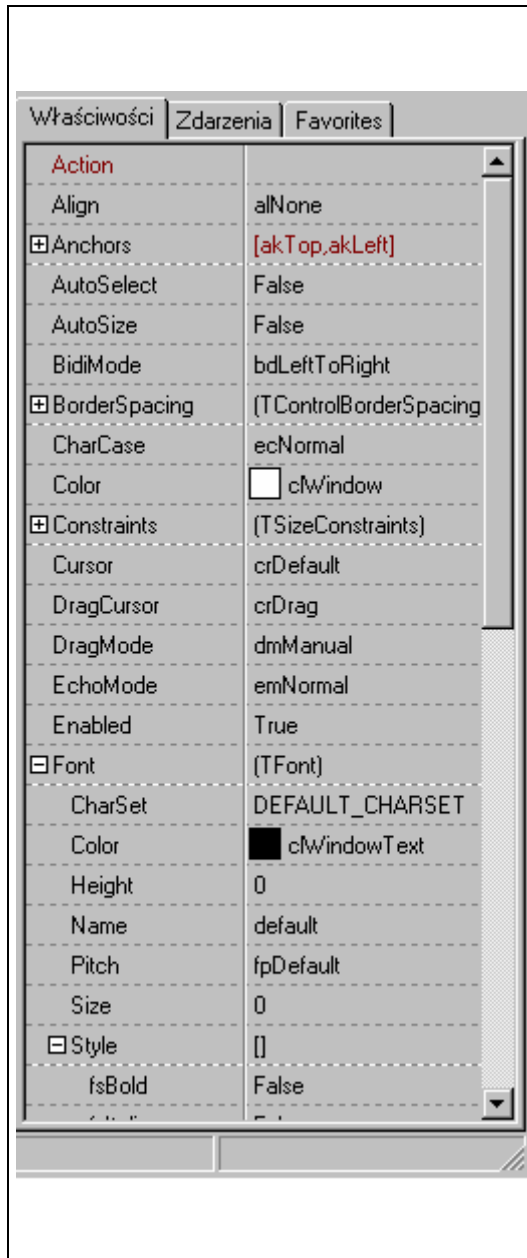


Wybrane właściwości i zdarzenia występujące w środowisku Delhi, Kylix i Lazarus

Właściwości

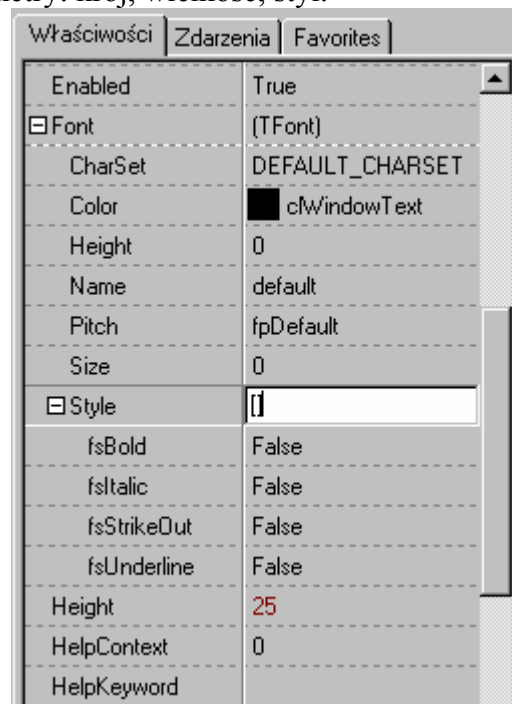


Z lewej strony ekranu znajduje się okno zawierające właściwości używanych obiektów.

Składa się z dwóch kolumn:

- Lewa zawiera nazwy poszczególnych własności
- Prawa zawiera ich wartości.

Niektóre z nich mogą mieć jeszcze inne podwartości. Np. FONT może ustalić dodatkowe parametry: krój, wielkość, styl.



