Part 2 - The Continue Modifier

Table of Contents

Intro

⚠ This is Part 2 of a 3-part tutorial series for the Employee Training scenario. It is recommended that you complete Part 1 - Importing Data before starting this section.

ⓘ **Following the Sample**
There is a downloadable PowerPointWriter_BasicTutorials.zip with completed templates and code. The completed example of the template is available under *EmployeeTraining/templates/part2_template.pptx*. The code for this part of the tutorial can be found in *Part2 .aspx.cs*.

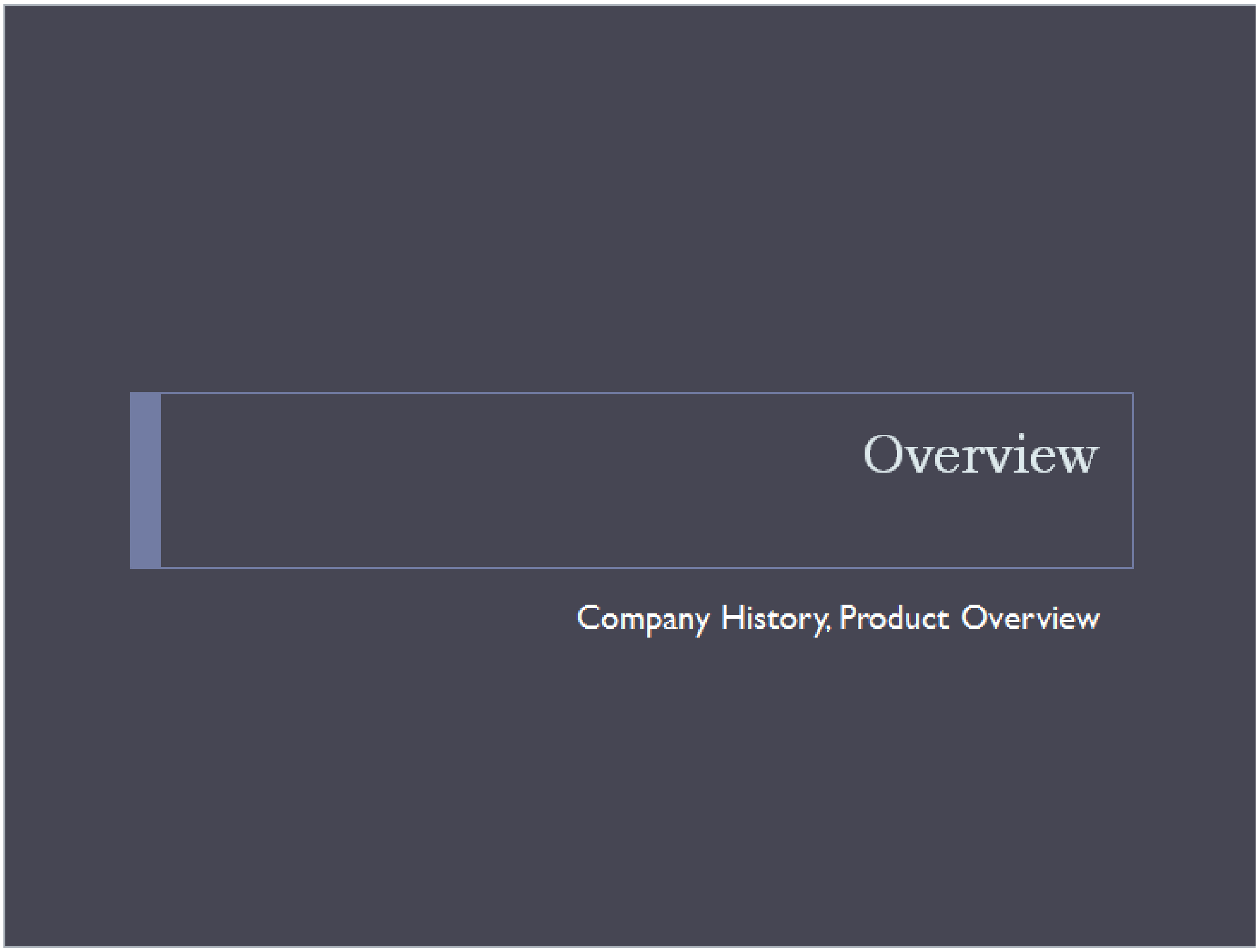
This part focuses on taking advantage of PowerPointWriter's *Continue* modifier.

