Part 1 - Importing Data

Table of Contents

Setting up the Template

About templates and data markers

A PowerPointWriter **template** is a PowerPoint presentation that contains PowerPointWriter **data markers**. A **data marker** is a line of text beginning with `%%=` that specifies a database column, variable, or array to insert into the presentation. PowerPointWriter supports data markers embedded in text. Data markers are added to a presentation in PowerPoint and then bound to data sources in code. PowerPointWriter populates the data markers with values from the data sources when the code is executed. Part 1 of this tutorial demonstrates how to use data markers to import single rows of data.

Data marker syntax

The basic syntax for a data marker is `%%=[DataSourceName].[ColumnName]`, where `DataSourceName` is the name of the data source and `ColumnName` is the name of the column in the data source. You need to follow these rules when naming data markers:

- Names must begin with a letter (A-Z, a-z)
- The brackets (`[ ]`) are optional, but must be used when the data marker contains spaces or Unicode characters
    - The following is a list of characters allowed in data marker names without brackets: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890_
- Names must exactly match the names in the data source.
    - If a column in a database is 'Street Address', the `ColumnName` must be `[Street Address]` to account for the space. Similarly, if a data source name is "DataSource1", the data marker name must be `DataSource1` or `[DataSource1]`.

For more specific information about creating data markers, see How to use Data Markers

Adding data markers to the the template

The template for Part 1 consists of 2 slides populated with text and data markers.

> ⓘ **Following the Sample Code**
> In the downloadable PowerPointWriter_BasicTutorials.zip under *ProjectProposal*, there is a completed template file located in *ProjectProposal/templates/part1_template.pptx*.

1. Open Microsoft PowerPoint and start with a blank .pptx file. Sace the file as part1_template.pptx.

2. (optional) Set the presentation theme by going to the Design tab and selecting on the the built-in themes.

3. A title slide should have been added to the presentation by default. If not, insert a title slide now.
This slide will hold our data markers for the employee name, company logo.

4. In the title text box add the text 'New Employee Training'.

5. Create a new text box underneath the title. This will hold our data marker for our employee name.

6. Place the text 'Welcome,' and the data marker **%%=Employee.Name** in this text box.
The data marker indicates where the data in the 'Employee' column from the 'Company' data source will be placed.

