

# Instalacja środowiska

Wymagania:

1. system •Windows Vista lub •Windows 7 (~2.2GB).  
Dla studentów Politechniki Wrocławskiej dostępny jest on na stronie, np.  
[https://msdn62.e-academy.com/elms/Security/Login.aspx?campus=msdnaa\\_pd6979](https://msdn62.e-academy.com/elms/Security/Login.aspx?campus=msdnaa_pd6979). Dane do logowania do tego konta prawdopodobnie powędrowały na konto poczty studenckiej <http://student.pwr.wroc.pl> zaraz po tym, kiedy zostaliśmy studentami.  
Mogłoby się wydawać, że możliwe jest zainstalowanie systemu jako maszyna wirtualna korzystając z platform takich jak VMWare VirtualBox, Xen, Citrix czy KVM. Jednakże w dokumentacji wspomniane jest o tym, że maszyny wirtualne nie są mile widziane  
<http://msdn.microsoft.com/en-us/library/ff402530%28v=vs.92%29.aspx>.
2. •Zune, służąca do synchronizacji zawartości telefonu z komputerem. Można ją pobrać ze strony <http://www.zune.net/en-US/products/software/download/default.htm> (100MB).
3. •Windows Phone SDK: Windows Phone and App Hub  
<https://www.dreamspark.com/Product/Product.aspx?productid=26>. Pakiet Microsoft Visual Studio 2010 Express + Windows Phone Emulator + inne narzędzia. Więcej na:  
<http://msdn.microsoft.com/library/ff402523%28v=VS.92%29.aspx>.
4. aktywny telefon (umożliwia to narzędzie *Windows Phone Developer Registration*, które można domyślnie znaleźć w "C:\Program Files\Microsoft SDKs\Windows Phone\v7.1\Tools\Phone Registration\PhoneReg.exe")

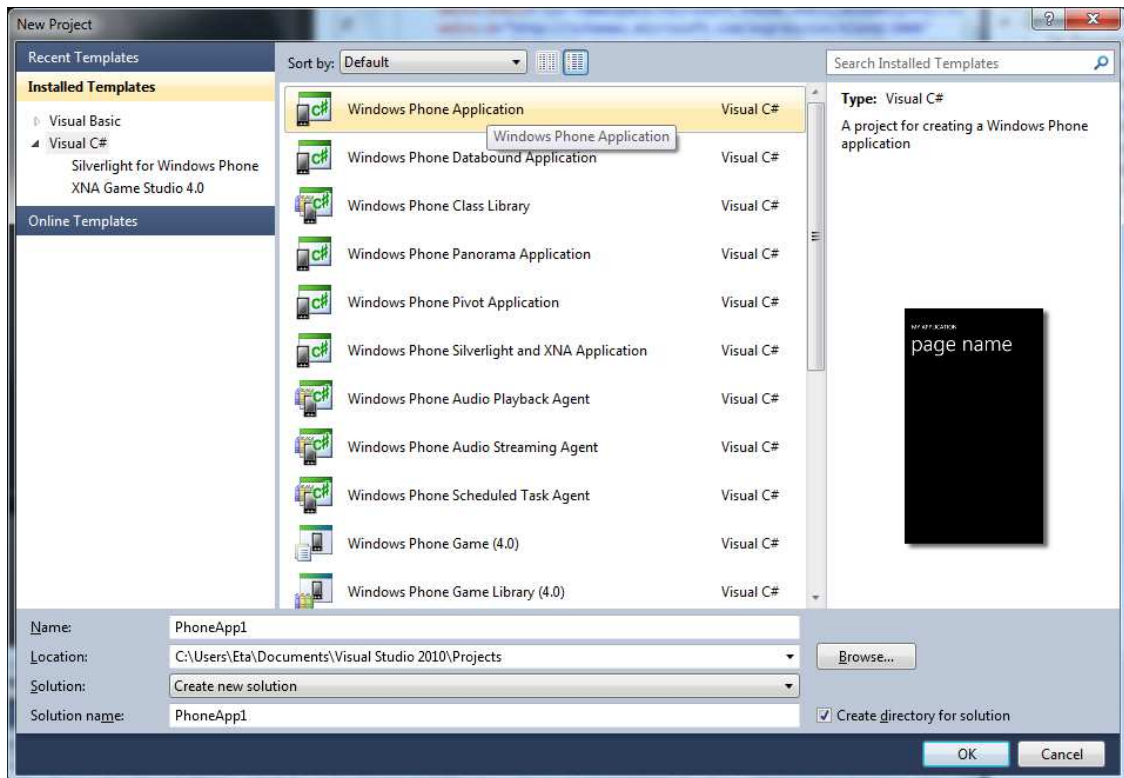
Alternatywnie można spróbować zainstalować Microsoft Visual Studio 2011.

