

# **PORADNIKI**

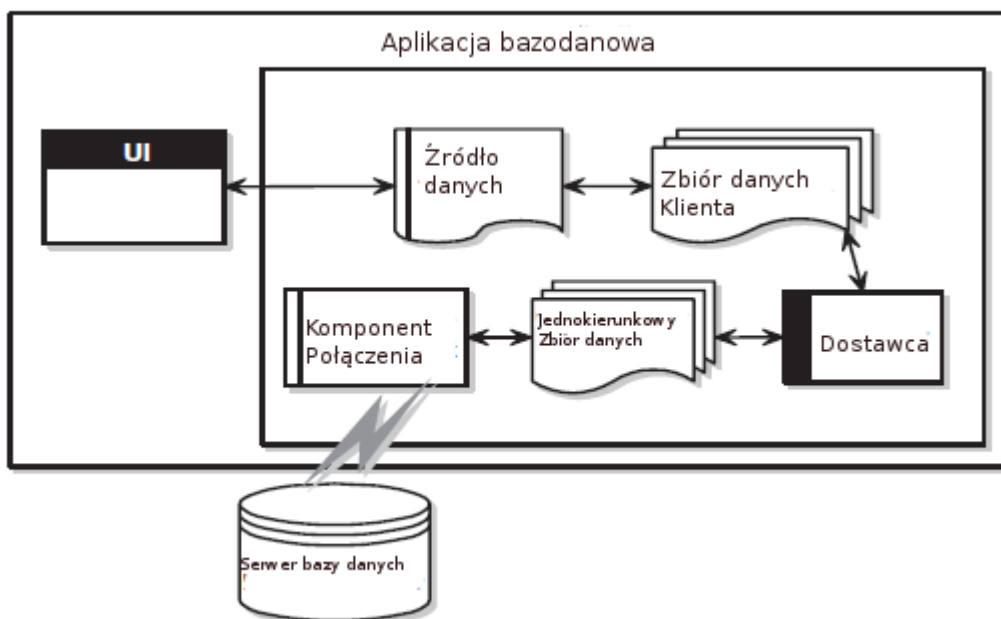
**Jak napisać bazę danych w Kylix**

## Wstęp

Przedstawię ci sposób tworzenia aplikacji bazowej InterBase. Użyjemy do tego IDE Delphi. Zakładam, że znasz Linux i czytałeś trochę o programowaniu w Kylix i IDE.

## Omówienie architektury bazodanowej

Architektura aplikacji bazodanowej może wydawać się skomplikowana na pierwszy rzut oka, ale zastosowanie wielu komponentów upraszcza projektowanie i zarządzanie rzeczywistymi aplikacjami bazy danych. Aplikacja bazodanowa zawiera trzy główne części: interfejs użytkownika, zbiór komponentów dostępu do danych i samą bazę danych. Tu stworzymy aplikację bazodanową dbExpress. Inne bazy danych mają podobną architekturę. Interfejs użytkownika zawiera dane kontrolki, takie jak siatka dzięki czemu użytkownicy mogą edytować i wysyłać dane do bazy danych. Komponenty dostępu do danych zawierają źródło danych, zbiór danych klienta, dostawcę danych, jednokierunkowy zbiór danych i komponenty połączenia. Źródło danych działa jako łącznik między interfejsem użytkownika a zbiorem danych klienta. Zbiór danych klienta jest sercem aplikacji ponieważ składa się ze zbioru rekordów z podstawowej bazy danych, która jest buforowana w pamięci. Dostawca przesyła dane między zbiorem danych klienta a jednokierunkowym zbiorem danych, pobierając dane bezpośrednio z bazy danych. W końcu element połączenia ustanawia połączenie z bazą danych. Każdy typ jednokierunkowego zbioru danych używa różnych typów komponentów połączenia.



## Tworzenie nowego projektu

Zanim zaczniemy, stworzymy folder przechowujący pliki źródłowe.

Potem otworzymy i zapiszemy nowy projekt

1. Stwórz folder BazaD do przechowywania plików projektu jakie stworzysz podczas pracy
2. Użyj domyślnego projektu już stworzonego kiedy uruchamiasz IDE Delphi lub zacznij nowy projekt wybierając File | New Application
3. Wybierz File | Save As aby zapisać pliki na dysk. Kiedy pojawi się okno dialogowe Save As, przejdź do folderu BazaD i zapisz każdy plik używając nazw domyślnych.

