

Robert BRZESKI, Damian PYTEL, Piotr BAJERSKI  
Politechnika Śląska, Instytut Informatyki

## MOŻLIWOŚCI PRZEPROWADZENIA PROCESU IMPORTU DANYCH Z BAZY VISUAL FOXPRO DO MSSQL SERVER

**Streszczenie.** W artykule przedstawiono metodologię oraz algorytmy umożliwiające przeprowadzenie importu danych z bazy Visual FoxPro do MSSQL Server. Częściowo przedstawiona została również aplikacja realizująca taki proces. Szczególna uwaga położona jest na zagadnienia związane z bezpieczeństwem oraz wydajnością importu danych.

**Słowa kluczowe:** Visual FOXPRO, MSSQL, import danych

## THE POSSIBILITY OF EXECUTION OF PROCESS OF THE IMPORT OF DATA FROM THE VISUAL FOXPRO DATABASE TO MSSQL SERVER

**Summary.** The paper presents methodology and algorithms enabling execution of import of the data from the Visual FoxPro database to MSSQL Server. Application realizing such process is also introduced partly. The special attention is laid on questions connected with safety and the efficiency of the import of data.

**Keywords:** Visual FOXPRO, MSSQL, import of the data

### 1. Geneza realizacji tematu

Szybki rozwój systemów informatycznych we współczesnym świecie z jednej strony daje użytkownikom końcowym coraz większe możliwości, m.in. w zakresie swojej funkcjonalności, z drugiej natomiast, w związku z tym, że wiele nowych rozwiązań i powstających produktów jest niekompatybilnych z używanymi już systemami, taki rozwój powoduje czasami utrudnienia w korzystaniu z całego zestawu produktów informatycznych. W wielu przedsiębiorstwach część kosztów jest przeznaczana na dostosowanie do siebie różnych, niekompatybilnych wzajemnie rozwiązań. Dotyczy to np. systemów bazodanowych. To, jaka baza da-

nych jest używana w takich systemach, zależy od wielu czynników, przy czym wybór jest ograniczony do serwerów dostępnych w danym czasie na rynku. Ten wybór w miarę upływu czasu się zmienia. Prowadzi to do sytuacji, w której po jakimś czasie konieczne staje się zastosowanie serwera baz danych zupełnie niekompatybilnego z tym używanym do tej pory. Taka sytuacja wystąpiła w Spółdzielczej Kasie Oszczędnościowo-Kredytowej (SKOK) „Piast”. Dane gromadzone i przetwarzane przez bazę Visual FoxPro zaczęły być potrzebne na Microsoft SQL Server 2008 w wersji 64-bitowej oraz Microsoft SQL Server 2005 w wersji 32-bitowej. Ta potrzeba spowodowała rozpoczęcie prac związanych z analizą możliwości przeprowadzenia importu danych z jednej bazy do drugiej.

## 2. Wymagania użytkownika systemu realizującego proces importu

Obecna sytuacja w SKOK „Piast” wymusza dalsze używanie bazy Visual FoxPro 9. Jednocześnie gromadzone w tej bazie dane muszą być systematycznie, tzn. po każdym dniu roboczym, w godzinach wieczornych, kopiowane do Microsoft SQL Server. W danych tych mogą pojawiać się pewne błędy, które powinny być wykrywane w czasie importu, tak aby istniała później możliwość ich ręcznej korekty. Głównymi wymogami aplikacji realizującej import danych powinny być niezawodność, szybkość działania oraz możliwość wykrywania błędów. Wymóg niezawodności wynika z bardzo ważnej roli systemów wykorzystujących dane SQL Serwera. Procesy działania aplikacji importującej dane muszą być weryfikowalne, aby można było określić poprawność ich wykonania. W procesie tym nie ma miejsca na błędy, gdyż mogą powodować one duże opóźnienia i straty finansowe. Aplikacja powinna umożliwiać weryfikację wykonanych procedur, tak aby pracownik mógł podjąć decyzję o zakończeniu pracy. Prędkość działania aplikacji także jest bardzo ważnym aspektem, gdyż aplikacja używana będzie w godzinach wieczornych, gdzie jej zadaniem będzie import dużej ilości danych. Proces ten jest kluczowy, ponieważ skopiowane dane w następnym dniu roboczym umożliwią pracę wielu aplikacjom zewnętrznym.

