



Wstęp + Podstawy

Kilka słów wstępu..

Zapewne kupiłeś/-aś ten poradnik, aby móc samodzielnie pisać keygeny, generatory czy inne programy, a później dzięki promowaniu czerpać z nich niezłą kasę. Ja właśnie chcę Tobie to umożliwić. Nie mogę jednak Ci zagwarantować, że to co stworzysz będzie miało popyt. Więc proszę nie składać później żadnych zażaleń. Natomiast jeżeli będziecie mieli jakieś pytania, prośby, czy inne zachcianki odsyłam na forumowe PW.

Poświęciłem swój czas na napisanie tego poradnika, więc zabraniam kopiowania jego treści, sprzedaży go, jego czy pół-treści, wymieniania się nim, szantażowania nim Microsoft bez mojej zgody. Powstał on w celu edukacyjnym, podczas pisania nie ucierpiały żadne zwierzęta, a cały tekst został napisany przez profesjonalistów, proszę nie powtarzać tego w domu.

ROCKNDRUG

Założenia

Czyli co tak naprawdę otrzyma osoba, która kupiła ten poradnik?

Osoba po przeczytaniu tego poradnika powinna

- Nie mieć problemów z podstawową obsługą VB
- Umieć tworzyć bardzo wiarygodne programy
- Bez problemów tworzyć bardziej „profesjonalne” generatory
- Czerpać korzyści z napisanych programów

Osoba po przeczytaniu nie powinna:

- Uważać czasu poświęconego na przeczytanie poradnika, oraz wydane na niego pieniądze za stracone.

wiec zaczniemy od..



A więc zacznijmy od..

..od zaopatrzenia się oczywiście w Virtual Basic, jeżeli wcześniej tego nie zrobiliśmy. Microsoft Visual Basic 2010 Express będzie bardzo dobrym rozwiązaniem na początek.

Pobierz Visual Basic 2010 Express

<http://www.microsoft.com/express/Downloads/#2010-Visual-Basic>

lub

<http://www.dobreprogramy.pl/Visual-Basic,Program,Windows,12110.html>

Jeśli nam się skończy „trial” możemy za darmo otrzymać key, poprzez rejestrację na stronie

Visual Basic 2010 może być zainstalowany na:

Windows XP (x86) z Service Pack 3 – wszystkie edycje
Windows Vista (x86 & x64) z Service Pack 2 – wszystkie edycje
Windows 7 (x86 & x64)
Windows Server 2003 (x86 & x64) z Service Pack 2
Windows Server 2003 R2 (x86 & x64)
Windows Server 2008 (x86 & x64) z Service Pack 2
Windows Server 2008 R2 (x64)

Sprzętowe wymagania:

Komputer z procesorem 1.6GHz lub szybszym
1 GB (32 Bit) lub 2 GB (64 Bit) RAM'u (+ dodatkowe 512 RAM jeśli odpalicie Wirtualną Maszynę)
3GB wolnego miejsca na waszym HDD
DirectX 9 kompatybilny z kartą graficzną
Myszka, Klawiatura, Monitor itd..

+

Może być dodatkowo potrzebny NET Framework 4.0

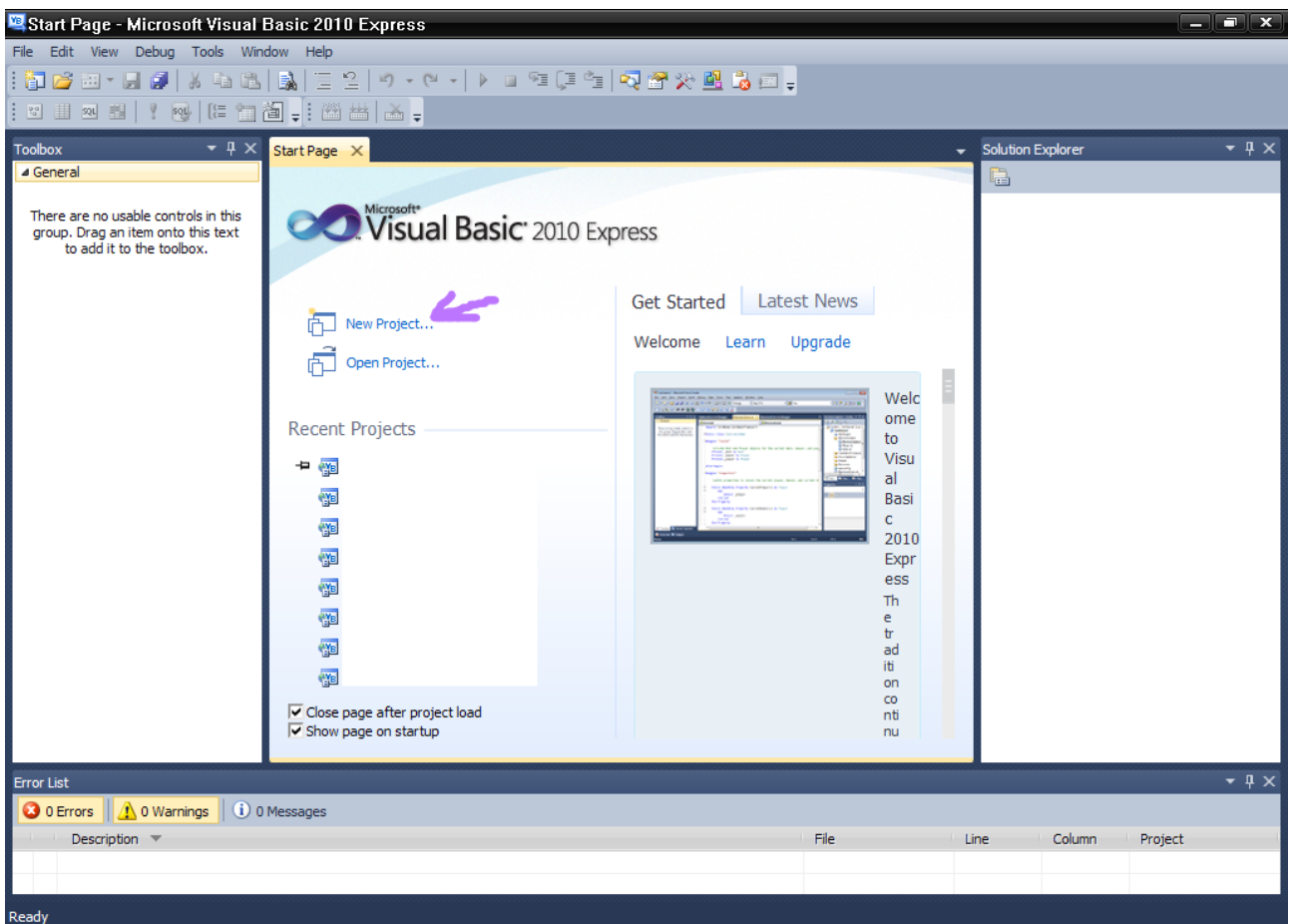
Ale co to w ogóle jest Virtual Basic?

Przepisania z Wikipedii i innych stron co to jest za program, jak powstał, przez kogo i w jakim garażu sobie odpuszczam. Zbyteczne info, jeśli weźmiemy pod uwagę, że chcemy zarabiać na naszych „programach” a nie wiedzieć wszystkiego nt. twórców Visual Basica. Ktoś będzie chciał wiedzieć więcej, wie gdzie szukać.



1.0 – Podstawy – podstawowe elementy

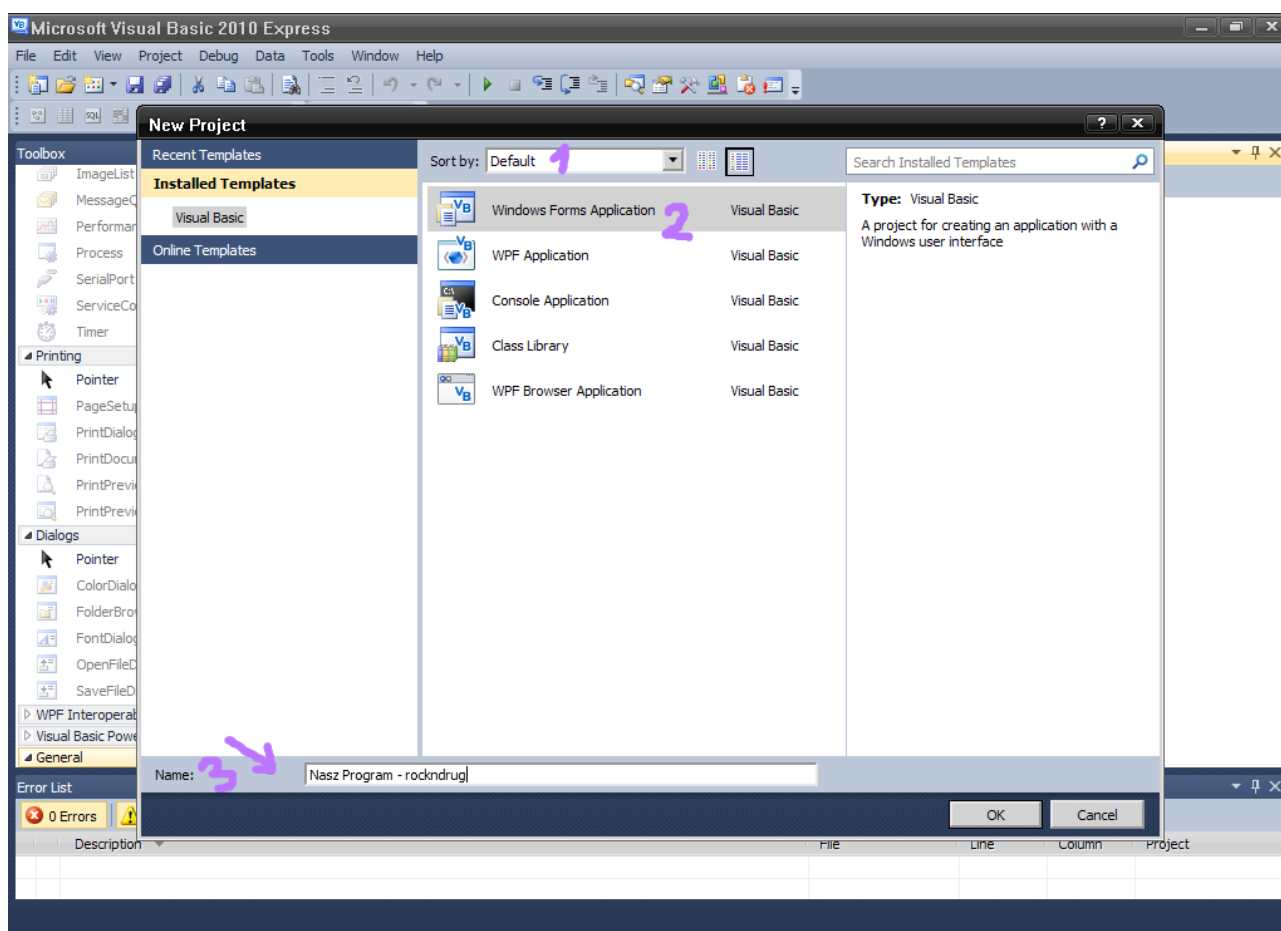
Pierwszą rzeczą jaką musimy zrobić, to odpalić naszego Visual Basic'a. Następne postępujemy wg. kroków(jak na screenie) Czyli klikamy na przycisk "**New Project**"





1.1 – Podstawy – podstawowe elementy

Naszym oczom okazują się kolejne okienko. Postępujemy zgodnie ze screenem. Na początek ustawiamy Sort by: na "Default", wybieramy "Windows Forms Application" no i ustawiamy nazwę naszego projektu

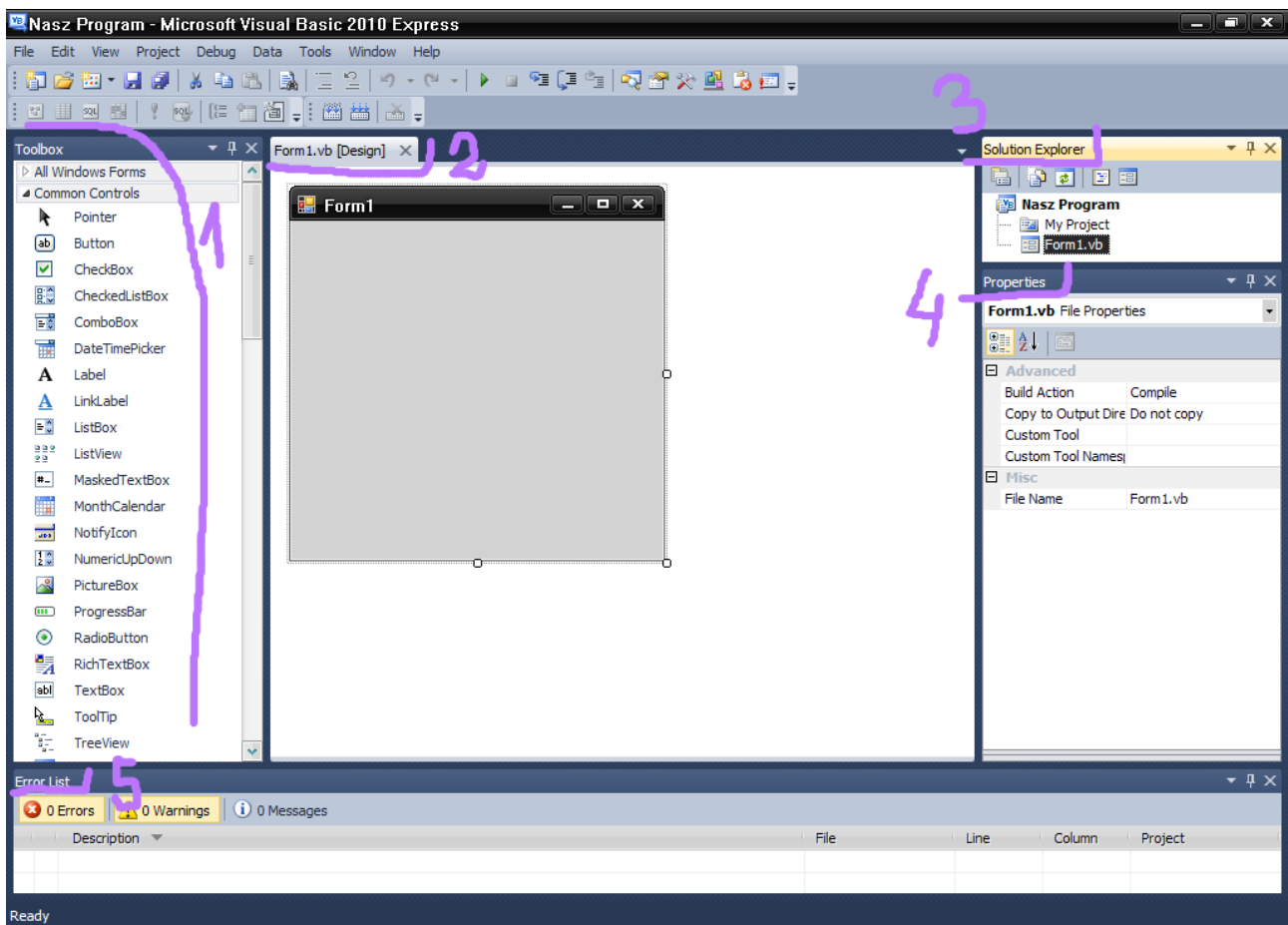




1.2 – Podstawy – podstawowe elementy

I oto okno, z którym będziemy mieli do czynienia przez długi czas. Na początku wydają się to bardzo nieprzejrzyste, ale troszkę wprawy, i będziecie śmiagać po VB jak zawodowcy.