## Pierwszy program. Contacts Serializer.

System Windows Phone 7 nie umożliwia wyeksportowania kontaktów do pliku tekstowego. Co prawda lista kontaktów synchronizuje się z <http://profile.live.com>, ale wymaga to połączenia z internetem. Celem jest program umożliwiający wyeksportowanie listy kontaktów do formy tekstowej, z którą później będziemy mogli co zrobić (np. wysłać pocztą elektroniczną czy smsem - wszystko dzięki możliwości kopiowania zawartości pola tekstowego).

W celu zapoznania się z systemem polecam uruchomienie i zrozumienie, i być może nawet zmodyfikowanie przykładowego *Walkthrough: Accessing Contact and Calendar Data for Windows Phone*  
<http://msdn.microsoft.com/en-us/library/hh286414%28v=vs.92%29.aspx>.

1. Tworzymy projekt (*File>New Project*    *Ctrl+Shift+N*). Typ projektu: *Visual C#, Windows Phone Application*.



2. W oknie *Solution Explorer* (*View>Other windows>Solution Explorer* *Ctrl+Alt+L*) widzimy pliki *App.xaml* oraz *Main.xaml*. Pierwszy zawiera informacje o aplikacji, a drugi o ułożeniu elementów na wyświetlaczu.  
 Póki co, należy dodać przycisk obsługujący kliknięcie (wkrótce posłuży on nam jako pobieracz listy kontaktów i aktualizator pola tekstowego). Wprowadzenie do używania przycisków znajduje się np. na <http://www.silverlightshow.net/items/Button-Controls-in-Silverlight-2-Beta-1.aspx>.
3. W pliku *MainPage.xaml* dopisujemy w tagu `<StackPanel>` kod  

```
<Button x:Name="getContacts" Content="przetwarzaj"></Button>
```

 W wyniku w widoku strony powinien zostać dodany przycisk.  
 Zmienne tekstowe wyświetlane na ekranie powinny znajdować się w resource file - pliku z zasobami. Informacje o tym jak należy to zrobić można znaleźć na *How to: Build a Localized Application for Windows Phone*  
<http://msdn.microsoft.com/en-us/library/ff637520%28v=vs.92%29.aspx>. Chcąc uprościć wszystko jak się tylko da - tekst zostanie wprowadzony hard-coded, tj. na stałe.  
 Pozostaje tylko dodanie obsługi kliknięcia klawisza. Wprowadzając atrybut `Click` środowisko samo nam proponuje dodanie nowego handler'a.  

```
<Button x:Name="getContacts" Content="przetwarzaj" Click="getContacts_Click"></Button>
```
4. klikając prawym na `getContacts_Click` możemy wybrać opcję *View Code*, która zaprowadzi nas do implementacji kodu.  
 Przy okazji: aby szybko wrócić do ostatniego miejsca (czyli np. miejsca z którego przenieśliśmy się do innego miejsca, możemy użyć polecenia *View.NavigateBackward* (*Ctrl+-*), które jest odpowiednikiem *Ctrl+Q* w środowisku *Eclipse*. Polecenie działające odwrotnie do

*View.NavigateBackward* to *View.NavigateForward* (**Ctrl+Shift+-**). Każdy z tych skrótów można zdefiniować w menu *Tools>Options>zakładka Environment>Keyboard*. Skróty te mogą pomóc w sprawnym poruszaniu się po kodzie źródłowym.

5. W celu sprawdzenia czy wciśnięcie przycisku jest poprawnie obsługiwane spowodujemy, aby aplikacja wyświetliła prosty komunikat. Więcej o powiadomieniach można znaleźć w [1]. Użyjemy metody `MessageBox.Show` w tym celu przechodzimy do implementacji metody klikając prawym na `Click="getContacts_Click"` > View Code (skrót to **F7**, lecz by go móc używać należy zmienić w menu *Tools>Settings>Expert Settings*):

```
private void getContacts_Click(object sender, RoutedEventArgs e)
{
    (sender as Button).Content = "nowy napis";
    (sender as Button).UpdateLayout();
    MessageBox.Show("udało się - działa");
}
```

