

CLARION 5

Pierwsze kroki

Dane o oryginale:

COPYRIGHT 1994, 1995, 1996, 1997, 1998 by TopSpeed Corporation

All rights reserved.

This publication is protected by copyright and all rights are reserved by TopSpeed Corporation. It may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from TopSpeed Corporation.

This publication supports Clarion 5. It is possible that it may contain technical or typographic errors. TopSpeed Corporation provides this publication „as is”, without warranty of any kind, either expressed or implied.

TopSpeed Corporation

150 East Sample Road
Pompano Beach, Florida 33064
(954) 785-4555

Authorized translation from English Language Edition.

Translation © Speed
Poznań, 2000

Speed

ul. Ratajczaka 18
61-815 Poznań

Trademarks Acknowledgements:

TopSpeed® is a registered trademark of TopSpeed Corporation.

Clarion 5™ is a trademark of TopSpeed Corporation.

Btrieve® is a registered trademark of Pervasive Software

Microsoft® Windows® and Visual Basic® are registered trademarks of Microsoft Corporation

Wise Installation System is registered trademark of GLBS, Inc.

All other products and company names are trademarks of their respective owners

SPIS TREŚCI

SPIS TREŚCI	3
1 - WPROWADZENIE.....	5
WITAJ.....	5
Poziomy użytkownika/projektanta	5
Co znajdziesz w tej książce	7
Gdzie szukać więcej informacji	7
KONWENCJE PRZYJĘTE W OPISACH	10
Style czcionki	10
Style klawiszy.....	10
INFORMACJE O PRODUKCIE	11
Rejestrowanie produktu.....	11
Wsparcie techniczne.....	11
PROGRAM INSTALACYJNY.....	11
Wymagania systemowe.....	12
Uruchamianie instalacji.....	12
Opcje instalacji	13
Uruchamianie Clariona.....	13
2 - SZYBKI START	14
QUICK START WIZARD.....	15
Tworzenie słownika danych i aplikacji.....	15
QUICK LOAD WIZARD	22
Tworzenie nowego pliku	22
Definiowanie relacji	25
KREATOR PROCEDUR.....	28
Zastosowanie kreatora procedury Browse	30
NA KONIEC, KREATOR APLIKACJI	36
Korzystanie z kreatora aplikacji.....	37
OK, co udało nam się zrobić?	40
Co robić teraz?.....	41

3 - PRZEGLĄD ŚRODOWISKA.....	42
TAO CLARIONA	42
Clarion to język programowania	42
Poziomy abstrakcji	42
Programowanie w oparciu o szablony	43
Poziomy abstrakcji Clariona	45
Klucz do maksymalnej wydajności.....	46
CLARION, ŚRODOWISKO WYTWARZANIA APLIKACJI.....	48
Enterprise Tools	51
Co dalej?.....	52

1 - WPROWADZENIE

Witaj

Dziękujemy za zakup Clariona! Masz teraz w ręku narzędzie wytwarzania aplikacji o takim bogactwie możliwości i takiej elastyczności, z jakimi nigdy wcześniej nie udało Ci się spotkać.

Poziomy użytkownika/projektanta

Clarion jest doskonale adaptowalnym środowiskiem, stworzonym do oferowania szybkich rozwiązań na każdym poziomie abstrakcji: od małego biznesu do obsługi wielkich systemów korporacyjnych. Clarion może operować na wielu różnych poziomach i na każdym z nich wspiera zarówno użytkownika, jak i projektanta aplikacji. Na każdym z tych poziomów posiadasz pełną kontrolę nad danymi firmy, przy doskonałym współczynniku kosztu do efektywności i trzy razy szybciej, niż oferują to inne narzędzia. A oto dlaczego...

Narzędzie automatycznego tworzenia aplikacji oparte na szablonach

Jeśli jest Ci potrzebna prosta aplikacja, obsługująca bazę danych, możesz ją stworzyć, za pomocą Clariona, dosłownie w ciągu kilku minut. Kluczem do tego jest słownik danych. Gdy Generator Aplikacji wie, jakie pliki lub tabele są potrzebne w Twojej aplikacji oraz jak są wzajemnie powiązane, może tą aplikację wygenerować. Wszystko co musisz zrobić, to wskazać te pliki oraz określić ich wzajemne relacje. W rozdziale **Szybki start** pokazany został przykład takiego zastosowania, możesz samodzielnie przekonać się, jakie to proste.

Kreator aplikacji może utworzyć w pełni funkcjonalną aplikację działającą w konwencji *widok-formularz*, która jest domyślną konwencją przyjmowaną w Clarionie. Wynika ona prosto ze struktury Twojej bazy danych. Aplikacja działa, w przybliżeniu, następująco: (1) Użytkownik porusza się po bazie danych – całej lub określonej jej części, po pojedynczym pliku danych lub wielu, wzajemnie powiązanych plikach danych. Odbywa się to przez przeglądanie rekordów w okienkach listowych, widokach, nazywanych tutaj okienkami typu Browse. (2) użytkownika zaznacza wybrany rekord w liście i przeprowadza jakąś akcję, na przykład aktualizacji tego rekordu. Odbywa to się na ogół w oddzielnym oknie, w którym pojawiają się pola wybranego rekordu. Okno to jest formularzem (okienko typu Form). Za pomocą formularza możemy także wprowadzać nowe dane. (3) Opcjonalnie, użytkownik może pobierać wartości dla pól z innych plików danych (powiązanych na przykład relacją). Powoduje to otwarcie oddzielnego okienka typu Browse, w którym zaznaczamy ten rekord, którego pole chcemy pobrać i umieścić jego wartość w polu formularza.

Tak wygląda najbardziej ogólny opis. Dodatkowo, każde okno widoku może być sterowane za pomocą przycisków paska narzędzi aplikacji, może być wielokrotnie otwierane (w oddzielnych wątkach) i posiadać własny bufor rekordu wskazujący na aktualnie wybrany rekord. Możesz wybierać różne porządki sortowania rekordów w oknie, poprzez zwykłe wybranie jednej z reprezentujących klucze zakładek. Podwójne kliknięcie rekordu powoduje otwarcie okienka formularza i umożliwia aktualizację jego pól. Odbywa się przy tym kontrola, czy dany rekord nie jest już aktualizowany przez innego użytkownika (praca w sieci) oraz kontrola integralności danych.

Każdy może to zrobić. Wystarczy zacząć od wskazania pliku danych.

Środowisko programowania wizualnego

W Clarionie, umieszczając kontrolkę w okienku, uzyskujesz znacznie więcej, niż w przypadku innych narzędzi błyskawicznego wytwarzania aplikacji (RAD – Rapid Application Development). One również pozwalają na umieszczanie kontrolek w okienkach, oczekują jednak od programisty kodu implementującego ich działanie. W Clarionie, tak naprawdę wstawiasz *szablon*, który zawiera nie tylko kontrolkę, ale także wszystkie niezbędne dane i kod. Oznacza to, że nie musisz oprogramowywać działania kontrolek samodzielnie, jedno kliknięcie myszką powoduje dostarczenie kompletnego rozwiązania. Co więcej, każdy szablon posiada swój własny interfejs użytkownika. Gdy sprawdzisz właściwości szablonu, znajdziesz tam zakładkę „Actions”. Zaznaczając pole wyboru, wybierając element z listy lub wypełniając pole wprowadzania, możesz dostosować sposób działania szablonu dokładnie do własnych wymagań. Możesz określić akcje dla szablonów w wielu różnych miejscach, przekonasz się o tym studiując ten podręcznik. Opis **Generatora Aplikacji** znajdziesz natomiast w podręczniku *Learning Clarion*.

Gdy określisz już działanie szablonów w oparciu o ich interfejsy, Generator Aplikacji wygeneruje na tej podstawie kod aplikacji. Kod ten jest kodem źródłowym języka Clarion, jest w pełni obiektowy, bazuje bowiem na bibliotece *Application Builder Class (ABC) Library*, dzięki czemu jest zwarty i efektywny. Używając szablonów unikasz wszystkich niedogodności związanych z pisaniem kodu źródłowego programu wiersz po wierszu.

Zakres zastosowań języka Clarion jest bardzo szeroki. Opis języka zajmuje ponad 900 stron. Jedną z największych zalet korzystania z systemu szablonów wzbogaconą możliwością ręcznego kodowania fragmentów aplikacji jest to, że dzięki temu łatwo poznajemy sam język programowania i to w takim zakresie, na ile jest nam to potrzebne. Jeśli nie odpowiada Ci sposób rozwiązania problemu przez szablon, możesz go najpierw użyć do wygenerowania szkieletu procedury, którą możesz następnie zmodyfikować w oparciu o interfejs szablonu. Teraz pozostaje już tylko wygenerowanie kodu źródłowego i wykorzystanie go, po wprowadzeniu pożądanых zmian, we własnym rozwiązaniu. (Uwaga: istnieją szablony niezależnych producentów, być może one zapewnią takie działanie, jakie Ci właśnie odpowiada).

Nawet użytkownicy, którzy zwykle nie piszą programów, mogą teraz to zrobić!

Kompletny język programowania

Pracując na najbardziej podstawowym poziomie możesz zapisać aplikację w kodzie źródłowym języka Clarion – języka czwartej generacji. Clarion operuje na wyższym poziomie abstrakcji, dzięki czemu jest „czytelny” i pozwala na łatwe operowanie na rekordach i ich zbiorach. Nie musisz znać języka Clarion,

by utworzyć aplikację, jednak jego znajomość pozwoli Ci na lepsze rozumienie tego, w jaki sposób aplikacja działa, a przez to na tworzenie lepszych aplikacji. Wprowadzenie do pisania kodu źródłowego języka Clarion znajdziesz w podręczniku *Learning Clarion*.

Profesjonaliści wysoko cenią możliwości języka Clarion. Został on przygotowany z myślą o twórcach aplikacji biznesowych. Biorąc pod uwagę łatwość jego opanowania (jest to w końcu język czwartej generacji, a nie język programowania niskopoziomowego), wkrótce osiągniesz ogromną wydajność w swojej pracy. Dodatkowo, kompilator TopSpeed zamieni Twój kod, niezależnie od tego, czy napiszesz go ręcznie, czy wygenerujesz za pomocą Generатора Aplikacji, w optymalny kod maszynowy.

Nie zależnie od tego, na jakim poziomie zamierzasz pracować, najlepiej zrobisz, jeśli zaczniesz od dokładnego zapoznania się z rozdziałami tego podręcznika.

Co znajdziesz w tej książce

Podręcznik Pierwsze kroki stanowi pierwsze wprowadzenie do środowiska Clarion. Poniżej zostały wymienione poszczególne jego części i krótkie opisy ich zawartości.

Wprowadzenie	<i>Rozdział pierwszy</i> ; wprowadza nas w zagadnienia związane z Clarionem i udostępnia informacje o sposobie instalacji tego środowiska.
Szybki start	<i>Rozdział drugi</i> ; za pomocą kilku kroków kreatora Quick Start Wizard utworzysz kompletną aplikację w pięć minut. Poświęć mniej niż godzinę na lekturę tego rozdziału i zapoznaj się z działaniem kreatora aplikacji i kreatorów procedur. Utworzysz za ich pomocą prostą aplikację, a następnie rozbudujesz ją samodzielnie, tak by zarządzała dwoma wzajemnie powiązаныmi plikami danych. Wszystko to bez pisania nawet jednej linii kodu.
Przegląd środowiska	<i>Rozdział trzeci</i> ; przegląd koncepcji tworzenia aplikacji stosowanych w Clarionie. Opisane są w nim także wszystkie narzędzia środowiska.

Gdzie szukać więcej informacji

Pierwszym miejscem, w którym powinno się szukać dodatkowych informacji jest system pomocy i jego sekcja **How do I ...?**. Jej tematy są odpowiedzią na najczęstsze pytania stawiane przez tych, którzy stykają się z Clarionem po raz pierwszy. Sekcja ta jest dostępna po wciśnięciu przycisku **How do I ...?** znajdującego się w spisie treści systemu pomocy.

System pomocy Clariona jest bardzo rozbudowany. Hipertekstowa pomoc pojawia się zawsze po naciśnięciu klawisza **F1** lub wybraniu polecenia **Help** w systemie menu.

Oprócz tego masz do dyspozycji szereg podręczników dostarczanych wraz z Clarionem:

Pierwsze kroki (Getting Started)

Tę książkę właśnie czytasz.

Learning Clarion

Ta książka składa się z dwóch poradników prowadzących czytelnika krok-po-kroku. Pierwszy dotyczy Generators Aplikacji i jest rozbudowanym wprowadzeniem do wszystkich dostępnych w środowisku Clarion narzędzi. Drugi z kolei jest wprowadzeniem do programowania w języku Clarion.

Application Handbook

Podręcznik dotyczący wszystkich narzędzi wykorzystywanych w procesie tworzenia aplikacji (z wersją Standard jest dostarczany w formacie .PDF na dysku CD). Opisuje szablony dostarczane standardowo wraz z produktem, w pełni dokumentuje bibliotekę Application Builder Class (ABC) Library wykorzystywaną przy generowaniu kodu obiektowego, dostarcza dokładne informacje na temat wykorzystywanych przez Clarion sterowników formatów danych.

Language Reference

Kompletny przewodnik po języku Clarion. Książka ta zawiera dokładne opisy wszystkich instrukcji języka, z przykładami dla każdej z nich. Jest podzielona tematycznie. Pełny jej tekst jest także zawarty w systemie pomocy. Podczas pracy w Edytorze tekstu możesz umieścić kursor na dowolnej instrukcji języka i wcisnąć klawisz **F1**, by uzyskać jej opis.

User's Guide

Opis środowiska Clarion zorientowany na wykonywane w nim zadania i zaaranżowany wg podstawowych jego składników (z wersjami Standard i Professional jest dostarczany w formacie .PDF na dysku CD).

Programmer's Guide

Kolekcja rozdziałów szczegółowo opisujących różnorodne aspekty programowania w języku Clarion, a także sposoby dostosowania środowiska do indywidualnych potrzeb (dostępny tylko w wersjach Professional i Enterprise; z wersją Professional jest dostarczany w formacie .PDF na dysku CD). Zawiera również kompletny podręcznik języka szablonów, z opisami wszystkich instrukcji i funkcji, przykładami dla każdej z nich i przejrzystym wyjaśnieniem zagadnień związanych z pisaniem własnych szablonów.

ReportWriter User's Guide

Dokumentacja ReportWriter for Windows – samodzielnego narzędzia, za pomocą którego końcowy użytkownik aplikacji może tworzyć i drukować własne raporty (dołączona do wersji Professional i Enterprise; w przypadku wersji Professional w formacie .PDF na dysku CD – dostępna także jako oddzielny produkt).

Enterprise Tools

Dokumentacja narzędzi takich, jak Documents the Data Dictionary Synchronizer, Version Control/Team Developer, and Data Modeller tools. Znajduje się tam również opis Business Math Library

(standardowe funkcje biznesowe i statystyczne) oraz Wise for Clarion (zestawu narzędzi służących do tworzenia programów instalacyjnych). Ten podręcznik występuje tylko w Enterprise Edition (niektóre wymienione produkty są również sprzedawane oddzielnie).

Wizatron Handbook

Dokumentuje Wizatrony – nowe kreatory w technologii opracowanej specjalnie dla Clariona, które potrafią „uczyć się” wymagań programisty i stają się jego bardzo samodzielnym asystentem. Ta książka jest dostarczana tylko wraz Enterprise Edition.

Master Index

Kompletny indeks do wszystkich podręczników i to zarówno tych drukowanych, jak i dostępnych w formacie .PDF (w formie drukowanej – z wersją Enterprise – w przypadku pozostałych wersji w formacie .PDF na dysku CD).

Wszystkie podręczniki zapisane w formacie .PDF na dysku CD można przeglądać za pomocą aplikacji Adobe Acrobat Reader, którą również można zainstalować z dysku CD Clariona. Pliki te umożliwiają wyszukiwanie tekstu, jest również możliwe ich wydrukowanie na drukarce.

Ważne: Jeśli jakieś informacje znajdujące się w pliku pomocy wykluczają się wzajemnie z informacjami zawartymi w dokumentacji, należy polegać na tych, które pochodzą z pliku pomocy.

TopSpeed Corporation stara się uczynić wszystko, by zapewnić aktualność drukowanej dokumentacji. Jednakże aktualizacje zawsze łatwiej, szybciej i taniej wprowadza się w systemie pomocy, stąd mogą występować pewne różnice.

Konwencje przyjęte w opisach

Style czcionki

<i>Kursywa</i>	Tekst, który należy wpisać z klawiatury.
Kapitaliki	Klawisze, które należy wcisnąć na klawiaturze.
Pogrubienie	Polecenia lub opcje, które należy wybrać w systemie menu; teksty okienek dialogowych.

Style klawiszy

F1	Pojedynczy klawisz, który należy nacisnąć.
Alt+x	Kombinacja klawiszy, które należy jednocześnie nacisnąć. W tym wypadku: wciskamy i przytrzymujemy klawisz alt i wciskamy klawisz x . Później zwalniamy oba klawisze.

Informacje o produkcie

Rejestrowanie produktu

Zanim zaczniesz korzystać z Clariona, poświęć chwilę czasu na wypełnienie i wysłanie karty rejestracyjnej. Pozwoli Ci to na skorzystanie z kilku istotnych udogodnień. Po zarejestrowaniu, możesz korzystać z usług wsparcia użytkowników świadczonych przez TopSpeed, automatycznie będziesz również otrzymywać informacje o nowych produktach i aktualizacjach istniejących.

