Structure of a SpreadsheetML document

01/14/2025

The document structure of a SpreadsheetML document consists of the <workbook/> element that contains <sheets/> and <sheet/> elements that reference the worksheets in the workbook. A separate XML file is created for each worksheet. These elements are the minimum elements required for a valid spreadsheet document. In addition, a spreadsheet document might contain , <chartsheet/>, <pivotTableDefinition/>, or other spreadsheet related elements.

① Note

Interested in developing solutions that extend the Office experience across multiple platforms? Check out the new <u>Office Add-ins model</u>. Office Add-ins have a small footprint compared to VSTO Add-ins and solutions, and you can build them by using almost any web programming technology, such as HTML5, JavaScript, CSS3, and XML.

Important Spreadsheet Parts

Using the Open XML SDK for Office, you can create document structure and content that uses strongly-typed classes that correspond to SpreadsheetML elements. You can find these classes in the DocumentFormat.OpenXML.Spreadsheet namespace. The following table lists the class names of the classes that correspond to some of the important spreadsheet elements.

Expand table

Package Part	Top Level SpreadsheetML Element	Open XML SDK Class	Description
Workbook	<workbook></workbook>	Workbook	The root element for the main document part.
Worksheet	<worksheet></worksheet>	Worksheet	A type of sheet that represent a grid of cells that contains text, numbers, dates or formulas. For more information, see Working with sheets.
Chart Sheet	<chartsheet></chartsheet>	Chartsheet	A sheet that represents a chart that is stored as its own sheet. For more information, see Working with sheets.
Table		Table	A logical construct that specifies that a range of data belongs to a single dataset. For more information, see Working with SpreadsheetML tables.
Pivot Table	<pre><pivottabledefinition></pivottabledefinition></pre>	PivotTableDefinition	A logical construct that displays aggregated view of data in an understandable layout. For more information, see Working with PivotTables.

Package Part	Top Level SpreadsheetML Element	Open XML SDK Class	Description
Pivot Cache	<pre><pivotcachedefinition></pivotcachedefinition></pre>	PivotCacheDefinition	A construct that defines the source of the data in the PivotTable. For more information, see Working with PivotTables.
Pivot Cache Records	<pre><pivotcacherecords></pivotcacherecords></pre>	PivotCacheRecords	A cache of the source data of the PivotTable. For more information, see Working with PivotTables.
Calculation Chain	<calcchain></calcchain>	CalculationChain	A construct that specifies the order in which cells in the workbook were last calculated. For more information, see Working with the calculation chain.
Shared String Table	<sst></sst>	SharedStringTable	A construct that contains one occurrence of each unique string that occurs on all worksheets in a workbook. For more information, see Working with the shared string table.
Conditional Formatting	<conditionalformatting></conditionalformatting>	ConditionalFormatting	A construct that defines a format applied to a cell or series of cells. For more information, see Working with conditional formatting.
Formulas	<f></f> />	CellFormula	A construct that defines the formula text for a cell that contains a formula. For more information, see Working with formulas.

Minimum Workbook Scenario

The following text from the Standard ECMA-376 introduces the minimum workbook scenario.

The smallest possible (blank) workbook must contain the following:

A single sheet

A sheet ID

A relationship Id that points to the location of the sheet definition

© Ecma International: December 2006.

Open XML SDK Code Example

This code example uses the classes in the Open XML SDK to create a minimum, blank workbook.

C#

```
static void CreateSpreadsheetWorkbook(string filepath)
   // Create a spreadsheet document by supplying the filepath.
   // By default, AutoSave = true, Editable = true, and Type = xlsx.
   using (SpreadsheetDocument spreadsheetDocument = SpreadsheetDocument.Create(filepath,
SpreadsheetDocumentType.Workbook))
       // Add a WorkbookPart to the document.
       WorkbookPart workbookPart = spreadsheetDocument.AddWorkbookPart();
       workbookPart.Workbook = new Workbook();
       // Add a WorksheetPart to the WorkbookPart.
       WorksheetPart worksheetPart = workbookPart.AddNewPart<WorksheetPart>();
       worksheetPart.Worksheet = new Worksheet(new SheetData());
       // Add Sheets to the Workbook.
       Sheets sheets = workbookPart.Workbook.AppendChild(new Sheets());
       // Append a new worksheet and associate it with the workbook.
       Sheet sheet = new Sheet() { Id = workbookPart.GetIdOfPart(worksheetPart), SheetId
= 1, Name = "mySheet" };
       sheets.Append(sheet);
}
```

Typical Workbook Scenario

A typical workbook will not be a blank, minimum workbook. A typical workbook might contain numbers, text, charts, tables, and pivot tables. Each of these additional parts is contained within the .zip package of the spreadsheet document.

The following figure shows most of the elements that you would find in a typical spreadsheet.

Figure 2. Typical spreadsheet elements

