

### **Co zrobić z programami "To zależy" oraz "Niezgodne"?**

DM2000 nie może określić, czy właściwie zainstalowałeś lub używałeś procedur obsługi i zapisu daty programu 'wrażliwego' na datę. Trzy przykłady służą ilustracji tego punktu i wytłumaczeniu symbolu: "To zależy." (It depends)

1. Arkusze kalkulacyjne. Większość obecnych arkuszy kalkulacyjnych (Excel, Lotus itp) wykonuje poprawne obliczenia dat przy założeniu, że wybrałeś prawidłowe, 4-cyfrowe ustawienia daty dla arkusza. Jeśli używasz 2-cyfrowego zapisu daty, procedury obliczające datę często uznają 00 za rok 1900. Nie możemy zagwarantować, że wybór opcji, którego dokonałeś jest prawidłowy.
2. Języki programowania i kompilatory. Większość z nich (Visual Basic, Delphi, Foxpro, itp.) wykona poprawne obliczenia daty przy założeniu, że wybrałeś odpowiednią, 4-cyfrową formę zapisu daty w tworzonym oprogramowaniu. Nie możemy zagwarantować, że wybór opcji, którego dokonałeś jest prawidłowy.
3. Nawet gdy dokonałeś prawidłowego wyboru używanego formatu daty, musisz dodatkowo wprowadzić i zapisać rok w postaci 4-cyfrowej. Jeśli używasz formatu roku RRRR, lecz wpisujesz tylko 87, 96 bądź 00 operacje obliczania daty mogą dać nieprawidłowe wyniki. DM2000 nie weryfikuje Twoich danych, w związku z tym nie może on określić, czy wybrałeś prawidłową opcję. Program DM2000 może Ci pomóc w prawidłowym ustawieniu parametrów oprogramowania.

### **Co możesz zrobić, jeśli DM2000 znajdzie jakiś program o statusie "To zależy".**

W prawie wszystkich programach o statusie "To zależy" występuje ten sam problem i prawie wszystkie mogą być poprawione w ten sam prosty sposób. Wiele aplikacji o tym statusie przetwarza lata zapisane w postaci 4 cyfr prawidłowo, ale pozwalają ustawić dwucyfrowy zapis roku, który jest potem nieprawidłowo przetwarzany w przypadku roku 2000 i następnych.

#### ***Stwórz kopię bezpieczeństwa swoich danych zanim przejdziesz dalej!***

Rozpocząć należy od właściwego ustawienia parametrów daty w programie. Następujące trzy warunki muszą zostać spełnione, aby dało się to przeprowadzić możliwie bezboleśnie:

1. Aplikacja przetwarza lata zapisane w formacie 4-cyfrowym prawidłowo,
2. Możesz dokonać zamiany lat na format 4-cyfrowy za pomocą opcji programu,
3. Oprogramowanie może automatycznie poprawić istniejące daty ewentualnie pytając użytkownika w niejasnych sytuacjach.

To są 3 ważne wymagania. Jednakże, warto poeksperymentować aby zobaczyć, jak dużo może zostać naprawione wewnątrz istniejącego programu.

Niektóre programy pobierają datę i format daty z systemu operacyjnego, dlatego też może pomóc ustawienie długich i krótkich dat prawidłowo na poziomie systemu.

#### ***Stwórz kopię bezpieczeństwa systemu zanim przejdziesz dalej!***

W systemie Windows 95 użyj Start | Ustawienia | Panel Sterowania | Ustawienia Regionalne | Data i wybierz ustawienia daty, w którym rok jest reprezentowany przez RRRR. Jeśli tworzysz własne ustawienia upewnij się, że rok jest reprezentowany przez RRRR.

W systemie Windows 3.x użyj Panelu Sterowania | Międzynarodowe, w Krótkim Formacie Daty naciśnij Wiek (1990 90), aby ukazał się (x), W Długim Formacie Daty wybierz Czterocyfrowy zapis daty w prawym dolnym rogu.

