# <%=Model.Title%>

## Introduction

You can specify a view model when you create a document:

var document = DocumentFactory.Create(viewPath, model);

document.Generate(documentPath);

The specified model will be used for every generated document. If you want to generate the same document multiple times but with different models, specify a model when calling Generate:

document.Generate(documentPath, model);

The assembly containing the model will automatically be referenced. Also, the namespace of the model will be imported automatically.

From the view (i.e. this Word document), the model can be accessed through the Model property. This property is strongly typed, so you can access all fields/properties/methods without casting. In this example, the model is of type Model.Models.MyViewModel and looks like this:

public class MyViewModel

{

public string Title { get; set; }

public string Date { get; set; }

public List<Country> Countries { get; set; }

public DateTime? AverageDateProclaimed { get; set; }

public int? AveragePopulation { get; set; }

}

Because we’ll be using the extension method MyToString, we need to import its namespace using the Import directive.

<%@ Import Namespace="Model" %>

## DoSomething()

Now let’s do something with our model.

<% if (Model.Countries.Count > 0) { %>

The following table consists of <%= Model.Countries.Count + 2 %> rows.

<% foreach (var c in Model.Countries) {! %>

|  |  |  |
| --- | --- | --- |
| **Country** | **Population** | **Date proclaimed** |
| <%=c.Name%> | <%=c.Population / 1000000%> M | <%= c.DateProclaimed.MyToString() %><% AppendRow(); %> |
| *Average* | *<%= Model.AveragePopulation / 1000000 %> M* | *<%= Model.AverageDateProclaimed.MyToString() %>* |

*Source: Wikipedia.org, 2017.*

<% } %>  
<% } %>

It’s also possible to generate paragraphs instead of table rows, like this:

<% if (Model.Countries.Count > 0) { %>

<% foreach (var c in Model.Countries) { %>

**Country**   
Name: <%= c.Name %>  
Population: <%= c.Population / 1000 %> K  
Date proclaimed: <%= c.DateProclaimed.MyToString() %>

<% } %>

**Average population:** <%= Model.AveragePopulation / 1000 %> K  
**Average date proclaimed:** <%= Model.AverageDateProclaimed.MyToString() %>

<% } %>

Document generated on <%= Model.Date %> in CLR runtime version <%= Environment.Version %>.

## ImageBase64Encoded:

### Base64 encoded image, default extension=jpg:

**<%ImageBase64Encoded(Model.Image64);%>**

### Base64 encoded image, jpg extension specified:

**<%ImageBase64Encoded(Model.Image64, extension: "jpg");%>**

### Base64 encoded image, percentage specified:

**<%ImageBase64Encoded(Model.Image64, percentage: 20);%>**

### Base64 encoded image, jpg extension and percentage specified:

**<%ImageBase64Encoded(Model.Image64, extension: "jpg", percentage: 20);%>**