Jak napisaś edytor tekstowy w Delphi
(wersja Delphi 2009)

I. Start nowej aplikacji

Przed stworzeniem nowej aplikacji, stwórzmy katalog w którym będziemy przechowywać pliki źródłowe Projektu:

1. Tworzymy katalog Edytor albo w domyślnym folderze zainstalowanego Delphi albo tworzymy własny folder, np. C:\ Edytor Tekstu
2. Zaczynamy nowy projekt przez wybranie ***File | New | VCL Forms Application***

   Każda aplikacja jest przedstawiana przez Projekt. Kiedy uruchamiamy Delphi, tworzy on pusty, domyślny Projekt, i tworzy automatycznie poniższe pliki:

   - ***Project1.dpr*** : plik kodu źródłowego związanego z projektem.Nazywany jest plikiem projektu
   - ***Unit1.pas*** : plik kodu źródłowego powiązany z główną formatką projektu. Nazywany jest plikiem modułu.
   - ***Unit1.dfm*** : plik zasobów, który przechowuje informacje o głównej formatce projektu. Nazywany jest plikiem formatki.

   Każda formatka ma swoje własne pliki modułu (***Unit1.pas***) i formatki (***Unit1.dfm***). Jeśli tworzymy druga formatkę, automatycznie jest tworzony drugi plik modułu (***Unit2.pas***) i formatki (***Unit2.dfm***)

3. Wybieramy ***File | Save All*** aby zapisać nasze pliki na dysku. Kiedy pojawi się okienko dialogowe Save As:

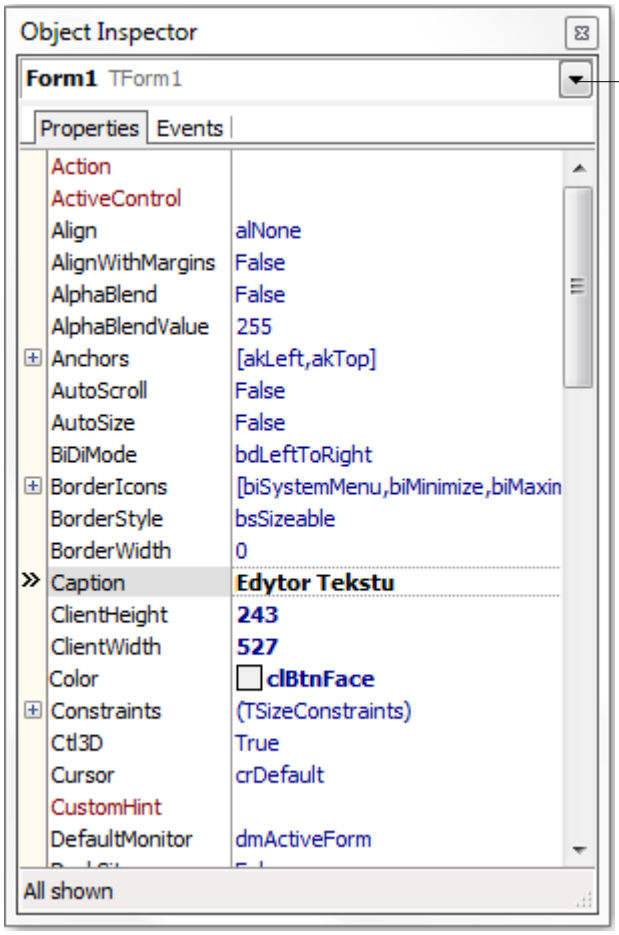
   - Przechodzimydo folderu ***C:\ Edytor Tekstu)*** (w naszym przypadku)
   - Zapisujemy pierwszy Unit z domyślną nazwą ***Unit1.pas***
   - Zapisujemy projekt używając nazwy Edytor.dpr (Plik wykonywalny będzie miał taką samą nazwę jak nazwa projektu, ale z rozszerzeniem .exe)

   Później można zpaisywać swoją pracę przez wybranie ***File | Save All***.
   Kiedy zapiszemy projekt, Delphi stworzy kilka dodatkowych plików w katalogu projektu. Nie usuwamy tych plików.

II. Ustawianie wartości właściwości

Kiedy otwieramy nowy projekt, Delphi wyświetla główną formatkę Projektu, nazwaną domyślnie ***Form1***. Tworzymy interfejs użytkownika i inne części aplikacji poprzez umieszczanie komponentów na tej formatce. Obok formatki, widać ***Object Inspector,*** którego możemy używać do ustawiania wartości właściwości dla formatki i komponentów na niej umieszonych. Kiedy ustawiamy te właściwości, Delphi samo zarządza kodem źródłowym.Wartości ustawiane w ***Object Inpector*** są nazywane ustawieniami czasu projektowania.

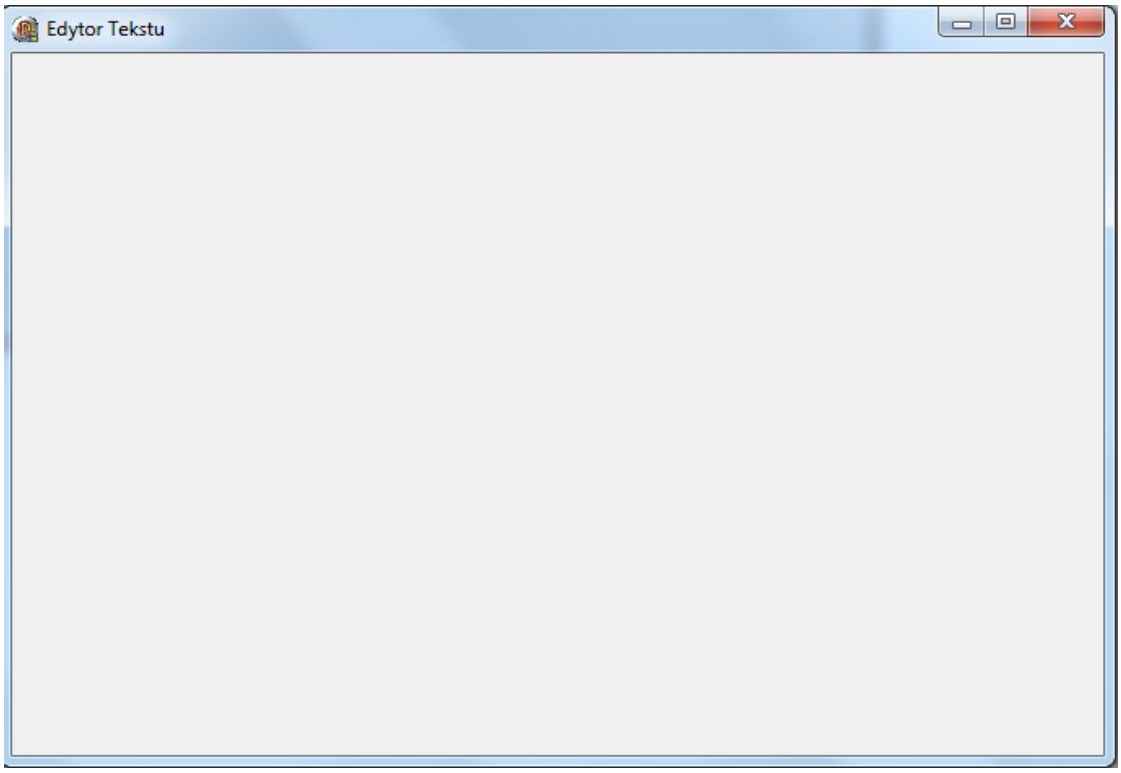
1. Znajdźmy właściwość ***Caption*** formatki w ***Object Inspector*** i wpiszmy Edytor Tekstu, zastępując wartość domyślną. Zwróć uwagę ,że zmienia się nagłówek formatki.



