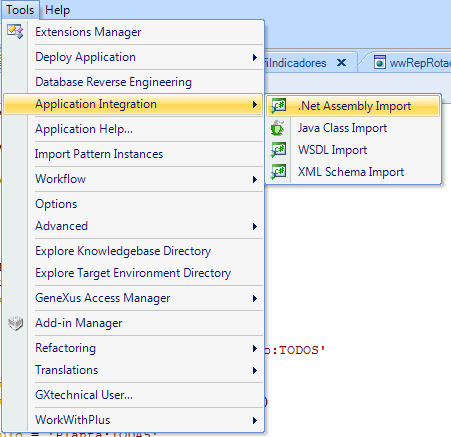
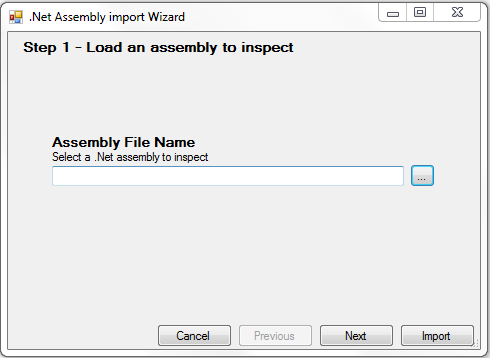
Documentacion EPPlusMTK

Descripción.- Clase de C# para crear archivo de Excel utilizando las funciones mas requeridas en la creación de un documento xls.

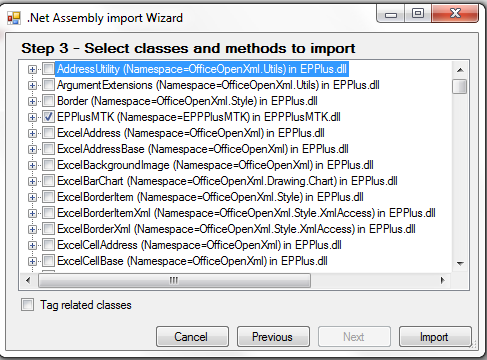
Importar la clase en “tolos/Application Integration/.Net Assembly Import”



Seleccionar el archivo EPPPlusMTK.dll



Seleccionar la Clase EPPlusMTK



Los Métodos son los siguientes:

Open

Descripción:

Método para definir el la ruta y el nombre con el que se creará el archivo.

Parámetros:

Directory.-Direccion donde se creará el archivo

Name.- Nombre del Archivo

Extension.- Extencion del archivo (“xls”,”xlsx”)

public void Open(string Directory, string Name, string Extension)

{

string File = Directory + '/' + Name + "." + Extension;

package = new ExcelPackage(new FileInfo(File));

}

CreateSheet

Descripción:

Método para crear hojas en archivo Excel (Siempre se tiene que crear para iniciar a escribir en el documento), este se llama cada que se quiera llamar una hoja nueva y empezar a escribir en ella.

Parámetros:

NameSheet.- Nombre de la hoja

public void CreateSheet(string NameSheet)

{

worksheet = package.Workbook.Worksheets.Add(NameSheet);

}

CellsHorizontalSize

Descripción:

Se define un ancho para la columna indicada.

Parámetros:

ColumnStart.-Numero de la columna a modificar lo ancho.

Size.- Medida de la Columna

public void CellsHorizontalSize( Int32 ColumnStart, Int32 Size)

{

worksheet.Column(ColumnStart).Width = Size;

}

CellsVerticalSize

Descripción:

Se define un alto para la fila indicada.

Parámetros:

RowStart.- Numero de la fila a modificar el alto.

Size.- Medida de la Columna

public void CellsVerticalSize(Int32 RowStart, Int32 Size)

{

worksheet.Row(RowStart).Height = Size;

}

Cells

Descripción:

Ingresar texto en la celda indicada. El sistema llama al método indicado según el numero de parámetros ingresados (En genexus le agrega un numero consecutivo al nombre del método)

Parámetros:

RowStart.-Numero de fila donde se imprimirá el texto.

ColumnStart.- Numero de columna donde se imprimirá el texto.

Text.- Texto a ingresar en la celda

public void Cells(Int32 RowStart, Int32 ColumnStart, string Text)

{

worksheet.Cells[RowStart, ColumnStart].Value = Text;

}

Parámetros:

RowStart.-Numero de fila donde se Inicia la impresión del texto.

ColumnStart.- Numero de Columna donde se Inicia la impresión del texto.

RowLast.- Numero de fila donde Termina la impresión del texto.

