedTemplate(params ITemplate[])

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

```
public MergedTemplate(params ITemplate[] templates)
```

Parameters

templates [ITemplate](#)[]

The templates.

Exceptions

[ArgumentNullException](#)

templates

MergedTemplate(IEnumerable<ITemplate>)

The merged template. Merge templates into new template.

```
public MergedTemplate(IEnumerable<ITemplate> templates)
```

Parameters

templates [IEnumerable](#)<[ITemplate](#)>

The templates.

Exceptions

[ArgumentNullException](#)

templates

See Also

[ITemplate](#)

Methods

GetContext()

Gets the context.

```
public TemplateContext GetContext()
```

Returns

[TemplateContext](#)

The template context.

Exceptions

[ArgumentNullException](#)

templates

See Also

[ITemplate](#)

Class TemplateContext

Namespace: [CloudyWing.SpreadsheetExporter.Templates](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The template context.

```
public class TemplateContext
```

Inheritance

[object](#) ← TemplateContext

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

TemplateContext(IEnumerable<Cell>, int,
 IReadOnlyDictionary<int, double?>)

The template context.

```
public TemplateContext(IEnumerable<Cell> cells, int rowSpan, IReadOnlyDictionary<int, double?> rowHeights)
```

Parameters

cells [IEnumerable](#)<Cell>

The cells.

rowSpan [int](#)

The row span.

rowHeights [IReadOnlyDictionary](#)<[int](#), [double](#)?>

The row heights.

Properties

Cells

Gets the cells.

```
public IReadOnlyList<Cell> Cells { get; }
```

Property Value

[IReadOnlyList](#)<[Cell](#)>

The cells.

FreezePanes

Gets or sets the freeze panes position. The column and row at this position will be the first unfrozen cell. For example, (1, 1) freezes the first column and first row (A1 becomes the frozen pane).

```
public Point? FreezePanes { get; set; }
```

Property Value

[Point](#)?

The freeze panes position. Null means no freeze panes.

IsAutoFilterEnabled

Gets or sets a value indicating whether to enable auto filter for the data range. Auto filter is typically applied to header rows in data tables.

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

Property Value

[bool](#)

`true` if auto filter is enabled; otherwise, `false`. Default is `false`.

RowHeights

Gets the height of rows.

```
public IReadOnlyDictionary<int, double?> RowHeights { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [double](#)?>

The height of rows.

RowSpan

Gets the row span.

```
public int RowSpan { get; }
```

Property Value

[int](#)

The row span.

Namespace CloudyWing.SpreadsheetExporter.Templates.DataTable

Classes

[DataTableColumn](#)

The DataTable column.

[DataTableColumnCollection](#)

The DataTable column collection.

[DataTableTemplate](#)

The DataTable template. Create cell information using DataTable as data source.

Class DataTableColumn

Namespace: [CloudyWing.SpreadsheetExporter.Templates.DataTable](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The DataTable column.

```
public class DataColumn
```

Inheritance

[object](#) ← DataColumn

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ColumnName

Gets or sets the name of the column in the DataTable.

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

Property Value

[string](#)

The name of the column.

FieldFormulaGenerator

Gets or sets the field formula generator.

```
public Func<object, string> FieldFormulaGenerator { get; set; }
```

PropertyValue

[Func<object, string>](#)

The field formula generator.

FieldStyleGenerator

Gets or sets the field style generator.

```
public Func<object, CellStyle> FieldStyleGenerator { get; set; }
```

PropertyValue

[Func<object, CellStyle>](#)

The field style generator.

FieldValueGenerator

Gets or sets the field value generator.

```
public Func<object, object> FieldValueGenerator { get; set; }
```

PropertyValue

[Func<object, object>](#)

The field value generator.

HeaderStyle

Gets or sets the header style.

```
public CellStyle HeaderStyle { get; set; }
```

PropertyValue

[CellStyle](#)

The header style.

HeaderText

Gets or sets the header text.

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

Property Value

[string](#)

The header text.

Class DataTableColumnCollection

Namespace: [CloudyWing.SpreadsheetExporter.Templates.DataTable](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The DataTable column collection.