Our slide so far should look something like the one below:
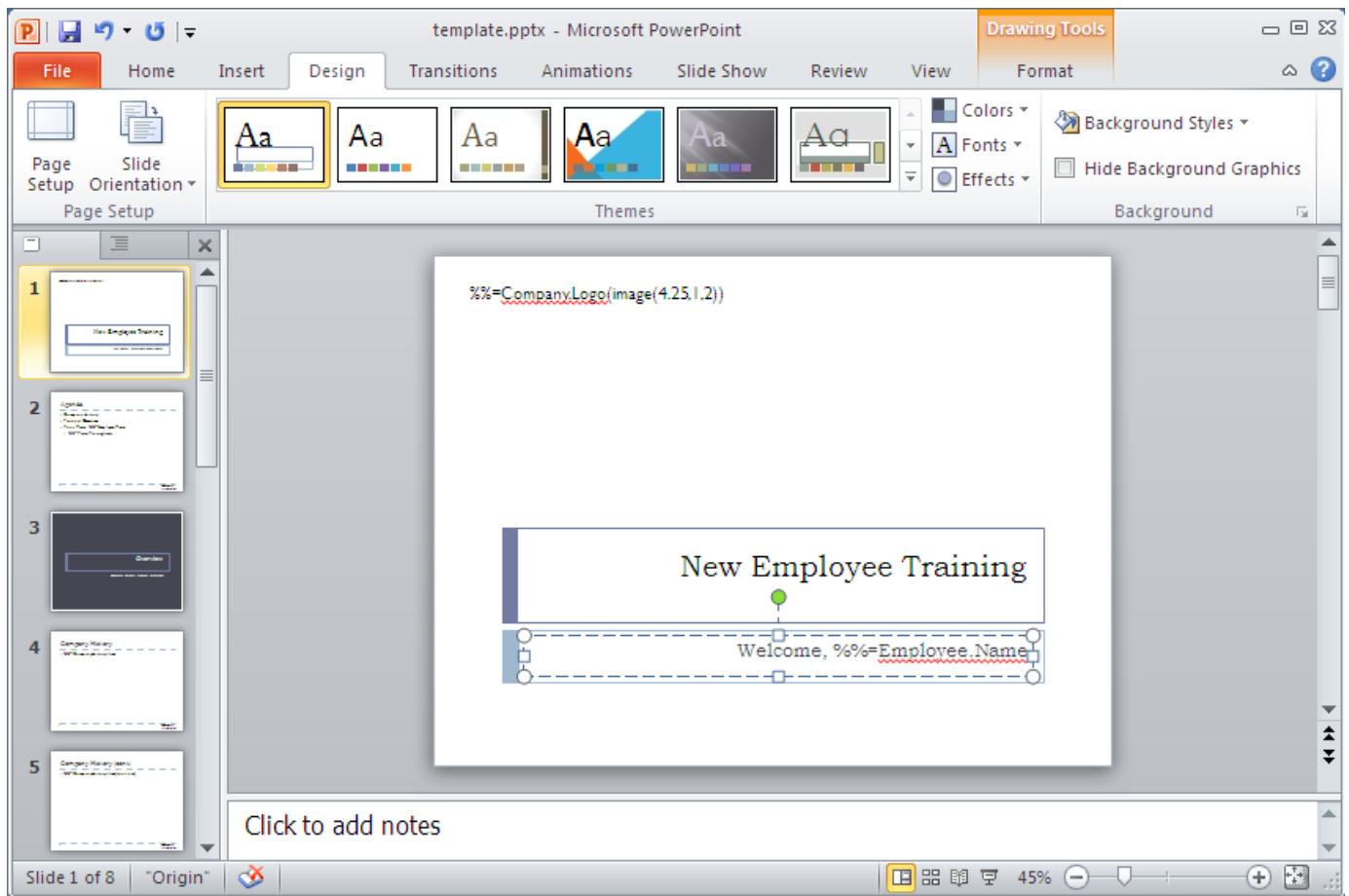
Using the image modifier

Next we are going to add a data marker that holds the company logo to our slide. To place an image using a data marker into your presentation, the data marker must include the image modifier.

7. Add a text box to the upper left hand corner of the presentation. This will hold our company logo data marker.

8. Place the data marker **%%=Company.Logo(image(4.25,1,2))** in the text box.

We are using the image modifier with 3 parameters. The first and second parameters are the width and height of the image respectively. All dimensions are in inches.

The third parameter is the scaling mode, which tells PowerPointWriter how to scale the image. In this example we are using scaling mode 2, meaning that the image will be enlarged or shrunk to fit within the bounds specified, but it will keep its natural aspect ratio.

For more information on scaling modes and the image modifier see the Quick Guide Importing Images.

Our first slide is now complete. It should look similar to the one below.

%%=Company.Logo(image(4.25,1,2))

New Employee Training

Welcome, %%=Employee.Name

Our next slide will contain an agenda for the rest of the training presentation.

9. Create a new content slide. The default content slide will consist of two text boxes: one for the title and a large one for the body of the text.

10. Give this slide the title 'Agenda'.

11. Add a third text box in the bottom right of the slide to hold our company logo.

12. Place the data marker **%%=Company.Logo(image(2))** in this text box.
This time, our image modifier is only taking one parameter, the scaling mode. This means that the image will take on the dimensions of the text box containing the data marker and be scaled according to the chose mode. Therefore, the text box should be the desired size for the imported image.

Importing Multiple Rows of Data

The body of our slide is going to contain a list of contents for the remainder of the presentation. It will include a list of training items imported for the specific role that the employee is training for.

