POLITECHNIKA KOSZALIŃSKA
Wydział Mechaniczny

PRACA DYPLOMOWA
MAGISTERSKA

Temat :

„Pakiet aplikacji do optymalizacji systemu Windows 9x i NT poprzez weryfikację i modyfikację parametrów konfiguracyjnych zapisanych w plikach rejestrów”

Promotor: prof.dr hab. inż. Wojciech Kacalak
Dyplomant: Przemysław Kleszczewski

Kierunek: ***Inżynierskie Zastosowania Komputerów***
Specjalność: ***Podstawy Systemów Komputerowych***

Najpiękniejszą rzeczą, jakiej możemy doświadczyć,
jest oczarowanie tajemnicą. Jest to uczucie, które stoi u kolebki
prawdziwej sztuki i nauki. Ten kto go nie zna i nie potrafi się dziwić,
nie potrafi doznawać zachwytu, jest martwy jak zdmuchnięta świeczka.

Albert Einstein

Szczególne podziękowania składam :
prof. dr hab. inż. Wojciechowi Kacalakowi,
mgr inż. Stefanowi Nawrockiemu
bez pomocy których praca ta nigdy by nie powstała,

oraz moim rodzicom za to że, zawsze wierzyli we mnie
i którym to wszystko zawdzięczam.

Spis treści

CEL PRACY 6

WPROWADZENIE 7

1 ARCHITEKTURA SYSTEMU WINDOWS 9X 8

Rys.1 Schemat budowy Windows 9x *8*

1.1 Sterowniki urządzeń 9

Rys.2 Schemat pośredniczenia sterowników pomiędzy urządzeniem a systemem *9*

1.2 Menedżer konfiguracji 10

Rys 3 Schemat Menedżera konfiguracji *10*

1.3 Rejestr a urządzenia Plug & Play 11

Rys 4 Faza boot-strappingu *12*

Rys 5 Graficzna reprezentacja uruchamiania Windows 9x zgodnie z PnP *13*

Rys 6 Sposób współdziałania Menedżera konfiguracji z innymi elementami systemu *14*

2 STRUKTURA REJESTRU 15

2.1 ANALIZA HIERARCHII REJESTRU: 16

3 RÓŻNICE POMIĘDZY REJESTREM SYSTEMU WINDOWS 9X I NT 16

3.1 TYPY PLIKÓW BINARNYCH W WINDOWS NT: 17

4 POWIĄZANIA PLIKÓW 18

4.1 Rozszerzenia plików z podkluczami definicji klasy 18

4.2 Podklucze definicji klas 19

4.3 Szybki podgląd 21

4.4 Obsługa plików nieznanych typów 21

4.5 Pozycje rejestru związane z menu kontekstowym 21

4.6 Statyczne elementy menu kontekstowego 22

4.7 Dynamiczne elementy menu 22

4.8 Identyfikatory CLSID 23

4.9 Podklucz InprocServer32 23

4.10 Podklucz ProgID 24

5 PLIKI REG 24

6 MODYFIKOWANIE REJESTRU Z POZIOMU SYSTEMU OPERACYJNEGO 26

Rys 7. Okno programu regedit.exe *27*

7 NAPRAWA USZKODZONEGO REJESTRU 28

7.1 Windows 9x 28

Uruchamianie systemu w trybie awaryjnym: *28*

Rys 8. Okno Panel Sterowania *28*

Rys 9. Okno Dodaj nowy sprzęt *29*

Odbudowa rejestru za pomocą programu regedit.exe *30*

Przywracanie kopii zapasowej rejestru *30*

Odtwarzanie kopii z dysków bezpieczeństwa *30*

7.2 Windows NT 31

Odtwarzanie systemu z nośników danych *31*

Odtwarzanie rejestru programem REGREST *31*

8 PROGRAMOWANIE OPERACJI NA REJESTRZE WINDOWS 32

8.1 Zarządzanie kluczami 33

8.2 Zarządzanie wartościami 35

8.3 Wyliczanie kluczy i wartości ... 37
8.4 Funkcje kopii zapasowych ... 39
8.5 Ochrona rejestru ... 40

9 PROGRAMOWANIE STANDARDÓW ... 41

Wymagania dotyczące systemu operacyjnego: ... *41*
Wymagania dotyczące 32- bitów ... *41*
Instalowanie ... *41*
Ogólne ustawienia programu ... 42
Rejestracja rozszerzeń plików ... 42
Rejestracja definicji plików ... 42
Rejestracja wspólnych komponentów ... 42
Rejestracja procedury deinstalacyjnej ... 42
Procedura deinstalacyjna ... *43*
Interfejs użytkownika ... *43*
OLE ... *43*
Obsługa UNC/LFN ... *43*
9.1 Dodawanie do rejestru ustawień programu ... 43
Ogólne ustawienia programu ... *43*
Rys 10. Dane ogólne programu Lotus Organizer 4.1 ... *44*
Preferencje użytkownika ... *44*
Rys 11. Dane szczegółowe programu Lotus Organizer 4.1 ... *45*
9.2 Integracja aplikacji z systemem Windows ... 45
Rys. 12. Podklucz definicji typu pliku. ... *46*
Rys. 13. Podklucz definicji pliku dla rozszerzenia MP3. ... *46*
Zaawansowane rozszerzenia powłoki ... *47*
Obsługa OLE ... *47*
9.3 Instalowanie wspólnych komponentów ... 47
Rys. 14. Pozycje podklucza SharedDLLs. ... *48*
9.4 Informacje o deinstalacji ... 48

10 FUNKCJE REGISTRY API DOSTĘPNE DLA PROGRAMISTÓW ... 49

- RegCloseKey() ... 49
- RegConnectRegistry() ... 49
- RegCreateKey() ... 50
- RegCreateKeyEx() ... 51
- RegDeleteKey() ... 53
- RegDeleteValue() ... 54
- RegEnumKey() ... 54
- RegEnumKeyEx() ... 55
- RegEnumValue() ... 56
- RegFlushKey() ... 58
- RegGetKeySecurity() ... 58
- RegLoadKey() ... 59
- RegNotifyChangeKeyValue() ... 60
- RegOpenKey() ... 61
- RegOpenKeyEx() ... 62
- RegQueryInfoKey() ... 63
- RegQueryMultipleValues() ... 64
- RegQueryValue() ... 65
- RegQueryValueEx() ... 66
- RegSetKeySecurity() ... 68
- RegReplaceKey() ... 68
- RegRestoreKey() ... 69
- RegSaveKey() ... 70
- RegSetValue() ... 71
- RegSetValueEx() ... 72
- RegUnLoadKey() ... 73

WNIOSKI 74

LITERATURA 76

DODATKI 77

Cel pracy

Praca ta powstała z myślą o poprawieniu niektórych parametrów systemu Windows 9x i NT w celu optymalizacji z punktu widzenia komfortu pracy z systemem. Ma ona na celu upowszechnienie wiedzy o rejestrze systemowym tak by przeciętny użytkownik miał możliwość wnoszenia poprawek i dostosowywania systemu do swoich upodobań i potrzeb. Za główny cel pracy postawiono sprawę skojarzenia plików, która jest bardzo ważna, gdyż każdy użytkownik prędzej czy później trafia na problemy z otwieraniem plików w niewłaściwych aplikacjach. Dziej się to na skutek przedefiniowania skojarzeń plików. System operacyjny pozwala na bardzo łatwe skojarzenie nowych typów plików z aplikacjami w których mają być otwierane ale w bardzo nikłym stopniu umożliwia zmianę już istniejących skojarzeń. Także nie informuje użytkownika o niepoprawnych skojarzeniach czy skasowanych typach plików. Brak tak podstawowych funkcji nakłonił do powstania aplikacji ułatwiającej i integrującej wszystkie funkcje zarządzania skojarzeniami plików. By dopełnić całości powstały też narzędzia pomocnicze takie jak edytor rejestru i system wnoszenia poprawek. Równie ważną sprawą poruszoną w tej pracy jest omówienie wszystkich funkcji udostępnianych dla programistów przez twórców systemu Windows. Zostały one zebrane w jedną całość i omówione tematycznie by umożliwić jak najłatwiejsze wyszukiwanie określonych funkcji w zależności od potrzeb. Nie pominięto także ważnego aspektu jakim jest integracja pisanych aplikacji z rejestrem systemowym.

Wprowadzenie

Obecna postać rejestru nie jest czymś nowym i nieznanym. Z bazą rejestrową systemu użytkownik spotkał już się w wersji Windows 3.1x. Dużym krokiem naprzód było natomiast scentralizowanie zarządzania zasobami systemu przez pojedynczą bazę. Została tak zaprojektowana aby stanowić fundament zarządzania całym systemem, siecią i użytkownikami . Historyczne podwaliny obecnie znanego rejestru mają swoje korzenie w Windows NT. Jego zadaniem było zastąpienie mnożących się plików .INI oraz konfiguracyjnych, które aplikacje, sieci i sterowniki urządzeń rozrzucały po całym dysku. Rejestr zgromadził wszystkie ustawienia w jednym miejscu zapewniając efektywną metodę zarządzania środowiskiem NT.

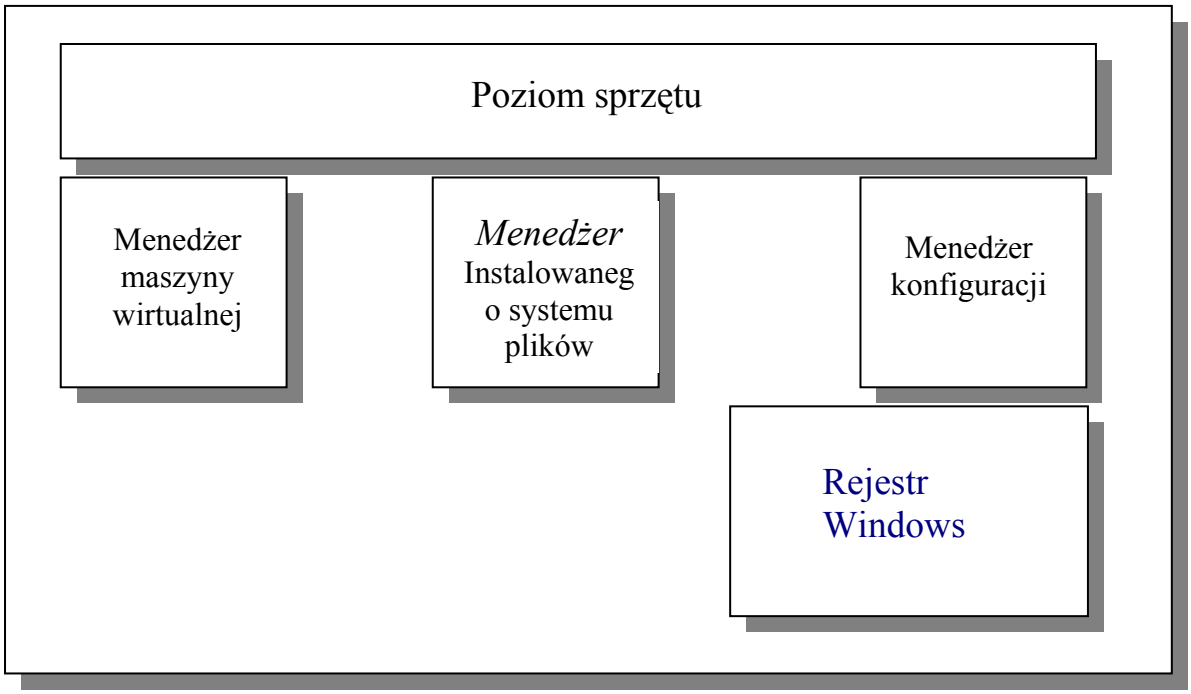
Administratorzy sieciowi zawsze dążyli do tego, aby zapewnić różnym użytkownikom posiadanie własnych konfiguracji. Sposób w jaki Windows pozwala użytkownikom dostosować Pulpit do własnych wymagań, staje się problemem, gdy w sieci znajduje się kilkaset lub kilka tysięcy użytkowników korzystających z różnych tematów, wygaszaczy ekranu, tapet itp. Podczas kolejnych aktualizacji lub napraw systemu administratorzy mieli by problem, by zapewnić każdemu użytkownikowi jego osobiste ustawienia systemu. Dzięki scentralizowaniu ustawień systemowych w Rejestrze administratorzy NT i Windows 9x mają wreszcie sposób pozwalający centralnie zarządzać komputerami.

Zarządzanie plikami INI stwarzało wiele innych problemów i niedogodności. O ile główne pliki systemu Windows nie sprawiały zazwyczaj większych kłopotów, to podczas instalacji i deinstalacji aplikacji system zostawał wypełniany coraz większą ilością plików INI. Powstawały problemy gdy zawarte w nich informacje stawały się nieaktualne albo rozrzucone po tak wielu katalogach, że trudno było odnaleźć ten, który wymagał akurat edycji. Na serwerach posiadających wielu użytkowników ich ilość dochodziła często do kilu tysięcy.

Rejestr Windows NT i Windows 9x zapewnia gromadzenie wszystkich ustawień w scentralizowanym archiwum. Rejestr korzysta tylko z dwóch plików : USER.DAT i SYSTEM.DAT. Oba pliki są plikami „ukrytymi" przechowywanymi w folderze systemu Windows. Są tworzone podczas instalacji systemu i dla każdego z nich powstaje kopia bezpieczeństwa. Kopie te również są plikami z atrybutem „ukryty" i mają odpowiednio nazwy : USER.DA0 i SYSTE.DA0. Podczas każdego startu systemu Windows automatycznie aktualizuje kopie bezpieczeństwa kopiując bieżące pliki Rejestru i nadając im rozszerzenia DA0. Teoretycznie dzięki temu zawsze znajduje się ostatnia konfiguracja, która zapewniała poprawne uruchomienie systemu.

1 Architektura systemu Windows 9x

Architektura systemu Windows ma postać modułową. Windows posiada 32 bitowe podsystemy drukowania, multimedialne i komunikacyjne, 32 bitowy system graficzny, nowy system plików oraz nowy rodzaj sterowników urządzeń.



Rys.1. Schemat budowy Windows 9x.