Lista rozwijana na górze Object Inspector pokazuje aktualnie wybrany koponent. W tym przypadku jest to komponent Form1 a jego typ to TForm1

Kiedy komponent jest wybrany, Object Inspector wyświetla jego właściwości

2. Uruchomimy przycisk ***F9*** , chcociaż nie ma na formatce jeszcze żadnych elementów



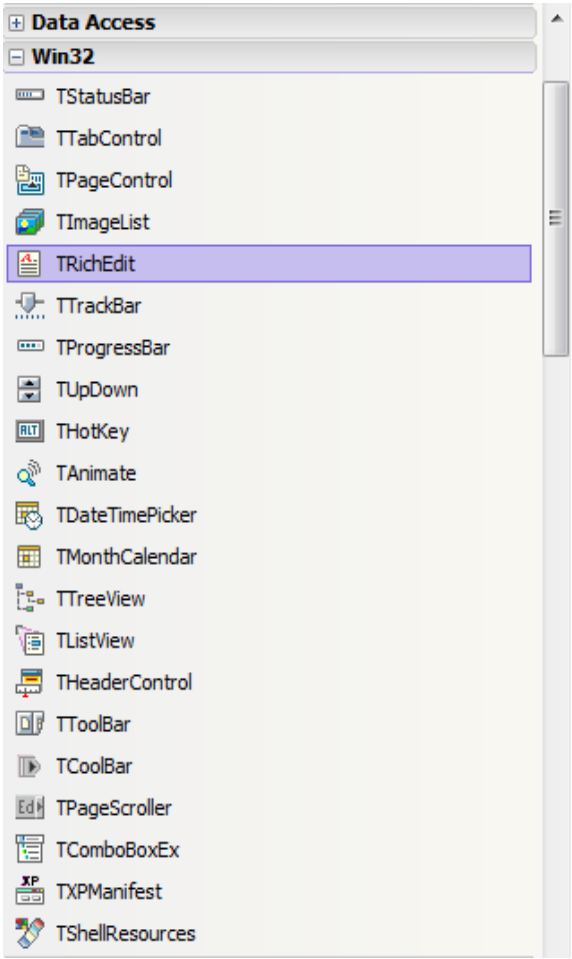
3. Wrócimy teraz do obszaru projektowania ***Form1***, wykonując jedną z poniższych rzeczy:

- klikamy **X** w prawym górnym rogu paska tytułowego aplikacji
- klkamy przycisk ***Exit*** apliakcji w lewym górnym rogu paka tytułowego i klikamy ***Close***
- wybieramy ***View*** | ***Forms*** , zaznaczamy ***Form1*** i klikamy OK
- wybieramy ***Run*** | ***Program Reset***

III. Dodawanie komponentów na formatkę

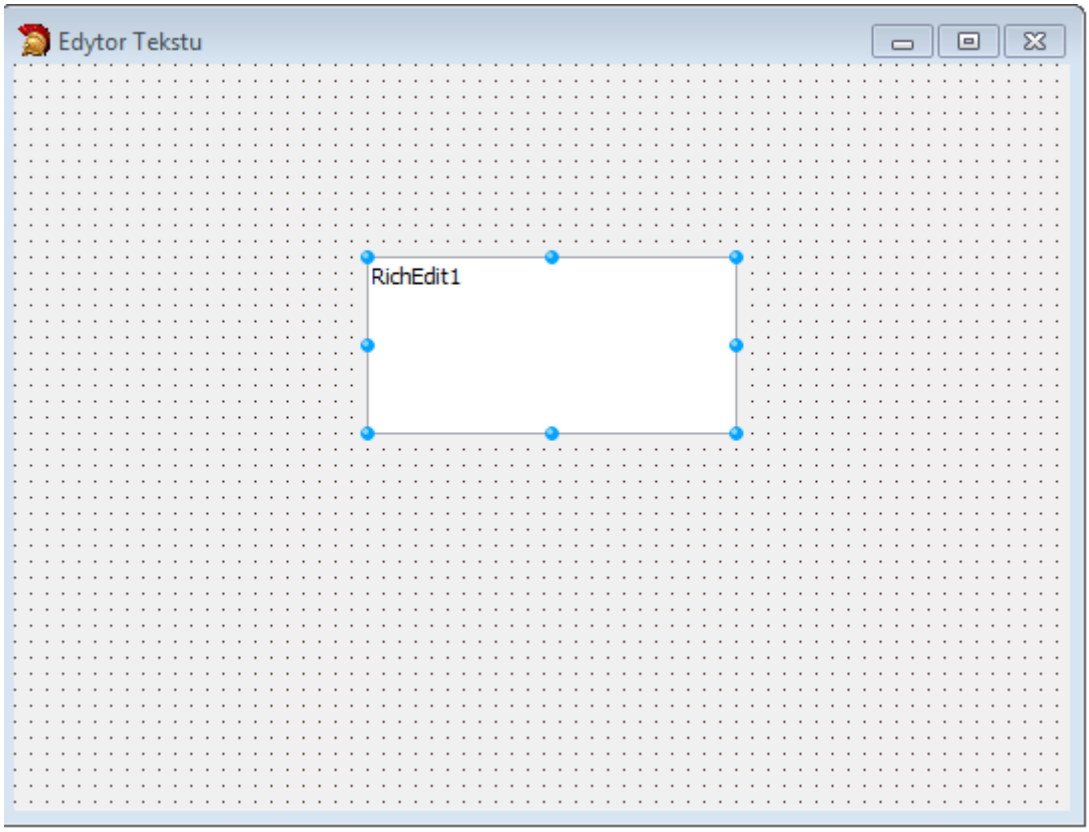
Zanim zaczniemy dodawać komponenty na formatkę, musiy najpierw pomyśleć o najlepszym spoosbie na stworzenie interfejsu użuytkownik (UI) dla naszej aplikacji. UI jest tym co powala użytkownikowi na współpracę z programem i powinno być zaprojektowane dla łatwego użytkowania. Delphi posiada wiele komponentów, które można umieścić w naszej aplikacji. Na przykład, są komponenty ,które (pochodne ***obiektów***) na palecie ***Component*** , które pozwalają na stworzenie menu, paska narzędziowego, okienek dialogowych i wielu innych wizualnych i niewizualnych elementów programu. Apliakcja edytora tekstu wymaga obszaru edycynego, paska satnu dla wyświetlania informacji takich jak nazwa edytowanego pliku, menu i paska narzędziowego z przyciskami dla łatwiejszego dostępu do poleceń. Na początek dodamy obszar edycyjny i pasek stanu na formatkę.

1. Na palecie ***Component*** znajdźmy komponent ***RichEdit***



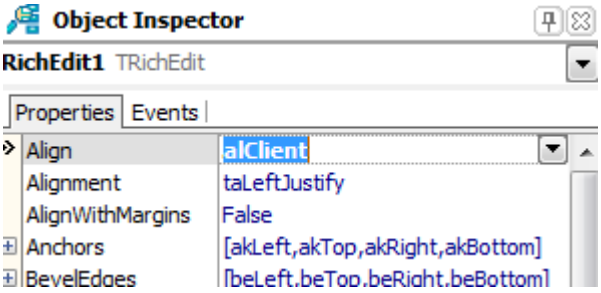
Kiedy już został znaleziony, możemy

- Zaznaczyć komponent na palecie a potem kliknąć na formatce w miejscu gdzie ma zostać umieszczony
- Kliknąć dwukrotnie komponent aby został umieszczony na formatce