Szukając gotowych rozwiązań, umożliwiających import danych z plików Visual FoxPro na SQL Server 2005/2008, można znaleźć Visual FoxPro OLEDB Provider oraz inne aplikacje, np. DBConvert for FoxPro & MS SQL.

Visual FoxPro OLE DB Provider (VfpOleDB.dll) udostępnia interfejs dający możliwość dostępu do plików bazy danych Visual FoxPro przy użyciu innych języków programowania oraz aplikacji. Jednak Visual FoxPro OLE DB umożliwia pracę wyłącznie na systemach 32-bitowych (brak wersji dla systemów 64-bitowych). Działanie Visual FoxPro OLE DB jest jednak bardzo powolne, a kolejnym minusem zaimportowanych danych jest niepoprawne interpretowanie danych tekstowych oraz pól typu ‘datetime’, posiadających wartości ‘null’.

Natomiast DBCConvert for FoxPro & MS SQL jest komercyjną aplikacją niekorzystającą z Provider-a VFP OLE DB. Zaletą aplikacji jest możliwość importu kluczy oraz indeksów. Minusem aplikacji jest jednak brak wykonywania testów i możliwości raportowania poprawności wykonania importu danych.

Z powodu wad wspomnianych aplikacji, zdecydowano się na realizację własnej, rozpoznającej i raportującej znalezione błędy, aplikacji, nad którą będzie się miało pełną kontrolę i pozwalającą na wprowadzenie zmian w jej działaniu w miarę pojawiania się nowych wymagań. W ten sposób uzyska się możliwość zapewnienia większego bezpieczeństwa związanego z jej używaniem.

### **3. Analiza możliwości przeprowadzenia procesu importu danych**

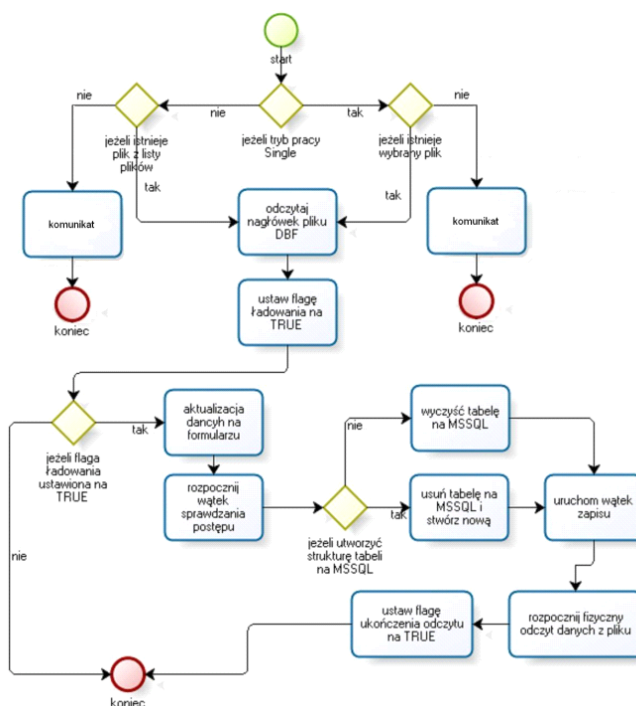
#### **3.1. Struktura pliku Visual FoxPro**

Realizację aplikacji realizującej import danych rozpoczęto od analizy struktury pliku Visual FoxPro [1]. Odczytanie danych z pliku DBA musi być zawsze poprzedzone odczytaniem niezbędnych danych z nagłówka. Z analizy nagłówka pliku DBA widać, że pierwsze 32 bajty zawierają wszystkie niezbędne informacje dotyczące struktury tabeli. Z nagłówka można odczytać informacje, takie jak: liczba kolumn, liczba rekordów, pozycja pierwszego rekordu w pliku, typ bazy danych itp. Informacjami potrzebnymi przy odczytywaniu danych z pliku oraz przy migracji do innych baz danych, są także opisy definicji poszczególnych kolumn tabeli. W przypadku gdy tabela zawiera pola typu MEMO, zawartość tych pól przechowywana jest w plikach o rozszerzeniu FPT. Plik DBF w polu MEMO zawiera jedynie „odnośnik” prowadzący do odpowiedniego miejsca w pliku FPT. W związku z tym, do poprawnego odczytania kompletu danych wymagana jest praca na dwóch typach plików – DBF oraz FPT.

