Part 2 - Repeat Behavior

Table of Contents

Intro

⚠ This is Part 2 of a 3-part tutorial series for the Project Proposal scenario. It is recommended that you complete Part 1 - Getting Started 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 *ProjectProposal/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 repeat slide behavior.

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 content slide with the title 'Team Members'. This slide will contain data from the 'Team' data source about the proposed team members for the new project.

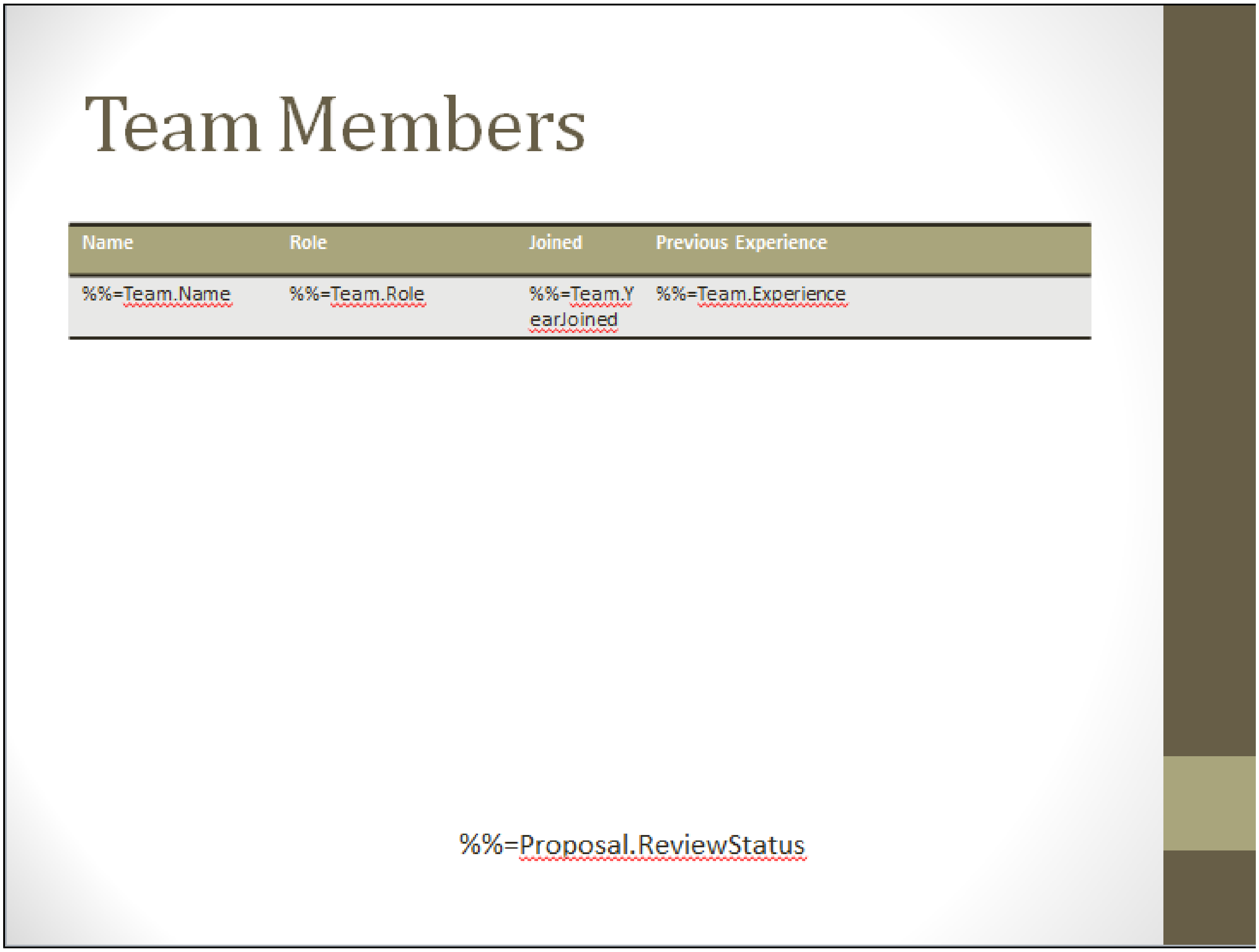
   In Part 1, our data source only contained a single row of data. However, the 'Team' data source contains multiple rows of data that we wish to import into the presentation. In order for multiple rows of data to be imported using a data marker, the data marker must be placed in a list entry or table row. The list or table will expand dynamically to hold all of the imported rows.
   For more information see Importing Multiple Rows of Data

2. On the 'Team Members' slide, add a new table with 2 rows and 4 columns by going to *Insert* > *Table* and choosing the appropriate size. The table will automatically be formatted based on the theme of your presentation.

