# Uni-Excel

My Unity Dream Created Uni-Excel Component For Create - Edit And Read Excel File. In Uni-Excel Use Multi Component Such As NPOI - ExcelDataReader And Arabic Support

For Creating And Editing Excel File We Use NPOI Component.

For Reading Excel File We Use NPOI And ExcelDataReader Component.

**Support:**

Excel 2003

Windows

Arabic And Persian

**Sample Code:**

```csharp
using UnityEngine;
using System.Collections;

using System.Collections.Generic;          → Library For List Of Excel Data

using NPOI.HSSF.UserModel;                 → Importing NPOI Library
using NPOI.SS.UserModel;
using NPOI.HSSF.Util;

using System.IO;                            → For Address Of Excel File

using Excel;                                → Importing ExcelDataReader Library
using System.Data;

using ArabicSupport;                        → For Arabic Support

public class MyExcel : MonoBehaviour
{
    private string MySheetName = "Sheet_Test";          → Sheet Name Of Your Excel File

    public List<string> MyCellArray;                     → Data List For The First And Second Column
    public List<string> MyCellArray02;

    public List<string> MyString;                        → String Of Data For ExcelDataReader Only

    private bool ReadExcelEnable_NPOI = false;           → Enable And Disable NPOI And ExcelDataReader Component
    private bool ReadExcelEnable_ExcelDataReader = false;

    void OnGUI()
    {
        GUIStyle style = new GUIStyle();

        GUI.Label(new Rect(10, 10, 100, 30), "Sheet Name:");          → Text Field For Change Sheet Name
        MySheetName = GUI.TextField(new Rect(90, 10, 200, 30), MySheetName, 25);

        style.richText = true;                                         → Address Of Excel File

        FileStream MyAddress = new FileStream(Application.dataPath + "/My First Excel.xls", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.ReadWrite);

        if (GUI.Button(new Rect(300, 10, 200, 30), "Create Excel Files With " + "<color=red>NPOI</color>"))
        {
            HSSFWorkbook MyWorkbook = new HSSFWorkbook();              → Create An Empty Excel File

            HSSFSheet Sheet01 = (HSSFSheet)MyWorkbook.CreateSheet(MySheetName);   → Adding The First Sheet

            for (int i = 0; i < 5; i++)
            {
                HSSFRow Row = (HSSFRow)Sheet01.CreateRow((short)i);       → Define Rows For Your Sheet

                HSSFCell cell = (HSSFCell)Row.CreateCell((short)0);       → Define Columns For Rows

                cell.SetCellValue(MyCellArray[i]);                        → Adding Value To Cells

                if (i < MyCellArray02.Count)
                {
                    HSSFCell cell02 = (HSSFCell)Row.CreateCell((short)1);

                    cell02.SetCellValue(MyCellArray02[i]);
                }
                else
                {
                    HSSFCell cell02 = (HSSFCell)Row.CreateCell((short)1);

                    cell02.SetCellValue("");
                }

                Row.RowStyle = MyWorkbook.CreateCellStyle();              → Define Style Of Row

                Row.RowStyle.BorderBottom = BorderStyle.Double;           → Change Border Of Row

                cell.CellStyle = MyWorkbook.CreateCellStyle();            → Define Style Of Cells

                cell.CellStyle.BorderRight = BorderStyle.Thin;            → Change Border Of Cells
                cell.CellStyle.BorderBottom = BorderStyle.Dashed;
                cell.CellStyle.BottomBorderColor = HSSFColor.Red.Index;

                HSSFFont MyFont = (HSSFFont)MyWorkbook.CreateFont();      → Define Font

                MyFont.FontName = "Tahoma";                              → Change Font, Font Height, Font Color And etc
                MyFont.FontHeightInPoints = 14;
                MyFont.Color = HSSFColor.Gold.Index;
                MyFont.Boldweight = (short)FontBoldweight.Bold;

                cell.CellStyle.SetFont(MyFont);                          → Setting Font For Cell
            }

            MyWorkbook.Write(MyAddress);                                 → Writting Your Data In Excel File

            MyWorkbook.Close();                                          → Closing My Work Book
        }

        if (GUI.Button(new Rect(60, 120, 200, 30), "Reading Excel Files With " + "<color=yellow>NPOI</color>"))
        {
            ReadExcelEnable_NPOI = true;                                 → Enabling NPOI Component

            ReadExcelEnable_ExcelDataReader = false;                     → Enabling ExcelDataReader Component

            MyString.Clear();                                            → Reset The MyString List
        }

        if (ReadExcelEnable_NPOI == true)
        {
            HSSFWorkbook MyBook;                                         → Finding Excel File For Reading

            using (FileStream MyAddress_Read = new FileStream(Application.dataPath + "/My First Excel.xls", FileMode.Open, FileAccess.Read, FileShare.ReadWrite))
            {
                MyBook = new HSSFWorkbook(MyAddress_Read);
            }

            ISheet Sheet_Read = MyBook.GetSheet(MySheetName);            → Finding Sheet With Sheet Name

            for (int row = 0; row <= Sheet_Read.LastRowNum; row++)
            {
                IRow Row_Read = Sheet_Read.GetRow(row);                 → Finding Rows Of Sheet

                for (int cells = 0; cells < Row_Read.LastCellNum; cells++)
                {
                    GUI.Label(new Rect(400 + cells * 200, 100 + row * 40, 200, 200),Row_Read.GetCell(cells).ToString());   → Showing Content Of Excel

                    //Arabic And Persian Support
                    //GUI.Label(new Rect(400 + cells * 200,100+ row * 40, 200, 20),ArabicFixer.Fix(Row_Read.GetCell(cells).ToString()));   → Arabic Support
                }
            }
        }

        if (GUI.Button(new Rect(10, 160, 300, 30), "Reading Excel Files With " + "<color=yellow>ExcelDataReader</color>"))
        {
            ReadExcelEnable_ExcelDataReader = true;

            ReadExcelEnable_NPOI = false;

            MyString.Clear();
        }

        if (ReadExcelEnable_ExcelDataReader == true)
        {
            IExcelDataReader excelReader = ExcelReaderFactory.CreateBinaryReader(MyAddress);

            DataSet Result = excelReader.AsDataSet();

            excelReader.IsFirstRowAsColumnNames = true;

            int NumberOfRows = Result.Tables[0].Rows.Count;

            int NumberOfCols = Result.Tables[0].Columns.Count;

            for (int i = 0; i < NumberOfRows; i++)
            {
                for (int j = 0; j < NumberOfCols; j++)
                {
                    if (i == NumberOfRows - 1 && j == NumberOfCols - 1)
                    {
                        ReadExcelEnable_ExcelDataReader = false;
                    }

                    MyString.Add(Result.Tables[0].Rows[i][j].ToString());
                }
            }

            for (int b = 0; b < MyString.Count; b++)
            {
                GUI.Label(new Rect(400,100+ 20 * b, 200, 20),MyString[b]);

                //Arabic And Persian Support
                //GUI.Label(new Rect(400, 100 + 20 * b, 200, 20), ArabicFixer.Fix(MyString[b]));
            }
        }
    }
}
```