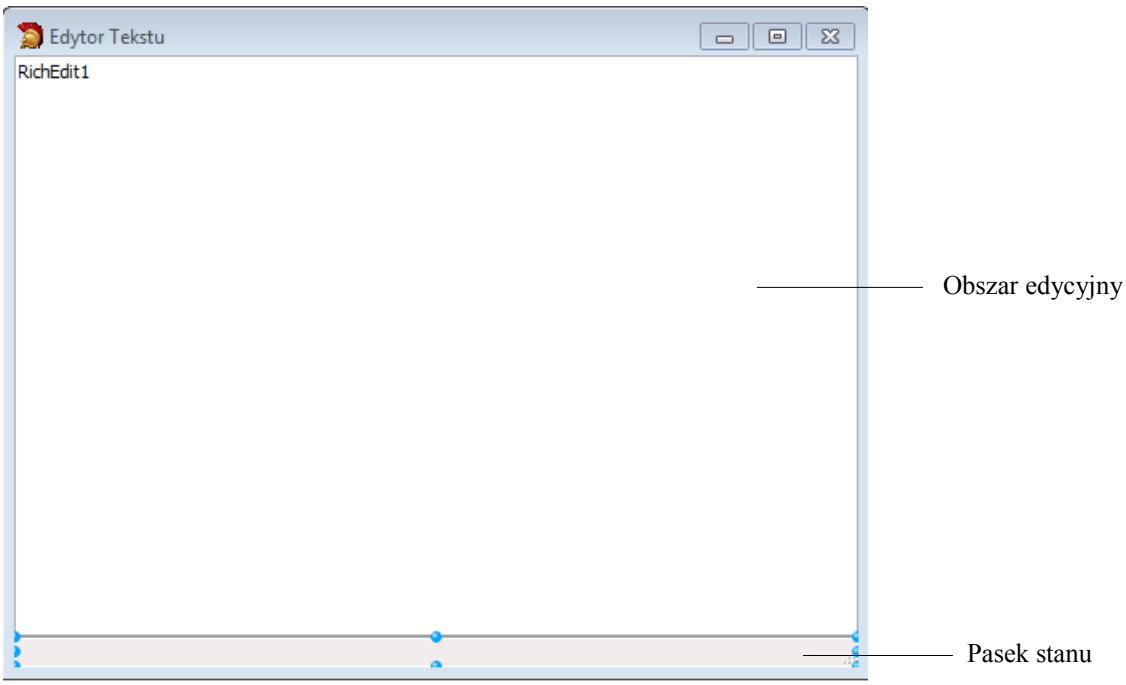
Każdy komponent Delphi jest ***klasą***; umieszczenie komponentu na formatce tworzy ***instancję*** tej klasy. Kiedy komponent znajduje się na formatrce, Delphi generuje kod potrzebny do zbudowaniu instancji obiektu, kiedy aplikacja zostaje uruchamiana.

2. Po zaznaczeniu komponentu ***RichEdit*** ,w Object Inspector, klikamy strzałkę przy właściwości ***Align*** i ustawiamy ją na ***alClient***

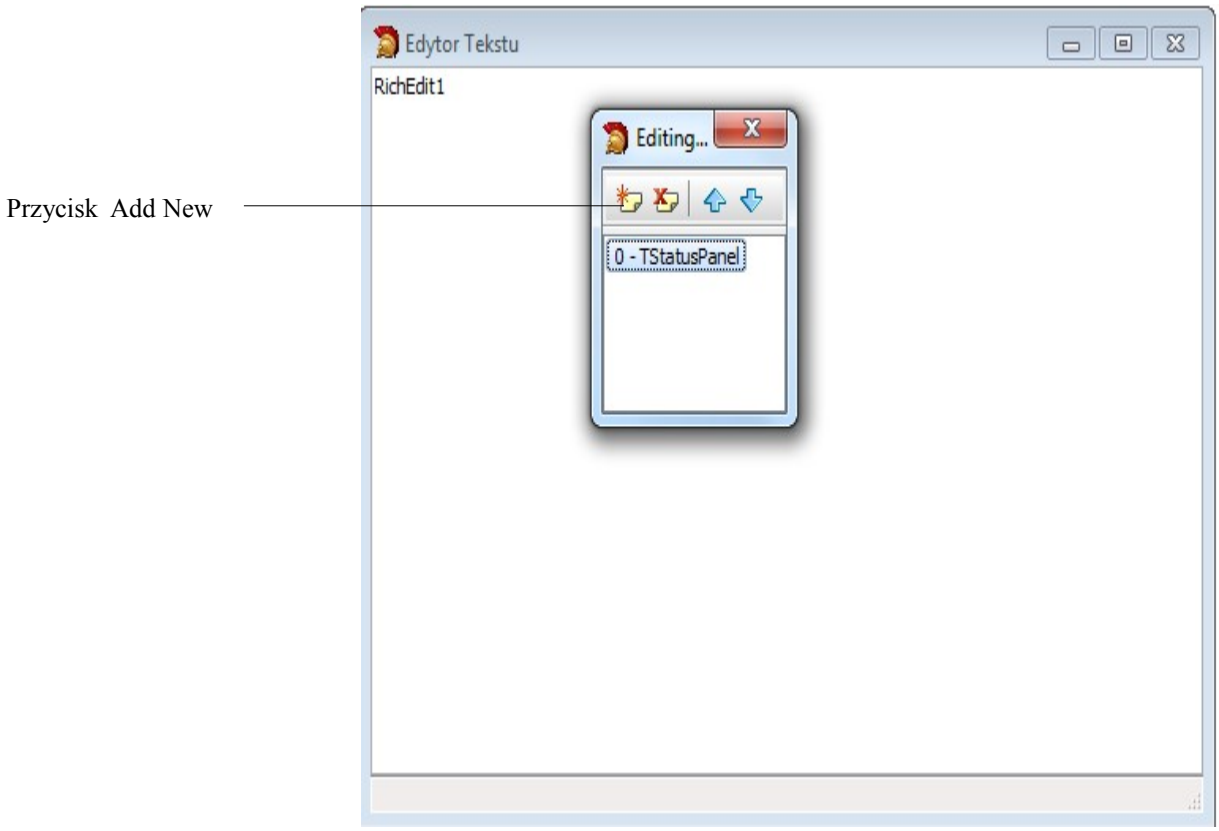


Teraz komponent ***RichEdit*** wypełnia całą formatkę, więc mamy duży obszar edycyjny. Po ustawieniu właściwości ***Align*** na ***alClient*** , komponent ***RichEdit*** będzie wypełniał całą formatkę nawet po zmianie jej rozmiaru

3. Na stronie Win32 palety komponentów, kliknijmy komponent ***StatusBar*** dodając go na dół formatki



4. Kliknij dwukrotnie na pasku stanu aby otworzyć okienko dialogowe ***Editing StatusBar1.Panels***
5. Klinij na przycisk ***Add New***, aby dodać nowy panel do paska stanu, na któym będzie wyświetlana ścieżka dostępu do edytowanego pliku



6. W Object Inspector, ustaw właściwość ***Text*** na ***beztytulu.txt.*** Kiedy użyjemy edytora tekstu, jeśli edytowany plik nie został jeszcze zapisany, nazwą pliku będzie ***beztytulu.txt*** .
7. Kliknij **X** aby zamknąć okno dialogowe Editing StatusBar1.Panels

Teraz mamy już utworzony główny obszar edycyjny interfeju użytkownika dla naszego edytora tekstu.

IV. Dodawanie menu i paska narzędziowego

Aplikacja edytora tekstu potrzebuje menu, poleceń,i paska narzędziowego. Chociaż można zakodować polecenia oddzielnie, Delphi dostarcza ***action list*** lub ***action manager*** dla scentralizowanie działań, obrazków i kodu dla menu i pasków zadań. Poniższa lista pokazuje polecenia menu jakich potrzebuje i pokazuje czy działanie ma pozwiązanie z przycikiem z paska narzędziowego

Menu	**Polecenie**	**Na pasku zadań?**	**Opis**
File	Nowy	Tak	Tworzenie nowego pliku
File	Otwórz	Tak	Otwarcie istniejącego pliku do edycji
File	Zapisz	Tak	Zapis bieżącego pliku na dysk
File	Zapisz jako	Nie	Zapis pliku przy użyciu nowej nazwy
File	Zamknij	Tak	Wyjście z edytora
Edit	Wytnij	Tak	Usunięcie tekstu i przeniesienie do Schowka
Edit	Kopiuj	Tak	Kopiowanie tekstu u przeniesienie do Schowka
Edit	Wklej	Tak	Wstawienie tekstu ze Schowka
Help	O programie	Nie	Wyświetlenie informacji o aplikacji