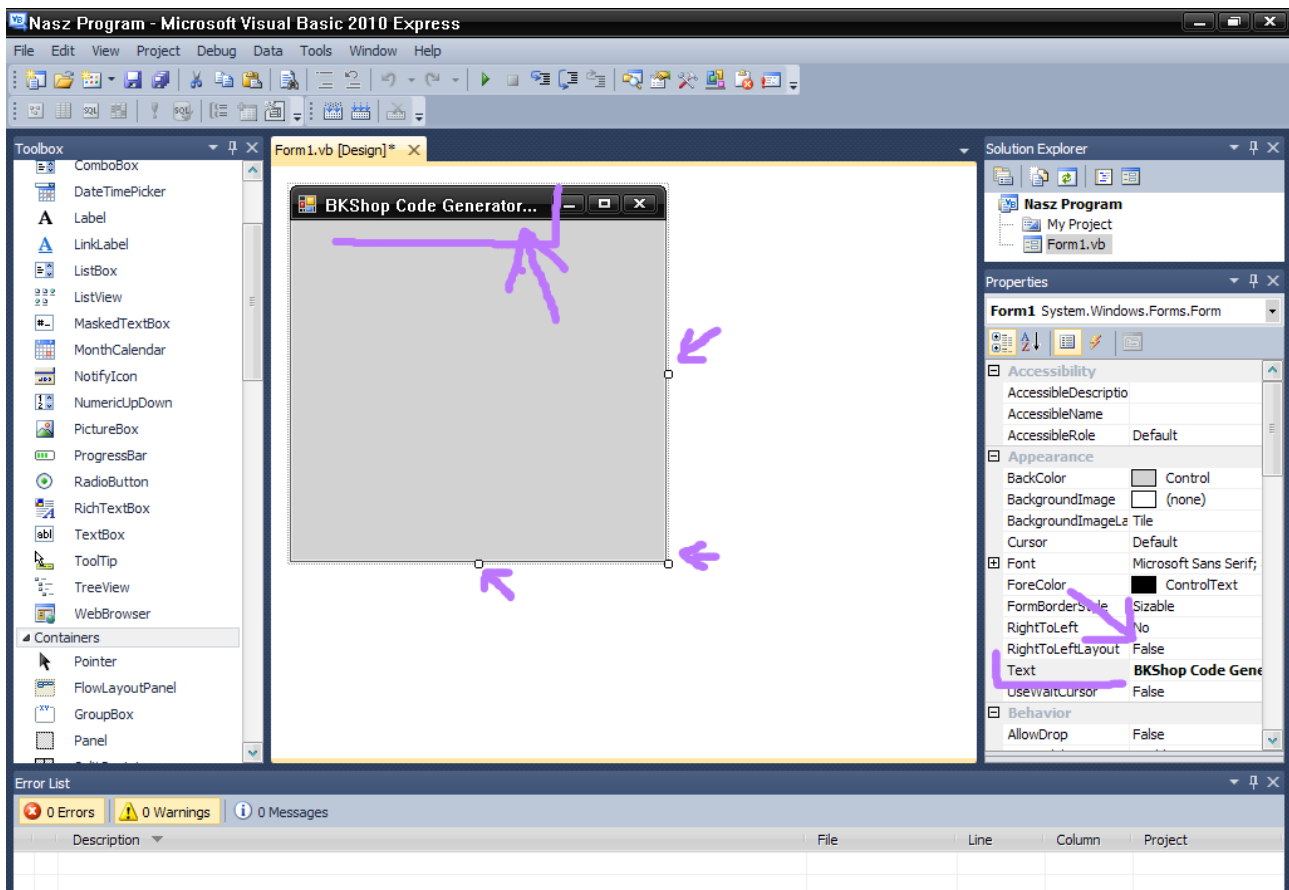
1. Toolbox – czyli taki "zbiór" wszystkich narzędzi. To właśnie z tego narzędzia korzystamy, kiedy chcemy dodać jakiś przycisk, tekst, link, pasek ładowania czy inne "meczyje"
2. Form1.vb[Design] – ja to nazywam w skrócie Podglądem. Okno te pozwala graficznie zobaczyć jak wygląda nasz program i wprowadzać na nim zmiany [np. dodanie nowego przycisku]
3. Solution Explorer – czyli cały nasz projekt, bardzo ładnie pokazany w "drzewku"
4. Properties – czyli właściwości danego elementu, możecie teraz zauważyć, że mam na screenie zaznaczone "Form1.vb" w "Solution Explorer". Properties pozwala mi na podgląd i zmianę właściwości danego aktywnego elementu. Będzie to nam niezbędne w późniejszej części (np. przy progressbar'ze)
5. Error List – czyli jeżeli po próbie testu naszego programu, aplikacja nie będzie chciała zadziałać, to tu powinniśmy zobaczyć co tak naprawdę nam się posypało, i co musimy naprawić.





1.3 – Podstawy – podstawowe elementy

Skoro już mniej-więcej orientujemy się, gdzie co leży pora na stworzeniu jakiejś prostej "aplikacji". Weźmy się za generowaniu kilku kluczy (cd-key'ów). Program będzie miał za zadanie wygenerować klucz, po wciśnięciu przycisku "generate" Prosty, banalnie wyglądający programik. Jak widzimy na poprzednik screenie, nasz program aktualnie nazywa się "Form1". Musimy to zmienić zakładając że chcemy stworzyć "BKShop Code Generator v1.2" (wymyślone w kilka sekund, nie czepiać się). W tym celu w "Properties" wyszukujemy pola „Text” i zmieniamy jego wartość na "BKShop Code Generator v1.2". Tak jak jest to widoczne na screenie poniżej.



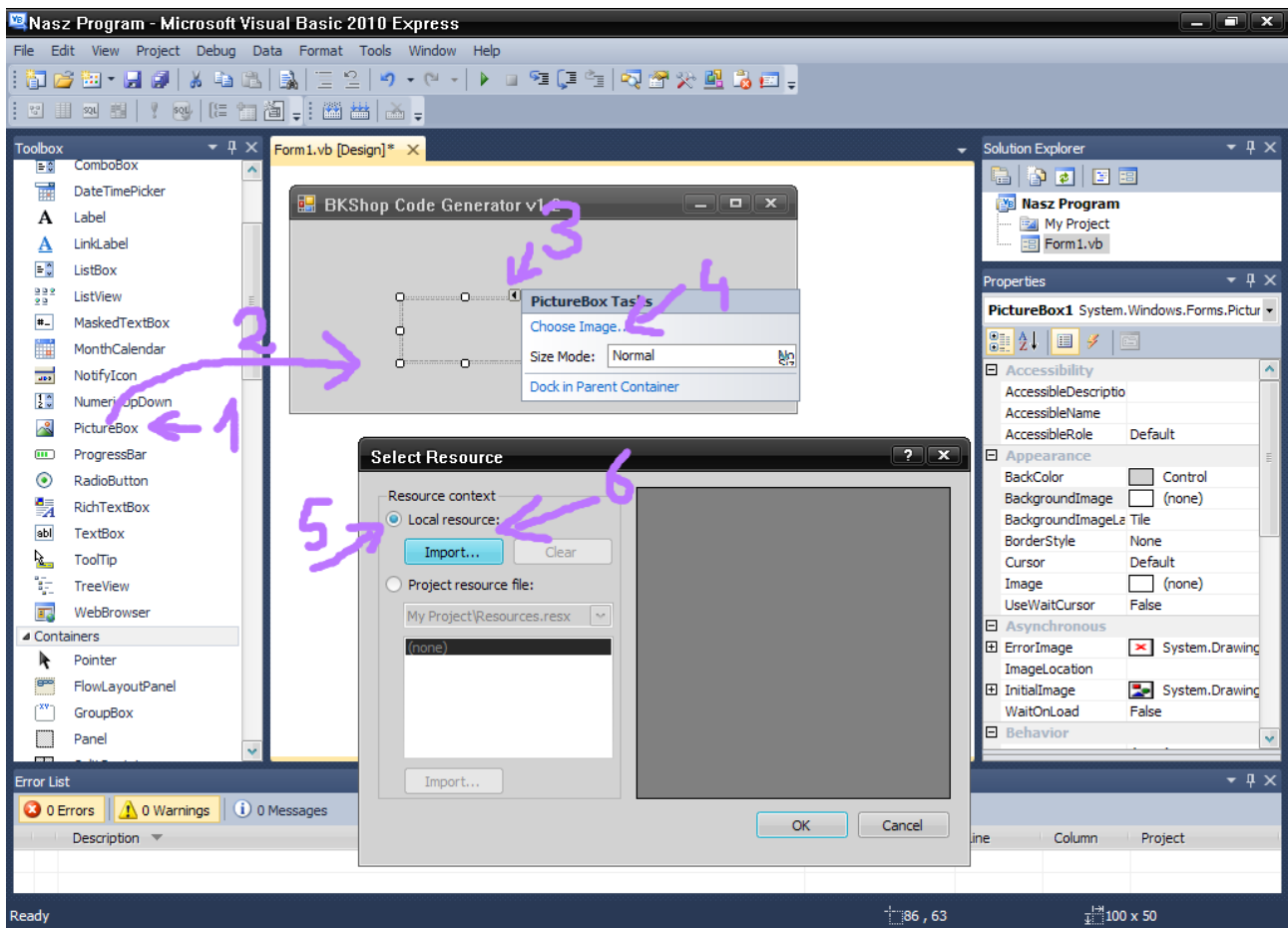
Zauważmy, że nazwa zmieniła się na tą, którą chcieliśmy, ale nie zmieściła się. Możemy temu zaradzić poszerzając okno programu, używając tych "kwadracików", które zostały zaznaczone na screenie, lub mając aktywne całe okno programu wyszukać w "Properties" pola "Size" i zmienić jego wartość.



1.4 – Podstawy – podstawowe elementy

Mamy już zmienioną wielkość okna. Teraz czas na dodanie jakiegoś tła. Przecież nie chcemy aby nasz program był "szary". Na potrzeby tego poradnika pobrałem losowe tło z grafiki google, po wpisaniu słowa kluczowego "background", czyli "tło".

1. Nasz obrazek dodajemy poprzez wybór z "Toolbox'a" narzędzia "PictureBox",
2. Następnie umieszczamy go na naszym "projekcie".
3. Wybieramy taką małą strzałeczkę, która jest zaznaczona na screenie.
4. Klikamy na "Choose Image" czyli "Wybierz Obrazek"
5. Zaznaczamy opcję, że chcemy wybrać obrazek z naszego dysku
6. Klikamy na opcję "Import", po czym wybieramy nasz obrazek.



Tło naszej aplikacji możemy dodać również na drugi sposób, poprzez wybór z pola "Properties", "Background Image" oraz podanie ścieżki do tła, analogicznie jak przy dodawaniu tła poprzednim sposobem (od punktu czwartego) Jest to **LEPSZY SPOSÓB**, ponieważ nie będziemy mieli później problemem z ustawieniem "przezroczystości" jakiegoś tekstu. ;)



1.4.1 – Podstawy – podstawowe elementy

Żeby nasze tło nie wyglądało tak "biednie" postanowiłem dodać jeszcze jakieś logo, zwykle dodaje tam np logo z mojej strony, jeżeli dodaje generator, którego dodatkowo umieszczam na mojej stronie. Używam w tym celu "PictureBox'a". Warto pobawić się też opcjami, które są widoczne na screenie, a mianowicie:

BackColor – Odpowiada za tło (Jeśli jest ustawiony odpowiedni Flat Style, systemowy przycisk nie będzie miał tła – dot. przycisków)

BackgroundImage – Jeśli chcemy aby zamiast jednolitego tła, był widoczny obrazek, wybieramy go z dusku

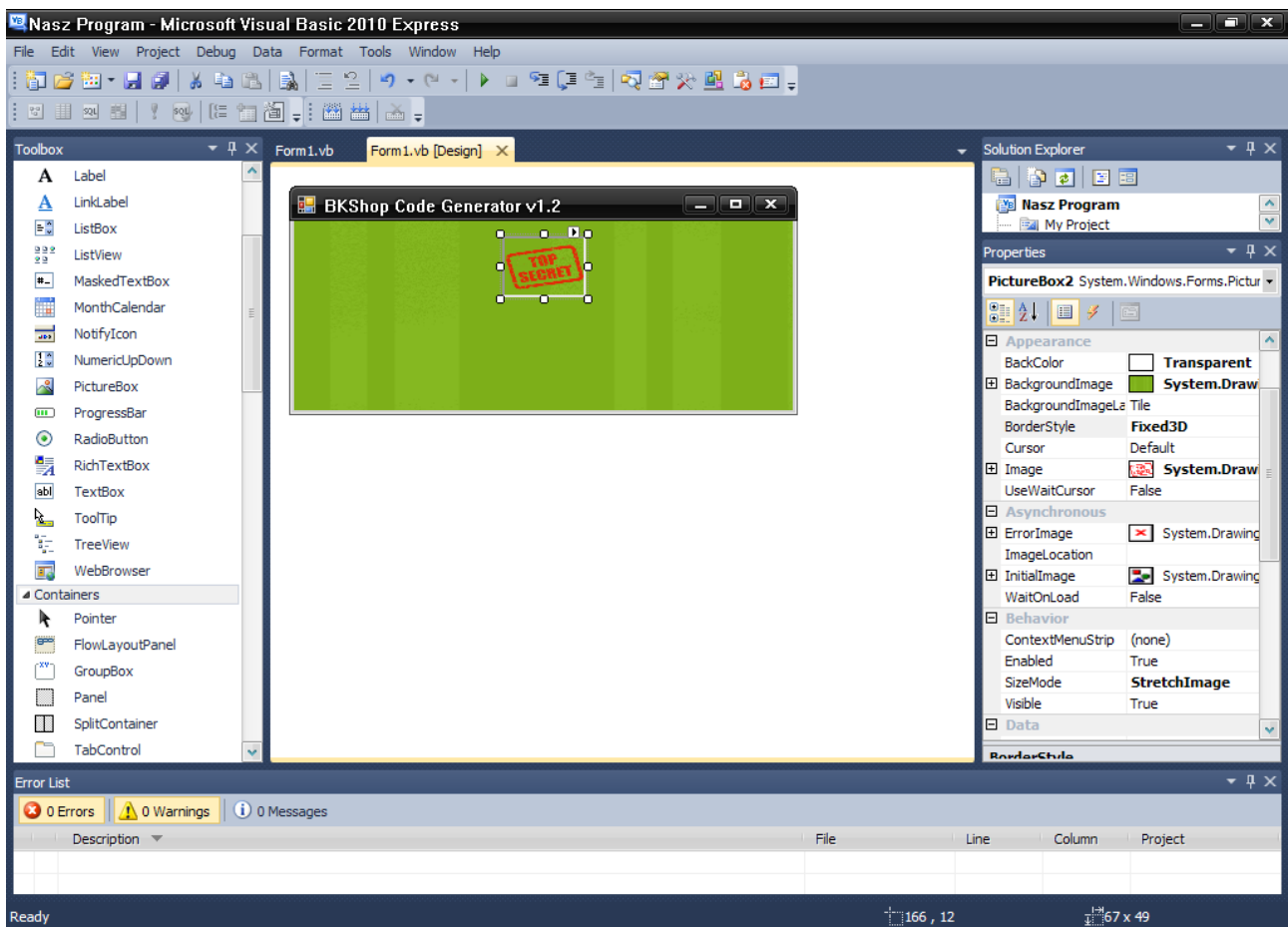
Flat Style – Zmieniamy wygląd naszego okna, przycisku, całkiem ciekawa i użyteczna opcja
Font – Czcionka

Fore Color – Kolor tekstu (np. W przycisku, teksie itp)

Text – Czyli tekst, przykładowo jeżeli chcemy stworzyć przycisk „Exit” wpisujemy tutaj „Exit”

