

Data Binding Properties

Intro

The `DataBindingProperties` object provides the means to change how data is imported using `ExcelTemplate`.

The `DataBindingProperties` object controls how the data is bound when the `BindData` method is called. The `DataBindingProperties` can control the maximum number of rows that are imported, whether the data is transposed and which worksheet the data is imported to.

In this sample, we use two `DataBindingProperties` objects-`topEmployeeProperties` and `allEmployeeProperties`-to create two different worksheets: one containing only the 10 employees with the most sales and one containing all the sales employees. By setting the `MaxRows` property to 10 with the `topEmployeeProperties` object, we are able to display only the top 10 sales people in the AdventureWorks database.

Code

```
public void GenerateReport()
{
    //Open the template file
    ExcelTemplate xlt = new ExcelTemplate();
    xlt.Open(@"..\..\ExcelTemplateFiles\BindingPropertiesTemplate.xlsx");

    //Get the data to bind to the template
    DataTable salesPeople = getSalesPeople();
    DataTable salesPeople2 = getSalesPeople();

    //We will bind the data in two places, using the same data source name.
    //In the "Top 10" worksheet we will place the 10 sales people who have
    //earned the most money from sales this year
    DataBindingProperties topEmployeesProperties =
xlt.CreateDataBindingProperties();
    topEmployeesProperties.MaxRows = 10;
    topEmployeesProperties.WorksheetName = "Top 10";

    //In the "Employees" worksheet, we will place all sales people
    DataBindingProperties allEmployeesProperties =
xlt.CreateDataBindingProperties();
    allEmployeesProperties.WorksheetName = "Employees";

    //Bind the data to the template. While the same data object cannot be
bound
    //in two different places, we can have data sources on two different
worksheets
    //that have the same name, so long as we specify which worksheet we want
to bind
    //to in the DataBindingProperties object.
    xlt.BindData(salesPeople, "SalesPerson", topEmployeesProperties);
    xlt.BindData(salesPeople2, "SalesPerson", allEmployeesProperties);

    //Process the template, and save it.
    xlt.Process();
    xlt.Save(@"..\..\ExcelOutputFiles\BindingProperties_output.xlsx");
}

private DataTable getSalesPeople()
{
    DataTable dt = new DataTable();
```

```
dt.Columns.Add("FirstName", typeof(string));
dt.Columns.Add("LastName", typeof(string));
dt.Columns.Add("SalesYTD", typeof(double));
dt.Rows.Add("Linda", "Mitchell", 5200475.231);
dt.Rows.Add("Jae", "Pak", 5015682.231);
dt.Rows.Add("Michael", "Blythe", 4557045.0459);
dt.Rows.Add("Jillian", "Carson", 3857163.6332);
dt.Rows.Add("Ranjit", "Varkey Chudukatil", 3827950.238);
dt.Rows.Add("David", "Campbell", 3587378.426);
dt.Rows.Add("Jose", "Saraiva", 3189356.247);
dt.Rows.Add("Shu", "Ito", 3018725.486);
dt.Rows.Add("Tsvi", "Reiter", 2811012.715);
dt.Rows.Add("Rachel", "Valdez", 2241204.042);
dt.Rows.Add("Tete", "Mensa-Annan", 1931620.184);
dt.Rows.Add("Garrett", "Vargas", 1764938.986);
dt.Rows.Add("Lynn", "Tsoflias", 1758385.926);
dt.Rows.Add("Stephen", "Jiang", 677558.4653);
dt.Rows.Add("Amy", "Alberts", 636440.251);
dt.Rows.Add("Syed", "Abbas", 219088.8836);
dt.Rows.Add("Pamela", "Ansman-Wolfe", 0);
return dt;
}
```

```
}
```

Downloads

- Template: [BindingPropertiesTemplate.xlsx](#)
- Output: [BindingProperties_output.xlsx](#)