```
public class DataTableColumnCollection : Collection<DataTableColumn>,
IList<DataTableColumn>, ICollection<DataTableColumn>, IReadOnlyList<DataTableColumn>,
IReadOnlyCollection<DataTableColumn>, IEnumerable<DataTableColumn>, IList,
ICollection, IEnumerable
```

Inheritance

[object](#) ← [Collection](#)<[DataTableColumn](#)> ← [DataTableColumnCollection](#)

Implements

[IList](#)<[DataTableColumn](#)>, [ICollection](#)<[DataTableColumn](#)>, [IReadOnlyList](#)<[DataTableColumn](#)>,
 [IReadOnlyCollection](#)<[DataTableColumn](#)>, [IEnumerable](#)<[DataTableColumn](#)>, [IList](#), [ICollection](#),
[IEnumerable](#)

Inherited Members

[Collection](#)<[DataTableColumn](#)>.Add([DataTableColumn](#)) , [Collection](#)<[DataTableColumn](#)>.Clear() ,
[Collection](#)<[DataTableColumn](#)>.ClearItems() ,
[Collection](#)<[DataTableColumn](#)>.Contains([DataTableColumn](#)) ,
[Collection](#)<[DataTableColumn](#)>.CopyTo([DataTableColumn](#)[], [int](#)) ,
[Collection](#)<[DataTableColumn](#)>.GetEnumerator() ,
[Collection](#)<[DataTableColumn](#)>.IndexOf([DataTableColumn](#)) ,
[Collection](#)<[DataTableColumn](#)>.Insert([int](#), [DataTableColumn](#)) ,
[Collection](#)<[DataTableColumn](#)>.InsertItem([int](#), [DataTableColumn](#)) ,
[Collection](#)<[DataTableColumn](#)>.Remove([DataTableColumn](#)) ,
[Collection](#)<[DataTableColumn](#)>.RemoveAt([int](#)) , [Collection](#)<[DataTableColumn](#)>.RemoveItem([int](#)) ,
[Collection](#)<[DataTableColumn](#)>.SetItem([int](#), [DataTableColumn](#)) ,
[Collection](#)<[DataTableColumn](#)>.Count , [Collection](#)<[DataTableColumn](#)>.this[[int](#)] ,
[Collection](#)<[DataTableColumn](#)>.Items , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

See Also

[Collection](#)<[T](#)>

Class DataTableTemplate

Namespace: [CloudyWing.SpreadsheetExporter.Templates.DataTable](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The DataTable template. Create cell information using DataTable as data source.

```
public class DataTableTemplate : ITemplate
```

Inheritance

[object](#) ← DataTableTemplate

Implements

[ITemplate](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

DataTableTemplate(DataTable)

The DataTable template. Create cell information using DataTable as data source.

```
public DataTableTemplate(DataTable dataTable)
```

Parameters

dataTable [DataTable](#)

The data table.

Exceptions

[ArgumentNullException](#)

dataTable

See Also

[ITemplate](#)

Properties

ColumnSpan

Gets the column span.

```
public int ColumnSpan { get; }
```

Property Value

[int](#)

The column span.

Exceptions

[ArgumentNullException](#)

dataTable

Columns

Gets the columns.

```
public DataTableColumnCollection Columns { get; }
```

Property Value

[DataTableColumnCollection](#)

The columns.

Exceptions

[ArgumentNullException](#)

dataTable

DataTable

Gets or sets the data table.

```
public DataTable DataTable { get; set; }
```

Property Value

[DataTable](#)

The data table.

Exceptions

[ArgumentNullException](#)

dataTable

HeaderHeight

Gets or sets the height of the header.

```
public double? HeaderHeight { get; set; }
```

Property Value

[double](#)?

The height of the header.

Exceptions

[ArgumentNullException](#)

dataTable

RecordHeight

Gets or sets the height of the record.

```
public double? RecordHeight { get; set; }
```

Property Value

[double](#)?

The height of the record.

Exceptions

[ArgumentNullException](#)

dataTable

RowSpan

Gets the row span.

```
public int RowSpan { get; }
```

Property Value

[int](#)

The row span.

Exceptions

[ArgumentNullException](#)

dataTable

Methods

[GetContext\(\)](#)

Gets the context.

```
public TemplateContext GetContext()
```

Returns

[TemplateContext](#)

The template context.

Exceptions

[ArgumentNullException](#)

dataTable

See Also

[ITemplate](#)

Namespace CloudyWing.SpreadsheetExporter.Templates.Grid

Classes

[GridTemplate](#)

The grid template. Create cell information using `CreateRow()` and `CreateCell()`.

Class GridTemplate

Namespace: [CloudyWing.SpreadsheetExporter.Templates.Grid](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The grid template. Create cell information using `CreateRow()` and `CreateCell()`.