#### **3.2. Algorytm wątku odczytu**

Jednym z dwóch kluczowych algorytmów wykorzystanych do realizacji procesu importu jest algorytm odczytu danych z plików Visual FoxPro (rys. 1). Wątek odczytu w pierwszej kolejności sprawdza, czy plik istnieje, w zależności od trybu pracy sprawdzane są różne lokalizacje plików. Jeżeli plik został odnaleziony, ustawiana jest flaga zezwalająca na fizyczny odczyt danych z pliku. W przypadku gdy flaga ta nie została ustawiona, system kończy wątek nie odczytując danych z pliku i w związku z tym nie jest uruchamiany wątek zapisu danych na SQL Server. Przy ustawionej fladze aplikacja przeprowadza aktualizację danych w formularzu i następnie uruchamia wątek sprawdzający postęp odczytu oraz zapisu danych.

W kolejnym kroku aplikacja sprawdza, czy struktura tabeli powinna zostać utworzona od nowa. Jeżeli tak, to tabela na serwerze jest usuwana i tworzona od nowa, w przeciwnym wypadku następuje jedynie usunięcie danych z tabeli. Po tej operacji aplikacja uruchamia wątek zapisu danych (rys. 3)(podrozdz. 3.4) i przechodzi do fizycznego odczytu danych z plików (podrozdz. 3.3). Po zakończeniu odczytu ustawiana jest flaga informująca o zakończeniu procesu odczytu danych z pliku.