Wsparcie techniczne

Możesz otrzymać nielimitowane, darmowe wsparcie techniczne za pośrednictwem serwisów uruchomionych na forum CompuServe i w Internecie. Pracownicy TopSpeed, jak również wolontariusze będący certyfikowanymi partnerami wsparcia TopSpeed (TopSpeed Certified Support Partners), znani także jako *Team TopSpeed*, odpowiedzą na Twoje pytania. Dodatkowo możesz otrzymywać porady i odpowiedzi na pytania od innych użytkowników Clariona. Zalecamy wszystkim naszym klientom korzystanie z takich usług wymienionych poniżej.

- ◆ Dostęp do wsparcia technicznego TopSpeed na CompuServe uzyskujemy poprzez GO TOPSPEED.
- ◆ W Internecie korzystamy z grupy dyskusyjnej *comp.lang.clarion*.
- ◆ W Internecie możemy uzyskać dostęp do *topspeed.products family* poprzez grupy dyskusyjne funkcjonujące na serwerze *tsnews.clarion.com*.
- ◆ W Internecie możemy również odwiedzić stronę firmy TopSpeed znajdującą się pod adresem <http://www.topspeed.com>.

Istnieje możliwość skorzystania z płatnych usług wsparcia za pośrednictwem łączy telefonicznych TopSpeed Corporation (954) 785-4556. Zadzwoń do biura obsługi klienta (800) 354-5444 lub (954) 785-4555 w celu uzyskania szczegółowych informacji na temat programów wsparcia oferowanych przez TopSpeed.

Program instalacyjny

Program instalacyjny dekompresuje i kopiuje pliki Clariona na dysk twardy.

- ◆ Dla każdego z obsługiwanych systemów operacyjnych, program instalacyjny dostarcza opcje umożliwiające instalowanie wybranych komponentów środowiska, takich jak na przykład programy przykładowe.
- ◆ W odniesieniu do każdej wersji Windows, program instalacyjny pozostawia użytkownikowi decyzję, czy ma automatycznie zmodyfikować systemową ścieżkę dostępu (parametr PATH w pliku AUTOEXEC.BAT), tak by dołączyć do niej katalog zawierający pliki środowiska Clarion.

- ◆ W przypadku Windows 3.x, program instalacyjny pozwala użytkownikowi na podjęcie decyzji, czy ma być utworzona grupa Menedżera programów z ikonami środowiska Clarion, Debuggera, plików pomocy i plików opisów (ReadMe).
- ◆ W Windows 95 i Windows NT program instalacyjny pozwala na utworzenie ikon reprezentujących elementy środowiska Clarion w menu Start.

Wymagania systemowe

Środowisko Clarion może być uruchamiane na dowolnym komputerze spełniającym minimalne wymagania stawiane przez Windows 3.x, Windows 95 lub Windows NT. Poniżej zostały wymienione minimalne konfiguracje:

- ◆ Windows 3.x, 8 MB RAM.
- ◆ Windows 95, 16 MB RAM.
- ◆ Windows NT, 32 MB RAM.
- ◆ Przynajmniej 12-35 MB wolnego miejsca na dysku twardym; w zależności od wybranej opcji instalacji.

Aplikacja opracowana za pomocą Clariona może być uruchamiana na komputerach spełniających minimalne wymagania stawiane przez działający na nich system operacyjny.

Uruchamianie instalacji

By uruchomić instalację Clariona w Windows 3.x:

1. Wprowadź dysk instalacyjny do napędu CD-ROM.
2. Z poziomu Menedżera programów, Menedżera plików lub innej nakładki umożliwiającej uruchamianie programów, wybierz polecenie **Plik ▸ Uruchom**.
3. Wpisz *D:\SETUP* w okienku dialogowym **Uruchom** (*D:* oznacza literę Twojego napędu CD-ROM) i wciśnij przycisk **OK**. W efekcie zostanie wyświetlone okienko powitalne programu instalacyjnego i informacje o dalszych czynnościach.

By uruchomić instalację Clariona w Windows 95:

1. Wprowadź dysk instalacyjny do napędu CD-ROM.
2. Z menu Start wybierz polecenie **Ustawienia ▸ Panel sterowania**.
3. Wybierz opcję **Dodaj/Usuń Programy** i wciśnij przycisk **Instaluj**. Pozostałe kroki wykonaj w oparciu o instrukcje kreatora instalacji Windows 95.

Opcje instalacji

Po uruchomieniu programu instalacyjnego zobaczysz ekran z różnymi opcjami:

1. Wybierz opcję instalacji proponowaną przez kreator instalacji (np. docelowy dysk i katalog), wciskaj przycisk **Next**, by przechodzić do kolejnych opcji. Program instalacyjny umieszcza główne składniki środowiska Clarion w podkatalogu o nazwie BIN; jest to oczywiście podkatalog tego katalogu, który został wskazany jako docelowy. *Wszystkie* pliki są umieszczane we wskazanym katalogu docelowym i jego podkatalogach. *Żaden* plik nie znajdzie się w innym katalogu. Podczas instalacji jest wyświetlany wskaźnik wykonania operacji kopiowania plików.
2. Wybierz **Yes** lub **No** w momencie, gdy program instalacyjny zapyta Cię o modyfikowanie zmiennej PATH określającej systemową ścieżkę dostępu. W przypadku Windows 3.x i Windows 95 środowisko Clariona wymaga, by ścieżka dostępu do katalogu BIN była wymieniona w systemowej ścieżce dostępu reprezentowanej przez parametr PATH w pliku AUTOEXEC.BAT. Jeśli wybierzesz **No**, *musisz* później samodzielnie zmodyfikować plik AUTOEXEC.BAT. Inną zmianą wprowadzaną w plikach systemowych jest dodanie sekcji Clariona; po pierwszym jego uruchomieniu, w pliku WIN.INI.
3. Wybierz **Yes** lub **No** w momencie, gdy program instalacyjny zapyta Cię o instalowanie ikon reprezentujących poszczególne elementy środowiska w grupie Menedżera programów lub w menu Start.
4. Wybierz **Yes** lub **No** w momencie, gdy program instalacyjny zapyta Cię o wyświetlenie zawartości pliku ReadMe. Jeśli nie chcesz zapoznać się z zawartością tego pliku w tej chwili, możesz zrobić to później, ikonę reprezentującą ten plik znajdziesz w grupie Menedżera programów lub w menu Start; o ile oczywiście została wybrana opcja powodująca utworzenie tej grupy. Zalecamy przeczytanie pliku ReadMe w czasie, gdy program instalacyjny będzie kopiował pliki na twardy dysk.
5. Wciśnij przycisk **OK**, gdy zakończy się proces instalacji.

Uruchamianie Clariona

By uruchomić Clarion, znajdź jego ikonę w grupie i podwójnie kliknij ją lewym przyciskiem myszki; w menu Start wystarczy ją tylko kliknąć.

Pojawi się wówczas okno środowiska Clarion, które jest w tym momencie gotowe do działania.

- ◆ Krótki opis poszczególnych składników środowiska Clarion – takich, jak na przykład paski narzędzi - znajdziesz w rozdziale trzecim.
- ◆ Przejdź teraz do rozdziału drugiego, znajdziesz tam opis tworzenia *dwóch* imponujących aplikacji – od słownika danych, poprzez okna przeglądania, aż do formularzy wprowadzania i modyfikowania danych – wszystko w czasie *krótszym niż godzina*.

2 - SZYBKI START

Clarion jest narzędziem zorientowanym na dane. Biznes opiera się na danych. Oznacza to, że Clarion jest zoptymalizowany pod kątem tworzenia aplikacji biznesowych. Wszystko w Clarionie zaczyna się od słownika Data Dictionary.

Za pomocą Clariona możesz stworzyć słownik danych i działającą aplikację bez żadnego kodowania. Wszystko, co musisz zrobić, to zdefiniować pola w pliku danych i uruchomić kreatora Quick Start Wizard. Kreator ten utworzy kompletną aplikację Windows i to w czasie krótszym niż pięć minut, wszystko zależy od tego, jak szybko potrafisz pisać na klawiaturze. Twoja aplikacja będzie posiadała formularz umożliwiający aktualizację danych w pliku, okienko do przeglądania tych danych w oparciu o różne klucze sortowania oraz raporty dla każdego ze zdefiniowanych kluczy.

Posługując się kreatorem Quick Start Wizard definiujemy jedynie każde pole pliku danych określając jego nazwę, format wyświetlania i informacje o kluczu. Quick Start Wizard zajmie się całą resztą.

W tym rozdziale:

- ◆ Wykorzystanie kreatora **Quick Start Wizard** do utworzenia bazy klientów i aplikacji umożliwiającej zarządzanie danymi w tej bazie; kompilowanie i uruchamianie aplikacji.
- ◆ Wykorzystanie kreatora **Quick Load** – narzędzia umożliwiającego w łatwy i szybki sposób definiowanie plików w słowniku danych.
- ◆ Definiowanie relacji między plikami danych w słowniku danych **Dictionary Editor**, określanie zasad zachowania integralności danych.
- ◆ Wykorzystanie generatora aplikacji; wzbogacanie funkcji aplikacji za pomocą kreatorów procedur **Procedure Wizards**.
- ◆ Na zakończenie, wykorzystanie kreatora aplikacji **Application Wizard** to utworzenia drugiej, kompletnej aplikacji obsługującej relacyjną bazę danych w oparciu o ten sam słownik danych.

To wszystko zrobisz w jakieś pół godziny – bez żadnego kodowania z Twojej strony. Po zapoznaniu się z tym rozdziałem, będziesz mieć na swoim koncie dwie kompletne aplikacje obsługujące bazę danych z dwoma powiązаныmi relacją plikami.

A więc, do roboty!

Quick Start Wizard

Tworzenie słownika danych i aplikacji

Punkt startowy

Masz przed sobą otwarte środowisko Clarion. Musisz widzieć okienko dialogowe Pick.

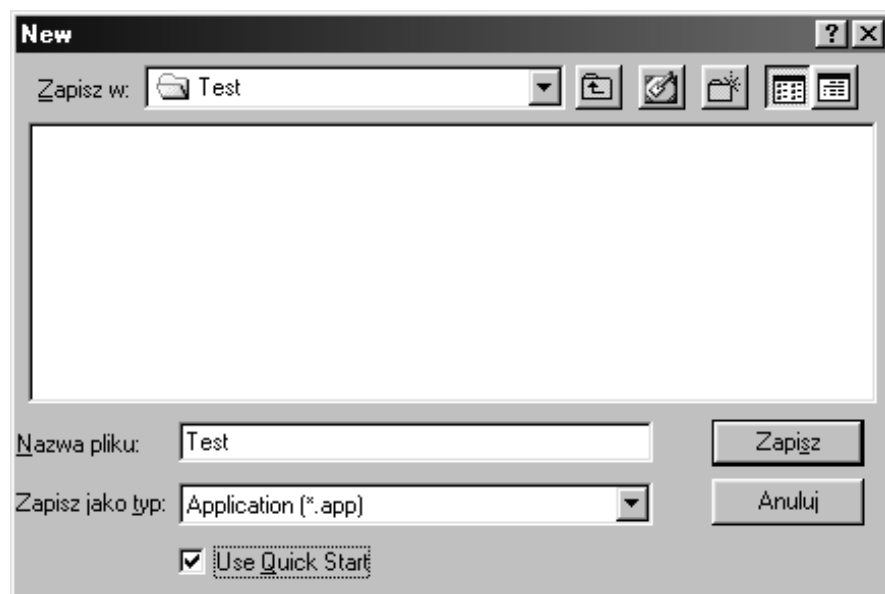
Zacniemy od utworzenia katalogu C:\CLARION5\TEST, w którym będziemy przechowywali tworzone przez siebie, w oparciu o wskazówki zawarte w tej książce, aplikacje. Zakładamy oczywiście, że Clarion został zainstalowany w katalogu C:\CLARION5.

Tworzenie katalogu roboczego

1. Utwórz, na przykład za pomocą Menedżera plików lub Eksploratora Windows, nowy podkatalog o nazwie TEST w katalogu CLARION5.
2. Powróć do Clariona

Tworzenie pierwszej aplikacji w Clarionie

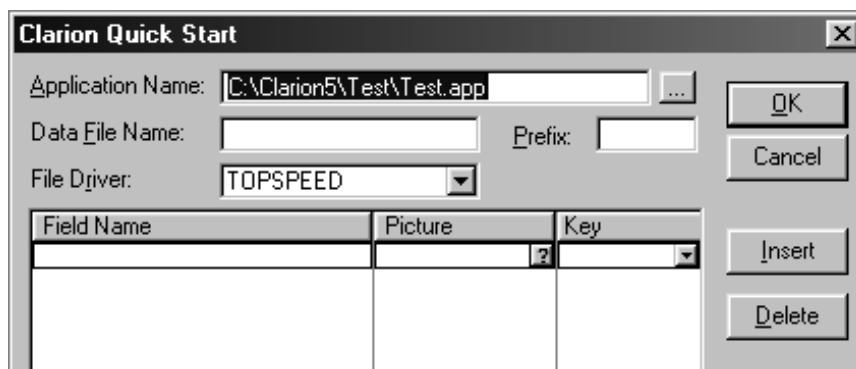
1. W okienku dialogowym **Pick** wciśnij przycisk **New**. Pojawi się wówczas okienko dialogowe **New**. Jest to standardowe okienko Windows umożliwiające utworzenie nowego pliku. Za jego pomocą można wskazać katalog i nazwę nowo tworzonego pliku.



2. Wybierz *Application (*.app)* w polu **Zapisz jako typ**. Przejdź do katalogu C:\CLARION5\TEST.
3. Wpisz nazwę aplikacji – *TEST* - w polu **Nazwa pliku**. Quick Start Wizard wykorzysta wpisaną nazwę zarówno dla pliku aplikacji, jak i dla pliku słownika danych. W efekcie powstaną dwa pliki: *TEST.DCT* (plik słownika danych) i *TEST.APP* (plik aplikacji). W systemach Windows 95 i Windows NT możesz stosować długie nazwy plików.

- Zaznacz pole **Use Quick Start** i wciśnij przycisk **Zapisz**.

Pojawi się teraz okienko dialogowe **Clarion Quick Start**.



Definiowanie pliku danych

- Wciśnij dwa razy klawisz **Tab**, wpisz *Klienci* w polu **Data File Name** i wciśnij ponownie klawisz **Tab**.

Quick Start Wizard wykorzysta wpisaną nazwę, jako nazwę pliku danych. Zauważ, że jej pierwsze trzy litery – *KLI* – pojawiają się w polu **Prefix**. Prefiks pliku służy do tego, by nazwy pól w nim występujące były unikalne. Dzięki temu, w różnych plikach, o różnych prefiksach, możemy stosować te same nazwy dla pól.

- Wciśnij **Tab**, by zaakceptować prefiks *KLI*.

Teraz wybierzemy format pliku danych. Domyślną zawartością pola **File Driver** jest **TOPSPEED** (jeden z dwóch systemów plików Clariona).

- Wciśnij **Tab**, by zaakceptować **TOPSPEED** jako przyjęty format pliku danych.

W tym momencie znajdziesz się w liście, w której będziesz definiować pola.

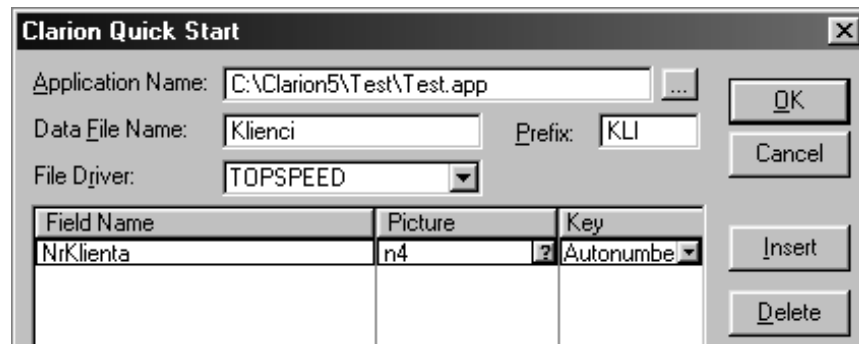
UWAGA: W Windows 3.1 musisz uruchomić program SHARE.EXE lub też załadować sterownik VSHARE.386, tylko wtedy jest możliwe stosowanie plików w formacie TopSpeed. Nowsze wersje Windows zrobią to za Ciebie.

- Wpisz *NrKlienta* w pierwszym wierszu kolumny **Field Name** i wciśnij **Tab**.

Spowoduje to utworzenie pola *NrKlienta*. Quick Start Wizard wykorzysta tę nazwę również jako opis poprzedzający pole (Prompt) w okienku, w którym je umieścimy oraz jako nagłówek kolumny w liście (Column Heading) wyświetlanej w oknie lub w raporcie.

- W kolumnie **Picture** wpisujemy *N4* i wciskamy klawisz **Tab**.

Określamy w ten sposób domyślny wzorzec wyświetlania zawartości pola w oknach i raportach, definiując jednocześnie typ pola.



6. Znajdując się w kolumnie **Key** wciskamy klawisz strzałki w dół, spowoduje to wyświetlenie dostępnych opcji. Podświetlamy *Autonumber* i wciskamy klawisz **Tab**.

Taki wybór spowoduje, że numer umieszczany w polu będzie nadawany automatycznie i będzie to numer unikalny dla każdego rekordu. Określając klucz jako *Autonumber* możemy być pewni, że każdemu klientowi zostanie przyporządkowany unikalny numer, automatycznie inkrementowany dla każdego dopisywanego rekordu. Quick Start Wizard tworzy okienka przeglądania i raporty dla każdego utworzonego klucza. Ten klucz pozwala na przeglądanie danych klientów posortowanych wg numeru reprezentowanego przez *NrKlienta*.