Do trzech najważniejszych elementów architektury Windows 9x należą:

- menedżer maszyny wirtualnej
- menedżer instalowanego systemu
- menedżer konfiguracji

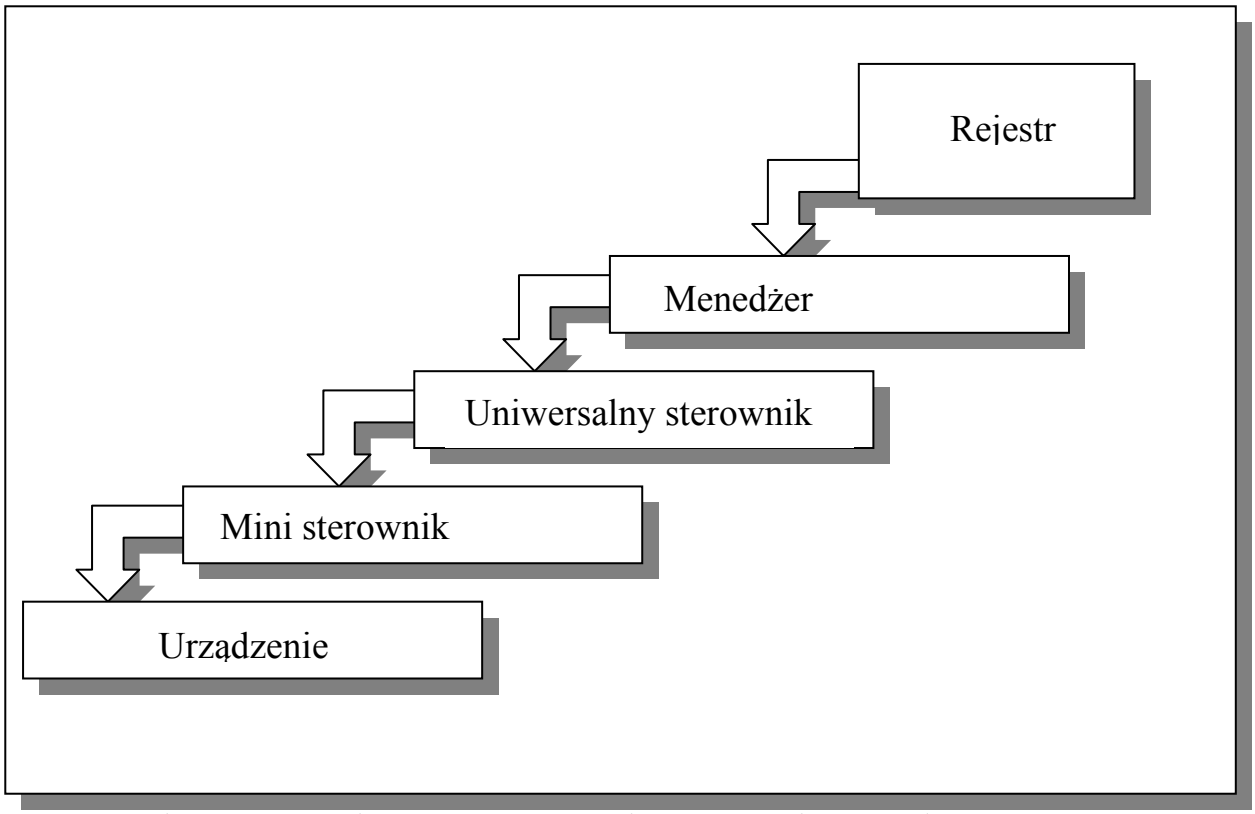
Menedżer maszyny wirtualnej zajmuje się tworzeniem i sterowaniem środowiskiem maszyny wirtualnej. Maszyny wirtualne to miejsca w pamięci operacyjnej systemu, które traktowane są jak niezależne komputery i w których działają aplikacje i procesy systemowe.

Menedżer instalowanego systemu plików (IFS Manager) decyduje o dostępie do urządzeń systemu plików i innych jego składników.

Menedżer konfiguracji jest natomiast głównym narzędziem pomagającym w konfiguracji sprzętu w Windows 9x

1.1 Sterowniki urządzeń

Sterownik urządzenia to program, który umożliwia komunikację pomiędzy urządzeniem a systemem operacyjnym, w tym przypadku Windows. W Windows 9x dostępny jest uniwersalny sterownik urządzenia przeznaczony do pracy z niektórymi urządzeniami. Zawiera on większość kodu niezbędnego do działania tych urządzeń w Windows. Sterowniki tego typu dostępne są np.: dla drukarek, modemów itp.



Rys.2. Schemat pośredniczenia sterowników pomiędzy urządzeniem a systemem.

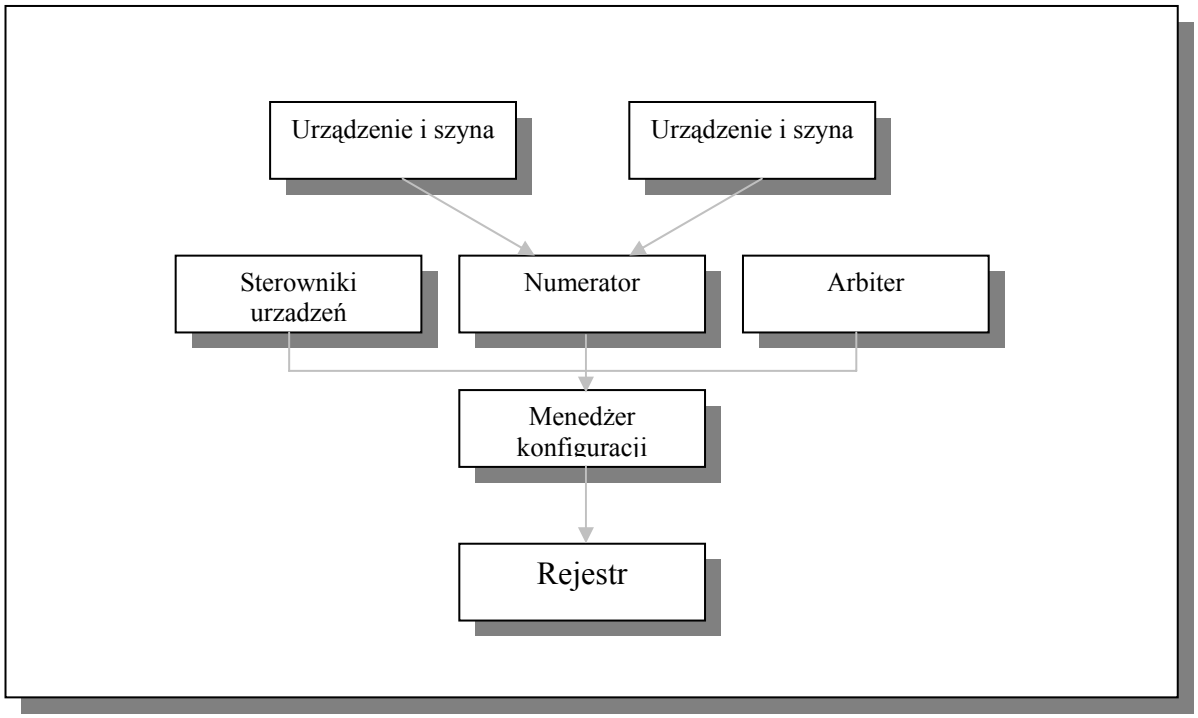
Główną przyczyną dla których Microsoft opracował uniwersalne sterowniki urządzeń było wyeliminowanie konieczności pisania sterowników dla każdego z nich. Producenci oprogramowania mogą korzystać z uniwersalnych sterowników pisząc tzw. mikro sterowniki . Jest to architektura pozwalająca na pisanie małych sterowników które zawierają dodatkowy kod charakterystyczny dla danego urządzenia , w większości przypadków wystarcza jednak sterownik uniwersalny.
Innym typem sterowników w Windows 9x są sterowniki wirtualne , które są oznaczone symbolem VxD. Są to 32 bitowe sterowniki pracujące w trybie chronionym procesora i tak zarządzają zasobami systemu , że jednocześnie z danego zasobu może korzystać więcej niż jedna aplikacja.

Dostępne są między innymi następujące rodzaje wirtualnych sterowników:

- sterowniki ekranu(VDD)
- zegary (VTD)
- drukarki (VPD)
- itp.

1.2 Menedżer konfiguracji

Menedżer konfiguracji jest swoistym strażnikiem pilnującym konfiguracji urządzeń i jest najważniejszym narzędziem do konfiguracji urządzeń Plug & Play. Jego zadaniem jest lokalizacja wszystkich urządzeń, umieszczanie ich na drzewie urządzeń i przydzielenie każdemu z nich odpowiedniej ilości zasobów.



Rys. 3. Schemat Menedżera konfiguracji.

Menedżer konfiguracji nie tylko zajmuje się przygotowaniem i konfigurowaniem urządzenia podczas startu ale monitoruje również stan komputera podczas pracy. Jeżeli podczas pracy jakieś urządzenie zostanie dodane lub usunięte menedżer konfiguracji dokonuje ponownej konfiguracji sprzętu. Powiadamia on o tym pracujące aplikacje i umieszcza odpowiednią informację w Rejestrze.

Aby prawidłowo identyfikować urządzenia, menedżer konfiguracji opiera się na kilku komponentach podrzędnych. Musi on też umieć zidentyfikować i współpracować z architekturą szyny danego komputera. Komponent podrzędny, zwany numeratorem, identyfikuje wszystkie urządzenia i szyny,

do których są one podłączone. Numerator szyny przekazuje następnie te informacje do menedżera, wraz z danymi o zasobach wymaganych przez urządzenie. Wreszcie menedżer konfiguracji przydziela urządzeniu przerwania IRQ, adresy portów I/O, parametry DMA i inne wymagane zasoby.

Podczas gdy numerator szyny zbiera dane o urządzeniach i przekazuje je menedżerowi konfiguracji, włącza się jeszcze jeden komponent podrzędny – arbiter zasobów. Informuje on menedżera, w jaki sposób ma on alokować zasoby dla poszczególnych urządzeń. Rozwiązuje on wszelkie konflikty powstające między urządzeniami żądającymi tych samych zasobów. Gdy dwa urządzenia żądają tego samego zasobu arbiter próbuje przydzielić dla jednego z nich inną wartość zasobu lub zgłasza konflikt urządzeń. Istnieje wtedy możliwość ręcznego przyznania zasobów za pomocą odpowiednich narzędzi Windows. Po przekazaniu przez arbitra i numerator szyny wszystkich informacji do menadżera konfiguracji dokonuje on konfiguracji sterowników urządzeń.

1.3 Rejestr a urządzenia Plug & Play

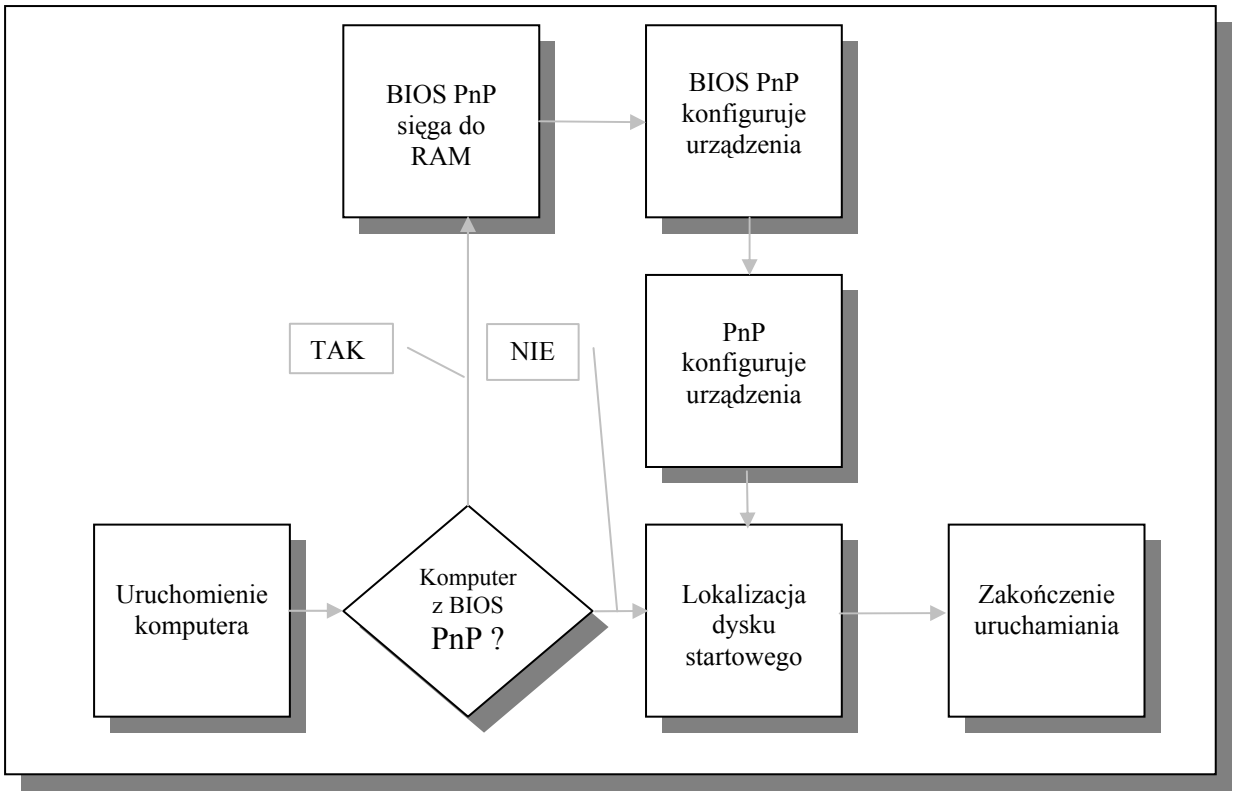
Plug & Play to rodzaj specyfikacji urządzeń, za pomocą której Windows może dokonywać automatycznej konfiguracji urządzeń. Dzięki systemowi w pełni zgodnemu z tą specyfikacją użytkownicy nie muszą się martwić o ręczne konfigurowanie instalowanych urządzeń. Sercem Plug & Play jest oczywiście baza rejestrowa systemu.

System Plug & Play tworzą trzy składniki :

- system operacyjny Plug & Play
- BIOS Plug & Play
- urządzenia Plug & Play

Nawet nie posiadające tych wszystkich elementów komputery (np. posiadające stary BIOS) mogą wykorzystywać niektóre możliwości systemu Plug & Play.

