

CASE STUDIES

Aspose.Cells for .NET Case Study



The Digital Group

Using Aspose.Cells for .NET to Generate Financial Excel Reports

Author:

- The Digital Group
4th September 2013

About The Digital Group

The Digital Group (T/DG) is a premier provider of information technology services with extensive expertise in all facets of software development. With capabilities at all points of the software life-cycle, we can provide support from concept to development to operations. For the last decade our teams have been delivering industry leading solutions for our customers.

The Digital Group, with its headquarters in Princeton, New Jersey, was incorporated in 1999 having a global talent pool of top notch professionals providing technology and management consulting services and solutions, with offices in US, India, Fiji and Australia. TDG's Services & Solutions delivery capability is anchored in an integrated set of core competencies that span, people, processes and technology coupled with practice competencies in System Administration, Web Software Engineering, Enterprise Applications, Enterprise Integration, Corporate Compliance and Legacy Migration. We provide a cost-optimized global delivery model with onsite, near-site and off-shore capabilities.

Solution

One of our client has requirement to automate the generation of Complex Financial reports in Excel format. Considering the complexity of Reports and required performance criteria, Technical team evaluated few options and final decision has to be made between Couple of options before finalizing the solution. Options were –

- 1. Aspose.Cells for .NET (7.5.2)**
- 2. Microsoft.Office.Interop.Excel**

We decided to do POC of small part of functionality with above two options. Functionality was, data tabled data had to be extracted from dataset and exported to an Excel sheet.

How does it work?

ASPOSE: For exporting data to Excel we have used Aspose.Cells for .NET, which provides the most flexible group of components that enable .NET applications to create and manage Excel® spreadsheets without requiring Microsoft Excel® to be installed on the server.

It has **ImportDataTable** method that imports a System.Data.DataTable object into a worksheet. We need just a few lines of code to get the export data task done.

```
string targetFilePathAspose = @"D:\AsposeData\ASPOSE_Excel.xlsx";  
Workbook workBook = new Workbook();  
Worksheet workSheet = workBook.Worksheets[0];  
workSheet.Cells.ImportDataTable(ds.Tables["Info"], true, "A1");  
workBook.Save(targetFilePathAspose);
```

With **INTEROP**, same task is done using looping which takes more time in completing the task.

We need to loop through the data table columns and rows and write data to cells in Excel sheet.

```
Microsoft.Office.Interop.Excel.Application excel =  
    new Microsoft.Office.Interop.Excel.Application();  
  
string filePath = @"d:\AsposeData\INTREOP_Excel.xlsx";  
var workbook = excel.Workbooks.Open(filePath);  
var worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.Worksheets.Add();  
  
//System.Data.DataTable table = ds.Tables["Info"].DefaultView;  
int ColumnIndex = 0;  
  
foreach (DataColumn col in ds.Tables["Info"].Columns)  
{  
    ColumnIndex++;  
    excel.Cells[1, ColumnIndex] = col.ColumnName;  
}  
  
int rowIndex = 0;  
  
foreach (DataRow row in ds.Tables["Info"].Rows)  
{  
    rowIndex++;  
    ColumnIndex = 0;  
  
    foreach (DataColumn col in ds.Tables["Info"].Columns)  
    {  
        ColumnIndex++;  
        excel.Cells[rowIndex + 1, ColumnIndex] = row[col.ColumnName].ToString();  
    }  
}  
  
workbook.Save();  
workbook.Close();  
excel.Quit();
```

Following is the result of comparison made between both the approaches.

We ran the test with different number of records and noticed the time difference between ASPOSE and INTEROP.

Scenario 1

To export 2500 records **ASPOSE** took '0.7252908' **seconds** where **INTEROP** took **2 minutes**.

State

Export Data To Excel Using ASPOSE And INTEROP

Total Records : 2500

ASPOSE :

	FirstName	MiddleName	LastName	Address1	Address2	City
▶	David	J	White	street1	moon road	ca
	James	A	Wills	east street	sun road	ca
	John	J	Smith	street1	moon road	ca
	Glyn	A	Wills	east street	sun road	ca
	Pierre	A	Gauthier	east street	sun road	ca
	Louis	S	Boucher	east street	sun road	ca

Time Taken Using ASPOSE : 00:00:00.7252908
[Hours/Minutes/Seconds/MiliSeconds]

INTEROP:

	FirstName	MiddleName	LastName	Address1	Address2	City
▶	David	J	White	street1	moon road	ca
	James	A	Wills	east street	sun road	ca
	John	J	Smith	street1	moon road	ca
	Glyn	A	Wills	east street	sun road	ca
	Pierre	A	Gauthier	east street	sun road	ca
	Louis	S	Boucher	east street	sun road	ca
	Antoine	E	Caron	east street	sun road	ca

Time Taken Using INTEROP : 00:02:07.3607468
[Hours/Minutes/Seconds/MiliSeconds]

Scenario 2

To export 1050 records **ASPOSE** took '0.7243860' **seconds** where **INTEROP** took **55 seconds**.