Bądź ostrożny, niektóre programy są napisane do tego stopnia niepoprawnie, że jeśli system operacyjny pierwotnie był przygotowany na używanie dwucyfrowego zapisu roku a ty zmienisz to na czterocyfrowy zapis, program może w ogóle się nie uruchomić lub zacznie przekazywać błędne wyniki. Jest to częstsze w programach pisanych na zamówienie niż tych zakupionych w sklepie.

**OSTRZEŻENIE:** Jeśli pracujesz w sieci nie dokonuj żadnych powyższych zmian bez porozumienia z administratorem sieci.

### ***Rozpoznanie.***

Możesz nam pomóc zidentyfikować oprogramowanie posiadające status **Nie wiadomo** lub **Nierozpoznany Program**. Kiedy czytasz jeden z raportów na ekranie widoczny jest odpowiednio przycisk Zasygnalizuj status (**Indicate Status**) bądź Zidentyfikuj Produkty (**Identify Products**).

### **Co możesz zrobić, jeśli DM2000 znajdzie jakiś program Niezgodny z rokiem 2000?**

Nie istnieje uniwersalna odpowiedź dla wszystkich zagadnień oraz wszystkich programów. Na początek nie należy kupować pakietów oprogramowania, których zgodność z rokiem 2000 nie jest zagwarantowana. Nie należy pisać programów, które nie poradzą sobie w roku 2000 i później. Nie powinno się tworzyć nowych procedur (arkuszy kalkulacyjnych, baz danych itp), które nie używają oraz nie zapisują roku za pomocą 4 cyfr. Możliwości zostały przedstawione poniżej:

- **Uaktualnienie** – Jeśli oprogramowanie, którego używasz nie jest aktualną wersją weź pod uwagę uaktualnienie do wersji najnowszej. Zwykle uaktualnienie przekształci stare dane, a niektóre mogą nawet przekształcić starą strukturę zapisu daty na nową. Uaktualniaj jednak tylko wtedy, kiedy wiesz, że nowa wersja jest kompatybilna z rokiem 2000.
- **Zakup innego pakietu oprogramowania** – Można zakupić oprogramowanie od innego producenta i zastąpić nim aktualnie używany pakiet oprogramowania. Dla przykładu, możesz kupić nowy pakiet finansowy, który jest zgodny z rokiem 2000 i zacząć go używać już teraz. Możesz zakupić pakiet, aby zastąpić wykonaną na zamówienie lub napisaną we własnej firmie aplikację. Upewnij się że to, co kupujesz jest zgodne z rokiem 2000.
- **Naprawa** – Można naprawić istniejące oprogramowanie i datę do wymaganego poziomu zgodności z rokiem 2000. W każdym systemie i dla każdej aplikacji, które umieścisz w tej kategorii będziesz musiał wybrać sposób naprawy, aby osiągnąć wymagany poziom zgodności z rokiem 2000. Zobacz [“Są tylko trzy sposoby naprawy”](#) aby dowiedzieć się czegoś więcej na ten temat.

- **Napisanie od nowa** – Jeśli w Swojej pracy używasz oprogramowania pisanego na zamówienie, możesz napisać go od nowa lub zlecić jego napisanie w całości. Jest oczywiste, że program napisany od nowa musi być zgodny z rokiem 2000. Może to być bardzo czasochłonne jak również drogie i wciąż istnieje niebezpieczeństwo wynikające z chęci “rozbudowania” nowej wersji. To może spowodować jeszcze większe wydatki i pochłonięcie jeszcze większej ilości czasu.

W następujących dwóch możliwościach albo oddasz kontrolę nad aplikacjami komuś innemu albo nie będziesz używał tych aplikacji w ogóle.