Później możesz zapisywać swoją pracę w dowolnym czasie wybierając File | Save All

## **Ustawienie komponentów dostępu do danych**

Komponenty dostępu do danych są komponentami, które reprezentują dane (zbiór danych), i komponenty które łączą ten zbiór danych z innymi częściami twojej aplikacji. Każdy komponent dostępu do danych wskazuje kolejny niższy komponent. Na przykład, źródło danych wskazuje źródło danych klienta, ten wskazuje dostawcę i tak dalej. Po skonfigurowaniu komponentów dostępu do danych, można dodać najniższy komponent jako pierwszy. W kolejnych sekcjach będziemy dodawać komponenty aby utworzyć połączenia z bazą danych, jednokierunkowym zbiorem danych, dostawcą, zbiorem danych klienta i źródłem danych. Potem stworzymy interfejs użytkownika dla tej aplikacji. Komponenty te są umieszczone na stronach dbExpress, Data access i Data Controls palety Component

## **Konfigurowanie połączeń z bazą danych**

Strona dbExpress zawiera zbiór komponentów, które dostarczają szybkiego dostępu do serwera baz danych SQL. Musisz dodać komponent połączenia tak abyś mógł łączyć się z tą bazą danych. Typ użytego komponentu zależy od tego jakiego komponentu zbioru danych użyjesz. Tu będziemy używali komponentów TSQLConnection i TSQLDataSet.

Aby dodać komponent połączenia dbExpress:

1. Upewnij się, że serwer InterBase jest uruchomiony, tak abyś mógł się łączyć ze swoją bazą danych, jaką stworzysz.
2. Kliknij stronę dbExpress na palecie Component i kliknij dwukrotnie komponent TSQLConnection aby umieścić go na formatce. Znajdź komponent TSQLConnection, wskaż jego ikonkę na tej palecie na chwilę; podpowiedź Pomocy pokaże ci nazwę tego komponentu. Domyślnie jest to SQLConnection1. Komponent ten jest niewidoczny, więc nic się nie dzieje kiedy wstawiasz go na formatkę. Dla ułatwienia będziemy wstawiać wszystkie komponenty niewidoczne w jednej linii u góry formatki. Aby wyświetlić tytuły komponentów umieszczonych na formatce, wybierz Tools | Environment Options | Designer i kliknij Show component captions
3. W Object Inspector, ustaw właściwość ConnectionName na IBConnection (z listy rozwijanej).

4. Ustaw właściwość LonginPrompt na False. Przez ustawienie tej właściwości na False nie będziesz zmuszony logować się do bazy danych przy każdej próbie dostępu do bazy.
5. Kliknij dwukrotnie komponent TSQLConnection a wyświetli się edytor Connection. Używamy edytora Connection dla wybrania konfiguracji połączenia dla komponentu TSQLConnection lub edycji połączenia przechowywanego w pliku dbxconnections w katalogu .borland. W edytorze tym, określamy ścieżkę dostępu do pliku bazodanowego nazwanego employee.gdb w systemie. Stworzymy tu prostą bazę danych InterBase, employee.gdb, dostarczaną wraz z Kylix. Domyślnie, instalacja InterBase umieszcza employee.gdb w /opt/interbase/examples
6. Sprawdź pola User\_Name i Password dla dopuszczalnych wartości. Jeśli nie masz zmodyfikowanych tych wartości domyślnych, nie musisz zmieniać tych pól. Jeśli dostęp do bazy danych będzie zarządzany przez kogoś innego, będziesz musiał ustawić nazwę użytkownika i hasło dla dostępu do tej bazy.
7. Kiedy sprawdziłeś i ustawiłeś te pola, kliknij OK aby zamknąć edytor Connection i zapisz swoje zmiany. Zmiany są zapisane do pliku dbxconnections a wybrane połączenie jest przypisane jako wartość właściwości ConnectionName komponentu SQLConnection.
8. Wybierz File | Save All aby zapisać swój projekt

### **Konfiguracja jednokierunkowego zbioru danych**

Podstawowa aplikacja bazodanowa używa zbioru danych dla dostępu do informacji bazy danych. W aplikacji dbExpress, używamy jednokierunkowego zbioru danych. Jednokierunkowy zbiór danych odczytuje danych z bazy danych ale nie aktualizuje danych

Aby dodać jednokierunkowy zbiór danych:

1. Ze strony dbExpress, dodaj TSQLDataSet na górę formatki.
2. W Object Inspector ustaw właściwość SQLConnection ma SQLConnection1 (połączenie bazy danych stworzony poprzednio)
3. Ustaw właściwość CommandText na Select \* from sales dla określenia polecenia, które wykonuje baza danych. Możesz wpisać albo instrukcję Select w Object Inspector albo kliknij wielokropkę po prawej stronie CommandText dla wyświetlenia edytora CommandText, gdzie możesz zbudować swoją własną instrukcję zapytania.
4. Ustaw Active na True aby otworzyć zbiór danych
5. Wybierz File | Save All aby zapisać projekt.

