

Jacek Matulewski  
<http://www.phys.uni.torun.pl/~jacek/>

# Tworzenie aplikacji Windows

## Kontrolki ActiveX

# Ćwiczenia

Toruń, 1 grudnia 2002

Najnowsza wersja tego dokumentu znajduje się pod adresem  
[http://www.phys.uni.torun.pl/~jacek/dydaktyka/rad/rad4\\_ocx.pdf](http://www.phys.uni.torun.pl/~jacek/dydaktyka/rad/rad4_ocx.pdf)

Źródła opisanych w tym dokumencie programów znajdują się pod adresem  
[http://www.phys.uni.torun.pl/~jacek/dydaktyka/rad/rad4\\_ocx.zip](http://www.phys.uni.torun.pl/~jacek/dydaktyka/rad/rad4_ocx.zip)

# I. Spis treści

I. Spis treści.....	2
II. Kontrolka StoperX .....	3
1. Komponent TStoper (RAD1.VII.5) .....	3
2. Kontrolka StoperX .....	3
3. Import kontrolki do środowiska Delphi/C++ Builder i jej testowanie .....	5
4. Testowanie kontrolki StoperX w Visual Basic 98 .....	6
5. Wykorzystywanie kontrolki w kodzie HTML .....	6
III. Kontrolka DiskInfoPanelX.....	8

## II. Kontrolka StoperX

Nazwa ActiveX wprowadzona została przez Microsoft w miejsce skrótów OCX znanego wszystkim programistom związanym z Visual Basic lub Visual C++. Oznacza ona technologię systemu Windows, która wykorzystując standard COM (*Component Object Model*) dostarcza aplikacjom klasy gotowe do wykorzystania przez programistów. Każda zarejestrowana klasa ActiveX posiada swój unikalny w systemie identyfikator (GUID – *Globally Unique Identifier*) o nazwie CLSID (*Class Identifier*), który aplikacja-klient musi podać systemowi, aby wskazać na interesującą ją klasę-serwer ActiveX. Wielką zaletą tej technologii jest jej uniwersalność – zarejestrowana w systemie kontrolka może być wykorzystywana w niemal każdej platformie programistycznej (nawet w kodzie HTML). Jej wadą jest jednak angażowanie systemu do jej kontroli – nie jest integralną częścią skompilowanej aplikacji i musi być dostarczona niezależnie.

### 1. Komponent TStoper (RAD1.VII.5)

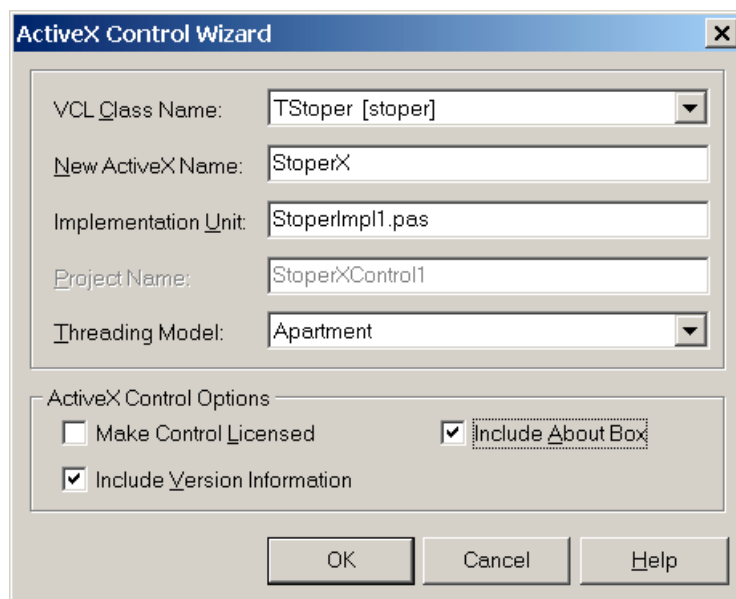
#### Uwaga!

Delphi wspiera automatyczne tworzenie kontrolki ActiveX od wersji 3. Ponadto automatyczne wspomaganie ich tworzenia nie jest dostępne w wersjach Standard.