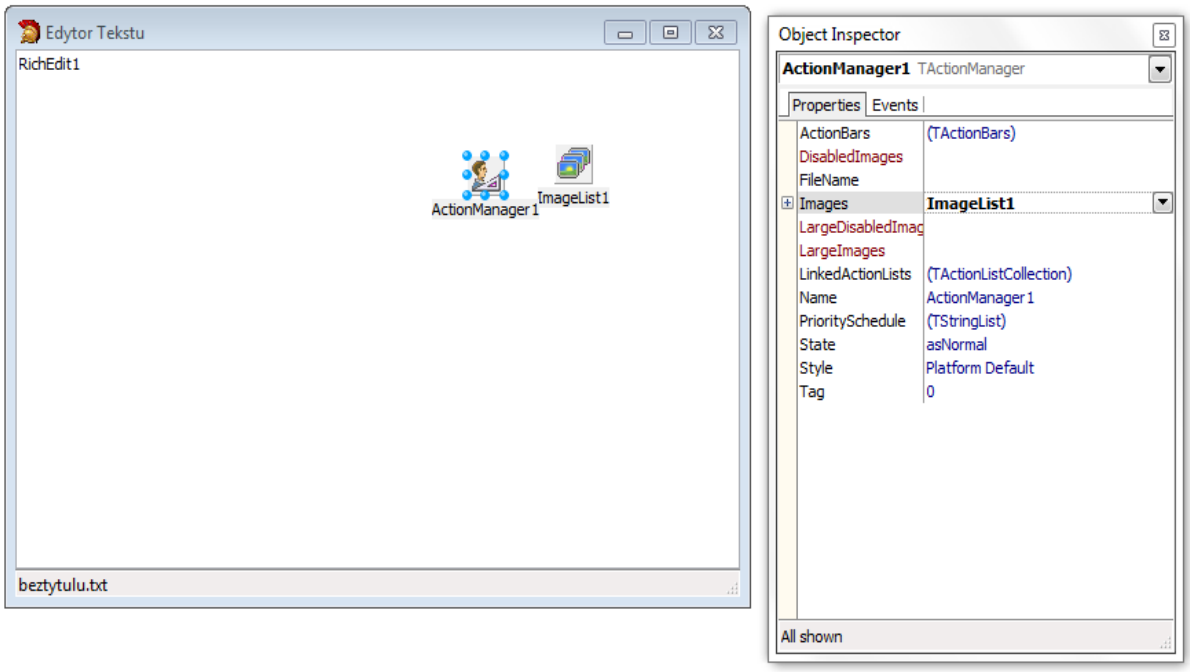
Różnice między Action Manager a Action List

Delphi dostarcza dwóch sposobów na zarządzanie działaniami dla menu i pasków zadań:

- ***Action List*** jest dostępny we wszystkich wersjach Delphi. Może być używany zarówno w aplikacjach pod Windows jak i Linux
- ***Action Manager*** jest dostępny dla wersji Professional , Enterprise i Architect. Jest łatwiejszy w stosowaniu i dostarcza większej funkcjonalności niż ***Action List****i* ,jednak może być używany tylko w aplikacjach Windows

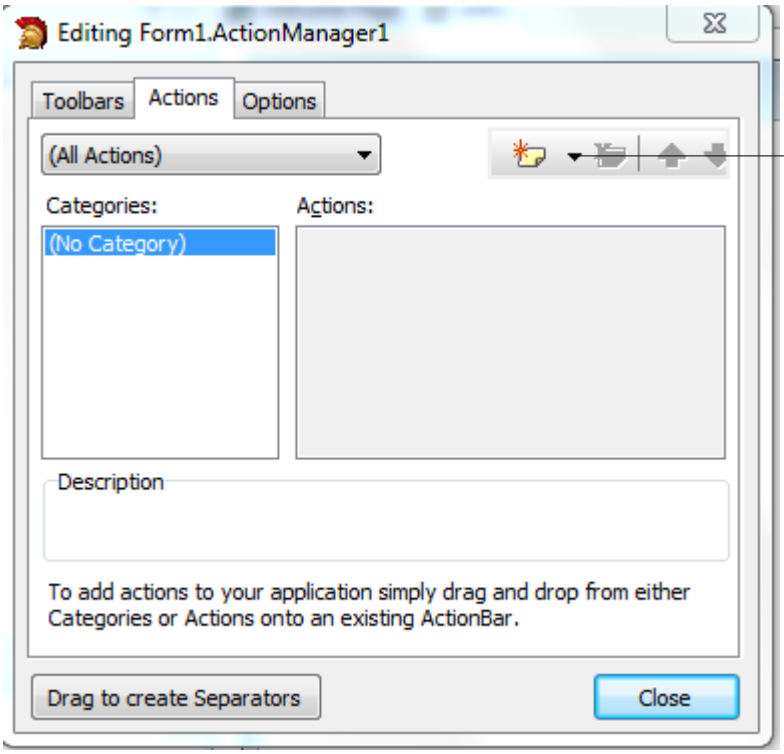
Dodawanie działań do Action Managera

1. Ze strony ***Win32*** palety ***Component,*** wybierz komponent ***ImageList*** i kliknij na formatce aby dodać go na formatkę. ***ImageList*** jest komponentem niewidocznym, więc można go umieścić gdziekolwiek na formatce
2. Na stronie ***Additional*** palety ***Component*** kliknij dwukrotnie komponent ***ActionManager*** i przeciągnij go na formatkę. On również jest komponentem niewidocznym więc również można go umieścić gdziekolwiek na formatce.
3. W zaznczonym na formatce komponencie ***ActionManager1*** ,ustaw właściwość ***Images*** w ***Object Inspector*** na ***ImageList1.***



Teraz dodamy działania do ***Action Managera*** i ustawimy ich właściwości. Dodamy zarówno działania niestandardowe, dla których ustawimy wszystkie właściwości, jak i działania stadnardowe dla których właściwości są ustawiane automatycznie.

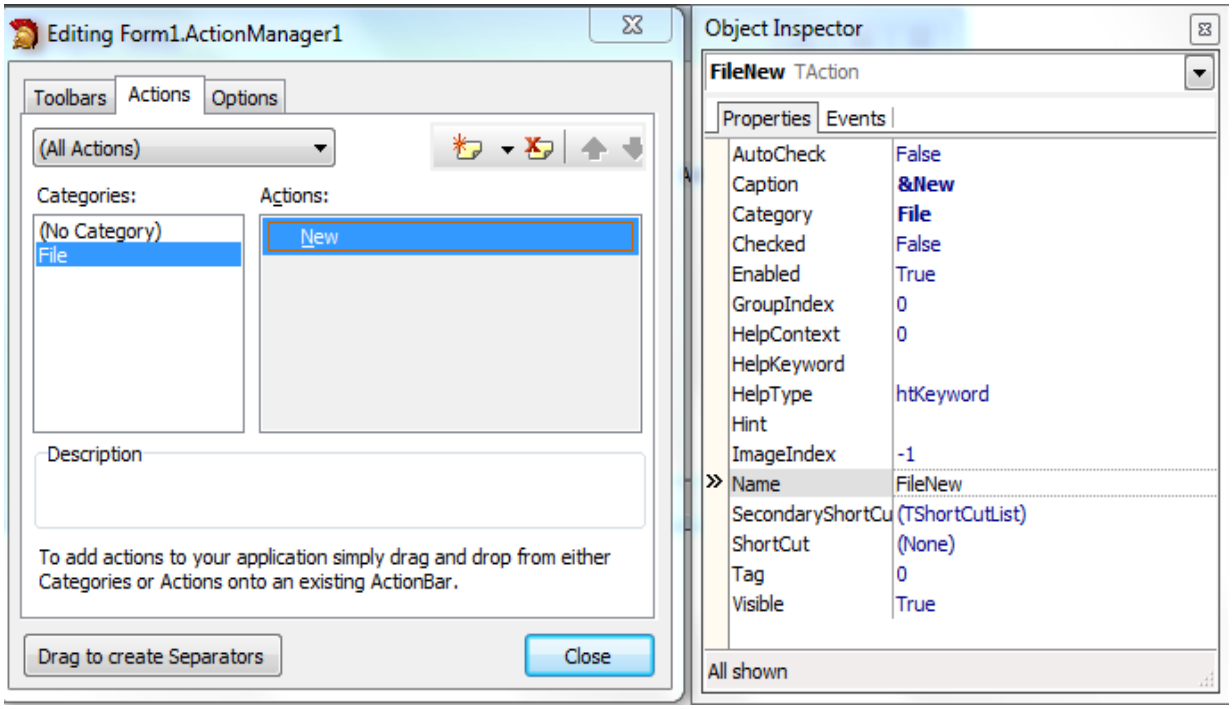
4. Dwukrotnie klikamy na komponent ***Action Manager*** aby się otworzył. Pojawi się okno dialogowe ***Editing Form1ActionManager1***.
5. Upewnijmy się ,że wybraliśmy zakładkę ***Actions*** . Klikamy w strzałkę obok przycisku ***New Action*** i klkamy ***New Action*** .