Adding to the Template

Importing Multiple Rows

We are going to start with the template file as it was at the end of Part 1.

1. Add a new title slide to the presentation and give it the title 'Overview'. This slide is not going to to contain any data markers, just text. The 'Overview' slide in our template file is shown below.



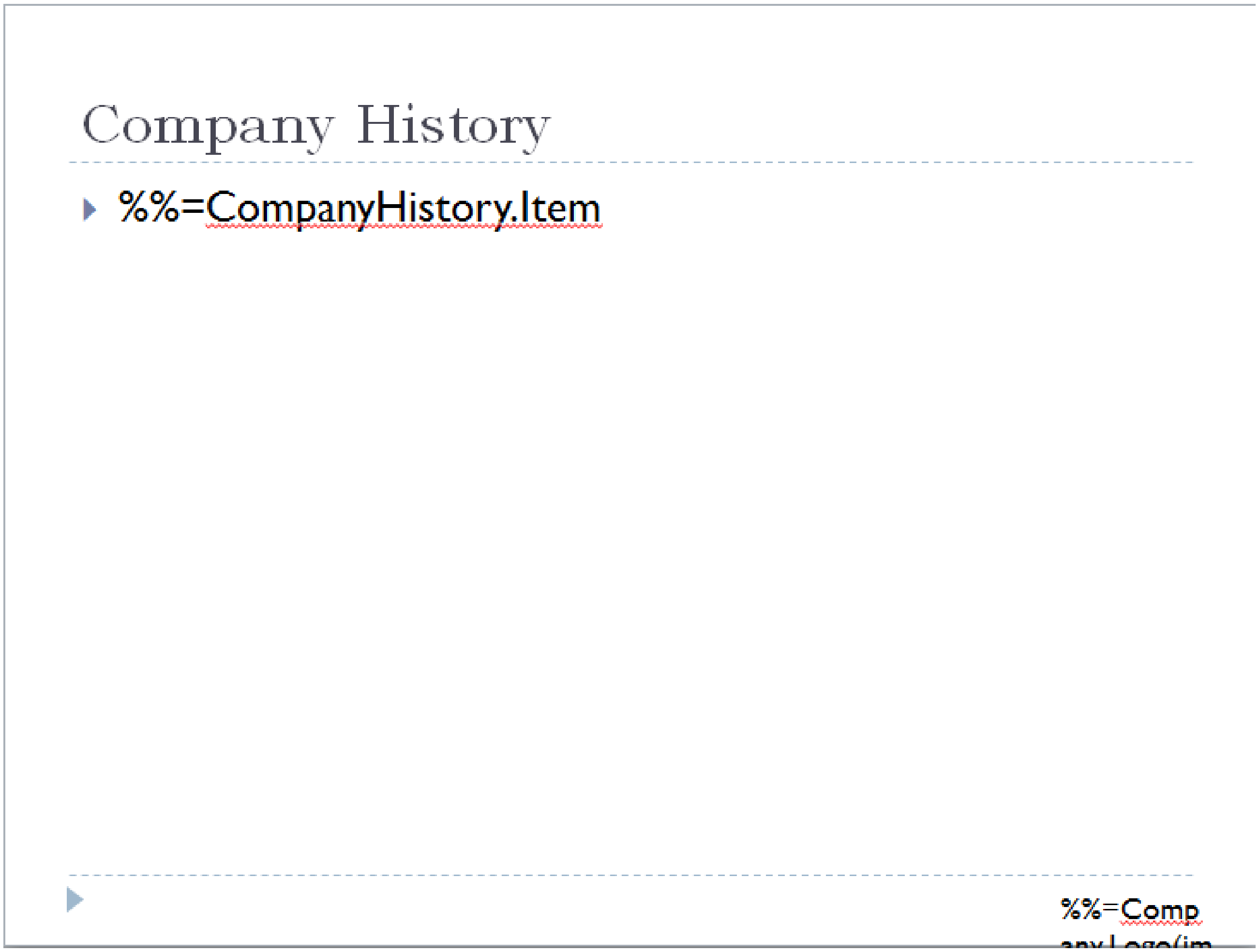
Overview

Company History, Product Overview

2. Add a new content slide with the title 'Company History'
3. Add the company logo to the bottom right corner of this slide by creating a text box and inserting the data marker *%%=Company.Logo(image(2)).