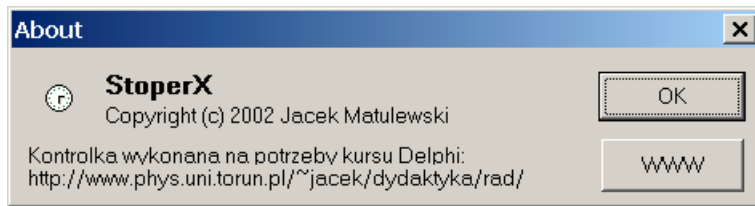
Wykorzystamy komponent `TStoper`, który jest opisany w pierwszej części skryptu [RAD1](#), rozdział VII, paragraf 5. Należy go zainstalować w środowisku Delphi lub C++ Builder.

### 2. Kontrolka StoperX

Chcąc automatycznie stworzyć kontrolkę należy w menu File, New, Other wybrać zakładkę ActiveX, a następnie pozycję „ActiveX Control”. W okienku dialogowym ActiveX Control Wizard wybieramy klasę bazową kontrolki spośród zainstalowanych komponentów VCL, w naszym przypadku jest to `TStoper`. Nazwa komponentu i pliku zostanie stworzona automatycznie na podstawie nazwy klasy VCL, ale można ją dowolnie modyfikować. Do kontrolki dodajemy informację o wersji oraz formę „about” zaznaczając odpowiednie checkboxy.



Formę (domyślnie `About1`) zawierającą informację o kontrolce możemy zaprojektować dowolnie (przykład na rysunku poniżej).



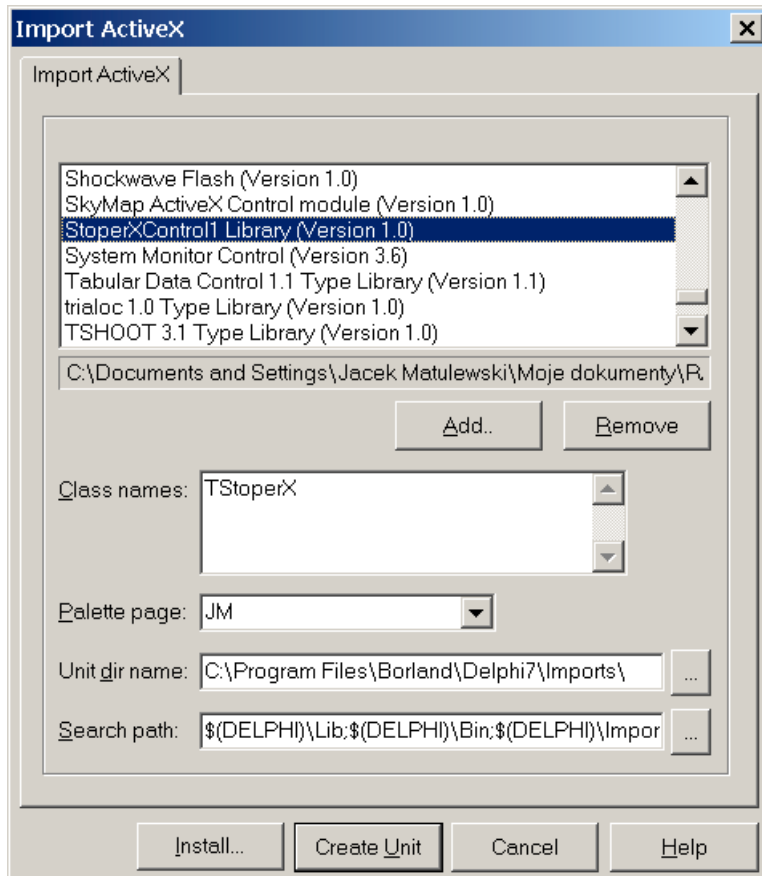
Projektując komponent TStoper wykorzystałem zdarzenie OnClick. Niestety nie można tego zdarzenia uprzywilejować, a nie ma komponentu typu TCustomButton, który miałby to zdarzenie zadeklarowane jedynie jako publiczne. Podczas tworzenia kontrolki ActiveX zdarzenia komponentu są przekazywane do mechanizmu OCX, co umożliwia ich kontrolę (zob. pliku StoperImpl1.pas). Jednak przypisania, które to robią (zob. kod poniżej) przesłaniają przypisanie do tego zdarzenia, metody którą zdefiniowaliśmy „we wnętrzu” komponentu i konieczne jest jego usunięcie. Inaczej stoper nie reagowałby prawidłowo na kliknięcie.

```
procedure TStoperX.InitializeControl;  
begin  
  FDelphiControl := Control as TStoper;  
  //FDelphiControl.OnClick := ClickEvent;  
  //FDelphiControl.OnKeyPress := KeyPressEvent;  
  FDelphiControl.OnReset := ResetEvent;  
  FDelphiControl.OnStart := StartEvent;  
  FDelphiControl.OnStop := StopEvent;  
end;
```

Pozostaje skompilować projekt kontrolki (Ctrl+F9). Powstanie plik OCX, który trzeba zarejestrować w systemie. Ta ostatnia czynność również jest wspierana przez środowisko Delphi/C++ Builder. Wystarczy w menu Run wybrać pozycję Register ActiveX Server.