Kursor został automatycznie umieszczony w następnym wierszu, możemy zatem zdefiniować kolejne pola.

7. W kolumnie **Field Name** wpisujemy *Nazwa* i wciskamy klawisz **Tab**.

Pole *Nazwa* będzie reprezentowało nazwę klienta.

8. W kolumnie **Picture** wpisujemy *S20* i wciskamy klawisz **tab**.

Określamy w ten sposób domyślny wzorec wyświetlania zawartości pola w okienkach i raportach. W tym przypadku będzie to pole *STRING* mogące liczyć 20 dowolnych znaków.

9. W kolumnie **Key** wciskamy klawisz strzałki w dół, by wyświetlić dostępne opcje, wybieramy *Duplicate* i wciskamy klawisz **tab**.

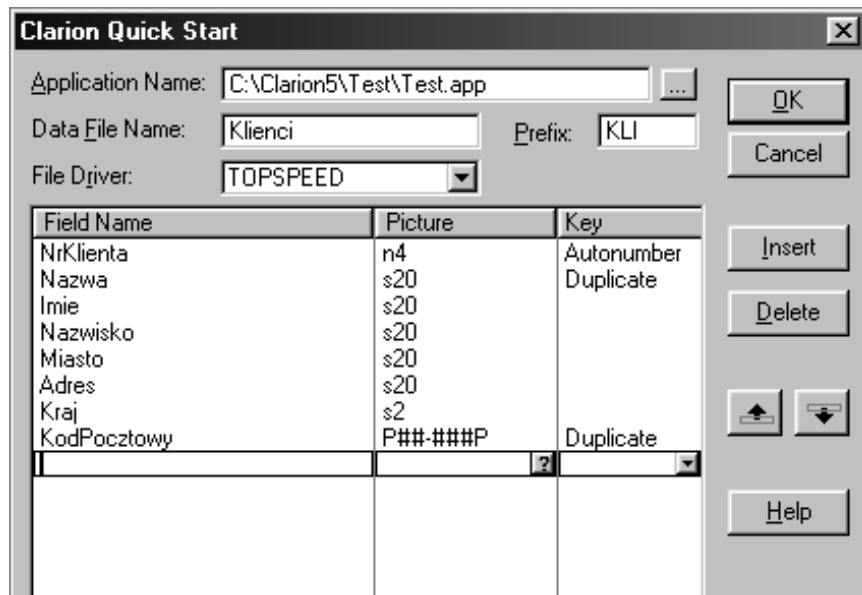
Taki klucz będzie sortował klientów wg nazwy dodatkowo dopuszczając, by nazwy te się powtarzały.

10. Samodzielnie zakończ teraz definiowanie pliku danych, powtarzając opisane powyżej czynności, tak, by uzyskać strukturę przedstawioną w tabeli poniżej.

Field Name	Picture	Key
<i>NrKlienta</i>	N4	Autonumber
<i>Nazwa</i>	S20	Duplicate
<i>Imie</i>	S20	(no key)
<i>Nazwisko</i>	S20	(no key)

<i>Miasto</i>	S20	(no key)
<i>Adres</i>	S20	(no key)
<i>Kraj</i>	S2	(no key)
<i>KodPocztowy</i>	P##-###P	Duplicate

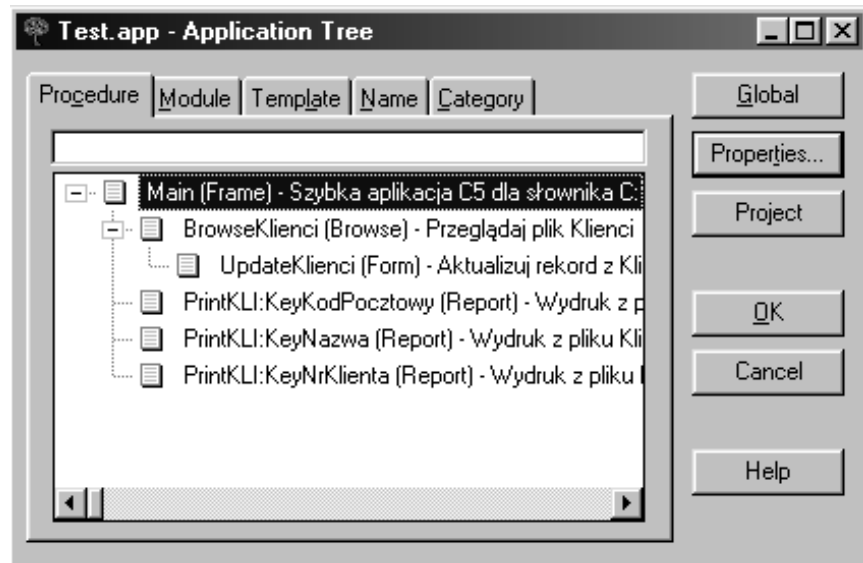
Zwróć uwagę na wzorec przyjęty dla pola *KodPocztowy*. Jest to tzw. maska wprowadzania; w tym wypadku umożliwia wprowadzanie w polu tylko cyfr, pierwsze dwie są oddzielone od pozostałych znakiem myślnika.



Zakończ działanie Quick Start Wizard i utwórz aplikację

1. Po zdefiniowaniu wszystkich pól wciśnij przycisk **OK**.

Quick Start Wizard zapyta Cię najpierw, czy na pewno wszystkie pola zostały już określone. Po wciśnięciu w wyświetlonym okienku dialogowym przycisku **OK**, zostanie utworzona aplikacja, a na ekranie, w okienku **Application Tree**, pojawi się jej struktura.



Przyjrzyj się strukturze aplikacji. Na samym szczycie znajduje się procedura *Main* typu Frame. Procedura ta steruje aplikacją, jest to tzw. ramka aplikacji MDI (Multiple Document Interface), w której oknie można otwierać wiele okien podrzędnych. Procedura ta zawiera system menu z poleceniami umożliwiającymi wywoływanie innych procedur. Poniżej procedury *Main* znajduje się procedura *BrowseKlienci* (Browse) oraz podłączona do niej procedura *UpdateKlienci* (Form). Procedura typu Browse służy do wyświetlania danych w oparciu o wszystkie zdefiniowane dla danego pliku klucze. W strukturze aplikacji widać także procedury wyświetlające raporty dla każdego z tych kluczy.

Aplikacja jest kompletna i gotowa do działania, nie jest wymagana żadna Twoja ingerencja w jej kod.

Przyciski paska narzędzi, od lewej: Pick, New, Open, Save, Make, Run i Debug



- Wybierz polecenie **Project ▸ Run** (lub wciśnij przycisk *Run* na pasku narzędzi).

Generator Aplikacji wygeneruje kod źródłowy naszej aplikacji, który następnie zostanie skompilowany, skonsolidowany i uruchomiony.

Gratulacje! Twoja pierwsza aplikacja już działa.

Testowanie aplikacji

Zacniemy, na początek, od dodania kilku rekordów.

- Wybierz polecenie **Browse ▸ Browse the Klienci File**.
- Wciśnij przycisk **Dopisz** w pasku nawigacyjnym.

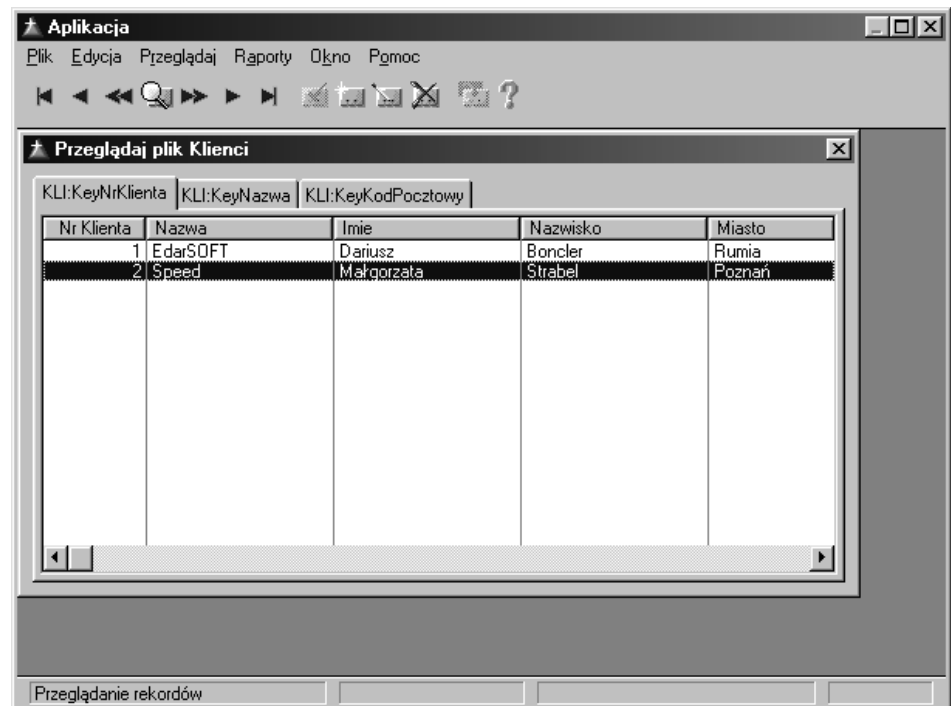
Pasek nawigacyjny, od lewej przyciski: Pierwszy Rekord, Poprzednia Strona, Poprzedni Rekord, Szukaj, Następny Rekord, Następna Strona, Ostatni Rekord, Wybierz, Dopisz, Popraw, Usuń, Przywróć, Pomoc



3. Po wpisaniu danych pierwszego klienta, wciskamy przycisk **OK**, by zamknąć okno formularza i zapisać wprowadzone dane na dysku.

Przykład ten ilustruje konwencję *Browse-Form*, przyjętą domyślnie w Clarionie. Okienko Browse wyświetla rekordy pliku danych, z kolei okienko Form jest używane do aktualizacji (dopisywania, poprawiania i usuwania) wybranego rekordu. W Clarionie można również zastosować konwencję *ScrollingForm*, pozwala ona na aktualizację rekordów bezpośrednio w liście, w której są przeglądane (to zagadnienie omówimy później).

4. Dodaj teraz kilka rekordów (możesz skorzystać, oprócz sposobów przedstawionych powyżej, z menu podręcznego – kliknij tylko okienko przeglądania prawym przyciskiem myszki i wybierz polecenie **Insert**). Zauważ, że za każdym razem, gdy dodajesz nowy rekord, automatycznie zwiększa się wartość pola *NrKlienta*. Sprawdź wyświetlanie rekordów wg różnych porządków sortowania (zakładki reprezentują poszczególne klucze), powiększ okienko, spróbuj zmienić rozmiary poszczególnych kolumn. Możesz także wyświetlić wybrane raporty; poprzez wskazanie odpowiedniego polecenia w menu **Browse**.



5. Wybierz polecenie **File ▶ Exit**, by zakończyć działanie aplikacji.

Możesz teraz uruchomić swój program, TEST.EXE, spoza środowiska Clarion, na przykład z poziomu Eksploratora Windows. Upewnij się, że

dysponujesz kompletną, gotową do działania, aplikacją Windows. Teraz zapiszemy efekty naszej pracy i opuścimy Generator Aplikacji, by zmodyfikować nasz słownik danych. Zdefiniujemy w nim nowy plik danych, który następnie powiążemy relacją z już istniejącym.

6. Wybierz polecenie **File** ▶ **Close** lub wciśnij przycisk **OK**, by zamknąć Generator Aplikacji.

Quick Load Wizard

Nowy plik w słowniku danych możemy w łatwy i szybki sposób utworzyć za pomocą specjalnego kreatora – Quick Load Wizard. Wymaga to użycia opcji Quick Load w Edytorze słownika danych. Działanie kreatora Quick Load Wizard jest podobne do działania kreatora Quick Start Wizard – definiujemy pola (nazwy, wzorce wyświetlania, klucze), a w efekcie, w słowniku danych, pojawia się definicja pliku danych. Omawiana opcja jest dostępna za każdym razem, gdy dodajemy nowy plik do słownika danych.

Tworzenie nowego pliku

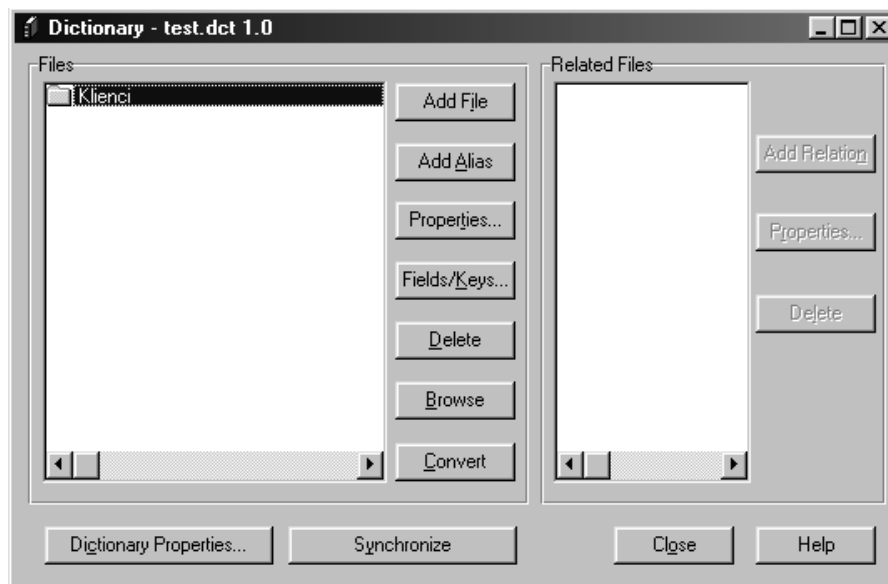
W tym punkcie dodamy do słownika danych plik, w którym będziemy przechowywali numery telefoniczne klientów. Zdefiniujemy go tak, by można było zapisywać wiele różnych telefonów dla jednego klienta (utworzymy zatem relację jeden-do-wielu).

Zastosowanie Quick Load w Edytorze słownika

1. Wybierz polecenie **File ▶ Open** (lub wciśnij przycisk *Open* w pasku narzędzi).

Na ekranie pojawi się okienko dialogowe **Open**.

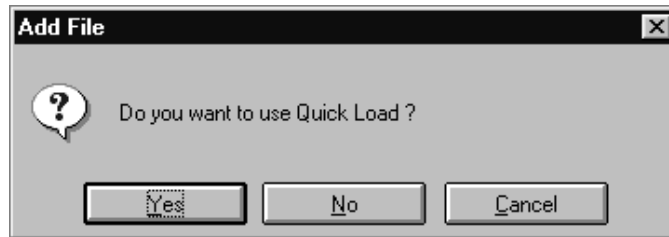
2. Wybierz opcję *Dictionary (*.dct)* w polu **Pliki typu** i podświetl w okienku plik *TEST.DCT*, na zakończenie wciśnij przycisk **Otwórz**.



Na ekranie pojawi się okienko słownika danych **Dictionary Editor**.

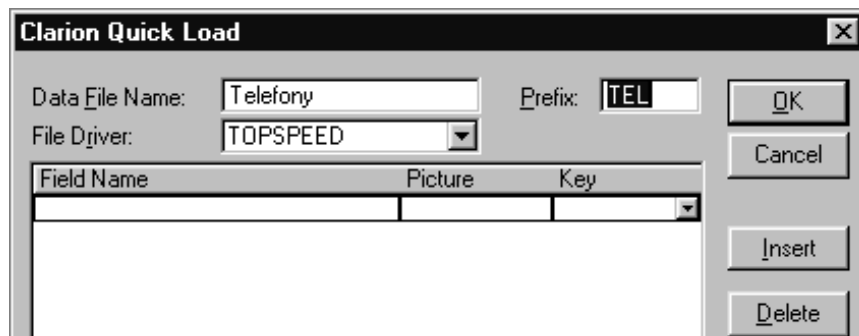
3. Wciśnij teraz przycisk **Add File**.

Spowoduje to wyświetlenie okienka dialogowego umożliwiającego zastosowanie opcji Quik Load.



4. Wciśnij przycisk **Yes**.

Spowoduje to wyświetlenie okienka dialogowego **Clarion Quick Load**. Zauważ, że jest ono bardzo podobne do okienka Quick Start Wizard.



5. Przejdź do pola **Data File Name** i wpisz w nim nazwę tworzonego pliku – *Telefony*, następnie wciśnij klawisz **Tab**.
6. Wciśnij ponownie klawisz **Tab**, by zaakceptować prefiks *TEL*.

Teraz przystąpimy, w polu **File Driver**, do wskazania sterownika dla naszego pliku danych.

7. Wciśnij klawisz **Tab**, by zaakceptować sterownik **TOPSPEED**.

W tym momencie znajdziesz się w liście, w której definiuje się pola rekordu.

8. W pierwszym wierszu wpisz *NrKlienta* i wciśnij klawisz **Tab**.

W ten sposób utworzysz pole o nazwie *NrKlienta*. Pole to zostanie później przez nas powiązane z polem *NrKlienta* w pliku *Klienci*. Używając takich samych nazw pól ułatwimy sobie rozpoznanie, które pola mają uczestniczyć w relacji. W kodzie wygenerowanym przez Clarion pola te będą się różniły prefiksem, po którym dodatkowo wystąpi znak dwukropka, i dzięki temu będą unikalne. Tak więc, tak naprawdę, nasze pole nazywa się *TEL:NrKlienta*, podczas gdy pole z pliku *Klienci* – *KLI:NrKlienta*.