```
public class GridTemplate : ITemplate
```

Inheritance

[object](#) ← GridTemplate

Implements

[ITemplate](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ColumnSpan

Gets the column span.

```
public int ColumnSpan { get; }
```

Property Value

[int](#)

The column span.

RowSpan

Gets the row span.

```
public int RowSpan { get; }
```

Property Value

[int](#)

The row span.

Methods

CreateCell(Action<Cell>, int, int, CellStyle?)

Creates a cell with custom configuration using a configurator action. This method provides full flexibility for configuring cell properties including value, formula, data validation, and style.

```
public GridTemplate CreateCell(Action<Cell> configurator, int columnSpan = 1, int rowspan = 1, CellStyle? cellStyle = null)
```

Parameters

configurator [Action](#)<[Cell](#)>

The action to configure the cell. The action receives the cell instance to configure.

columnSpan [int](#)

The column span.

rowSpan [int](#)

The row span.

cellStyle [CellStyle](#)?

The cell style. The default is [SpreadsheetManager.DefaultCellStyles.GridCellStyle](#).

Returns

[GridTemplate](#)

The self.

Exceptions

[ArgumentOutOfRangeException](#)

`columnSpan` must be greater than 0.

[ArgumentOutOfRangeException](#)

`rowSpan` must be greater than 0.

CreateCell(Func<int, int, string>, int, int, CellStyle?)

Create cell that contain formula.

```
public GridTemplate CreateCell(Func<int, int, string> formulaGenerator, int columnSpan = 1,  
int rowspan = 1, CellStyle? cellStyle = null)
```

Parameters

`formulaGenerator Func<int, int, string>`

The formula generator. Pass the cell index and row index to the generator. The index start at 0.

`columnSpan int`

The column span.

`rowSpan int`

The row span.

`cellStyle CellStyle?`

The cell style. The default is `SpreadsheetManager.DefaultCellStyles.GridCellStyle`.

Returns

[GridTemplate](#)

The self.

Exceptions

[ArgumentException](#)

`columnSpan` must be greater than 0.

[ArgumentException](#)

`rowSpan` must be greater than 0.

CreateCell(object, int, int, CellStyle?)

Creates the cell.

```
public GridTemplate CreateCell(object value, int columnSpan = 1, int rowspan = 1, CellStyle? cellStyle = null)
```

Parameters

`value` [object](#)

The value.

`columnSpan` [int](#)

The column span.

`rowSpan` [int](#)

The row span.

`cellStyle` [CellStyle](#)?

The cell style. The default is `SpreadsheetManager.DefaultCellStyles.GridCellStyle`.

Returns

[GridTemplate](#)

The self.

Exceptions

[ArgumentException](#)

`columnSpan` must be greater than 0.

[ArgumentOutOfRangeException](#)

`rowSpan` must be greater than 0.

CreateRow(double?)

Creates the row.

```
public GridTemplate CreateRow(double? height = null)
```

Parameters

`height double?`

The height.

Returns

[GridTemplate](#)

The self.

GetContext()

Gets the context.

```
public TemplateContext GetContext()
```

Returns

[TemplateContext](#)

The template context.

Namespace CloudyWing.SpreadsheetExporter.Templates.RecordSet

Classes

[DataColumnBase<T>](#)

The data column base.

[DataColumnCollection<T>](#)

The data column collection.

[DataColumnCollection<T>.GeneratorProvider<TRecord, TContext>](#)

The generator provider.

[FieldContext<TRecord, TField>](#)

The field context.

[RecordContext<T>](#)

The record context.

[RecordSetTemplate<T>](#)

The recordset template. Create cell information using set data source and data column.

Class DataColumnBase<T>

Namespace: [CloudyWing.SpreadsheetExporter.Templates.RecordSet](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The data column base.

```
public abstract class DataColumnBase<T>
```

Type Parameters

T

The type of the record.

Inheritance

[object](#) ← DataColumnBase<T>

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ChildColumns

Gets the child columns.

```
public DataColumnCollection<T> ChildColumns { get; }
```

Property Value

[DataColumnCollection](#)<T>

The child columns.

ColumnLayers

Gets the column layers.

```
public int ColumnLayers { get; }
```

Property Value

[int↗](#)

The column layers.

Remarks

自己和子層共幾層 Column, 用來計算 RowSpan

ColumnSpan

Gets the column span.

```
public int ColumnSpan { get; }
```

Property Value

[int↗](#)

The column span.

HeaderStyle

Gets or sets the header style.

