

УДК 004.5

ОСОБЕННОСТИ ИНТЕГРАЦИИ СЕРВИСА BING MAPS В ПРИЛОЖЕНИЕ WPF

Н.В. Карпенко¹, Д.С. Попов²

¹кандидат физико-математических наук, доцент кафедры электронных вычислительных машин, Днепропетровский национальный университет имени Олеся Гончара, г. Днепропетровск, Украина, e-mail: karpenko_nadija@mail.ru

²студент группы KI-13-1, кафедра электронных вычислительных машин, Днепропетровский национальный университет имени Олеся Гончара, г. Днепропетровск, Украина, e-mail: sonarnoir@gmail.com

Аннотация. В статье рассмотрены варианты интеграции онлайн карт, разработки пользовательских элементов управления, с подключением надстроек в приложение, созданное с использованием технологии Windows Presentation Foundation.

Ключевые слова: WPF, онлайн карты, элементы управления, слои, интеграция

PECULIARITIES IN BING MAPS SERVICES INTEGRATION INTO WPF-APPLICATION

Nadiia Karpenko¹, Dmitriy Popov²

¹ Ph.D. in Physics and Mathematical Sciences, Associate Professor, Department of Computer, Dnipropetrovs'k National University named by Oles Honchar, Dnipropetrovs'k, Ukraine, e-mail: karpenko_nadija@mail.ru

² student, Department of Computer, Dnipropetrovs'k National University named by Oles Honchar, Dnipropetrovs'k, Ukraine, e-mail: sonarnoir@gmail.com

Abstract. The article describes the various options for integrate online map, developing control API.

Keywords: WPF, online map, layers, integration, control API

Введение: В настоящее время существует множество картографических сервисов и поисково-информационных приложений, например, Яндекс.Карты, Рамблер-Карты, Карты Google и др. Для полноценной работы с этими сервисами необходимо подключение к сети Интернет, что не всегда осуществимо. К тому же хотелось бы иметь возможность добавления личных заметок (текста, медиа-контента и др.) к маршруту или месту на карте в любое удобное для пользователя время или, вообще, использовать карты как дополнение к приложению, основной функцией которого вовсе не является работа с картами.

В настоящее время наиболее интересной технологией для создания как автономных, так и запускаемых в браузере Windows-приложений

является Windows Presentation Foundation (WPF). Благодаря применению стилей и шаблонов, решена проблема настройки элементов управления, что позволяет создать приложение с насыщенным пользовательским интерфейсом [1]. Исходя из этого, хотелось бы рассмотреть особенности интеграции сервиса Bing Maps в приложение, созданное с использованием технологии WPF.

Цель работы: Интегрировать сервис в приложение WPF, разработать пользовательский интерфейс взаимодействия с приложением, использовать надстройки и слои для расширения функционала приложения.

Материалы и результаты исследования: Для работы с онлайн сервисами карт (технология Microsoft Bing Maps) необходимо выполнение ряда условий:

- ✓ наличие аккаунта разработчика Bing Map [2];
- ✓ наличие уникального ключа картографического сервиса – необходимо заполнить информацию о цели использования картографических сервисов, ссылки на приложение, выбрать тип ключа;
- ✓ наличие .NET Framework версии 4.0 и выше и Windows SDK.

Рассмотрим процесс интеграции картографического сервиса в стандартное приложение WPF. Для начала необходимо загрузить динамическую библиотеку классов Microsoft.Maps.MapControl.WPF.dll (поставляется в комплекте при установке Bing API) и добавить ссылку на библиотеку в проекте. Далее работаем с кодом XAML:

1. Следует сделать объявление:

```
xmlns:m="clr-namespace:Microsoft.Maps.MapControl.WPF;assembly=Microsoft.Maps.MapControl.WPF"
```

2. Добавить следующий код, чтобы поместить карту в поле Grid:

```
<Grid>  
  <m:Map x:Name="myMap" CredentialsProvider="INSERT_YOUR_BING_MAPS_KEY"  
  Mode="Aerial"/>  
</Grid>
```

где вместо фразы "INSERT_YOUR_BING_MAPS_KEY" следует указать ранее полученный ключ разработчика.