Podczas startu komputera i uruchamiania systemu Windows ten ostatni wykonuje następujące czynności:



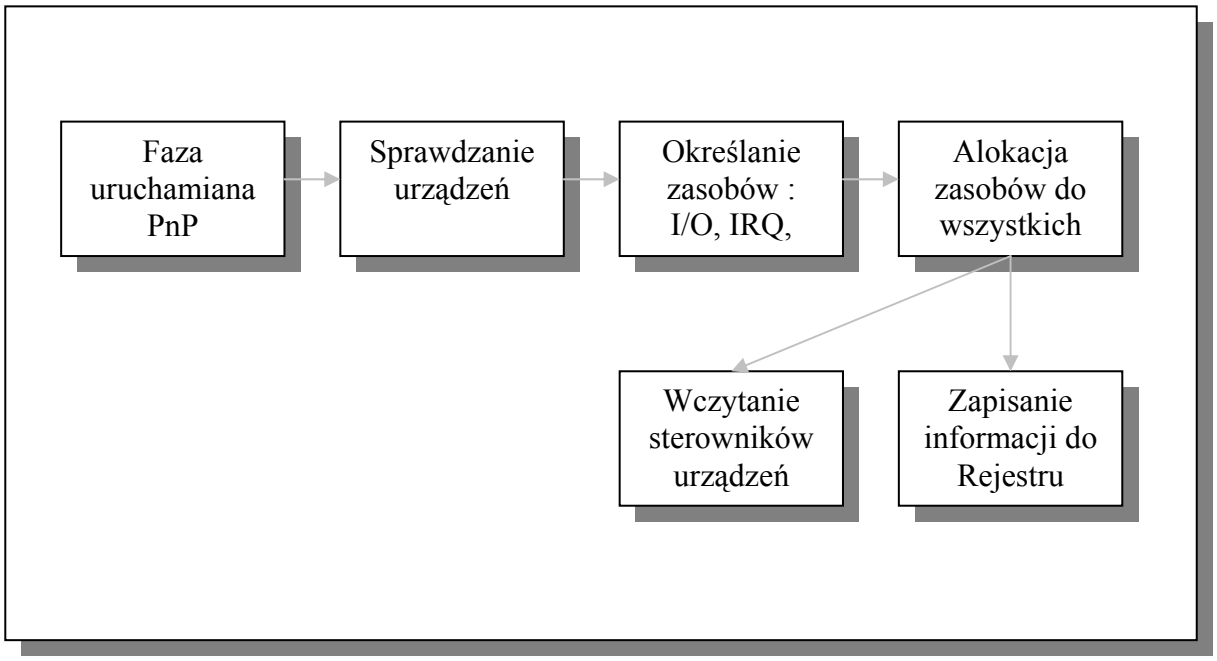
Rys. 4. Faza boot-strappingu.

Zaraz po uruchomieni komputera wykonywane jest najpierw jego samotestowanie (tzw. POST). Testy POST odbywają się pod kontrolą BIOS-u i mogą różnie przebiegać w zależności od komputera. Gdy pojawia się jakiś błąd, POST generuje unikalną serię dźwięków która jednoznacznie określa typ problemu ze sprzętem lub konfiguracją.

Testy POST wykonują następujące czynności:

- sprawdzenie pamięci systemowej
- sprawdzenie karty graficznej
- odszukanie dysku startowego

Zgodnie z rysunkiem nr 4 pierwszy test wykonywany przez POST ma stwierdzić czy BIOS jest typu Plug & Play. Jeżeli tak, to przed wczytaniem systemu operacyjnego wykonywanych jest jeszcze kilka procedur pomocniczych pobieranych z pamięci ROM.



Rys. 5. Graficzna reprezentacja uruchamiania Windows 9x zgodnie z PnP.

Jeżeli system jest wyposażony we wszystkie trzy składniki Plug & Play Windows wykona następujące czynności w celu skonfigurowania zainstalowanych urządzeń:

- BIOS identyfikuje urządzenia zamontowane na płycie głównej , takie jak szyna, dyski twarde, klawiatura, karta grafiki i tym podobne
- BIOS określa zasoby wykrytych urządzeń. (IRQ, I/O, DMA)
- Windows alokuje zasoby dla urządzeń W systemach z urządzeniami starszej generacji (nie Plug & Play) i kilkoma PnP wiąże się to z dużą ilością iteracji co znacząco wydłuża uruchamianie
- do Rejestru zostają zapisane informacje o konfiguracji. Ustawienia te można znaleźć w Rejestrze w kluczu HKEY_DYN_DATA\ConfigManager\Enum.
- do pamięci wczytywane są sterowniki urządzeń.

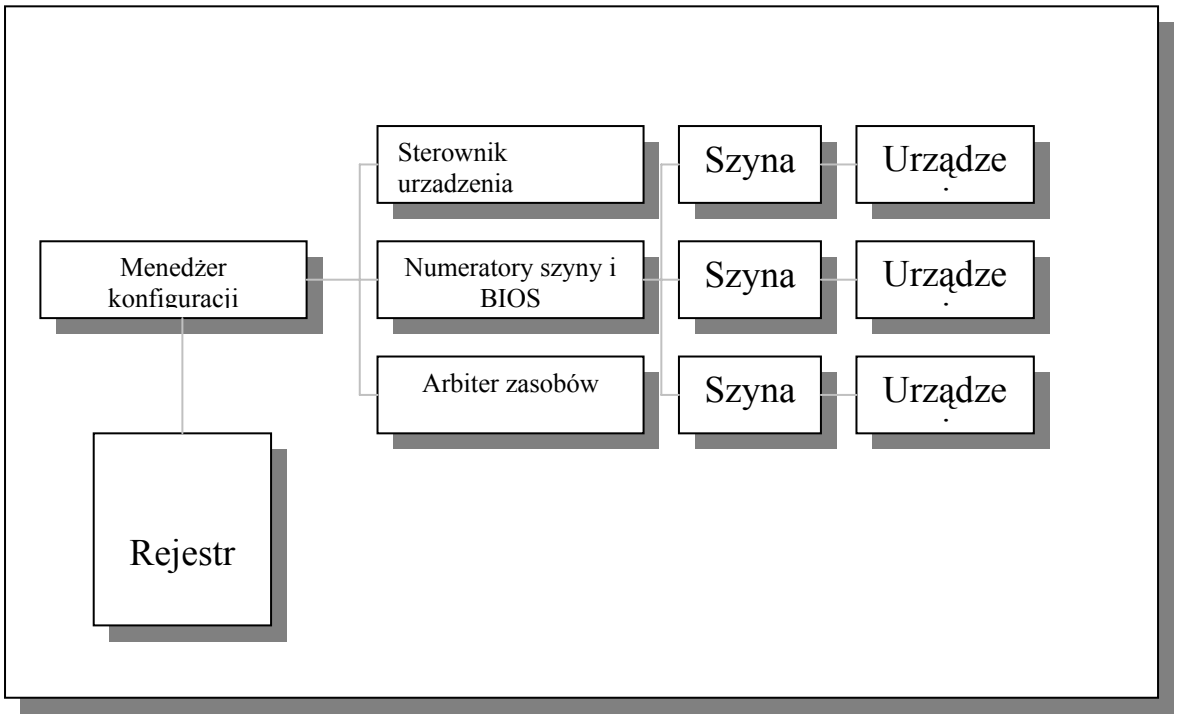
Aby umożliwić prawidłowe działanie Plug & Play system musi być elastyczny i musi pozwalać na dokonywanie zmian. Zmiany te zachodzą w następujących elementach:

- numerator szyny
- BIOS
- drzewo urządzeń i Rejestr
- system operacyjny Windows
- arbiter zasobów
- menedżer konfiguracji
- aplikacje

`Numerator szyny` buduje w systemie drzewo urządzeń. By urządzenia mogły efektywnie pracować, muszą znać szczegóły implementacyjne wszystkich możliwych architektur szyny. Dzięki temu mogą one identyfikować urządzenia, odczytywać ich wymagania co do zasobów oraz konfigurować je za pomocą menedżera konfiguracji. Dla każdego rodzaju szyny potrzebny jest inny numerator. Podczas określania zainstalowanego sprzętu posługują się one zarówno BIOS'em jak i zainstalowanymi sterownikami. Głównym zadaniem numeratora szyny jest przyporządkowanie każdemu urządzeniu unikalnego identyfikatora „ID". Musi być on jednoznaczny, co oznacza, że musi być on taki sam podczas każdego uruchamiania systemu.

`BIOS Plug & Play` zawiera rozszerzenia które umożliwiają konfigurację urządzeń podczas startu oraz zapewniają usługi dynamicznego wysłania komunikatów. Usługi te są niezbędne aby Windows mógł reagować na zmiany konfiguracji w czasie trwania sesji.

BIOS współpracuje z systemem na trzy sposoby:



Rys. 6. Sposób współdziałania Menedżera konfiguracji z innymi elementami systemu.

2 Struktura Rejestru

Rejestr zawiera trzy rodzaje obiektów:
- klucze
- wartości
- dane

Klucze Rejestru mogą zawierać jeden lub kilka innych kluczy lub wartości. Każdy klucz lub wartość musi mieć unikalną nazwę wewnątrz danego klucza lub klucza podrzędnego. Wielkość liter w nazwach kluczy jest zapamiętywana ale nie jest rozróżniana. Nazwa klucza może składać się z dowolnych znaków, które dają się wyświetlić, w tym spacji, podkreśleń, symboli. Nie może jednak zawierać znaków lewego ukośnika (\). Znak ten jest używany jako separator w ścieżce definiującej położenie elementu w Rejestrze. Klucze i klucze podrzędne zawierają jedną wartość o specjalnej nazwie (Domyślna)

Wartości opisują trzy elementy:
- typ danych
- nazwa wartości
- dana wartości

Typ danych jest typem reprezentującym sposób zapisu danych w kluczu. Rejestr wykorzystuje następujące typy danych:

1. Ciągi znaków: ciąg znaków to mająca zmienną długość tablica znaków, zakończona znakiem o kodzie zero. Mogą być w nim przechowywane słowa, zdania, ścieżki i inne teksty.
2. Binarne: wartość binarna to mający zmienną długość ciąg cyfr heksadecymalnych (0-9 i A-F). Każdy bajt jest reprezentowany przez dwie heksadecymalne cyfry.
3. DWORD : wartość DWORD (podwójne słowo) to pojedyncza liczba 32 bitowa reprezentowana przez osiem cyfr heksadecymalnych.
4. W systemie Windows NT wykorzystuje się jeszcze REG_BINARY, REG_DWORD, REG_EXPAND_SZ, REG_MULTI_SZ, REG_SZ.

2.1 Analiza hierarchii rejestru:

Rejestr zawiera następujące klucze główne :

- HKEY_CLASSES_ROOT zapewniający wsteczną zgodność obsługi OLE i DDE z Windows 3.x
- HKEY_CURRENT_USER zapewniający że, Windows i aplikacje zachowują się nie zależnie od tego jaki użytkownik jest obecnie zalogowany do systemu.
- HKEY_LOCAL_MACHINE przechowuje informacje o konfiguracji sprzętu i oprogramowaniu związanym z danym komputerem. Informacje te są charakterystyczne dla danej maszyny nie zależnie od tego który użytkownik wpisze się do sytemu.
- HKEY_USERS zawiera informacje o profilu bieżącego użytkownika
- HKEY_CURRENT_CONFIG wskazuje na klucz podrzędny HKEY_LOCAL_MACHINE\Config, zawierający informacje specyficzne dla danego komputera
- HKEY_DYN_DATA przechowuje informacje, które muszą przez cały czas znajdować się w pamięci RAM

3 Różnice pomiędzy Rejestrem systemu Windows 9x i NT

Windows NT nie przechowuje wartości rejestru w dwóch plikach. Zamiast tego, zawartość przechowywana jest w plikach reprezentujących poszczególne gałęzie rejestru rozpoczynające się na samej górze hierarchii. Pliki te znajdują się w folderze Windows NT w podkatalogu SYSTEM32\CONFIG. Żaden z plików gałęzi rejestru nie posiada rozszerzenia i jest powiązany z dwoma typami plików o rozszerzeniach SAV oraz LOG. Plik SAV stanowi kopię zapasową pliku gałęzi rejestru i jest tworzony podczas każdej pomyślnie zakończonej sekwencji startowej. Plik LOG natomiast zawiera wszystkie zmiany i modyfikacje, które system dokonuje w zawartości pliku gałęzi rejestru. Windows NT przechowuje konfiguracje specyficzne dla użytkowników w innym miejscu, podobnie jak Windows 9x z włączonymi profilami. Windows NT przechowuje dane konfiguracyjne każdego użytkownika w folderze PROFILES*USERNAME* , gdzie USERNAME to nazwa użytkownika podawana podczas logowania. Plik ten nosi nazwę NTUSER.DAT.

Rejestry Windows 9x i Windows NT są pod wieloma względami podobne, szczególnie gałęzie HKEY_CLASSES_ROOT i HKEY_USERS.
Różnią się natomiast tym że:

- WinNT przechowuje zawartość rejestru w wielu plikach podzielonych na gałęzie rejestru
- WinNT stosuje pełną ochronę, zarówno samych gałęzi rejestru jak i poszczególnych kluczy
- struktura klucza HKEY_LOCAL_MACHINE\System znacznie się różni

W dalszej części pracy gałęzie rejestru będą zwane ulami. Jest to termin przyjęty zwyczajowo przez Microsoft.

3.1 Typy plików binarnych w Windows NT:

Tabela 1. Znaczenie plików w folderach z ulami

Rozszerzenie pliku	Opis
Brak	Zawiera bieżącą kopię ula
.ALT	Zawiera zapasową kopię ula
.LOG	Zawiera rejestr zmian w ulu
.SAV	Zawiera zapasową kopię ula po instalacji (w trybie tekstowym)

Tabela 2. Powiązania plików binarnych uli

Ul	ALT	LOG	SAV
SAM		SAM.LOG	SAM.SAV
Security		Security.LOG	Security.SAV
Software		Software.LOG	Software.SAV
System	SYSTEM.ALT	SYSTEM.LOG	SYSTEM.SAV

Tabela nr 1 wyjaśnia znaczenie rozszerzeń plików zawierających dane o poszczególnych ulach systemu Win NT. Natomiast w tabeli nr 2 zamieszczone są powiązania każdego ula z plikami binarnymi.

4 Powiązania plików

Oprócz innych ustawień systemowych, systemy Win NT i 9x przechowują w rejestrze informacje o powiązaniach plików. Informacje o rozszerzeniach plików i instrukcjach OLE są zawarte w podkluczach klucza HKEY_CLASSES_ROOT. Każdy typ rozszerzenia jest reprezentowany w rejestrze przez osobny klucz zaczynający się od kropki (TXT, .BMP itd.). Wewnątrz podklucza rozszerzenia pliku zawarte są zwykle różne rodzaje danych. Mogą to być dane wskazujące na definicje klasy rozszerzenia pliku, nazywane czasem identyfikatorami aplikacji, oprócz tych danych klucz może zawierać informacje instruujące Windows jak powinien otworzyć czy też wydrukować dany dokument.

4.1 Rozszerzenia plików z podkluczami definicji klasy

Do podkluczy rozszerzeń plików, zwierających wskaźniki do podkluczy definicji klasy należą podklucze oraz wartości określające typ pliku powiązany z rozszerzeniem. Na przykład rozszerzenie pliku .TXT jest powiązane z plikiem tekstowym. Wskaźniki w podkluczach rozszerzeń plików nie określają jednak co powinno zostać zrobione z plikiem, wskazują jedynie na inny podklucz rejestru, informujący Windows jak powinien obsłużyć taki plik. Aby ułatwić wyszukiwanie, Windows przechowuje te podklucze w tym samym podkluczu co typy rozszerzeń.
By przybliżyć nieco ten temat należy posłużyć się przykładem. Typowa pozycja rejestru dla pliku JPG może zawierać następujące wartości:

(Domyślna) „jpegfile”
ContentType „image/jpeg”

Wartość (Domyślna) zawiera typ pliku powiązany z plikami o rozszerzeniu JPG. Wartość ContentType zawiera tzw. Typ MIME, czyli rozszerzenie poczty internetowej ogólnego przeznaczenia(ang. Multipurpose Intrnet Mail Extensions). Typ pliku z jakim jest skojarzone dane rozszerzenie to jpegfile i jest to nazwa klucza zawierającego informacje o aplikacji lub aplikacjach w których ten typ pliku można otworzyć, wydrukować itp.