- **Serwis** – Oddanie systemu do serwisu. Serwis musi mieć system zgodny z rokiem 2000, aby było to w ogóle warte zachodu. Systemy, które mogą zostać oddane do serwisu to z reguły mało strategiczne, powtarzalne pospolite systemy (np. oprogramowanie listy płac, księgowość itp).
- **Eliminacja** – Zlikwidowanie aplikacji. Oznacza to, że funkcjonalność, jaką daje aplikacja przestanie istnieć. We wszystkich innych możliwościach funkcjonalność reprezentowana przez aplikację nadal istnieje w jakiejś formie. Decyzja ta jest właściwa tylko dla oprogramowania, które i tak miało zostać wkrótce wyeliminowane na skutek innych decyzji np. finansowych lub pracowniczych. Decyzja ta może również być spowodowana faktem, że nie ma już czasu aby naprawiać wszystkie systemy.
- **Nie robienie niczego** – Jeśli zdecydujesz się zignorować sytuację, jest to Twoja decyzja. Chociaż jest to standardowy wybór, nie powinieneś na to pozwolić. Niech to będzie przemyślana decyzja. Wybranie tej opcji zależy od twojej wiedzy na temat systemu oraz tego, jak bardzo pewny jesteś tego, że on nie zawiedzie albo kiedy zawiedzie, uszkodzenie będzie znośne i będziesz mógł wrócić do stanu poprzedniego.

### Istnieją tylko 3 możliwości naprawy.

Poniższe informacje przeznaczone są dla tych, którzy pragną sami naprawić błędy związane z rokiem 2000.

Trzema możliwościami naprawy są **Zmiany algorytmu (okienkowanie)**, **Rozszerzanie oraz Pakowanie**. Istnieją pomiędzy nimi różnice, lecz podstawowe zasady pozostają takie same. Mocno upraszczając można powiedzieć, że Rozszerzanie oraz Pakowanie (wytłumaczone poniżej) wymagają zmian w danych i w kodzie programu. Zmiany algorytmu (okienkowanie) wymaga jedynie zmian w kodzie programu.

**Zmiana algorytmu (okienkowanie):** Podczas modyfikowania algorytmu *nie* zmienia się struktur danych (plików, baz danych itp), formatu daty czy wartości opisujących datę. Zamiast tego, wszystkie daty są przetwarzane przez procedury zapisane w programie. Procedury te zawierają algorytmy, które odpowiednio ustalają stulecie. Data jest nadal ma format RR dla roku, nadal przechowuje się jedynie 2 cyfry, a rok 2000 jest reprezentowany przez 00. Prosty przykładem może być:

```
if RR < 40 then
    stulecie := 20
else
    stulecie := 19
```

Jest to tylko tymczasowa poprawka, lecz w zależności od zakresu dat w danej aplikacji możesz nie potrzebować niczego więcej. Wiele napisano o algorytmach obliczania dat, są również programiści specjalizujący się w tych algorytmach. Zalecamy skontaktowanie się z dostawcą lub producentem

kompilatora i zapytanie o propozycje dotyczące takiego algorytmu. Porozum się z innymi twórcami oprogramowania w twoim rejonie. Być może oni mają jakieś interesujące sugestie.

### **Zalety**

Twoje bazy danych pozostają niezmiennione.

Jest ono 40%-60% tańsze od innych rozwiązań.

Można dzięki niemu odroczyć problemy w kilku systemach.

Niektóre systemy mogą zostać wymienione zanim problem tysiąclecia znów stanie się problemem.

### **Wady**

Jest to rozwiązanie tymczasowe.

Nie jest uważane za prawidłowe przez purystów programowania

Może ono wymagać napisania wielu dziwnych (egzotycznego) fragmentów kodu, które mogą wkrótce potem zostać usunięte.

Dziwne fragmenty kodu mogą wprowadzać błędy działania programu. Indeksowanie i sortowanie może być skomplikowane.