Теперь можно выбрать режим отображения карты. Существует три стандартных режима отображения: Road, Aerial, AerialWithLabels. Режим отображения можно указать в XAML коде приложения, например, Mode="AerialWithLabels", но гораздо практичнее будет объявить режим карты при инициализации приложения, что позволит в будущем изменять режим карты без вмешательства в код XAML:

```
public MainWindow()
```

```
{
```



```
InitializeComponent();  
myMap.Mode = new AerialMode(true);  
}
```

После проведенных манипуляций карта появилась в приложении (рис. 1), однако у нас еще нет интерфейса для взаимодействия с ней. Чтобы исправить это, необходимо создать пользовательский интерфейс и передать подключенной карте право захватывать нажатия клавиш. Это делается добавлением строки

```
myMap.Focus();
```

Теперь нажимая клавиши "+" и "-" карту можно приблизить и отдалить.



Рисунок 1 – Отображение карты в приложении WPF

Для реализации центрирования карты на определенные страны (города, улицы) необходимо передать следующие параметры:

1. Координаты – именно они задают условную точку центрирования;
2. Степень увеличения (зумирования);
3. Степень разворота карты (не используется).

Существует два способа реализации центрирования по заданной координате: через код XAML, либо через код C#:

Код XAML :

```
<Button x:Name="btnNewYork" Click="ChangeMapView_Click" Tag="40.7199,-  
74.0030,0.0000 12.0000" Margin="5">
```

```
<TextBlock>New York</TextBlock>  
</Button>
```

Код WPF:

```
private void ChangeMapView_Click(object sender, RoutedEventArgs e)  
{  
    string[] tagInfo ;  
    if (comboBox1.SelectedIndex == 0)  
    {  
        tagInfo = "48.4618,35.0224,0 16".Split(' ');  
        Location center = (Location)locConverter.ConvertFrom(tagInfo[0]);  
        double zoom = System.Convert.ToDouble(tagInfo[1]);  
        myMap.SetView(center, zoom);  
    }  
}
```

В переменную tagInfo можно поместить любую координату, после чего произвести центрирование по ней. Второй вариант более практичен и функционален.

С помощью BingMaps можно отображать различные фигуры на слое поверх карты, для этого необходимо указать начальную и конечную точку линии, а так же её толщину (в виде координат), например:

```
<m:MapPolyline Stroke="Blue" StrokeThickness="5"  
    Locations="47.6424,-122.3219 47.8424,-122.1747 47.5814,-122.1747  
47.67856,-122.130994"  
    Opacity="0.7"/>
```

Либо же можно использовать функцию, которая будет создавать линии в зависимости от выбранных точек, что опять же, будет более практично:

```
void addNewPolyline()  
{  
    MapPolyline polyline = new MapPolyline();  
    polyline.Stroke = new System.Windows.Media.SolidColorBrush(System.Windows.Media.Colors.Blue);  
    polyline.StrokeThickness = 5;  
    polyline.Opacity = 0.7;  
    polyline.Locations = new LocationCollection() {  
        new Location(47.6424, -122.3219),  
        new Location(47.8424,-122.1747),  
        new Location(47.67856,-122.130994)};  
    myMap.Children.Add(polyline);  
}
```

В поля new Location() можно передавать параметры, тем самым задавая точки, через которые будет проведена линия.

Для упрощения процесса создания ломаных линий, можно привязать задание точек к двойному клику мыши. Для этого необходимо создать событие: `MapWithPolygon_MouseDoubleClick` и описать функцию для «отлавливания» текущего положения курсора:

```
Point mousePosition = e.GetPosition(this);
```

и передачи координаты в функцию задания точки:

```
Location polygonPointLocation = MapWithPolygon.ViewportPointToLocation(mousePosition);
```

Сервис `BingMaps` поддерживает многослойность картографического приложения, значит можно использовать данную особенность для отображения медиа контента на слое поверх карты. Для этого необходимо привязать к пользовательскому событию, например клику по кнопке «Добавить», функцию добавления изображения. Следует обратить внимание на то, что можно указывать источник изображения в виде `Url` адреса изображения:

```
BitmapImage myBitmapImage = new BitmapImage();
myBitmapImage.BeginInit();
myBitmapImage.UriSource = new Uri("http://upload.wikimedia.org/wikipedia/commons/d/d4/Golden_Gate_Bridge10.JPG");
```

...

```
imageLayer.AddChild(image, location, position);
myMap.Children.Add(imageLayer);
```

Изображение будет размещено на дочернем слое. Каждый новый слой при перекрытии скрывает предыдущий, однако сохраняет его функционал.