### **Konfiguracja dostawcy, zbioru danych klienta i źródła danych**

Strona Data Access zawiera komponenty, które mogą być używane z mechanizmem dostępu do danych, nie tylko dbExpress. Komponenty Dostawcy są sposobem w jaki zbiór danych klienta uzyskuje swoje dane z innych zbiorów danych. Dostawca odbiera żądanie danych ze zbioru danych klienta, pobiera dane, pakuje je i zwraca dane do

zbioru danych klienta. W dbExpress, Dostawca odbiera aktualizacje ze zbioru danych klienta i serwuje je serwerowi bazy danych.  
Aby dodać dostawcę :

1. Ze strony Data Access ,wstaw komponent TdataProvider na górę formatki
2. W Object Inspector ustaw właściwość Dostawcy DataSet na SQLDataSet1.

Zbiór danych klienta buforuje swoje dane w pamięci. Buforuje także aktualizacje wysyłane do bazy danych. Możesz użyć zbioru danych klienta do kontrolki data-aware na interfejsie użytkownika korzystając z komponentu źródła danych  
Aby dodać zbiór danych klienta:

1. Ze strony Data Access wstaw komponent TclientDataSet.
2. Ustaw właściwość ProviderName na DataSetProvider1
3. Ustaw właściwość Active na True pozwalając na przekazywanie danych do twojej aplikacji.

Połączenie źródła danych zbioru danych klienta z kontrolkami data-aware. Każda kontrolka data-aware musi być powiązana z komponentem źródła danych mając dane do wyświetlenia i manipulowania. Podobnie, wszystkie zbiory danych muszą być powiązane z komponentem źródła danych dla swoich danych dla wyświetlenia i manipulowania w kontrolkach data-aware na formatce.  
Aby dodać źródło danych:

1. Ze strony Data Access wstaw komponent TdataSource.
2. Ustaw właściwość źródła danych na DataSet na ClientDataSet1
3. Wybierz File | Save All dla zapisania projektu

Tak więc masz dodane niewizualną infrastrukturę bazy danych do swojej aplikacji. Teraz zarysujemy interfejs użytkownika