Przycisk ***New Actions***. Kiedy aktywny jest przycisk ***Delete*** można usuwać działania z listy działań

6. Upewnijmy się ,że wybrane jest ***(No Category)*** w edytorze ***Action Manager*** i ***Action1*** w ***Actions.*** W ***Object Inspector*** ustawiamy następujące właściwości:

- W ***Caption*** wpisujemy ***&New***. Znak ampersandu ("&") przed N , oznacza ,że ta litera będzie przypisana do skrótu klawiaturowego
- W ***Category*** wpisujemy ***File*** (ustawia to polecenie ***File*** na pierwszym miejscu)
- W ***Hint*** wpisujemy ***Create file*** (będzie to "dymek" Pomocy)
- W ***Name*** wpisz ***FileNew*** (dla polecenia ***File*** | ***New***) i naciśnij ***Enter*** aby zapisać zmiany



7. Upewnijmy się ,że zanazczony jest ***File*** w edytorze ***ActionManger.*** Kliknij strzałkę obok przycisku ***New Action*** i wybierz ***New Action***
8. W ***Object Inspector*** ustaw właściwości:

- W ***Caption*** wpisujemy ***&Save***
- Upewnij się ,że ***Category*** jest ustawione na ***File***
- W ***Hint*** wpisz ***Zapisz plik***
- W ***Name*** wpisz ***FileSave*** (dla polecenia ***File*** | ***Save***)

9. Kliknij strzałkę obok przycisku ***New Action*** i kliknij ***New Action***
10. W ***Object Inspector*** ustaw właściwości:

- W ***Caption*** wpisujemy ***&About***
- W ***Category*** wpisz ***Help***
- W ***Name*** wpisz ***HelpAbout*** (dla polecenia ***Help***| ***About***)

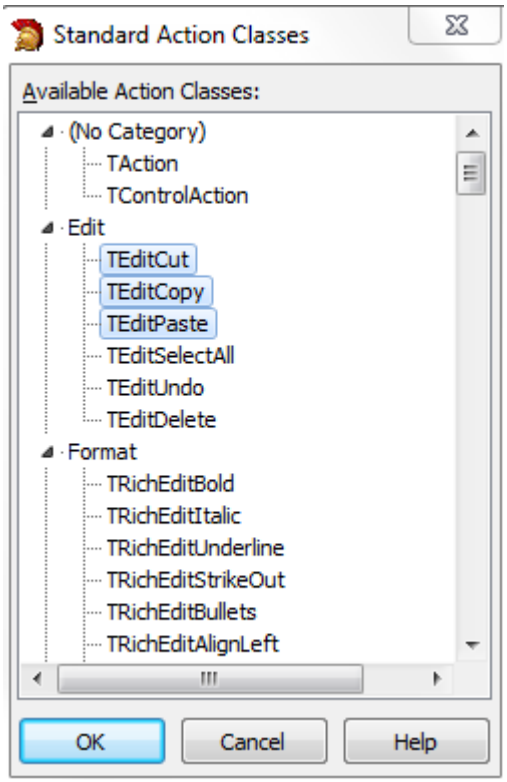
11. Kliknij strzałkę obok przycisku ***New Action*** i kliknij ***New Action***
12. Zachowaj edytor ***Action Manager*** na ekranie

13. Zapiszmy swoją pracę przez klkinięcie ***File | Save All***

Dodawanie standardowych działań

Teraz dodamy standardowe działania (open, save as, exit, cut, copy, paste)

1. Otwórz edytor ***Action Manager***
2. Kliknij strzałkę obok ***New Action*** i wybierz ***New Standard Action.*** Pojawi się okienko ***Standard Action Class***
3. Przewiń do kategorii ***Edit*** i naciśnij klawisz ***Ctrl*** aby wybrać ***TEditCut, TEditCopy i TEditPaste***. Kliknij OK aby dodać te działania do nowej kategorii Edytuj na listę ***Categories*** edytora ***Action Manager***



4. Kliknij strzałkę obok ***New Action*** i wybierz ***New Standard Action.***
5. Przewiń do kategorii ***File*** i wybierz ***TFileOpen,TFileSaveAs*** i ***TFileExit.*** Kliknij OK aby dodać te działania do kategorii Plik
6. Kliknij ***File | Save All*** aby zapisać zmiany

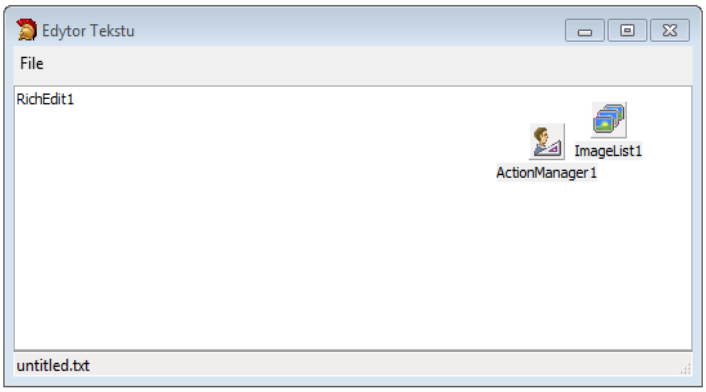
Dodawanie menu

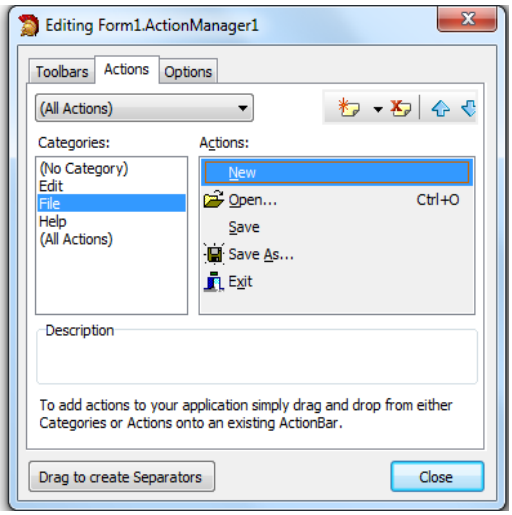
Teraz dodamy masek menu i pasek narzędziowy. Menu edytora zawiera trzy rozwijane nemu – ***File, Edit*** i ***Help*** i ich polecenia. Z edytorem ***Action Manager*** można dodać polecenia do paska menu.

1. Na stronie ***Additional*** palety ***Component*** kliknij dwukrotnie na komponencie ***ActionMainMenuBar*** i dodaj go na formatkę. Pojawi się pusty pasek menu u góry formatki
2. Otwórz edytor ***Action Manager*** i wybierz ***File*** z listy ***Categories.*** Polecenia podmenu nie są uproządkowane tak jak chcemy, ale można zmienić ich kolejność używając przycicków ***W***

górę W dół lub ***Ctrl*** + ↑ lub ***Ctrl*** + ↓

3. Wybierz ***Open*** i kliknij przycisk ***W górę*** w edytorze ***Action Manager*** tak aby polecenia były uporządkowane tak: ***New, Open, Save, Save As i Exit***
4. Przeciągnij ***File*** na pasek menu. ***File*** i polecenia podmenu pojawią się na pasku menu