Wiele wartości jest ustalanych domyślnie w trakcie inicjalizacji komponentu.

Wybór wartości może też być dokonywany ręcznie, lub wewnątrz kodu programu.

Nazwa właściwości	Opis i zakres wartości
<i>Action</i>	Pole zawiera nazwę akcji przypisanej do danego komponentu.
<i>Align</i>	<p>Domyślne położenie danego komponentu.</p> <ul style="list-style-type: none"> • alNone – bez zmian. Pozostanie w takiej wielkości jak ustawił użytkownik. Można ją przesunąć. • <i>alBottom</i> – na dole, na całą szerokość • <i>alTop</i> - na górze, na całą szerokość • <i>alLeft</i> - z lewej strony, na całą wysokość • <i>alRight</i> - z prawej strony, na całą wysokość • <i>alClient</i> - wypełnia całą dostępną przestrzeń – całą formatka. • <i>alCustom</i> - Pozostanie w takiej wielkości jak ustawił użytkownik i nie można jej przesuwać.
<i>Anchors</i>	<p>Położenie komponentu względem komponentu-rodzica.</p> <ul style="list-style-type: none"> • <i>AkTop</i> – utrzymuje stałą odległość od górnej krawędzi formatki • <i>akLeft</i>– utrzymuje stałą odległość od lewej krawędzi formatki • <i>akRight</i>– utrzymuje stałą odległość od prawej krawędzi formatki • <i>akBottom</i>– utrzymuje stałą odległość od dolnej krawędzi formatki
<i>Autosize</i>	<p>Automatyczne utrzymywanie pierwotnego rozmiaru elementu</p> <ul style="list-style-type: none"> • False – nie utrzymywanie • <i>True</i> - utrzymywanie
<i>BidiMode</i>	<p>Obsługa pisma od prawej do lewej</p> <ul style="list-style-type: none"> • <i>BdLeftToRight</i> - od lewej do prawej • <i>bdRightToLeft</i> – od prawej do lewej • <i>bdLeftToRightNoAlign</i> – bez dopasowywania • <i>bdRightToLeftNoAlign</i> - bez dopasowywania
<i>BorderSpacing</i>	
<i>Cancel</i>	<ul style="list-style-type: none"> • True • False
<i>Caption</i>	<p>Napis widoczny na elemencie.</p> <p>Może to być dowolny tekst</p>

Color	Kolor elementu Pewien zbiór standardowych kolorów lub możliwość wyboru z pełnej palety barw
Constraints	pozwała określić maksymalną i minimalną wielkość okna. Domyślnie jest zero, co znaczy, że nie ma żadnych ograniczeń. <ul style="list-style-type: none"> • <i>MaxWidth</i> - maksymalna szerokość • <i>MaxHeight</i> - maksymalna wysokość • <i>MinWidth</i> - minimalna szerokość • <i>MinHeight</i> - minimalna wysokość
Cursors	Możliwość przypisania dowolnego kursora do komponentu
Default	<ul style="list-style-type: none"> • False • True
DragCursor	Określa rodzaj kursora widocznego podczas przeciągania.
DragMode	Określa sposób przeciągania komponentu. <ul style="list-style-type: none"> • <i>dmAutomatic</i> – przeciąganie jest rozpoczynane przez przeciąganie myszką • <i>dmmanual</i> - przeciąganie jest rozpoczynane przez wywołanie StartDrag
Enabled	Informuje, czy element jest aktywny. <ul style="list-style-type: none"> • True - aktywny • <i>False</i> - nieaktywny
Font	Pozwala zmienić czcionkę domyślną. Jej zmiana zmieni parametry czcionki na wszystkich komponentach znajdujących się na formie. <ul style="list-style-type: none"> • CHARSET - zestaw znaków • COLOR - Zmienia kolor czcionki. Poza zdefiniowanymi kolorami, można też wybrać swoje. • HEIGHT - Wysokość napisu. Od własności SIZE różni się tym, że w przypadku dużych liter będą widoczne postrzępienia (litery są powiększane) • NAME - Nazwa czcionki. • SIZE - Wielkość napisu. W odróżnieniu od własności HEIGHT ta nie strzępi brzegów napisu. • STYLE - efekty czcionki: <ul style="list-style-type: none"> ○ <i>fsBold</i> - pogrubienie, ○ <i>fsUnderline</i> – podkreślenie, ○ <i>fsStrikeOut</i> – przekreślenie, ○ <i>fsItalic</i> - pochylenie
Height	Wysokość komponentu. Przybiera wartości z zakresu 0 – 9999