## **Projektowanie interfejsu użytkownika**

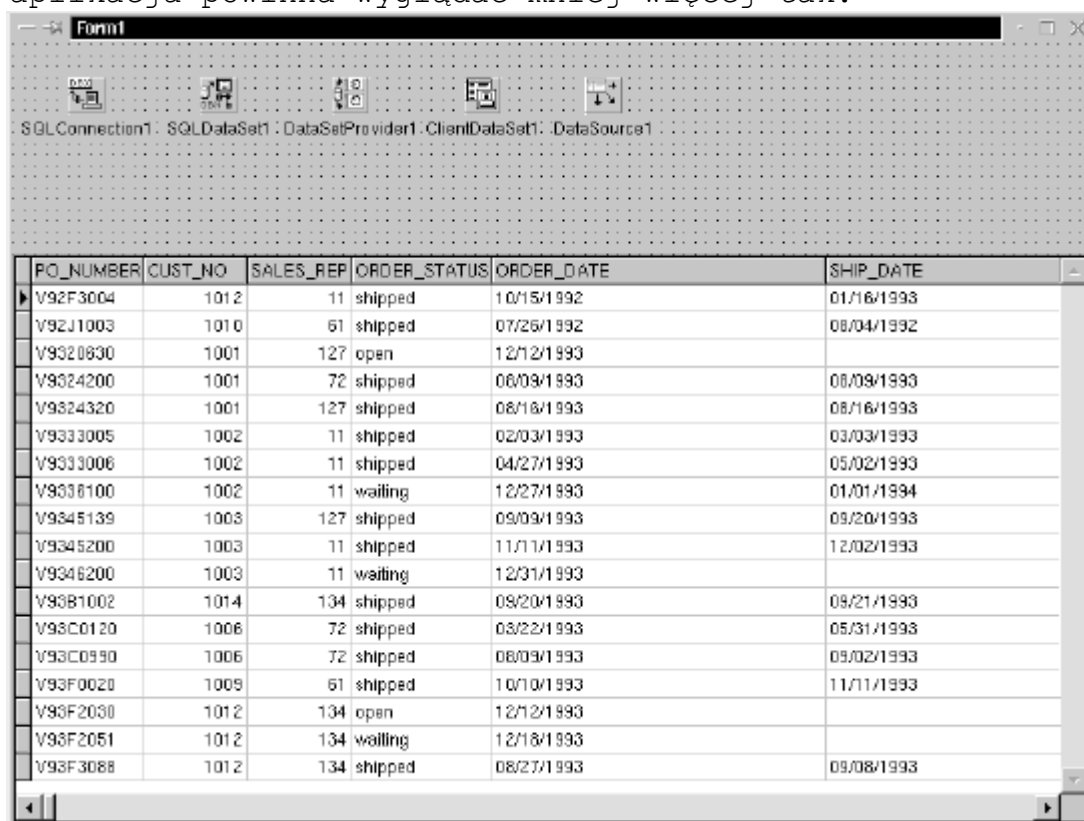
Teraz musimy dodać wizualne kontrolki do aplikacji aby użytkownik mógł zobaczyć dane, edytować i zapisywać je. Strona Data Controls dostarcza kontrolki data-aware, które działają z danymi w bazie danych i budują interfejs użytkownika. Wyświetlimy bazę danych na siatce i dodamy kilka poleceń i pasek nawigacyjny.

## **Tworzenie siatki i paska nawigacyjnego**

Aby stworzyć interfejs dla tej aplikacji:

1. Możesz zacząć od dodania siatki na formatkę. Ze strony Data Controls, wstaw komponent TDBGrid na formatkę
2. Ustaw właściwość DBGRID na kotwicę siatki. Kliknij + obok Anchors w Object Inspector aby wyświetlić allLeft,akTop,akRight i akBottom;ustaw je wszystkie na True. Najłatwiejszym sposobem zrobienia tego jest podwójne kliknięcie na False obok każdej właściwości w Object Inspector.

3. Wyrównaj siatkę do dołu formatki przez ustawienie właściwości Align na alBottom. Możesz również powiększyć rozmiar siatki przez przeciągnięcie jej lub ustawienie właściwości Height na 400.
4. Ustaw właściwość DataSource na DataSource1. Kiedy to zrobisz, siatka wypełni się danymi z bazy danych pracowników. Jeśli siatka nie wyświetla danych upewnij się, że poprawnie ustawiłeś właściwości wszystkich obiektów na formatce. Twoja aplikacja powinna wyglądać mniej więcej tak:



PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE	SHIP_DATE
V92F3004	1012	11	shipped	10/15/1992	01/16/1993
V92J1003	1010	61	shipped	07/26/1992	08/04/1992
V9320630	1001	127	open	12/12/1993	
V9324200	1001	72	shipped	06/09/1993	06/09/1993
V9324320	1001	127	shipped	08/16/1993	08/16/1993
V9333005	1002	11	shipped	02/03/1993	03/03/1993
V9333006	1002	11	shipped	04/27/1993	05/02/1993
V9336100	1002	11	waiting	12/27/1993	01/01/1994
V9345139	1003	127	shipped	09/09/1993	09/20/1993
V9345200	1003	11	shipped	11/11/1993	12/02/1993
V9346200	1003	11	waiting	12/31/1993	
V93B1002	1014	134	shipped	09/20/1993	09/21/1993
V93C0120	1006	72	shipped	03/22/1993	05/31/1993
V93C0990	1006	72	shipped	08/09/1993	09/02/1993
V93F0020	1009	61	shipped	10/10/1993	11/11/1993
V93F2030	1012	134	open	12/12/1993	
V93F2051	1012	134	waiting	12/18/1993	
V93F3088	1012	134	shipped	08/27/1993	09/08/1993

DBGrid wyświetla dane w czasie projektowania, podczas gdy pracujesz w IDE. Pozwala to na weryfikację połączeń z bazą danych. Nie możesz jednak edytować danych w czasie projektowania; edycja danych w tabeli jest możliwa po uruchomieniu aplikacji.