TextAlign – Opcja która pozwala nam wybrać, gdzie umiejscowić tekst na dany element

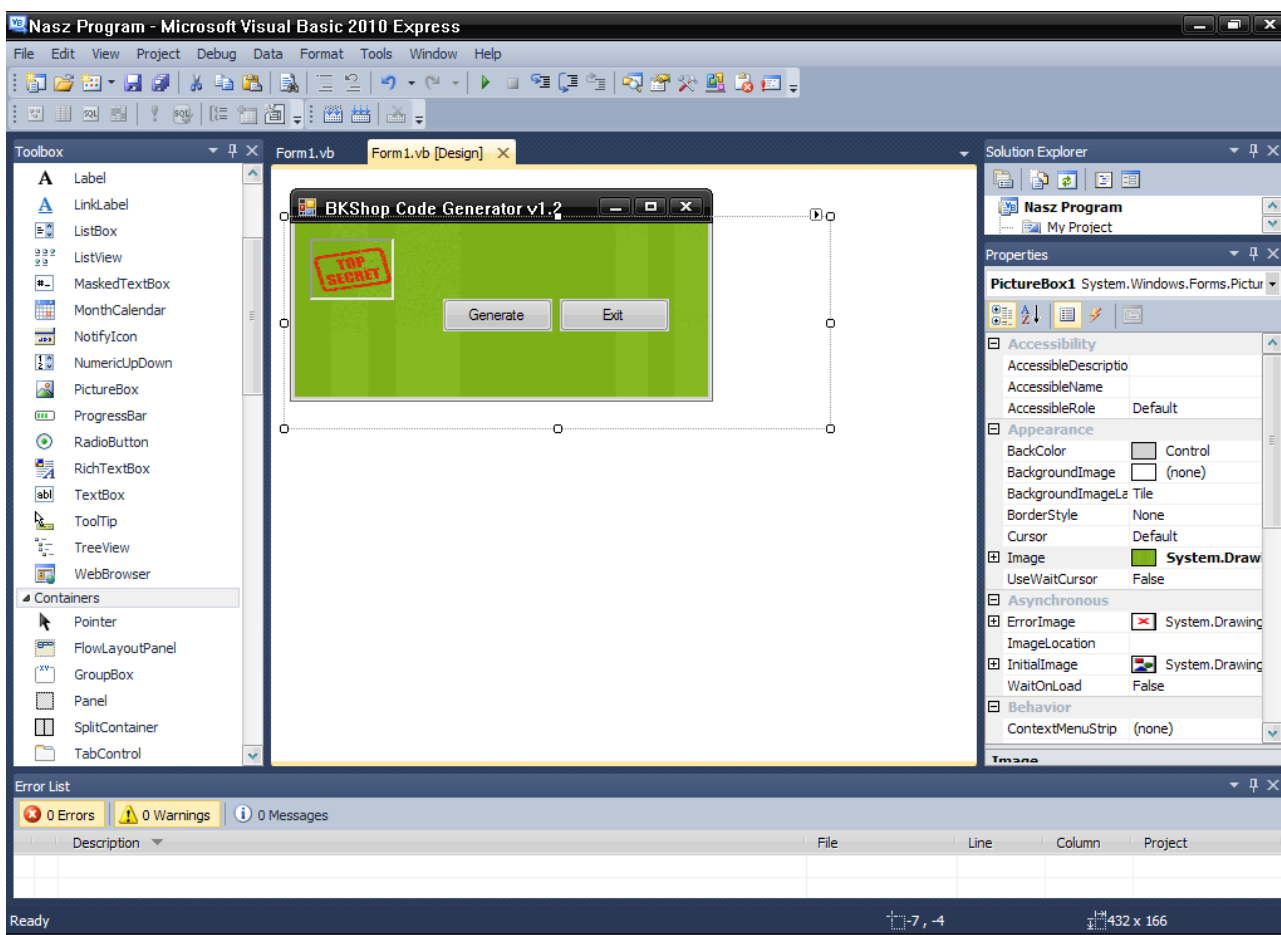
Location – jeżeli nie możemy „myszką” wstawić elementu w odpowiednie miejsce, zmieniamy wartości X,Y. Przykładowo (152; 321).





1.5 – Podstawy – podstawowe elementy

Czas na dodanie przycisku "Generate", który ma służyć do odpalenia całego generatora kluczy. Bez niego nie da się obejść :). Tak samo jak szukaliśmy PictureBox'a, znajdujemy narzędzie "Button" i umieszczamy go na naszym projekcie. Aby edytować napis w "przycisku" należy postąpić tak, jak robiliśmy to z nazwą naszej aplikacji. Ja dodałem 2 przyciski. Generate i Exit.. wiadomo do czego mają służyć. Po czym stwierdziłem że aplikacja jest za "długa". Troszkę ją zmodernizowałem.

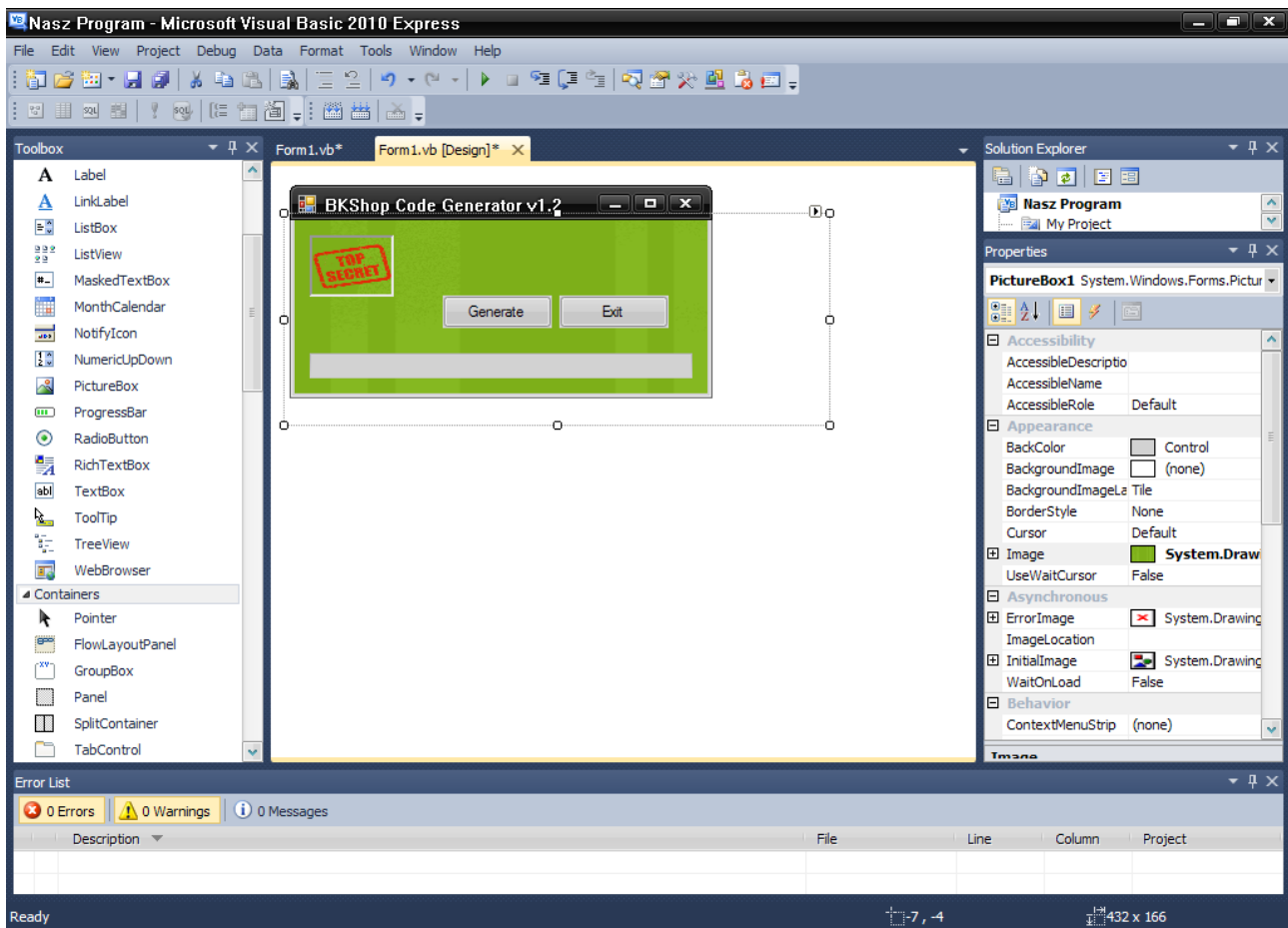


Tutaj warto wspomnieć o tym, że każdy kolejny element przez nas dodawany ma swoją analogiczną nazwę. Jeżeli dodajemy "Button" to ma on nazwę "Button1", natomiast jeżeli dodamy kolejny przycisk, będzie miał on już nazwę "Button2". Musimy o tym pamiętać, przy późniejszym nadawaniu funkcji, oraz pisaniu aplikacji.



1.6 – Podstawy – podstawowe elementy

Prawie ostatnią rzeczą jaką nasz "generator" potrzebuje jest pole, w którym będzie się generował kluczyk. Przecież co to za generator, który nic nie generuje. :) W tym celu z "Toolbox'a" wybieramy "TextBox" i umieszczamy go. Przeważnie po umieszczeniu go, ustawiam pole "ReadOnly" na "True", po co ktoś ma pisać po tym polu, wystarczy jak skopiuje sobie wygenerowany kluczyk. Całość u mnie wygląda następująco.



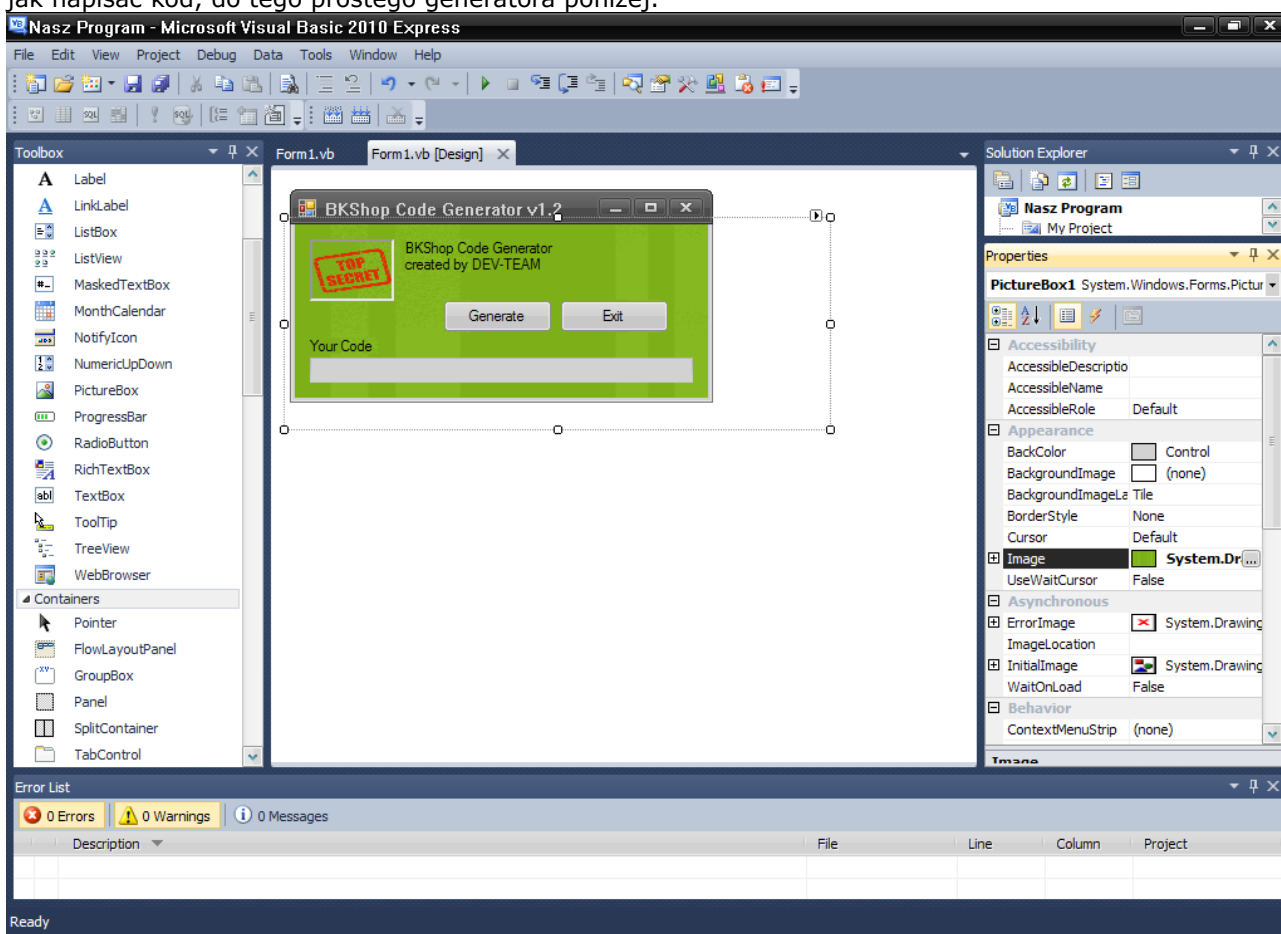


1.6.1 – Podstawy – podstawowe elementy

Wszystko mamy prawie dodane, ale dobrze by było, jakbyśmy dodali jakiś tekst. Powiedzmy "Your Code", co ten generator robi itp. Do tego służy narzędzie "Label" z "Toolbox'a" Ja dodałem "Label" dwa razy, zmieniłem trochę położenie i ustawiłem tło na przezroczyste

Background -> Web -> Transparent

Teraz możemy nacisnąć klawisz "F5" i zobaczyć jak nasz program działa. A raczej jak on "nie-działa" przecież nie napisaliśmy ani jednej linijki kodu. Wstęp i Podstawy miały na celu tylko zapoznanie was z moim zdaniem niezbędnymi funkcjami, w kolejnych wersjach wszystko będzie uaktualniane, dodane zostaną nowe funkcje oraz postaram się pokazać więcej opcji. Tymczasem podstawy masz już opanowane, więc myślę, że możesz zabrać się za czytanie kolejnego poradnika, w którym wytłumaczę, jak napisać kod, do tego prostego generatora poniżej.



1.6.2 – Podstawy – zmiana ikony

Szczerze mówiąc to trochę się "pośpieszyłem" i zapomniałem wspomnieć o zmianie ikonki. Jak możemy zauważyć, obok nazwy naszego programu widnieje standardowa ikona, jaką dodaje Visual Basic. Aby ją zmienić klikamy na nasze okno (możemy np. Na tytuł okienka) w wersji graficznej. Aby ją zmienić mając aktywne okno naszego programu wyszukujemy w Properties pola „ICON”. Teraz pozostaje nam tylko wybrać ikonkę. Jeżeli nie masz ikonki do swojego programu, i nie wiesz skąd ją wziąć najlepiej wyszukać w grafice google jakiegoś logo gry, strony czy do czego tam robicie generator. Ja wyszukałem swój wpisując – Tree icon

Wybieramy plik z komputera, i po chwili ściągamy ten sam obrazek w formacie ICO. Następnie, aby nasz program.exe miał też ikonę taką, jaką ustawiliśmy przed chwilą w



Myślę, że będzie odpowiednia. Tylko pobrałem ją z google w formacie .png. Aby przekonwertować ten obrazek na plik.ico Wchodzimy na stronę:
<http://www.coolutils.com/online/image-converter/>
i konwertujemy nasz obrazek, po czym pobieramy go w formacie ".ico".