ColumnLast.- Numero de columna donde Termina la impresión del texto.

Text.- Texto a ingresar en la celda

public void Cells(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast, string Text)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Value = Text;

}

CellsNumber

Descripción:

Ingresar un numero en la celda indicada en formato numérico. El sistema llama al método indicado según el numero de parámetros ingresados (En genexus le agrega un numero consecutivo al nombre del método)

Parámetros:

RowStart.-Numero de fila donde se imprimirá el Número.

ColumnStart.- Numero de columna donde se imprimirá el Número.

Numero.- Numero a ingresar en la celda

public void CellsNumber(Int32 RowStart, Int32 ColumnStart, int Numero)

{

worksheet.Cells[RowStart, ColumnStart].Value = Numero;

}

Parámetros:

RowStart.-Numero de fila donde se Inicia la impresión del Número.

ColumnStart.- Numero de columna donde se Inicia la impresión del Número.

RowLast.- Numero de fila donde Termina la impresión el Número.

ColumnLast.- Numero de columna donde Termina la impresión el Número.

Numero.- Numero a ingresar en la celda

public void CellsNumber(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast, int Numero)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Value = Numero;

}

CellsHorizontalCenter

Descripción:

Centra el texto de forma horizontal

Parámetros:

RowStart.-Numero de fila donde centrará el contenido de la celda.

ColumnStart.- Numero de columna donde centrará el contenido de la celda.

public void CellsHorizontalCenter(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Center;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a centrar.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a centrar.

RowLast.- Numero de fila donde Termina el rango de celdas a centrar.

ColumnLast.- Numero de columna donde Termina el rango de celdas a centrar.

public void CellsHorizontalCenter(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Center;

}

CellsHorizontalLeft

Descripción:

Alinear el texto a la izquierda de forma horizontal

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsHorizontalLeft(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Left;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsHorizontalLeft(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Left;

}

CellsHorizontalRight

Descripción:

Alinear el texto a la Derecha de forma horizontal

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsHorizontalRight(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Right;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsHorizontalRight(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Right;

}

CellsHorizontalJustify

Descripción:

Justificar el texto

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsHorizontalJustify(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Justify;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsHorizontalJustify(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.HorizontalAlignment = OfficeOpenXml.Style.ExcelHorizontalAlignment.Justify;

}

CellsVerticalTop

Descripción:

Alinear el texto al tope de forma vertical

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsVerticalTop(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Top;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsVerticalTop(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Top;

}

CellsVerticalCenter

Descripción:

Alinear el texto al centro de forma vertical

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsVerticalCenter(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Center;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsVerticalCenter(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Center;

}

CellsVerticalBottom

Descripción:

Alinear el texto a la base de la celda.

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsVerticalBottom(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Bottom;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsVerticalBottom(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Bottom;

}

CellsVerticalJustify

Descripción:

Se justifica el texto de forma vertical.

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

public void CellsVerticalJustify(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Justify;

}

RowStart.-Numero de fila donde se Inicia el rango de celdas a alinear.

ColumnStart.- Numero de columna donde se Inicia el rango de celdas a alinear.

RowLast.- Numero de fila donde Termina el rango de celdas a alinear.

ColumnLast.- Numero de columna donde Termina el rango de celdas a alinear.

public void CellsVerticalJustify(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.VerticalAlignment = OfficeOpenXml.Style.ExcelVerticalAlignment.Justify;

}

CellsFormula

Descripción:

Se agregan formulas para interpretar en las celdas de Excel. (El símbolo de = al principio de la formula te lo agrega por default) Ejemplo: “SUMA(A1:A5)”

Parámetros:

RowStart.-Numero de fila a alinear el contenido de la celda.

ColumnStart.- Numero de columna a alinear el contenido de la celda.

Formula.- Texto con la formula a imprimir.

public void CellsFormula(Int32 RowStart, Int32 ColumnStart, string Formula)

{

worksheet.Cells[RowStart, ColumnStart].Formula = Formula;

}

Descripcion:

Puedes realizar distintas operaciones enviando la celda donde se imprimirá, el rango del celdas que tomará en cuenta y el operador a utilizar. Ejemplo: “B1=(A1+A5)”

Parámetros:

RowStart.- Fila donde se imprime el resultado.

ColumnStart.- Celda donde se imprime el resultado.

RowFirstCell.- Fila del primer dato a tomar en cuenta.

ColumnFirstCell.- Columna del primer dato a tomar en cuenta.