9. Wpisz *N4* w kolumnie **Picture** i wciśnij klawisz **Tab**.
10. Przejdź do kolumny **Key** i wciśnij klawisz strzałki w dół, z wyświetlonych opcji wybierz *Duplicate* i wciśnij klawisz **Tab**.

W efekcie w kluczu będą mogły być powtarzane wartości danego pola. W ten sposób umożliwiamy przydzielanie wielu numerów telefonów pojedynczemu klientowi. Przygotowujemy tym samym grunt do zdefiniowania relacji jeden-do-wielu (*jeden Klient może mieć wiele*

Telefonów).

Kursor pojawi się w następnym wierszu umożliwiając zdefiniowanie kolejnego pola.

11. W kolumnie **Field Name** wprowadź nazwę *Kierunek*, następnie wciśnij klawisz **Tab**.
12. W kolumnie **Picture** wprowadź wzorzec dla numeru kierunkowego, niech to będzie *P(#-##)P*, następnie dwukrotnie wciśnij klawisz **Tab**.

Nasz numer kierunkowy będzie się składał z trzech cyfr, pierwsza będzie oddzielona od pozostałych znakiem myślnika, dodatkowo cały numer będzie zamknięty w nawiasach okrągłych.

13. Wprowadź pozostałe pola, zgodnie z poniższą tabelką:

Field Name	Picture	Key
<i>NrKlienta</i>	N4	Duplicate
<i>Kierunek</i>	P(#-##)P	(no key)
<i>NrTelefonu</i>	P###-##-##P	(no key)
<i>Opis</i>	S20	(no key)

14. Po zdefiniowaniu wszystkich pól wciśnij przycisk **OK**.

Quick Load zapyta Cię, czy to na pewno już wszystkie pola.

15. Wciśnij przycisk **OK**.

Quick Load utworzy definicję nowego pliku danych i umieści go w słowniku danych.

Przygotowanie pliku do aktualizacji w okienku przeglądania

Ponieważ w naszym nowo utworzonym pliku występują tylko cztery pola, jest on dobrym materiałem do zilustrowania przykładu aktualizacji rekordów w okienku przeglądania (konwencja *ScrollingForm*).

1. Podświetl plik *Telefony* w liście **Files**, następnie wciśnij przycisk **Properties**.
2. Przejdź do zakładki **Options**.
3. W okienku **User Options** wpisz *EDITINPLACE*.

W ten sposób poinformujesz kreator aplikacji, że ma wygenerować kod umożliwiający bezpośrednio aktualizację pól rekordu w okienku przeglądania (Browse).



4. Wciśnij przycisk **OK**.

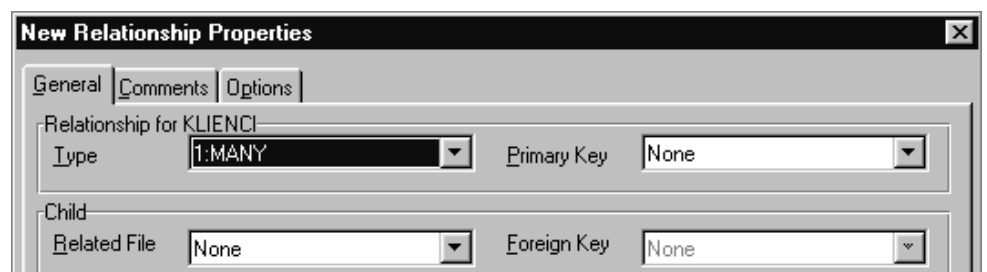
Definiowanie relacji

Oczywiście chcemy, żeby plik *Telefony* zawierał numery telefonów poszczególnych klientów. Znaczący to, że musi istnieć relacja pomiędzy tymi dwoma plikami. W naszym przypadku jeden klient może mieć wiele telefonów, czyli zachodzi relacja jeden-do-wielu. By zdefiniować taką relację, musimy powiązać wzajemnie oba pliki w słowniku danych. W ten sposób poinformujemy Generator Aplikacji o sposobie uzyskiwania dostępu do powiązanych ze sobą rekordów.

Ustawianie relacji między dwoma plikami

1. Podświetl plik *Klienci* w liście **Files**, następnie wciśnij przycisk **Add Relation**.

Na ekranie pojawi się okienko dialogowe **New Relationship Properties**. Za jego pomocą zdefiniujemy relację.



2. Upewnij się, że w polu **Type** jest wybrana opcja *1:MANY*.
3. Przejdź do pola **Primary Key** i wciśnij klawisz strzałki w dół, z wyświetlonych opcji wybierz *KeyNrKlienta* i wciśnij klawisz **Tab**.

Będzie to klucz pliku *Klienci*, który będzie wykorzystywany w definiowanej relacji.

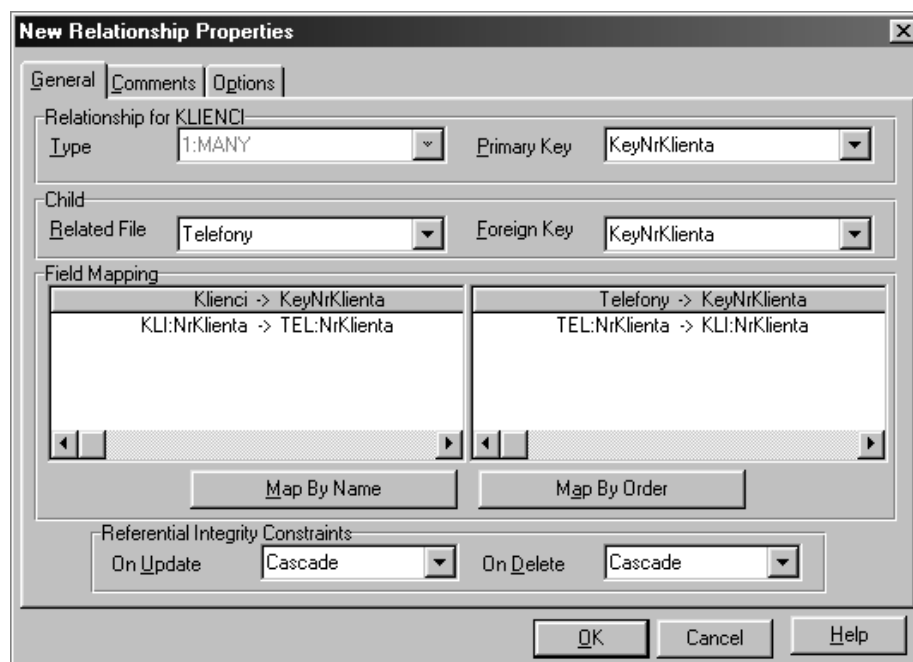
4. Przejdź do pola **Related File** i wciśnij klawisz strzałki w dół. Z listy wyświetlonych opcji wybierz nazwę pliku *Telefony*, następnie wciśnij klawisz **Tab**.
5. Przejdź teraz do kolejnego pola, **Foreign Key**, i wciśnij klawisz strzałki w dół. W wyświetlonej liście wskaż klucz *KeyNrKlienta*, następnie wciśnij klawisz **Tab**.

Ten klucz będzie uczestniczył w relacji od strony pliku *Telefony*.

Teraz należy dokonać mapowania pól kluczy uczestniczących w relacji, tzn. wskazać, które z tych pól są ze sobą powiązane. Jest to łatwe, między innymi dzięki temu, że użyliśmy identycznych nazw dla interesujących nas pól.

6. Wciśnij przycisk **Map By Name**.

Powiązane pola pojawiają się w okienku **Field Mapping**.



Określenie zasad zachowania integralności danych

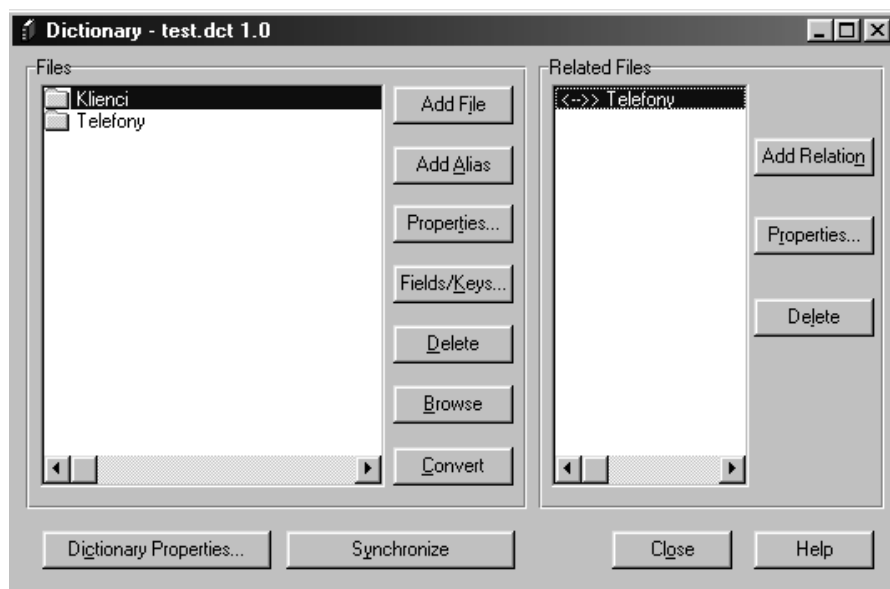
1. Wybierz opcję *Cascade* w polu **On Update** grupy **Referential Integrity Constraints**.
2. Wybierz opcję *Cascade* w polu **On Delete** grupy **Referential Integrity Constraints**.

Wygenerowany kod źródłowy będzie automatycznie przeprowadzał operacje zapewniające zachowanie integralności danych. W dużym skrócie: jeśli skasujemy rekord w pliku *Klienci*, to zostaną automatycznie skasowane wszystkie rekordy zawierające telefony usuniętego klienta. Jeśli poprawimy numer klienta, to automatycznie zostanie poprawiona wartość pola *NrKlienta* we wszystkich rekordach z telefonami danego klienta.

Więcej informacji na ten temat można znaleźć w podręcznikach *Learning Clarion* oraz *User's Guide* (rozdział *Using the Dictionary Editor*).

3. Wciśnij przycisk **OK**, by zamknąć okienko dialogowe **New Relationship Properties**.

Nasz słownik danych wygląda teraz tak:



4. Wybierz teraz polecenie **File** ▶ **Save** lub wciśnij przycisk *Save* na pasku narzędzi, by zapisać słownik danych.
5. Wybierz polecenie **File** ▶ **Close** lub wciśnij przycisk **Close** w okienku dialogowym **Dictionary**, by zamknąć Edytor słownika danych.

Kreator procedur

Oprócz kreatora Quick Start Wizard, Clarion daje nam do dyspozycji kreatora procedur pomagającego w tworzeniu okien przeglądania danych, formularzy, czy raportów. Kreator ten jest dostępny w momencie, gdy tworzymy nową procedurę w istniejącej już aplikacji.

Skorzystamy zatem z kreatora procedury typu Browse, by oprogramować przeglądanie zawartości pliku *Telefony*. Kreator ten utworzy również procedurę formularza, a to dlatego, że, jak pamiętamy, podczas definiowania pliku *Telefony*, wybraliśmy konwencję *ScrollingForm*, przy której aktualizacja rekordów ma miejsce w oknie przeglądania.

Uruchomienie Generатора Aplikacji

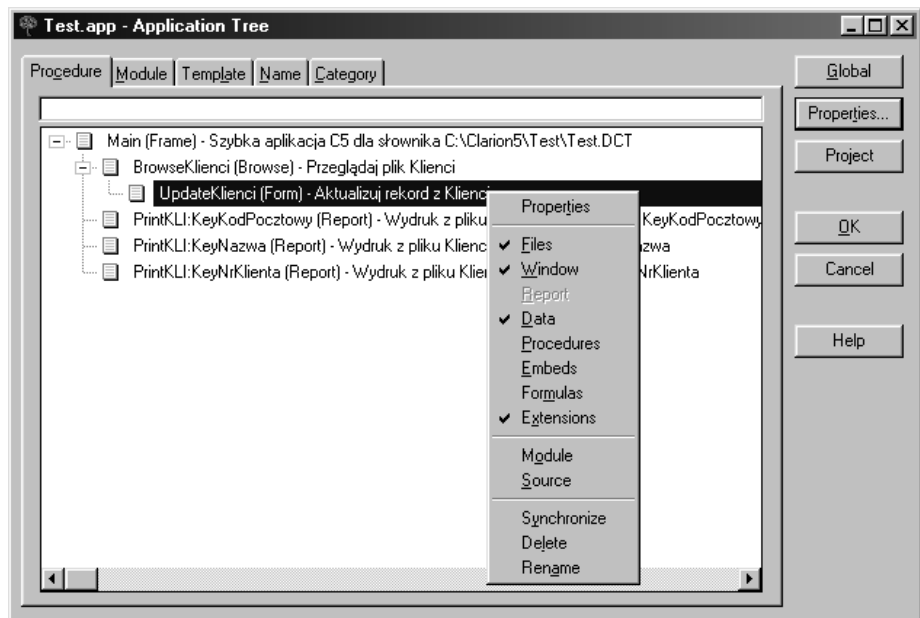
1. Wybierz polecenie **File** ▶ **Pick** lub wciśnij przycisk *Pick* w pasku narzędzi.
2. Przejdź do zakładki **Application** i podświetl *c:\clarion5\test\test.app*, następnie wciśnij przycisk **Select**.

Na ekranie pojawi się struktura aplikacji wyświetlona w okienku **Application Tree**.

Poprawianie procedury formularza

1. Podświetl w strukturze aplikację procedurę *UpdateKlienci* i kliknij ją prawym przyciskiem myszki; zostanie wówczas wyświetlone menu podręczne.

Menu to pozwala na dostęp do narzędzi Generатора Aplikacji, bez potrzeby otwierania okienka właściwości danej procedury.

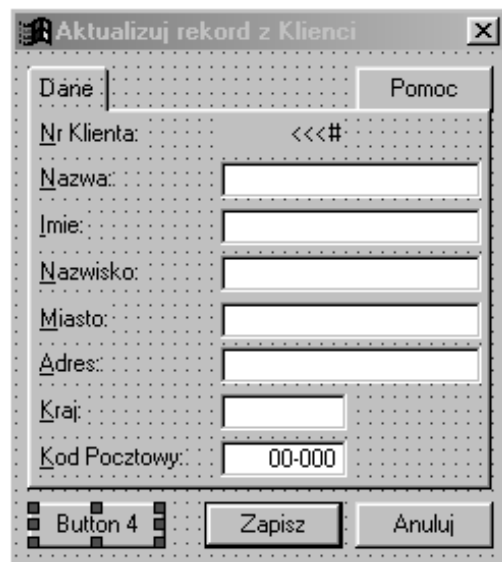


Obok nazw tych narzędzi, za pomocą których wprowadzono już jakieś dane, znajdują się symbole zaznaczenia.

2. Kliknij w menu podręcznym opcję **Window**.

Na ekranie pojawi się narzędzie umożliwiające wizualne projektowanie wyglądu okna procedury – Window Formatter.

3. Wybierz polecenie **Control ▶ Push Button** lub kliknij przycisk Button w okienku narzędzi (ikona tego przycisku wygląda tak, jak przycisk **OK**).
4. Kliknij obszar okienka w pobliżu lewego, dolnego rogu. We wskazanym przez Ciebie miejscu zostanie umieszczony nowy przycisk.

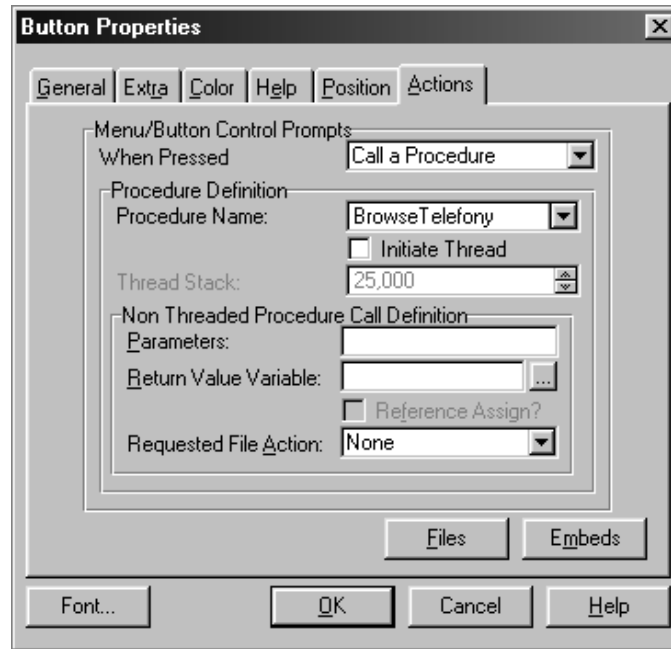


5. Kliknij nowy przycisk, tak by pojawiło się wokół niego obramowanie, następnie kliknij go prawym przyciskiem myszki i z wyświetlonego menu podręcznego wybierz polecenie **Properties**. W efekcie na ekranie pojawi się okienko dialogowe **Button Properties**.
6. W polu **Text** wpisz nową nazwę przycisku – *Telefony*. Nazwa ta będzie wyświetlana na przycisku.
7. Przejdź do zakładki **Actions**.