HelpContext	<p>Indeks pomocy</p> <p>Jest to kolejny numer od 0.</p>
HelpType	<p>Typ pomocy</p> <ul style="list-style-type: none"> • htContext – pomoc kontekstowa • <i>htKeyWord</i> - Słowo kluczowe
Hint	<p>Podpowiedź do danego elementu. Pojawia się gdy kursor się nad nim zatrzyma.</p> <p>Dowolny tekst.</p>
Left	<p>Odległość od lewej strony formatki.</p> <p>Wartość to dowolna liczba całkowita (może być ujemna)</p>
ModalResult	<p>Rodzaj domyślnej odpowiedzi skojarzonej z tym przyciskiem.</p> <ul style="list-style-type: none"> • mrNone – brak wartości • <i>mrOK</i> • <i>mrCancel</i> • <i>mrAbort</i> • <i>meRetry</i> • <i>mrIgnore</i> • <i>meYes</i> • <i>mrNo</i> • <i>mrAll</i> • <i>mrNoToAll</i> • <i>mrYesToAll</i>
Name	<p>Nazwa komponentu</p> <p>Edytor nadaje każdemu komponentowi nazwę domyślną. Użytkownik może ją zmienić.</p>
ParentBiDi Mode	<p>Dziedziczenie obsługi pisma od prawej do lewej od komponentu nadrzędnego</p> <ul style="list-style-type: none"> • False – niedziedziczone • <i>True</i> - dziedziczone
ParentFont	<p>Dziedziczenie rodzaju czcionki od komponentu nadrzędnego</p> <ul style="list-style-type: none"> • False – niedziedziczone • <i>True</i> - dziedziczone
ParentShowHint	<p>Dziedziczenie trybu wyświetlania podpowiedzi od komponentu nadrzędnego</p> <ul style="list-style-type: none"> • False – niedziedziczone • <i>True</i> - dziedziczone
PopupMenu	<p>Informacja, jakie menu <i>PoPuMenu</i> jest skojarzone z tym komponentem</p>

ShowHint	<p>Pozwala na pojawienie się podpowiedzi do danego elementu</p> <ul style="list-style-type: none"> • False – niewidoczna podpowiedź • True – widoczna podpowiedź
TabOrder	<p>Kolejność przeskakiwania fokusa za pomocą klawisza TAB.</p> <p>Wartość to kolejny numer zaczynając od 0.</p>
TabStop	<p>Informacja, czy komponent ma być uwzględniany przy przeskakiwaniu klawiszem TAB.</p> <ul style="list-style-type: none"> • True – TAB uwzględnia ten komponent • False – TAB nie uwzględnia
Tag	<p>Dodatkowa własność komponentu. Pozwala na wprowadzenie dowolnej wartości.</p> <p>Może zastępować zmienną w kodzie.</p> <p>Wartość to dowolna liczba całkowita.</p>
Top	<p>Odległość od górnej krawędzi formatki.</p> <p>Wartość to dowolna liczba całkowita (może być ujemna)</p>
Visible	<p>Informacja czy element jest widoczny.</p> <ul style="list-style-type: none"> • True – widoczne • False – niewidoczne
Width	<p>Szerokość komponentu.</p> <p>Przybiera wartości z zakresu 0 – 9999</p>

Zdarzenia



Druga zakładka zawiera zdarzenia dla danego obiektu.