   In the body of this slide we are going to insert a list outlining the history of the company.

4. Create a new list. Add the data marker **%%=CompanyHistory.Item** as the first list item.



Company History

- %%=CompanyHistory.Item

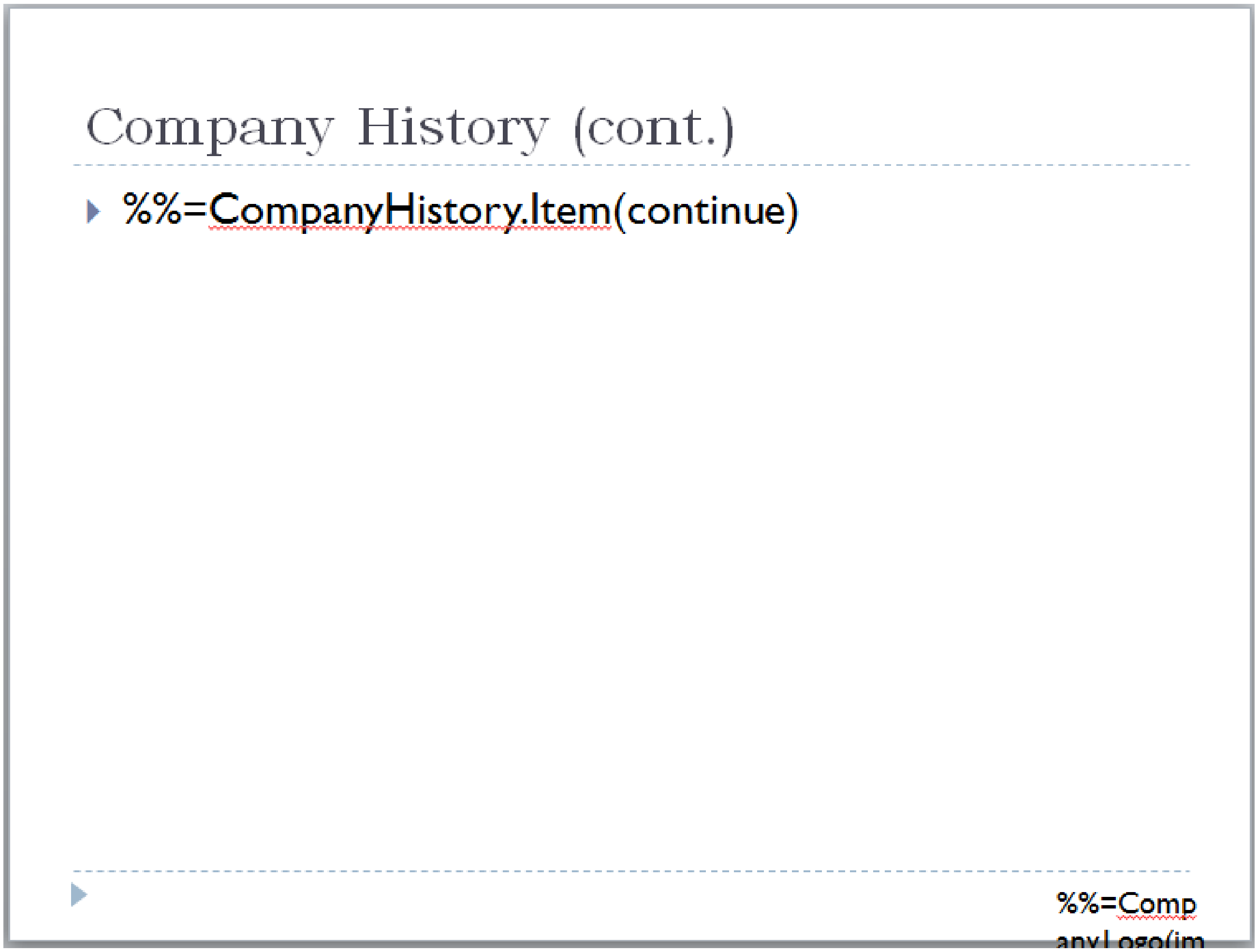
%%=Comp
anyLogo(im

Our company has a long history - too long to fit all the list items on a single slide. In order to keep the list from spilling off the slide, we can use PowerPointWriter's *Continue* modifier and MaxRowsPerSlide data binding property. The MaxRowsPerSlide data binding property specifies how many rows of data will be imported into each slide. This property will be set in the Writing the Code section. For more information about Data Binding Properties see the How to Use Data Binding Properties Quick Guide.