RowSecundCell.- Fila del segundo dato a tomar en cuenta.

ColumnSecundCell.- Columna del segundo dato a tomar en cuenta.

Operation.- Operador para utilizar en la formula (+,-,\*,/)

public void CellsFormula(Int32 RowStart, Int32 ColumnStart, Int32 RowFirstCell, Int32 ColumnFirstCell, Int32 RowSecundCell, Int32 ColumnSecundCell, string Operation)

{

worksheet.Cells[RowStart, ColumnStart].Formula = "(" + worksheet.Cells[RowFirstCell, ColumnFirstCell].Address + Operation + worksheet.Cells[RowSecundCell, ColumnSecundCell].Address + ")";

}

Descripcion:

Puedes realizar distintas formulas enviando la celda donde se imprimirá, el tipo de formula que se utilizara y el rango del celdas que tomará. Ejemplo: “SUM=(A1:A5)”

Parámetros:

RowStart.- Fila donde se imprime el resultado.

ColumnStart.- Celda donde se imprime el resultado.

Formula.- Formula que se utilizará (SUM,REST,PROM, ETC)

RowStartRead.- Fila del primer dato a tomar en cuenta.

ColumnStartRead.- Columna del primer dato a tomar en cuenta.

RowlastRead.- Fila del segundo dato a tomar en cuenta.

ColumnLastRead.- Columna del segundo dato a tomar en cuenta.

public void CellsFormula(Int32 RowStart, Int32 ColumnStart, string Formula, Int32 RowStartRead, Int32 ColumnStartRead, Int32 RowLastRead, Int32 ColumnLastRead)

{

worksheet.Cells[RowStart, ColumnStart].Formula = Formula + "(" + worksheet.Cells[RowStartRead, ColumnStartRead].Address + ":" + worksheet.Cells[RowLastRead, ColumnLastRead].Address+")";

}

Descripcion:

Puedes igualar dos Celdas. Ejemplo: “A1=A5”

Parámetros:

RowStart.- Fila donde se imprime el resultado.

ColumnStart.- Celda donde se imprime el resultado.

RowRead.- Fila Que se igualará

ColumnRead.- Columna que se igualará

public void CellsFormula(Int32 RowStart, Int32 ColumnStart, Int32 RowRead, Int32 ColumnRead)

{

worksheet.Cells[RowStart, ColumnStart].Formula = worksheet.Cells[RowRead, ColumnRead].Address; }

CellsBord

Descripción:

Se dibuja el borde de una celda o rango de celdas.

Parámetros:

RowStart.- Fila donde se dibujará el contorno de la celda.

ColumnStart.- Columna donde se dibujará el contorno de la celda

public void CellsBord(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.Border.Left.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

worksheet.Cells[RowStart, ColumnStart].Style.Border.Right.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

worksheet.Cells[RowStart, ColumnStart].Style.Border.Top.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

worksheet.Cells[RowStart, ColumnStart].Style.Border.Bottom.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

}

Parámetros:

RowStart.- Fila donde se iniciará a dibujar el rango del contorno de la celda.

ColumnStart.- Columna donde se iniciará a dibujar el rango del contorno de la celda.

RowLast.- Fila donde terminará de dibujar el contorno de la celda.

ColumnLast.- Columna donde terminará de dibujar el contorno de la celda.

public void CellsBord(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Border.Left.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Border.Right.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Border.Top.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Border.Bottom.Style = OfficeOpenXml.Style.ExcelBorderStyle.Thin;

}

CellsColor

Descripción:

Color de fondo de la celda o rango de celdas.

Parámetros:

RowStart.- Fila donde se Coloreará la celda.

ColumnStart.- Columna donde se Coloreará la celda.

Color.- Color en RGB que se agregará a la celda “#FFFFFF”

public void CellsColor(Int32 RowStart, Int32 ColumnStart, string Color)

{

worksheet.Cells[RowStart, ColumnStart].Style.Fill.PatternType = OfficeOpenXml.Style.ExcelFillStyle.Solid;

worksheet.Cells[RowStart, ColumnStart].Style.Fill.BackgroundColor.SetColor(System.Drawing.ColorTranslator.FromHtml(Color));

}

Parámetros:

RowStart.- Fila donde se Iniciará el rango a colorear.

ColumnStart.- Columna donde se Iniciará el rango a colorear.

RowLast.- Fila donde finalizará el rango a colorear.

ColumnLast.- Columna donde finalizará el rango a colorear.