Jeżeli udało nam się to zrobić, a obok naszej nazwy programu widnieje już nowa ikona, pozostaje nam jeszcze zmienić "ikonę" naszego programu, który będzie zapisany w postaci ".exe"

Aby to zrobić wchodzimy to w oknie „Solution Explorer”, które powinno znajdować się po prawej stronie klikamy 2xLPM na „My Project” i klikając na pasek pod napisem „icon” wybieramy Browse → a następnie naszą ikonę.

*Jeżeli nie masz „Solution Explorer” w oknie programu
Wejść w View → Other Windows → Solution Explorer*

1.7 – Podstawy – plik.exe

Teraz pozostaje nam tylko zapisać program w postaci pliku .exe. Aby to zrobić klikamy na "File" -> Save All. Ustawiamy ścieżkę, gdzie ma zapisać się program. Domyślnie są to moje dokumenty. Nasz plik .exe powinien się pojawić tutaj.

C:\D & S\HD2\Moje dokumenty\Visual Studio 2010\Projects\Nasz Program\Nasz Program\bin\Debug

Jeżeli np. Zapomnieliśmy zmienić ikonkę, to wchodzimy w Debug -> Build. I nasz plik.exe wyląduje w:

C:\D & S\HD2\Moje dokumenty\Visual Studio 2010\Projects\Nasz Program\Nasz Program\bin\Release

Ja zawsze zapisuję klikając Save All, później Build. Zdarza się, kiedy nie chce się zmienić ikona naszego programu, wtedy zapisuję kilka razy.

Przeczytaj następną część pt. "Kodujemy prosty generator"



Kodujemy prosty generator

Kilka słów wstępu..

Zapewne kupiłeś/-aś ten poradnik, aby móc samodzielnie pisać keygeny, generatory czy inne programy, a później dzięki promowaniu czerpać z nich niezłą kasę. Ja właśnie chcę Tobie to umożliwić. Nie mogę jednak Ci zagwarantować, że to co stworzysz będzie miało popyt. Więc proszę nie składać później żadnych zażaleń. Natomiast jeżeli będziecie mieli jakieś pytania, prośby, czy inne zachcianki odsyłam na forumowe PW.

Poświęciłem swój czas na napisanie tego poradnika, więc zabraniam kopiowania jego treści, sprzedaży go, jego czy pół-treści, wymieniania się nim, szantażowania nim Microsoft bez mojej zgody. Powstał on w celu edukacyjnym, podczas pisania nie ucierpiał żadne zwierzęta, a cały tekst został napisany przez profesjonalistów, proszę nie powtarzać tego w domu.

ROCKNDRUG

2.0 – Kodujemy prosty generator

Mamy już „zaprojektowany” nasz pierwszy generator. Jednak sam projekt nie wystarczy. Aby ludzie uwierzyli, że on naprawdę działa. Zapewne większość z was nie miała nigdy styczności z jakimkolwiek programowaniem, więc chciałem tylko zaznaczyć że tego nie da się nauczyć. Można znać regułki pisania programów na pamięć, a nie napiszecie nic. Tutaj chodzi o logiczne myślenie. Jeżeli będziecie mieli jakiś problem z kodem, spróbujcie najpierw sami posiedzieć chwilę, i się zastanowić zanim polecicie na forum i zapytacie userów o gotowy kod. Tak naprawdę komputer nie może was oszukać, więc jeżeli coś wam nie chce działać, to tylko i wyłącznie z waszej winy. Bo komputer widzi tylko dwie opcje „TAK” i „NIE” ← prościej nie mogłem tego wytłumaczyć. On nie „zastanawia się” nad waszymi decyzjami i nie próbuje wprowadzać samodzielnych zmian. Powiedzmy:

```
If TextBox1.Text = "" Then
    MessageBox.Show("blad", "error")
end If
```

W skrócie:

```
Jeżeli TextBox1.Text = "" to           (jeżeli TextBox1 będzie pusty to)
    MessageBox.Show("blad", "error")     (wyświetl taki i taki błąd)
Koniec Jeżeli                           (tutaj kończy się „jeżeli”, czyli dalszy
                                         kod nie będzie podlegał temu warunkowi)
```

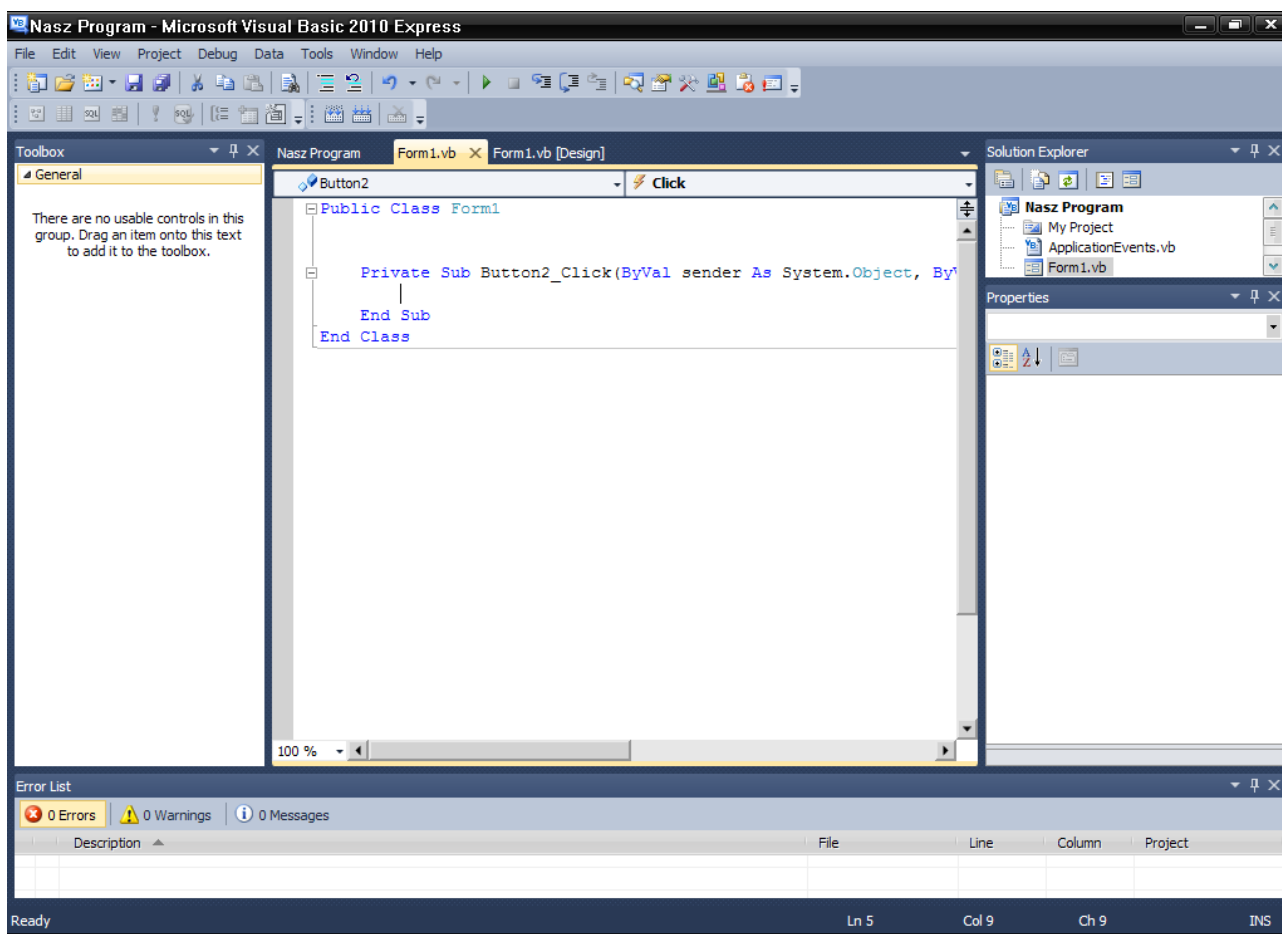
Jeżeli byśmy nie dodali „MessageBox.Show(„blad”, „error”)” to komputer sam by za nas tego błędu nie wygenerował. On dostałby tylko informację, że jeżeli TextBox1.text będzie pusty to.. nic.

To tyle na „początek”



2.1 – Kodujemy prosty generator

Nasz generator z poprzedniego poradnika „nie działa” a chcemy, żeby on jednak generował jakieś klucze. W tym generatorze użyliśmy 2 przycisków „**Generate**” oraz „**Exit**”. Jednak po kliknięciu w jakikolwiek przycisk nic się nie dzieje. Zaczynamy od tego prostszego, czyli „**Exit**”. W tym celu klikamy dwukrotnie w przycisk, aby przejść do edytora kodu. Waszym oczom powinno się ukazać takie okno:



Między Private Sub, a End Sub dopisujemy po prostu „**End**”, całość ma wyglądać tak:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    End
End Sub
```

Następnie naciskamy klawisz „**F5**” i sprawdzamy funkcjonalność przycisku „**Exit**” w naszym generatorze. Jeżeli wszystko wykonałeś poprawnie, program się zamknie.

GRATULACJE, WŁAŚNIE UDAŁO CI SIĘ ZAPROGRAMOWAĆ PIERWSZY PRZYCISK!



2.1 – Kodujemy prosty generator

Jeden przycisk już działa, teraz czas na drugi. Będzie on dużo bardziej skomplikowany, ponieważ ma on w „Textbox1” dodatkowo pokazać nam wygenerowany klucz. Niemniej jednak zacznijmy od naprawdę „prostego” generowania. Klikamy dwukrotnie na przycisk „Generate” z poziomu podglądu graficznego (tak jak to robiliśmy z przyciskiem exit) i piszemy tam taki oto kod:

```
Dim key As Integer
key = (Rnd() * 3+1)
Select Case key
    Case 1
        TextBox1.Text = "AG7V3BG895SR"
    Case 2
        TextBox1.Text = "NS2OP5XV5B11"
    Case 3
        TextBox1.Text = "41S9VBG7YT2P"
End Select
```

No i sprawdzamy jak dany program działa. Jak widzimy nasz „kluczyk” się generuje, kiedy klikamy kilkakrotnie w przycisk generate nasz klucz będzie „losowo” wybierany spośród tych trzech. Kiedy byśmy chcieli losować klucz z 4, lub więcej wystarczyłoby przekształcić kod na:

```
Dim key As Integer
key = (Rnd() * 4+1)
Select Case key
    Case 1
        TextBox1.Text = "AG7V3BG895SR"
    Case 2
        TextBox1.Text = "NS2OP5XV5B11"
    Case 3
        TextBox1.Text = "41S9VBG7YT2P"
    Case 4
        TextBox1.Text = "G6U23KNVPA4A"
End Select
```

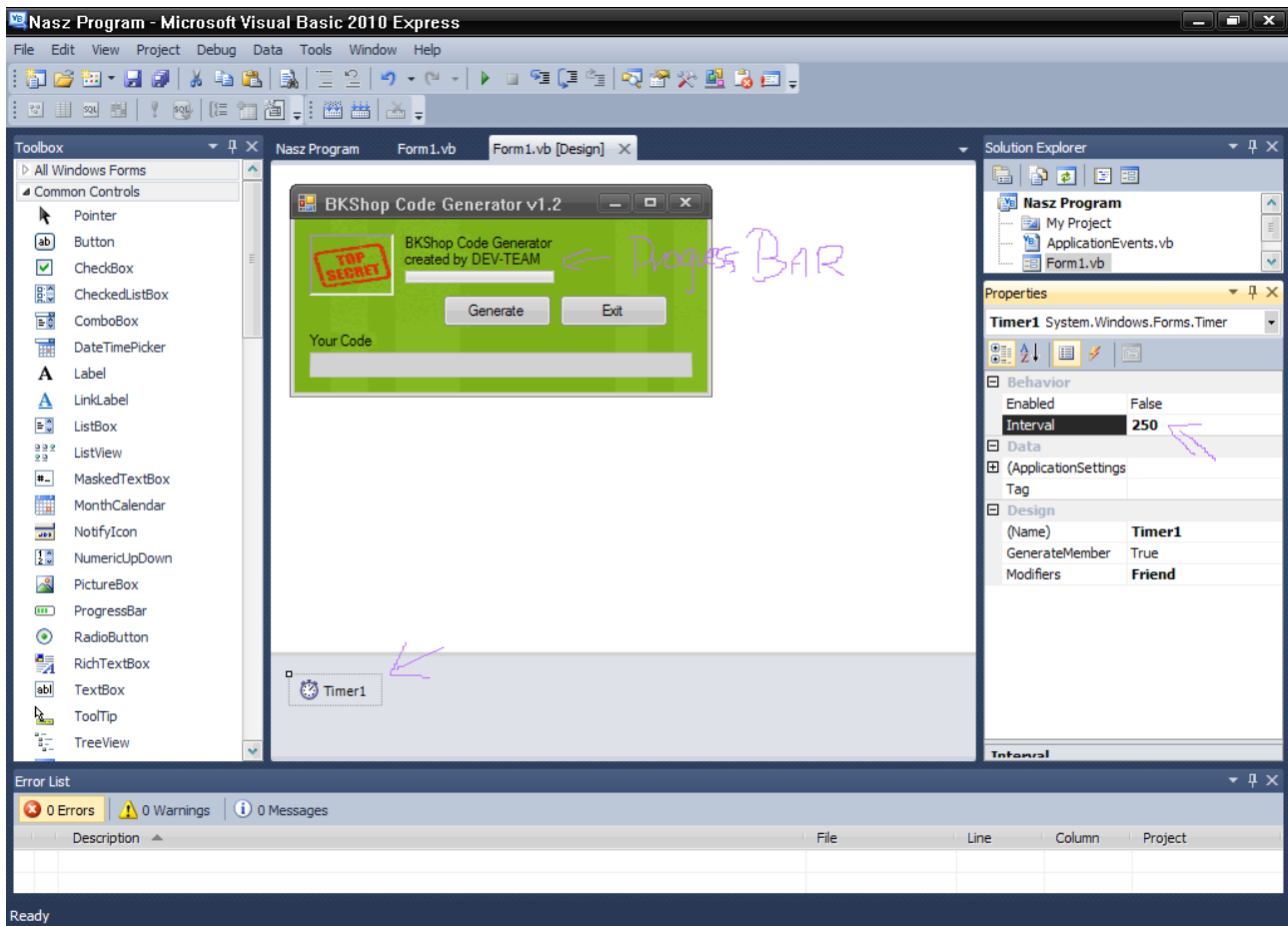
Teraz faktycznie generują się cztery klucze, ale to takie „proste generowanie” . Trzeba za każdym razem bawić się w dopisywanie kluczy itp. Do bardziej „złożonego” generatora przejdę później, teraz czas na pasek ładowania.

2.1.1 – Kodujemy prosty generator – pasek ładowania

Żeby nasz pasek ładowania działał, potrzebujemy nieco zmodyfikować nasz kod oraz niezbędne dla nas będą niektóre narzędzia z „**toolbox'a**”. Mam na myśli „**ProgressBar**” - pasek ładowania, oraz „**Timer**” - czyli coś, co będzie kierowało naszym ładowaniem.

Wybieramy z menu „**toolbox'a**” **Progressbar**” i umieszczamy go gdziekolwiek, rozciągamy itp. A następnie wybieramy „**Timer**” i upuszczamy go: **UWAGA!** Na przycisku „**Generate**”. Czyli nasz przycisk będzie odpowiadał za Timer.

Powinniśmy też ustawić Interval naszego Timer'a. W tym celu klikamy na nasz „dodany” już Timer, i w polu „**Interval**” zmieniamy tego wartość np. na „**200**” – 20 sec. Tak jak na screenie poniżej.



2.2 – Kodujemy prosty generator

Przejdźmy do kodowania. Klikamy dwukrotnie na „Timer1” i wklejamy tam ten oto kod:

```
ProgressBar1.Increment(1)
If ProgressBar1.Value = ProgressBar1.Maximum Then
    Timer1.Stop()
    Dim key As Integer
    key = (Rnd() * 3)
    Select Case key
        Case 1
            TextBox1.Text = "AG7V3BG895SR"
        Case 2
            TextBox1.Text = "NS2OP5XV5B11"
        Case 3
            TextBox1.Text = "41S9VBG7YT2P"
    End Select
End If
```

Ok, kod wklejony, ale przecież już dodawałem taki sam kod przy buttonie..? O tym wspominałem wcześniej, musimy nieco zmodyfikować nasz kod. Tym razem to Timer ma decydować, kiedy pokaże się kod. Przecież nie chcemy sytuacji, w której kod się pojawia, zanim załaduje się pasek, prawda?



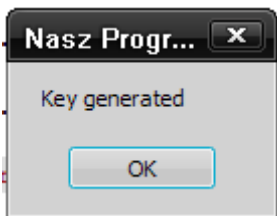
Lecz to nie wszystko, teraz przecież musimy edytować nasz przycisk **"Generate"**
Oto kod:

```
Timer1.Start()  
ProgressBar1.Value = ProgressBar1.Minimum
```

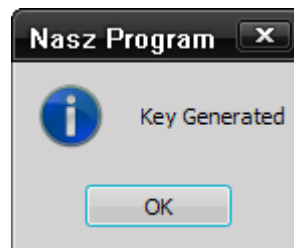
Sprawdzamy czy nasz pasek ładowania działa - „F5”

2.2.1 – Kodujemy prosty generator

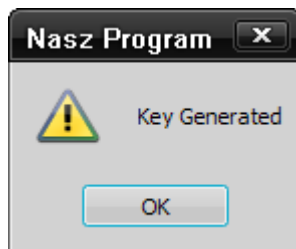
Nasz program powinien działać, ale wzbogaćmy go jeszcze o komunikat, który potwierdzi „że kod został wygenerowany”. Później będziemy często używać komunikatów w celu napisania np. „Cash Added”, „Account Hacked” itp. Zanim podam gotowy kod, jaki należy wkleić do Timer'a muszę zaznaczyć, że nasz komunikat może wyglądać różnie, zależnie od kodu. Przykładowo



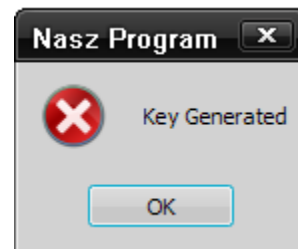
```
MsgBox("Key generated")
```



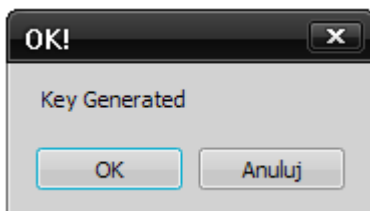
```
MsgBox("Key Generated", MsgBoxStyle.Information)
```



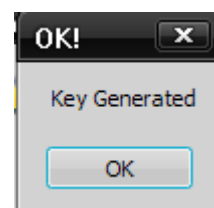
```
MsgBox("Key Generated", MsgBoxStyle.Critical)
```



```
MsgBox("Key Generated", MsgBoxStyle.Exclamation)
```



```
MessageBox.Show("Key Generated", "OK!", MessageBoxButtons.OKCancel)
```



```
MessageBox.Show("Key Generated", "OK!")
```

Możemy sobie pokombinować, ja jednak zostanę przy „`MessageBox.Show("Key Generated", "OK!")`” i zamiast OK, napiszę SUCCESS!.



A tak wygląda na chwilę obecną mój kod do „Timer'a”

```
ProgressBar1.Increment(1)
If ProgressBar1.Value = ProgressBar1.Maximum Then
    Timer1.Stop()
    Dim key As Integer
    key = (Rnd() * 3)
    Select Case key
        Case 1
            TextBox1.Text = "AG7V3BG895SR"
        Case 2
            TextBox1.Text = "NS2OP5XV5B11"
        Case 3
            TextBox1.Text = "41S9VBG7YT2P"
    End Select
    MessageBox.Show("Key Generated", "Success!")
End If
```

Po wklejeniu tego kodu do programu naciskamy „**F5**” i możemy być już zadowoleni z pierwszego „generatora” mimo że jego zasada działania jest banalna. Niestety klucze zawsze będą takie same, ale można użyć „**Case'ów**” do prawdziwych generatorów, które generują działające klucze. Następna część poradnika będzie jeszcze brała pod uwagę generatory kluczy, ale będą już one bardziej praktyczniejsze, i użyte zostaną w nich bardziej profesjonalne opcje.



Kodujemy bardziej zaawansowany keygen

Kilka słów wstępu..

Zapewne kupiłeś/-aś ten poradnik, aby móc samodzielnie pisać keygeny, generatory czy inne programy, a później dzięki promowaniu czerpać z nich niezłą kasę. Ja właśnie chcę Tobie to umożliwić. Nie mogę jednak Ci zagwarantować, że to co stworzysz będzie miało popyt. Więc proszę nie składać później żadnych zażaleń. Natomiast jeżeli będziecie mieli jakieś pytania, prośby, czy inne zachcianki odsyłam na forumowe PW.

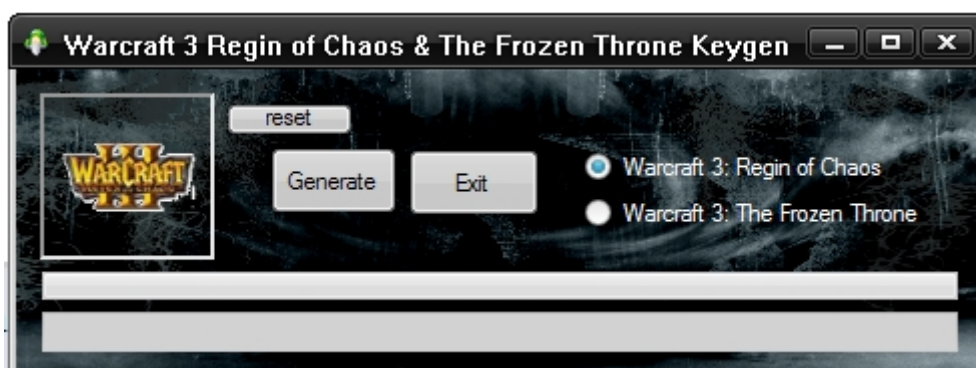
Poświęciłem swój czas na napisanie tego poradnika, więc zabraniam kopiowania jego treści, sprzedaży jego treści czy pół-treści, wymieniania się nim, szantażowania nim Microsoft bez mojej zgody. Powstał on w celu edukacyjnym, podczas pisania nie ucierpiały żadne zwierzęta, a cały tekst został napisany przez profesjonalistów, proszę nie powtarzać tego w domu.

ROCKNDRUG

3.0 – Kodujemy bardziej zaawansowany keygen

Mamy gotowy nasz pierwszy keygen, mimo jego prostoty można spokojnie już na nim zarobić. Jednak nie będziemy tworzyć tak banalnych aplikacji. Aby program wyglądał realistycznie musimy dodać mu bardziej profesjonalne funkcje, ale nie tracąc przy tym na „wyglądzie” przecież z góry wiadomo że większość ludzi to wzrokowcy, jeżeli nie spodoba im się „dizajn” naszego programu, bo zasłonimy twarz postaci z Assassins Creed naszym buttonem, a resztę przysłonimy textboxem to wiadomo że nikt tego nie ściągnie. (zdarzają się wyjątki, jak to w tym „sporcie”)

Po przeczytaniu poprzednich części, wiecie już jak „zaprojektować” keygen. W tej części stworzę nowy, żeby nie pracować tylko na jednym. Zróbmy generator kluczy go gry, w którą grałem nocami a mianowicie „Warcraft 3”. Czyli zaczniemy tego, że tworzymy nowy projekt, dodajemy tło, nazwę programu, ikonę itp. Ja już to zrobiłem, a oto efekt mojej pracy (było robione na szybko)





3.1 – Kodujemy bardziej zaawansowany keygen

Jak widzicie teraz użytkownik musi wybrać, jaką wersję gry posiada. Użyłem do tego narzędzia z „Toolbox’a” o nazwie „RadioButton”, odpowiednio ustawiłem przezroczystość, kolor tekstu, położenie itp. Lecz nie możemy użyć takiego samego kodu jak w poprzedniej wersji poradnika. Teraz klucz musi być zależny od tego, jaką wersję ktoś wybierze. Kod dla przycisku Exit pozostaje bez zmian, lecz Generate musi przejść mały „tuning”:

Przycisk Generate:

```
    If RadioButton1.Checked = False And  
        RadioButton2.Checked = False Then  
        MessageBox.Show("Choose game version", "Error")  
    Else  
        Timer1.Start()  
        ProgressBar1.Value = ProgressBar1.Minimum  
        If RadioButton1.Checked Then  
            RadioButton2.Enabled = False  
        End If  
        If RadioButton2.Checked Then  
            RadioButton1.Enabled = False  
        End If  
    End If
```

.Checked pozwala nam na sprawdzenie czy dany „Radiobutton” został zaznaczony. True – prawda, False – fałsz. Czyli logicznie True – zaznaczony, False – niezaznaczony. Musimy użyć tutaj warunku „If” jeżeli jeden zostanie zaznaczony, wtedy drugi zostanie wyłączony „za to odpowiada .Enabled = False”

Przycisk Exit:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button2.Click  
    End ← dopisujemy tylko end, przecież aplikacja ma zostać zamknięta.  
End Sub
```

Przycisk Reset:

```
ProgressBar1.Value = ProgressBar1.Minimum  
If RadioButton1.Checked Then  
    RadioButton2.Enabled = True  
End If  
If RadioButton2.Checked Then  
    RadioButton1.Enabled = True  
End If
```

Ta część kodu pozwala nam na „zresetowanie” paska postępu. Także jego wartość musimy zredukować do minimum. Następnie musimy wziąć pod uwagę, to że ktoś wybrał jakiś „Radiobutton”, jeżeli program już się wykonał to jeden z nich będzie wyłączony. Musimy „oba” włączyć. Robimy to warunkiem „If”. Jeżeli Radiobutton1.zaznaczony to Radiobutton1.włączony = prawda oraz jeżeli Radiobutton2.zaznaczony to Radiobutton2.włączony = prawda.

Dzięki takiemu kodowaniu, program w ogóle nie zacznie działać, jeżeli user nie wybierze „wersji” gry. Musi wybrać któryś radiobutton. Dodatkowo, jeżeli wybierze „The Frozen Throne” i naciśnie generate, to nie będzie mógł w trakcie wybrać już tego „drugiego”. Po naciśnięciu „reset” będzie mógł powtórzyć całą zabawę od początku.



3.2 – Kodujemy bardziej zaawansowany keygen

Pora na zakodowanie Timer'a. Pokażę wam jak stworzyć „keygen” który będzie losowo tworzył klucze, oraz keygen, który będzie tylko wybierał klucze, jakie mu wcześniej podaliście. Pierwsza opcja jest wygodniejsza – wiadomo, nie trzeba szukać po sieci działających kluczy, natomiast jeżeli chcemy stworzyć keygen, który naprawdę działa ta opcja mija się z celem...

Skupmy się na razie, na stworzeniu losowego klucza z tekstu. Aby tego dokonać, odpowiednio kodujemy nasz timer, oto gotowy kod:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
    ProgressBar1.Increment(1)
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Timer1.Stop()
        MessageBox.Show("Key generated! Have Fun!", "Success!")
        If RadioButton1.Checked Then
losowanie_klucza:
            Dim AA As Double = Int(Rnd() * 9999)
            Dim BB As Double = Int(Rnd() * 9999)
            Dim CC As Double = Int(Rnd() * 9999)
            Dim DD As Double = Int(Rnd() * 9999)
            Dim EE As Double = Int(Rnd() * 9999)
            If AA > 1 And BB > 1 And CC > 1 And DD > 1 And EE > 1 Then
                If AA + BB + CC + DD + EE = 25560 Then
Dim klucz As String = "Your key: " & AA & "-" & BB & "-" & CC & "-" & DD & "-" & EE
                    klucz = klucz.Replace("2", "G")
                    klucz = klucz.Replace("4", "Z")
                    klucz = klucz.Replace("6", "S")
                    klucz = klucz.Replace("8", "N")
                    TextBox1.Text = klucz
                    Exit Sub
                Else
                    GoTo losowanie_klucza
                End If
            Else
                GoTo losowanie_klucza
            End If
        End If
    End If
End Sub
```

Jeżeli nie pracowałeś wcześniej nigdy w Visual Basic, to zapewne po tym jak zobaczyłeś ten kod musiałeś być trochę przerażony. Ale zapewniam że nie ma się czego bać. Zacznę tłumaczyć od:

„If RadioButton1.Checked Then”, a więc jeżeli ktoś zaznaczy ten RadioButton1 to..

I tutaj dość długo myślałem jak mogę to w „prosty” sposób wytłumaczyć. Doszedłem do wniosku że zrobię tak. Dim ← czyli przypisanie „czemuś” - „coś”. Czyli Dim AA As Double będzie oznaczało że AA zostaje przypisane jako „Double” czyli zmienna liczba rzeczywista (więcej info nt. Rzeczywistych i ich przedziałów tutaj: http://www.e-programme.info/index.php?show=kurs_zmienne_i_stale). Natomiast część po „=” mówi nam, że ma zostać wylosowana liczba z *99999. Logicznie myśląc, to każdy nasz dodany element „AA, BB, CC, DD, EE” będzie losowo wybraną liczbą. Dalej, jeżeli AA jest większe od 1 (tego nie muszę chyba tłumaczyć). Następnie nasze liczby zostają dodane, i jeżeli ich suma wyniesie 25560 (takie zabezpieczenie, aby liczby nie były małe) to przypisz „klucz” jako String (czyli jako łańcuch znaków), który ma wyglądać następująco: "Your key: " & AA & "-" & BB & "-" & CC & "-" & DD & "-" & EE (może on wyglądać tak: 3423 – 2365 – 9343 itd.) A następne cztery linijki kodu mają za zadanie zamienić cyfry, na litery. (klucze seryjne z reguły mają litery mieszane z cyframi). Później cały „klucz” ma zostać wprowadzony do TextBox'a. I na tym kończy się losowanie, w przeciwnym wypadku całość się zapętla (wraca do punktu, w którym mamy napisane „losowanie klucza:” i ta część wykonywana jest aż do skutku.



3.2.1 – Kodujemy bardziej zaawansowany keygen

To samo robimy z „Radiobutton2” czyli naszym „Warcraft 3: The Frozen Throne” jeżeli chcemy, aby nie wyskakiwały podobne klucze, możemy zmienić kilka wartości np. 9999 na 8888. Cały kod Timer'a powinien wyglądać następująco:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
    ProgressBar1.Increment(1)
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Timer1.Stop()
        MessageBox.Show("Key generated! Have Fun!", "Success!")
        If RadioButton1.Checked Then
losowanie_klucza:
            Dim AA As Double = Int(Rnd() * 9999)
            Dim BB As Double = Int(Rnd() * 9999)
            Dim CC As Double = Int(Rnd() * 9999)
            Dim DD As Double = Int(Rnd() * 9999)
            Dim EE As Double = Int(Rnd() * 9999)
            If AA > 1 And BB > 1 And CC > 1 And DD > 1 And EE > 1 Then
                If AA + BB + CC + DD + EE = 35560 Then
                    Dim klucz As String = "Your key: " & AA & "-" & BB & "-" & CC & "-" & DD & "-" & EE
                    klucz = klucz.Replace("2", "G")
                    klucz = klucz.Replace("4", "Z")
                    klucz = klucz.Replace("6", "S")
                    klucz = klucz.Replace("8", "N")
                    TextBox1.Text = klucz
                    Exit Sub
                Else
                    GoTo losowanie_klucza
                End If
            Else
                GoTo losowanie_klucza
            End If
        Else
            End If
        ProgressBar1.Increment(1)
        If ProgressBar1.Value = ProgressBar1.Maximum Then
            Timer1.Stop()
            If RadioButton2.Checked Then ← tutaj zaczyna się RadioButton2
losowanie_klucza2:
                Dim AA As Double = Int(Rnd() * 9999)
                Dim BB As Double = Int(Rnd() * 9999)
                Dim CC As Double = Int(Rnd() * 9999)
                Dim DD As Double = Int(Rnd() * 9999)
                Dim EE As Double = Int(Rnd() * 9999)
                If AA > 1 And BB > 1 And CC > 1 And DD > 1 And EE > 1 Then
                    If AA + BB + CC + DD + EE = 38560 Then ← zmieniłem sumę liczb
                        Dim klucz2 As String = "Your key: " & AA & "-" & BB & "-" & CC & "-" & DD & "-" & EE
                        klucz2 = klucz2.Replace("3", "F") ← zmieniłem troszkę liczby, które
                        klucz2 = klucz2.Replace("4", "Z") mają się wymieniać
                        klucz2 = klucz2.Replace("5", "T")
                        klucz2 = klucz2.Replace("8", "N")
                        TextBox1.Text = klucz2
                        Exit Sub
                    Else
                        GoTo losowanie_klucza2
                    End If
                Else
                    GoTo losowanie_klucza2
                End If
            End If
        End If
    End If
End Sub
```