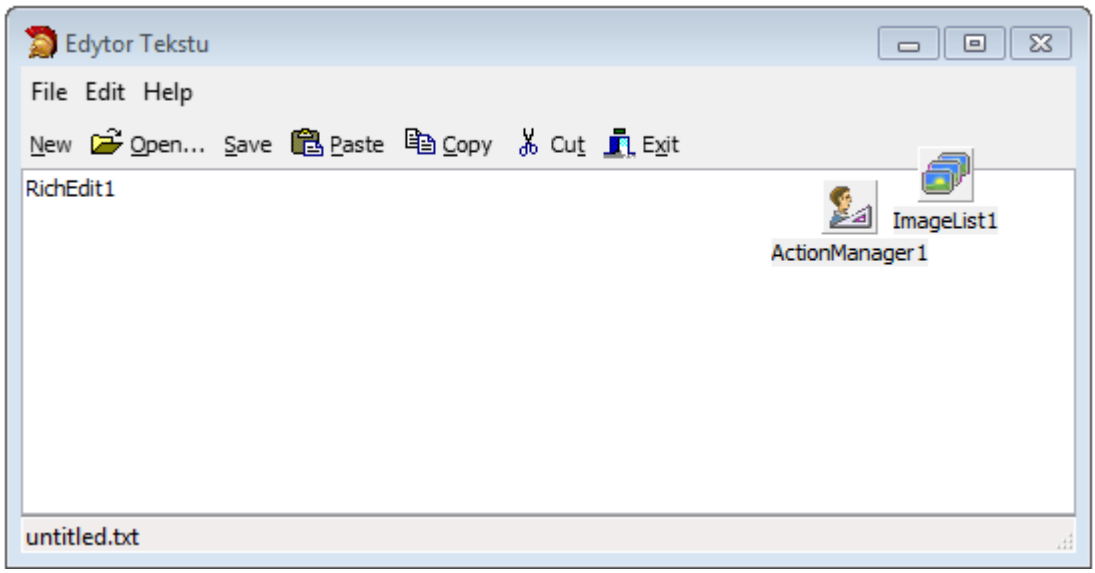


5. Z listy ***Categorties*** edytora ***Action Manager,*** przeciągnij ***Edit*** po prawej stronie ***File*** na pasku menu
6. Następnie zrób to samo z menu ***Help***
7. Kliknij w ***Help*** aby zobaczyć polecenia podmenu. Przeciągnij polecenie ***Contents*** powyżej polecenia ***Index***.
8. Naciśnij ***Esc*** lub ponownie kliknij ponownie menu ***Help*** aby go zamknąć
9. Wybierz ***File*** | ***Save All*** aby zapisać zmiany

Dodawanie paska narzędziowego

Teraza, kiedy skonfigurowaliśmy działania w edytorze ***Action Manager*** , możemy dodać kilka działań do paska narzędziowego

1. Na stronie ***Additional*** palety ***Component*** klkinij dwukrotnie na komponencie ***ActionToolBar*** aby dodać go do formatki.Pojawi się puste pasek narzędziowy pod paskiem menu
2. Otwórz edytor ***Action Manager*** wybierz ***File*** na liście ***Categories.*** Z listy ***Actions*** , wybierz ***New Open, Save*** i ***Exit*** , i przeciągnij te pozycje na pasek narzędziowy. Pojawią się na nim przyciski
3. W edytorze ***Action Manager*** przeciągnij kategorię ***Edit*** na pasek narzędziowy. Wszystkie polecenia ***Edit*** pojawią się na pasku narzędziowym



Jeśli przeciągnęliśmy złe polecenie na pasek narzędziowy, można go przeciągnąć ponownie. Lub można wybrać pozycję w ***Object TreeView*** i nacisnąć klawisz ***Del.***

4. Wybierz ***File*** | ***Save All*** aby zapisać zmiany
5. Zamknij edytor ***Action Manager***
6. Naciśnij ***F9*** aby skompilować i uruchomić projekt

Czyszczenie obszaru tekstowego

Aby wyczyścić obszar tekstowy (RichEdit):

1. Na głównej formatce kliknij komponent ***RichEdit1***.
2. W ***Object Inspector*** , obok właściwości ***Lines*** i, kliknij dwukrotnie wartość (***TStrings)*** aby wyświetlić edytor ***StringList.***
3. W tym edytorze, wybierz i usuń tekst (***RichEdit1***) i kliknij OK
4. Zapisz zmiany i uruchom ponownie program

Teraz obszar tekstowy jest pusty kiedy wyświetlana jest główna forma

Zapisywanie obsługi zdarzeń

Do tego miejsca tworzyliśmy naszą aplikację bez pisania jakiegokolwiek kodu. Przez użycie ***Object Inspector*** ustawialiśmy wartości właściwości w czasie projektowania, wykorzystując w pełni możliwości środwiska Delphi. W tej części napiszemy procedury nazywane ***obsługą zdarzeń*** ,które reagują na działania użytkownika, podczs gdy aplikacja jest uruchomiona. Połączymy obsługę zdarzeń do pozycji w menu i pasku narzędziowym, kiedy pozycja zostanie wybrana aplikacja wykona kod w programie obsługi. Dla działań niestandardowych, musimy utworzyć obsługę zdarzeń. Dla działań stadnardowych, takich jak polecenia ***File*** | ***Exit*** i ***Edit*** | ***Paste,*** kod obsługi zadrzeń jest generowany dla nas automatycznie. Jednak dla niektórych działań standardowych, takich jak polecenie ***File*** | ***Save As***,musimy napisać własną obsługę zdarzeń. Ponieważ wszystkie pozycje menu i paska narzędziowego znajdują się w edytorze ***Action Manager*** lub ***Action List*** , tam możemystworzyć obsługi zdarzeń.

Tworzenie obsługi zdarzenia dla polecenia New

1. Wybierz ***View*** | ***Units*** i wybierz ***Unit1*** aby wyświetlić kod związany z ***Form1***
2. Najpierw musimy zadeklarować nazwę pliku, który będzie używany w obsłudze zdarzenia, dodając odpowiednią właściwość dla nazwy pliku,aby stał się dostępny globalnie dla innych metod. W pliku ***Unit1.pas*** znajdujemy sekcję deklaracji publicznych dla kalsy ***TForm1*** i dodajemy linię po

   {Public declarations }

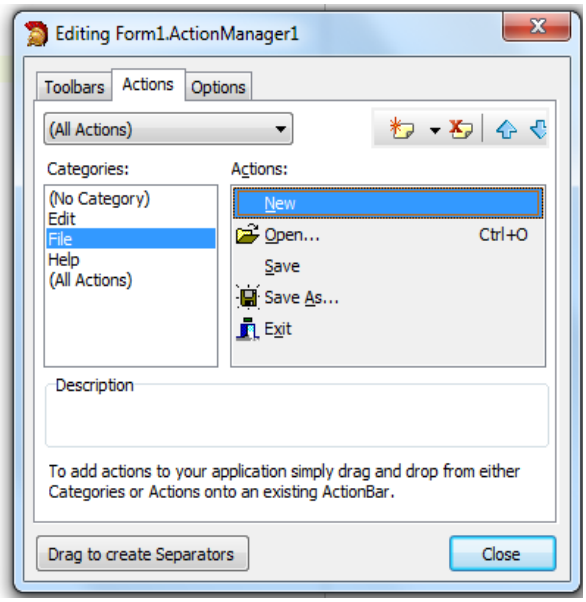
   wpisujemy

   FileName: String;

```
  private
    { Private declarations }
  public
    { Public declarations }
    FileName: String
  end;

var
  Form1: TForm1;

implementation
```

3. Klikamy ***F12*** aby wrócić do głównej formatki
   F12 przełącza między formatką a powiązanym kodem. Można również wybrać ***View*** | ***Forms*** i wybrać ***Form1***
4. Kliknij na komponencie ***Action Manager*** aby go otworzyć
5. Kliknij na ***New*** w edytorze ***Action Manager*** . Otworzy się edytor kodu, z kursorem wewnątrz obsługi zdarzenia

```
procedure TForm1.FileNewExecute(Sender: TObject);
begin

end;

end.
```



6. Umieść kursor między ***begin*** a ***end*** i wpisz poniższe linie:

```
RichEdit1.Clear;
FileName := 'untitled.txt';
StatusBar1.Panels[0].Text := FileName;
```

Nasza obsługa zdarzenia powinna wyglądać tak:

```
procedure TForm1.FileNewExecute(Sender: TObject);
begin
RichEdit1.Clear;
  FileName := 'untitled.txt';
  StatusBar1.Panels[0].Text := FileName;
end;
```

