

GemBox Support Center

Portal > Knowledgebase > GemBox.Spreadsheet > Read a CSV file with more than 1,048,576 rows

Read a CSV file with more than 1,048,576 rows

Mario - GemBox - 2017-05-08 - 0 Comments - in GemBox.Spreadsheet

GemBox.Spreadsheet's content model has the same limitations as the XLSX file format (1,048,576 rows and 16,384 columns per sheet), the file format closest to the GemBox.Spreadsheet model and Microsoft Excel.

Because of these limitations, we cannot load a CSV file with more than 1,048,576 rows into a single [ExcelWorksheet](#). However, a workaround exists. We can load a CSV file into multiple [ExcelWorksheets](#), so that when the limitation of 1,048,576 rows is passed, the following rows are loaded into separate [ExcelWorksheet](#).

For this workaround we can use the [ExcelFile.Load\(TextReader, CsvLoadOptions\)](#) method and provide a custom [TextReader](#) to read the CSV sheets.

The code for the custom [TextReader](#) ([CsvTextReader](#)) and an example of its usage are provided below.

C# code

```
public sealed class CsvFileReader : TextReader
{
    private const int MaxRow = 1048576;
    private readonly TextReader reader;

    private int currentRow;
    private bool isCompleted;

    public static ExcelFile Read(string path, CsvLoadOptions options)
    {
        var workbook = new ExcelFile();
        int sheetIndex = 0;

        using (var reader = new CsvFileReader(path))
            while (reader.CanReadNextSheet())
            {
                ++sheetIndex;
                workbook.Worksheets.AddCopy(
                    string.Format("Sheet {0}", sheetIndex),
                    ExcelFile.Load(reader, options).Worksheets.Active
```

```

Worksheet);
    }

    return workbook;
}

private CsvFileReader(string path)
{
    this.reader = File.OpenText(path);
}

private bool CanReadNextSheet()
{
    if (this.isCompleted)
        return false;

    this.currentRow = 0;
    return true;
}

public override string ReadLine()
{
    if (this.currentRow < MaxRow)
    {
        ++this.currentRow;
        string line = this.reader.ReadLine();

        if (line == null)
            this.isCompleted = true;

        return line;
    }

    return null;
}

protected override void Dispose(bool disposing)
{
    this.reader.Dispose();
}
}

```

```

class Sample
{
    static void Main(string[] args)
    {
        SpreadsheetInfo.SetLicense("FREE-LIMITED-KEY");
        SpreadsheetInfo.FreeLimitReached += (sender, e) => e.FreeLimitReachedAction = FreeLimitReachedAction.ContinueAsTrial;
        ExcelFile ef = CsvFileReader.Read("Sample.csv", LoadOptions.CsvDefault);
        foreach (ExcelWorksheet ws in ef.Worksheets)
        {
            Console.WriteLine("Sheet's name: {0} | Sheet's rows count : {1}", ws.Name, ws.Rows.Count);
        }
    }
}

```

VB.NET code

```

Public NotInheritable Class CsvFileReader
    Inherits TextReader
    Private Const MaxRow As Integer = 1048576
    Private ReadOnly reader As TextReader

    Private currentRow As Integer
    Private isCompleted As Boolean

    Public Shared Function Read(path As String, options As CsvLoadOptions) As ExcelFile
        Dim workbook = New ExcelFile()
        Dim sheetIndex As Integer = 0

        Using reader = New CsvFileReader(path)
            While reader.CanReadNextSheet()
                sheetIndex += 1
                workbook.Worksheets.AddCopy(
                    String.Format("Sheet {0}", sheetIndex),
                    ExcelFile.Load(reader, options).Worksheets.ActiveWorksheet)
            End While
        End Using
    End Function

```

```

        Return workbook
    End Function

    Private Sub New(path As String)
        Me.reader = File.OpenText(path)
    End Sub

    Private Function CanReadNextSheet() As Boolean
        If Me.isCompleted Then
            Return False
        End If

        Me.currentRow = 0
        Return True
    End Function

    Public Overrides Function ReadLine() As String
        If Me.currentRow < MaxRow Then
            Me.currentRow += 1
            Dim line As String = Me.reader.ReadLine()

            If line Is Nothing Then
                Me.isCompleted = True
            End If

            Return line
        End If

        Return Nothing
    End Function

    Protected Overrides Sub Dispose(disposing As Boolean)
        Me.reader.Dispose()
    End Sub
End Class

Module Sample
    Sub Main(args As String())
        SpreadsheetInfo.SetLicense("FREE-LIMITED-KEY")
        AddHandler SpreadsheetInfo.FreeLimitReached, Sub(sender, e)
e.FreeLimitReachedAction = FreeLimitReachedAction.ContinueAsTrial
    End Sub
End Module

```

```
    Dim ef As ExcelFile = CsvFileReader.Read("Sample.csv", LoadOptions.CsvDefault)
    For Each ws As ExcelWorksheet In ef.Worksheets
        Console.WriteLine("Sheet's name: {0} | Sheet's rows count : {1}", ws.Name, ws.Rows.Count)
    Next
End Sub
End Module
```

Attachments

- [CsvFileReader.cs \[1.44 KB\]](#)
- [CsvFileReader.vb \[1.58 KB\]](#)