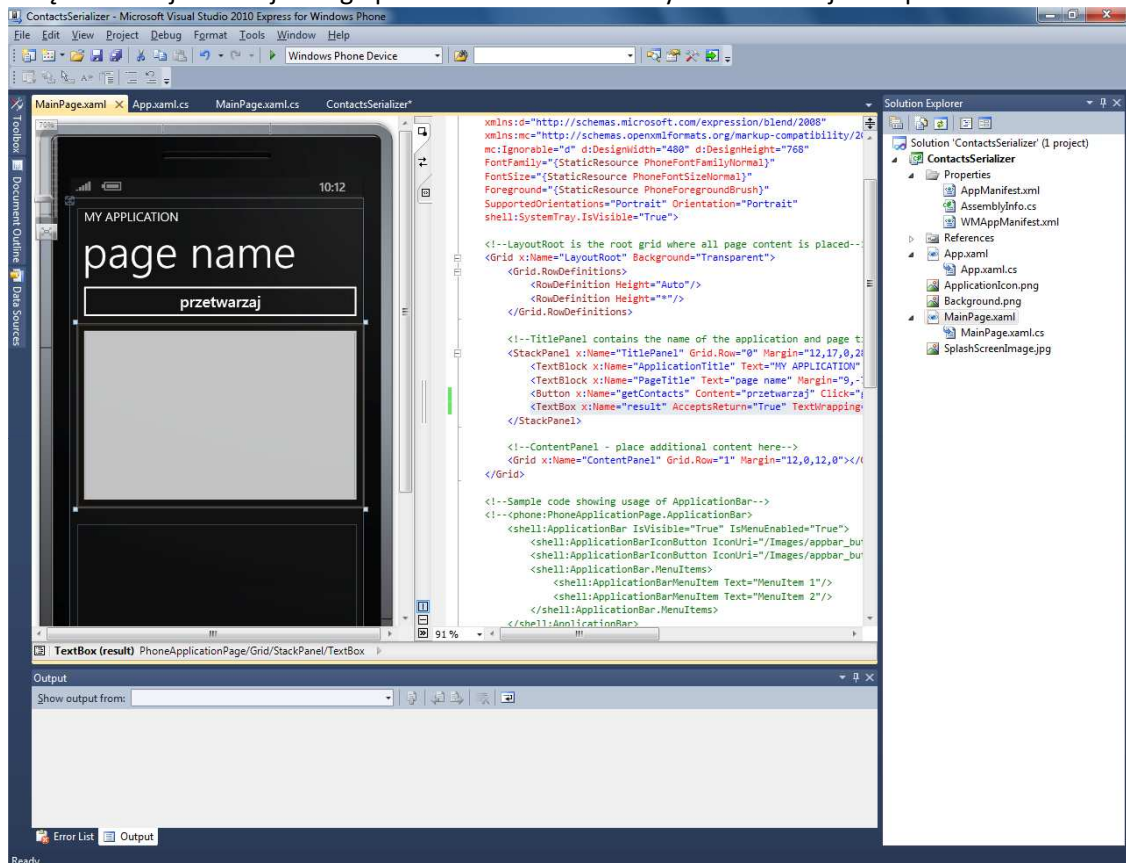
Wracamy do pliku *MainApp.xaml*, np. używając skrótu **Ctrl+-**.

6. Pozostało dodanie pola tekstowego, które będzie prezentować rezultat naszego prostego programu. W tym celu można dodać pole tekstowe

```
<TextBox x:Name="result" AcceptsReturn="True" TextWrapping="Wrap"
Height="300"></TextBox>
```

`AcceptsReturn` umożliwia wprowadzanie nowych linii, a `TextWrapping` realizuje zawijanie tekstu.

Dzięki temu z jednolinijkowego pola `TextBox` można otrzymać wielolinijkowe pole tekstowe.



7. Można sprawdzić czy pole tekstowe daje się rzeczywiście wypełnić tekstem, w tym celu dodamy dodanie tekstu do pola tekstowego `TextBox x:Name="result"`. Definicja metody `getContacts_Click` wygląda następująco:

```
private void getContacts_Click(object sender, RoutedEventArgs e)
```

```

{
    (sender as Button).Content = "Hello World!\n";
    (sender as Button).UpdateLayout();
    (FindName("result") as TextBox).Text += "How are you?\n";
    (FindName("result") as TextBox).UpdateLayout();
    MessageBox.Show("udało się");
}

```

8. Nadszedł czas na pobieranie kontaktów i ich wypisywanie.

Wyszukiwanie kontaktów polega na zdefiniowaniu metody callback, która zostanie wywołana po znalezieniu kontaktów

```

cons.SearchCompleted += new
EventHandler<ContactsSearchEventArgs>(Contacts_SearchCompleted);

```

i wywołaniu przeszukania kontaktów z parametrami, które wskazują na to, że każdy rekord ma być zwrócony.

