

# Sales Report Sample

## Intro

Demonstrates usage of the WordApplication and WordTemplate objects together. Generates a nested-data sales report from AdventureWorks data.

This sample uses the WordApplication object to create a new Word document that displays nested salesperson/sales item data. The data for this demo comes from the AdventureWorks database. First, a query for a list of salespeople is executed and a table for each salesperson is created with a surrounding bookmark. Then, the document is opened as a WordTemplate object and the bookmarks for each sales person are bound and populated with sales line-item data. The end-result is a document that displays five sales line items for each employee in a nested format.

```
DataTable salesData;

public void GenerateDocument()
{
    // Initialize the WordApplication and create our document.
    WordApplication app = new WordApplication();
    Document doc = app.Create();

    // Add the date to the header
    Element hdr = doc.Sections[0].GetHeader(Section.HeaderFooterType.All);
    Paragraph paragraph = hdr.InsertParagraphAfter(null);
    paragraph.Formatting.TextJustification = ParagraphFormatting.Justification.Right;
    paragraph.InsertTextAfter(new DateTime().ToString(), true);

    // Add a title to the document
    paragraph = doc.InsertParagraphAfter(null);
    paragraph.Formatting.TextJustification = ParagraphFormatting.Justification.Center;
    CharacterRun run = paragraph.InsertTextAfter("AdventureWorks Sales Report", true);
    run.Font.FontName = "Arial";
    run.Font.Italic = true;
    run.Font.Shadow = true;
    run.Font.FontSize = 26.0f;

    // Add a sub title to the document
    doc.InsertParagraphAfter(null).InsertTextAfter("The last five sales for each sales
person", true);

    // Create a font to be used when inserting names
    Font nameFont = doc.CreateFont();
    nameFont.Bold = true;
    nameFont.FontSize = 16.0;
    nameFont.FontName = "Arial";

    // Get DataTable
    salesData = getData();

    // Get unique salespeople
    string[,] people = getSalesPeople();

    // Insert the repeat block for each salesperson
    for (int i=0; i<people.GetLength(0); i++)
    {
        // Create a table that only contains a header.
        Table table = doc.InsertTableAfter(1, 1);
        TableCell cell = table[0, 0];
```

```

Shading shading = cell.Shading;
shading.Pattern = Shading.ShadingPattern.Percent25;
cell.InsertTextAfter(people[i,0], nameFont);

// create the repeat block table
Table repeatTable = table.InsertTableAfter(1, 2);
repeatTable.InsertParagraphAfter(null);

// Insert the merge fields
TableCell dateCell = repeatTable[0, 0];
dateCell.InsertMergeFieldAfter("OrderDate", "OrderDate");
TableCell totalCell = repeatTable[0, 1];
totalCell.InsertMergeFieldAfter("TotalSale", "TotalSale");

// Create the bookmark on the row. Use the id to create a unique name. This is our
repeat block
repeatTable.CreateBookmarkOnRow(0, "SalesPerson"+people[i,1]); // was using a unique
id
}

// Init our template from the application object.
WordTemplate template = new WordTemplate();
template.Open(app, doc);

// Iterate through all bookmarks and get the sales person id from the bookmark name
string[] bookmarks = template.Bookmarks;
for (int x = 0; x < bookmarks.Length; x++)
{
// Create a prepared statement and associate the result set to the repeat block
int salesPersonID = Convert.ToInt32(bookmarks[x].Substring(11));
DataTable resultTable = this.GetSalesData(salesPersonID);

template.SetRepeatBlock(resultTable, bookmarks[x], 5);
}

// Process the template and save it
template.Process();
template.Save(@"..\..\WordOutputFiles\SalesReport_out.doc");
}

// Gets data for particular salesperson
private DataTable GetSalesData(int salesID)
{
    DataTable dt = new DataTable();
    dt.Columns.Add("name");
    dt.Columns.Add("OrderDate");
    dt.Columns.Add("TotalSale");
    dt.Columns.Add("SalesPersonID");
    foreach (DataRow row in salesData.Rows)
    {
        if (Int32.Parse((string)row["SalesPersonID"]) == salesID)
            dt.ImportRow(row);
    }
    return dt;
}

// Gets a 2D array of unique salespeople's name and their SalesPersonID
private string[,] getSalesPeople()