The Continue modifier offers a powerful way of determining how data is imported into a presentation. When importing data into a presentation, PowerPointWriter will stop importing onto a slide when the MaxRowsPerSlide limit is reached. However, if the same data marker appears in the presentation with the *Continue* modifier, the data will continue being imported to this data marker. If the same data marker appears in the presentation without the *Continue* modifier, the data will be imported from the beginning of the data source. For more information see Fitting Data on to Multiple Slides.

5. Create a new content slide. This time the title of the slide should be 'Company History (cont.)'.

6. Add a text box for the company logo in the bottom right. The data marker in this text box should be **%%=Company.Logo(image(2))**
7. Add a new list to the body of the slide.
8. The first item in this list will be our data marker with the continue modifier: **%%=CompanyHistory.Item(continue)**



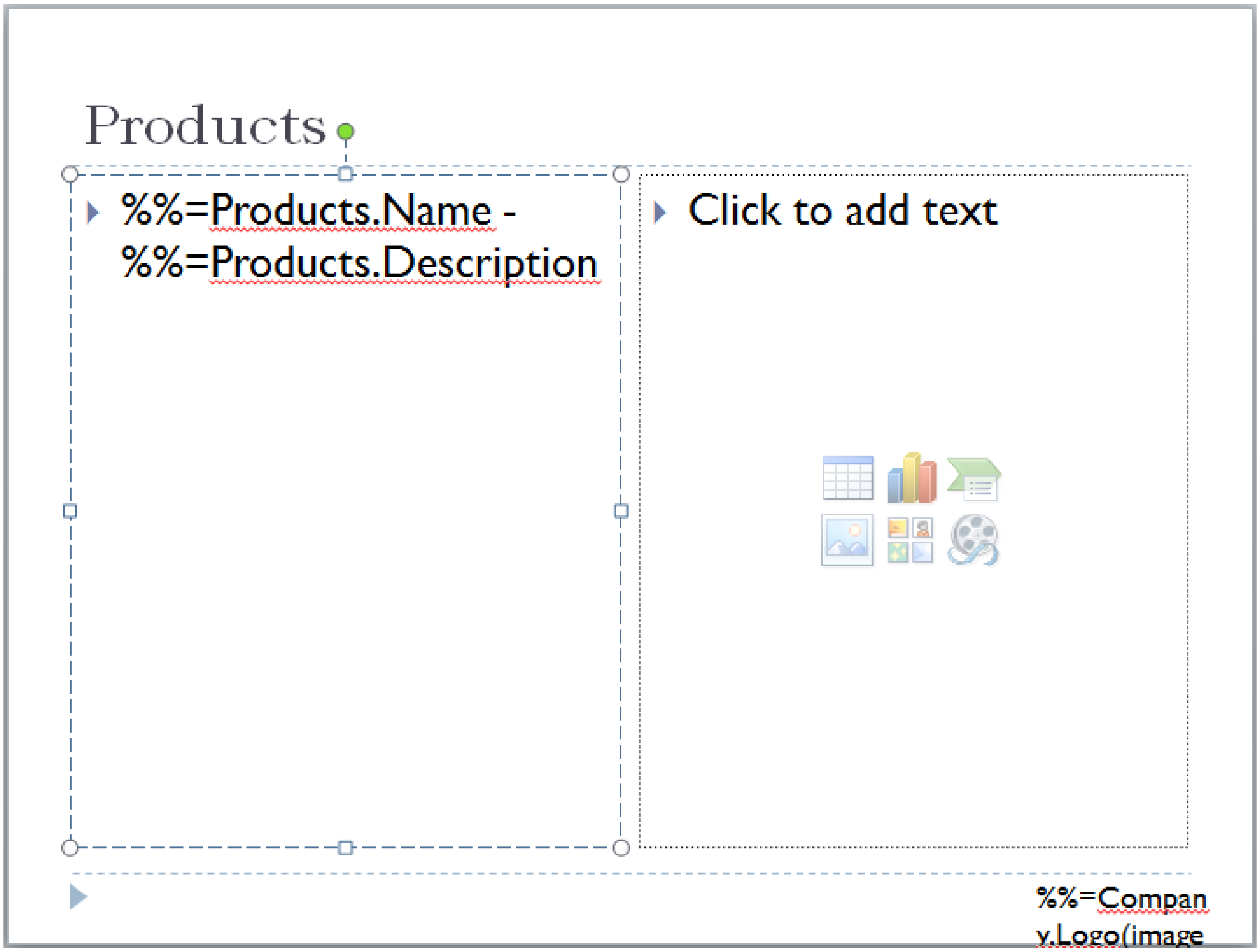
Company History (cont.)