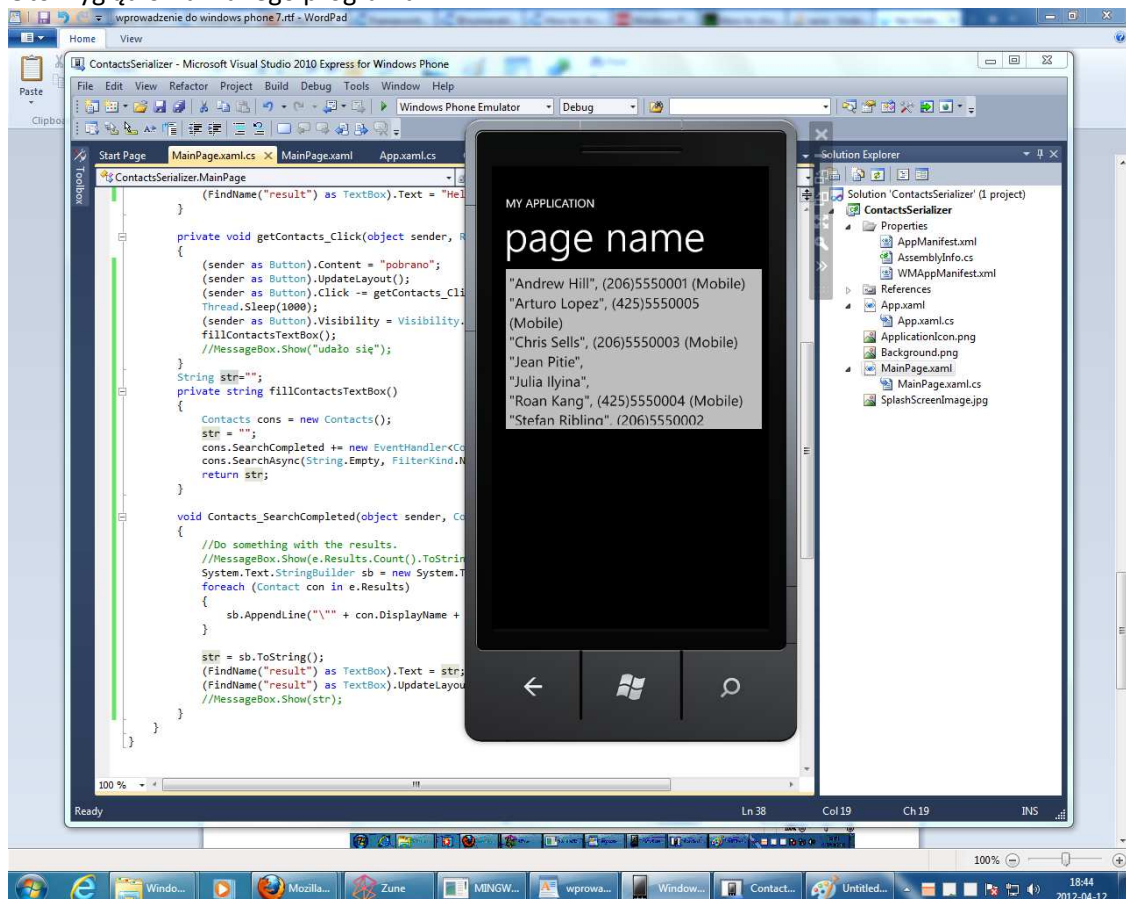
```

cons.SearchAsync(String.Empty, FilterKind.None, "Contacts Test #1");

```

9. Do przekazania zmiennej tekstowej zawierającej wszystkie kontakty do metody obsługującej zakończenie przeszukiwania kontaktów użyjemy pola klasy str.

10. Oto wygląd omawianego programu:



Kody źródłowe można znaleźć na <https://github.com/arjamizo/Contacts-Serializer>.

## FAQ

1. Nie można wysłać aplikacji na telefon. Phone is pin locked.

**Rozwiązanie:**

Wystarczy odblokować telefon.

2. Nie można wysłać aplikacji na telefon. Niezarejestrowane urządzenie.

Błąd o zablokowanym telefonie dla developera.



Studio 2010 Express for Windows Phone

Failed to connect to device as it is developer locked. For details on developer unlock, visit <http://go.microsoft.com/fwlink/?LinkId=195284>.

**Rozwiązanie:**

Aby odblokować telefon wchodzimy na <https://users.create.msdn.com/account/profile>, założyć tam konto musimy zarejestrować się używając konta poczty elektronicznej z uczelni na <http://dreamspark.com>, następnie zweryfikować to, że jesteśmy studentami.

Oficjalna instrukcja znajduje się na stronie

<http://msdn.microsoft.com/library/ff769508%28v=VS.92%29.aspx>.

Można już wgrywać aplikacje na telefon używając Microsoft Visual Studio 2010 Express for Windows Phone.

## Warto zobaczyć

[1]

<http://channel9.msdn.com/Series/Kurs-programowania-Windows-Phone-7/Kurs-Windows-Phone-7-Cz-9-Push-notifications?format=html5>

[2]

<http://www.windowsphonegeek.com/articles/How-to-choose-a-Contact-and-get-Contact-details-in-a-WP7-app>

Opracował Artur Zochniak (184725@student.pwr.wroc.pl)

ISAA, Politechnika Wrocławska 2012.