Sample Expense Report

Intro

This sample uses ExcelTemplate to populate a generic expense report template.

This example demonstrates how to use single object data markers (ones that start with %%=$) to import the data for the header by calling the BindCellData method multiple times. It also demonstrates how to use 2-D array data markers to import expense data from an Object array by calling the BindData method once.

Requirements
This sample requires OfficeWriter Enterprise Edition to be installed because the OfficeWriter Grouping and Nesting is only available in the Enterprise Edition of the product.

Code

```
public class ExpenseReport
    {
        /// <summary>
        /// Build the report with ExcelTemplate
        /// </summary>
        public void GenerateReport()
        {
            // Here we create some data to populate into
            //the template for the sample In your application
            //this data could possibly come straight from a database


            // These strings will bind to variable data markers
            string empPurpose = "Quarterly Expense Report";
            string empName = "Sammy McDougal";
            string empDept = "Sales (Eastern region)";
            string empSSN = "111-222-1212";
            string empPosition = "VP";
            string empID = "242";
            string empManager = "C.M. Burns";
            string dateFrom = "1/1/2002";
            string dateTo = "5/31/2002";

            // These arrays hold data that bind to the array
            //data markers and will be repeated
            //in the template for each row imported

            string[] colNames = { "Date", "Description", "Hotel", "Fuel", "Meals",
"Misc" };

            object[,] values = {{"1/2/2002", "2/5/2002", "3/15/2002", "4/10/2002",
"5/17/2002"},
                                {"Dev Conference", "Lunch w/ Client", "Delivery",
"Dinner w/ Client", "Sales Seminar"},
                                {200.04, 223.43, 184.23, 331.24, 112.11},
                                {10.00, 0.99, 11.23, 32.23, 10.00},
                                {30.00, 23.04, 44.44, 19.99, 5.00},
                                {5.00, 11.00, 13.03, 3.24, 33.12}};
```

```
            // Create an instance of ExcelTemplate
            ExcelTemplate xlt = new ExcelTemplate();

            // Open the template workbook
            xlt.Open(@"..\..\ExcelTemplateFiles\ExpenseReportTemplate.xlsx");

            // Pass the strings to the BindCellData method
            DataBindingProperties cellBindingProperties =
xlt.CreateDataBindingProperties();
            xlt.BindCellData(empPurpose, "EmpPurpose", cellBindingProperties);
            xlt.BindCellData(empName, "EmpName", cellBindingProperties);
            xlt.BindCellData(empDept, "EmpDept", cellBindingProperties);
            xlt.BindCellData(empSSN, "EmpSSN", cellBindingProperties);
            xlt.BindCellData(empPosition, "EmpPosition", cellBindingProperties);
            xlt.BindCellData(empID, "EmpID", cellBindingProperties);
            xlt.BindCellData(empManager, "EmpManager", cellBindingProperties);
            xlt.BindCellData(dateFrom, "DateFrom", cellBindingProperties);
            xlt.BindCellData(dateTo, "DateTo", cellBindingProperties);

            // Limit the number of rows to the maximum number of rows that Excel
             //can support. ExcelTemplate expects the array's first index to be
             //the row number and the second index to be the column number. Our
             //array is the opposite, so we have to transpose the data.

            DataBindingProperties bindingProperties =
xlt.CreateDataBindingProperties();
            bindingProperties.MaxRows = ExcelTemplate.ALL_ROWS;
            bindingProperties.Transpose = true;
            xlt.BindData(values, colNames, "Exp", bindingProperties);


            // Call the Process() method to populate the template
            //with the data source values

            xlt.Process();

            // Save the report
            xlt.Save(@"..\..\ExcelOutputFiles\ExpenseReport_output.xlsx");
        }

    }
```

Downloads

- Template: ExpenseReportTemplate.xlsx
- Output: ExpenseReport_output.xlsx