Budowa takiego klucza dla pliku JPG może być następująca:

```
Jpegfile        DefaultIcon (Domyślna) „c:\Windows\System\jpeg.ico,0”
                Shell
                  └──Open (Domyślna) „Otwórz”
                        └── Command (Domyślna) „c:\Win\paint.exe %1”
```

Pozycja DefaultIcon zawiera ikonę wyświetlaną w Exploratorze dla danego typu pliku. Pozycja shell jest to podklucz zawierający podklucze tzw. reakcji w momencie dwukrotnego kliknięcia na danym typie pliku. Są to "Open" otwarcie pliku, "Print" wydruk pliku i inne zdefiniowane przez użytkownika. Jeżeli jest kilka pozycji to wykonywana jest domyślnie pierwsza. By skorzystać z innych opcji należy kliknąć prawym klawiszem i wybrać z menu kontekstowego właściwą reakcję. By nazwać poszczególne pozycje w menu kontekstowym należy wypełnić wartość (Domyślna) dla określonej reakcji. Na przykład:

Open – (Domyślna) „Otwórz”
Print – (Domyślna) „Drukuj”

Niektóre podklucze rozszerzeń plików nie zawierają wartości wskazujących na definicję klasy. Zamiast tego zawierają dane instruujące Windows jak otworzyć lub wydrukować zawartość pliku. Jest to jakby połączenie rozszerzenia pliku z typem pliku w kluczu rozszerzenia pliku.

Istnieje też grupa podkluczy skojarzonych z definicjami klas które zawierają dodatkowe informacje. Te dodatkowe informacje to najczęściej informacje o tym jak stworzyć nowy typ pliku. Informacje te są zawarte w podkluczu ShellNew\Command.

4.2 Podklucze definicji klas

Powodem dla którego w Windows zastosowano w podkluczu rozszerzenia wskaźnik do podklucza definicji klasy, była chęć zachowania pewnej elastyczności. Windows może przechowywać w pojedynczym kluczu wszystkie informacje o typie pliku , łącznie z jego rozszerzeniem i poleceniami służącymi do jego obsługi. Jednak w ten sposób rejestr byłby pełen powtarzających się informacji. Pojedyncza deklaracja klasy potrafi obsłużyć dwa lub więcej rozszerzeń. Windows obsługuje długie nazwy plików, gdzie rozszerzenia nie są ograniczone do trzech znaków. I tak na przykład pliki formatu JPG miewają rozszerzenia .JPG jak i JPEG. W ten

sposób dwa różne rozszerzenia wskazują na tą samą klasę. Innym powodem zastosowania odniesień jest przypadek gdy dana aplikacja obsługuje różne formaty plików. Przykładem może tu być procesor tekstu, przeglądarka plików graficznych itp. Wadą takiego sposobu obsługi plików jest brak możliwości otwarcia tego samego typu pliku przez różne aplikacje. Podklucze definicji klas zawierają także informacje o domyślnych ikonach wyświetlanych przez Windows.

Te informacje są wykorzystywane w celu:

- wyświetlania ikony na przycisku paska zadań podczas działania aplikacji
- wyświetlania ikony na pulpicie, gdy tworzysz połączenie lub skrót do pliku bądź aplikacji
- wyświetlania ikony w Eksploratorze Windows
- wyświetlania ikony na belce tytułowej aplikacji
- wyświetlania ikony w Menu Start

Ustawienia rejestru dotyczące ikon znajdują się w podkluczach definicji klas, w zawartych w nich podkluczach "DefaultIcon".

W podkluczu tym Windows może przechowywać następujące typy wartości:

- określenie ścieżki oraz nazwy pliku EXE lub DLL zawierającego ikony. Oprócz ścieżki trzeba podać numer ikony w pliku liczony od zera
- określenie ścieżki oraz nazwy pliku EXE lub DLL zawierającego ikony, dodanie znaku minus oraz identyfikatora zasobu ikony. Aby użyć tej metody, należy znać numer zasobu.
- określenie ścieżki oraz nazwy pliku obrazka, który Windows może wyświetlić jako ikonę. Mogą to być następujące typy plików:

1. BMP – plik bitmapy
2. ICO – plik ikony
3. CUR – plik kursora
4. SCR – plik wygaszacza ekranu
5. ANI – plik animowanego kursora
6. RLE – spakowana bitmapa

- dołączenie ciągu %1 instruującego Windows by użył uchwytu ikony, w celu określenia ikony dla danego typu pliku. Ta metoda jest stosowana w przypadku aplikacji EXE. Dzięki temu w przypadku różnych aplikacji nie jest wyświetlana ta sama ikona.

4.3 Szybki podgląd

Wraz z Windows jest dostarczany program Szybki podgląd, który pozwala na przeglądanie niektórych plików bez konieczności uruchamiania aplikacji w których zostały one stworzone. Dla plików dopuszczających szybki podgląd w menu kontekstowym systemu znajduje się pozycja szybki podgląd. Dane na temat szybkiego podglądu znajdują się w podkluczu HKEY_CLASSES_ROOT\QickView. W tym podkluczu znajduje się lista podkluczy definicji klas typów plików wraz z podkluczami, zawierającymi identyfikatory klas (CLSID). Typ pliku który może być przeglądany za pomocą szybkiego podglądu jest zdefiniowany w postaci .XXX , gdzie xxx to nazwa rozszerzenia danego typu pliku. Zaczyna się on od kropki i zawiera podklucz z identyfikatorem klasy która obsługuje dany szybki podgląd; (CLSID) jako wartość domyślą zawiera opis tekstowy przeglądarki dla szybkiego podglądu. Pozwala to na łatwe odczytanie jaka aplikacja obsługuje szybki podgląd danego typu plików.

4.4 Obsługa plików nieznanych typów

Gdy użytkownik dwukrotnie klinie na pliku, Windows szuka w kluczu HKEY_CLASSES_ROOT podklucza zgodnego z rozszerzeniem pliku. Gdy zdarzy się że, nie znajdzie takiego podklucza automatycznie przechodzi do podklucza \Unknown i w jego podkluczu \shell\openas\command szuka instrukcji obsługi takiego pliku. Domyślną wartością tego podklucza jest „C:\Windows\rundll32.exe shaell32.dll,OpenAs_RunDLL %1”, nakazująca wyświetlenie okna dialogowego „Otwórz z” Wewnątrz tego okna użytkownik może wybrać aplikację którą może użyć do otwarcia takiego pliku.

4.5 Pozycje rejestru związane z menu kontekstowym

Bardzo dużym udogodnieniem w Windows 9x i NT są menu kontekstowe. Takie menu jest wyświetlane w momencie kliknięcia prawym przyciskiem na ikonie pliku widzianej w oknie eksploratora. Informacje używane do sterowania menu kontekstowym przechowywane są w kluczu HKEY_CLASSES_ROOT. Windows korzysta z następujących elementów menu kontekstowego:

- elementy statyczne: Te elementy są oparte na powiązaniach pliku i należą do nich polecenia „Otwórz z”, „Szybki podgląd” itp.
- elementy dynamiczne: Te elementy zmieniają się na podstawie zawartych w rejestrze procedur obsługi menu kontekstowego. Procedury te to pozycje serwera OLE, określające jakie elementy dynamiczne dodawane są do menu.

- elementy definiowane przez program: Te elementy obsługiwane są przez eksploratora Windows i nie powinny być modyfikowane. Niektóre z popularnych poleceń tej kategorii to „Wyślij do”, ”Wytnij”, ”Wstaw”, ”Kopiuj” itp.

4.6 Statyczne elementy menu kontekstowego

Proces postępowania podczas budowy menu kontekstowego zaczyna się od zajrzenia do podklucza HKEY_CLASSES_ROOT*\shell, w celu zlokalizowania statycznych elementów menu. Standardowo ten podklucz nie występuje ale użytkownik może go dodać by wprowadzić pozycję menu kontekstowego występującą nie zależnie od typ pliku. Następnym krokiem jest przejrzenie podkluczy definicji klasy wskazanego pliku i zlokalizowanie podklucza \shell. Windows wie jak obsłużyć standardowe polecenia takie jak open, print i nie trzeba w ich wartości domyślnej umieszczać nazwy pojawiającej się w menu kontekstowym. I tak dla open jest to Otwórz, dla print Drukuj itp. W ten sposób Windows kończy ustalanie statycznych elementów menu i rozpoczyna ustalanie dynamicznych elementów.

4.7 Dynamiczne elementy menu

Podobnie jak w przypadku elementów statycznych, elementy dynamiczne wynikają z typu pliku oraz z bieżącego stanu aplikacji i systemu. Dynamiczne elementy menu kontekstowego są definiowane by uelastycznić obsługę systemu typów plików. Na przykład dla plików typu EXE, COM, DLL, SYS można zdefiniować pozycję sprawdzania wirusów, dla plików spakowanych odpowiednie polecenie rozpakowywania w określonym programie depakującym. Tworząc określoną aplikację można tak zaprogramować element menu kontekstowego by np. dla plików o określonych atrybutach były wyświetlane pozycje pozwalające na ich zdejmowanie.

Podklucze odpowiedzialne za pozycje dynamiczne menu kontekstowego znajdują się w podkluczach \shellx\ContextMenuHandlers. Tam umieszczane są dopiero podklucze np. WinZip wewnątrz których znajdują się identyfikatory klas pliku DLL, zawierający informacje o procedurze obsługi menu kontekstowego. W oparciu o identyfikator CLSID przeszukiwany jest podklucz HKEY_CLASSES_ROOT\CLSID i pobierana jest nazwa oraz położenie pliku DLL, do którego odnosi się identyfikator w podkluczu procedury obsługi menu. Nazwa DLL’a może zostać zlokalizowana w podkluczu \InProcServer32.

4.8 Identyfikatory CLSID

Podklucz CLSID zawiera identyfikatory klas dla typów plików. Każdy rodzaj obiektu OLE w Windows posiada unikalny identyfikator klasy CLSID, zawarty w tym podkluczu. Identyfikatory te pomagają systemowi w organizowaniu różnych obiektów OLE, włącznie z plikami DLL, EXE, funkcjami Windows oraz typami plików. Identyfikator klas składa się z ciągu cyfr szesnastkowych, ujętych w nawiasy klamrowe, podzielonego kreskami na grupy:

Osiem cyfr, trzy grupy po cztery cyfry i jedna grupę dwunastu cyfr. Są one tworzone przez programistów podczas pisania programów OLE. Ponieważ CLSID musi być unikalny dla wszystkich aplikacji występujących w pojedynczym komputerze to numery CLSID są generowane losowo za pomocą specjalnych programów, tak zoptymalizowanych by wygenerowany numer był niepowtarzalny. Takim programem jest guidgen.exe z pakietu Microsoft VC++. W podkluczu CLSID występuje jeden lub kilka podkluczy. Do najczęściej spotykanych należą:

- DefaultIcon – określa ikonę stosowaną do oznaczenia obiektu, do którego odnosi się dany CLSID
- InprocHandler – wskazuje na procedurę obsługi obiektu, Plik DLL współpracujący z plikiem EXE przy obsłudze danego typu pliku.
- InprocServer - Zawiera wskaźnik do In-Process Server obiektu, który jest plikiem DLL obsługującym zdefiniowany typ obiektu CLSID
- InprocServer32 – Wskazuje na 32 bitowy In-Process Server
- LocalServer – zawiera ścieżkę i nazwę pliku aplikacji serwera , czyli plik EXE obsługujący obiekty danego typu.
- ProgID – zawiera czytelny dla człowieka identyfikator CLSID
- ShellEx – zawiera podklucze definiujące, które DLL'e i polecenia używane są do wywoływania menu kontekstowego obiektu lub okna dialogowego właściwości.

4.9 Podklucz InprocServer32

Podklucze te są tworzone dla 32 bitowych aplikacji i są podkluczami kluczy CLSID. InprocServer32 zawiera wskaźniki do serwera in-process obiektu, czyli odpowiedniej biblioteki DLL obsługującej ten typ obiektów. Aby serwery in-process mogły działać wielowątkowo, podklucz inprocServer32 zawiera także informacje o modelu wielowątkowości serwera. Wartość dla tego typu ustawienia nosi nazwę TheradingModel. Ta pozycja może mieć następujące wartości:

- Apartment - wskazuje ze używany jest tzw. Apartment therading model. Tryb ten nakazuje aby każdy obiekt utworzony przez serwer OLE mógł istnieć tylko w jednym wątku. W tym typie wielowątkowości możliwe jest

tworzenie wielu obiektów, a każdy z nich posiada indywidualną kolejkę komunikatów.

- Both - Wskazuje, że server OLE obsługuje dwa modele wielowątkowości – apartment oraz free. Tryb free oznacza, że obiekt COM jest dostępny z kilu wątków pojedynczego procesu.

4.10 Podklucz ProgID

Zawiera programowe identyfikatory komponentów OLE. Te identyfikatory umożliwiają programom i innym obiektom odwoływanie się do obiektu w sposób alternatywny do użycia CLSID. W większości przypadków ProgID jest dużo bardziej czytelny dla człowieka niż numery CLSID. Identyfikatory ProgID są podobne do identyfikatorów znajdujących się w podkluczach definicji klas dla typów plików.
ProgID występuje w dwóch postaciach:

- niezależnej od wersji – zawierającej dane tylko o producencie i typie bez wersji np. Soft.Paint
- zależnej od wersji – zawierającej dane o wersji np. SoftPaint.8

Jednym z zastosowań ma miejsce wtedy gdy użytkownik ma wybrać z listy obiekt OLE.