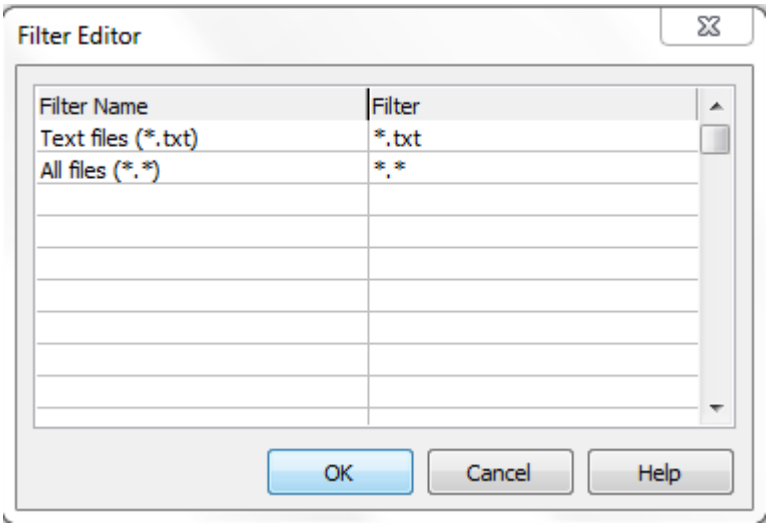
7. Wybierz ***File*** | ***Save All***

<u>Tworzenie obsługi zdarzenia dla polecenia Open</u>

Aby otworzyć plik w edytorze tekstu, użyjemy standardowego okna dialogowego Windows. Mamy już standardowe polecenie ***File*** | ***Open*** w edytorze ***Action Manager*** **,** które automatycznie zawiera to okno. Jednak musimy dopracować obsługę zdarzenia do tego polecenia.

1. Naciśnij ***F12*** aby przejść do głównej formatki i kliknij dwukrotnie ***Action Manager***
2. Wybierz działanie ***FileOpen1***
3. W ***Object Inspector*** , kliknij znak plus obok ***Dialog*** aby rozwinąć jego właściwości. ***Dialog*** odnosi się do komponentu, któy tworzy okno dialogowe ***Open.*** Delphi domyślnie nazywa to okno ***FileOpen1.OpenDialog.*** Kiedy wywoływana jest metoda ***Execute OpenDialog,*** wywołuje ona standardowe okno dialogowe dla otwierania plików.
4. Ustaw ***DefaultExt*** na ***txt***
5. Kliknij dwukrotnie w obszarze tekstu obok ***Filter*** aby wyświetlić edytor ***Filter***

- W pierwszym wierszu kolumny ***Filter Name*** wpisz ***Text files (*.txt)***, W kolumnie ***Filter*** wpisz ****.txt***
- W drugim wierszu kolumny ***Filter Name*** wpisz ***All files (*.*)*** a w kolumnie ***Filter*** wpisz ****.****
- Kliknij OK



6. Ustaw ***Title*** na ***Otwórz plik .*** Te słowa pojawią się na górze okan dialogowego ***Open***.

7. Kliknij zakładkę ***Eventsi***.Kliknij dwukrotnie w miejscu zdarzenia ***OnAccept*** abyu pojawiło się okienko dialogowe ***FileOpen1Accept***
8. Umieść kursor między ***begin*** a ***end*** i wpisz poniższe linie:

   RichEdit1.Lines.LoadFromFile(FileOpen1.Dialog.FileName);
   FileName := FileOpen1.Dialog.FileName;
   StatusBar1.Panels[0].Text := FileName;

   Teraz obsługa zdarzenia ***FileOpen*** powinna wygladać tak:

```
procedure TForm1.FileOpen1Accept(Sender: TObject);
begin
RichEdit1.Lines.LoadFromFile(FileOpen1.Dialog.FileName);
FileName := FileOpen1.Dialog.FileName;
StatusBar1.Panels[0].Text := FileName;
end;
```

Tworzenie obsługi zdarzenia dla polecenia Save

1. Naciśnij ***F12*** aby przejść do głównej formatki i kliknij dwukrotnie ***Action Manager***
2. Kliknij dwukrotnie na działaniu ***Save*** w edytorze ***Action Manager***
   Pojawi się edytor kodu z kursorem wewnątrz obsługi zdarzenia
3. Umieść kursor między ***begin*** a ***end*** i wpisz poniższe linie:

   if (FileName = 'untitled.txt') **then**
   FileSaveAs1.Execute
   else
   RichEdit1.Lines.SaveToFile(FileName);

   Kod ten mówi edytorowi tekstu aby wyświetlił okno dialogowe ***Save As*** jeśli plik nie ma nazwy, więc może ją przypisać uzzytkownik. W przeciwnym razie, zapisuje plik używając nazwy aktualnej. Okno to jest zdefiniowane w obsłudze zdarzenia dla polecenia ***Save As. FileSaveAs1BeforeExecute*** automatycznie generuje nazwę dla polecenia ***Save As***. Obsługa tego zdarzenia powinna wyglądać tak:

```
procedure TForm1.FileSaveExecute(Sender: TObject);
begin
if (FileName = 'untitled.txt') then
  FileSaveAs1.Execute
  else
  RichEdit1.Lines.SaveToFile(FileName);
end;
```

   Jest to polecenie ***File*** | ***Save***

Tworzenie obsługi zdarzenia dla polecenia Save As

Kiedy jest wywoływana metoda ***Execute SaveDialog,*** wywołuje ona standarddowe okno dialogowe Zapisz jako Windows. Stworzymy obsługę zdarzenia dla polecenia ***Save As***

1. Naciśnij ***F12*** aby i kliknij dwukrotnie komponent ***Action Manager***
2. Wybierz akcję ***Save As*** w edytorze ***Action Manager***
3. W ***Object Inspector*** przejdź na zakładkę ***Properties*** .Kliknij znak plus obok ***Dialog*** aby

roszwinąć jego właściwości. ***Dialog*** odnosi się do okna dialogowego ***Save As*** i wyświetla właściwości tego okna.

4. Ustaw ***DefaultExt*** na ***txt***
5. Kliknij dwukrotnie w obszar obok ***Filter*** aby wyświetlić edytor ***Filter.*** W edytorze tym określamy typy plików podobnie jak w oknie dilaogowym ***Open.***

- W pierwszym wierszu kolumny ***Filter Name*** wpisz ***Text files (*.txt)***, W kolumnie ***Filter*** wpisz ****.txt***
- W drugim wierszu kolumny ***Filter Name*** wpisz ***All files (*.*)*** a w kolumnie ***Filter*** wpisz ****.****
- Kliknij OK

6. Ustaw ***Title*** na ***Zapisz jako***
7. Kliknij zakładkę ***Events.*** Kliknij dwukrotnie obszar obok ***BeforeExecute*** aby otworzył się edytor kodu z kursorem w ***FileSaveAs1BeforeExecute***.
8. Między ***begin*** a ***end*** umieść tekst:

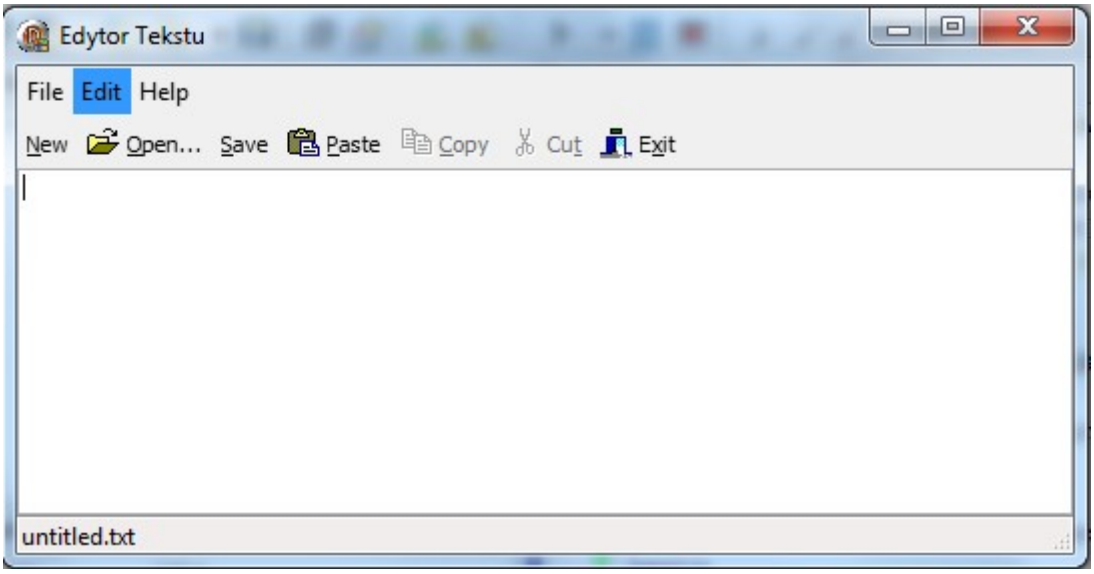
   FileSaveAs1.Dialog.InitialDir := ExtractFilePath(Filename);
9. Teraz kliknij obszar obok ***OnAccept*** aby pojawił się w edytorze kodu ***FileSaveAs1Accept***
10. Między ***begin*** a ***end*** wpisz tekst:

    FileName := FileSaveAs1.Dialog.FileName;
    RichEdit1.Lines.SaveToFile(FileName);
    StatusBar1.Panels[0].Text := FileName;

    Teraz obsługa zdarzenia ***FileSAveAs*** powinna wyglądać tak:

```
procedure TForm1.FileSaveAs1Accept(Sender: TObject);
begin
FileName := FileSaveAs1.Dialog.FileName;
RichEdit1.Lines.SaveToFile(FileName);
StatusBar1.Panels[0].Text := FileName;
end;

procedure TForm1.FileSaveAs1BeforeExecute(Sender: TObject);
begin
FileSaveAs1.Dialog.InitialDir := ExtractFilePath(Filename);
end;
```

11. Wybierz ***File |Save All*** aby zapisać zmiany
12. Aby zobaczyć działającą aplikację naciśnij ***F9.***

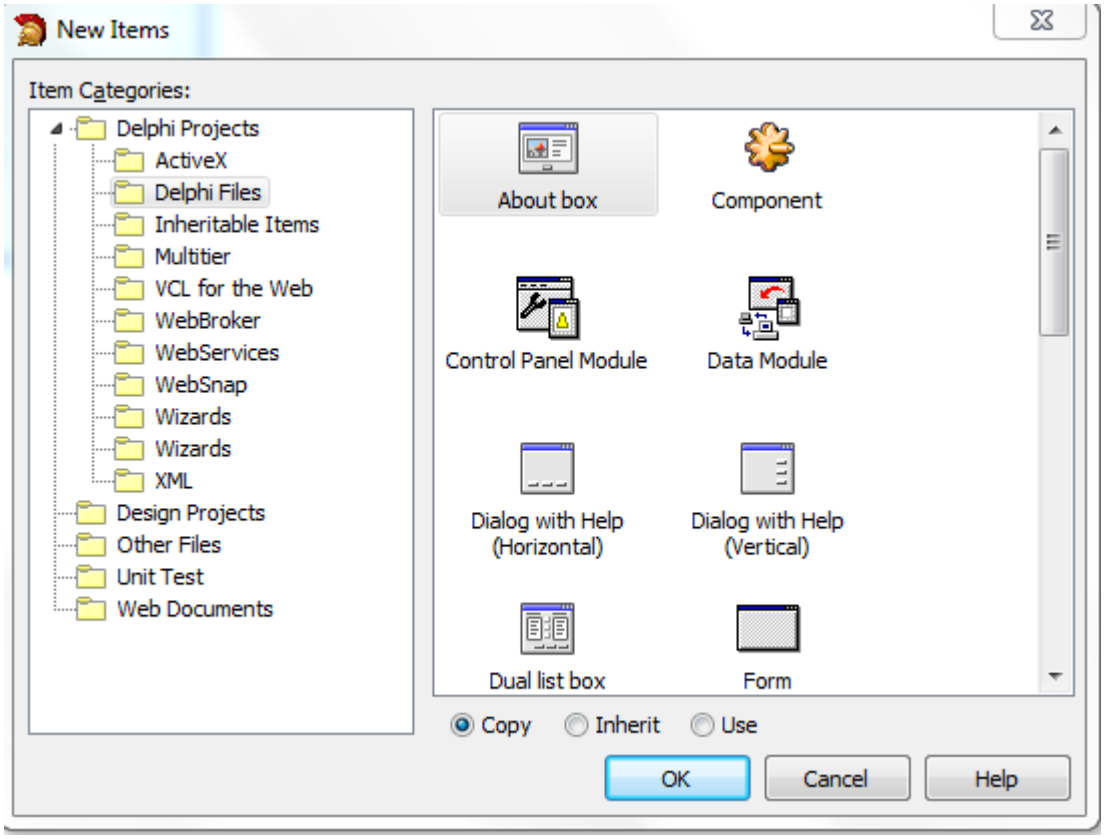


Tworzenie pola O programie

Wiele apliakcji zawiera okno ***O programie*** któe wyświetla informacje takie jak nazwa, wersja, logo i wiele innych informacji o prawach autorskich.
Ustawiamy polecenie ***HelpAbout*** w ***Action Manager***

Aby dodać ***About***

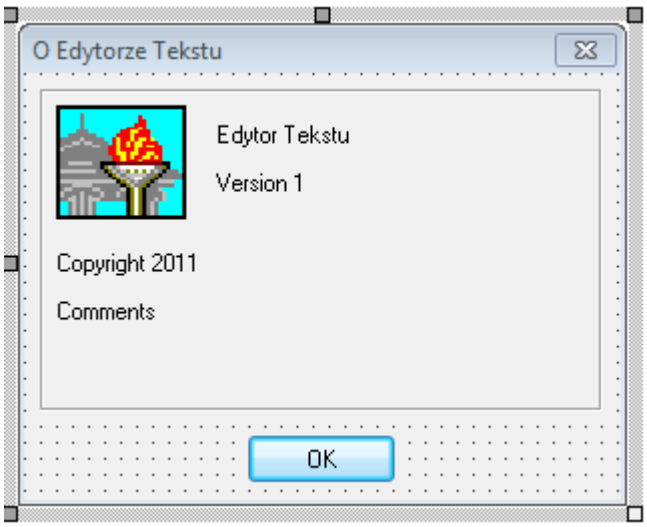
1. Wybierz ***File*** | ***New*** | ***Other*** aby wyświetlić okno dialogowe ***New Items***
2. Przejdź do ***Items Categories*** | ***Delphi Files*** i kliknij dwukrotnie ikonę ***About box***



Pojawi się wstępnie zdefiniowane pole ***O programie*** .

3. Wybierz samą formatkę (kliknij szarą siatkę) i w ***Object Inspector*** , kliknij zakładkę ***Properties*** i zmień właściwość ***Caption*** na ***O Edytorze Tekstu.***
4. Kliknij z powrotem na formatce. Aby zmienić dowolną wartość na formatce, kliknij na niej aby ją podświetlić i wpisz nową wartość.

- Zmień ***Prodcut Name*** na ***Edytor Tesktu***
- Zmień ***Version*** na ***Version 1***
- Zmień ***Copyright*** na ***Copyright 2011***



5. Zapisz pracę przez wybranie ***File | Save As*** i zapisz ją jako ***About.pas*** w katalogu
6. Kliknij zakładkę ***Unit1*** i przewiń na górę edytora kodu. Dodaj nowy moduł ***About*** i przez wpisanie słowa ***About*** na końcu listy dołączonych modułów w klauzuli ***uses.***

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, PlatformDefaultStyleActnCtrls, ActnList, ActnMan,
  ImgList, StdActns, ToolWin, ActnCtrls, ActnMenus, About;
```

7. Naciśnij ***F12*** aby wrócić do trybu projektowania .Kliknij dwukrotnie na ***Action Manager*** lub ***Action List*** aby go otworzyć
8. Kliknij dwukrotnie działanie ***Help | O programie*** aby stworzyć obsługę zdarzenia. Umieść kursor między ***begin end*** i wpisz poniższą linię:

   AboutBox.ShowModal;

9. Wybierz ***File | Save All***