How to import from jagged array in C# with ByteScout Spreadsheet SDK

Tutorial on how to import from jagged array in C#

Learn how to import from jagged array in C# with this source code sample. ByteScout Spreadsheet SDK is the library (SDK) that is capable of writing, reading, modifying and calculating Excel and CSV spreadsheets. Most popular formulas can be calculated and reculculated with Excel installed. You may import or export data to and from CSV, XML, JSON as well as to and from databases, arrays. It can import from jagged array in C#.

This rich sample source code in C# for ByteScout Spreadsheet SDK includes the number of functions and options you should do calling the API to import from jagged array. In order to implement the functionality, you should copy and paste this code for C# below into your code editor with your app, compile and run your application. Test C# sample code examples whether they respond your needs and requirements for the project.

Free trial version of ByteScout Spreadsheet SDK is available on our website. Documentation and source code samples are included.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Spreadsheet SDK

Explore API Documentation

Get Free Training for ByteScout Spreadsheet SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Program.cs

```csharp
using System.Diagnostics;
using System.IO;

namespace Bytescout.Spreadsheet.Demo.Csharp.ImportFromJaggedArray
{
    class Program
    {
        static void Main(string[] args)
        {
            const string fileName = "CSharpImportFromJaggedArray.xls";

            // Create a new spreadsheet
            Spreadsheet spreadsheet = new Spreadsheet();

            // Get the data from the jagged array that we want to import
            string[][] periodicTable = GetJaggedArray();

            // Import data into spreadheet
            spreadsheet.ImportFromJaggedArray(periodicTable);

            // Save the spreadsheet
            if (File.Exists(fileName)) File.Delete(fileName);
            spreadsheet.SaveAs(fileName);

            // Close spreadsheet
            spreadsheet.Close();

            // Open the spreadsheet
            Process.Start(fileName);
        }

        /// <summary>
        /// Creates a jagged array of the period table of elements
        /// </summary>
        /// <returns>Jagged array of the periodic table of elements</returns>
        private static string[][] GetJaggedArray()
        {
            string[][] periodicTable = {
                        new string[] {"H","He"},
                        new string[] {"Li","Be","B","C", "N", "O", "F", "Ne"},
                        new string[] {"Na","Mg","Al","Si", "P", "S", "Cl", "Ar"},
                        new string[] {"K", "Ca", "Sc", "Ti", "V", "Cr", "Mn", "Fe",
                    };
            return periodicTable;
        }
    }
}
```

VIDEO

https://www.youtube.com/watch?v=nm_7I0PN1TY

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Spreadsheet SDK Home Page
Explore ByteScout Spreadsheet SDK Documentation
Explore Samples
Sign Up for ByteScout Spreadsheet SDK Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co