GemBox Support Center

Portal > Knowledgebase > GemBox.Spreadsheet > How to unmerge cells

How to unmerge cells

Mario - GemBox - 2017-02-09 - 0 Comments - in GemBox.Spreadsheet

Merging cells is achieved by setting a CellRange's Merged property to true.

Unmerging cells through CellRange's MergedRange property is not allowed (an exception will be thrown).
To unmerge cells we need to create a range from the ExcelWorksheet and then unmerge it. This kind of implementation was chosen for performance reasons.

Here is the sample code which demonstrates how to properly unmerge cells:

C# code

```
static void Main(string[] args)
{
   // If using Professional version, put your serial key below.
   SpreadsheetInfo.SetLicense("FREE-LIMITED-KEY");

   ExcelFile ef = new ExcelFile();
   ExcelWorksheet ws = ef.Worksheets.Add("Sheet1");

   // Merge some cells
   ws.Cells.GetSubrangeAbsolute(0, 0, 2, 2).Merged = true;

   // Set some value (value will be set to merged cell range - not cell alone)
   ws.Cells[0, 0].Value = 1;

   // UnmergeCells_BadWay(ws);

   UnmergeCells_GoodWay(ws);

   ef.Save("Merge and unmerge cells.xlsx");
}

private static void UnmergeCells_GoodWay(ExcelWorksheet ws)
{
   // Get some cell in the range
   ExcelCell cell = ws.Cells[0, 0];

   // Get merged range in which that cell is.
   CellRange mergedRange = cell.MergedRange;

   // You need to get new range that is made from the worksheet (it's a bit different than
previous - which was made from cell).
   CellRange wsMergedRange =
ws.Cells.GetSubrangeAbsolute(mergedRange.FirstRowIndex,
mergedRange.FirstColumnIndex, mergedRange.LastRowIndex,
mergedRange.LastColumnIndex);

   // Unmerge cells (note that all cells would now have the value that was set to merged
range).
```

```
    wsMergedRange.Merged = false;

    // We want that only cell in the upper-left corner has merged value -> so delete values from other cells.
    for (int i = mergedRange.FirstRowIndex; i <= mergedRange.LastRowIndex; i++)
        for (int j = mergedRange.FirstColumnIndex; j <= mergedRange.LastColumnIndex; j++)
            if (i != mergedRange.FirstRowIndex || j != mergedRange.FirstColumnIndex)
                ws.Cells[i, j].Value = null;
}


private static void UnmergeCells_BadWay(ExcelWorksheet ws)
{
    // Get some cell in the range
    ExcelCell cell = ws.Cells[0, 0];

    // Get merged range in which that cell is.
    CellRange mergedRange = cell.MergedRange;

    // The following line will throw an exception!
    mergedRange.Merged = false;
}
```

VB.NET code

```
Sub Main(args As String())
    ' If using Professional version, put your serial key below.
    SpreadsheetInfo.SetLicense("FREE-LIMITED-KEY")

    Dim ef As New ExcelFile()
    Dim ws As ExcelWorksheet = ef.Worksheets.Add("Sheet1")

    ' Merge some cells
    ws.Cells.GetSubrangeAbsolute(0, 0, 2, 2).Merged = True

    ' Set some value (value will be set to merged cell range - not cell alone)
    ws.Cells(0, 0).Value = 1

    ' UnmergeCells_BadWay(ws)

    UnmergeCells_GoodWay(ws)

    ef.Save("Merge and unmerge cells.xlsx")
End Sub

Private Sub UnmergeCells_GoodWay(ws As ExcelWorksheet)
    ' Get some cell in the range
    Dim cell As ExcelCell = ws.Cells(0, 0)

    ' Get merged range in which that cell is.
    Dim mergedRange As CellRange = cell.MergedRange

    ' You need to get new range that is made from the worksheet (it's a bit different than previous - which was made from cell).
    Dim wsMergedRange As CellRange = ws.Cells.GetSubrangeAbsolute(mergedRange.FirstRowIndex, mergedRange.FirstColumnIndex, mergedRange.LastRowIndex, mergedRange.LastColumnIndex)
    ' Unmerge cells (note that all cells would now have the value that was set to merged range).
    wsMergedRange.Merged = False

    ' We want that only cell in the upper-left corner has merged value -> so delete values from other cells.
    For i As Integer = mergedRange.FirstRowIndex To mergedRange.LastRowIndex
        For j As Integer = mergedRange.FirstColumnIndex To mergedRange.LastColumnIndex
            If i <> mergedRange.FirstRowIndex OrElse j <> mergedRange.FirstColumnIndex Then
                ws.Cells(i, j).Value = Nothing
            End If
        Next
    Next
End Sub

Private Sub UnmergeCells_BadWay(ws As ExcelWorksheet)
    ' Get some cell in the range
    Dim cell As ExcelCell = ws.Cells(0, 0)

    ' Get merged range in which that cell is.
    Dim mergedRange As CellRange = cell.MergedRange

    ' The following line will throw an exception!
    mergedRange.Merged = False
End Sub
```