3.3 – Kodujemy bardziej zaawansowany keygen

Nasz keygen działa! Losuje losowy klucz, po czym możemy go skopiować, w razie pomyłki naciskamy reset i wybieramy drugie rozszerzenie gry. Ale jak pisałem wcześniej, szansę że zostanie wylosowany klucz, który rzeczywiście będzie działał są znikome. Więc musimy mieć też stworzyć działające keygeny. (nikt z góry nie ustalił, że wszystkie nasze programy to muszą być fejki :))

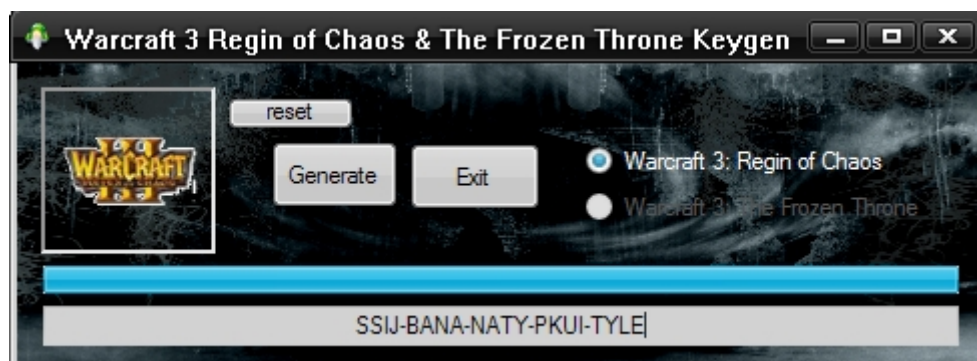
Aby „stworzyć” działający keygen posłużę się kodem, którego używaliśmy w części drugiej. Mała modyfikacja, i będzie on się nadawał. Oto on:

```
Dim key As Integer
key = (Rnd() * 3 + 1)
Select Case key
    Case 1
        TextBox1.Text = "AG7V3BG895SR"
    Case 2
        TextBox1.Text = "NS2OP5XV5B11"
    Case 3
        TextBox1.Text = "41S9VVG7YT2P"
End Select
```

Na początku zmodyfikujmy go tylko na potrzeby „RadioButton'a1”. Wtedy kod timer'a będzie o wiele „prostszy”

```
ProgressBar1.Increment(1)
If ProgressBar1.Value = ProgressBar1.Maximum Then
    Timer1.Stop()
    MessageBox.Show("Key generated! Have Fun!", "Success!")
    If RadioButton1.Checked Then
        Dim key As Integer
        key = (Rnd() * 3 + 1)
        Select Case key
            Case 1
                TextBox1.Text = "Your Code: "[podajemy działający klucz]"
            Case 2
                TextBox1.Text = "Your Code: "[np FS23-35CS-CS26-TY6T-3SAX]"
            Case 3
                TextBox1.Text = "Your Code: inny działający klucz"
        End Select
    End If
End If
End Sub
```

Oczywiście wiemy co robić, jak chcemy podać więcej kluczy(było podawane w cz.2) A w praktyce wygląda to tak:





Teraz pozostaje nam tylko dopisać do Timer'a kod dla „Radiobutton2”, podać działające key'e i mamy z głowy. Tak wygląda cały kod:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Increment(1)
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Timer1.Stop()
        MessageBox.Show("Key generated! Have Fun!", "Success!")
        If RadioButton1.Checked Then
            Dim key As Integer ← („key” będzie teraz liczbą rzeczywistą
            key = (Rnd() * 3+1) tutaj key, losowana jest jako jedna z 4 liczb
            Select Case key      (0,1,3,4) +1 jest po to, aby zapobiec 0. bo 0
                Case 1          nie uwzględniliśmy w Case :)
                    TextBox1.Text = "AG45-SFXV-SWTF-W4FV-SO2A"
                Case 2
                    TextBox1.Text = "VS47-S21A-BG3T-ESKL-WYPL"
                Case 3
                    TextBox1.Text = "SSIJ-BANA-NATY-PKUI-TYLE"
            End Select
        Else
            MessageBox.Show("Key generated! Have Fun!", "Success!")
            If RadioButton2.Checked Then
                Dim key2 As Integer
                key2 = Int(Rnd() * 3 + 1) ← to samo, tylko że z key2
                Select Case key2
                    Case 1
                        TextBox1.Text = "G335-SEXX-TO3S-W4SO-2AFV"
                    Case 2
                        TextBox1.Text = "VS47-T4PG-BDVT-ESKA-W094"
                    Case 3
                        TextBox1.Text = "NAWE-TNIE-DLUB-WN05-1EFU"
                End Select
            End If
        End If
    End If
End Sub
```

Nasz generator powinien działać. Używanie „RadioButton’ów” powinniśmy mieć opanowane. Jest to bardzo użyteczne narzędzie jeżeli chcemy dodać jakiś „wybór”. Często stosowane są te buttony do paypal hacków, lub itunes. Generator kluczy do Warcrafta to tylko przykładowa aplikacja, jeżeli chcesz zarabiać to twórz programy do gier, które dopiero pojawiają się na torrentach. Na stronach takich jak www.epidemz.net , www.warez-bb.org , lub sprawdzaj premiery gier na stronie www.ultima.pl



3.4 – Kodujemy bardziej zaawansowany keygen

Nasz keygen działa! Możemy urozmaicić troszkę wygląd naszej aplikacji, i dodać jej wczytywanie w „%”. User będzie wiedział w jak szybkim tempie ładuje się aplikacja. Całkiem przydatne dla efektu wizualnego, później będziemy mogli bardzo ciekawie jeszcze to wykorzystać.

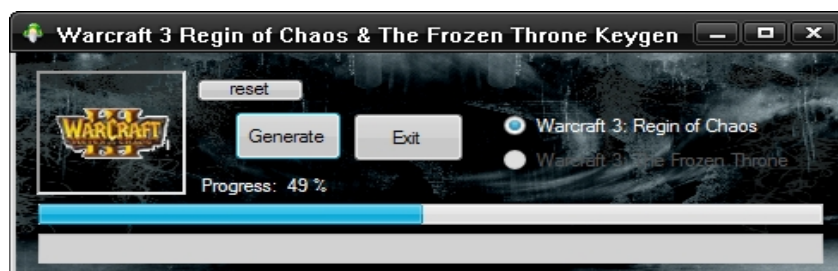
Aby zaprogramować wczytywanie procentowe do naszego „Timer'a” przed „End Sub” czyli praktycznie na końcu dodajemy jedną linijkę kodu:

```
Label1.Text = ProgressBar1.Value & " %" ← Czyli tekst label1 będzie  
End Sub                                odpowiadał wartości paska progress
```

Natomiast do przycisku „Reset” dopisujemy:

```
Label1.Text = "" ← czyli po kliknięciu w button Reset, label1 będzie puste
```

Żeby wszystko wyglądało i działało jak należy, musimy dodać co najmniej 2x „Label”. Pierwszy podpisać jako „Progress” czy jak tam chcemy, a drugi ma służyć jako „nasze wczytywanie w %”. U mnie label1 posłużył jako procenty. Wcześniej jego „TEXT” ustawiłem tylko na spację, więc nie widać go pierwotnie w programie.



3.5 – Kodujemy bardziej zaawansowany keygen

Pora na podjęcie kroków, które mogą pomóc nam w zwiększeniu zarobku! Pierwszą rzeczą będzie odnośnik do waszej strony w generatorze. Ja wymyśliłem jakiś Team „NaviRed Team”, można też dopisać sobie że to zespół który zajmują się profesjonalnym crackowaniem gier czy nawymyślać coś innego. Ma to na celu zachęcić ludzi, aby odwiedzili stronę tego zespołu. Wy natomiast zamiast tworzyć bloga na blogspocie o nazwie megahackscracksmorefun.blogspot.com. Robicie sobie stronę o nazwie NaviRed-Team i tam wrzucacie wszystkie wasze „hacki”. Ludzie pomyślą że jesteście wiarygodni i pobiorą Twój generator. Dobrym pomysłem jest „wejście” na waszą stronę w trakcie kręcenia filmiku. Kiedy wasz klucz się „generuje” wchodzicie na stronę, dodajecie jakiś tekst żeby inni też odwiedzili (można też tak dorabiać sobie reklamami na ad-sense) No i dodajemy odnośnik do naszego generatora i strony pod filmikiem.

Przykładowo:

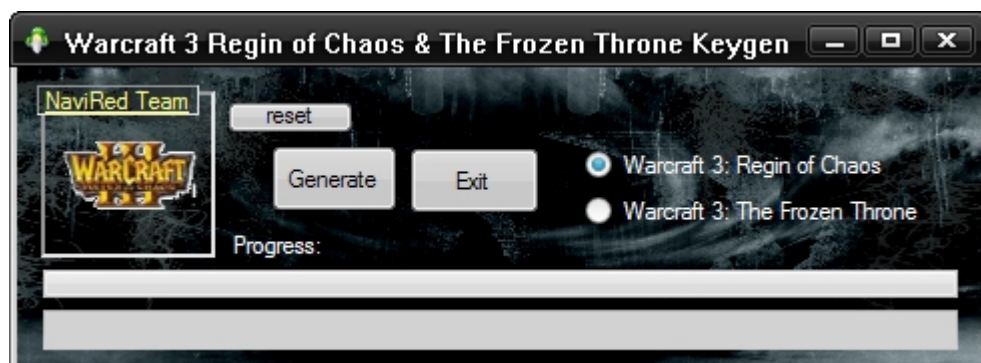
Direct Download: <http://bit.ly/W3GenNaviRed>
or
Download from our Site: <http://NaviRedTeam.tk>



Aby dodać odnośnik do naszej strony wybieramy z menu „Toolbox'a” narzędzie „LinkLabel” i umieszczamy je gdzie chcemy. Ja polecam „rogi” programu. Ja ustawiłem „BorderStyle” na 3d, kolor, przezroczystość i troszkę go wydłużyłem. No tak, ale on wciąż nie działa, trzeba go zaprogramować. W tym celu klikamy dwukrotnie w nasz „LinkLabel” i po przejściu do naszego edytora dopisujemy coś takiego:

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
    System.Diagnostics.Process.Start("http://NaviRedTeam.com")
End Sub
```

Czyli krótko mówiąc, jeżeli ktoś kliknie w nasz link, zostanie przeniesiony do „NaviRedTeam.com”. Pamiętajcie o „http://” bez tego nasz link nie będzie działał.



3.6 – Kodujemy bardziej zaawansowany keygen

W trakcie czytania następných części, poznasz nowe narzędzia oraz nauczysz się je wykorzystywać. Zaznaczę tylko, że jeżeli umiesz wykorzystać już „RadioButton” to nie będziesz miał większych problemów z wykorzystaniem „CheckBox'a” z naszego „ToolBox'a”. Programowanie go będzie opierało się ja takiej samej zasadzie.



Pora na generator Kilka słów wstępu..

Zapewne kupiłeś/-aś ten poradnik, aby móc samodzielnie pisać keygeny, generatory czy inne programy, a później dzięki promowaniu czerpać z nich niezłą kasę. Ja właśnie chcę Tobie to umożliwić. Nie mogę jednak Ci zagwarantować, że to co stworzysz będzie miało popyt. Więc proszę nie składać później żadnych zażaleń. Natomiast jeżeli będziecie mieli jakieś pytania, prośby, czy inne zachcianki odsyłam na forumowe PW.

Poświęciłem swój czas na napisanie tego poradnika, więc zabraniam kopiowania jego treści, sprzedaży jego treści czy pół-treści, wymieniania się nim, szantażowania nim Microsoft bez mojej zgody. Powstał on w celu edukacyjnym, podczas pisania nie ucierpiały żadne zwierzęta, a cały tekst został napisany przez profesjonalistów, proszę nie powtarzać tego w domu.

ROCKNDRUG

4.0 – Pora na generator

Potrafimy już napisać keygen, ale samo wrzucanie keygenów może okazać się mało efektywne. Nadszedł czas abyśmy zaczęli tworzyć coś, czego ludzie będą szukać bo chcą być lepsi od innych na tyle, że sięgną do „programów trzecich” które znacznie ułatwią im zdobycie tego celu. No tak, ale mówiąc generator muszę mieć coś konkretnego na myśli, nie zrobię przecież generatora do wszystkiego (dlaczego nie :p). Zaczniemy od Generatora Gold'a go gier MMORPG. Na to zawsze jest popyt, w szczególności jeśli chodzi o produkcje, które posiadają rzesze fanów. Mowa o World of Warcraft, Silkroad, Metin2, Lineage 2, Aion, Flyff i wiele innych. Lecz z doświadczenia wiem, że nowe gry, te które mają niedługo premierę, są w fazie Closed-Open Beta też cieszą się sporym zainteresowaniem na yt. Warto tutaj zaznaczyć że nikt inny nie wrzuca przykładowych generatorów do takich gier, więc jest duże prawdopodobieństwo że będziecie jednymi userami na youtube.

Nasz generator powinien brać pod uwagę realne nazwy serwerów, walutę w grze [nie w każdej walutą jest „gold”, a może raczej powinienem powiedzieć że w prawie żadnej] Dla zaprezentowania wam generatora do gry MMORPG wybrałem produkcję „Cabal Online”. Dlaczego? Posiada bardzo dużą liczbę graczy i przyciąga nowych, cały czas jest rozwijana.



4.0.1 – Pora na generator

Generator, który widzicie poniżej wykonałem dosyć szybko. Screen przedstawia odpalony proces „dodawania ALZ – waluty w grze” do postaci w grze. I tutaj kolejna uwaga.. Bardzo dużo użytkowników youtube sprawdza, czy trzeba podać hasło do konta w takich generatorach, przez co większość jeżeli widzi takie pole do wpisania to rezygnuje z pobierania z obawy o własne konto.