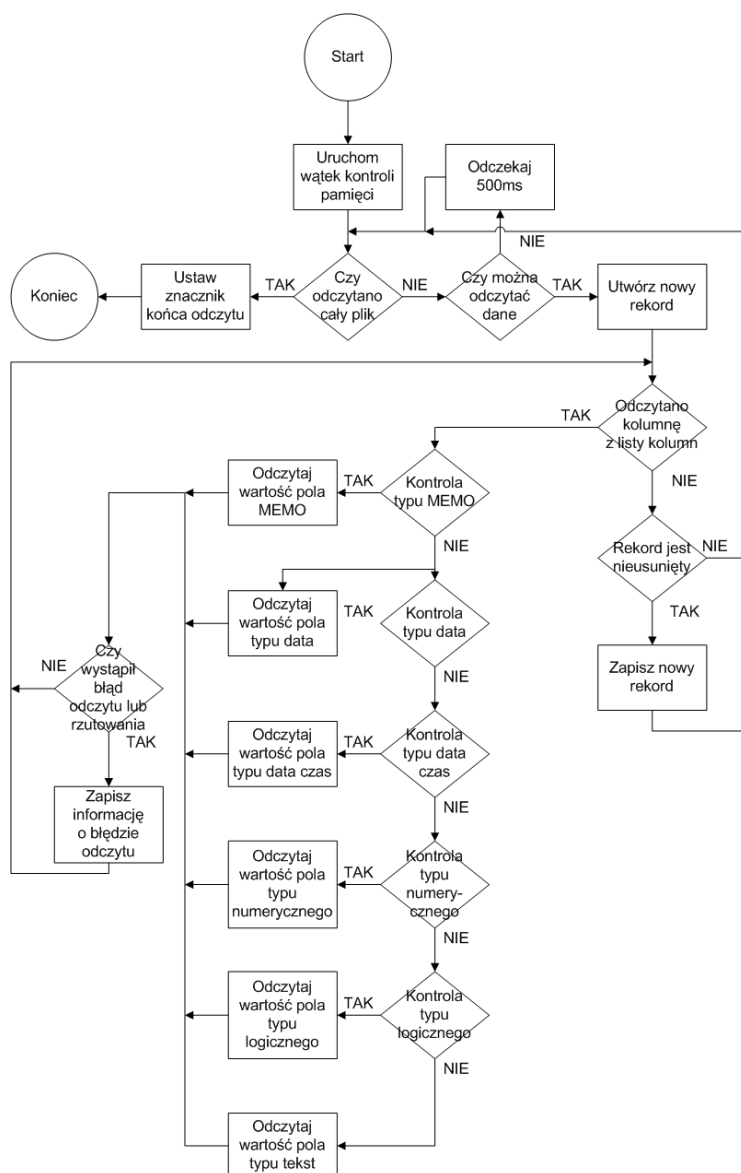


Rys. 1. Algorytm wątku odczytującego dane z pliku

Fig. 1. The algorithm of the thread of reading the data from the file

### 3.3. Algorytm odczytu danych z pliku Visual FoxPro.

Algorytm (rys. 2) sprawdza typ danych dla każdej kolumny i próbuje dokonać konwersji odczytanych danych na wskazany typ. Jeżeli rzutowanie nie powiedzie się, zapisywany jest błąd informujący o numerze rekordu i nazwie kolumny, której dotyczy. Kontrolę nad odczytem danych sprawuje wątek kontroli pamięci. Wątek ten ma na celu sprawdzanie zajętej pamięci komputera wykonującego import danych. Kontrola wiąże się z zachowaniem równowagi pomiędzy wątkiem odczytu i zapisu, zapobiega to zajmowaniu dużej liczby pamięci w przypadku, gdyby wątek zapisu był dużo wolniejszy od wątku odczytu.