Color.- Color en RGB que se agregará a la celda “#FFFFFF”

public void CellsColor(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast, string Color)

{

worksheet.Cells[RowStart, ColumnStart,RowLast,ColumnLast].Style.Fill.PatternType = OfficeOpenXml.Style.ExcelFillStyle.Solid;

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Fill.BackgroundColor.SetColor(System.Drawing.ColorTranslator.FromHtml(Color));

}

CellsBold

Descripción:

Letra en negritas en la o las celdas seleccionadas.

Parámetros:

RowStart.- Fila donde se Iniciará las letras en negritas.

ColumnStart.- Columna donde se Iniciará las letras en negritas.

public void CellsBold(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.Font.Bold = true;

}

Parámetros:

RowStart.- Fila donde se Iniciará el rango de las letras en negritas.

ColumnStart.- Columna donde se Iniciará el rango de las letras en negritas.

RowLast.- Fila donde finalizará el rango de las letras en negritas.

ColumnLast.- Columna donde finalizará el rango de las letras en negritas.

public void CellsBold(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Font.Bold = true;

}

CellsItalic

Descripción:

Letra Itálica en la o las celdas seleccionadas.

Parámetros:

RowStart.- Fila donde se Iniciará las letras Itálica.

ColumnStart.- Columna donde se Iniciará las letras Itálica.

public void CellsItalic(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.Font.Italic = true;

}

Parámetros:

RowStart.- Fila donde se Iniciará el rango de las letras Itálica.

ColumnStart.- Columna donde se Iniciará el rango de las letras Itálica.

RowLast.- Fila donde finalizará el rango de las letras Itálica.

ColumnLast.- Columna donde finalizará el rango de las letras Itálica.

public void CellsItalic(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Font.Italic = true;

}

CellsUnderLine

Descripción:

Subrayado en la o las celdas seleccionadas.

Parámetros:

RowStart.- Fila donde se realizará el Subrayado.

ColumnStart.- Columna donde se realizará el Subrayado.

public void CellsUnderLine(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.Font.UnderLine = true;

}

Parámetros:

RowStart.- Fila donde se Iniciará el rango del Subrayado.

ColumnStart.- Columna donde se Iniciará el rango del Subrayado.

RowLast.- Fila donde finalizará el rango del Subrayado.

ColumnLast.- Columna donde finalizará el rango del Subrayado.

public void CellsUnderLine(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Font.UnderLine = true;

}

CellsFontColor

Descripción:

Agregar color a la fuente de la celda o rango de celdas.

Parámetros:

RowStart.- Fila donde se realizará el coloreado de la fuente.

ColumnStart.- Columna donde se realizará el coloreado de la fuente.

Color.- Color en RGB que se agregará a la celda “#FFFFFF”

public void CellsFontColor(Int32 RowStart, Int32 ColumnStart, string Color)

{

worksheet.Cells[RowStart, ColumnStart].Style.Font.Color.SetColor(System.Drawing.ColorTranslator.FromHtml(Color));

}

Parámetros:

RowStart.- Fila donde se Iniciará el rango del coloreado de la fuente.

ColumnStart.- Columna donde se Iniciará el rango del coloreado de la fuente.

RowLast.- Fila donde finalizará el rango del coloreado de la fuente.

ColumnLast.- Columna donde finalizará el rango del coloreado de la fuente.

Color.- Color en RGB que se agregará a la celda “#FFFFFF”

public void CellsColor(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast, string Color)

{

worksheet.Cells[RowStart, ColumnStart,RowLast,ColumnLast].Style.Font.Color.SetColor(System.Drawing.ColorTranslator.FromHtml(Color));

}

CellsRotation

Descripción:

Rotar el texto en 360 grados.

Parámetros:

RowStart.- Fila donde se realizará la rotación del texto.

ColumnStart.- Columna donde se realizará la rotación del texto.

Rotacion.- Grados en los que realizará la rotación del texto.

public void CellsRotation(Int32 RowStart, Int32 ColumnStart, Int32 Rotacion)

{

worksheet.Cells[RowStart, ColumnStart].Style.TextRotation = Rotacion;

}

CellsFontSize

Descripción:

Se puede cambiar el tamaño de la fuente en una celda o en un rango de celdas.

Parámetros:

RowStart.- Fila donde se aplicará el cambio de tamaño de la fuente.

ColumnStart.- Columna donde se aplicará el cambio de tamaño de la fuente.