13. In the large text box, create a new list. This can be done from the *Home* tab under *Paragraph*.

14. Add three items to this list:

- 'Company History'
- 'Product Overview'
- 'Your Team:' followed by the data marker **%%=Employee.Team**

Our slide so far should look like the following:
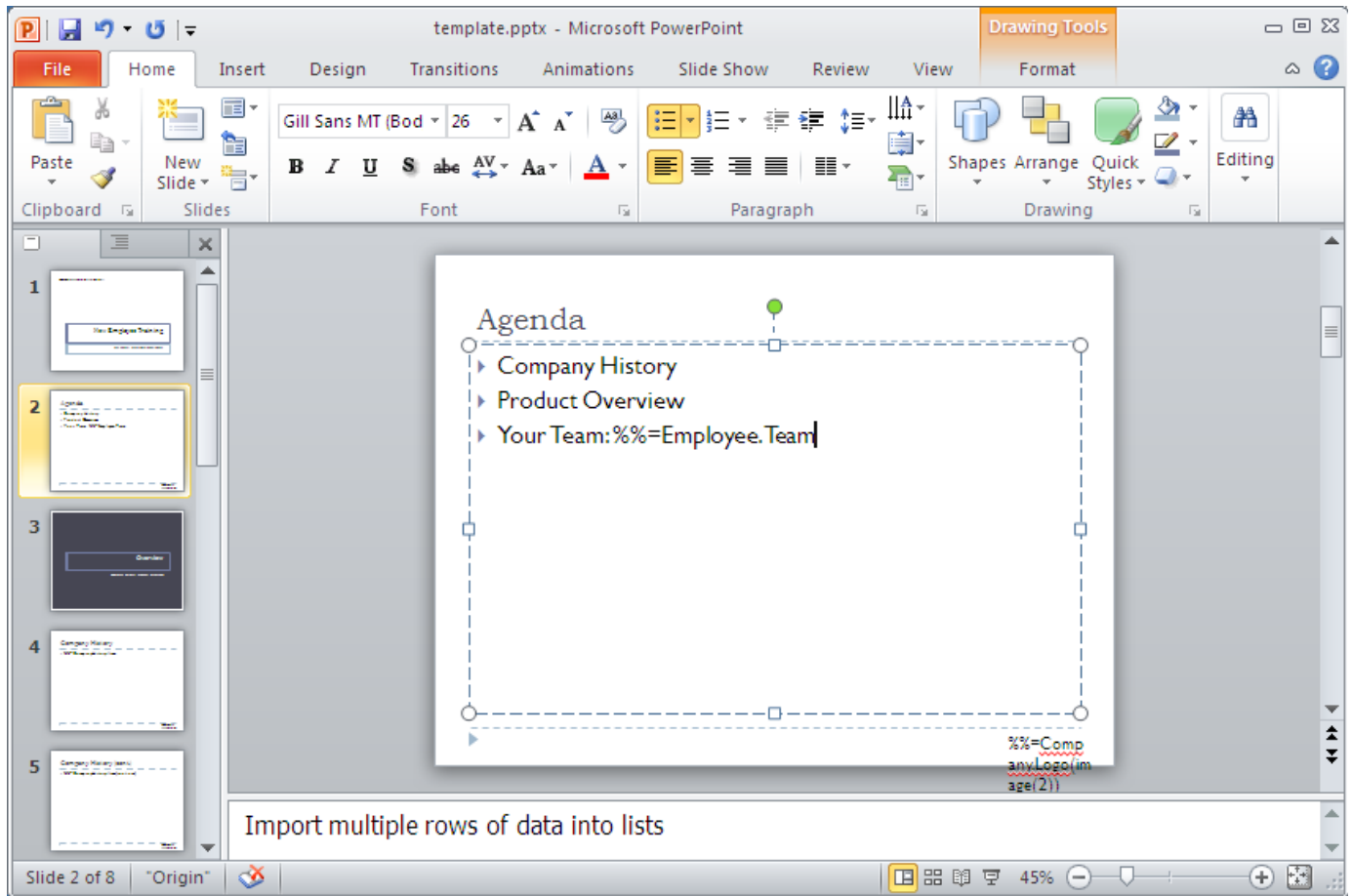
Under the 'Your Team' bullet, we are going to import a list of training items based on the position that the new employee was hired for. This will require that we import multiple rows from our data source. To import multiple rows in PowerPointWriter, the data marker must be placed in a list entry or table row. see Importing Multiple Rows of Data for more information.

15. Create a new list under the 'Your Team' bullet.

16. Add the data marker **%%=Team.TrainingItems** as the first entry in the new list.
The entirety of the 'TrainingItems' column will be imported into this list.

This completes our second slide. The completed slide is shown below.