- %%=CompanyHistory.Item(continue)

%%=Comp anyLogo(im

Now our entire company history will fit neatly in our presentation. Next we will show how the *Continue* modifier can be used to arrange data on a single slide. Rather than having our data continue onto a new slide, we will populate a second list.

9. Create a new content slide with the title 'Products'. Don't forget to add the company logo to the bottom right of the slide.
10. In the body of this slide create two text boxes side-by-side.
11. In the left text box add a list.
12. The first item in this list will contain two data markers. Because PowerPointWriter handles data markers embedded in text, we can easily display multiple columns of data together in our presentation.
13. Add '**%%=Products.Name - %%=Products.Description**' as the first item in our list. Our slide thus far should look like the following:



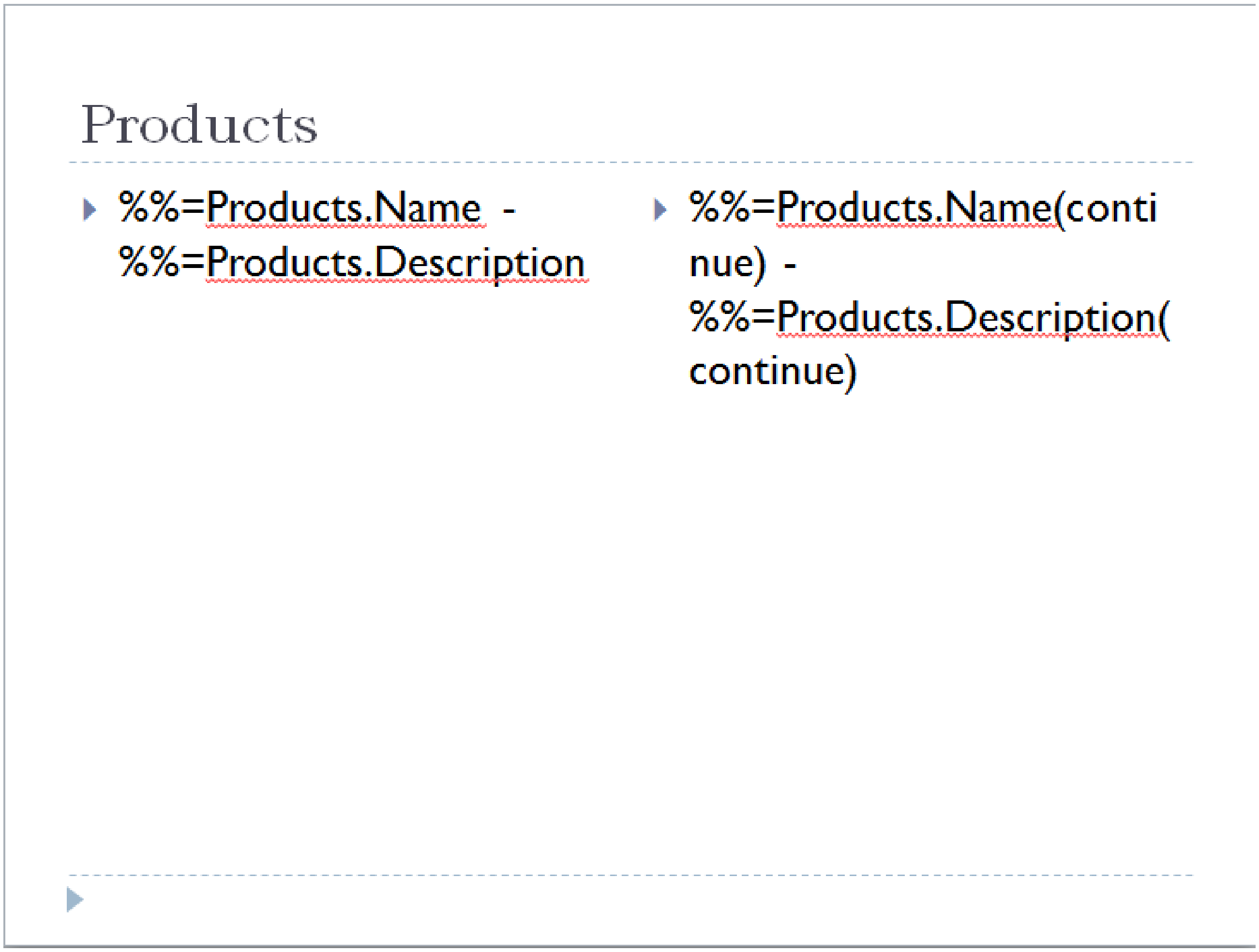
Products

%%=Products.Name - %%=Products.Description

Click to add text

%%=Company.Logo(image

14. Next, add a list to the right text box.
15. Insert '**%%=Products.Name(continue) - %%=Products.Description(continue)**' as the first item in the new list.
    Our completed slide is shown below:



Products

- %%=Products.Name - %%=Products.Description
- %%=Products.Name(continue) - %%=Products.Description(continue)

Writing the Code

We will reuse the code from Part 1 of this tutorial. We will take out the Process and Save calls and add our new code for Part 2.

```
//Code from part 1

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);
```

1. Get the data for the CompanyHistory.

Following the Sample
In the sample project, we are parsing CSV files with query results, rather than querying a live database. The CSV files are available

under the *data* directory. There is a copy of the CSV parser, `GenericParsing.dll` in the *bin* directory of the project `GetCSVData` is defined in *Part1.aspx.cs* in a `region` marked *Utility Methods*.

These calls are to a helper method `GetCSVData` that parses the CSV files and returns a `DataTable` with the values.

