Calendar Sample

Intro

Create a calendar document for any month/year combination.

The template document for this demo is a blank calendar with Word merge fields in every date block. The code in this demo determines the current month and year using a DateTime Object, does some calendar calculations (eg, figuring out which day of the week the month begins on and how many days are in the month), and finally pushes the date, month, and year data into the document with WordTemplate.

Code

```
 public class CalendarGenerator
    {
        private int year;
        private int month;


        /// <summary>
        /// Build the report with WordTemplate
        /// </summary>
        public void GenerateDocument()
        {
            //Set the year and month to the current year and month
            DateTime today = DateTime.Now;
            year = today.Year;
            month = today.Month;

            // Form the name array that will be passed to WordWriter.  The elements
            // of this array are the names of the merge fields in the calendar
document

            string[] arrNames = {"Block1", "Block2", "Block3", "Block4", "Block5",
"Block6",
                                 "Block7", "Block8", "Block9", "Block10",
"Block11", "Block12",
                                 "Block13", "Block14", "Block15", "Block16",
"Block17", "Block18",
                                 "Block19", "Block20", "Block21", "Block22",
"Block23", "Block24",
                                 "Block25", "Block26", "Block27", "Block28",
"Block29", "Block30",
                                 "Block31", "Block32", "Block33", "Block34",
"Block35", "Block36",
                                 "Block37", "Block38", "Block39", "Block40",
"Block41", "Block42", "MonthYear"};

            // Determine on which day of the week the month begins
            DateTime thisDate = new DateTime(year, month, 1);
            DayOfWeek startday = thisDate.DayOfWeek;

            // Get the number of days in the selected month
            int daysinmonth = DateTime.DaysInMonth(year, month);
```

```
            //Declare the object array that will be used to hold the
            // values inserted into the calendar template

            object[] arrDayBlocks = new object[43];

            // The first date of the month is, as always, 1
            int dayofmonth = 1;

            // Loop through all of the available blocks in the calendar
            for(int iDayBlock = 0; iDayBlock <= 41; iDayBlock++)
            {
                // If the block exists before the first day of the month, enter an
empty string
                if((iDayBlock < (int)startday) || (dayofmonth > daysinmonth))
                    arrDayBlocks[iDayBlock] = String.Empty;
                else
                {
                    // Otherwise, it's a valid block, so enter the day of the month
                    arrDayBlocks[iDayBlock] = dayofmonth;
                    dayofmonth++;
                }
            }

            // The last element of the value array is reserved for a DateTime struct.
             // Using MergeField date formatting codes, the template merge fields will
             // display the Month name and 4-digit year in the appropriate places

            arrDayBlocks[42] = thisDate;

            // Instantiate a WordTemplate object
            WordTemplate wt = new WordTemplate();

            // Open the template file
            string templatePath = @"..\..\WordTemplateFiles\CalendarTemplate.docx";
            wt.Open(templatePath);

            // Set the datasource with the name and value arrays defined above
            wt.SetDataSource(arrDayBlocks, arrNames);

            // Process the template
            wt.Process();

            // Save the document to the desired location
            string docName = String.Format("Calendar-{0}-{1}_output.docx", this.month,
this.year);
            wt.Save(@"..\..\WordOutputFiles\"+docName);
        }
```

```
    }
```

Downloads

Template: CalendarTemplate.docx

Output: Calendar-7-2013_output.docx