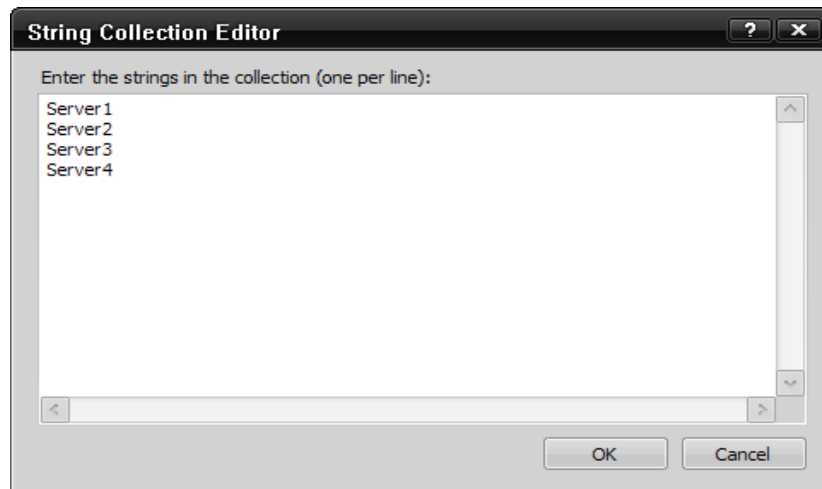
Jak widzicie dodałem „**TextBox**”, dwa „**Button'y**”, „**ProgressBar**” „**LinkLabel**” itd. Ale waszą uwagę powinny przykuć tzw. „listy” wyboru. Te dwa narzędzia, które widnieją na screenie pod nazwą „**Server**” oraz „**Alz Amount**” nazywają się **ComboBox'y**. Ich użycie jest bardzo przydatne, jeśli chodzi o generatory tego typu, można wybrać coś z bardzo długiej listy, nie tracąc przy tym wiele miejsca w naszym programie. Kolejną rzeczą jaką powinna was zainteresować jest „**Status: Connecting to game server..**”. Ten „**bajer**” bardzo pomaga nam zdobyć serca user'ów którzy oglądają filmik, wiem to z doświadczenia.

Na screenie tego nie widać, ale można sobie odpalić aplikację z folderu (dodałem wszystko do paczki z poradnikiem) i zobaczyć jak działa program. Otóż nie zadziała on, jeżeli użytkownik zostawi jakieś pole puste. Nie wpiszę nazwy postaci, nie wybierze serwera czy ilości ALZ jaką ma dostać.



4.1 – Pora na generator

Zacznijemy od „**ComboBox'a**”. Wybieramy go z menu „**Toolbox**” i upuszczamy na nasz projekt. Po umieszczeniu go w projekcie, i kliknięciu możemy dostrzec taką małą, białą strzałkę nad jego „górną” krawędzią. Klikamy w nią, a następnie w „**Edit Items**”. Pojawi nam się okienko, w które powinniśmy wpisać nazwę serwerów. Jedna linijka = jeden serwer. Wpisujemy kilka nazw serwerów, po czym w „**Properties**” ustawiamy wartość pola „**DropDownStyle**” na „**DropDownList**”, aby nikt nie mógł pisać po naszym „**ComboBox'ie**”. Możemy też pobawić się wyglądem tego ComboBox'a. **BorderStyle**, zmienić czcionkę..



^Tak mają wyglądać wpisane nazwy serwerów

4.1.1 – Pora na generator

Mamy dodany „**ComboBox**”, po naciśnięciu „**F5**” możemy wywołać test naszego programu i sprawdzić czy działa on jak należy. Teraz musimy powiązać wybór użytkownika z całym programem. I tak zaprogramować aplikację, aby w ogóle nie działała, jeżeli użytkownik nie wybierze żadnego serwera, nie wpisze nazwy postaci czy ilości ALZ, jaką ma dostać. Zacznijemy od przycisku „**START**”

Czysty kod:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    Dim a As Double
    a = 0
    If TextBox1.Text = "" Then
        MessageBox.Show("Put Character name!", "Error")
    Else
        a = (a + 1)
    End If
    If ComboBox1.Text = "" Then
        MessageBox.Show("Select Server!", "Error")
    Else
        a = (a + 1)
    End If
    If ComboBox2.Text = "" Then
        MessageBox.Show("Select ALZ Amount!", "Error")
    Else
        a = (a + 1)
    End If
    If a = (3) Then
        Timer1.Start()
        ProgressBar1.Value = ProgressBar1.Minimum
    End If
End Sub
```



4.1.1 – Pora na generator

Mamy dodany „**ComboBox**”, po naciśnięciu „**F5**” możemy wywołać test naszego programu i sprawdzić czy działa on należy. Teraz musimy powiązać wybór użytkownika z całym programem. I tak zaprogramować aplikację, aby w ogóle nie działała, jeżeli użytkownik nie wybierze żadnego serwera, nie wpisze nazwy postaci czy ilości ALZ, jaką ma dostać. Zaczniemy od przycisku „**START**”

Opisany kod:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim a As Double ← przypisujemy a jako liczbę rzeczywistą
    a = 0 ← a równa się na samym początku zero = 0
    If TextBox1.Text = "" Then ← jeżeli textbox będzie pusty wtedy
        MessageBox.Show("Put Character name!", "Error") ← pokazuję się błąd
    Else ← w przeciwnym wypadku (else) do naszego a zostanie dodane 1 (a+1)
        a = (a + 1)
    End If
    If ComboBox1.Text = "" Then ← tak samo jak poprzednio.
        MessageBox.Show("Select Server!", "Error")
    Else
        a = (a + 1) ← tym razem, jeżeli poprzedni warunek został spełniony (a) to
    End If a będzie wynosiło (1), po dodaniu będzie to już 2 (a=a+1)
    If ComboBox2.Text = "" Then
        MessageBox.Show("Select ALZ Amount!", "Error")
    Else
        a = (a + 1) ← teraz trzecie sprawdzenie warunku, jeżeli wszystko poszło OK to
    End If a powinno równać się 3.
    If a = (3) Then ← jeżeli a jest równe 3 to Timer zaczyna działać.
        Timer1.Start()
        ProgressBar1.Value = ProgressBar1.Minimum
    End If
End Sub
```

Aby wszystko działało jak należy musieliśmy dodać pewien warunek. Jeżeli zostanie on spełniony, dopiero wtedy program zacznie działać. Tutaj tym warunkiem jest „If a = (3) Then” czyli jeżeli a jest równe 3. Samo „a” będzie równe tej liczbie dopiero wtedy, kiedy spełnione zostaną trzy inne warunki, czyli w każdym polu będzie coś wybrane, wpisane.

Przy kopiowaniu i wklejaniu kodu ode mnie, do waszego programu musicie też uwzględnić to, że ja np. użyłem TextBox'a i 2 ComboBox'ów, Jeżeli wy użyjecie np. 1 ComboBox'a i 1 TextBox'a będziecie musieli przerobić kod na:

```
Dim a As Double
a = 0
If TextBox1.Text = "" Then
    MessageBox.Show("Put Character name!", "Error")
Else
    a = (a + 1)
End If
If ComboBox1.Text = "" Then
    MessageBox.Show("Select Server!", "Error")
Else
    a = (a + 1)
End If
If a = (2) Then
    Timer1.Start()
    ProgressBar1.Value = ProgressBar1.Minimum
End If
```

Jeżeli ktoś będzie miał z tym kłopot, proszę o podanie GG na PW. Pomogę :)