Zakładka ta umożliwia określenie działań wykonywanych w odpowiedzi na akcje użytkownika związane z przyciskiem, na przykład wciśnięcie go. W naszym przypadku, chcemy, by po wciśnięciu przycisku była wywoływana procedura wyświetlająca wszystkie rekordy pliku *Telefony* powiązane z aktualnie wyświetlanym rekordem pliku *Klienci*. Innymi słowy, by wyświetlała ona wszystkie telefony danego klienta.

8. W polu **When Pressed** wybierz opcję **Call a Procedure**. W ten sposób, po wciśnięciu przycisku, będzie wykonywana wskazana procedura.

W zakładce **Actions** pojawi się grupa pól umożliwiających zdefiniowanie tej procedury.



9. W polu **Procedure Name** wpisz nazwę nowo tworzonej procedury – *BrowseTelefony*, następnie wciśnij przycisk **OK**.

W ten sposób wskazaliśmy procedurę, która zostanie uruchomiona po wciśnięciu przycisku przez użytkownika.

10. Wybierz teraz polecenie **Exit!** z menu Window Formattera, by powrócić do okna struktury aplikacji. Wcześniej program zapyta nas, czy ma zapisać wprowadzone zmiany. Odpowiadamy twierdząco wciskając przycisk **Yes**.

Nowa procedura – *BrowseTelefony* pojawiła się w naszej strukturze aplikacji. Jest ona opisana jako (ToDo), co oznacza, że trzeba zdefiniować jej działanie.

Zastosowanie kreatora procedury Browse

Kreatory procedur potrafią wygenerować kompletny kod procedur i to bazując na minimalnej ilości informacji dostarczonych przez programistę. Wszystko odbywa się w kolejnych krokach precyzujących zakres czynności, które procedura ma wykonywać.

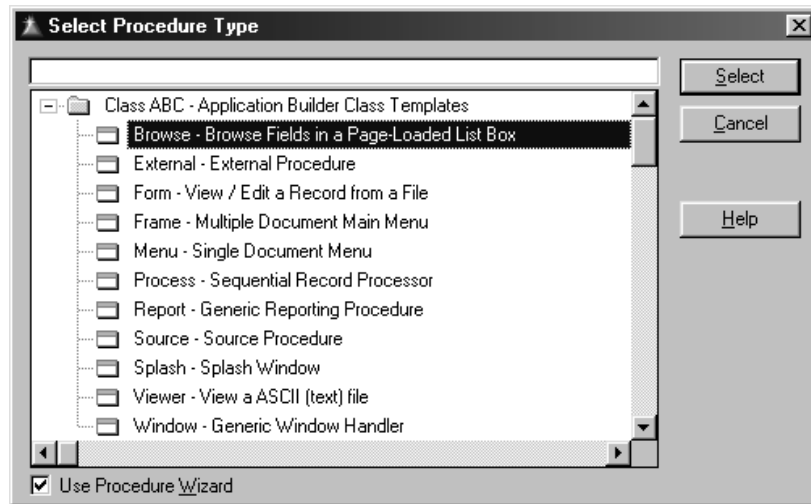
Wywołanie kreatora Browse Wizard

1. Kliknij podwójnie nazwę procedury *BrowseTelefony* (ToDo).

W efekcie pojawi się okienko **Select Procedure Type**. Umożliwia ono wybranie szablonu Generators Aplikacji, w oparciu o który ma zostać wygenerowany kod procedury.

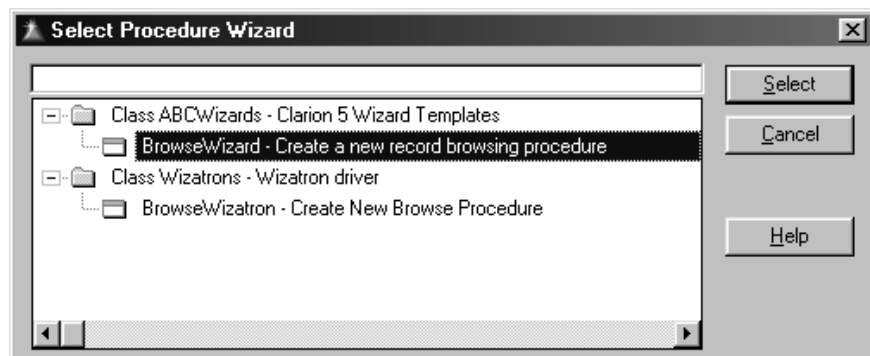
2. Podświetl szablon o nazwie **Browse**, następnie zaznacz pole **Use Procedure Wizard** znajdujące się w dolnej części okienka.

Zaznaczając pole **Use Procedure Wizard** powodujemy uruchomienie właściwego kreatora.



3. Wciśnij przycisk **Select**.

UWAGA: Jeśli korzystasz z Clarion 5 Professional Edition, przejdź od razu do punktu 5.

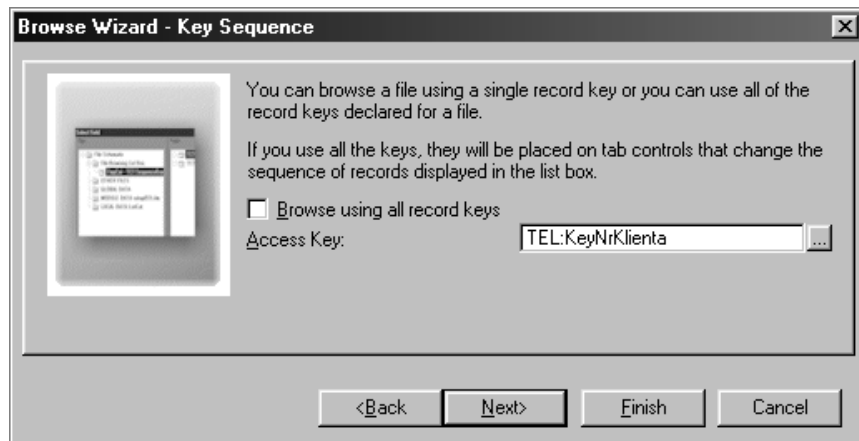


4. Podświetl BrowseWizard – Create a new recording procedure, następnie wciśnij przycisk **Select**.
5. Po przeczytaniu informacji podawanych przez kreator w jego pierwszym oknie, wciśnij przycisk **Next**, by zacząć tworzenie nowej procedury.



6. Wpisz w polu wprowadzania nazwę *Telefony* (możesz również wcisnąć przycisk ... i wybrać nazwę pliku z wyświetlonej wówczas listy). W ten

sposób określasz, który plik danych ma być przeglądany w okienku Browse.

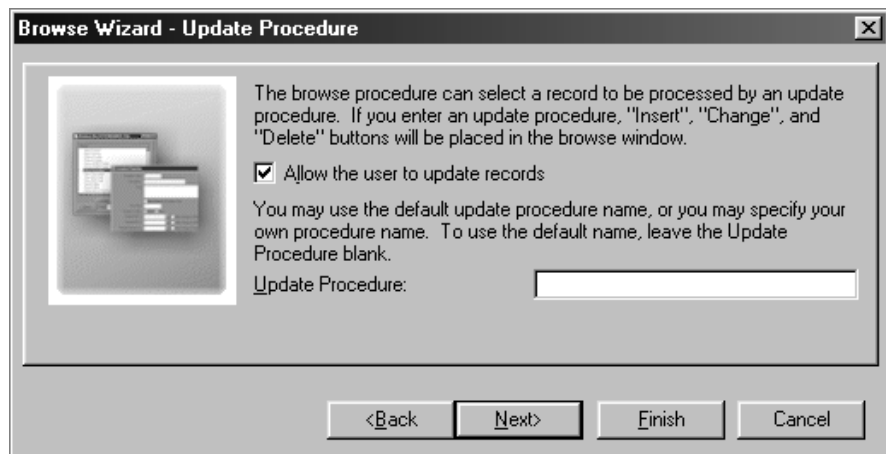


7. Usuń zaznaczenie pola **Browse using all record keys**, następnie wciśnij przycisk ... i wybierz z listy *TEL:KeyNrKlienta*, następnie wciśnij przycisk **Next**.

Określasz tym samym klucz, wg którego będą posortowane przeglądane rekordy.

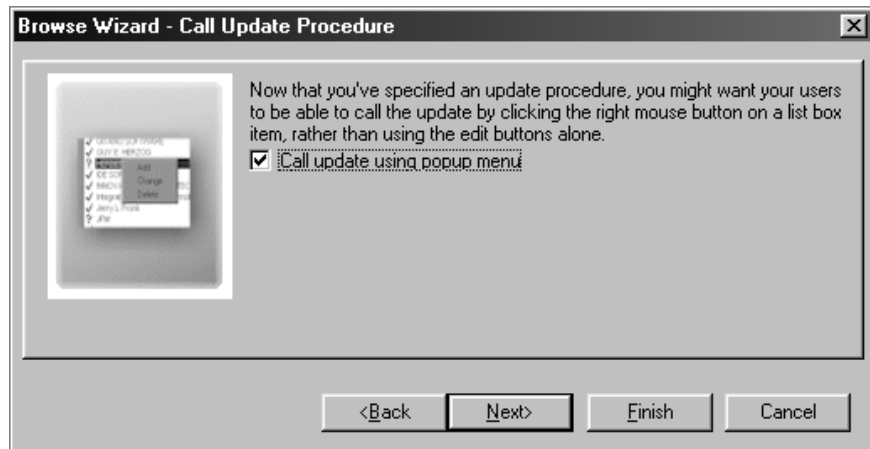
8. Pozostaw puste pole **Update Procedure** i po prostu wciśnij przycisk **Next**.

To pole służy do wprowadzenia nazwy procedury, którą utworzy kreator Browse Wizard, a która będzie służyła do aktualizacji rekordów w pliku *Telefony*. W naszym przypadku nie jest to konieczne, gdyż definiując plik w słowniku danych podaliśmy informację dla kreatora, by wygenerował taką procedurę, która będzie umożliwiała aktualizację rekordów bezpośrednio w oknie przeglądania. Zrobiliśmy to wprowadzając dyrektywę EDITINPLACE w zakładce User Options okienka, w którym definiowaliśmy plik. W przypadku, gdybyśmy tak nie zrobili, Browse Wizard automatycznie wywoła kreator formularza Form Wizard, który utworzy procedurę aktualizacji rekordów. Z kolei, wprowadzając nazwę procedury aktualizacji rekordów w omawianym polu, powodujemy, że nie będzie dla niej generowany kod. Możemy zdefiniować ją później, wywołując na przykład kreatora Form Wizard.

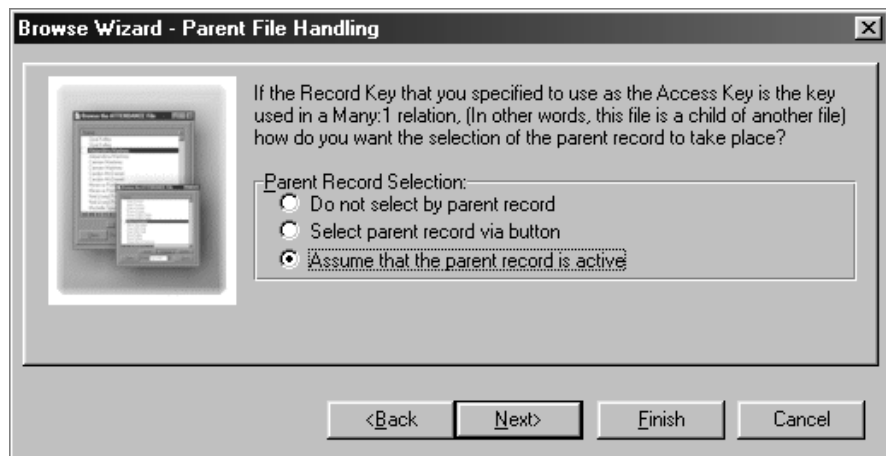


9. Zaznacz teraz pole **Call update using popup** menu i wciśnij przycisk **Next**.

Dzięki temu użytkownik aplikacji będzie mógł korzystać z podręcznego menu do wybierania poleceń aktualizacji rekordów (dopisywanie, poprawianie i usuwanie).

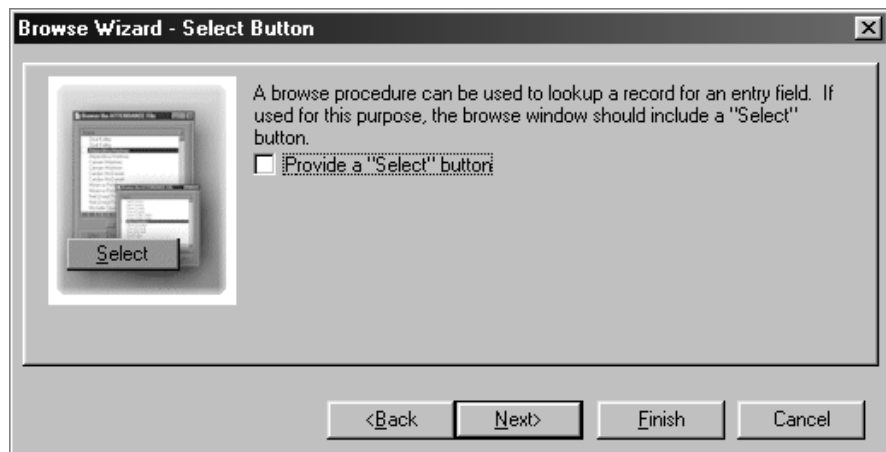


10. Usuń zaznaczenie pola **Provide buttons for child files**, następnie wciśnij przycisk **Next**.



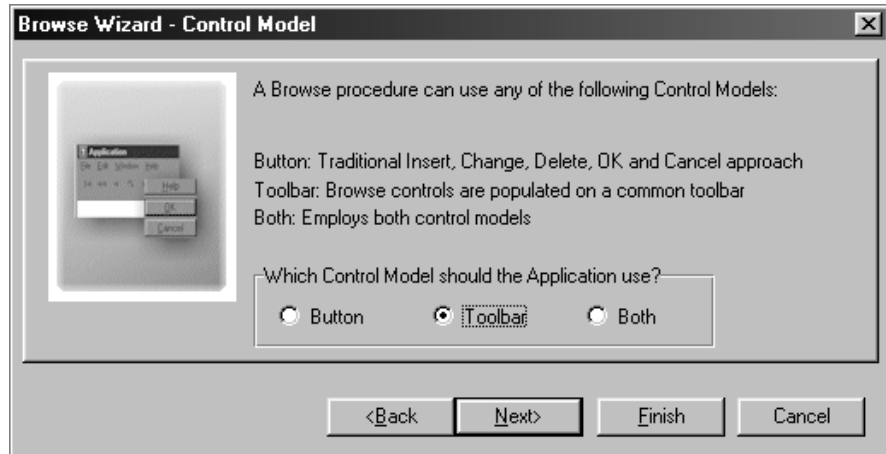
11. Zaznacz opcję **Assume that the parent record is active** i wciśnij przycisk **Next**.

W ten sposób spowodujemy, że w okienku tworzonej przez nas procedury będą wyświetlane tylko te rekordy pliku *Telefony*, które są powiązane z aktualnie wybranym, w momencie wywołania procedury, rekordem pliku *Klienci*.



12. Usuń zaznaczenie pola **Provide a “Select” button** i wciśnij przycisk **Next**.

Ponieważ nasza procedura nie będzie służyła do wybierania rekordów, przycisk **Select** (Wybierz) nie będzie nam potrzebny.

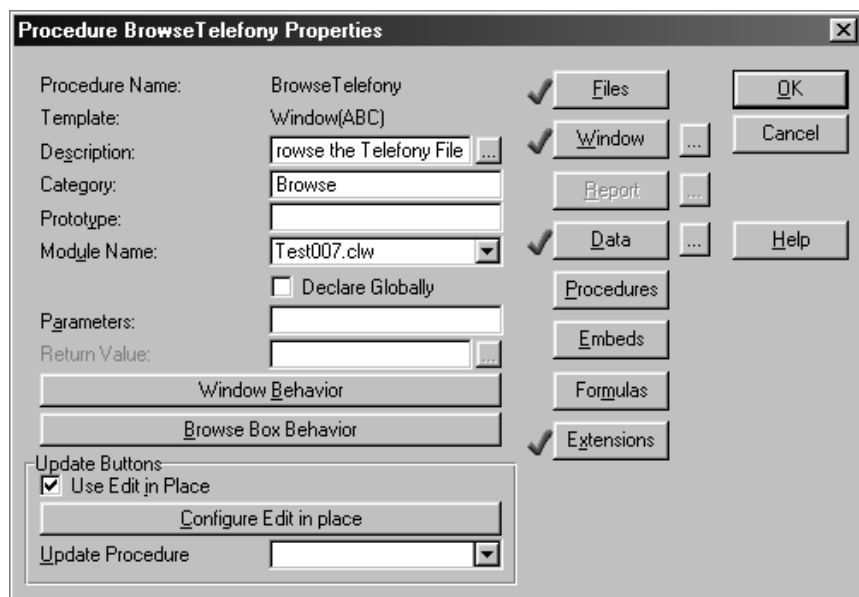


13. Pozostaw zaznaczoną opcję *Toolbar* w grupie **Which Control Model Should the Application Use** i wciśnij przycisk **Next**.