5 Pliki REG

Plik REG są plikami tekstowymi zawierającymi dane o kluczach i ich wartościach. Można je w łatwy sposób importować do rejestru. Swą strukturą przypominają pliki INI znane z Windows 3.x. Plik ten jest podzielony na kilka sekcji. Pierwszą linię stanowi REGEDIT4 a po niej następuje pusty wiersz i pierwsza sekcja. Sekcja jest zamknięta pomiędzy nawiasami kwadratowymi i zawiera pełną ścieżkę wpisywanego klucza łącznie z nazwą symboliczną klucza głównego. Po sekcji dotyczącej nazwy klucza następuje sekcja wartości w kluczu. Każdy z typów danych ma inny format. Dla wartości tekstowej składnia jest następująca:
„Nazwa wartości” = ”tekst”
Wartości DWORD zapisane są jako liczba ośmiu cyfrowa w zapisie heksadecymalnym poprzedzona symbolem DWORD:
„Nazwa wartości”=DWORD:00000000
Wartości binarne są konwertowane do ciągu znaków poprzedzonych symbolem HEX: .Liczby w tym ciągu są zapisane jako dwucyfrowe i oddzielone znakiem przecinka. Długość takiego ciągu znaków nie powinna przekraczać 80 znaków łącznie z nazwą wartości i dodatkowymi symbolami. Wynika z tego, że w wierszu można praktycznie zapisać nie więcej jak 76

znaków. Gdy ciąg znaków przekracza 76 należy zakończyć go przecinkiem oraz znakiem kontynuacji wiersza którym jest <\>. Następny wiersz należy poprzedzić dwoma znakami spacji. W nazwie zmiennej jak i w jej wartości nie mogą występować znaki <”> i <\> . By je tam umieścić należy poprzedzić je znakiem <\>.
Chcąc umieścić wartość w (Wartości domyślnej) należy zamiast nazwy zmiennej użyć znaku <@> np. @=”tekst”

Przykład pliku REG :

```
REGEDIT4

[HKEY_CLASSES_ROOT\.aaa]
@=”Plik.AAA”
„Opis”=”To jest typ pliku użytkownika”
„Flags”=HEX:00,00,11,10
„Wartosc”=DWORD:0011AAF0
```

Plik ten umieści w rejestrze klucz .AAA w kluczu głównym HKEY_CLASSES_ROOT. W nim umieści wartość domyślną „Plik.AAA”, wartość tekstową o nazwie „Opis” i wartość binarną „Flags” oraz wartość DWORD o nazwie „Wartość”.

Nazwy kluczy głównych:
- HKEY_LOCAL_MACHINE
- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_CURRENT_CONFIG
- HKEY_USERS

Pliki REG pozwalają w bardzo prosty sposób wnosić poprawki do rejestru ale nie pozwalają na zarządzanie nim. Nie ma możliwości kasowania kluczy czy też wartości z rejestru. Nadają się doskonale natomiast do sporządzania kopii zapasowych części lub całości rejestru. Taką kopię zapasową można potem wczytać nawet z poziomu dyskietki systemowej za pomocą regedit.exe. Niestety dopiero Windows 95OSRB posiada poprawioną wersję tego programu która powala na importowanie dużych kluczy do rejestru.

6 Modyfikowanie rejestru z poziomu systemu operacyjnego

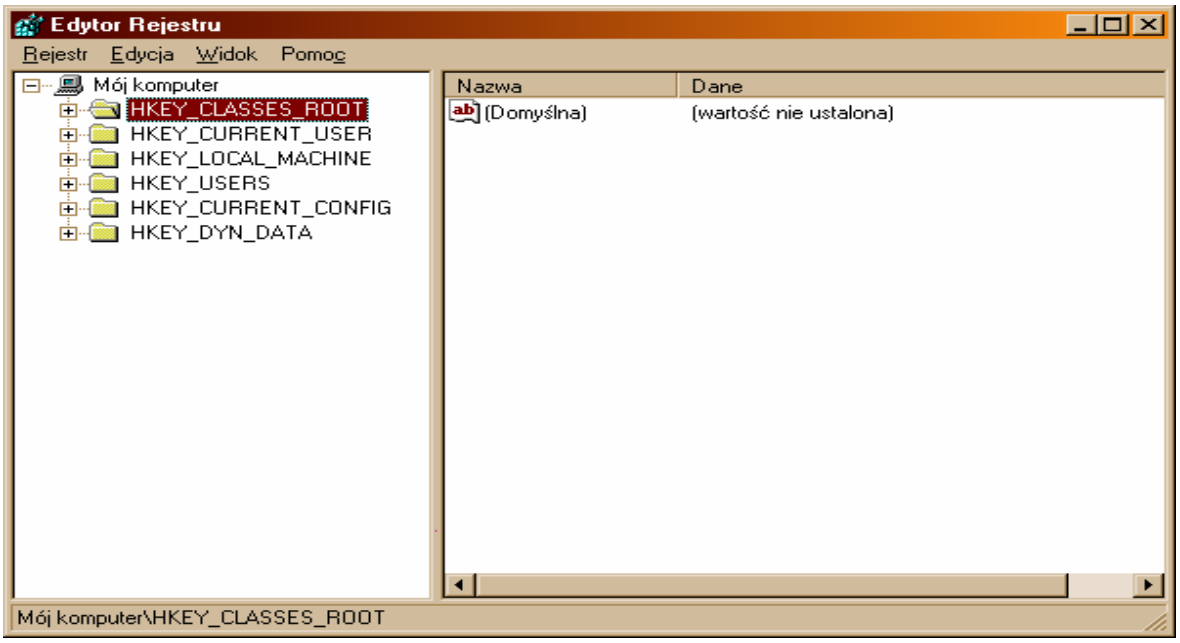
Wnoszenie poprawek do rejestru zawsze wiąże się z ryzykiem zmiany jakiś „żywotnych" parametrów systemu, więc przed każdą taką operacją należy wykonać kopie bezpieczeństwa danych które zmieniamy. Edytory rejestru dołączone do systemu Windows nie posiadają polecenia <Undo> (<Cofnij>). Każda zmiana nie jest buforowana, operacje są prowadzone bezpośrednio na rejestrze.

Tworzenie kopii rejestru może być przeprowadzone dwojako :

- skopiowanie w inne miejsce plików rejestru (możliwe w praktyce tylko w Win 9x)
- wykonanie eksportu części lub całości rejestru do pliku REG.

Ta ostatnia metoda ma tą zaletę że, działa we wszystkich systemach Windows. Pliki REG są plikami tekstowymi, o formacie opisanym wcześniej w rozdziale „Pliki REG". Należy tu pamiętać o jednej zasadzie. W Windows 95 do wersji OSR B ponowny import dużych kluczy do rejestru nie działa poprawnie ze względu na błąd jaki zakradł się do programu regedit.exe. Dopiero od wersji 95OSRB działa poprawnie. Można jednak w poprzednich wersjach spróbować użyć regedit.exe z wersji OSRB. W WinNT ten problem nie występuje.

Gdy zachodzi potrzeba bezpośredniej ingerencji w rejestr, należy spróbować skorzystać z aplikacji które zakulisowo dokonują tych poprawek. Do takich narzędzi należą Tweak UI firmy Microsoft, Microsoft Plus, i wiele innych narzędzi z serii shareware i freeware. Doświadczeni użytkownicy mogą korzystać z edytora rejestru dołączonego do systemu regedit.exe. Filozofia korzystania z tego ostatniego jest bardzo prosta. Program jest podzielony na dwie sekcje. Po lewej stronie znajduje się drzewo kluczy a po prawej panel z zawartymi w wskazanym kluczu wartościami. Pozwala on ponadto na wyszukiwanie danych, kluczy i wartości w rejestrze. Każda z sekcji posiada menu podręczne pozwalające na kilka podstawowych operacji dla niej charakterystycznych . Każda zmiana wnoszona w tym programie jest od razu wykonywana na rejestrze. Nieuważne operowanie kasowaniem kluczy może mieć bardzo zgubny skutek. Może to doprowadzić do tego że system nie uruchomi się poprawnie ponownie. Na rysunku poniżej zamieszone jest główne okno programu regedit.exe.



Rys. 7. Okno programu regedit.exe.

W systemie Windows NT jest jeszcze dołączony jeden edytor rejestru : regedt32.exe. Jest programem bardzo użytecznym i zarazem prostym. Nie posiada paska narzędzi. Każdy z głównych kluczy jest otwierany w osobnym oknie, jako dokument MDI. Program ten nie pozwala na zmianę nazw kluczy i wartości, wyszukiwać nazw wartości, poszukiwać danych. Pozwala natomiast na szybkie wyszukiwanie kluczy, oraz na zmiany wartości, tworzenie nowych kluczy oraz nowych wartości a także na kasowanie kluczy i wartości. Ma możliwość zapisu kluczy w postaci uli na dysku. Istnieje też możliwość odtwarzania kluczy na podstawie zapisanych uli. W zasadzie jest on używany do tworzenia kopii zapasowych kluczy lub całych uli, jego inne funkcje nie są zbyt dobrze napisane i do innych operacji używa się regedit.exe.

7 Naprawa uszkodzonego rejestru

W systemie Windows 9x naprawianie rejestru jest o wiele prostsze niż w Windows NT. Windows 9x przechowuje wszystkie dane w dwóch plikach: system.dat i user.dat. Nie są one zabezpieczone przed dostępem z zewnątrz, można je swobodnie kopiować. Natomiast w WinNT pliki przechowujące dane rejestru są zbudowane trochę inaczej. Nie można ich kopiować i przenosić z miejsca na miejsce, chroni je system.

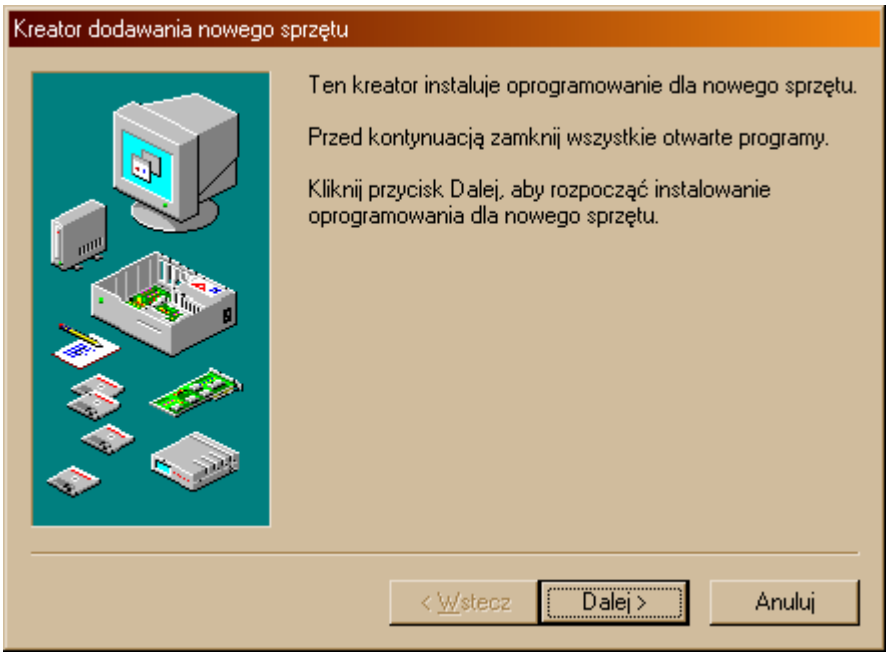
7.1 Windows 9x

Uruchamianie systemu w trybie awaryjnym:

Tryb ten to jedna z opcji uruchamiania systemu, powodujący, że Windows ładowany jest bez większości sterowników. Używane są w tym trybie jedynie sterowniki standardowe dla : karty VGA, myszy i klawiatury. W tym trybie system nie tworzy kopii bezpieczeństwa plików rejestru, więc nie trzeba tworzyć kopii zapasowych tych plików. W większości przypadków nie trzeba ręcznie zmieniać danych w rejestrze, można skorzystać z narzędzi systemowych.

Gdy mamy kłopoty z sprzętem to należy skorzystać z opcji ponownego wykrywania sprzętu. Dokonać tego można za pomocą kreatora <Dodaj nowy sprzęt> w panelu sterowania.

Rys. 8. Okno Panel Sterowania.



Rys. 9. Okno Dodaj nowy sprzęt.

1. Uruchom program zaznaczony na rysunku nr 8
2. Kliknij na przycisku <Dalej>
3. Wybierz opcje <Tak> pozwalając na to by Windows automatycznie wykrył sprzęt zainstalowany w komputerze
4. Kliknij na przycisku <Zakończ>

Gdy pojawiają się problemy z określoną aplikacją, najprostszym sposobem jest ponowna instalacja tego programu. Można nainstalować go na starą wersję co pozwoli odtworzyć jego poprzednie ustawienia. Niektóre programy proszą o deinstalację przed ponowną instalacją danej aplikacji. Zdarza się że, uszkodzona aplikacja nie daje się odinstalować za pomocą <Dodaj\ Usuń programy >. Wtedy nie pozostaje nam nic innego jak żmudna deinstalacja ręczna . Najpierw usuwamy program z dysku, potem odszukujemy w programie regedit.exe w następujących lokalizacjach kluczy naszego programu:

- HKEY_CURRENT_USER\SOFTWARE – klucza nazywającego się jak firma która napisała ten program np. Adobe. Szukamy w nim nazwy klucza podobnego do nazwy programu, robimy z niego kopię i kasujemy go.
- HKEY_LOCAL_MACHINE\Software - patrz wyżej.
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\ \CurrentVersion\Uninstall – nazwy klucza podobnej do nazwy programu, i sprawdzamy jego nazwę pojawiającą się w oknie <Dodaj/ Usuń programy> z nazwą wartości „Display Name” – gdy są takie same usuwany ten klucz.

Powyższe zabiegi powinny pozwolić usunąć większość zapisów z rejestru dotyczących naszej aplikacji.

Odbudowa rejestru za pomocą programu regedit.exe

Nie jest to najdoskonalsza metoda ale pozwala uruchomić komputer który nie startuje już w trybie „okienkowym". Sposób postępowania jest następujący:

- z linii poleceń trybu MS-DOS należy wprowadzić komendę: regedit /e registry.reg – która wyeksportuje do tego pliku wszystkie poprawne zapisy
- wprowadzić polecenie regedit /c registry.reg które zaimportuje poprawne zapisy do rejestru

Ta metoda ma dużą wadę, traci się wszystkie ustawienia w uszkodzonych częściach rejestru . Czasami są to na tyle istotne dane że, pomimo zastosowania tej metody Windows nie uruchamia się lub pracuje bardzo niestabilne. W takim przypadku jeśli nie dysponowaliśmy kopią zapasową to niestety czeka nas ponowna instalacja systemu.

Przywracanie kopii zapasowej rejestru

Jeżeli wykonaliśmy kopię zapasową plików system.dat i user.dat to możemy łatwo odtworzyć rejestr, przekopiowując je zamiast starych plików. Gdy natomiast nie posiadamy własnej kopii bezpieczeństwa należy spróbować odtworzyć rejestr z systemowych kopii. Noszą one nazwy system.da0 i user.da0. Należy zmienić ich nazwy na system.dat i user.dat. Często ten zabieg pomaga skutecznie odtworzyć ostatnią stabilną konfigurację systemu.