4.2 – Pora na generator

Pora na nasz status, czyli chcemy aby program pokazywał, co aktualnie się wykonuje (np. Trwa Logowanie do serwera facebook.com.. Proszę Czekać :D) Kod sam w sobie nie jest bardzo skomplikowany, jeżeli do tej pory rozumiałeś „cokolwiek” z poprzednich kodów to sądzę że nie będziesz miał/-a problemów z tym:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
    ProgressBar1.Increment(1)
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Timer1.Stop()
        Dim key As Integer
        key = (Rnd() * 1)
        Select Case key
            Case 1
        End Select
        MessageBox.Show("Please check your gold in-game", "Success")
    Else
        If ProgressBar1.Value = 1 Then
            Label7.Text = "Loading.."
        End If
        If ProgressBar1.Value = 15 Then [czyli kiedy wartość paska będzie 15 to
            Label7.Text = "Generator loaded!" w label7 pojawi się tekst w "]
        End If
        If ProgressBar1.Value = 25 Then
            Label7.Text = "Connecting to game server.."
        End If
        If ProgressBar1.Value = 45 Then
            Label7.Text = "Connected!"
        End If
        If ProgressBar1.Value = 55 Then
            Label7.Text = "Searching character on selected server.."
        End If
        If ProgressBar1.Value = 80 Then
            Label7.Text = "Characted found!"
        End If
        If ProgressBar1.Value = 86 Then
            Label7.Text = "Transfer ALZ to character.."
        End If
        If ProgressBar1.Value = 95 Then
            Label7.Text = "Success!"
        End If
    End If
End Sub
```

Początek taki, jak w poprzednich kodach. Zmiana następuje dopiero od linijki MessageBox.Show. Otóż cała magia działania tego kodu, polega na tym że jeżeli wartość paska ładowania będzie równa np. 25, wtedy w Label ma się pojawić „Connecting to game server.". Teraz wystarczy ustawić tylko odpowiedni interwał naszego timera, ustawić odpowiednie wartości oraz tekst, połączyć z odpowiednim „labelem” i mamy gotowy status :)



Upiększamy naszą aplikację **Kilka słów wstępu..**

Zapewne kupiłeś/-aś ten poradnik, aby móc samodzielnie pisać keygeny, generatory czy inne programy, a później dzięki promowaniu czerpać z nich niezłą kasę. Ja właśnie chcę Tobie to umożliwić. Nie mogę jednak Ci zagwarantować, że to co stworzysz będzie miało popyt. Więc proszę nie składać później żadnych zażaleń. Natomiast jeżeli będziecie mieli jakieś pytania, prośby, czy inne zachcianki odsyłam na forumowe PW.

Poświęciłem swój czas na napisanie tego poradnika, więc zabraniam kopiowania jego treści, sprzedaży jego treści czy pół-treści, wymieniania się nim, szantażowania nim Microsoft bez mojej zgody. Powstał on w celu edukacyjnym, podczas pisania nie ucierpiały żadne zwierzęta, a cały tekst został napisany przez profesjonalistów, proszę nie powtarzać tego w domu.

ROCKDRUG

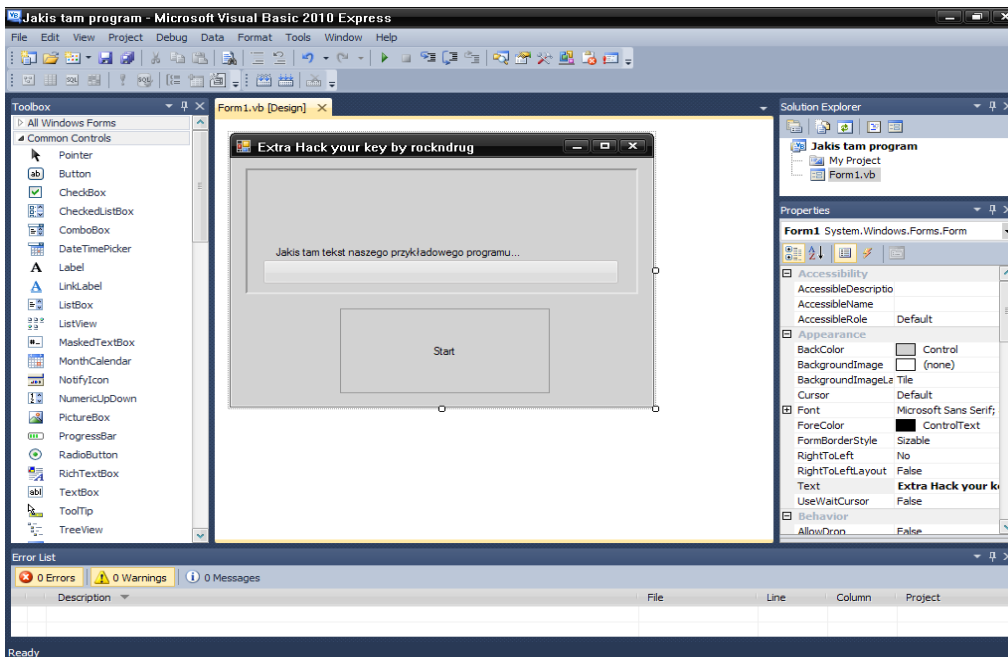
5.0 – Upiększamy naszą aplikację

Po dokładnym przestudiowaniu poprzednich części nie powinniśmy mieć problemów z tworzeniem keygena, generatora czy hacka. Teraz chciałbym pokazać kilka ciekawych „tricków” które pomogą nam upiększyć nasz program, i co z tym idzie.. zwiększyć nasz zarobek. Jest to ostatnia część dla tej wersji. Przewidywana jest w przyszłym czasie bardziej rozbudowana wersja, ale z braku czasu nie mam jak się za nią zabrać. Oczywiście Ci którzy kupili tą wersję nie będą musieli wcale dopłacać, lub będą mieli super zniżkę.

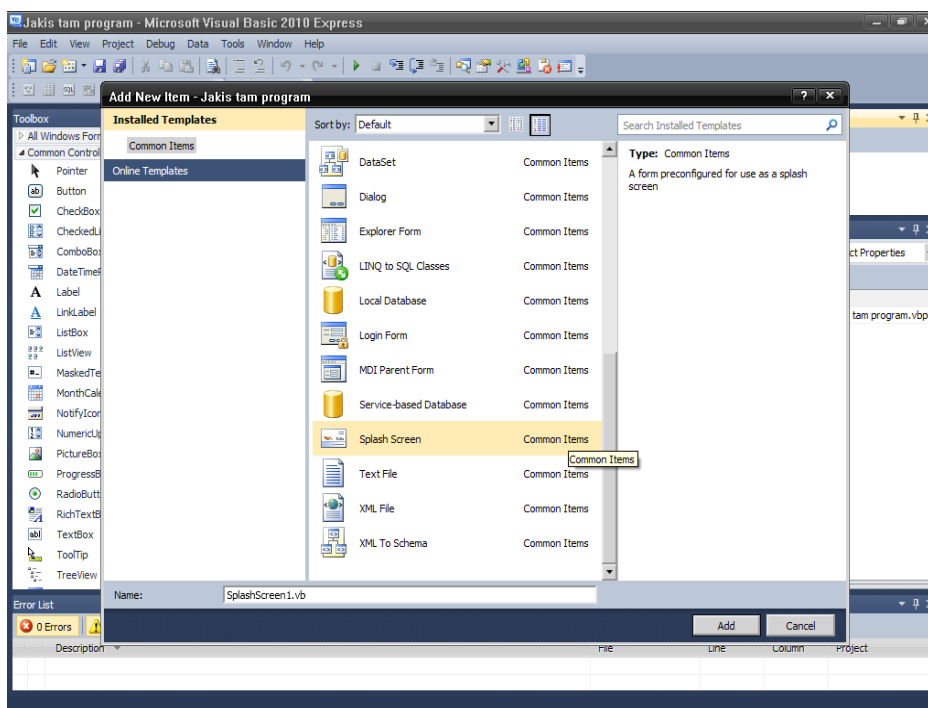


5.0.1 – Upiększamy naszą aplikację

A więc zaczynamy, stworzyłem sobie w minutę taki prosty programik, dodałem button, jakiś tekst itp. Ale nie sęć teraz w jego funkcjonalności.. Skupmy się teraz na polu Solution Explorer. Klikamy prawym przyciskiem na nazwę naszego projektu (u mnie jest to „Jakis tam program”) i klikamy na **ADD** → **New item**



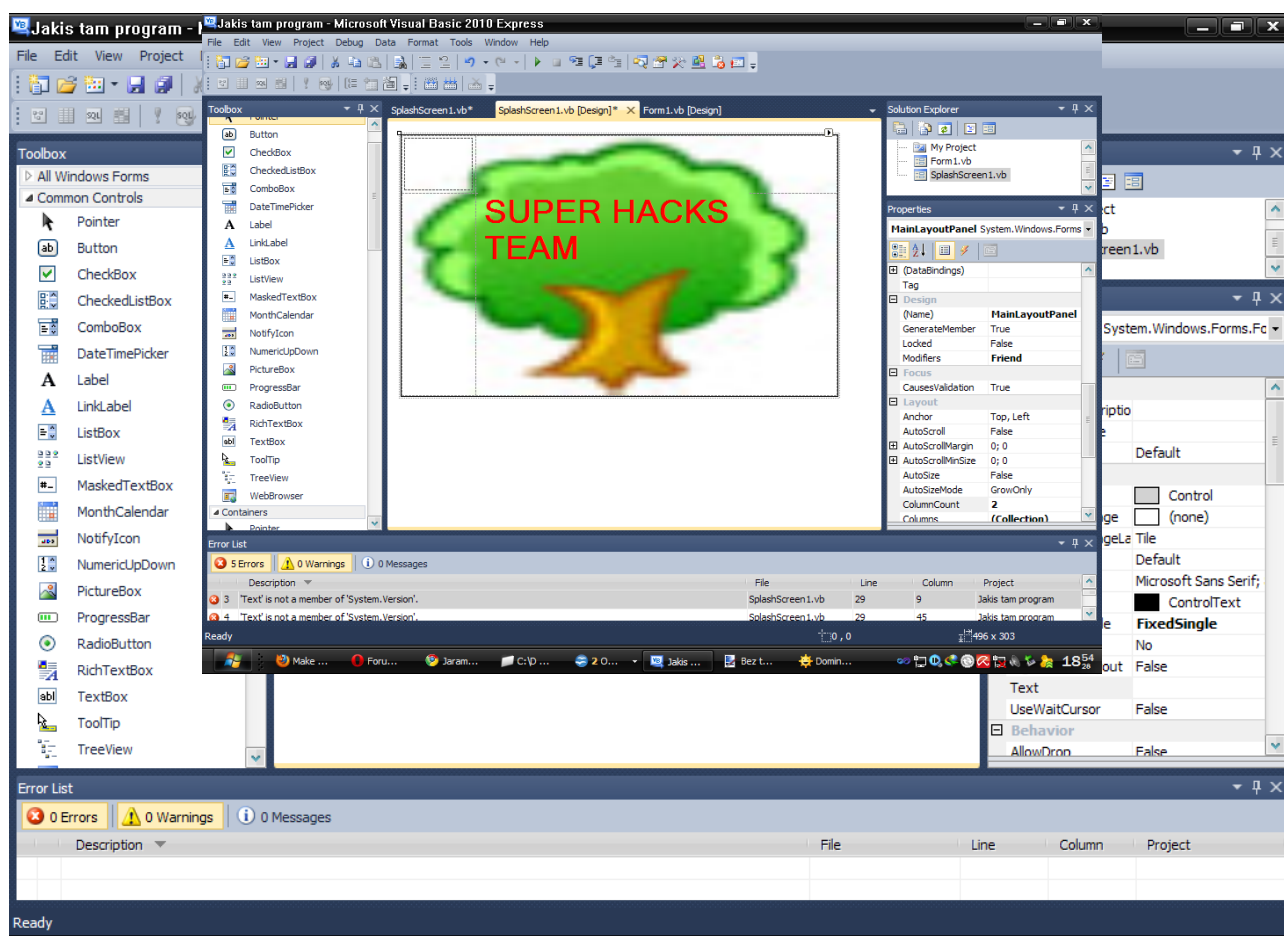
> Z całej listy wybieramy „Splash Screen”





5.1 – Upiększamy naszą aplikację

Naszym oczom ukazało się całkiem dziwne okienko. Jest ono niezwykle „użyteczne” możemy tutaj dodać logo naszego fake-teamu, który tworzy cracki, hacki do gier. Jakieś firmowe logo, że nasz program jest oryginalny (z tym bym uważał :) lub coś w stylu (WELCOME IN WORLD OF HACKERS). Pomysłów jest naprawdę wiele. Edytujemy go przez dodawanie pictureboxów, zmienianie tła, dodawanie tekstu itp. itd. U mnie po krótkiej chwili wyglądało to tak.

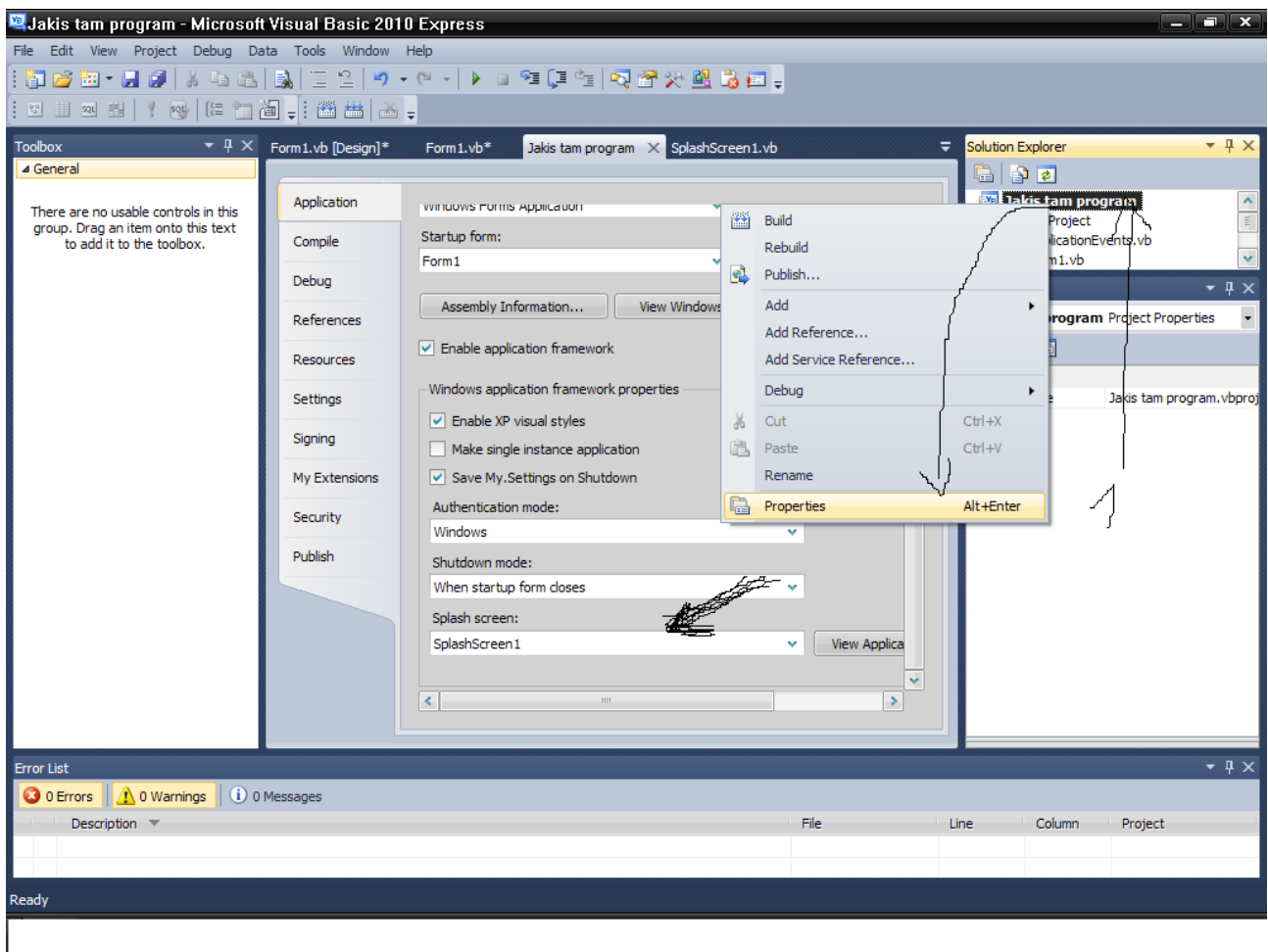


Wiem, może i nie jest to dzieło artystyczne, ale nie chciałem pokazywać wam mojego „gotowego” splash screena. Moglibyście szybko wyłączyć moje programy na yt. Widziałem już sporo ciekawych wykonań tego efektu. Pamiętajcie jednak żeby nie było tego za dużo. (W sensie nawalonego tekstu czy obrazków) To ma upewniać użytkownika, że pobierał on hack, który stworzony został przez naprawdę profesjonalną ekipę.



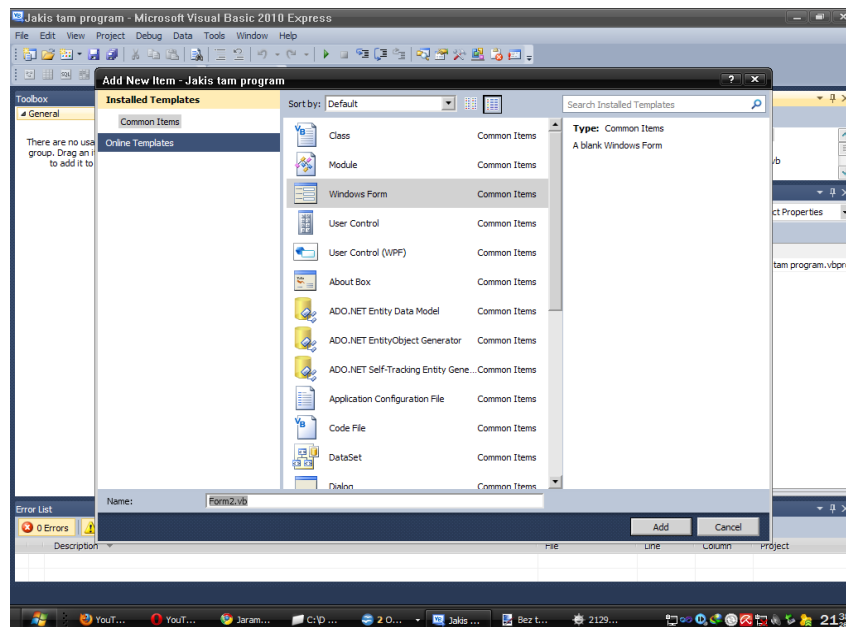
5.1.1 – Upiększamy naszą aplikację

Postępujemy zgodnie ze screenem, czyli: Klikamy PPM na nasz projekt, następnie wchodzimy w „**Properties**” i na samym dole okienka szukamy „**Splash Screen**” wybieramy z listy tego, którego przed chwilą stworzyliśmy. Teraz możemy zobaczyć efekt. Wciskamy „**F5**” i widzimy jak nasz „**Splash Screen**” się pojawia, a później dopiero widać nasz program. Warto używać tej opcji :)

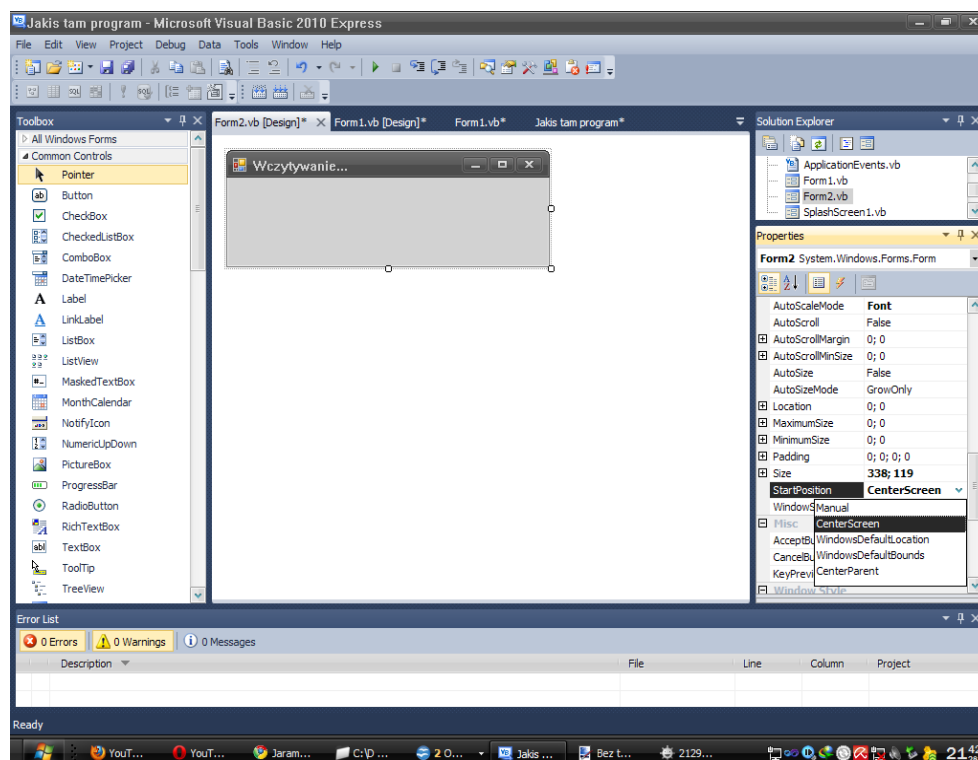


5.2 – Upiększamy naszą aplikację

Następnie zajmiemy się bardzo ciekawym efektem, a mianowicie „Zaawansowanym Loadingiem”. Aby to zrobić postępujemy tak samo jak poprzednio, tylko nie dodajemy teraz „nowego itemu” tylko „Windows Form” później wybieramy „Windows Form” tak jak to jest na screenie poniżej:

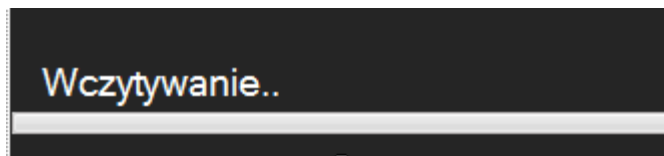


Następnie robimy takie okienko, jak zrobiłem ja. Ustawiamy nazwę na wczytywanie (Loading..) i StartPosition ustawiamy na „Center Screen” - czyli środek ekranu. Od razu wypełniamy jakimś tłem, obrazkiem lub czymkolwiek. Ja użyję koloru czarnego. Robimy też napis Loading.. albo Wczytywanie.. (Ustawiamy Maximize oraz Minimize box na wartość False!). Natomiast **FormBorderStyle** ustawiamy na „none”





Powinno to wyglądać mniej więcej tak:



Przejdźmy do kodowania, klikamy 2x w okienko. Nie żaden progressbar, czy napis. Tylko w okienko. Dodajemy tam kod
Timer1.start ()

Kolejną rzeczą jaką powinniśmy zrobić, to dodać Timer. Teraz uwaga! Timer dodajemy na nasze okno numer2, czyli Form2.vb. Dodajemy go gdziekolwiek (najlepiej na jakieś nie użyte miejsce, żeby nie wrzucić go do tekstu). Klikamy 2x na timer, i dodajemy kod:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Increment(1)
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Me.Hide()
        Form1.Show()
        Timer1.Stop()
    End If
```

Jeżeli oczywiście weźmiemy pod uwagę to, że dodaliśmy wszystko, a nasze okno „Wczytywania” to form2, a nie form1 :). Jeśli ktoś będzie miał z tym kłopot, to odsyłam na gg. Kod prosty, kiedy skończy się ładowanie, ukryj te okno, i pokaż okno Form1.

Następnie w naszym „Form1” czyli programie. Do przycisku dodajemy

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Form2.Show()
End Sub
```

Czyli że jak ktoś kliknie, to pojawi się Form2, czyli nasze wczytywanie, które powinno działać.

Ode mnie na razie tyle. Ci, którzy kupili poradnik dostaną następnie rozszerzenie za darmo. Poprawię te małe błędy i postaram się wszystko dokładniej omówić. Niestety z braku czasu, nie mogę wam napisać więcej elementów, jakie możecie wykorzystać Ale zrobię to jak najszybciej będę mógł.

Live support: 10520903 jeżeli będą jakieś pytania.
Pozdrawiam i dziękuję za zakup poradnika