5. Ze strony Data Controls, ustaw kontrolkę TDBNavigator na formatce. Navigator bazy danych jest narzędziem do przechodzenia przez dane w zbiorze danych (używając na przykład strzałek w przód, w tył) i wykonywania operacji na danych.
6. Ustaw właściwość DataSource paska nawigatora na DataSource1 aby nawigator mógł zajrzeć w dane w zbiorze danych klienta.
7. Ustaw właściwość ShowHint paska nawigatora na True (Ustawienie to pozwala na pojawianie się podpowiedzi Pomocy kiedy kursor jest umieszczony nad każdą pozycją w pasku nawigatora w czasie uruchamiania).
8. Wybierz File | Save All aby zapisać projekt.
9. Naciśnij F9 aby skompilować i uruchomić projekt. Możesz również uruchomić projekt przez kliknięcie przycisku Run na pasku narzędziowym Debug, lub przez wybranie Run z menu Run.

PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE	SHIP_DATE	DA
V32F3004	1012	11	shipped	10/15/1992	01/18/1993	01/
V92J1003	1010	61	shipped	07/26/1992	08/04/1992	09/
V9320630	1001	127	open	12/12/1993		12/
V9324200	1001	72	shipped	08/09/1993	05/09/1993	08/
V9324320	1001	127	shipped	08/16/1993	08/16/1993	09/
V9333005	1002	11	shipped	02/03/1993	03/03/1993	
V9333006	1002	11	shipped	04/27/1993	05/02/1993	05/
V9336100	1002	11	waiting	12/27/1993	01/01/1994	01/
V9345139	1003	127	shipped	09/09/1993	09/20/1993	10/
V9345200	1003	11	shipped	11/11/1993	12/02/1993	12/
V9346200	1003	11	waiting	12/31/1993		01/
V93B1002	1014	134	shipped	09/20/1993	09/21/1993	09/
V93C0120	1006	72	shipped	03/22/1993	05/31/1993	04/
V93C0990	1006	72	shipped	08/09/1993	09/02/1993	
V93F0020	1009	61	shipped	10/10/1993	11/11/1993	11/
V93F2030	1012	134	open	12/12/1993		
V93F2051	1012	134	waiting	12/18/1993		03/
V93F3008	1012	134	shipped	08/27/1993	09/08/1993	

Kiedy uruchomisz projekt, otworzy się program w oknie jakie zaprojektowałeś na formatce. Możesz przetestować pasek nawigacyjny z bazą danych pracowników. Na przykład, możesz przechodzić z rekordu na rekord używając strzałek poleceń, dodawać rekordy używając polecenia + i usuwać rekordy używając polecenia -.

### Dodanie obsługi menu

Chociaż program ma już sporo funkcji, brak mu wielu funkcji związanych z GUI aplikacji. Na przykład, większość aplikacji implementuje menu i przyciski aby uczynić je łatwiejszymi do użycia. W tym miejscu dodamy listę działań. Chociaż możemy tworzyć menu, paski narzędzi i przyciski bez używania listy akcji, lista akcji ułatwia projektowanie i zarządzanie przez zcentralizowanie poleceń użytkownika

1. Jeśli aplikacja jest jeszcze uruchomiona, kliknij X w prawym górnym rogu aby zamknąć aplikację i wrócić do formy projektowania.
2. Ze strony Common Controls palety Component, wstaw komponent ImageList na formatkę. Przesu go do zbioru komponentów niewizualnych. ImageList będzie zawierała ikony przedstawiające standardowe działania takie jak wytnij i wklej.
3. Ze strony Standard palety Component wstaw komponent

- ActionList na formatkę. Ustaw właściwość listy akcji Images na ImageList1.
4. Kliknij dwukrotnie na listę akcji aby wyświetlić edytor Action List.
  5. Kliknij w edytor Action List i wybierz New Standard Action. Pojawi się pole lista Standard Actions.
  6. Wybierz następujące działania: TEditCopy, TeditCut i TeditPaste. Potem kliknij OK. Te standardowe działania pojawiają się w edytorze Action List z domyślnymi obrazkami już z nimi powiązanymi.
  7. Kliknij prawym klawiszem w edytorze Action List i wybierz New action i dodaj inne działania (nie wspierane domyślnie) Action1 jest dodawane domyślnie. W Object Inspector ustaw jego właściwość Caption na Update Now!. To same działania będą używane w menu i przyciskach. Później dodamy obsługę zdarzeń aktualizujące bazę danych
  8. Kliknij (No Category), kliknij prawym klawiszem i wybierz New Action i dodaj inne działanie. Dodaj Action2. Ustaw właściwość Caption na E&xit
  9. Kliknij X (w prawym górnym rogu) aby zamknąć edytor Action List
  10. Wybierz File | Save All aby zapisać ten projekt

### **Dodawanie menu**

W tej części dodamy paske menu głównego z dwoma rozwijanymi menu - File i Edit - i dodamy pozycje menu do każdej używając działań w liście akcji.