Помимо изображений также можно добавлять видео:

```
MediaElement video = new MediaElement();
video.Source = new Uri(@"http://mschnline.vo.llnwd.net/d1/ch9/9/2/9/9/1/4/TCS2NBCOlympics_ch9.wmv",
```

```
UriKind.RelativeOrAbsolute);
```

Одной из востребованных функций является создание заметок прямо на карте. Как и в примерах выше, можно привязать функцию с захватом координаты в положении курсора к событию двойного щелчка мышью.

Для отображения, например, названий улиц (рис. 2), следует использовать надстройку (новый слой). Для этого необходимо подгружать надстройку к карте, а значит необходимо указать адрес к ней:

```
TileSource tileSource = new TileSource();
tileSource.UriFormat = "{UriScheme}://ecn.t0.tiles.virtualearth.net/tiles/r{quadkey}.jpeg?g=129&mkt=en-us&shading=hill&stl=H";
```

Теперь с картой можно полноценно работать.

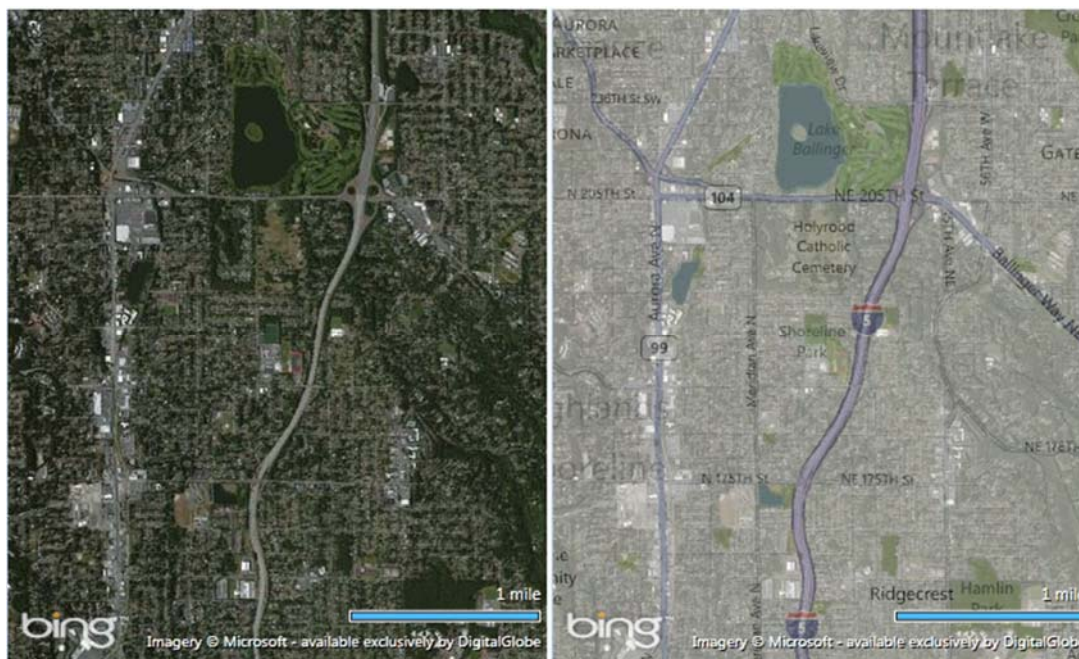


Рисунок 2 – Отображение названий улиц

Для нахождения местности по названию следует воспользоваться внешней надстройкой. Это означает, что поисковый запрос будет отправлен в виде строки, а надстройка будет возвращать результат поиска – координаты. Следует учесть то, что поисковый сервис работает лишь с региональным стандартом. Таким образом, в поисковом запросе необходимо заменить запятые на точки. Пример генерации запроса:

```
public XmlDocument Geocode(string addressQuery)
{
    string geocodeRequest = "http://dev.virtualearth.net/REST/v1/Locations/" + addressQuery + "?o=xml&key=" + BingMapsKey;
    XmlDocument geocodeResponse = GetXmlResponse(geocodeRequest);
    return (geocodeResponse);
}
```