Adding a PowerPointWriter Reference in Visual Studio

> ⓘ **Following the Sample Code**
> In the sample code, the reference to *SoftArtisans.OfficeWriter.PowerPointWriter.dll* has already been added to the *ProjectProposal* project.

Create a .NET project and add a reference to the PowerPointWriter library.

1. Open Visual Studio and create a .NET project.
    - The sample code uses a web application.
2. Add a reference to SoftArtisans.OfficeWriter.PowerPointWriter.dll
    - SoftArtisans.OfficeWriter.PowerPointWriter.dll is located under **Program Files > SoftArtisans > OfficeWriter > dotnet > bin**

Writing the Code

> ⓘ **Following the Sample Code**
> There is a sample web application page `Part1.aspx` and code behind `Part1.aspx.cs` available in the *ProjectProposal/templates/part1_template.pptx* directory that shows the completed code.

1. Include the SoftArtisans.OfficeWriter.PowerPointWriter namespace in the code behind

```
using SoftArtisans.OfficeWriter.PowerPointWriter;
```

2. In the method that will create the presentation, instantiate the `PowerPointTemplate` object.

```
PowerPointTemplate pptt = new PowerPointTemplate();
```

3. Open the template file for the presentation.

```
pptt.Open(Page.MapPath("//templates//part1_template.pptx"));
```

4. Create a DatabindingProperties object for binding data to the template.

```
DataBindingProperties dataProps = pptt.CreateDataBindingProperties();
```

5. Create the 'Company' data source. When importing data with arrays, the values and column names are placed in separate arrays. This first array will contain the column names.

```
//The column names are the same as the data markers
 string[] columnNamesArray = {"Logo"};
```

6. Create a second array that contains the values for the data source. When importing an image, the image value must be placed in a byte array. This can be done by using the File.ReadAllBytes method. See Importing Images for more information.

```
Byte[] logoArray =
System.IO.File.ReadAllBytes((Page.MapPath("//data//logo.png")));
//Place the image byte[] into an object[] for the BindData call
object[] valuesArray = {logoArray};
```

7. Bind the 'Company data to the template. Make sure the data source name is the same as the name used in the template.

```
pptt.BindData(valuesArray, columnNamesArray, "Company", dataProps);
```

8. Create the 'Employee' data source. This contains the Employee's name and team

```
string[] employeeColNames = {"Name", "Team"};
object[] employeeValues = {"Amy Alberts", "Development"};
```

9. Bind the Employee data to the presentation

```
pptt.BindData(employeeValues, employeeColNames, "Employee", dataProps);
```

10. Get the data for the Team data source

To create the DataTable, we will call a helper method `GetCSVData` that parses the CSV files and returns a `DataTable` with the values.

```
DataTable teamData = GetCSVData((Page.MapPath("//data//Team.csv")));
```

11. Bind the Team data to the presentation

```
pptt.BindData(teamData, "Team", dataProps);
```

12. Call process to import the data into the template and save the file.

```
pptt.Process();
```

13. Stream the output in the response as an attachment

```
pptt.Save(Page.Response, "Part1_Output.pptx", false);
```

Final Code

```
PowerPointTemplate pptt = new PowerPointTemplate();
pptt.Open(Page.MapPath("//templates//part1_template.pptx"));
DataBindingProperties dataProps = pptt.CreateDataBindingProperties();
string[] columnNamesArray = {"Logo"};
Byte[] logoArray = System.IO.File.ReadAllBytes((Page.MapPath("//data//logo.png")));
object[] valuesArray = {logoArray};
pptt.BindData(valuesArray, columnNamesArray, "Company", dataProps);
string[] employeeColNames = {"Name", "Team"};
             object[] employeeValues = {"Amy Alberts", "Development"};
pptt.BindData(employeeValues, employeeColNames, "Employee", dataProps);
DataTable teamData = GetCSVData((Page.MapPath("//data//Team.csv")));
pptt.BindData(teamData, "Team", dataProps);
pptt.Process();
pptt.Save(Page.Response, "Part1_Output.pptx", false);
```

Downloads

You can download the code for the Basic PowerPointWriter Tutorials as a Visual Studio solution, which includes the Project Proposal.

- [PowerPointWriter_BasicTutorials.zip](#)

Next Steps

Continue on to