Odtwarzanie kopii z dysków bezpieczeństwa

Dobrym zwyczajem jest wykonywanie kopii rejestru i ważnych plików systemowych, na zewnętrznych nośnikach danych, takich jak taśmy, dyski ZIP, JAZ, CD-ROM, innych partycjach, lub fizycznie odrębnych dyskach twardych. Wtedy bardzo łatwo przywrócić stan pierwotny systemu. Kopie zapasowe można tworzyć ręcznie lub lepiej za pomocą specjalnych programów. Do tej grupy aplikacji należą: ERU, CfgBack i wiele innych dołączonych do napędów dysków wymiennych czy CDR/W.

7.2 Windows NT

Odtwarzanie rejestru w Windows NT nie jest już takie łatwe, jak w przypadku Win 9x. Nie można kopiować uli podczas działania systemu. Są jednak inne wyjścia.

Odtwarzanie systemu z nośników danych

Odbywa się na podobnej zasadzie jak w Windows 9x tylko za pomocą innych programów np.: RegBack

Odtwarzanie rejestru programem REGREST

Program ten wchodzi w skład Windows NT Resource Kit i można go używać do odtwarzania uli z plików stworzonych za pomocą programu z punktu pierwszego. Posiada on trzy różne tryby wywołania:

- Regrest - wyświetla pomoc programu
- Regrest BackupDir SaveDir - odtwarza ule zawarte w plikach w folderze BackupDir i zapisuje poprzednie wersje zastępowanych uli w folderze SaveDir
- Regrest Filename SaveFileName hivetype hivename – odtwarza ul zawarty w pliku Filename robiąc kopię poprzedniego ula w pliku SaveFileName. Parametr hivetype to Machine lub User, zaś hivename to nazwa ula, który chcemy odtworzyć

8 Programowanie operacji na Rejestrze Windows

Dostęp do rejestru z poziomu języka programowania jest możliwy za pomocą SDK dla programistów. API udostępnia szereg funkcji pozwalających na podstawowe operacje na rejestrze, takie jak tworzenie i otwieranie kluczy, kasowanie kluczy, dodawanie i modyfikacja wartości w kluczach, śledzenie zmian, zarządzanie dostępem do poszczególnych gałęzi i kluczy. Gdy aplikacje są tworzone z myślą o wykorzystaniu w Win32 oraz Windows 9x i NT to można skorzystać z funkcji występujących w tych systemach wspólnie. Ten zbiór funkcji nosi nazwę Win32s. Porównanie funkcji dla poszczególnych systemów jest umieszczone w poniższej tabeli:

Tabela 3. Funkcje Reg API dostępne dla poszczególnych systemów

Funkcja	***NT***	***9x***	***Win32s***
RegCloseKey	Tak	Tak	Tak
RegConectRegistry	Tak	Tak	Nie
RegCreateKey	Tak	Tak	Tak
RegCreateKeyEx	Tak	Tak	Tak
RegDeleteKey	Tak	Tak	Tak
RegDeleteValue	Tak	Tak	Nie
RegEnumKey	Tak	Tak	Tak
RegEnumKeyEx	Tak	Tak	Nie
RegEnumValue	Tak	Tak	Tak
RegFlushKey	Tak	Tak	Nie
RegGetKeySecurity	Tak	Nie	Nie
RegLoadKey	Tak	Tak	Nie
RegNotifyChangeKeyValue	Tak	Nie	Nie
RegOpenKey	Tak	Tak	Tak
RegOpenKeyEx	Tak	Tak	Tak
RegQueryInfoKey	Tak	Tak	Nie
RegQueryValue	Tak	Tak	Tak
RegQueryValueEx	Tak	Tak	Tak
RegRepalceKey	Tak	Tak	Nie
RegRestoreKey	Tak	Nie	Nie
RegSaveKey	Tak	Tak	Nie
RegSetKeySecurity	Tak	Nie	Nie
RegSetValue	Tak	Tak	Tak
RegSetValueEx	Tak	Tak	Tak
RegEnLoadKey	Tak	Tak	Tak

Dostęp do kluczy rejestru jest zorganizowany tradycyjnie za pomocą uchwytu. Podczas otwierania lub tworzenia klucza otrzymujemy uchwyt do niego. Uchwyt ten jest używany podczas wywoływania funkcji operujących

na tym kluczu i jego wartościach. Wszystkie dostępne funkcje można podzielić na następujące kategorie:

- zarządzanie kluczami
- zarządzanie wartościami
- wyliczenia
- kopie zapasowe/ odtwarzanie
- narzędzia
- ochrona

8.1 Zarządzanie kluczami

Zarządzanie kluczami na podstawowym poziomie wymaga użycia kilku funkcji. By dostać się do rejestru należy użyć funkcji RegOpenKeyEx() lub RegCreateKeyEx(). Obie te funkcje potrafią otworzyć istniejący klucz, dodatkowo funkcja RegCreateKeyEx() potrafi stworzyć nowy klucz.
Przykład:

```
RegCreateKeyEx(HKEY_CLASSES_ROOT, // uchwyt otwartego klucza
".aaa",                           // podklucz
0,                                // zarezerwowane
„mój opis"                        // łańcuch klasy (ignorowany w Win9x)
REG_OPTION_NON_VOLATILE,          // typ klucza (ignorowany w Win9x)
KEY_ALL_ACCESS,                   // pełny dostęp do klucza
 NULL,hKey,                       // zwracany chwyt podklucza
hDisp);                           //dyspozycja – tworzenie lub otwieranie
```

Tabela 4. Funkcje Reg API zarządzające kluczami

Funkcja	***Opis***
RegCloseKey()	Zamyka podany klucz rejestru i zwalnia uchwyt
RegCreateKey()	Otwiera podany klucz lub podklucz. Jeżeli podany klucz nie istnieje funkcja podejmie próbę stworzenia go. Funkcja ta jest obecnie nie używana i do tej operacji powinno się używać RegCreateKeyEx()
RegCreateKeyEx()	Otwiera podany klucz lub podklucz. Jeżeli podany klucz nie istnieje funkcja podejmie próbę stworzenia go. Zwracana wartość wskazuje czy klucz został otwarty, czy też dopiero stworzony.
RegDeleteKey()	Usuwa z rejestru wskazany klucz. Nie może usuwać głównych kluczy.
RegOpenKey()	Funkcja otwiera wskazany klucz rejestru Nie używana obecnie i zastąpiona RegOpenKeyEx()
RegOpenKeyEx()	Otwiera podany klucz lub podklucz. Każdy otwarty klucz powinien być zamknięty po użyciu.

Podczas tworzenia lub otwierania kluczy aplikacja określa pożądane prawa dostępu, nazywane maską SAM. Otrzymane prawa dostępu decydują o rodzaju dostępu, jaki aplikacja będzie miała do tego klucza. Prawa dostępu są zamieszczone w poniższej tabeli:

Tabela 5. Prawa dostępu do klucza

Nazwa	***Opis***
KEY_ALL_ACCESS	Stała reprezentująca kombinację praw dostępu: KEY_CREATE_LINK, KEY_CREATE_SUB_KEY, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY, KEY_QUERY_VALUE, KEY_SET_VALUE
KEY_CREATE_LINK	Prawo do tworzenia symbolicznego łącza.
KEY_CREATE_SUB_KEY	Prawo do tworzenia podkluczy.
KEY_ENUMERATE_SUB_KEYS	Prawo do wyliczania podkluczy.
KEY_EXECUTE	Prawo do odczytu klucza.
KEY_NOTIFY	Prawo do otrzymywania komunikatów o zmianie klucza.
KEY_QUERY_VALUE	Prawo do pobierania wartości z klucza.
KEY_READ	Stała będąca kombinacją: KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY
KEY_SET_VALUE	Prawo do dodawania nowych wartości.
KEY_WRITE	Stała będąca kombinacją : KEY_SET_VALUE oraz KEY_CREATE_SUB_KEY

Uwaga :
Prawa dostępu do klucza funkcjonują tylko w Windows NT. W Windows 9x są ignorowane.

Klucze stworzone przez funkcję RegCreateKeyEx() w środowisku Windows 9x są domyślnie nie ulotne. Parametr dotyczący typu klucza jest ignorowany. Klucz nie ulotny to taki klucz który przy zamykaniu systemu jest zapisywany na dysk do plików rejestru.

Funkcja RegDeleteKey() służy do usuwania z rejestru kluczy i zawartych w nich wartości. Cechuje ją odmienny sposób działania w Windows 9x i NT. O ile w Win9x usuwa klucz wraz z podkluczami i ich wartościami, to w WinNT nie usunie klucza jeżeli ten zawiera jakiekolwiek podklucze. Komplikuje to trochę pisanie aplikacji dla dwóch systemów jednocześnie, gdyż wymaga przeskanowania klucza w „dół” i rozpoczęcia kasowania od

najniższego podklucza posuwając się do góry aż do osiągnięcia klucza od którego rozpoczynamy kasowanie.

Wykorzystując funkcje do zarządzania kluczami należy pamiętać o kilku sprawach:

- klucze nie mogą być tworzone na głównym poziomie
- w środowisku NT można stosować klucze ulotne, dla danych które nie muszą być dostępne po ponownym załadowaniu systemu
- kasowanie kluczy przebiega odmiennie w Windows 9x i NT

8.2 Zarządzanie wartościami

Każdy klucz rejestru może zawierać powiązane z nim wartości. Wartości umieszczane w kluczach umożliwiają aplikacjom przechowywanie danych w różnych formatach. W środowisku programistów panuje zasada, że dane o długości powyżej 2 KB nie są przechowywane bezpośrednio w rejestrze, lecz w pliku, a w kluczu przechowuje się jedynie wskaźnik do pliku.

Tabela 6. Funkcje zarządzania wartościami

Funkcja	***Opis***
RegDeleteValue()	Usuwa wartość z podanego klucza rejestru
RegQueryValue()	Zwraca dane z pierwszej wartości zawartej w kluczu (wartości domyślnej)
RegQueryValueEx()	Zwraca daną i jej typ
RegSetValue()	Ustala wartość pierwszej wartości zawartej w kluczu (wartości domyślnej)
RegSetValueEx()	Ustala wartość i typ danej

Odczyt danych z rejestru odbywa się przy pomocy funkcji RegQueryValueEx(). Ta funkcja wymaga podania uchwytu żądanego klucza oraz nazwy wartości, zwracając w zamian daną wartość, jej rozmiar oraz typ. Aby móc odwołać się do klucza w celu pobrania wartości musi być on otwarty z prawem dostępu KEY_QUERY_VALUE.
Przykład:

```
RegQueryEx(hKey,            //uchwyt otwartego klucza
lpszValueName,              //wskaźnik do nazwy wartości
lpdwReservd,                //zarezerwowane
lpdwType,                   //bufor dla zwracanego typu wartości
lpbData,                    //bufor dla zwracanych danych
lpcbDataSize                //bufor dla zwracanego rozmiaru
                            danych

);
```

Funkcja ta doskonale nadaje się do czytania danych o znanych nazwach. Natomiast gdy aplikacja analizuje klucze i ich wartości musi mieć możliwość odczytania wszystkich nazw wartości danego klucza. Do tego służy funkcja RegEnumValueEx(), która jest opisana w podpunkcie *„Wyliczanie kluczy i wartości”*.

Tworzenie i ustalanie wartości odbywa się z zastosowaniem funkcji RegSetValueEx(). Jak widać w poniższym przykładzie funkcja wymaga podania uchwytu klucza, nazwy wartości, rozmiaru danych oraz typu danych. By operacja ta powiodła się klucz musi być otwarty z prawem dostępu KEY_SET_VALUE.

Przykład:

```
RegSetValueEx(hKey,          //uchwyt otwartego klucza
lpszValueName,               //wskaźnik do nazwy wartości
lpdwReservd,                 //zarezerwowane
lpdwType,                    //bufor dla zwracanego typu wartości
lpbData,                     //bufor dla zwracanych danych
cbDataSize                   //rozmiar danych
);
```

Jeżeli typem danych jest REG_SZ, REG_EXPAND_SZ, lub REG_MULTI_SZ w rozmiarze danych musi być uwzględniony znak Null kończący ciąg.

8.3 Wyliczanie kluczy i wartości

Aplikacje analizujące rejestr w większości muszą być w stanie przeglądać klucze i wartości. Aby to umożliwić, Win32 API zawiera grupę funkcji służących do tego celu.

Tabela 7. Funkcje wyliczeniowe rejestru

Funkcja	***Opis***
RegQueryInfoKey()	Zwraca informacje na temat klucza. Informacja ta zawiera nazwę klasy, ilość podkluczy, długość najdłuższej nazwy podklucza, długość najdłuższej nazwy podklucza, rozmiar najobszerniejszej wartości, długość deskryptora ochrony oraz czas ostatniego zapisu do klucza.
RegEnumKey()	Zwraca szczegółowe informacje na temat podkluczy klucza rejestru.
RegEnumKeyEx()	Zwraca szczegółowe informacje na temat podkluczy klucza rejestru.
RegEnumValue()	Każde wywołanie tej funkcji zwraca informacje o kolejnej wartości w kluczu.

Funkcje z tej grupy pozwalają na:

- poznanie szczegółów na temat kluczy i ich wartości
- stworzenie listy podkluczy klucza
- stworzenie listy wartości klucza

Jako przykład wykorzystania funkcji jest zaprezentowana funkcja która służy do skanowania rejestru i wyliczania jego kluczy i wartości.

Przykład:

```
ScanKeys(HKEY hKeyRoot, LPCTSTR KeyName)
{
 ...
 ...  // tu definicje zmiennych itp.

  //otwarcie podklucza
        retCode=RegOpenKey(hKeyRoot,KeyName,&hey);
 //gromadzenie informacj o kluczu

        retCode=RegQueryValue(
     hKey,
        lpszClass,
        &cchClass,
        &dwReservd,
```

```
        &cSubKeys,
        &cchMaxSubKeys,
        &cchValues,
      &cchMaxClass,
        cchMaxValueName,
        &cchMxValueData,
&cbSecurityDescriptor,
&ftLastWriteTime
};

//analiza wartości w kluczu
        if(cchValues>0) //są wartości w kluczu
        {
while((retCode=RegEnumValue(hKey,iValue,valuenamne,
&valuesize,&res,&datatype,pdata_buffer,&datasize))!=ERROR_NO_MORE_ITEMS)
{
 // tu operacje na wartościach podklucza
}

}

// analiza podkluczy

if(sSubKeys>0) // są podklucze
{
while((retCode=RegEnumKeyEx(hKey,iSubKey,sbkkeyname,
&cchMaxSubKeys,&res,classname,&cchMaxClass,&ftLastWriteTime))!=ERROR_NO_M
ORE_ITEMS)
{
  ScanKeys(hKeylpszNextKey);
}
}
   }
}
```

Funkcja ScanKeys() jest funkcją rekurencyjną, tzn. wywołuje samą siebie z swojego wnętrza. Takie rozwiązanie pozwala na mało skomplikowane rozwiązanie analizy kluczy ich wartości. Co prawda spada czytelność programu, ale kod jest o wiele krótszy i szybszy od kodu analizującego kolejne podklucze za pomocą wielokrotnych instrukcji if. Niestety to rozwiązanie nie jest idealne, i może doprowadzić do przepełnienia stosu systemowego i obszaru zmiennych programu. Każde wywołanie funkcji z poziomu jej samej powoduje zapamiętanie na stosie adresu wywołania tej funkcji, oraz stworzeniu kolejnych zmiennych tymczasowych. Przy wielu wywołaniach może to więc pomniejszyć zasoby systemowe.

