Data Validation

Intro

Define and apply data validation rules to cells with ExcelApplication.

Excel Data Validation (in Excel the menu, Data > Validation...) allows you to control what kind of data can be entered into cells by creating validation rules. You can define and apply these rules with ExcelApplication.

Code

```
  public class DataValidationDemo
    {

        // Declare an ExcelApplication object, workbook and a worksheet. The document
created
        will only have one worksheet.
        private ExcelApplication xlw;
        private Workbook wb;
        private Worksheet ws;
        // Previously selected by user in a webform
        private string borrowerName = "Sample J. Borrower";
        private string loanStartDate = "6/19/2013";
        private int Downpayment = 20;
        private double InterestRate = .05;
        private int PurchasePrice =  200000;
        private int LoanPeriod = 15;

        ///
        /// Build the report with ExcelApplication
        ///
        public void GenerateReport()
        {
            // Get path to template.
            //string templatePath ="Templates/FirstYearPayment.xlsx";

            // Create an instance of ExcelApplication,
            open the correct Workbook template based on loan period,
            and create one Worksheet.
            xlw = new ExcelApplication();
            wb = xlw.Open(@"..\..\ExcelTemplateFiles\FirstYearPayment.xlsx");
            ws = wb.Worksheets[0];

            // Insert some data into the template
            ws.Name = borrowerName;
            ws["A2"].Value = "Prepared for " + borrowerName;
            ws["A3"].Value = "Generated on " +
System.DateTime.Now.ToShortDateString();
            ws["C5"].Value = Convert.ToInt32(PurchasePrice);
            ws["C6"].Value = Convert.ToInt32(Downpayment) //.01;
            ws["C10"].Value = Convert.ToDouble(InterestRate);
            ws["C11"].Value = loanStartDate;
            ws["C9"].Value = Convert.ToInt32(LoanPeriod) //12;

            // Add data validation.
```

```
            //Define the type of validation you want (in this case, List),
            //and specify the valid values in an array.

            int[] purchasePriceValues = { 200000, 250000, 300000, 350000, 400000 };
            SoftArtisans.OfficeWriter.ExcelWriter.DataValidation dvPurchasePrice =

wb.CreateDataValidation(SoftArtisans.OfficeWriter.ExcelWriter.DataValidation.Validatio
nType.List, purchasePriceValues);

            // After you define your data validation rule, apply it to a Cell,
             //Area, or Range

            ws["C5"].DataValidation = dvPurchasePrice;

            // Define data validation rules for percentages
            string[] downPaymentValues = { "20.00%", "25.00%", "30.00%", "35.00%",
"40.00%" };
            DataValidation dvDownpayment =
                wb.CreateDataValidation(DataValidation.ValidationType.List,
downPaymentValues);
            ws["C6"].DataValidation = dvDownpayment;

            string[] intRateValues = { "5.00%", "6.00%", "7.00%", "8.00%", "9.00%",
"10.00%" };
            DataValidation dvInterestRate =
                wb.CreateDataValidation(DataValidation.ValidationType.List,
intRateValues);
            ws["C10"].DataValidation = dvInterestRate;

            // Define a date-based comparison rule.  The entered date must be on or
after
             //the current date.


            DataValidation dvFirstPayment =
                wb.CreateDataValidation(DataValidation.ValidationType.Date,
                    DataValidation.ComparisonType.GreaterThanOrEqualTo,
System.DateTime.Now);

            ws["C11"].DataValidation = dvFirstPayment;

            // Save the report by streaming it
            to the client's browser
            xlw.Save(wb, @"..\..\ExcelOutputFiles\DataValidation_output.xlsx");

    }
```

```
}
```

Downloads

- Template: FirstYearPayment.xlsx
- Output: DataValidation_output.xlsx