3. In the first row of the table, add the headers **'Name'**, **'Role'**, **'Joined'**, and **'Previous Experience'**.

4. In the second row, add the following data markers under their corresponding headers:
   - %%=Team.Name
   - %%=Team.Role
   - %%=Team.YearJoined
   - %%=Team.Experience

5. To finish this slide, add the data marker %%=Proposal.ReviewStatus somewhere at the bottom of the slide. Your completed slide should look like the following:



Team Members

Name	Role	Joined	Previous Experience
%%=Team.Name	%%=Team.Role	%%=Team.YearJoined	%%=Team.Experience

%%=Proposal.ReviewStatus

Fitting Data on to Multiple Slides

Our next slide will contain a list of estimated costs for the project.

1. Create a new slide and give it the title 'Expected Costs'.

2. Add %%=Proposal.ReviewStatus to the bottom of the slide.

3. Add a table to this new slide with 2 rows and 3 columns. This table will contain data from the 'Cost' data source.

4. In the first row of the table, label the table columns **'Type'**, **'Item'**, and **'Amount'**.

5. In the second row, add the following data markers under their corresponding headers:

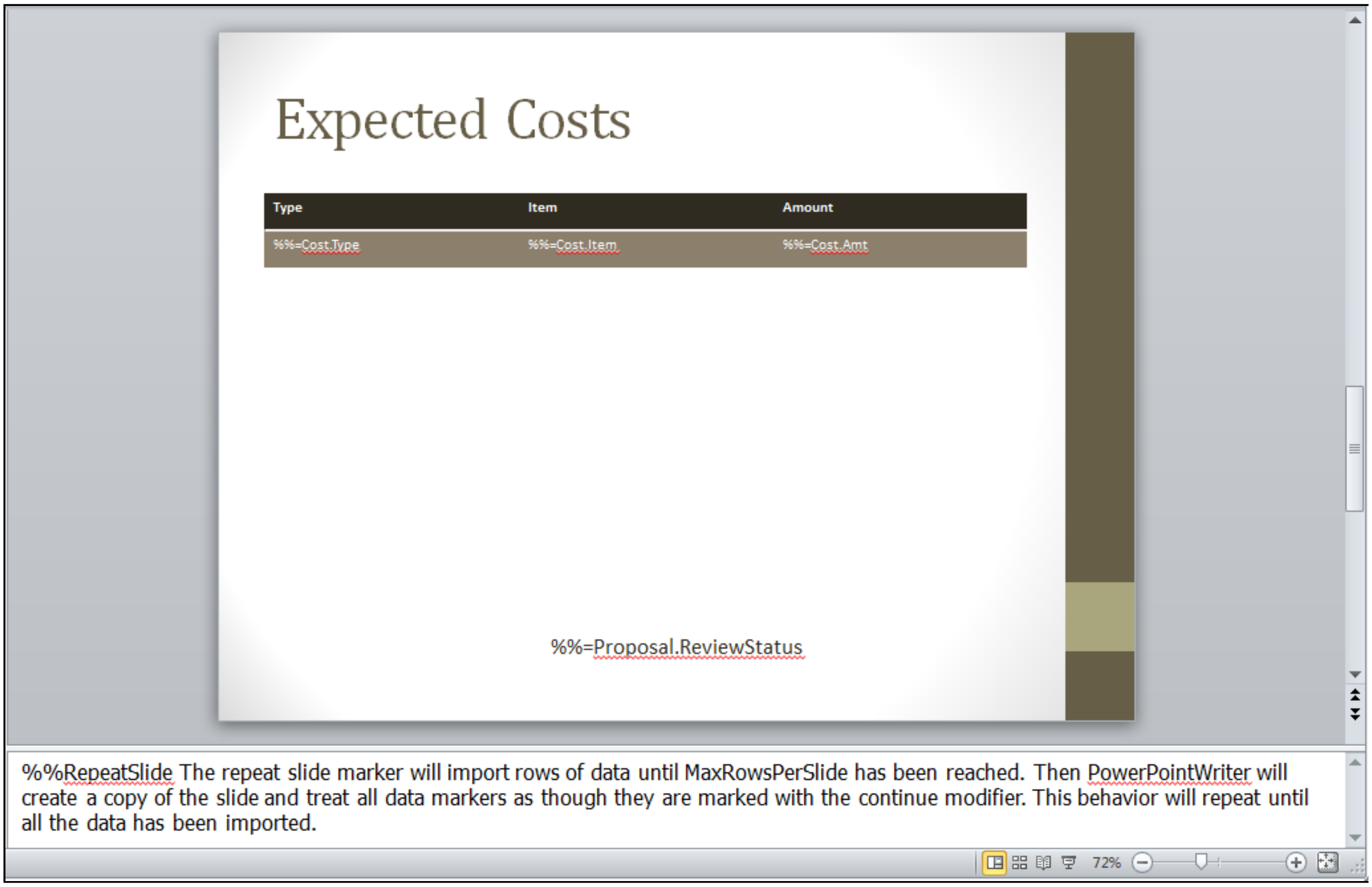
- %%=Cost.Type
- %%=Cost.Item
- %%=Cost.Amt.