```
public CellStyle HeaderStyle { get; set; }
```

Property Value

[CellStyle](#)

The header style.

HeaderText

Gets or sets the header text.

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

Property Value

[string](#)

The header text.

ParentColumns

Gets the parent columns.

```
public DataColumnCollection<T> ParentColumns { get; }
```

Property Value

[DataColumnCollection](#)<T>

The parent columns.

Point

Gets the point.

```
public Point Point { get; }
```

Property Value

[Point](#)

The point.

RowSpan

Gets the row span.

```
public int RowSpan { get; }
```

Property Value

[int↗](#)

The row span.

Methods

GetFieldDataValidation(RecordContext<T>)

Gets the field data validation.

```
public abstract DataValidation GetFieldDataValidation(RecordContext<T> context)
```

Parameters

[context RecordContext<T>](#)

The context.

Returns

[DataValidation](#)

Returns the field data validation, or [null](#) if no validation is specified.

GetFieldFormula(RecordContext<T>)

Gets the field formula.

```
public abstract string GetFieldFormula(RecordContext<T> context)
```

Parameters

```
context RecordContext<T>
```

The context.

Returns

```
string
```

Returns the field formula.

GetFieldStyle(RecordContext<T>)

Gets the field style.

```
public abstract CellStyle GetFieldStyle(RecordContext<T> context)
```

Parameters

```
context RecordContext<T>
```

The context.

Returns

```
CellStyle
```

Returns the field style.

GetFieldValue(RecordContext<T>)

Gets the field value.

```
public abstract object GetFieldValue(RecordContext<T> context)
```

Parameters

```
context RecordContext<T>
```

The context.

Returns

object ↗

Returns the field value.

Class DataColumnCollection<T>

Namespace: [CloudyWing.SpreadsheetExporter.Templates.RecordSet](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The data column collection.

```
public class DataColumnCollection<T> : Collection<DataColumnBase<T>>,
IList<DataColumnBase<T>>, ICollection<DataColumnBase<T>>, IReadOnlyList<DataColumnBase<T>>,
IReadOnlyCollection<DataColumnBase<T>>, IEnumerable<DataColumnBase<T>>, IList,
ICollection, IEnumerable
```

Type Parameters

T

The type of the record.

Inheritance

[object](#) ↗ ← [Collection](#) ↗ <[DataColumnBase](#)<T>> ← [DataColumnCollection](#)<T>

Implements

[IList](#) ↗ <[DataColumnBase](#)<T>>, [ICollection](#) ↗ <[DataColumnBase](#)<T>>,
[IReadOnlyList](#) ↗ <[DataColumnBase](#)<T>>, [IReadOnlyCollection](#) ↗ <[DataColumnBase](#)<T>>,
[IEnumerable](#) ↗ <[DataColumnBase](#)<T>>, [IList](#), [ICollection](#), [IEnumerable](#)

Inherited Members

[Collection](#)<[DataColumnBase](#)<T>>.Clear() ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.Contains([DataColumnBase](#)<T>) ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.CopyTo([DataColumnBase](#)<T>[], int) ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.GetEnumerator() ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.IndexOf([DataColumnBase](#)<T>) ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.Insert(int, [DataColumnBase](#)<T>) ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.Remove([DataColumnBase](#)<T>) ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.RemoveAt(int) ↗ , [Collection](#)<[DataColumnBase](#)<T>>.Count ↗ ,
[Collection](#)<[DataColumnBase](#)<T>>.this[int] ↗ , [Collection](#)<[DataColumnBase](#)<T>>.Items ↗ ,
[object.Equals](#)([object](#)) ↗ , [object.Equals](#)([object](#), [object](#)) ↗ , [object.GetHashCode](#)() ↗ , [object.GetType](#)() ↗ ,
[object.MemberwiseClone](#)() ↗ , [object.ReferenceEquals](#)([object](#), [object](#)) ↗ , [object.ToString](#)() ↗

Properties

ColumnSpan

Gets the column span.

```
public int ColumnSpan { get; }
```

Property Value

[int](#)

The column span.

DataSourceColumns

Get the column containing the properties of the data source.

```
public IEnumerable<DataColumnBase<T>> DataSourceColumns { get; }
```

Property Value

[IEnumerable](#) <[DataColumnBase](#)<T>>

The data columns.

RootColumns

Gets the root columns.

```
public DataColumnCollection<T> RootColumns { get; }
```

Property Value

[DataColumnCollection](#)<T>

The root columns.

RowSpan

Gets the row span.