1. Ze strony Standard palety Component wstaw komponent TmainMenu na formatkę. Przeciągnij ją do innym niewizualnych komponentów.
2. Ustaw właściwość Images menu głównego na ImageList1 powiązanej listy obrazków z pozycjami menu
3. Kliknij dwukrotnie komponent TmainMenu aby wyświetlić Menu Designer
4. Wpisz &File aby ustawić właściwość Caption pozycji menu pierwszego górnego poziom i naciśnij Enter
5. Wybierz pustą pozycję menu poniżej menu File. Ustaw właściwość Action pustej pozycji menu na Action2. Menu Exit pojawi się poniżej File.
6. Kliknij pozycję menu z drugiego poziomu (na prawo od File). Ustaw właściwość Caption na &Edit i naciśnij Enter. Zaznacz pustą pozycję menu, która pojawiła się poniżej menu Edit.
7. W Object Inspector ustaw właściwość Action na EditCut1 i naciśnij Enter. Nagłówek pozycji jest automatycznie ustawiany na Cut i domyślna bitmapa nożyczek pojawi się w menu.
8. Zaznacz kolejną pustą pozycję menu (pod Cut) i ustaw jej właściwość Action na EditCopy1 (domyślna bitmapa kopiowania pojawia się w menu)
9. Zaznacz kolejną pustą pozycję menu (pod Copy) i ustaw jego właściwość Action na EditPaste1 (domyślna bitmapa wklejania pojawi się w menu)



10. Zaznacz kolejną pustą pozycję menu (pod Paste) i ustaw jego właściwość Caption na myślnik ( - ) dla stworzenia linii oddzielającej w menu. Naciśnij Enter
11. Zaznacz kolejną pustą pozycję menu (pod linią oddzielającą) i ustaw jego właściwość Action na Action11. Pozycja menu wyświetli Update Now!
12. Kliknij X aby zamknąć Menu Designer
13. Wybierz File | Save All aby zapisać projekt
14. Naciśnij F9 lub Run na pasku narzędziowym aby uruchomić program i zobaczyć jak wygląda

PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE
V932F3004	1012	11	shipped	10/15/1992
V932J1003	1010	61	shipped	07/26/1992
V9320630	1001	127	open	12/12/1993
V9324200	1001	72	shipped	08/09/1993
V9324320	1001	127	shipped	08/16/1993
V9333005	1002	11	shipped	02/03/1993
V9333006	1002	11	shipped	04/27/1993
V9338100	1002	11	waiting	12/27/1993
V9345139	1003	127	shipped	09/09/1993
V9345200	1003	11	shipped	11/11/1993
V9348200	1003	11	waiting	12/31/1993
V93B1002	1014	134	shipped	09/20/1993
V93C0120	1006	72	shipped	03/22/1993
V93C0990	1006	72	shipped	08/09/1993
V93F0020	1009	61	shipped	10/10/1993
V93F2030	1012	134	open	12/12/1993
V93F2051	1012	134	waiting	12/18/1993
V93F3088	1012	134	shipped	08/27/1993

Wiele z poleceń menu Edit i paska nawigacyjnego funkcjonują tym razem. Copy i Cut są szare w menu Edit dopóki nie zaznaczysz jakiegoś tekstu w bazie danych. Możesz użyć paska nawigacyjnego do przechodzenia od rekordu do rekordu w bazie danych, wstawiać rekord lub usuwać rekord. Polecenie Update jeszcze nie działa. Zamknij aplikację kiedy jesteś gotowy do dalszego działania.

### **Dodawanie przycisku**

Opisze tu jak dodać przycisku Update Now do aplikacji. Przycisk ten jest używany do stosowania dowolnej edycji jaką wykonuje na bazie danych, takich jak edytowanie rekordów, dodawanie nowych rekordów lub usuwania rekordów. Aby dodać przycisk:

1. Ze strony Standard palety Component, wstaw Tbutton na formatkę (Zaznacz komponent a potem kliknij formatkę obok paska nawigacyjnego)
2. Ustaw właściwość Action na Action1

Nagłówek przycisku zmieniamy na Update Now! Kiedy uruchamiasz aplikację, będzie on szary dopóki nie dodamy obsługi zdarzeń dla jego działania.