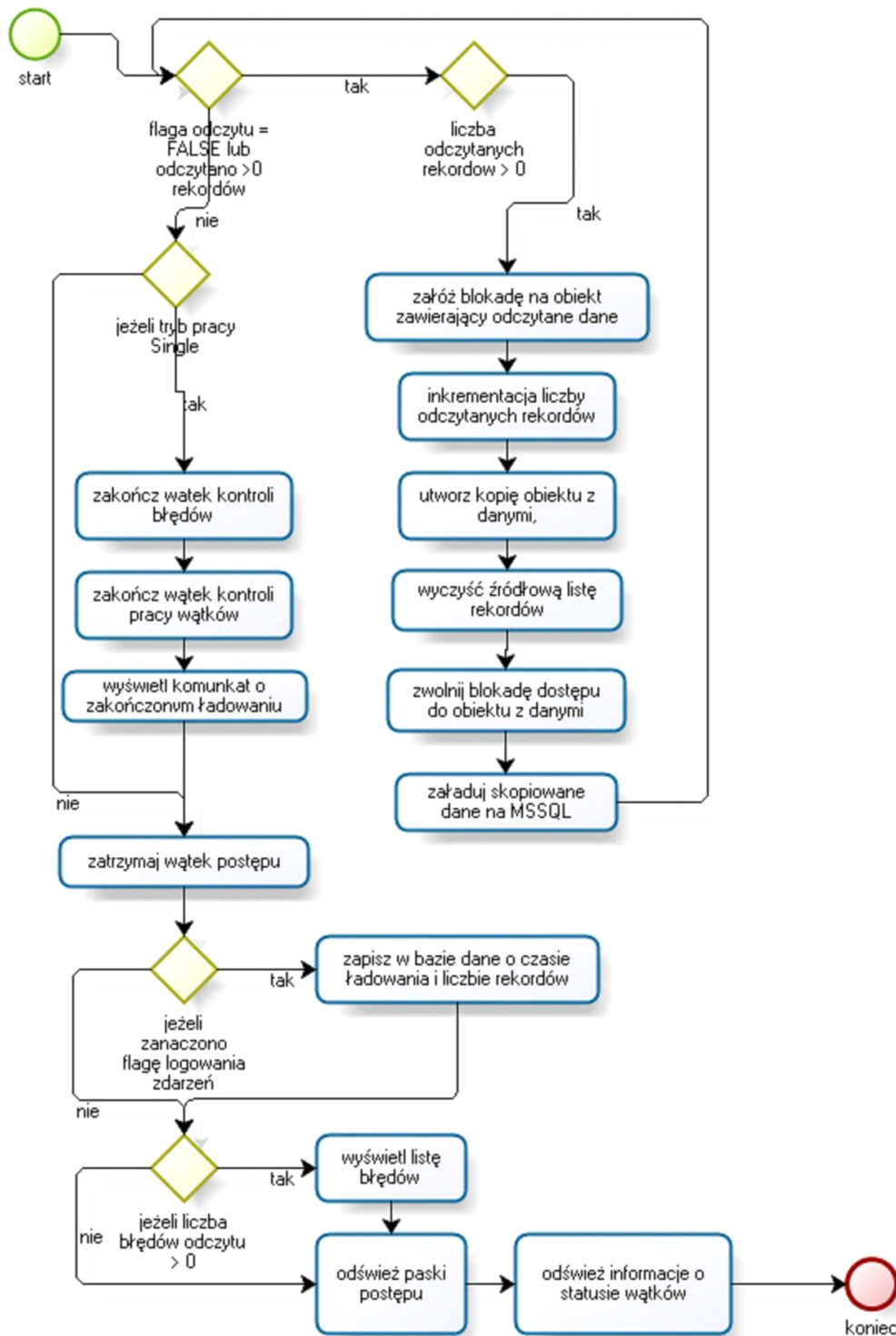


Rys. 2. Algorytm odczytu danych z pliku Visual FoxPro

Fig. 2. The algorithm of reading the data from Visual FoxPro file

### 3.4. Wątek zapisu

Algorytm wątku zapisującego dane na serwerze SQL (rys. 3) jest drugim najważniejszym algorytmem omawianej aplikacji. Chcąc rozpocząć pracę, wątek zapisu sprawdza flagę pracy wątku odczytującego dane lub sprawdza, czy wątek ten posiada odczytane dane, które nie zostały jeszcze zapisane na serwerze SQL. Jeżeli wątek odczytu posiada niezapisane jeszcze rekordy, wątek zapisu zakłada blokadę na obiekt zawierający listę odczytanych rekordów. Po założeniu blokady wątek odczytu wstrzymuje odczytywanie nowych danych.



Rys. 3. Algorytm wątku zapisującego dane na SQL Server

Fig. 3. The algorithm of the thread of writing down the data to SQL Server

Wątek zapisu tworzy kopię listy rekordów, tak aby mógł importować dane, równocześnie pozwalając na odczytywanie nowych danych z plików Visual FoxPro. Po utworzeniu kopii rekordów następuje usunięcie wszystkich elementów źródłowej listy rekordów, zwolnienie blokady dostępu i rozpoczęcie procesu zapisywania danych na serwer SQL.

Cały proces jest wykonywany tak długo, jak długo flaga odczytu danych ma wartość 'true' lub obiekt z danymi posiada liczbę rekordów większą od zera. Podczas importu danych na serwer, wątek odczytu równocześnie odczytuje dane, zapelniając bufor danych.

Po załadowaniu wszystkich rekordów przez wątek zapisujący i zakończeniu odczytu przez wątek odczytujący, aplikacja kończy wątek kontroli błędów oraz wątek kontroli statusu wątków (podrozdz. 3.5), a następnie wyświetlany jest komunikat o zakończeniu procesu ładowania danych.

Kolejnym krokiem aplikacji jest zatrzymanie wątku postępu, po czym sprawdzane jest, czy informacja o importowanej tabeli powinna zostać zapisana w logu na serwerze SQL. Jeżeli użytkownik zaznaczył żądanie zapisywania informacji w logu, aplikacja dopisze w logu informację o zapisanej tabeli, czasie zapisania oraz liczbie rekordów znajdujących się w pliku DBF. Następnie sprawdzana jest liczba błędów znalezionych podczas odczytu danych z plików. Jeżeli liczba błędów jest większa od zera, to nastąpi wyświetlenie tej listy.

Kończąc działanie wątku zapisu, odświeżane są wartości na paskach postępu oraz informacje o statusie wszystkich wątków aplikacji.