Funkcja RegQueryInfoKey() udostępnia szczegółowe informacje na temat klucza rejestru, włącznie z ilością wartości i podkluczy oraz maksymalnymi

rozmiarami buforów na dane, wartości oraz nazwy klas. Natomiast funkcja RegEnumValue() zwraca nazwę, typ oraz rozmiar wskazanej wartości. W każdym wywołaniu zwraca dane na temat jednej wartości. Gdy nie masz już innych wartości to funkcja zwraca błąd ERROR_NO_MORE_ITEMS. Na podobnej zasadzie działa funkcja RegEnumKeyEx() tylko służy do wyliczania wszystkich podkluczy danego klucza.

8.4 Funkcje kopii zapasowych

Rejestr systemu Windows jest jego integralną bazą danych, więc powstało kilka funkcji do wykonywania kopii zapasowych, części lub całości rejestru.

Tabela 8. Funkcje kopii zapasowej

Funkcja	***Opis***
RegLoadKey()	Odczytuje informacje z pliku i ładuje je do podklucza rejestru.
RegReplaceKey()	Odczytuje dane z pliku i zastępuje stare wpisy, danymi z pliku. Działa tylko na bezpośrednich podkluczach kluczy HKY_LOCAL_MACHINE i HKEY_USERS.
RegRestoreKey()	Odczytuje informacje z pliku i kopiuje je w miejsce wskazanego klucza. Funkcja nie dostępna w Windows 9x.
RegSaveKey()	Zapisuje do pliku wskazany klucz i wszystkie zawarte w nim podklucze z wartościami.

Nawet aplikacje, które nie służą do tworzenia i odtwarzania kopii zapasowych mogą skorzystać z tych funkcji. Aplikacja może zapisać poddrzewo informacji do pliku, tak by móc odczytać je później, gdy użytkownik wybierze inną opcję.

Określenie funkcji które należy użyć do wykonania kopii zapasowej rejestru zależy zazwyczaj od :

- platformy na której działa aplikacja
- rodzaju kluczy które będą zapisywane i odtwarzane
- postaci danych

Używanie funkcji kopii zapasowej nie zawsze jest możliwe w danej aplikacji. Wszystkie funkcje kopii zapasowej działają tylko na bezpośrednich podkluczach kluczy HKEY_LOCAL_MACHINE i HKEY_USERS. Nie można za ich pomocą zapisać i odtworzyć kluczy w gałęziach HKEY_CLASSES_ROOT gdzie znajdują się zapisy dotyczące między

innymi skojarzeń plików, definicji klas. Problem ten można ominąć stosując dołączoną do pracy funkcję zapisu danych do formatu rozpoznawanego przez regedit.exe. Tworzy ona pliki tekstowe REG, które łatwo importować powtórnie do rejestru zarówno z poziomu systemu jak i poziomu aplikacji.

8.5 Ochrona rejestru

Jest możliwa tylko w Windows NT. Windows 9x nie zapewnia żadnej ochrony na poziomie kluczy. Wszystkie opisane w poniższej tabeli funkcje są dostępne tylko w WinNT.

Tabela 9. Funkcje ochrony rejestru

Funkcja	***Opis***
RegGetKeySecurity()	Zwraca informacje o ochronie wskazanego klucza.
RegSetKeySecurity()	Zmienia i ustawia opcje ochrony klucza.

Ochrona w WinNT to bardzo skomplikowane zagadnienie. Funkcje ochrony rejestru mogą zmieniać uprawnienia dostępu do klucza na poziomie grup i użytkowników oraz na poziomie systemu. Działa ona w oparciu o obiekty. Każdy obiekt rejestru posiada atrybuty ochrony.

Tabela 10. Informacje o ochronie

Typ	***ID***	***Opis***
Owner	OWNER_SECURITY_INFO	Identyfikuje głównego właściciela obiektu.
	GROUP_SECURITY_INFO	Identyfikuje główną grupę do której należy obiekt.
	DACL_SECURITY_INFO	Dyskretna lista kontroli dostępu określająca prawa dostępu do obiektu specyficzne dla użytkowników oraz grup.
	SACL_SECURITY_INFO	Systemowa lista kontroli dostępu określająca ochronę powiązanego ze sobą obiektu na poziomie systemu.

Funkcje ochrony wymagają także wskazania bufora na strukturę SECURITY_DES-CRIPTOR. Ta struktura zawiera informacje, dostarczane lub otrzymywane, na temat obiektu. Rodzaj informacji odpowiada rodzajowi informacji w strukturze SECURITY_NFORMATION. Struktura może zawierać wskaźniki do listy DACL, listy SACL, informacji o użytkowniku i inne. Te struktury są dość skomplikowane .

9 Programowanie standardów

Microsoft przyznaje certyfikat *przeznaczony dla Windows*, który pomaga użytkownikom w identyfikacji sprzętu i oprogramowania zgodnego z Windows NT i 9.x. Certyfikat otrzymują jedynie te produkty które przeszły szereg testów przeprowadzonych przez niezależną firmę VeriTest. Najważniejszymi wymaganiami koniecznymi do uzyskania tego certyfikatu są:

- wymagania dotyczące systemu operacyjnego
- wymagania dotyczące 32-bitów
- sposób instalowania
- procedura deinstalacyjna
- interfejs użytkownika
- OLE
- obsługa UNC/LFN

Wymagania dotyczące systemu operacyjnego:

Aby uzyskać certyfikat *Designed for Windows NT i Windows 9.x*, aplikacja musi działać zarówno w systemie WinNT jak i w Win9.x. Musi działać w systemie Windows NT 3.51 oraz 4.0 Workstation . Nie ma certyfikatu tylko dla aplikacji dla systemu Windows NT, ale istnieje certyfikat *Designed for Windows 95*.

Wymagania dotyczące 32- bitów

Aplikacje muszą być kompilowane za pomocą 32- bitowego kompilatora i muszą korzystać z interfejsu Win32. Wszystkie pliki wykonywalne i biblioteki DLL muszą być 32- bitowe (z kilkoma wyjątkami).

Instalowanie

Produkt musi posiadać w pełni zautomatyzowany program instalacyjny z graficznym interfejsem. Produkty rozprowadzane na płytach CD-ROM muszą korzystać z mechanizmu AutoPlay, który powoduje automatyczne uruchamianie programu instalacyjnego po włożeniu płytki do czytnika. Programy umieszczane na dyskietkach muszą zawierać plik uruchomieniowy o nazwie *setup.exe* który jest aplikacją inicjującą proces instalacji. Produkt musi rozpoznawać wersję Windows i instalować odpowiednie pliki w zależności od wersji. Dane o aplikacji muszą być umieszczone w odpowiednim miejscu rejestru systemowego. Dane te są podzielone na dwie grupy.

Ogólne ustawienia programu

Ogólne ustawienia programu to dane dostępne dla wszystkich użytkowników aplikacji. Umieszcza się je w kluczu:

HKEY_LOCAL_MACHINE\SOFTWARE\Nazwa firmy\Nazwa produktu\Wersja

Rejestracja rozszerzeń plików

Każda aplikacja która używa plików własnych typów musi je zarejestrować w systemie. Dane o rozszerzeniu pliku umieszcza się w kluczu:

HKEY_CLASSES_ROOT\ .<rozszerzenie>

Rejestracja definicji plików

Definicje plików są wymagane w przypadku własnej obsługi pliku o określonym rozszerzeniu. Dane umieszcza się w kluczu:

HKEY_CLASSES_ROOT\ <Nazwa definicji klasy>

Rejestracja wspólnych komponentów

Wspólne komponenty to komponenty wykorzystywane przez kilka aplikacji. Najpopularniejszymi komponentami są biblioteki DLL, kontrolki ActiveX, OCX.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsft\Windows\CurrentWersion\ SharedDLLs

Rejestracja procedury deinstalacyjnej

Procedura deinstalacyjna musi być odpowiednio zarejestrowana by była widoczna w oknie <Dodaj\ Usuń programy> w <Panelu sterowania>.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsft\Windows\CurrentWersion\ Uninstal\ Nazwa programu

Procedura deinstalacyjna

Aplikacja musi zapewniać w pełni zautomatyzowaną procedurę deinstalacyjną, która usunie wszystkie pliki, foldery i pozycje rejestru stworzone przez aplikację. Program deinstalacyjny musi usunąć wszystkie skróty utworzone przez aplikację, zmniejszyć licznik odwołań do wspólnych komponentów i musi być dostępny przez ikonę <Dodaj\ Usuń programy> w panelu sterowania.

Interfejs użytkownika

Aplikacja do wymiarowania swoich elementów interfejsu użytkownika musi korzystać z wartości dostarczonych przez system operacyjny. Informacje systemowe umożliwiają aplikacji prawidłowe stworzenie okien, w oparciu o atrybuty systemu.

OLE

Aplikacja musi być kontenerem OLE lub serwerem. Wyjątkiem są tu programy narzędziowe i specjalizowane.

Obsługa UNC/LFN

Aplikacja musi obsługiwać konwencję UNC (Universal Naming Conversion), która umożliwia odwołanie się do zasobów sieciowych, bez konieczności używania litery dysku. Musi też obsługiwać LFN (Long Flie Name) długie nazwy plików które mogą zawierać do 260 znaków.

9.1 Dodawanie do rejestru ustawień programu

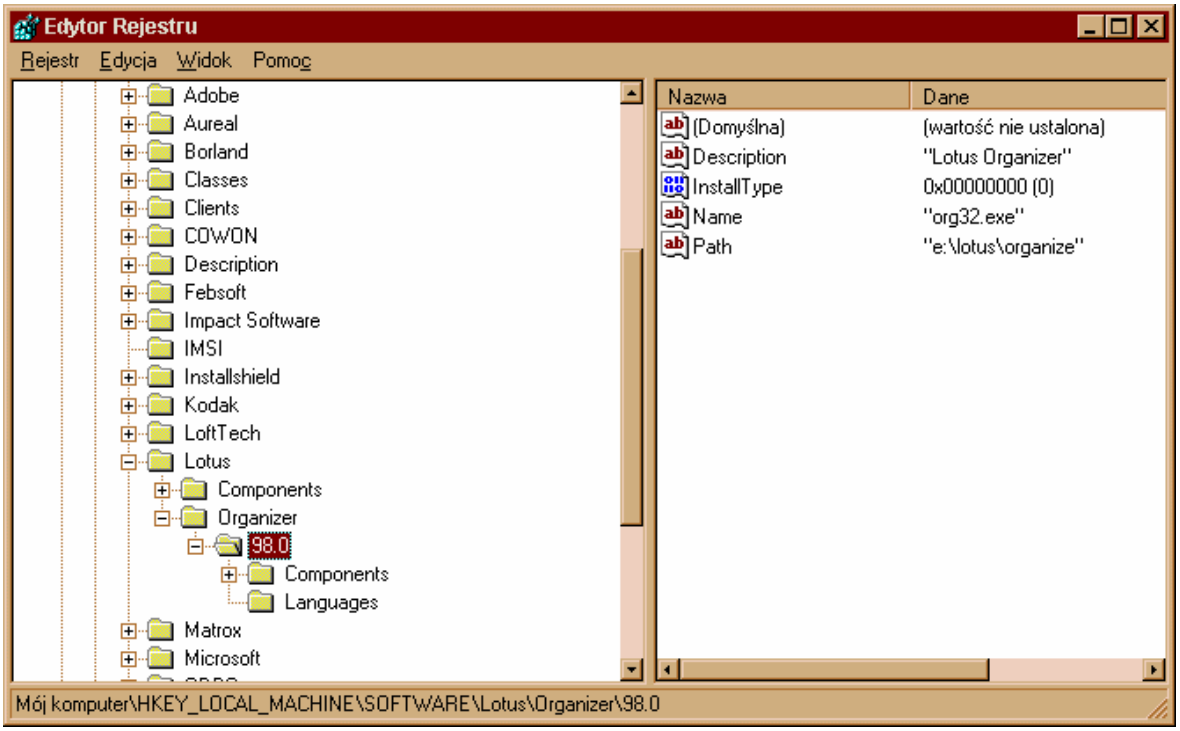
Kiedy aplikacja zapisuje i odczytuje swoje własne ustawienia, takie jak informacje o instalacji, nazwie użytkownika, położeniu okien itp. Należy korzystać z rejestru. Są dwa miejsca gdzie zapisuje się ustawienia programu. Jednym z nich są dane ogólne o aplikacji dostępne dla wszystkich użytkowników oraz dane szczegółowe zwane preferencjami użytkownika.