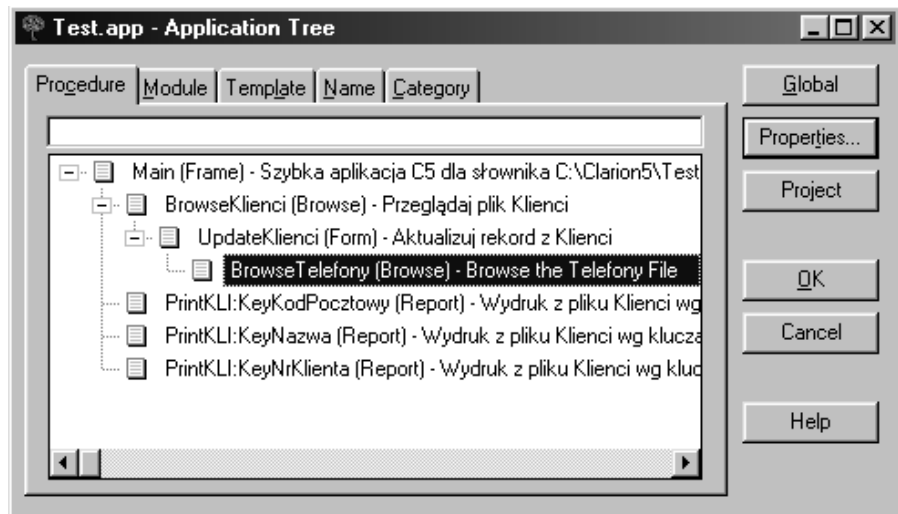
Taki wybór spowoduje, że wygląd okienka procedury *BrowseTelefony* będzie podobny do wyglądu okienka procedury *BrowseKlienci*. Do wywoływania określonych akcji, takich jak na przykład aktualizacja rekordu, będą wykorzystywane przyciski paska narzędzi okna aplikacji.

14. Wciśnij przycisk **Finish**, by wygenerować kod zdefiniowanej procedury.

Kreator przystąpi do generowania kodu. Jeśli nie określiliśmy dyrektywy *EDITINPLACE* w okienku *User Options* podczas definiowania pliku *Telefony*, kreator *Browse Wizard* utworzy również procedurę służącą do aktualizacji rekordów. Po zakończeniu działania kreatora, na ekranie pojawi się okienko właściwości zdefiniowanej procedury **Procedure BrowseTelefony Properties**.



15. Wciśnij przycisk **OK**, by zamknąć to okienko i powrócić do widoku drzewa aplikacji.



Zauważ, że w drzewie aplikacji pojawiła się nasza nowa procedura.

16. Wybierz polecenie **Project ▶ Run** (lub wciśnij przycisk *Run* w pasku narzędzi).

Po skompilowaniu, otrzymasz kompletną aplikację bazodanową umożliwiającą przeglądanie danych z pliku podstawowego oraz z pliku powiązanego relacją, wyposażoną dodatkowo w mechanizmy zapewniające zachowanie integralności danych, tak by w plikach nie pojawiały się „osierocone” rekordy.

Jeżeli chcesz dodać telefon dla wybranego klienta, wciśnij przycisk *Telefony*, po wcześniejszym otwarciu okienka, w którym dopisuje się lub poprawia dane klienta.

17. Gdy skończysz testowanie możliwości swojej aplikacji, wybierz w niej polecenie **File ▶ Exit**, by powrócić do środowiska Clarion.
18. Wybierz polecenie **File ▶ Close** lub wciśnij przycisk **OK**, by zamknąć Generator Aplikacji.

Na koniec, Kreator Aplikacji

Quick Start Wizard, a także kreatory Browse, Form i Report to narzędzia o dużych możliwościach. Jednak Clarion oferuje jeszcze jedno, które łączy w sobie wszystkie ich funkcje, a nawet więcej – Application Wizard, czyli kreator aplikacji.

Application Wizard buduje kompletną aplikację w oparciu o istniejący słownik danych. Słownik ten może zawierać definicje dowolnej ilości powiązanych wzajemnie plików. Pliki te powinny być w pełni zdefiniowane, włączając w to definicje powiązań i zasad zachowania integralności danych.

Aby ułatwić korzystanie z Application Wizard, w Edytorze słownika danych mamy możliwość określenia właściwości dla każdego pliku danych, ich pól i kluczy. W ten sposób możemy wpływać na wygląd i działanie wygenerowanych przez kreator aplikacji okienek.

Wykorzystanie Edytora słownika do określenia opcji dla kreatora

1. Wybierz polecenie **File ▶ Pick** (lub wciśnij przycisk *Pick* w pasku narzędzi).
2. Wybierz zakładkę **Dictionary**, podświetl *c:\clarion5\test\test.dct* i wciśnij przycisk **Select**.

Na ekranie pojawi się okienko dialogowe **Dictionary Editor**.

3. Podświetl w nim plik *Klienci*, następnie wciśnij przycisk **Fields/Keys...**
4. Przejdź do zakładki **Keys**, kliknij klucz *KeyNrKlienta* i wciśnij przycisk **Properties**.

Na ekranie zostanie wyświetlone okienko dialogowe **Edit Key Properties**.

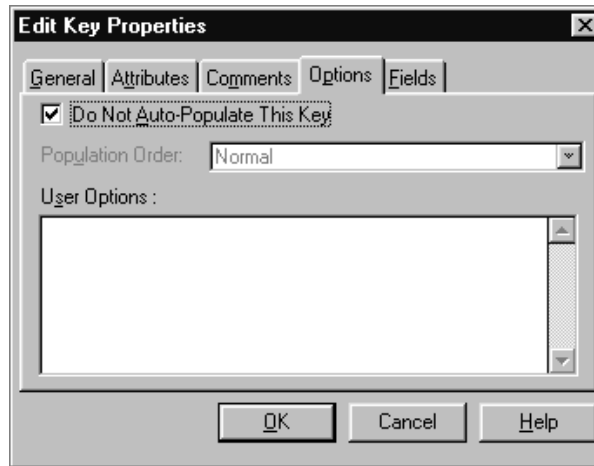
5. Przejdź w wyświetlonym okienku do zakładki **Options**.

Parametry tej zakładki pozwalają wpływać na działania kreatora. Opcje znajdujące się w górnej części okienka są wykorzystywane przez standardowy zestaw szablonów Clariona. Pole **User Option** pozwala na przekazywanie informacji do szablonów dostarczanych przez niezależnych producentów, pozwalają one na określenie opcji nie obsługiwanych przez szablony standardowe.

6. Zaznacz pole **Do Not Auto-Populate This Key**.

Zaznaczając to pole informujemy kreatora aplikacji, że dany klucz nie ma być wykorzystywany w okienku przeglądania rekordów pliku *Klienci*. Oznacza to, że nie pojawi się w tym okienku zakładka *KLI:KeyNrKlienta* pozwalająca na sortowanie rekordów wg wartości pola *NrKlienta*.

Oczywiście jest dużo więcej sposobów na dostosowanie pracy kreatora do naszych indywidualnych potrzeb. Więcej informacji na ten temat można znaleźć na przykład w rozdziale *Optimizing the Wizards* w podręczniku *Application Handbook*.



1. Wciśnij przycisk **OK**.
2. W okienku **Field / Key Definition** wciśnij przycisk **Close**.
3. Wciśnij przycisk **Close** w okienku dialogowym **Dictionary**. Gdy program zapyta Cię, czy zachować wprowadzone zmiany, wciśnij przycisk **Tak**.

To wszystko, co chcieliśmy zrobić, teraz pozwolimy kreatorowi aplikacji na jej wygenerowanie.

Korzystanie z kreatora aplikacji

Kreator Application Wizard generuje kompletną aplikację bazując na minimalnej ilości informacji wymaganej od programisty. Informacje te wprowadzamy w kolejnych krokach działania kreatora.

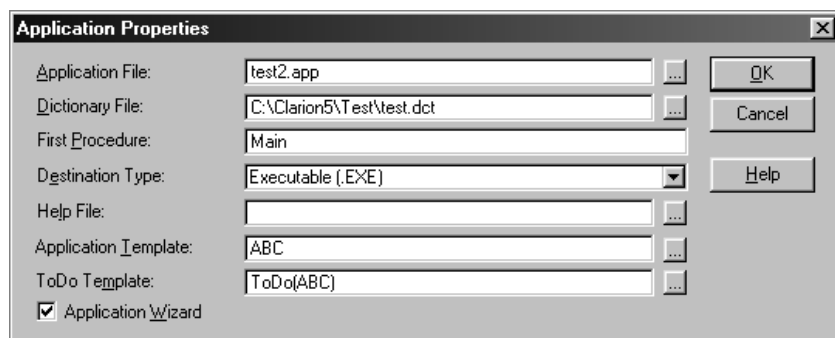
Tworzenie nowej aplikacji

1. Wybierz polecenie **File** ▶ **New** ▶ **Application**.

Na ekranie pojawi się okienko dialogowe **New**.

2. Przejdź do katalogu *C:\CLARION5\TEST*.
3. Wpisz nazwę tworzonej aplikacji – *TEST2* - w polu **Nazwa pliku**.
4. Usuń zaznaczenie pola **Use Quick Start**, następnie wciśnij przycisk **Zapisz**.

Na ekranie pojawi się okienko dialogowe **Application Properties**.



5. Wciśnij przycisk ... znajdujący się po prawej stronie pola **Dictionary File**.

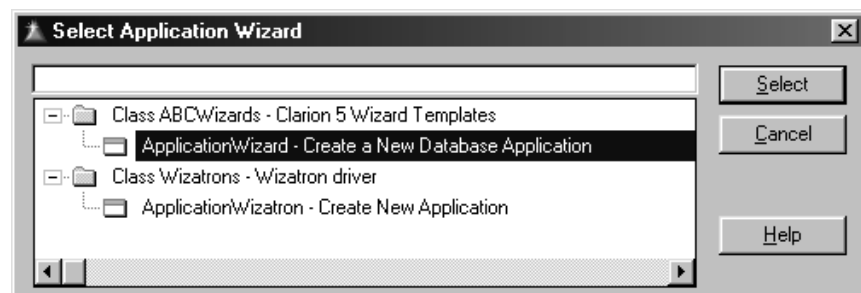
Zostanie wówczas wyświetlone okienko dialogowe **Select Dictionary**.

6. Zaznacz plik słownika *TEST.DCT* i wciśnij przycisk **Otwórz**.
7. Wciśnij przycisk ... znajdujący się po prawej stronie pola **Application Template**.

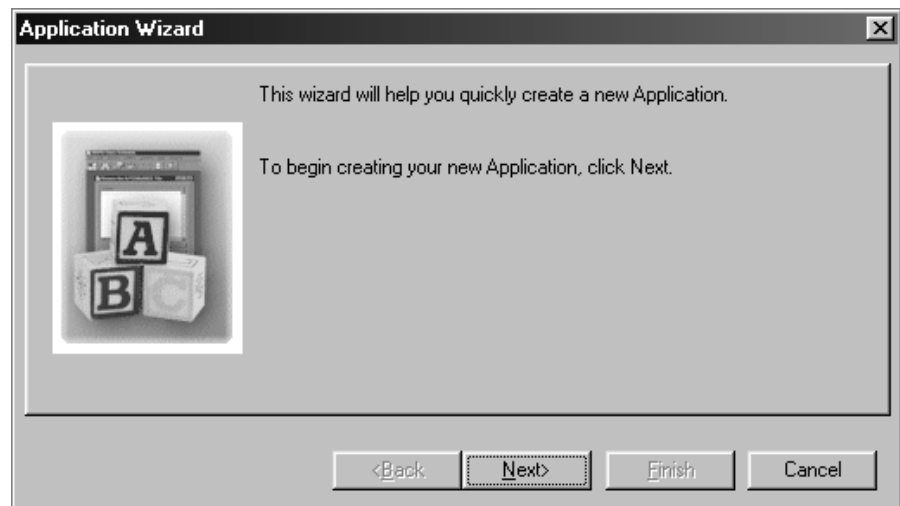
W wyświetlonym okienku **Application Type** zaznacz opcję *ABC* i wciśnij przycisk **Select**.

8. Zaznacz pole **Application Wizard**, następnie wciśnij przycisk **OK**.

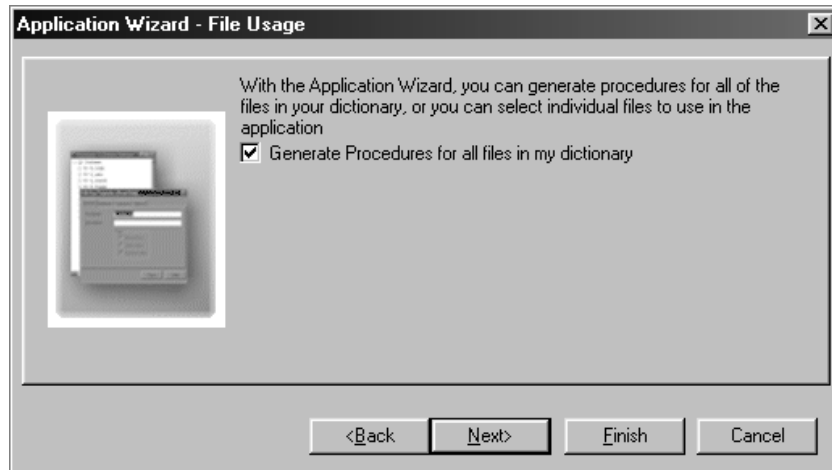
UWAGA: Jeśli korzystasz z Clarion 5 Professional Edition, pomijasz punkt 9.



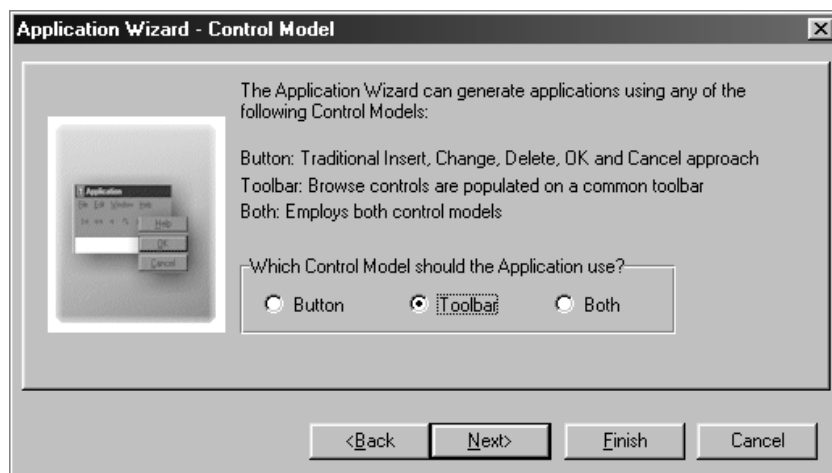
9. Zaznacz **Application Wizard – Create a new Database Application**, następnie wciśnij przycisk **Select**.



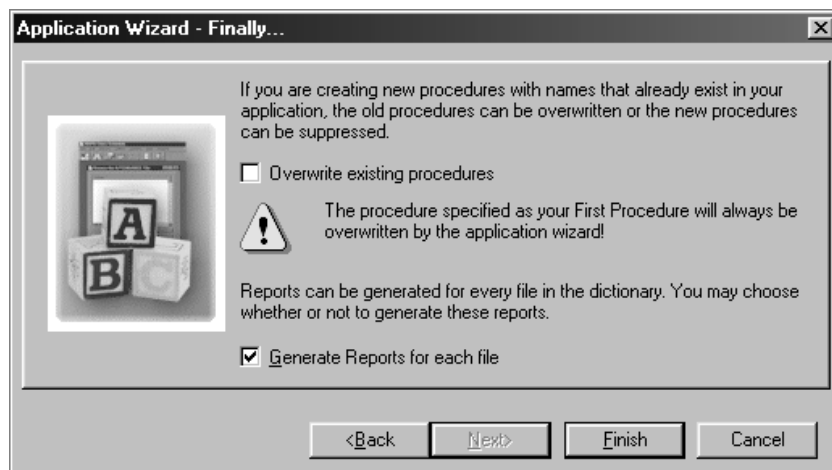
10. Wciśnij przycisk **Next**.



11. Wciśnij przycisk **Next**.

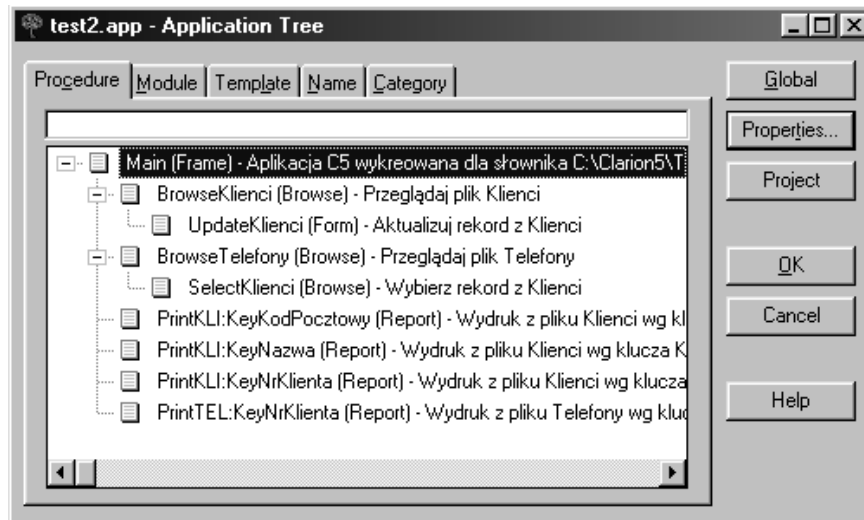


12. Wciśnij przycisk **Next**.



13. Wciśnij przycisk **Finish**.

Kreator Aplikacji wygeneruje teraz Twoją aplikację.