```
public int RowSpan { get; }
```

Property Value

[int↗](#)

The row span.

Methods

Add(DataColumnBase<T>)

Adds the specified header text.

```
public DataColumnCollection<T> Add(DataColumnBase<T> dataColumn)
```

Parameters

dataColumn [DataColumnBase<T>](#)

The data column.

Returns

[DataColumnCollection<T>](#)

The self.

Add(string, Action<GeneratorProvider<T, RecordContext<T>>>, CellStyle?, Func<RecordContext<T>, CellStyle>)

Adds the data column to the end of the DataColumnCollection<T>.

```
public DataColumnCollection<T> Add(string headerText,  
Action<DataColumnCollection<T>.GeneratorProvider<T, RecordContext<T>>> providerSetter,  
CellStyle? headerStyle = null, Func<RecordContext<T>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

providerSetter [Action](#)<[DataColumnCollection](#)<T>.GeneratorProvider<T, [RecordContext](#)<T>>>

The provider setter.

headerStyle [CellStyle](#)?

The header style. The default is [SpreadsheetManager.DefaultCellStyles.HeaderStyle](#).

fieldStyleGenerator [Func](#)<[RecordContext](#)<T>, [CellStyle](#)>

The field style generator. The default is `(context) =>`

[SpreadsheetManager.DefaultCellStyles.FieldStyle](#).

Returns

[DataColumnCollection](#)<T>

The self.

Add(string, CellStyle?, Func<RecordContext<T>, CellStyle>)

Adds the specified header text.

```
public DataColumnCollection<T> Add(string headerText, CellStyle? headerStyle = null,  
Func<RecordContext<T>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

`headerStyle` [CellStyle](#)?

The header style.

`fieldStyleGenerator` [Func](#)<[RecordContext](#)<T>, [CellStyle](#)>

The field style generator.

Returns

[DataColumnCollection](#)<T>

The self.

AddChildToLast(DataColumnBase<T>)

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast(DataColumnBase<T> childColumn)
```

Parameters

`childColumn` [DataColumnBase](#)<T>

The child data column.

Returns

[DataColumnCollection](#)<T>

The self.

Exceptions

[InvalidOperationException](#)

No columns available to add child to.

AddChildToLast(string, Action<GeneratorProvider<T, RecordContext<T>>>, CellStyle?, Func<RecordContext<T>,

CellStyle>)

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast(string headerText,  
Action<DataColumnCollection<T>.GeneratorProvider<T, RecordContext<T>>> providerSetter,  
CellStyle? headerStyle = null, Func<RecordContext<T>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

providerSetter [Action](#)<[DataColumnCollection](#)<T>.GeneratorProvider<T, [RecordContext](#)<T>>>

The provider setter.

headerStyle [CellStyle](#)?

The header style. The default is [SpreadsheetManager.DefaultCellStyles.HeaderStyle](#).

fieldStyleGenerator [Func](#)<[RecordContext](#)<T>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

AddChildToLast(string, CellStyle?, Func<RecordContext<T>, CellStyle>)

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast(string headerText, CellStyle? headerStyle =  
null, Func<RecordContext<T>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

headerStyle [CellStyle](#)?

The header style.

fieldStyleGenerator [Func](#)<[RecordContext](#)<T>, [CellStyle](#)>

The field style generator.

Returns

[DataColumnCollection](#)<T>

The self.

AddChildToLast<TField>(string, Expression<Func<T, TField>>, Action<GeneratorProvider<T, FieldContext<T, TField>>>, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast<TField>(string headerText, Expression<Func<T, TField>>> fieldKeyExpression, Action<DataColumnCollection<T>.GeneratorProvider<T, FieldContext<T, TField>>>> providerSetter, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

fieldKeyExpression [Expression](#)<[Func](#)<T, TField>>

The field key expression.

providerSetter [Action](#)<[DataColumnCollection](#)<T>.GeneratorProvider<T, FieldContext<T, TField>>>

The provider setter.

headerStyle [CellStyle](#)?

The header style. The default is `SpreadsheetManager.DefaultCellStyles.HeaderStyle`.

fieldStyleGenerator [Func](#)<[FieldContext](#)<T, TField>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

Type Parameters

TField

The type of the field.

AddChildToLast<TField>(string, Expression<Func<T, TField>>, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast<TField>(string headerText, Expression<Func<T, TField>> fieldKeyExpression, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

fieldKeyExpression [Expression](#)<[Func](#)<T, TField>>

The field key expression.