### **3.5. Kontrola poprawności procesu importu danych**

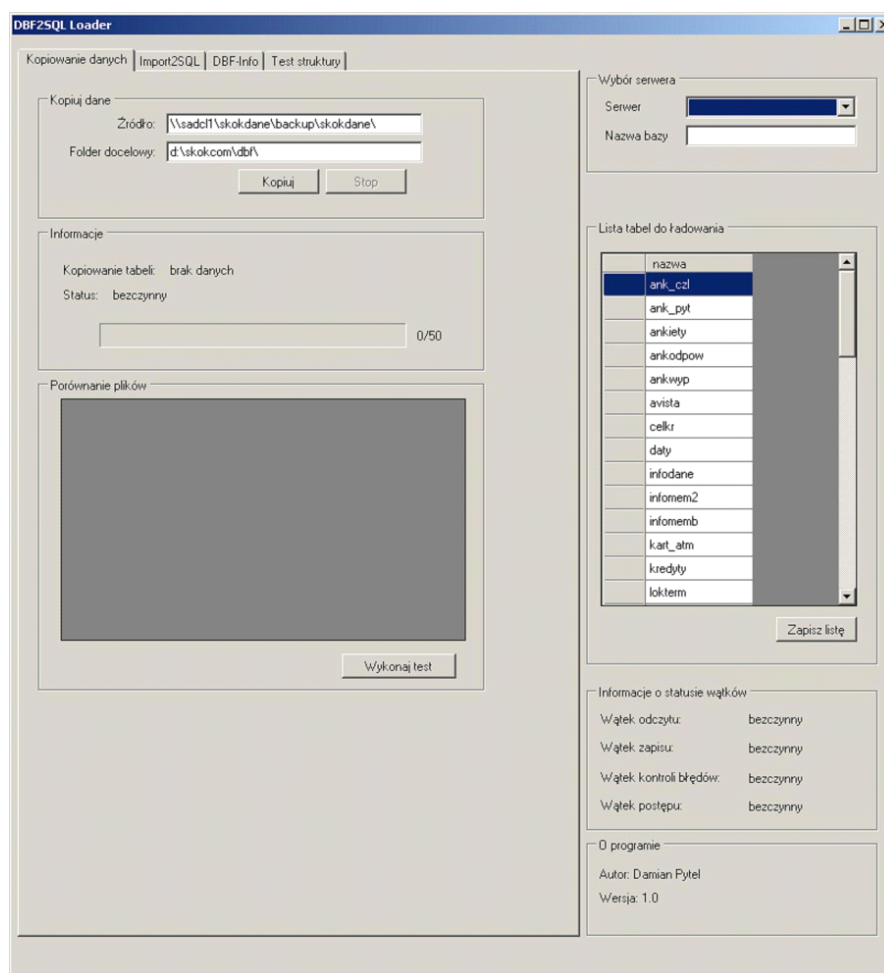
Dla zapewnienia większej kontroli nad działaniem aplikacji wprowadzono także wątek kontrolujący statusy wątków. Odpowiada on za nadzorowanie pracy wszystkich wątków aplikacji. Kolejny algorytm zapewniający bezpieczeństwo importu danych to kontrola błędów odczytu. Algorytm ten jest odpowiedzialny za wyłapywanie błędów pojawiających się podczas odczytu danych z pliku. Każdy błąd zapisywany jest kolejno na liście i wyświetlany w logu zdarzeń.

Jeszcze jeden ważny aspekt kontroli jest realizowany przez algorytm porównania struktury pliku DBF oraz tabeli na serwerze SQL. Algorytm ten znajduje swoje zastosowanie przy aktualizacjach struktury w plikach DBF. Działanie algorytmu obejmuje sprawdzanie zgodności typów danych, długości typów tekstowych oraz numerycznych. Algorytm przewiduje brak obecności pola w tabeli SQL, podczas gdy jest ono obecne w strukturze pliku DBF.

## **4. Aplikacja realizująca import danych oraz jej wydajność**

Do implementacji zaprezentowanych algorytmów (rozdz. 3) i napisania aplikacji importującej dane, zostały wybrane platforma .Net 3.5 i język C#. Aplikacja ta będzie mogła współpracować zarówno z Microsoft SQL Server 2005 w wersji 32-bitowej, jak i z SQL Server 2008 w wersji 64-bitowej. Aplikacja ta (rys. 4) umożliwia pracę w trybie Single, z jedną wybraną tabelą, lub w trybie Multi, z całym zestawem tabel. Algorytm importu danych w trybie

Multi tak naprawdę wykonuje wielokrotny import danych w trybie Single. Po wykonaniu importu danych (dla wszystkich tabel z listy) algorytm oczekuje na zakończenie wątków zapisu (podrozdz. 3.4) i odczytu (podrozdz. 3.2), kończy wątek kontroli błędów (podrozdz. 3.5) oraz wątek postępu, następnie wyświetla komunikat informujący o zakończeniu importu.



Rys. 4. Poglądowy widok aplikacji importującej dane

Fig. 4. The demonstrative view of the application of the data importing

Dla sprawdzenia wydajności napisanej aplikacji przeprowadzone zostały odpowiednie testy. Zostały one wykonane na komputerach o następującej konfiguracji:

K1 – komputer1:

- SQL Server 2005 32-bitowy.
- System operacyjny: Microsoft Windows Server 2003 R2 Standard Edition SP 2.
- Procesor Intel Xeon 5140 2.33 Ghz (4 rdzenie).
- 4 GB Ram.