**Rozszerzanie:** W przypadku naprawy przez rozszerzanie zamienia się każdy rok w dacie z zapisu 2 cyfrowego – RR na zapis 4 cyfrowy – RRRR.

### **Zalety**

Nie są potrzebne egzotyczne algorytmy.

Większość konwersji może zostać wykonana za pomocą narzędzi programowych (zależnie od języka programowania).

Nie będziesz musiał się martwić aż do 01/01/10000. Stanowi to dobrą okazję, aby użyć standardowych procedur obliczania daty.

Wymaga relatywnie łatwego do napisania programu do zmian struktury danych.

Indeksowanie i sortowanie działa bez problemów.

### **Wady**

Może mieć znaczący wpływ na rozmiar składowanych danych.

Wymaga, aby każdy program mający dostęp do struktur danych został naprawiony - nawet taki, który nie korzysta z dat.

Każdy program musi zostać przetestowany. Może to zwiększyć ilość tymczasowych łąć.

Każda procedura obsługująca datę musi zostać zmieniona.

**Pakowanie:** W naprawie przez pakowanie *nie* zmienia się długości pól zawierających rok, lecz zmienia się znaczenie danych zapisanych w polach lat (ich format). Jako prosty przykład możesz zmienić format zapisu pola rok na 'packed decimal', 'long integer' lub manipulować bitami. Wiele napisano o alternatywnym zapisie daty w postaci spakowanej i są programiści specjalizujący się w tym zagadnieniu. Zalecamy skontaktowanie się z dostawcą lub producentem kompilatora i zapytanie o propozycje dotyczące takiego algorytmu. Porozum się z innymi twórcami oprogramowania w twoim rejonie. Być może oni mają jakieś interesujące sugestie.

### **Zalety**

Może być uznawane za prawidłowe i zgodne.

Nie zwiększa rozmiaru składowanych

### **Wady**

Może nie być uznane za prawidłowe i zgodne.

Może być tymczasowym rozwiązaniem.

danych.

Kod programu wymagany do zmiany wartości daty w polach roku jest relatywnie prosty do napisania.

Część pracy może zostać wykonana za pomocą standardowych narzędzi.

Nie trzeba poprawiać programów, które nie odwołują się do pól roku.

Nie może wykorzystywać standardowych procedur przetwarzania dat. Może wymagać napisania dużych fragmentów skomplikowanego kodu.

Może wymagać napisania dużych ilości nieskomplikowanego kodu

Trzeba poprawić każdą procedurę obsługi daty. Indeksowanie i sortowanie może być wyjątkowo skomplikowane.

**Rozwiązanie idealne:** Niestety takie rozwiązanie nie istnieje. Rozszerzanie jest uznawane przez wiele organizacji za rozwiązanie idealne, gdyż pozwala uporać się z problemem na zawsze. Możesz jednak nie mieć wystarczającej ilości miejsca ani czasu aby zastosować rozszerzanie. Okienkowanie może być wystarczającym rozwiązaniem dla twojego środowiska. Znajduje to zastosowanie głównie wtedy, kiedy zakres lat jest relatywnie mały (kilka lat lub mniej niż dziesięciolecie) lub gdy aplikacja przestanie być używana, zanim zostanie osiągnięta data 'okna' (tak jak w przykładzie lata 40te następnego wieku).

Pakowanie może być najmniej pożądane. Jediną jego zaletą nad rozszerzaniem jest to, że nie wymaga ono większej ilości miejsca. Wymaga za to znacznie bardziej złożonego kodu programu, a co za tym idzie jest znacznie trudniejsze w testowaniu poprawności.

Należy pamiętać, że nie trzeba decydować się na jeden typ rozwiązania dla wszystkich systemów. Rozważyć należy zalety każdego rozwiązania osobno dla każdego systemu oraz aplikacji.

### **Znajdź pomoc.**

Jeśli nie jesteś 'złotą rączką' zastanów się, czy nie lepiej poszukać pomocy u kogoś innego.