If you are following in your own project and would like to parse the CSV files as well, you will need to:

- Add a reference to `GenericParsing.dll`
- Include `GeneringParsing` at the top of your code.
- Add the `GetCSVData` method that can be found in the sample code.

```
DataTable dtCompanyHistory = GetCSVData(Page.MapPath("//data//COmpanyHistory.csv"));
```

2. Set the MaxRowsPerSlide property. We will use the DataBindingProperties object created in Part 1 of this tutorial. Only 5 Company History items will be on a single slide.

```
dataProps.MaxRowsPerSlide = 5;
```

3. Bind the CompanyHistory data to the presentation

```
pptt.BindData(dtCompanyHistory, "CompanyHistory", dataProps);
```

4. Get the data for the Products data source

```
DataTable dtProducts = GetCSVData(Page.MapPath("//data//Products.csv"));
```

5. Create a new DataBindingProperties object and set the MaxRowsPerSlide property to be 2

```
DataBindingProperties dataProps2 = pptt.CreateDataBindingProperties();
dataProps2.MaxRowsPerSlide = 2;
```

6. Bind the Products data source to the presentation

```
pptt.BindData(dtProducts, "Products", dataProps2);
```

7. Process the template to import the data

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

8.Save the final presentation and stream it in the response.

```
ppta.Save(pres, Page.Response, "Part2_Output.pptx", false);
```

Final Code

For information on writing this code, see Part 1 - Importing Data.

```
//Code from Part 1
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);
 //Code from Part 2
DataTable dtCompanyHistory = GetCSVData(Page.MapPath("//data//CompanyHistory.csv"));
dataProps.MaxRowsPerSlide = 5;
pptt.BindData(dtCompanyHistory, "CompanyHistory", dataProps);
DataTable dtProducts = GetCSVData(Page.MapPath("//data//Products.csv"));
DataBindingProperties dataProps2 = pptt.CreateDataBindingProperties();
dataProps2.MaxRowsPerSlide = 2;
pptt.BindData(dtProducts, "Products", dataProps2);
pptt.Process();
pptt.Save(Page.Response, "Part2_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 Part 3 - Copy Slide