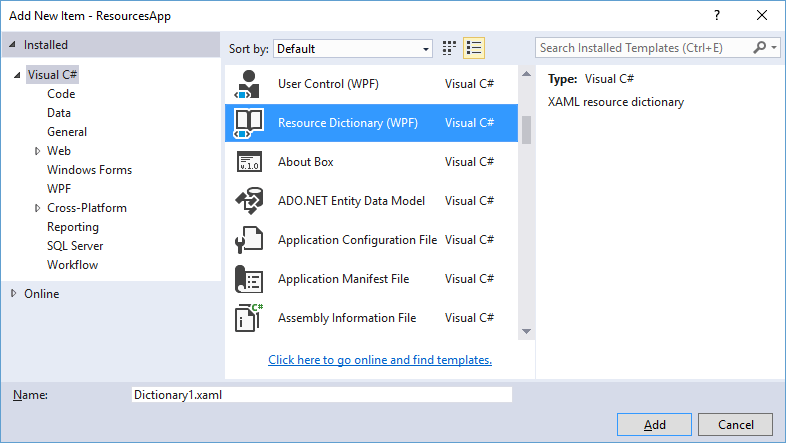
## Словари ресурсов

Мы можем определять ресурсы на уровне отдельных элементов окна, например, как ресурсы элементов Window, Grid и т.д. Однако есть еще один способ определения ресурсов, который предполагает использование **словаря ресурсов**.

Нажмем правой кнопкой мыши на проект и в контекстном меню выберем **Add -> New Item...**, И в окне добавления выберем пункт **Resource Dictionary (WPF)**:



Оставим у него название по умолчанию Dictionary1.xaml и нажмем на кнопку OK.

После этого в проект добавляется новый файл. Он представляет собой обычный xaml-файл с одним корневым элементом *ResourceDictionary*:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"                      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"                      xmlns:local="clr-namespace:ResourcesApp">    </ResourceDictionary> |

Изменим его код, добавив какой-нибудь ресурс:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"                      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"                      xmlns:local="clr-namespace:ResourcesApp">      <LinearGradientBrush x:Key="buttonBrush">          <GradientStopCollection>              <GradientStop Color="White" Offset="0" />              <GradientStop Color="Blue" Offset="1" />          </GradientStopCollection>      </LinearGradientBrush>  </ResourceDictionary> |

После определения файла ресурсов его надо подсоединить к ресурсам приложения. Для этого откроем файл *App.xaml*, который есть в проекте по умолчанию и изменим его:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <Application x:Class="ResourcesApp.App"               xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"               xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"               xmlns:local="clr-namespace:ResourcesApp"               StartupUri="MainWindow.xaml">      <Application.Resources>          <ResourceDictionary>              <ResourceDictionary.MergedDictionaries>                  <ResourceDictionary Source="Dictionary1.xaml" />              </ResourceDictionary.MergedDictionaries>          </ResourceDictionary>      </Application.Resources>  </Application> |

Элемент ResourceDictionary.MergedDictionaries здесь представляет колекцию объектов ResourceDictionary, то есть словарей ресурсов, которые добавляются к ресурсам приложения. Затем в любом месте приложения мы сможем сослаться на этот ресурс:

|  |  |
| --- | --- |
| 1 | <Button Content="OK" MaxHeight="40" MaxWidth="80" Background="{StaticResource buttonBrush}" /> |

При этом одновременно мы можем добавлять в коллекцию ресурсов приложения множество других словарей или параллельно с ними определять еще какие-либо ресурсы:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <Application x:Class="ResourcesApp.App"               xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"               xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"               xmlns:local="clr-namespace:ResourcesApp"               StartupUri="MainWindow.xaml">      <Application.Resources>          <ResourceDictionary>              <ResourceDictionary.MergedDictionaries>                  <ResourceDictionary Source="Dictionary1.xaml" />                  <ResourceDictionary Source="Dictionary2.xaml" />                  <ResourceDictionary Source="ButtonStyles.xaml" />                  <SolidColorBrush Color="LimeGreen" x:Key="limeButton" />              </ResourceDictionary.MergedDictionaries>          </ResourceDictionary>      </Application.Resources>  </Application> |

### **Загрузка словаря ресурсов**

Нам необязательно добавлять словарь ресурсов через ресурсы приложения. У объекта ResourceDictionary имеется свойство Source, через которое мы можем связать ресурсы конкретного элемента со словарем:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <Window x:Class="ResourcesApp.MainWindow"          xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"          xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"          xmlns:d="http://schemas.microsoft.com/expression/blend/2008"          xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"          xmlns:local="clr-namespace:ResourcesApp"          mc:Ignorable="d"          Title="Ресурсы" Height="250" Width="300">      <Window.Resources>          <ResourceDictionary Source="Dictionary1.xaml" />      </Window.Resources>      <Grid>          <Button Content="OK" MaxHeight="40" MaxWidth="80" Background="{StaticResource buttonBrush}" />      </Grid>  </Window> |

Также мы можем загружать словарь динамически в коде C#. Так, загрузим в коде C# вышеопределенный словарь:

|  |  |
| --- | --- |
| 1 | this.Resources = new ResourceDictionary() { Source = new Uri("pack://application:,,,/Dictionary1.xaml") }; |

При динамической загрузке, если мы определяем ресурсы через xaml, то они должны быть динамическими:

|  |  |
| --- | --- |
| 1 | <Button Content="OK" MaxHeight="40" MaxWidth="80" Background="{DynamicResource buttonBrush}" /> |