K2 – komputer2:

- SQL Server 2008 64-bitowy.
- System operacyjny Microsoft Windos Server 2003 R2 Standard x64 Edition SP 2.



- Procesor Intel Xeon 5140 2.33 Ghz (4 rdzenie).
- 8 GB Ram.

Dla tych dwóch komputerów przeprowadzono testy w seriach po 5 razy dla 3 zestawów danych o zupełnie odmiennej strukturze:

1. Tabela nr 1 – rozmiar 444 MB, 10 kolumn, ponad 9,7 miliona rekordów.
2. Tabela nr 2 - rozmiar 46 MB, 149 kolumn, ponad 51 tysięcy rekordów.
3. Wszystkie tabele – 50 tabel, rozmiar 3,37 GB.

Tabela nr1 reprezentuje tabele o małej liczbie kolumn i dużej liczbie rekordów, natomiast tabela nr2 jest tabelą o dużej liczbie kolumn i stosunkowo małej liczbie rekordów. Uzyskane wyniki (tabela 1) są następujące:

Tabela 1

## Wydajność procesu importu danych

Zestaw danych:	Test nr:	K1- SQL Server 2005 32-bitowy		K2- SQL Server 2008 64-bitowy	
		Czas ładowania [mm:ss]	Średnia prędkość importu [MB/s]	Czas ładowania [mm:ss]	Średnia prędkość importu [MB/s]
Tabela nr1	1	04:07	1,801	05:51	1,267
	2	04:04	1,823	05:27	1,360
	3	04:10	1,779	05:33	1,336
	4	04:06	1,808	05:31	1,344
	5	04:06	1,808	05:34	1,332
<b>Średnia:</b>		<b>04:06</b>	<b>1,804</b>	<b>05:35</b>	<b>1,328</b>
Tabela nr2	1	00:17	2,745	00:20	2,333
	2	00:16	2,916	00:22	2,121
	3	00:17	2,745	00:20	2,333
	4	00:16	2,916	00:20	2,333
	5	00:17	2,745	00:20	2,333
<b>Średnia:</b>		<b>00:17</b>	<b>2,813</b>	<b>00:20</b>	<b>2,291</b>
Wszystkie tabele	1	37:57	1,514	43:09	1,332
	2	38:09	1,506	43:41	1,315
	3	38:02	1,511	43:38	1,317
	4	37:57	1,514	43:41	1,315
	5	38:41	1,485	43:21	1,33
<b>Średnia:</b>		<b>38:09</b>	<b>1,506</b>	<b>43:33</b>	<b>1,322</b>

Analizując uzyskane wyniki, widzimy, że czasy importu na obu komputerach są porównywalne. Niewielka rozbieżność w czasach importu może wynikać z różnicy warunków obciążenia maszyn. Oba komputery są to maszyny produkcyjne, gdzie cały czas uruchomionych jest kilka instancji SQL Server 2005/2008. W ten sposób aplikacja była testowana w rzeczywistych warunkach swojej pracy. Z przeprowadzonych testów wynika, iż wydajność importu danych zależna jest od struktury tabel importowanych na SQL Server. Szybciej importowane

są tabelę o większej liczbie kolumn, a mniejszej liczbie rekordów. Co najważniejsze, testy wykazały, iż aplikacja pracuje stabilnie, uzyskując powtarzalne wyniki wydajnościowe. Czas importu kompletu danych (50 tabel) wynosi ok. 40 minut i jest czasem w pełni zadowalającym w kontekście obecnego zapotrzebowania.

