

Rozdział 4.

Kafelki - zapowiedź

Jacek Matulewski

*Materiały dla Podyplomowego Studium Programowania i Zastosowania Komputerów,
sekcja Projektowanie i tworzenie aplikacji dla platformy .NET (pod patronatem Microsoft)*

Kolejnym elementem aplikacji, jakim się zajmiemy, jest jej wygląd na ekranie Start, czyli kafelek. Nie będziemy tego wątku na razie rozbudowywali – poświęcony mu będzie inny rozdział. Tu dość lakonicznie chciałbym tylko przedstawić ideę kontaktu z użytkownikiem w ten sposób.

1. Dwukrotnie kliknij poznany już wcześniej plik *Package.appxmanifest* zawierający manifest aplikacji. Przejdź na zakładkę *Visual Assets*.
2. W drzewie z lewej strony podokna widoczne są typy plików przechowujące obraz logo, małego logo. Jest tam również miejsce na szerokie logo (pozycja *Wide 310x150 Logo*), czyli obraz o rozmiarze 310×150 pikseli. Żaden plik nie jest jednak tu przypisany.
3. Przygotujmy zatem nowy plik, np. za pomocą systemowego edytora Paint, powiększając obraz z pliku *Kolory\Kolory\Assets\Logo.scale-100.png* i zapisując go w podkatalogu *Assets* pod nazwą *WideLogo.scale-100.png*.
4. Następnie wróćmy do edycji manifestu i po zaznaczeniu pozycji *Wide 310x150 Logo* kliknijmy przycisk z trzema kropkami przy obrazku ze znakiem wodnym *Scale 100*. Wybierzmy nowy plik.
5. Następnie wracamy do okna Visual Studio i w klasie *MainPage* definiujemy metodę widoczną na listingu 3.16.

Listing 3.16. Metoda ta wymaga zadeklarowania użycia dwóch przestrzeni nazw: *Windows.Data.Xml.Dom* i *Windows.UI.Notifications*;

```
//using Windows.Data.Xml.Dom;
//using Windows.UI.Notifications;

#region Kafelki
TileUpdater tu = TileUpdateManager.CreateTileUpdaterForApplication();

public void zmienWygladKafelka()
{
    XmlDocument xml = TileUpdateManager.GetTemplateContent(TileTemplateType.
TileWide310x150Text01);
    XmlNode wezelTekst = xml.GetElementsByTagName("text").First();
    wezelTekst.AppendChild(xml.CreateTextNode("Kolory:"));
    wezelTekst = xml.GetElementsByTagName("text").Item(1);
    wezelTekst.AppendChild(xml.CreateTextNode(SkladoweKoloruDec(Ustawienia.Kolor)));
    wezelTekst = xml.GetElementsByTagName("text").Item(2);
}
```

```

        wezelTekst.AppendChild(xml.CreateTextNode("HEX: " +
        SkladoweKoloruHex(Ustawienia.Kolor)));
        tu.Update(new TileNotification(xml));
    }
#endregion

```

6. Metodę tę należy uruchomić przed zamknięciem aplikacji, a więc z metod `Current_VisibilityChanged` i `Button_Click_1`. Pokazuje to listing 3.17.

Listing 3.17. Zmiana wyglądu kafelka przed zamknięciem lub ukryciem aplikacji

```

void Current_VisibilityChanged(object sender, Windows.UI.Core.VisibilityChangedEventArgs e)
{
    if (e.Visible == false)
    {
        Ustawienia.Zapisz();
        zmienWygladKafelka();
    }
}

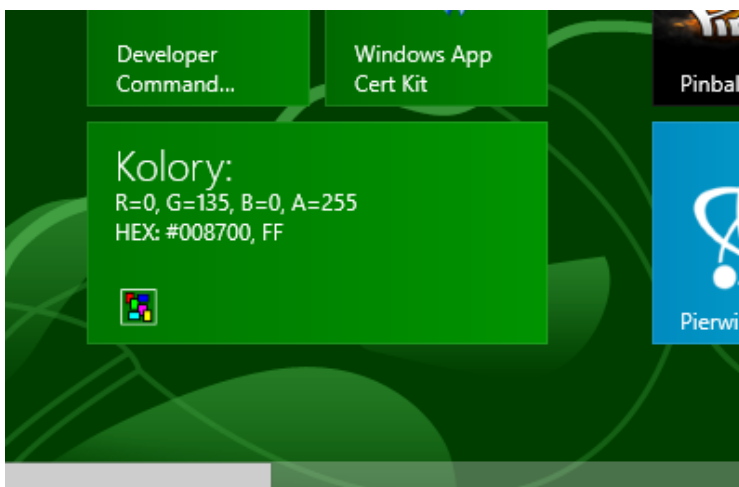
...

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    Ustawienia.Zapisz();
    zmienWygladKafelka();
    App.Current.Exit();
}

```

Użyty w tym przykładzie szablon kafelka to `TileTemplateType.TileWide310x150Text01`, czyli szeroki kafelek o rozmiarze 310×150 zawierający jedynie tekst. Przejdźmy do ekranu Start i zmieńmy rozmiar kafelka aplikacji. Należy kliknąć go prawym klawiszem myszy i wybrać pozycję *Zmień rozmiar*. W Windows 8.1 mamy do wyboru trzy rozmiary: mały, średni i szeroki. Wybieramy ten ostatni.

Po uruchomieniu aplikacji i jej zamknięciu, kafelek zmieni się wyświetlając składowe koloru wybranego za pomocą suwaków (rysunek 3.13).



Rysunek 3.13. Kafelki aplikacji z informacją o wybranych (i zapisanych w ustawieniach) składowych koloru