`headerStyle` [CellStyle](#)?

The header style. The default is `SpreadsheetManager.DefaultCellStyles.HeaderStyle`.

`fieldStyleGenerator` [Func](#)<[FieldContext](#)<T, TField>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

Type Parameters

TField

The type of the field.

`AddChildToLast<TField>(string, string, Action<GeneratorProvider<T, FieldContext<T, TField>>>, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)`

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast<TField>(string headerText, string fieldKey, Action<DataColumnCollection<T>.GeneratorProvider<T, FieldContext<T, TField>>> providerSetter, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle> fieldStyleGenerator = null)
```

Parameters

`headerText` [string](#)

The header text.

`fieldKey` [string](#)

The field key.

`providerSetter Action< DataColumnCollection<T>, GeneratorProvider<T, FieldContext<T, TField>>>`

The provider setter.

`headerStyle CellStyle?`

The header style. The default is `SpreadsheetManager.DefaultCellStyles.HeaderStyle`.

`fieldStyleGenerator Func<FieldContext<T, TField>, CellStyle>`

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

`DataColumnCollection<T>`

The self.

Type Parameters

`TField`

The type of the field.

`AddChildToLast<TField>(string, string, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)`

Adds the child data column at the end of the last data column.

```
public DataColumnCollection<T> AddChildToLast<TField>(string headerText, string fieldKey, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle> fieldStyleGenerator = null)
```

Parameters

`headerText string`

The header text.

`fieldKey string`

The field key.

headerStyle [CellStyle](#)?

The header style. The default is `SpreadsheetManager.DefaultCellStyles.HeaderStyle`.

fieldStyleGenerator [Func](#)<[FieldContext](#)<T, TField>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

Type Parameters

TField

The type of the field.

`Add<TField>(string, Expression<Func<T, TField>>, Action<GeneratorProvider<T, FieldContext<T, TField>>>, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)`

Adds the data column to the end of the `DataColumnCollection<T>`.

```
public DataColumnCollection<T> Add<TField>(string headerText, Expression<Func<T, TField>> fieldKeyExpression, Action<DataColumnCollection<T>.GeneratorProvider<T, FieldContext<T, TField>>> providerSetter, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

fieldKeyExpression [Expression](#)<[Func](#)<T, TField>>

The field key expression.

`providerSetter` `Action<DataColumnCollection<T>.GeneratorProvider<T, FieldContext<T, TField>>>`

The provider setter.

`headerStyle` `CellStyle?`

The header style. The default is `SpreadsheetManager.DefaultCellStyles.HeaderStyle`.

`fieldStyleGenerator` `Func<FieldContext<T, TField>, CellStyle>`

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

`DataColumnCollection<T>`

The self.

Type Parameters

`TField`

The type of the field.

`Add<TField>(string, Expression<Func<T, TField>>, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)`

Adds the data column to the end of the `DataColumnCollection<T>`.

```
public DataColumnCollection<T> Add<TField>(string headerText, Expression<Func<T, TField>>
    fieldKeyExpression, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle>
    fieldStyleGenerator = null)
```

Parameters

`headerText` `string`

The header text.

fieldKeyExpression [Expression](#)<[Func](#)<T, TField>>

The field key expression.

headerStyle [CellStyle](#)?

The header style. The default is `SpreadsheetManager.DefaultCellStyles.HeaderStyle`.

fieldStyleGenerator [Func](#)<[FieldContext](#)<T, TField>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

Type Parameters

TField

The type of the field.

Add<TField>(string, string, Action<GeneratorProvider<T, FieldContext<T, TField>>>, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)

Adds the data column to the end of the `DataColumnCollection<T>`.

```
public DataColumnCollection<T> Add<TField>(string headerText, string fieldKey,
Action<DataColumnCollection<T>.GeneratorProvider<T, FieldContext<T, TField>>>
providerSetter, CellStyle? headerStyle = null, Func<FieldContext<T, TField>, CellStyle>
fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

fieldKey [string](#)

The field key.

providerSetter [Action](#)< [DataColumnCollection](#)<T>.GeneratorProvider<T, [FieldContext](#)<T, TField>>>

The provider setter.

headerStyle [CellStyle](#)?

The header style. The default is [SpreadsheetManager.DefaultCellStyles.HeaderStyle](#).

fieldStyleGenerator [Func](#)<[FieldContext](#)<T, TField>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

Type Parameters

TField

The type of the field.

Add<TField>(string, string, CellStyle?, Func<FieldContext<T, TField>, CellStyle>)

Adds the data column to the end of the [DataColumnCollection](#)<T>.

