

Nested Product Catalog Demo

Intro

Use the new features of the WordTemplate object to create feature-rich reports that were previously only possible with many lines of code. This sample uses nested repeat blocks to produce a report very much like the Word Application Product Catalog.

This sample uses nested repeat blocks to produce a report very much like the [Word Application Product Catalog](#) sample, but with many fewer lines of code. Instead of using many database queries and manually constructing each piece of the report, this sample uses two queries and binds them to a template that builds the entire report.

Note that using nested repeat blocks is only supported by using docx files.

Code

```
public void GenerateDocument()
{
    WordTemplate wt = new WordTemplate();
    wt.Open(@"..\..\WordTemplateFiles\NestedCatalogTemplate.docx");

    DataTable dt = new DataTable();
    dt = GetCatalogData(@"..\..\WordData\NestedProductCatalogData.csv");
    dt.Columns.Add("LargePhoto", typeof(byte[]));

    //Retrieves each picture in the NestedProductCatalogImages folder
    //and get a byte[] for each one using the GetCatalogPictures Helper
method.
    //Then add this byte[] to the LargePhoto column in dt.
    for (int k = 0; k < dt.Rows.Count; k++)
    {
        string imageName = dt.Rows[k].ItemArray[6].ToString();
        DataRow dr = dt.Rows[k];
        dr[7] = GetCatalogPictures(imageName);
    }

    wt.SetDataSource(DateTime.Now, "HeaderData");
    //Sets the data source for the index of the catalog with bookmark string
"Index"

    wt.SetRepeatBlock(GetCatalogData(@"..\..\WordData\NestedProductCatalogIndexData.csv"),
"Index");

    //Sets the data source for the body of the catalog with bookmark string
"Catalog"
    wt.SetRepeatBlock(dt, "Catalog");

    wt.Process();
    //Save the product catalog in the desired location on the disc
    wt.Save(@"..\..\WordOutputFiles\NestedProductCatalog_output.docx");
}

//Helper method to parse the csv files with index and catalog data into datatables
private DataTable GetCatalogData(string csvFileName)
{
    DataTable dt = new DataTable();
```

```
        using (GenericParserAdapter parser = new
GenericParserAdapter(csvFileName))
        {
            parser.ColumnDelimiter = ',';
            parser.FirstRowHasHeader = true;

            dt = parser.GetDataTable();
        }
        return dt;
    }
    //Open desired file, picName, in Images\NestedProductCatalogImages and
    //return the bytearray that makes up the image.
    private byte[] GetCatalogPictures(string picName)
    {

        //Opens image and returns byte array
        byte [] imageArray =
File.ReadAllBytes(@"..\..\WordImages\NestedProductCatalogImages\"+picName);
        return imageArray;
    }
}
```

Downloads

Template: NestedCatalogTemplate.docx
Output: NestedProductCatalog_output.docx