Size.- Tamaño de la fuente.

public void CellsFontSize(Int32 RowStart, Int32 ColumnStart, int Size)

{

worksheet.Cells[RowStart, ColumnStart].Style.Font.Size = Size;

}

Parámetros:

RowStart.- Fila donde se Iniciará el rango donde se aplicará el cambio de tamaño de la fuente.

ColumnStart.- Columna donde se Iniciará el rango donde se aplicará el cambio de tamaño de la fuente.

RowLast.- Fila donde finalizará el rango donde se aplicará el cambio de tamaño de la fuente.

ColumnLast.- Columna donde finalizará el rango donde se aplicará el cambio de tamaño de la fuente.

Size.- Tamaño de la fuente.

public void CellsFontSize(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast, int Size)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Font.Size = Size;

}

CellsCombine

Descripción:

Combinación de varios celdas.

Parámetros:

RowStart.- Fila donde se Iniciará el rango de combinación de celdas.

ColumnStart.- Columna donde se Iniciará el rango de combinación de celdas.

RowLast.- Fila donde finalizará el rango de combinación de celdas.

ColumnLast.- Columna donde finalizará el rango de combinación de celdas.

public void CellsCombine(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Merge = true;

}

CellsWrapText

Descripción:

Ajustar texto a la celda.

Parámetros:

RowStart.- Fila donde se realizará el ajuste del texto a la celda.

ColumnStart.- Columna donde se realizará el ajuste del texto a la celda.

public void CellsWrapText(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.WrapText = true;

}

CellsFormatNum

Descripción:

Cambiar de formato de la celda o rango de celdas a numérico.

Parámetros:

RowStart.- Fila donde se formateará la celda a numérico.

ColumnStart.- Columna donde se formateará la celda a numérico.

public void CellsFormatNum(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.Numberformat.Format = "0.00";

}

Parámetros:

RowStart.- Fila donde Iniciará el rango donde se formateará la celda a numérico.

ColumnStart.- Columna donde Iniciará el rango donde se formateará la celda a numérico.

RowLast.- Fila donde finalizará el rango donde se formateará la celda a numérico.

ColumnLast.- Columna donde finalizará el rango donde se formateará la celda a numérico.

public void CellsFormatNum(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart,RowLast,ColumnLast].Style.Numberformat.Format = "0.00";

}

CellsFormatDate

Descripción:

Cambiar de formato de la celda o rango de celdas a Fecha.

Parámetros:

RowStart.- Fila donde se formateará la celda a Fecha.

ColumnStart.- Columna donde se formateará la celda a Fecha.

public void CellsFormatDate(Int32 RowStart, Int32 ColumnStart)

{

worksheet.Cells[RowStart, ColumnStart].Style.Numberformat.Format = "mm-dd-yy"; }

RowStart.- Fila donde Iniciará el rango donde se formateará la celda a Fecha.

ColumnStart.- Columna donde Iniciará el rango donde se formateará la celda a Fecha.

RowLast.- Fila donde finalizará el rango donde se formateará la celda a Fecha.

ColumnLast.- Columna donde finalizará el rango donde se formateará la celda a Fecha.

public void CellsFormatDate(Int32 RowStart, Int32 ColumnStart, Int32 RowLast, Int32 ColumnLast)

{

worksheet.Cells[RowStart, ColumnStart, RowLast, ColumnLast].Style.Numberformat.Format = "mm-dd-yy";

}

Image

Descripción:

Agregar imagen al archivo de Excel.

Parámetros:

RowStart.- Fila donde se inserta la imagen.

ColumnStart.- Columna donde se inserta la imagen.

imagePath.- Direccion donde se encuentra la imagen.

WidthSize.- Ancho que tendrá la imagen en el archivo de Excel.

HeighSize.- Alto que tendrá la imagen en el archivo de Excel.

public void Image(Int32 RowStart, Int32 ColumnStart, string imagePath, Int32 WidthSize, Int32 HeightSize)

{

Bitmap image = new Bitmap(imagePath);

ExcelPicture excelImage = null;

if (image != null)

{

excelImage = worksheet.Drawings.AddPicture("Imagen", image);

excelImage.From.Column = ColumnStart;

excelImage.From.Row = RowStart;

excelImage.SetSize(WidthSize, HeightSize);

// 2x2 px space for better alignment

excelImage.From.ColumnOff = Pixel2MTU(2);

excelImage.From.RowOff = Pixel2MTU(2);

}

}

public int Pixel2MTU(int pixels)

{

int mtus = pixels \* 9525;

return mtus;

}

Save

Descripción:

Al terminar de generar el archivo se guarda y se crea.

public void Save()

{ package.Save(); }