### 3. Import kontrolki do środowiska Delphi/C++ Builder i jej testowanie

Aby zaimportować stworzoną przez nas kontrolkę do środowiska Delphi lub C++ Builder (od wersji 3) należy w menu Components wybrać pozycję Import ActiveX Control. Następnie w okienku dialogowym należy wybrać StoperXControl1 Library (Version 1.0).



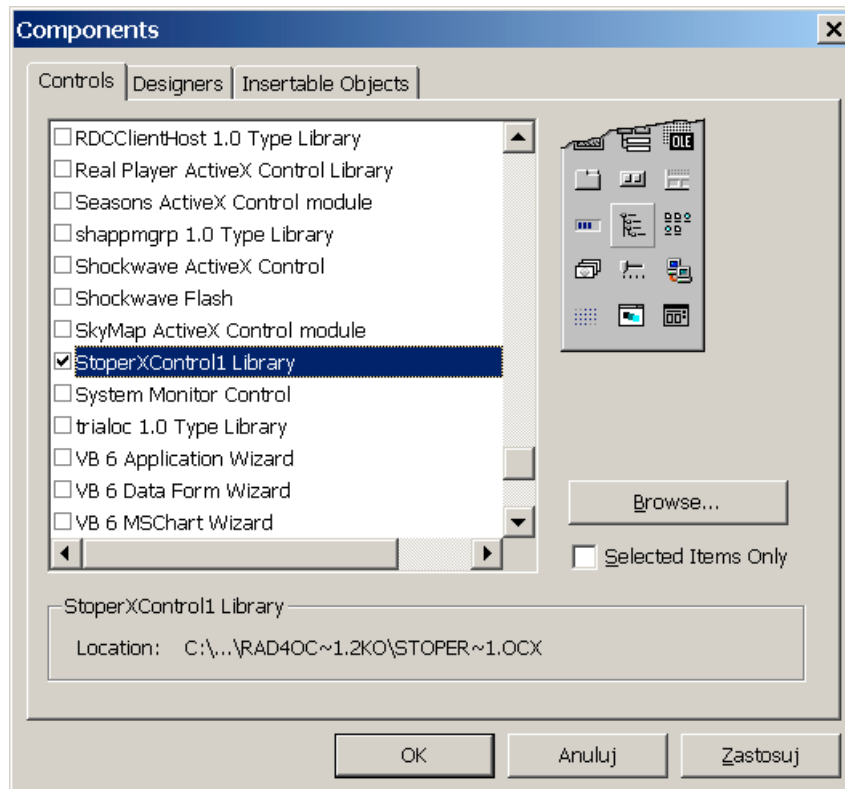
Kontrolka domyślnie pojawi się na paletce „ActiveX”, ale można to podczas importu lub potem dowolnie zmodyfikować.

I to wszystko. Kontrolka w trakcie projektowania zachowuje się jak zwykły komponent VCL, ale należy pamiętać, że nie będzie wkompiłowana w końcowy kod programu. Można ją identycznie jak komponenty umieszczać na formie, modyfikować własności i tworzyć metody zdarzeniowe.

## 4. Testowanie kontrolki StoperX w Visual Basic 98

Stworzenie kontrolki ActiveX może być, obok bibliotek DLL, sposobem przenoszenia klas między różnymi środowiskami programistycznymi. Jest tak dlatego, że format obu narzucony jest zewnętrznie przez Microsoft i system Windows.

Sprawdźmy np. działanie kontrolki w Visual Basicu: Spośród szablonów projektów wybieramy „Standard EXE”. Zarejestrowany w systemie komponent StoperX niczym nie różni się od pozostałych kontrolki Visual Basicu i powinien być obecny na liście dostępnych komponentów. W menu Project, Components ... (Ctrl+T) znajdujemy pozycję „StoperXControl1 Library” i zaznaczmy ją.



Na palecie dostępnych w projekcie kontrolki pojawi się ikona z opisem StoperX. Podobnie jak w Delphi możemy umieścić obiekt na formie i w okienku Properties zmieniać jego własności i definiować metody zdarzeniowe.