14. Wybierz polecenie **Project ▶ Run** (lub wciśnij przycisk *Run* na pasku narzędzi).

Gratulacje! Masz już na koncie dwie działające aplikacje stworzone w Clarionie.

Sprawdź swoją aplikację w działaniu i porównaj z aplikacją stworzoną wcześniej. Zauważysz, że występują między nimi pewne, niewielkie różnice. Wynikają one z tego, że poprzednią aplikację tworzyliśmy etapami, zamiast skorzystać z kreatora aplikacji.

OK, co udało nam się zrobić?

Podsumujmy krótko nasze działania:

- ◆ Zastosowaliśmy Quick Start Wizard do zdefiniowania pliku Klienci, w oparciu o który Quick Start Wizard utworzył słownik danych Data Dictionary i aplikację.
Gdy skompilowaliśmy i uruchomiliśmy tę aplikację, przekonaliśmy się, że Quick Start Wizard stworzył w pełni funkcjonalny, wielowątkowy program, wyposażony w okienka przeglądania i aktualizacji rekordów, a także w zestaw prostych raportów – wszystko bez pisania nawet jednej linii kodu!
- ◆ Zastosowaliśmy funkcję Quick Load w słowniku danych do bardzo szybkiego dodania nowej definicji pliku.
- ◆ Określiliśmy, że ten nowy plik ma korzystać z mechanizmu edit-in-place przy dodawaniu, aktualizacji i usuwaniu rekordów.
- ◆ Następnie zdefiniowaliśmy relację pomiędzy dwoma plikami i określiliśmy zasady zachowania integralności danych.
- ◆ Później wykorzystaliśmy kreator Browse Procedure Wizard do stworzenia procedury przeglądania rekordów z możliwością ich aktualizacji za pomocą mechanizmu edit-in-place. Po skompilowaniu i uruchomieniu aplikacji przekonaliśmy się, że procedura ta działa i jest wywoływana po wciśnięciu przycisku w okienku aktualizacji danych

klientów, pozwalając na aktualizację listy telefonów wybranego klienta. I znowu wszystko bez pisania nawet jednej linii kodu!

- ◆ Na zakończenie, wykorzystaliśmy kreator Application Wizard do utworzenia drugiej, kompletnej aplikacji bazodanowej operującej na tym samym słowniku danych.

Po jej skompilowaniu i uruchomieniu, przekonaliśmy się, że Application Wizard również stworzył w pełni funkcjonalny, wielowątkowy program, wyposażony w okienka przeglądania i aktualizacji rekordów, a także w zestaw prostych raportów – i znowu wszystko bez pisania nawet jednej linii kodu!

W jakieś trzydzieści minut – bez pisania jakiegokolwiek kodu – udało Ci się stworzyć dwie kompletne aplikacje operujące na dwóch wzajemnie powiązanych plikach danych. To prawdziwe osiągnięcie!

Jeśli możesz tak wiele zrobić przy zastosowaniu najprostszych funkcji Clariona, czego się można spodziewać, gdy nauczysz się korzystać z wszystkich funkcji, jakie daje Clarion?

Co robić teraz?

Skoro możesz zrobić tak wiele już za pomocą kilku kreatorów, czego możesz się spodziewać po zastosowaniu wszystkich pozostałych narzędzi, jakie daje Ci Clarion?

Oprócz kreatorów, Clarion daje Ci do dyspozycji zestaw produktywnych narzędzi Generators Aplikacji, występujących w postaci szablonów kontrolki, szablonów rozszerzeń, czy szablonów kodu. Z ich pomocą możesz w ogromnym stopniu zwiększyć funkcjonalność tworzonej aplikacji i dopasować ją do konkretnych wymagań. Narzędzia te są równie łatwe w zastosowaniu, jak kreatory. Pozwalają na modyfikację procedur i, na ogół, nie wymagają kodowania.

W następnym rozdziale tej książki została opisana filozofia tworzenia aplikacji w Clarionie. Zapoznaj się z nim dokładnie, gdyż jest to niezwykle ważny materiał. Clarion różni się od innych środowisk wytwarzania aplikacji. *Zrozumienie filozofii jego działania pozwoli na łatwiejsze opanowanie tego narzędzia.*

Opis Generators aplikacji można znaleźć w podręczniku *Learning Clarion*. Podręcznik ten przedstawia różne aspekty tworzenia aplikacji od podstaw, bez stosowania kreatorów, na przykładzie aplikacji obsługującej system zamówień. Pokazano w nim zastosowanie wszystkich narzędzi udostępnianych przez Clarion, jak również możliwości uzyskiwane poprzez tworzenie wstawek kodu źródłowego.

Do dzieła! To co zostało pokazane do tej pory, to tylko wierzchołek góry lodowej. *Jest tego dużo, dużo więcej!*

3 - PRZEGLĄD ŚRODOWISKA

Tao Clariona

Tao po chińsku oznacza drogę. Clarion tak naprawdę jest **nową drogą** w wytwarzaniu aplikacji dla środowiska Windows. Sposób na utworzenie kompletnych aplikacji w przeciągu 30 minut został pokazany w poprzednich rozdziałach. Jednak Clarion to dużo więcej, niż tylko kreatory. Tak więc teraz poświęcimy nieco czasu na przedstawienie podstawowych koncepcji zastosowanych w Clarionie.

Clarion to język programowania

Fundamentem Clariona (jako środowiska wytwarzania aplikacji) jest język programowania Clarion – język czwartej generacji (4GL), w pełni obiektowy i zorientowany na tworzenie aplikacji bazodanowych. Język Clarion to rzeczywiste narzędzie i masz je właśnie w swoich rękach. Ale Clarion to nie tylko język, to także pełny zestaw narzędzi opracowanych specjalnie dla celów tworzenia i zarządzania bazami danych dla biznesu.

Nadbudową języka programowania Clarion jest środowisko wytwarzania aplikacji – kompletny, zintegrowany zestaw narzędzi utworzony po to by uzyskać możliwość błyskawicznego tworzenia aplikacji (RAD- Rapid Application Development), błyskawicznego aktualizowania aplikacji (RAM – Rapid Application Maintenance) oraz błyskawicznego rozszerzania aplikacji (RAE- Rapid Application Enhancement). Wszystkie te narzędzia służą do wygenerowania kodu w języku Clarion, po to, by zrobić to za Ciebie. To co, czyni Clarion tak wydajnym, opiera się właśnie na zastosowaniu narzędzi do generowania kodu źródłowego. Ty z kolei możesz się skoncentrować na najważniejszym, czyli na określeniu co i jak ma robić tworzona aplikacja.

Poziomy abstrakcji

Zostało naukowo dowiedzione, że im dłużej będziesz się trzymać z daleka od „bebeczków” komputera, tym większą produktywność uda Ci się osiągnąć.

Nieco historii

Pierwsze komputery były programowane binarnie, w związku z czym praca programisty była bardzo trudna. Dużym ułatwieniem okazało się pojawienie języków assemblerowych, dzięki temu kod programów stał się bardziej czytelny. Każda instrukcja języka assemblera generowała jedną instrukcję maszynową.

Programiści posługujący się assemblerem szybko spostrzegli, że pewne fragmenty kodu często się powtarzają. Było to oczywiste, skoro wykonywały one podobne czynności, na przykład wyprowadzenie znaków na ekran. Było to przyczynkiem do powstania języków trzeciej generacji (3GL – Third Generation Languages), takich jak Fortran, C, PL/I i innych. Ich cechą

charakterystyczną było to, że pojedyncza instrukcja języka 3GL generowała cały zestaw instrukcji maszynowych umożliwiających wykonanie określonego zadania. Dzięki temu praca programistów stała się dużo bardziej wydajna. Osiągnęli oni wyższy poziom abstrakcji, odchodząc od wspomnianych „bebeczów” komputera. Negatywnym skutkiem ubocznym było to, że generowane były „standardowe” sekwencje kodu maszynowego, przez co utraciło się elastyczność dawaną przez assembler.

Dzień dzisiejszy – 4GL

Ten sam proces zaszedł w przypadku języków czwartej generacji, takich, jak Clarion. Dla przykładu, język Clarion automatycznie obsługuje wszystkie standardowe działania Windows (na przykład odtwarzanie wyglądu okna), które musiały być oprogramowane przez programistów posługujących się językami 3GL. W Clarionie program „Hello World” zajmie trzy linie kodu źródłowego, w C/C++ musisz ich napisać ponad 80!

Podsumowując, im wyższy poziom abstrakcji, tym wydajniejsza staje się praca programisty. Na każdym poziomie abstrakcji programiści stają się bardziej efektywni, pozwalając komputerowi pracować za siebie – poprzez generowanie standardowego kodu dla standardowych zadań. To jest właśnie fundamentalna zasada przyjęta w Clarionie.

Sekret wydajności

Podstawowym problemem, podczas operowania na wyższych poziomach abstrakcji, jest utrata elastyczności. Nie wszystko można zrobić dokładnie tak, jakby się chciało. Clarion jednak rozwiązuje ten problem.

Sekret prawdziwej wydajności pracy programisty bez utraty elastyczności polega na tym, by **zawsze operować na najwyższym poziomie abstrakcji** umożliwiającym wykonanie określonego zadania w taki sposób, w jaki powinno ono być wykonane. Clarion zawiera wiele poziomów abstrakcji umożliwiając Ci korzystanie z tego, który pozwoli na jak najlepsze wykonanie postawionego zadania. Nie zmusza Cię do operowania na wyższym poziomie, ani też do grzebania się w kodzie źródłowym, jeśli tego nie chcesz. Narzędzia generują zorientowany obiektowo kod źródłowy w oparciu o *Application Builder Class (ABC) Library* (wyższy poziom), a jeśli chcesz, możesz do tego dołożyć własny kod (niższy poziom), by dostosować do własnych potrzeb działanie funkcji określone przez standardowe szablony.

Innymi słowy, sekret wydajności polega na tym, by pracować mniej szybko i lżej. To właśnie umożliwia Ci Clarion – generuje standardowy kod, dzięki czemu nie musisz go samodzielnie pisać i jest na tyle elastyczny, że pozwala Ci na samodzielne wzbogacenie Twojej aplikacji o wszystkie niezbędne „wodotryski”. Jest to podstawowa reguła Clariona – osiągnęliśmy ją stosując programowanie oparte o szablony.

Programowanie w oparciu o szablony

Praca Generators Aplikacji polega na wykorzystaniu szablonów. Oznacza to, że aplikacja powstaje w oparciu o kod wygenerowany za pomocą predefiniowanych szablonów. Szablony procedur, kontrolki, rozszerzeń, czy kodu tworzą kod w języku Clarion ogromnie przyspieszając Twoją pracę. Szablony te udostępniają wszystkie zalety programowania zorientowanego obiektowo; domyślny zestaw szablonów oparty na *Application Builder Class (ABC) Library* pozwala na wygenerowanie w pełni obiektowego kodu aplikacji.

Czyni to z szablonów klucz do błyskawicznego tworzenia aplikacji (RAD- **Rapid Application Development**).

Co to jest szablon?

Szablon w Clarionie nie jest narzędziem służącym do jednorazowego wygenerowania fragmentu kodu, który musisz następnie „obrabiać” samodzielnie (tak właśnie traktują szablony inne narzędzia). W Clarionie jest to interaktywne narzędzie, umożliwiające wielokrotną zmianę parametrów, rozszerzanie i modyfikowanie danego fragmentu aplikacji. Jeśli zmienisz informacje udostępniane szablonowi, zostanie ponownie wygenerowany kod w pełni je uwzględniający. Ta interaktywność szablonów jest kluczem do uzyskania błyskawicznego aktualizowania aplikacji (RAM – **Rapid Application Maintenance**).

Szablony Clariona umożliwiają umieszczanie w nich wstawek kodu źródłowego; są na to przewidziane specjalne, w pełni opisane punkty, przez co umożliwiają łatwą identyfikację momentu wykonania wstawionego kodu. Tym samym łatwo jest kontrolować zachowanie danej procedury. Nasze zmiany nie będą anulowane przy kolejnym generowaniu procedury, gdyż są zapamiętywane w procedurze. Ta łatwa modyfikowalność działania szablonów jest kluczem do błyskawicznego rozszerzania aplikacji (RAE- **Rapid Application Enhancement**).

Wszystkie szablony są przechowywane w specjalnym pliku rejestru (REGISTRY.TRF). Plik ten zawiera kod wykonywalny i struktury danych, które można modyfikować i wielokrotnie wykorzystywać. Do modyfikacji szablonów służy specjalny Edytor Rejestru Szablonów (**Setup ▶ Template Registry**). Za jego pomocą można zmieniać domyślny wygląd okienek generowanych w oparciu o poszczególne szablony.

Możesz modyfikować zawartość samych szablonów. Możesz także korzystać z szablonów dostarczanych przez niezależnych producentów, czy też pisać własne szablony. Generator Aplikacji potrafi skorzystać z nich wszystkich, co powoduje, że jest on łatwo rozszerzalnym narzędziem.

Jak korzystać z szablonów?

Gdy stworzysz nową procedurę, wskazujesz dla niej szablon, który wygeneruje taki kod, by procedura mogła spełniać stawiane przed nią zadania. W kolejnym kroku dokonujesz aktualizacji tej procedury, tak by dostosować ją do specyficznych wymagań. Szablony procedur zawierają takie elementy, jak okienka przeglądania danych (Browse), czy też formularze aktualizacji rekordów (Form). Jeśli procedura ma otwierać okienko z systemem menu, akcje (procedury) dla poszczególnych poleceń tego menu są automatycznie dołączane do struktury aplikacji i oznaczane jako „ToDo”, czyli do oprogramowania.

Typowym sposobem na dopasowanie procedury do specyficznych wymagań, jest użycie Edytora okna (lub Edytora Raportu). Są to narzędzia wizualne, pozwalające na projektowanie wyglądu okna poprzez wskazywanie i klikanie myszką. Wystarczy wskazać kontrolkę w okienku narzędzi, a następnie kliknąć obszar okna, by ją w nim umieścić. Jej parametry możemy zmieniać za pomocą poleceń menu podręcznego otwieranego po kliknięciu jej prawym przyciskiem myszki.

Gdy już utworzysz procedurę, możesz użyć szablonów kontrolek, rozszerzeń lub kodu. Pozwalają one na rozszerzenie funkcjonalności Twojej procedury. Szablony kontrolek tworzą kontrolki i generują standardowy kod obsługujący

ich działanie. Wszystkie kontrolki pojawiające się domyślnie w okienku po wybraniu szablonu, są właśnie takimi kontrolkami wraz z przypisanymi im szablonami kontrolek. Szablony rozszerzeń dołączają do procedury kod rozszerzający jej funkcje, typowy przykład to wyświetlanie aktualnego czasu w pasku stanu okienka.

Inną drogą dopasowania działania procedury do specyficznych wymagań jest korzystanie ze wstawek kodu źródłowego. Generator Aplikacji wyświetla strukturę wszystkich punktów procedury, w których takie wstawki można umieszczać. Może to być przed, wewnątrz lub po fragmentach wygenerowanego kodu sterującego działaniem procedury, jak również dla każdego zdarzenia lub kontrolki obsługiwanej przez tą procedurę. Możesz także bezpośrednio edytować wstawiony kod w kontekście całego wygenerowanego kodu. Pozwala to precyzyjne określenie miejsca, w którym jest on wykonywany. Kod ten możesz pisać samodzielnie, możesz też użyć szablonu kodu, który wygeneruje go za Ciebie. Generator Aplikacji generuje kod aplikacji w oparciu o szablony użyte dla poszczególnych procedur i Twoje modyfikacje w nich wprowadzone (włączając w to oczywiście wstawki kodu źródłowego).

Clarion dostarcza bogaty asortyment standardowych szablonów wystarczających do błyskawicznego wytworzenia aplikacji. Tak jak ten podręcznik przygotowuje Cię do zastosowania Kreatorów, tak podręcznikiem wprowadzającym do Generatora Aplikacji jest *Learning Clarion*. Opisuje on sposoby użycia szablonów umożliwiające stworzenie kompletnej aplikacji Windows.

Poziomy abstrakcji Clariona

Do tej pory pracowaliśmy na najwyższym poziomie abstrakcji Clariona – z kreatorami. Oprócz nich będziesz korzystać i z innych elementów, opisanych pokrótce poniżej.

Edytor słownika danych (Dictionary Editor) i Data Modeller

Edytor słownika danych jest repozytorium, w którym są przechowywane definicje wszystkich plików danych wraz z domyślnymi parametrami poszczególnych pól. Jeżeli dysponujesz istniejącym plikiem danych, stworzenie jego definicji jest bardzo proste, wystarczy zaimportować go do słownika danych.

Kreator aplikacji (Application Wizard)

Generuje kompletną, „standardową” aplikację, w ciągu kilku minut, bazując na definicjach plików danych i parametrach pól zapisanych w słowniku danych.

Kreatory procedur (Procedure Wizards)

Generują „standardowe” procedury (okienka przeglądania, formularze) bazując na ustawieniach zdefiniowanych w słowniku danych.

Szablony procedur (Procedure Templates)

Generują „standardowe” procedury, takie jak okno sterujące aplikacją (Frame), raporty, bazując na oknach zdefiniowanych przez programistę za pomocą Edytora okien (Window Formatter) lub Edytora raportów (Report Formatter)

oraz opcjach określonych we właściwościach wybranego dla procedury szablonu.

Szablony kontrolek (Control Templates)

Dołączają „standardowe” kontrolki do Twoich okien i raportów (np. okno ze strukturą powiązanych plików) oraz generują kod niezbędny do obsługi działania tych kontrolek.