Zdarzenia to podprogramy, które reagują na określone wydarzenia związane z danym komponentem.

Składa się z dwóch kolumn:

- Lewa zawiera nazwy poszczególnych zdarzeń
- Prawa zawiera procedury i funkcje przypisane do nich.

Zdarzenie **OnClick** odpowiada sytuacji, gdy dany komponent zostanie kliknięty myszką. Przypisany podprogram wykona daną operację.

Różnym zdarzeniom można przypisać ten sam podprogram, np. zdarzeniom **OnClick** i **OnEnter**.

Nazwa zdarzenia	Opis
<i>OnActivate</i>	Aktywacja okna
<i>OnChangeBounds</i>	
Zdarzenia związane z kliknięciem	
<i>OnClick</i>	Zdarzenie <i>OnClick</i> występuje podczas kliknięcia klawiszem myszy w obszarze danej kontrolki — jest to chyba najczęściej używane zdarzenie VCL.
<i>OnDblClick</i>	Zdarzenie jest generowane podczas dwukrotnego kliknięcia danego obiektu. Obsługiwane jest tak samo jak zdarzenie <i>OnClick</i> — wraz ze zdarzeniem nie są dostarczane żadne dodatkowe parametry.
Zdarzenia związane z zamykaniem okna	

<i>OnClose</i>	<p>Zdarzenie związane z zamykaniem okna.</p> <p>Dzięki zdarzeniu <i>OnClose</i> można zareagować podczas próby zamknięcia okna. Wraz ze zdarzeniem dostarczany jest parametr <i>Action</i>, który określa „akcję” do wykonania.</p>	
	Właściwości klasy TCloseAction	
	Wartość	Opis
	<i>caNone</i>	Nic się nie dzieje — można zamknąć okno
	<i>caHide</i>	Okno nie jest zamykane, a jedynie ukrywane
	<i>caMinimize</i>	Okno jest minimalizowane zamiast zamykania
	<i>caFree</i>	Okno zostaje zwolnione, co w efekcie powoduje zamknięcie
<i>OnCloseQuery</i>	<p>Zdarzenia <i>OnCloseQuery</i> możesz użyć, aby zapytać użytkownika, czy rzeczywiście chce zamknąć okno. Zdarzenia posiada parametr <i>CanClose</i>; jeżeli nastąpi jego zmiana na <code>False</code>, okno nie zostanie zamknięte.</p>	
<i>OnPaint</i>	<p>Zdarzenie <i>OnPaint</i> występuje zawsze wtedy, gdy okno jest wyświetlane i umieszczane na pierwszym planie. W zdarzeniu tym będzie umieszczany kod, którego zadaniem będzie „malowanie” w obszarze formularza.</p>	
<i>OnResize</i>	<p>Zdarzenie <i>OnResize</i> występuje tylko wtedy, gdy użytkownik zmienia rozmiary komponentu. Możesz dzięki temu zdarzeniu odpowiednio zareagować na zmiany lub nie dopuścić do nich.</p>	
<i>OnShow</i> <i>OnHide</i>	<p>Te dwa zdarzenia informują o tym, czy aplikacja jest ukrywana czy pokazywana. Pokazanie lub ukrycie formularza dokonywane jest za pomocą metody <i>Show</i> lub <i>Hide</i> klasy <i>TForm</i>.</p>	
	Zdarzenia związane z obsługą myszy	
<i>OnMouseDown</i>	kliknięcie w obszarze komponentu	
<i>OnMouseMove</i>	przesunięcie kursora nad komponentem	
<i>OnMouseLeave</i>	Opuszczenie kursorem obszaru komponentu	
<i>OnMouseUp</i>	puszczenie klawisza myszy	
<i>OnMouseWheel</i>	użycie rolki myszki	
<i>OnMouseWheelDown</i>	przesunięcie rolki w górę	
<i>OnMouseWheelUp</i>	przesunięcie rolki w dół	
	<p>Wraz z tymi zdarzeniami do aplikacji może być dostarczana informacja o położeniu kursora myszy oraz o przycisku myszy, który został naciśnięty (lewy, środkowy, prawy). Informacje te zawiera parametr <i>Button</i> klasy <i>TMouseButton</i></p>	
	Możliwe wartości klasy TMouseButton	
	<i>mbLeft</i>	Naciśnięto lewy przycisk myszki
	<i>mbMiddle</i>	Naciśnięto środkowy przycisk myszki
	<i>mbRight</i>	Naciśnięto prawy przycisk myszki.