State

Export Data To Excel Using ASPOSE And INTEROP

Total Records : 1050

ASPOSE :

	FirstName	MiddleName	LastName	Address1	Address2	City
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE

Time Taken Using ASPOSE : 00:00:00.7243860
[Hours/Minutes/Seconds/MiliSeconds]

INTEROP:

	FirstName	MiddleName	LastName	Address1	Address2	City
▶	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE
	John	A	Cena	Add1	Add2	DE

Time Taken Using INTEROP : 00:00:55.4904617
[Hours/Minutes/Seconds/MiliSeconds]

Scenario 3

To export **5014** records **ASPOSE** took '0.9859248' **seconds** where **INTEROP** took **5 Minutes and 19 seconds**.

State

Total Records : 5014

ASPOSE :

	FirstName	MiddleName	LastName	Address1	Address2	State
▶	Charles	J	Girard	east street	sun road	NY
	Michel	S	Perrault	east street	sun road	NY
	Jacques	A	Tremblay	east street	sun road	NY
	Augustin	S	Giroux	east street	sun road	NY
	Jean	G	Demers	east street	sun road	NY
	Nicolas	D	Cloutier	east street	sun road	NY

Time Taken Using ASPOSE : 00:00:00.9859248
[Hours/Minutes/Seconds/MiliSeconds]

INTEROP:

	FirstName	MiddleName	LastName	Address1	Address2	State
▶	Charles	J	Girard	east street	sun road	NY
	Michel	S	Perrault	east street	sun road	NY
	Jacques	A	Tremblay	east street	sun road	NY
	Augustin	S	Giroux	east street	sun road	NY
	Jean	G	Demers	east street	sun road	NY
	Nicolas	D	Cloutier	east street	sun road	NY
	Marie	L	Allard	east street	sun road	NY

Time Taken Using INTEROP : 00:05:19.1495674
[Hours/Minutes/Seconds/MiliSeconds]

Experience

Excel Financial Report generation was one of the core functionality of the project.

Requirement was to have flexible financial report output with Performance as a crucial factor. Project time line was tight as well. We were looking for solution which not only meet all requirements but also save development time.

After POC We figured out that Aspose.Cells for .NET was the best fit with below highlights,

Functional Features: We can perform variety of Excel operations like

Worksheet scope – Add, Remove Rename worksheet. Sorting data, managing hyperlinks, pictures, comments.

Row-Column scope – We generated rows and columns with financial data for multiple years. Operations like Hide/Unhide, Freeze/Unfreeze, formatting of entire Row/Column can be easily performed.

Cell scope – Populating values, setting formula, Formatting, Replacing values in Cells or Range of cells can be achieved with ease. Even cells can merged/unmerged or use of Named ranges is possible.

Performance: As showcased above Aspose was clear winner in performance. Time taken to generate Thousands of rows with considerable columns was less than a second.

All above functional features can be achieved with great operational speed for higher number of rows/columns

Availability & Support: We used the Free Trial version of Aspose.Cells for .NET for feasibility check. It was easily available from Aspose. We discussed with Aspose Support representative regarding licensing queries and concluded on required license type and number. Aspose suggested Developer OEM license which meet all requirements with optimized cost.

We also did not require to install Microsoft Office on the Aspose server.

Rapid Development: We noticed that to achieve particular functionality, line of code required using Aspose is lesser than other options.

Aspose has smooth Learning curve as developer completed the POC within couple of Days. There are samples available on Aspose site almost for all functional features which were helpful. Overall it resulted into reduced Development efforts and invariably time.

We recommended client to use Aspose as it was clear winner in performance & features.

Next Steps

In current project, we will be using Aspose.Cells for .NET. We have also decided that, going forward, we will be considering to evaluate Aspose products wherever applicable.

We are using Aspose products for couple of other clients where client himself has recommended for Aspose.

Summary

We concluded on below points

1. INTEROP approach was taking significant amount of time as compared to ASPOSE. Performance was better in ASPOSE.
2. ASPOSE required fewer lines of code as compared to INTEROP which requires more coding which involves looping.
3. You require to purchase separate license for ASPOSE whereas there is no additional license required for INTEROP.

Aspose.Cells for .NET was the best fit for our project requirement due to its performance & features.

As mentioned earlier, developers can easily pick up on Aspose details, thanks to smooth learning curve and Samples available on Aspose site itself. Using Aspose, we are able to save some amount of development efforts as well.

Aspose team was supportive throughout the POC and implementation.

Our development team was satisfied with the ease, performance and reliability of Aspose products. We did finalize Aspose for future implementation for this client. Not only that we recommend Aspose products to other development teams.

We would like to recommend Aspose components whenever fits into requirement.