Szablony rozszerzeń (Extension Templates)

Generują dodatkowy kod w procedurze wzbogacając je w funkcje, które nie są standardowo obsługiwane (np. wyświetlanie aktualnej daty i czasu w pasku stanu okienka).

Szablony kodu (Code Templates)

Generują kod wstawiany do procedury, który zazwyczaj powoduje wykonanie określonej akcji związanej z konkretną kontrolką okna (np. walidacja danych wprowadzanych do pól).

Wstawki kodu źródłowego (Embedded Source Code)

Umożliwiają programiście dołączanie w określonych miejscach procedury własnego kodu źródłowego i w ten sposób modyfikować i rozszerzać standardowe działanie wybranego szablonu.

Szablony kodu źródłowego (Source Templates)

Szablony umożliwiające tworzenie kompletnych procedur i funkcji napisanych w języku Clarion, ale nadal uwzględnianych przez Generator Aplikacji.

Windows API oraz C Standard Library

W kodzie zapisanym w języku Clarion (w procedurach i we wstawkach kodu źródłowego), możesz bezpośrednio wywoływać funkcje Windows API (Application Programming Interface) oraz funkcje standardowej biblioteki języka C. Prototypy funkcji Windows API można wygenerować specjalnym programem narzędziowym, prototypy funkcji standardowej biblioteki C są udokumentowane w *Programmer's Guide*.

Kod w języku C/C++ lub Modula-2

Możesz utworzyć funkcje w języku TopSpeed C/C++ lub TopSpeed Modula-2 i dołączyć do programu tworzonego w Clarionie. Kompilatory te są dołączone do Clariona i przywoływane w momencie dołączenia do projektu modułu kodu źródłowego .C, .CPP, .MOD (można je również zakupić jako samodzielne produkty).

Jak widzisz, poziomy abstrakcji Clariona umożliwiają Ci programowanie na dowolnym poziomie, począwszy od najwyższego, reprezentowanego przez Kreatory, aż do najniższego, czyli kodu w języku C (a nawet w Assemblerze, o ile chce Ci się pracować tak ciężko).

Klucz do maksymalnej wydajności

Kluczem do osiągnięcia maksymalnej wydajności w Clarionie jest **praca na jak najwyższym poziomie abstrakcji**:

- ◆ Pozwól **Kreatorowi aplikacji** na utworzenie podstawowego szkieletu aplikacji.

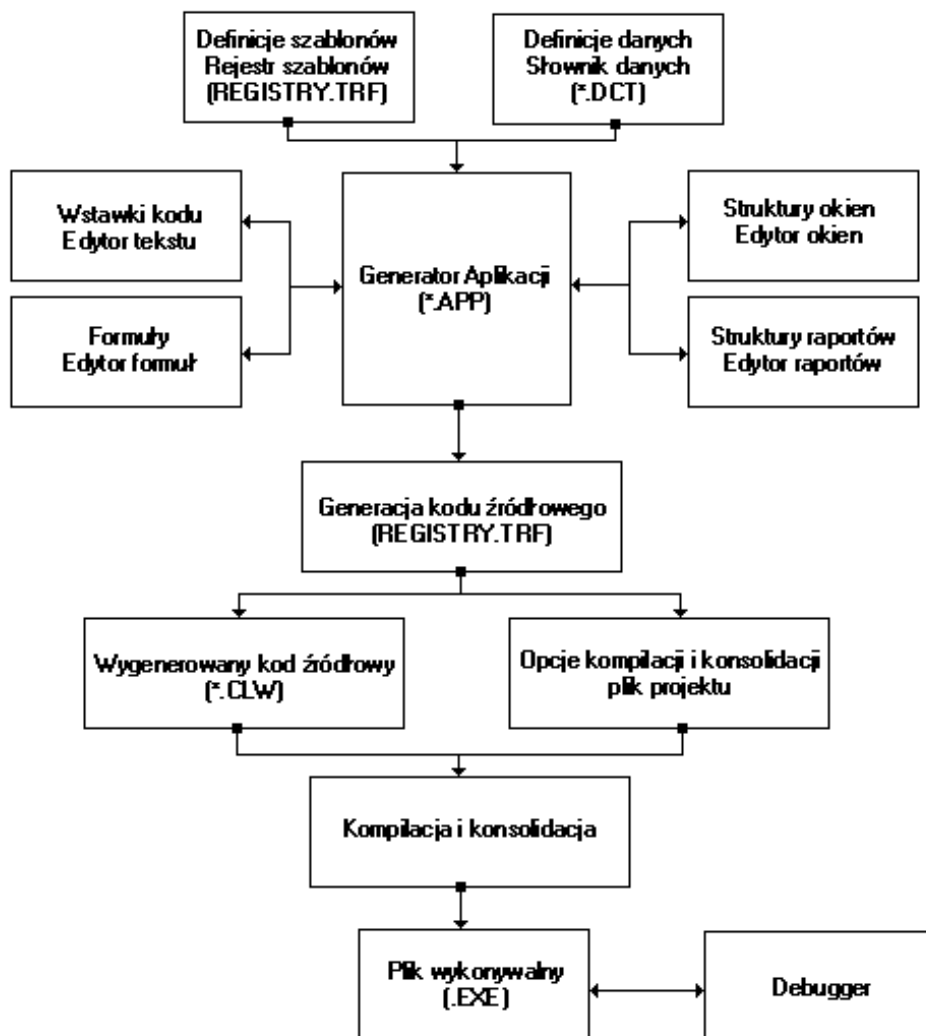
- ◆ Użyj **Kreatorów procedur** do tworzenia nowych procedur, o ile są potrzebne.
- ◆ Wprowadź modyfikacje w **procedurach** używając Edytora okna lub Edytora raportu.
- ◆ Dołącz do procedur **szablony kontrolek, rozszerzeń i kodu**.
- ◆ Dostosuj procedury do specyficznych wymagań za pomocą **wstawek kodu źródłowego**.
- ◆ Jeśli chcesz uzyskać efekt, którego nie zapewni Ci żaden standardowy szablon, użyj **szablonu kodu źródłowego** i napisz własną procedurę lub funkcję w języku Clarion; możesz też użyć szablonów dostarczanych przez niezależnych producentów.
- ◆ Możesz zejść do poziomu wywołań funkcji **Windows API i C/C++**, ale raczej tylko wtedy, gdy jest to niezbędnie konieczne.

Taka jest właśnie filozofia Clariona – pozwolić dostępnym narzędziom na wykonanie całej „brudnej” roboty. Działając w zgodzie z nią, przekonasz się, że trudno o bardziej wydajne narzędzie.

Clarion, środowisko wytwarzania aplikacji

Środowisko wytwarzania aplikacji składa się z kilku podstawowych narzędzi, które są wzajemnie ze sobą powiązane. Podczas korzystania z Generатора Aplikacji w okienkach dialogowych są dostępne przyciski i polecenia prowadzące nas do innych narzędzi.

Poniżej przedstawiono diagram ilustrujący współpracę poszczególnych narzędzi podczas tworzenia aplikacji; Generator Aplikacji jest centralnym punktem tego procesu.



W tym punkcie opiszemy pokrótce każde z narzędzi, w takim porządku, w jakim korzysta się z nich zazwyczaj podczas tworzenia aplikacji od podstaw. Każde z narzędzi zawiera okienka dialogowe, za pomocą których programista opisuje funkcje aplikacji, którą ma wygenerować Generator Aplikacji. Na Twoje polecenie, Generator Aplikacji generuje specyficzny kod źródłowy, a system obsługi projektu (Project System) kompiluje go i konsoliduje do postaci programu wykonywalnego.

Można powiedzieć, że programowanie w Clarionie polega, w wielu przypadkach, na swoistej podróży poprzez okienka dialogowe poszczególnych narzędzi. Przy czym podróż ta nie ma ściśle określonej trasy, wszystko zależy od tego, co wybierzesz w poszczególnych jej punktach.

Edytor słownika danych

Słownik danych (plik .DCT) jest tworzony za pomocą Edytora słownika danych. Przechowuje on opis bazy danych, włączając w to definicje plików, kluczy, indeksów, stosowanych sterowników baz danych, relacji między plikami, pól, reguł walidacji pól, zasad zachowania integralności danych itd. *Jest to pierwszy plik tworzony podczas tworzenia nowej aplikacji.*

Możesz tworzyć definicje plików od podstaw, możesz też zaimportować je z istniejących już plików danych. Jeśli dysponujesz wersją Enterprise Edition, możesz korzystać ze specjalnego narzędzia synchronizującego – Dictionary Synchronizer, którego zadaniem jest synchronizacja słownika danych Clariona ze słownikiem danych bazy SQL. Pozwala to na pobranie wszystkich definicji plików ze słownika danych bazy SQL, co zaoszczędzi nam wiele pracy, i synchronizowanie go z systemem zarządzania relacyjną bazą danych SQL w każdej chwili, niezależnie od zmian nanoszonych przez administratora bazy danych SQL.

Inną funkcją środowiska wytwarzania aplikacji jest umieszczanie w oknach informacji związanych z polami rekordów zdefiniowanych w słowniku danych. Generator Aplikacji tworzy kod dostępu do plików danych bazując na informacjach wprowadzonych w słowniku danych. Może on nawet wygenerować w pełni funkcjonalną aplikację opierając się tylko i wyłącznie na tych informacjach!

Generator Aplikacji

Generator Aplikacji generuje kod źródłowy aplikacji w oparciu o szablony wybrane dla poszczególnych procedur. Pozwala również na zdefiniowanie globalnych i lokalnych zmiennych i zmodyfikowanie procedur pod kątem specyficznych wymagań za pomocą narzędzi wizualnych i wstawek kodu źródłowego.

Generator Aplikacji daje dostęp do innych narzędzi środowiska wytwarzania aplikacji, dzięki czemu możesz zmieniać wygląd i działanie okien, systemu menu, raportów i innych elementów interfejsu użytkownika Twojej aplikacji. Szablony zawierają swoje własne kontrolki, pojawiają się one w Generatorze Aplikacji, pozwalające na dostosowanie ich funkcji do specyficznych wymagań.

Edytor okien

Wizualne narzędzie umożliwiające zmianę wyglądu oka i umieszczonych w nim kontrolki. Automatycznie generuje kod źródłowy dla elementów, które umieścisz w oknie.

Edytor raportów

Narzędzie podobne do Edytora okien, z tym, że dotyczy projektowania wyglądu raportów. Twoje zadanie polega na umieszczeniu kontrolki na stronie raportu. W czasie działania aplikacji specjalny moduł obsługi raportów przetwarza rekordy rozmieszczając dane na stronie raportu zgodnie ze zdefiniowanym wzorem.

Edytor tekstu

W pełni funkcjonalny edytor programisty, za pomocą którego możesz wprowadzać wstawki kodu źródłowego. Podczas korzystania z Generatora Aplikacji, Edytor tekstu jest stosowany właśnie do umieszczania w procedurach

fragmentów kodu w języku Clarion. Możesz także edytować kod źródłowy zapisany w plikach .CLW.

Edytor tekstu zawiera kilka udogodnień, między innymi kontrolę składni poprzez wyświetlanie instrukcji Clariona w różnych kolorach. Jest wyposażony również w mechanizmy wyszukiwania i zamiany tekstów i inne standardowe mechanizmy edycyjne.

Edytor formuł

Edytor formuł pomaga Ci w szybkim generowaniu i zarządzaniu prostymi i złożonymi formułami obliczeniowymi bazującymi na wyrażeniach matematycznych i łańcuchowych. Edytor formuł kontroluje składnię formuł, daje dostęp do zmiennych, funkcji, operatorów. Dzięki temu mamy pewność, że wygenerowana formuła jest poprawna składniowo.

System obsługi projektu

Generator Aplikacji tworzy automatycznie plik projektu (.PRJ) dla aplikacji. Plik ten zawiera opcje kompilacji i konsolidacji, włączając w to kod dla debuggera, ustawienia optymalizacyjne, dołączane pliki zewnętrzne, pliki źródłowe, biblioteki.

Debugger

Proces debugowania wymaga na ogół wielokrotnego zatrzymywania i wznowiania pracy aplikacji w jej różnych punktach w celu sprawdzenia w nich wartości zmiennych. Pozwala to na wykrywanie ewentualnych błędów funkcjonalnych. Debugery Clariona (16-bitowy i 32-bitowy) pozwalają na wyświetlanie różnych okienek zawierających fragmenty kodu źródłowego, wartości wybranych zmiennych, listy aktywnych procedur itd.

W dokumentacji znajdują się trzy poradniki dotyczące korzystania z debuggerów (znajdują się one w podręczniku *User's Guide*). Radzimy Ci zapoznać się z nimi po skończeniu lektury tego podręcznika oraz podręcznika *Learning Clarion*.

System pomocy

Clarion jest wyposażony w rozbudowany, kontekstowy system pomocy. By go wywołać, wystarczy wcisnąć przycisk **Help** lub wcisnąć klawisz **F1**.

Sekcja Common Question systemu pomocy zawiera instrukcje i przykłady wykonania najczęstszych zadań związanych z tworzeniem aplikacji. Możesz pobierać z niej przykładowy kod, poprzez zaznaczenie go i skopiowanie do Schowka, a następnie wklejenie we własnym programie. Wybierz polecenie **Help ▶ Contents**, następnie wcisnij przycisk **How do I ...?**.

System pomocy składa się z pliku głównego i plików uzupełniających. Plik główny można przeszukiwać wg indeksu lub za pomocą słów kluczowych. Jeśli takie wyszukiwanie ma być przeprowadzone w pliku pomocniczym, należy najpierw go otworzyć.

Enterprise Tools

Poniżej został zamieszczony krótki opis narzędzi dostarczanych wraz Enterprise Edition, a które nie są standardowo dostępne w wersji Professional Edition, Narzędzia te są w pełni udokumentowane w podręczniku *Enterprise Tools*.

Dictionary Synchronizer

Dictionary Synchronizer umożliwia synchronizację jednego słownika danych z drugim (dostępny tylko w Enterprise Edition). Można synchronizować ze sobą dwa słowniki Clarion (pliki .DCT files) lub też słownik Clarion ze słownikiem utrzymywanym przez serwer SQL (w obu kierunkach). Pozwala to na pobranie wszystkich informacji o definicjach plików i ich relacjach z serwera SQL eliminując konieczność „ręcznego” przeprowadzenia tej operacji, jak również zapewnia stałą synchronizację z serwerem SQL, niezależnie od zmian, jakie może wprowadzać jego Administrator.

Data Modeller

Słownik danych (plik .DCT) może być zarządzany również za pomocą Data Modeller (dostępny również jako produkt rozszerzający dla Professional Edition). Jest to narzędzie wizualnego modelowania danych, pozwalające na podgląd bazy danych z prespektywy diagramu powiązań relacyjnych.

Version Control/Team Developer

Narzędzie Version Control/Team Developer umożliwia wspólną pracę nad projektem. Pozwala na nadawanie numerów wersji aplikacji, przywracanie jej poprzednich wersji. Jest również dostępny jako rozszerzenie dla Professional Edition. W zespole programistów, każdy z jego członków może oznaczać jako własną część pliku aplikacji .APP i ją aktualizować. Wielu programistów, bez konfliktów, może wówczas pracować na jednym pliku aplikacji .APP. Jest to możliwe dzięki zastosowaniu wsparcia ze strony specjalizowanego produktu o nazwie PVCS, który jest standardem przemysłowym w tej dziedzinie.

Business Math Library

Business Math Library to zestaw standardowych funkcji biznesowych i statystycznych do zastosowania w aplikacjach Clarion. Jest także dostępna jako rozszerzenie dla Professional Edition. Zawiera także pełny zestaw szablonów ułatwiających zastosowanie tych funkcji w aplikacji.

Wise for Clarion

Wise for Clarion to kompletne narzędzie do tworzenia programów instalacyjnych opracowane przez Wise Solutions, Inc. Jest to jego specjalna edycja, przeznaczona dla Clariona, oparta na znanej linii produktów Wise.

Co dalej?

Co masz zrobić, jeśli chcesz dowiedzieć się więcej na temat programowania w Clarionie?

- ◆ Najlepszym rozwiązaniem będzie lektura podręcznika *Learning Clarion*. Zawiera on dwa poradniki, których treść wprowadzi Cię w zagadnienia związane z korzystaniem narzędzi udostępnianych przez Clarion.
- ◆ TopSpeed oferuje seminaria edukacyjne w różnych miejscach świata. Zadzwoń do serwisu obsługi klientów (800) 354-5444 lub (954) 785-4555.
- ◆ Przyłącz się do lokalnej grupy użytkowników Clariona, by móc wymieniać się doświadczeniami.
- ◆ Korzystaj z forum TopSpeed na CompuServe (GO TOPSPEED) lub z grupy dyskusyjnej *comp.lang.clarion* w Internecie. Tutaj wymieniają się informacjami programiści z całego świata.

Wewnętrzny serwer news firmy TopSpeed utrzymuje grupy dyskusyjne dla wszystkich produktów TopSpeed. Dzięki temu serwerowi można wymieniać doświadczenia, pomysły z innymi programistami Clarion, otrzymywać od nich, czy od pracowników TopSpeed, pomoc w rozwiązaniu pojawiających się problemów. Zaloguj się do serwera *tsnews.clarion.com* i zasubskrybuj grupy dyskusyjne, w których chesz brać udział.

Dodatkowo istnieje ogólnie dostępna w sieci Usenet grupa *comp.lang.clarion* pozwalająca na wymianę doświadczeń programistów Clarion z całego świata.

Powodzenia!