```

```

{
    string[, ] ppl = new string[8,2];
    string prev = "";
    int i=0;
    foreach (DataRow row in salesData.Rows)
    {
        if (!(prev == (string)row["name"]))
        {
            prev = (string)row["name"];
            Console.WriteLine(prev);
            Console.WriteLine(salesData.Rows.IndexOf(row));
            ppl[i, 0] = (string)row["name"];
            ppl[i, 1] = (string)row["SalesPersonID"];
            i++;
        }
    }
    return ppl;
}

//Get Data
// Returns the data for salesData DataTable
// This DataTable is programmatically defined for simplicity

private DataTable getData()
{
    DataTable dt = new DataTable();
    dt.Columns.Add("name");
    dt.Columns.Add("OrderDate");
    dt.Columns.Add("TotalSale");
    dt.Columns.Add("SalesPersonID");
    dt.Rows.Add(new string[] { "Syed Abbas", "01/01/2004", "$4650.73", "285" });
    dt.Rows.Add(new string[] { "Syed Abbas", "01/01/2004", "$590.61", "285" });
    dt.Rows.Add(new string[] { "Syed Abbas", "01/02/2004", "$1825.89", "285" });
    dt.Rows.Add(new string[] { "Syed Abbas", "01/05/2004", "$894.34", "285" });
    dt.Rows.Add(new string[] { "Syed Abbas", "01/05/2004", "$21479.49", "285" });
    dt.Rows.Add(new string[] { "David Campbell", "01/01/2002", "$43189.42", "283" });
    dt.Rows.Add(new string[] { "David Campbell", "01/01/2002", "$1854.20", "283" });
    dt.Rows.Add(new string[] { "David Campbell", "01/01/2002", "$5778.15", "283" });
    dt.Rows.Add(new string[] { "David Campbell", "01/01/2002", "$70303.53", "283" });
    dt.Rows.Add(new string[] { "David Campbell", "01/01/2002", "$2828.58", "283" });
    dt.Rows.Add(new string[] { "Garrett Vargas", "01/01/2002", "$29479.03", "278" });
    dt.Rows.Add(new string[] { "Garrett Vargas", "01/01/2002", "$7278.20", "278" });
    dt.Rows.Add(new string[] { "Garrett Vargas", "01/01/2003", "$46.03", "278" });
    dt.Rows.Add(new string[] { "Garrett Vargas", "01/01/2003", "$3737.68", "278" });
    dt.Rows.Add(new string[] { "Garrett Vargas", "01/01/2004", "$3752.24", "278" });
    dt.Rows.Add(new string[] { "Tsvi Reiter", "01/01/2002", "$97132.51", "279" });
    dt.Rows.Add(new string[] { "Tsvi Reiter", "01/01/2002", "$20991.87", "279" });
    dt.Rows.Add(new string[] { "Tsvi Reiter", "01/01/2002", "$31857.21", "279" });
    dt.Rows.Add(new string[] { "Tsvi Reiter", "01/01/2002", "$556.20", "279" });
    dt.Rows.Add(new string[] { "Tsvi Reiter", "01/01/2002", "$10821.08", "279" });
    dt.Rows.Add(new string[] { "Jillian Carson", "01/01/2002", "$18909.09", "277" });
    dt.Rows.Add(new string[] { "Jillian Carson", "01/01/2002", "$26167.11", "277" });
    dt.Rows.Add(new string[] { "Jillian Carson", "01/01/2002", "$37321.95", "277" });
    dt.Rows.Add(new string[] { "Jillian Carson", "01/01/2002", "$2781.01", "277" });
    dt.Rows.Add(new string[] { "Jillian Carson", "01/01/2003", "$27267.94", "277" });
    dt.Rows.Add(new string[] { "Michael Blythe", "01/01/2002", "$8122.50", "275" });
    dt.Rows.Add(new string[] { "Michael Blythe", "01/01/2002", "$3988.56", "275" });
    dt.Rows.Add(new string[] { "Michael Blythe", "01/01/2002", "$25606.63", "275" });
    dt.Rows.Add(new string[] { "Michael Blythe", "01/01/2002", "$51251.30", "275" });
}

```

```
dt.Rows.Add(new string[] { "Michael Blythe", "01/01/2002", "$14746.70", "275" });
dt.Rows.Add(new string[] { "Rachel Valdez", "01/01/2004", "$1735.65", "288" });
dt.Rows.Add(new string[] { "Rachel Valdez", "01/01/2004", "$49337.61", "288" });
dt.Rows.Add(new string[] { "Rachel Valdez", "01/01/2004", "$14612.14", "288" });
dt.Rows.Add(new string[] { "Rachel Valdez", "01/01/2004", "$2713.16", "288" });
dt.Rows.Add(new string[] { "Rachel Valdez", "01/01/2004", "$34728.17", "288" });
dt.Rows.Add(new string[] { "Amy Alberts", "01/01/2004", "$429.62", "287" });
dt.Rows.Add(new string[] { "Amy Alberts", "01/02/2003", "$32774.36", "287" });
dt.Rows.Add(new string[] { "Amy Alberts", "01/02/2003", "$6657.48", "287" });
dt.Rows.Add(new string[] { "Amy Alberts", "01/02/2004", "$251.35", "287" });
dt.Rows.Add(new string[] { "Amy Alberts", "01/02/2004", "$35356.74", "287" });
return dt;
```

```
}
```

## Downloads

Output: [SalesReport\\_out.doc](#)