## **Wyświetlanie tytułu i obrazka**

Możesz dodać nazwę firmy i obrazek czyniący twoją aplikację bardziej profesjonalną:

1. Ze strony Standard palety Component, wstaw komponent TLabel na formatkę. Kylix nazywa je domyślnie Label1
2. W Object Inspector, zmień właściwość Caption etykiety na World Corp lub inną nazwę firmy
3. Zmień czcionkę nazwy firmy przez kliknięcie właściwości Font. Kliknij kropki , które pojawiają się po prawej stronie a w oknie dialogowym Font, zmień czcionkę na Helvetica Bold, 16. Kliknij OK
4. Umieść etykietę w prawym górnym rogu
5. Ze strony Additional palety Component wstaw komponent TImage obok etykiety. Kylix nazywa ten komponent Image1 domyślnie
6. Aby dodać obrazek do komponentu Image1, kliknij właściwość Picture. Kliknij kropki aby wyświetlić Picture Editor
7. W Picture Editor, wybierz Load i nawiguj do ikony katalogu dostarczonego z tym produktem. Domyślnym położeniem jest {katalog instalacyjny}/images/icons. Na przykład, jeśli Kylix jest zainstalowany w katalogu /usr/local/kylix, spójrz do /usr/local/kylix/images/icons
8. Kliknij dwukrotnie na earth.ico. Kliknij OK aby załadować obrazek i zamknij Picture Editor
9. Domyślny rozmiar obszaru obrazka do rozmiar zdjęcia. Umieść obrazek blisko etykiety
10. Aby wyrównać tekst i obrazek, zaznacz oba obiekty na formatce, kliknij prawym klawiszem i wybierz Align. W oknie dialogowym Alignmnet, pod Vertical, kliknij Bottoms
11. Wybierz File | Save All aby zapisać projekt
12. Naciśnij F9 aby skompilować i uruchomić aplikację
13. Zamknij aplikację kiedy jesteś gotów kontynuować pracę.

## **Pisanie obsługi zdarzeń**

Większość komponentów na palecie Component ma zdarzenia, a większość komponentów ma domyślne zdarzenia. Najbardziej popularnym zdarzeniem jest OnClick, które zostaje wywołane, kiedy komponent, taki jak Tbutton zostaje wywołany. Jeśli zaznaczysz komponent na formatce i klikniesz zakładkę Events Object Inspector, zobaczysz listę zdarzeń komponentu

## **Pisanie obsługi zdarzenia polecenia Update Now!**

Najpierw napiszemy obsługę zdarzenia polecenia Update Now! I przycisku:

1. Kliknij dwukrotnie komponent ActionList aby wyświetlić edytor Action List
2. Wybierz (No Category) aby zobaczyć Action1 i Action2
3. Kliknij dwukrotnie Action1. W edytorze Code pojawi się szkieletowa obsługa zdarzenia:

```
procedure TForm1.Action1Execute(Sender: TObject);
begin
```

```
end;
```

Umieść kursor między begin a end i wpisz:

```
if ClientDataSet1.State in [dsEdit, dsInsert] then
    ClientDataSet1.Post; ClientDataSet1.ApplyUpdates(-1);
```

Ta obsługa zdarzenia najpierw sprawdza stan bazy danych. Kiedy po przeniesieniu zmienił się rekord, jest on automatycznie wysyłany. Ale jeśli nie przeniosłeś zmienionego rekordu, baza danych pozostaje w trybie edycji lub wstawiania. Instrukcja if wysyła dane które mogą być zmieniane ale nie przekazuje do zbioru danych klienta. Kolejna instrukcja dotyczy aktualizacji posiadanych w zbiorze danych klienta bazy danych. Zmiany nie są automatycznie wysyłane do bazy danych kiedy używamy dbExpress. Musimy wywołać metodę ApplyUpdates do pisania wszystkich aktualizacji, wstawiania i usuwania rekordów ze zbioru danych klienta bazy danych

### **Pisanie obsługi zdarzenia polecenia Exit**

Teraz napiszemy obsługę zdarzenia dla polecenia Exit:

1. Kliknij dwukrotnie komponent Action List aby wyświetlić edytor Action List jeśli już nie jest wyświetlony.
2. Kliknij (No Category) aby zobaczyć Action2
3. Kliknij dwukrotnie Action2. Wyświetli się edytor Code z szkieletową obsługą zdarzenia

```
procedure TForm1.Action2Execute(Sender: TObject);
begin
```

```
end;
```

Umieść kursor między begin a end i wpisz:

```
Close;
```

Ta obsługa zdarzenia zamknie aplikację , kiedy zostanie użyte polecenie File | Exit

4. Zamknij edytor Action List
5. Wybierz File | Save All aby zapisać projekt

### **Pisanie obsługi zdarzenia FormClose**

W koncu napiszemy inne zdarzenie obsługi, które jest wywoływane kiedy zamykana jest aplikacja. Aplikacja może być zamknięta przez użycie File | Exit lub kliknięcie X w prawym górnym rogu. Tak czy inaczej, program sprawdza, aby upewnić się, że nie ma zbliżającej się aktualizacji bazy danych i wyświetla okno komunikatu z pytaniem co zrobić, jeśli zmiany są w toku. Możesz umieścić ten kod w obsłudze zdarzenia Exit ale toczące się zmiany w bazie danych będą utracone jeśli użytkownik wybierze zamknięcie aplikacji używając X

1. Kliknij obszar siatki głównej formatki aby wybrać samą formatkę
2. Wybierz zakładkę Events w Object Inspectorze aby zobaczyć zdarzenia samej formatki
3. Kliknij dwukrotnie OnClose (lub wpisz FormClose obok zdarzenia OnClose i kliknij go) .Szkielet obsługi zdarzenia FormClose jest zapisywany i wyświetlany w edytorze Code po innych obsługach zdarzeń:

```
procedure TForm1.FormClose(Sender: TObject; var Action:
    TCloseAction);
begin
```

```
end;
```

Umieść kursor między begin i end, i wpisz:

```
Action := caFree;
if ClientDataSet1.State in [dsEdit, dsInsert] then
    ClientDataSet1.Post;
if ClientDataSet1.ChangeCount > 0 then
begin
    Option := Application.MessageBox('Zmiany w bazie
    danych w toku. Czy chcesz zapisać bazę danych',
    [smbYes, smbNo, smbCancel],
    smsWarning, smbYes);
    case Option of
        smbYes: ClientDataSet1.ApplyUpdates(-1);
        smbCancel: Action := caNone;
    end;
end;
```

Ta obsługa zdarzenia sprawdza stan bazy danych. Jeśli zmiany są dokonywane, są one wysyłane do zbioru danych klienta gdzie zmiany są zliczane i powiększane. Potem, przed zamknięciem aplikacji, pole komunikatu jest wyświetlane pytając cię jak obsłużyć zmiany. Opcje odpowiedzi to Yes, No lub Cancel. Odpowiedź Yes skutkuje aktualizacją bazy danych; nie zamyka aplikację bez zmian w bazie danych; a Cancel anuluje zamykanie ale nie anuluje zmian dokonywanych w bazie danych i pozostawia uruchomioną.

4. Musisz zadeklarować zmienną używaną wewnątrz procedury. W lini między procedure a begin wpisz:

```
var  
    Option: TmessageButton;
```

5. Cała procedura wygląda tak:

```
    procedure TForm1.FormClose(Sender: TObject; var Action:  
        TCloseAction);  
var  
    Option: TMessageButton;  
begin  
Action := caFree;  
    if ClientDataSet1.State in [dsEdit, dsInsert] then  
        ClientDataSet1.Post;  
    if ClientDataSet1.ChangeCount > 0 then  
    begin  
        Option := Application.MessageBox('Zmiany w bazie  
        danych w toku. Czy chcesz zapisać bazę danych?',  
        'Pending Updates',[smbYes, smbNo, smbCancel],  
        smsWarning, smbYes);  
    case Option of  
        smbYes: ClientDataSet1.ApplyUpdates(-1);  
        smbCancel: Action := caNone;  
        end;  
    end;  
end;
```

6. Kończąc, wybierz File | Save All aby zapisać zmiany w projekcie. Naciśnij F9 aby uruchomić aplikację

To wszystko! Możesz wypróbować tą aplikację aby zobaczyć jak działa. Kiedy chcesz zamknąć program, możesz użyć w pełni funkcjonalnego polecenia File | Exit