## 5. Wykorzystywanie kontrolki w kodzie HTML

Z kontrolką ActiveX można zrobić jeszcze więcej. Jeżeli znamy numer identyfikujący klasę kontrolki, a można go znaleźć w pliku StoperXControl1\_TLB.pas:

```
// *****//  
// GUIDS declared in the TypeLibrary. Following prefixes are used:  
//   Type Libraries      : LIBID_xxxx  
//   CoClasses          : CLASS_xxxx  
//   DISPInterfaces     : DIID_xxxx  
//   Non-DISP interfaces: IID_xxxx  
// *****//  
const  
  // TypeLibrary Major and minor versions  
  StoperXControl1MajorVersion = 1;  
  StoperXControl1MinorVersion = 0;
```

```
LIBID_StoperXControl1: TGUID = '{D56450D1-47B9-4473-9C67-A53C1AD19DFE}';  
  
IID_IStoperX: TGUID = '{47BBA39C-C208-4E68-91C7-4975E0609D74}';  
DIID_IStoperXEvents: TGUID = '{F6318A38-1324-48F1-83AC-B08B9771A23D}';  
CLASS_StoperX: TGUID = '{BE02B886-11A4-42C6-95CB-146CE904F38D}';
```

możemy wykorzystać kontrolkę także w kodzie HTML. Warunkiem jest jej zarejestrowanie na lokalnym komputerze. Umieszczenie kontrolki jest niezwykle proste i co najważniejsze zachowuje jej pełną funkcjonalność. Nic nie stoi na przeszkodzie, żeby w oparciu o tą technologię tworzyć applety. Jedynym, ale bardzo uciążliwym warunkiem jest konieczność rozprowadzania pliku OCX i jego rejestracja na komputerze kliencie.

```
<HTML>  
<BODY>  
<H2>Test kontrolki ActiveX - StoperX</H2>  
<OBJECT NAME = "StoperX"  
    WIDTH=200  
    HEIGHT=30  
    CLASSID = "CLSID:BE02B886-11A4-42C6-95CB-146CE904F38D">  
Błąd! Obiekt StoperX nie może być utworzony.  
</OBJECT>  
</BODY>  
</HTML>
```

### III. Kontrolka DiskInfoPanelX

Drugą kontrolkę ActiveX stworzymy na podstawie komponentu `TDiskInfoPanel` opisywanego w części o [WinAPI](#), część IV, paragraf 3. Komponent ten powinien być zainstalowany w Delphi/C++ Builderze, więc można od razu zająć się tworzeniem kontrolki zgodnie z procedurą opisaną w poprzednim paragrafie.

Tym razem komponent posiada publiczną własność `DriveLetter` pojawiającą się w Object Inspectorze. Podczas tworzenia kontrolki własność ta, oryginalnie typu `char` (litera), zostanie zamieniona we własność typu naturalnego – kod ASCII litery. Zatem zamiast litery 'C' należy podać jej kod – 67, zamiast 'D' – 68 itd.

Własność ta pojawi się w Visual Basicu po zainstalowaniu kontrolki. Można również zmieniać jej wartość z poziomu kodu HTML. W tym celu należy skorzystać ze znacznika `PARAM`.

W poniższym kodzie kontrolka `DiskInfoPanelX` wykorzystana jest dwukrotnie. Pierwszy raz z wartością domyślną, tj 67 ('C'), za drugim razem parametr `DriveLetter` ustalany jest na 68 (czyli 'D').

```
<HTML>
<BODY>
<H2>Test kontrolki ActiveX - DiskInfoPanelX</H2>

<OBJECT NAME="DiskInfoPanel"
  STANDBY="Proszę czekać! Obiekt DiskInfoPanel jest ładowany ..."
  WIDTH=800
  HEIGHT=50
  VSPACE=5
  CLASSID = "CLSID:11DCD049-D49E-4397-BD07-27FD825200F1">
Błąd! Obiekt DiskInfoPanelX nie może być stworzony
</OBJECT>

<BR>

<OBJECT NAME="DiskInfoPanelD"
  STANDBY="Proszę czekać! Obiekt DiskInfoPanel jest ładowany ..."
  WIDTH=800
  HEIGHT=50
  VSPACE=5
  CLASSID = "CLSID:11DCD049-D49E-4397-BD07-27FD825200F1">
<PARAM NAME="DriveLetter" VALUE=68 VALUETYPE="data">
Błąd! Obiekt DiskInfoPanelX nie może być stworzony
</OBJECT>
</BODY>
</HTML>
```