```
public DataColumnCollection<T> Add<TField>(string headerText, string fieldKey, CellStyle?  
headerStyle = null, Func<FieldContext<T, TField>, CellStyle> fieldStyleGenerator = null)
```

Parameters

headerText [string](#)

The header text.

fieldKey [string](#)

The field key.

headerStyle [CellStyle](#)?

The header style. The default is [SpreadsheetManager.DefaultCellStyles.HeaderStyle](#).

fieldStyleGenerator [Func](#)<[FieldContext](#)<T, TField>, [CellStyle](#)>

The field style generator. The default is `(context) => SpreadsheetManager.DefaultCellStyles.FieldStyle`.

Returns

[DataColumnCollection](#)<T>

The self.

Type Parameters

TField

The type of the field.

ClearItems()

Removes all elements from the [Collection<T>](#).

```
protected override void ClearItems()
```

InsertItem(int, DataColumnBase<T>)

Inserts an element into the [Collection<T>](#) at the specified index.

```
protected override void InsertItem(int index, DataColumnBase<T> item)
```

Parameters

index [int](#)

The zero-based index at which `item` should be inserted.

`item` [DataColumnBase<T>](#)

The object to insert. The value can be [null](#) for reference types.

Exceptions

[ArgumentOutOfRangeException](#)

`<code class="paramref">index</code> is less than zero.`

-or-

`index` is greater than [Count](#).

RemoveItem(int)

Removes the element at the specified index of the [Collection<T>](#).

`protected override void RemoveItem(int index)`

Parameters

`index` [int](#)

The zero-based index of the element to remove.

Exceptions

[ArgumentOutOfRangeException](#)

`<code class="paramref">index</code> is less than zero.`

-or-

`index` is equal to or greater than [Count](#).

SetItem(int, DataColumnBase<T>)

Replaces the element at the specified index.

```
protected override void SetItem(int index, DataColumnBase<T> item)
```

Parameters

index [int](#)

The zero-based index of the element to replace.

item [DataColumnBase](#)<T>

The new value for the element at the specified index. The value can be [null](#) for reference types.

Exceptions

[ArgumentOutOfRangeException](#)

```
<code class="paramref">index</code> is less than zero.
```

-or-

index is greater than [Count](#).

Class

DataColumnCollection<T>.GeneratorProvider<TRecord, TContext>

Namespace: [CloudyWing.SpreadsheetExporter.Templates.RecordSet](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The generator provider.

```
public sealed class DataColumnCollection<T>.GeneratorProvider<TRecord, TContext> where  
TContext : RecordContext<TRecord>
```

Type Parameters

TRecord

The type of the record.

TContext

The type of the context.

Inheritance

[object](#) ← DataColumnCollection<T>.GeneratorProvider<TRecord, TContext>

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

UseDataValidation(Func<TContext, DataValidation>)

Uses the data validation.

```
public void UseDataValidation(Func<TContext, DataValidation> dataValidationGenerator)
```

Parameters

`dataValidationGenerator` [Func](#)<TContext, [DataValidation](#)>

The data validation generator.

UseFormula(Func<TContext, string>)

Uses the formula.

```
public void UseFormula(Func<TContext, string> generator)
```

Parameters

`generator` [Func](#)<TContext, [string](#)>

The generator.

UseValue(Func<TContext, object>)

Uses the value.

```
public void UseValue(Func<TContext, object> generator)
```

Parameters

`generator` [Func](#)<TContext, [object](#)>

The generator.

See Also

[Collection](#)<T>

Class FieldContext<TRecord, TField>

Namespace: [CloudyWing.SpreadsheetExporter.Templates.RecordSet](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The field context.

```
public class FieldContext<TRecord, TField> : RecordContext<TRecord>
```

Type Parameters

TRecord

The type of the record.

TField

The type of the field.

Inheritance

```
object ↗ ← RecordContext<TRecord> ← FieldContext<TRecord, TField>
```

Inherited Members

```
RecordContext<TRecord>.CellIndex , RecordContext<TRecord>.RowIndex ,  
RecordContext<TRecord>.Record , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.GetHashCode() ↗ , object.GetType() ↗ , object.MemberwiseClone() ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.ToString() ↗
```

Constructors

FieldContext(RecordContext<TRecord>, string, TField)

The field context.