## 5. Podsumowanie

W artykule przedstawiono najważniejsze elementy metodologii umożliwiającej przeprowadzenie procesu importu danych z Visual FoxPro na MS SQL Server 2005 lub 2008. Na jej podstawie została zaimplementowana wielowątkowa aplikacja realizująca taki proces. Opracowane algorytmy umożliwiają import całej listy tabel lub pojedynczej tabeli na SQL Server. Jednocześnie opracowano metodę pełnej kontroli działania aplikacji (kontrola błędów, kontrola postępu odczytu/zapisu, kontrola pracy wątków). Wykonane przez użytkownika operacje są rejestrowane w celu weryfikacji poprawności wykonanych procedur. W związku z częstym występowaniem błędów danych w plikach DBA, opracowano algorytm sprawdzający zgodność odczytanych danych. Każde odczytane pole w rekordzie musi być zgodne z zadeklarowanym typem danych w nagłówku pliku DBF. Jeżeli nie można wykonać konwersji odczytanych danych na określony typ, taka informacja jest przechwytywana przez obsługę wyjątków i raportowana na formularzu aplikacji. Zabezpieczenie to jest wartościowym uzupełnieniem w stosunku do Provider-a VFP OLEDB, który po wykryciu niezgodności typu danych przerywa odczytywanie danych, natomiast omawiana aplikacja zastępuje błędnie odczytaną wartość wartością 'null'.

Ważnym elementem realizacji procesu importu danych było zapoznanie się ze strukturą plików DBF i FPT. Dopiero analiza struktury danych umożliwiła podjęcie decyzji o opracowaniu algorytmu pobierającego dane i importującego je na SQL Server.

Drugim ważnym elementem było opracowanie algorytmu umożliwiającego szybką i wydajną pracę. W tym celu zaprojektowany został algorytm wielowątkowej aplikacji, która w tym samym czasie wykonuje odczytywanie oraz zapisywanie danych.

Przeprowadzone testy wykazały, iż aplikacja działa w sposób stabilny, jednocześnie pracując wydajnie i niezawodnie. Czas importu kompletu danych (w obecnym zastosowaniu aplikacji), składającego się z 50 tabel (ponad 3 GB danych), wyniósł ok. 40 minut. Uzyskane średnie prędkości importu danych wyniosły około 1,5 MB/s.

Temat importu danych z Visual FoxPro jest często poruszonym problemem. Zrealizowana praca oraz przedstawione w tym artykule główne wyniki, mogą być dla szerszego grona odbiorców w znacznym stopniu pomocne w przeprowadzeniu poprawnego importu danych na MS SQL Server 2005 lub 2008.

**BIBLIOGRAFIA**

1. Microsoft: Table File Structure (.dbc, .dbf, .frx, .lbr, .mnx, .pjx, .scx, .vex), <http://msdn.microsoft.com/en-us/library/st4a0s68%28v=VS.80%29.aspx>.
2. Pytel D.: Aplikacja przeprowadzająca proces importu z bazy danych Visual FoxPro do MSSQL 2008. Praca dyplomowa magisterska, Politechnika Śląska, Wydział AEiI, Gliwice 2010.
3. DBConvert for FoxPro & MS SQL, <http://dbconvert.com/convert-foxpro-to-mssql-pro.php?DB=7>.

Recenzent: Prof. dr hab. inż. Mieczysław Muraszkiwicz

Wpłynęło do Redakcji 31 stycznia 2011 r.

**Abstract**

This paper presents the most important elements of methodology enabling execution of process of the import of data from Visual FoxPro database to MS SQL Server 2005 or 2008. Multithreaded application realizing such process was implemented using this methodology. The method of the full control of working of the application worked out (the control of errors, the control of the progress of the reading / writing, the control of the work of threads). Executed by the user operations are recorded in the aim of the verification of correctness of executed procedures. Because of frequent occurrence of data errors in DBA files, the algorithm verifying compatibility of read data is worked out.

Important element of realization of the import of data was the introduction about structure of DBF and FPT files. The analysis of the structure of data just made possible to create the algorithm of importing data from Visual FoxPro database to MS SQL Server.

Second important element was the study of the algorithm enabling the fast and effective work. To do this, the algorithm of the multithreaded application was created, which in the same time executes reading and recording of data.

Conducted tests demonstrated that the application works in the stable way, working efficiently and unfailingly. The time of import of set of the data (in the present use of application) consisting 50 tables (over 3 GB of the data) were approximately 40 minutes. The average speeds of import of the data were about 1,5 MB / s.

The subject of import of the data from Visual FoxPro is often discussed problem. Realized work and introduced main results in this article, can be for many recipients helpful, in the correct import of the data to MS SQL Server 2005 or 2008.

### **Adresy**

Robert BRZESKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, robert.brzeski@polsl.pl.

Damian PYTEL: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, damian@asp-net.pl.

Piotr BAJERSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, piotr.bajerski@polsl.pl.