<p>Wraz ze zdarzeniami obsługi myszy może być dostarczany również parametr <i>Shift</i>, który jest obecny także w zdarzeniach klawiaturowych (<i>OnKeyUp</i>, <i>OnKeyDown</i>).</p>	
<p>Możliwe wartości klasy TShiftState</p>	
<i>ssShift</i>	Klawisz Shift jest przytrzymany w momencie wystąpienia zdarzenia
<i>ssAlt</i>	Klawisz Alt jest przytrzymany w momencie wystąpienia zdarzenia
<i>ssCtrl</i>	Klawisz Ctrl jest przytrzymany w momencie wystąpienia zdarzenia
<i>ssLeft</i>	Przytrzymany jest również lewy przycisk myszki
<i>ssRight</i>	Przytrzymany jest także prawy przycisk myszki
<i>ssMiddle</i>	Przytrzymany jest środkowy przycisk myszy
<i>ssDouble</i>	Nastąpiło dwukrotne kliknięcie
<p>Zdarzenia związane z dokowaniem i przesuwaniam</p>	
<i>OnDockDrop</i>	Zdarzenie <i>OnDockDrop</i> generowane jest w momencie, gdy użytkownik próbuje osadzić jakiś inny komponent w obrębie naszego obiektu.
<i>OnDockOver</i>	Zdarzenie to występuje w momencie, gdy jakiś inny komponent jest przeciągany nad naszym obiektem.
<i>OnStartDock</i>	Zdarzenie występuje w momencie, gdy rozpoczynasz przeciąganie jakiegoś obiektu. Warunkiem wystąpienia tego zdarzenia jest ustawienie właściwości <i>DragKind</i> na wartość <i>dkDock</i> .
<i>OnStartDrag</i>	Zdarzenie występuje tylko wówczas, gdy właściwość <i>DragKind</i> komponentu jest ustawiona na <i>dkDrag</i> . Wykorzystaj to zdarzenie w momencie, kiedy chcesz zareagować na przeciąganie obiektu.
<i>OnEndDrag</i> , <i>OnEndDock</i>	Pierwsze ze zdarzeń wykorzystaj w przypadku, gdy chcesz zareagować na zakończenie procesu przeciągania; drugie natomiast występuje w przypadku zakończenia procesu „przeciągnij i upuść”.
<i>OnDragDrop</i>	Zdarzenie to generowane jest w momencie, gdy w komponencie następuje „upuszczenie” danych przeciąganych metodą drag and drop .
<i>OnDragOver</i>	Zdarzenie to generowane jest w momencie, gdy nad danym komponentem użytkownik przeciąga kursor z przeciąganymi danymi.
<p>Zdarzenia związane z obsługą klawiatury</p>	
<i>OnEnter</i>	Zdarzenie występujące po naciśnięciu klawisza ENTER
<i>OnKeyDown</i>	Zdarzenie występujące po naciśnięciu dowolnego klawisza
<i>OnKeyPress</i>	Zdarzenie występujące po przytrzymaniu dowolnego klawisza
<i>OnKeyUp</i>	Zdarzenie występujące po puszczeniu dowolnego klawisza
<p>Wraz ze zdarzeniami obsługi klawiatury może być dostarczany również parametr <i>Shift</i>.</p>	