The 'Cost' data source contains more data than can fit on to a single slide. In order to ensure that data does not fall off of a slide, the Data Binding Property MaxRowsPerSlide can be set. When importing data, PowerPointWriter will stop importing data onto a slide when it reaches the MaxRowsPerSlide limit.

Additionally, we want to be sure that all the data from the 'Cost' data source is imported into our presentation. To do this the **RepeatSlide** marker can be used in conjunction with the MaxRowsPerSlide property.
When placed as the first item in the notes section of the slide, the RepeatSlide marker will copy that slide in place and continue importing data on to the new slide. This behavior will continue until the end of the data source or the MaxRowsToImport limit is reached.
For more information see Fitting Data on to Multiple Slides.

6. To use the RepeatSlide marker, add the text '%%RepeatSlide' as the first string of text in the notes section of the 'Expected Costs' slide. Make sure there is a space between %%RepeatSlide and the next item of text. The completed slide should look like the one below. Note the

RepeatSlide marker in the notes section of the slide.



Expected Costs

Type	Item	Amount
%%=Cost.Type	%%=Cost.Item	%%=Cost.Amt

%%=Proposal.ReviewStatus

%%RepeatSlide The repeat slide marker will import rows of data until MaxRowsPerSlide has been reached. Then PowerPointWriter will create a copy of the slide and treat all data markers as though they are marked with the continue modifier. This behavior will repeat until all the data has been imported.

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();
object[] valuesArray = { "Project Name", "Project Date", "Review", "Leader Name",
"Start Date", "Cost Estimate", "Project Summary" };
string[] columnNamesArray = {"Name", "Date", "ReviewStatus", "Leader", "Start",
"Estimate", "Summary"};
pptt.BindData(valuesArray, columnNamesArray, "Proposal", dataProps);
```

1. Get the data for the Team and Cost data sets.

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.

```
DataTable dtTeam = GetCSVData(Page.MapPath("//data//Part1_Team.csv"));
DataTable dtCost = GetCSVData(Page.MapPath("//data//Part1_Cost.csv"));
```

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.

2. Set the MaxRowsPerSlide property to be 10.

```
dataProps.MaxRowsPerSlide = 10;
```

3. Bind the Team and Cost data sets to the template

```
pptt.BindData(dtTeam, "Team", dataProps);
pptt.BindData(dtCost, "Cost", dataProps);
```

4. Process the data

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

6. Use the Slides.Delete method to delete the last slide. It takes an integer representing the index of the slide to delete as a parameter.

```
pres.Slides.Delete(numSlides - 1);
```

7.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 - Getting Started.

```
//Code from Part 1
PowerPointTemplate pptt = new PowerPointTemplate();
pptt.Open(Page.MapPath("//templates//part1_template.pptx"));
DataBindingProperties dataProps = pptt.CreateDataBindingProperties();
object[] valuesArray = { "Surgery Schedule", "January 1, 2014", "Under Review",
"Pamela Blythe", "January 1, 2015", "6 months", "Software to schedule surgeries that
require multiple resources such as surgeons, assistants, nurses, pre-op and post-op
space, surgery theater, long-term recovery rooms." };
string[] columnNamesArray = { "Name", "Date", "ReviewStatus", "Leader", "Start",
"Estimate", "Summary" };
//Code from Part 2
DataTable dtTeam = GetCSVData(Page.MapPath("//data//Team.csv"));
DataTable dtCost = GetCSVData(Page.MapPath("//data//Cost.csv"));
dataProps.MaxRowsPerSlide = 10;
pptt.BindData(dtTeam, "Team", dataProps);
pptt.BindData(dtCost, "Cost", dataProps);
pptt.Process();
ppta.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 - Passing to PowerPointApplication