Получение результата запроса:

```
System.Diagnostics.Trace.WriteLine("Request URL (XML): " + requestUrl);
HttpRequest request = WebRequest.Create(requestUrl) as HttpRequest;
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    if (response.StatusCode != HttpStatusCode.OK)
        throw new Exception(String.Format("Server error (HTTP {0}: {1}).",
            response.StatusCode,
            response.StatusDescription));
    XmlDocument xmlDoc = new XmlDocument();
```

```
xmlDoc.Load(response.GetResponseStream());
return xmlDoc;
}
```

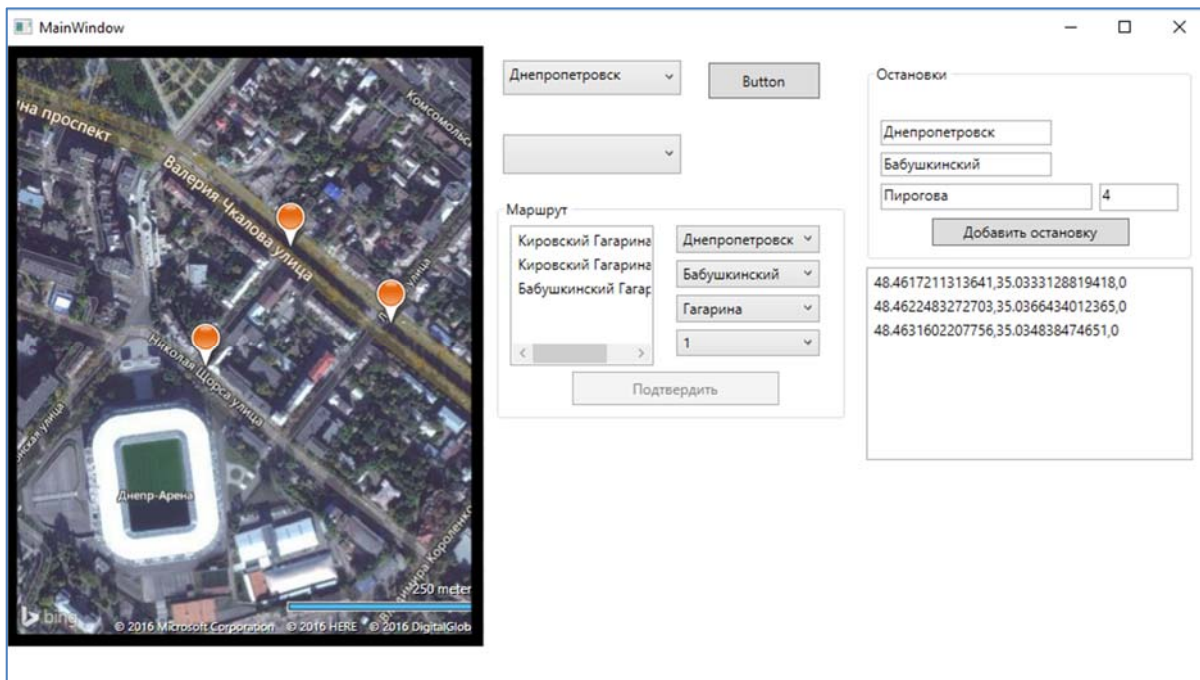


Рисунок 3 – Приложение WPF с интегрированным сервисом BING MAPS

Выводы: Выполняя вышеперечисленные действия можно интегрировать картографический сервис BING MAPS в WPF-приложение. С помощью данного приложения можно генерировать маршруты, производить поиск улиц/городов, создавать заметки к этим маршрутам, добавлять медиа контент.

ЛИТЕРАТУРА

1. Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 880 с.
2. Bing Maps Dev Center [Электронный ресурс]. – Режим доступа: <https://www.bingmapsportal.com/> — Загл. с экрана.
3. Техническая документация, поставляемая с пакетом BingMap API
4. Пользовательские элементы контроля [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/en-us/library/hh750210.aspx> — Загл. с экрана.