```
public FieldContext(RecordContext<TRecord> recordContext, string key, TField value)
```

Parameters

recordContext [RecordContext<TRecord>](#)

The record context.

key [string](#)

The key.

value [TField](#)

The value.

Exceptions

[ArgumentNullException](#)

key

Properties

Key

Gets the key.

```
public string Key { get; }
```

Property Value

[string](#)

The key.

Exceptions

[ArgumentNullException](#)

key

Value

Gets the value.

```
public TField Value { get; }
```

Property Value

TField

The value.

Exceptions

[ArgumentNullException](#)

key

Class RecordContext<T>

Namespace: [CloudyWing.SpreadsheetExporter.Templates.RecordSet](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The record context.

```
public class RecordContext<T>
```

Type Parameters

T

The type of the record.

Inheritance

[object](#) ← RecordContext<T>

Derived

[FieldContext<TRecord, TField>](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

RecordContext(int, int, T)

The record context.

```
public RecordContext(int cellIndex, int rowIndex, T record)
```

Parameters

cellIndex [int](#)

Index of the cell.

RowIndex [int↗](#)

Index of the row.

record T

The record.

Properties

CellIndex

Gets the index of the cell. The index start at 0.

```
public int CellIndex { get; }
```

Property Value

[int↗](#)

The index of the cell.

Record

Gets the record.

```
public T Record { get; }
```

Property Value

T

The record.

RowIndex

Gets the index of the row. The index start at 0.

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

Property Value

[int ↗](#)

The index of the row.

Class RecordSetTemplate<T>

Namespace: [CloudyWing.SpreadsheetExporter.Templates.RecordSet](#)

Assembly: CloudyWing.SpreadsheetExporter.dll

The recordset template. Create cell information using set data source and data column.

```
public class RecordSetTemplate<T> : ITemplate
```

Type Parameters

T

The type of the record.

Inheritance

[object](#) ← RecordSetTemplate<T>

Implements

[ITemplate](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

RecordSetTemplate(IEnumerable<T>)

The recordset template. Create cell information using set data source and data column.

```
public RecordSetTemplate(IEnumerable<T> dataSource)
```

Parameters

dataSource [IEnumerable](#)<T>

The data source.

Exceptions

[ArgumentNullException](#)

dataSource

See Also

[ITemplate](#)

Properties

ColumnSpan

Gets the column span.

```
public int ColumnSpan { get; }
```

Property Value

[int](#)

The column span.

Exceptions

[ArgumentNullException](#)

dataSource

Columns

Gets the columns.

```
public DataColumnCollection<T> Columns { get; }
```

Property Value

[DataColumnCollection](#)<T>

The columns.

Exceptions

[ArgumentNullException](#)

dataSource

DataSource

Gets or sets the data source.

```
public IEnumerable<T> DataSource { get; set; }
```

Property Value

[IEnumerable](#)<T>

The data source.

Exceptions

[ArgumentNullException](#)

dataSource

HeaderHeight

Gets or sets the height of the header.

```
public double? HeaderHeight { get; set; }
```

Property Value

[double](#)?

The height of the header.

Exceptions

[ArgumentNullException](#)

dataSource

IsAutoFilterEnabled

Gets or sets a value indicating whether to enable auto filter for the data range. Auto filter is applied to the header row and data rows.

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

Property Value

[bool](#)

`true` to enable auto filter; otherwise, `false`. Default is `false`.

Exceptions

[ArgumentNullException](#)

dataSource

IsFrozenHeader

Gets or sets a value indicating whether to freeze header rows. When enabled, freezes all header rows (determined by [RowSpan](#)).

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

Property Value

[bool](#)

`true` to freeze header rows; otherwise, `false`. Default is `false`.

Exceptions

[ArgumentNullException](#)

dataSource

RecordHeight

Gets or sets the height of the record.

```
public double? RecordHeight { get; set; }
```

Property Value

[double](#)?

The height of the record.

Exceptions

[ArgumentNullException](#)

dataSource

RowSpan

Gets the row span.

```
public int RowSpan { get; }
```

Property Value

[int](#)

The row span.

Exceptions

[ArgumentNullException](#)

dataSource

Methods

[GetContext\(\)](#)

Gets the context.

```
public TemplateContext GetContext()
```

Returns

[TemplateContext](#)

The template context.

Exceptions

[ArgumentNullException](#)

dataSource

See Also

[ITemplate](#)