Ogólne ustawienia programu

Ogólne ustawienia programu odnoszą się do każdego użytkownika aplikacji. Występuje tylko pojedyncza kopia ogólnych ustawień; zawarte w niej dane są inicjalizowane podczas instalacji programu. W celu

przechowywania danych ogólnych aplikacja powinna stworzyć klucz rejestru o następującej składni:

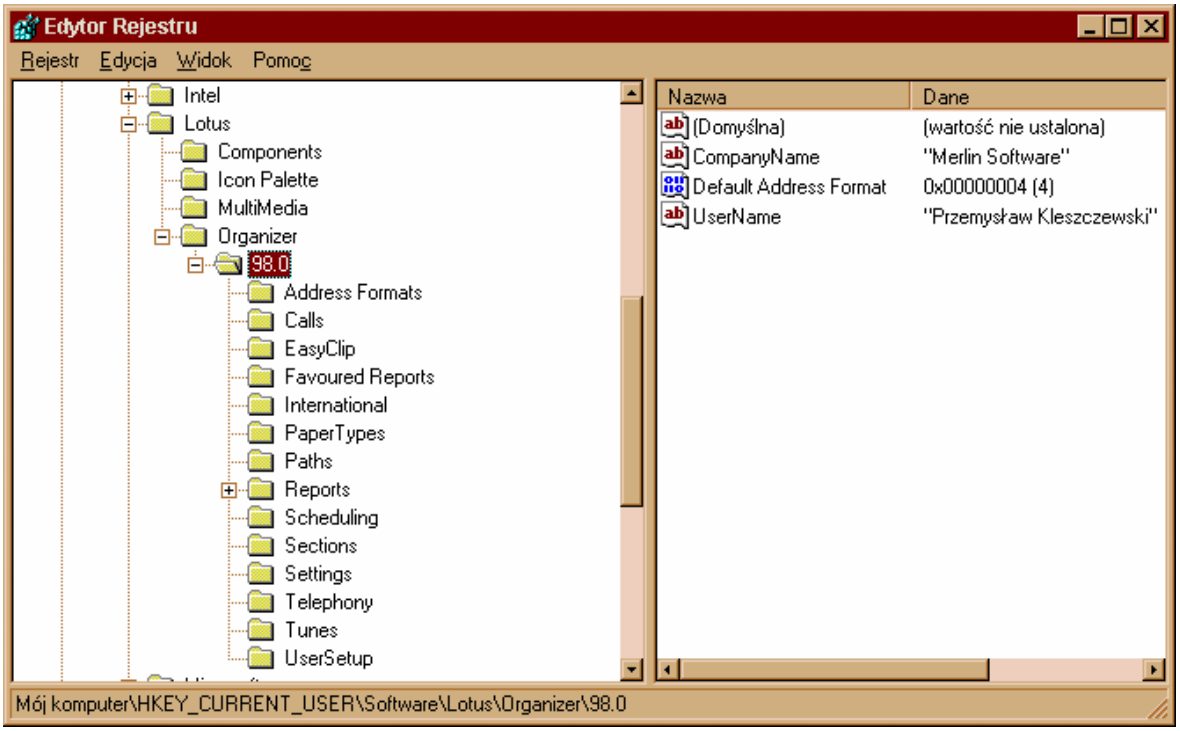
HKEY_LOCAL_MACHINE\SOFTWARE\< firma>\ < produkt >\ <wersja>



Rys 10. Dane ogólne programu Lotus Organizer 4.1

Preferencje użytkownika

Preferencje użytkownika to ustawienia programu specyficzne dla użytkowników aplikacji. Istnieje oddzielna kopia tych ustawień dla każdego z użytkowników, którzy uruchomili aplikację. Domyślne wartości ustawień użytkownika są inicjowane wtedy, gdy nowy użytkownik pierwszy raz uruchamia aplikację. Preferencje użytkownika są przechowywane w podkluczu o takim samym formacie jak ustawienia ogólne ale umieszczonym w gałęzi HKEY_CURRENT_USER.

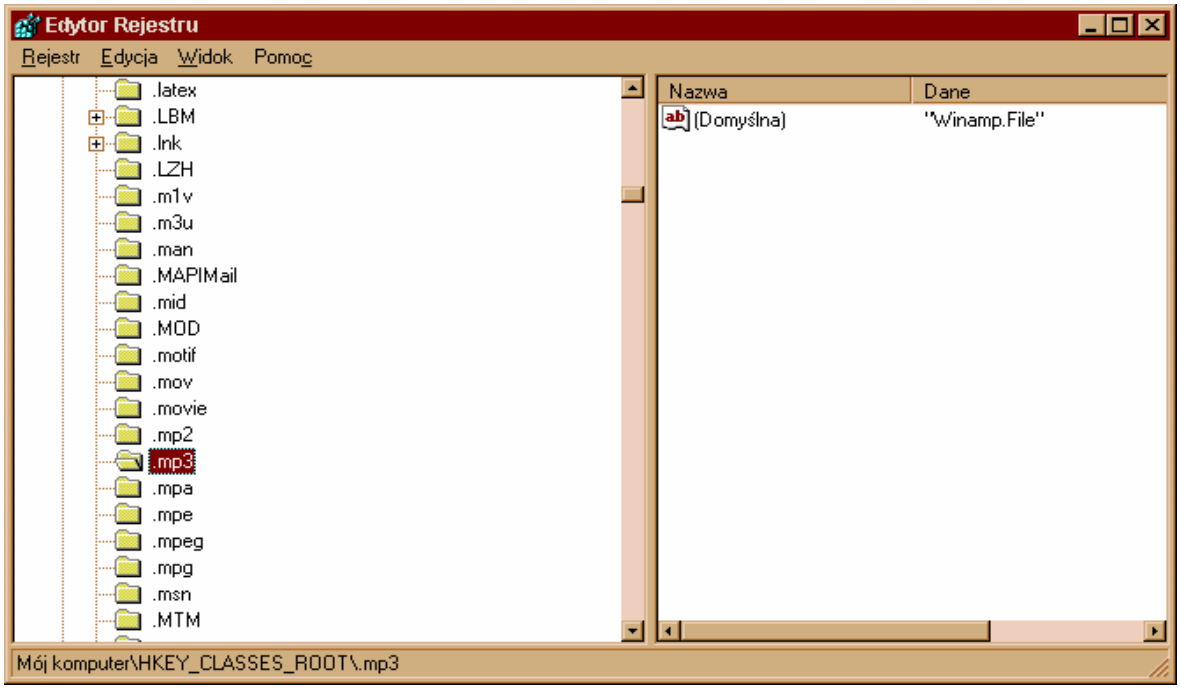


Rys 11. Dane szczegółowe programu Lotus Organizer 4.1

9.2 Integracja aplikacji z systemem Windows

Windows NT oraz Windows 9.x preferują zintegrowanie aplikacji z systemem i innymi aplikacjami. Korzystając z takich mechanizmów jak rozszerzenia powłoki i OLE, dane tworzone przez aplikację mogą zostać ściśle zintegrowane z programami powłoki Windows oraz innymi aplikacjami. Prostym przykładem może być dwukrotne kliknięcie na ikonie pliku w eksploratorze Windows. Dzięki prostym łączom stworzonym w rejestrze, użytkownik wybierający plik danych określonej aplikacji może spowodować automatyczne uruchomienie jej i załadowanie do niej wskazanego pliku. Inne użyteczne mechanizmy, takie jak OLE, pozwalają użytkownikowi na edycję wewnątrz jednej aplikacji obiektu stworzonego w innej aplikacji.

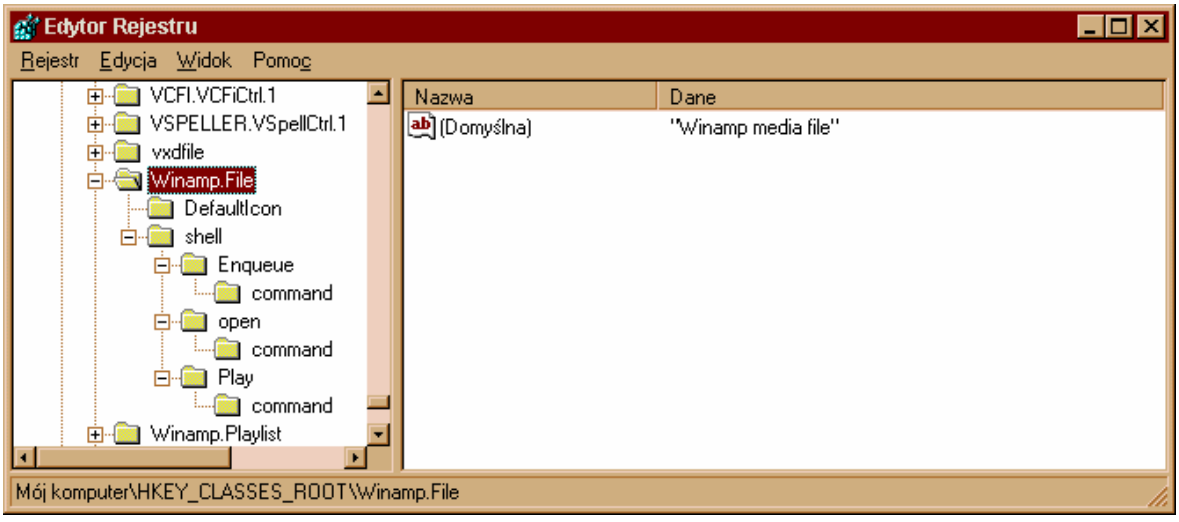
Podklucze typów plików zawierają informacje o typie pliku oraz wskazania na podklucz definicji typu pliku.



Rys. 12. Podklucz definicji typu pliku.

Na rysunku nr 12 przedstawiono typ pliku muzycznego MP3 zarejestrowanego w systemie. Wartość domyślna zawiera nazwę wskazującą na definicję typu pliku "Winamp.File" która zawiera informacje potrzebne do obsłużenia danego pliku.

Podklucze definicji plików zawierają informacje i polecenia wspomagające mechanizmy, takie jak rozszerzenia powłoki oraz OLE.



Rys. 13. Podklucz definicji pliku dla rozszerzenia MP3.

Rozszerzenia powłoki to mechanizmy dodane do programów powłoki Windows, takich jak eksplorator Windows czy menadżer zadań. Mechanizmy te umożliwiają twojej aplikacji ścisłe zintegrowanie z systemem.

Powiązania plików , rodzaj rozszerzenia powłoki pozwalający na uruchamianie plików określonego typu w przeznaczonych dla nich aplikacjach.

Zaawansowane rozszerzenia powłoki

Windows 95 i NT zawierają mechanizmy znane jako zaawansowane rozszerzenia powłoki. Te rozszerzenia łączą zmiany w rejestrze z tworzeniem DLL-i rozszerzeń powłoki w celu zapewnienia pełniejszej integracji z programami powłoki. Zaawansowane rozszerzenia mają moc tworzenia ikon związanych z instancją, dodawania zakładek do arkuszy właściwości, modyfikowania menu kontekstowych i wiele innych.

Obsługa OLE

OLE – Object Linking and Embedding – łączenie i osadzanie obiektów; umożliwia aplikacjom wspólnie i bezproblemowe korzystanie z danych. Przy pomocy OLE użytkownik może edytować wewnątrz jednej aplikacji pochodzące z innej aplikacji. Edycja rysunku wewnątrz Worda jest możliwa dzięki OLE. Polecenia OLE są przechowywane w podkluczu definicji aplikacji. Te polecenia powinny być tworzone podczas instalacji programu .

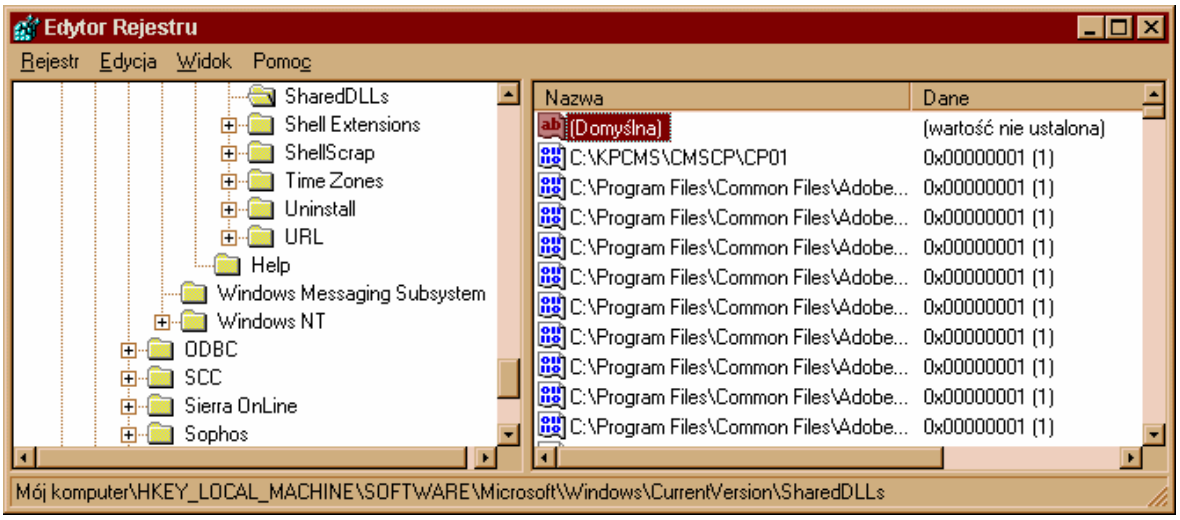
9.3 Instalowanie wspólnych komponentów

W wersjach poprzednich Windows aplikacje podczas instalacji miały trudności z utrzymaniem kontroli nad wersjami składników oprogramowania instalowanego w systemie. Nieomal standardem stawało się zastępowanie nowszych wersji bibliotek DLL starszymi, występowanie kilku kopii tej samej biblioteki, i tylko przypadek decydował która wersja biblioteki była ładowana pierwsza. Powodowało to wiele konfliktów, aplikacje które do tej pory działały poprawnie, po instalacji nowego oprogramowania przestawały działać, lub działały nie prawidłowo.

Aby wyeliminować takie kłopoty, Microsoft wprowadził w Windows 9x i NT koncepcję wspólnych komponentów. Komponenty te, wykorzystywane w aplikacji, umieszczane są w jednym miejscu. Lista wspólnych komponentów jest umieszczona w rejestrze. Zawiera ona także tzw. Licznik odwołań który określa ilość aplikacji w systemie korzystających ze wspólnej biblioteki. Lista ta znajduje się w kluczu

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\CurrentVersion\Shared DLLs. Podczas instalacji aplikacja jest odpowiedzialna za poprawne zainstalowanie wspólnych komponentów i zwiększenie licznika odwołań wykorzystywanych komponentów a podczas deinstalacji ma obowiązek zmniejszyć ten licznik. Postępowanie podczas instalacji aplikacji z wspólnymi komponentami jest następujące:

- sprawdzenie czy wspólny komponent jest już zainstalowany w systemie
- sprawdzenie czy jest to minimalna wystarczająca wersja komponentu
- gdy jest zainstalowany odpowiedni komponent to zwiększyć licznik odwołań
- gdy nie ma zainstalowanego komponentu należy go skopiować do wspólnego katalogu, dodać go do listy w rejestrze i ustawić licznik odwołań na 1



Rys. 14. Pozycje podklucza SharedDLLs.

9.4 Informacje o deinstalacji

Koniecznym wymaganiem do otrzymania certyfikatu od Microsoftu jest obecność programu deinstalacyjnego. Procedura deinstalacyjna powinna usuwać wszystkie pliki programu, pozycje rejestru, skróty oraz dane stworzone podczas instalacji lub działania programu.

W celu udostępnienia procedury deinstalacyjnej użytkownikowi, do rejestru muszą być wpisane nazwa i położenie tej procedury. Procedura deinstalacyjna jest dostępna dla wszystkich użytkowników, gdyż jest zdefiniowana w gałęzi HKEY_LOCAL_MCHINE. Informacje dla deinstalacji powinny zostać umieszczone w rejestrze przez procedurę instalacyjną. Sam projekt procedury deinstalacyjnej należy do projektanta aplikacji.

- RegCloseKey()

Funkcja RegCloseKey() zwalnia uchwyt otwartego klucza.

Składnia:
LONG RegCloseKey(HKEY hKey)

Parametry:

hKey	Uchwyt otwartego klucza przeznaczonego do zamknięcia.

Zwracane wartości:

ERROR SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Funkcja ta nie zawsze działa natychmiast, czasami mija sporo czasu zanim Windows zrzuci pamięć podręczną.

- RegConnectRegistry()

Funkcja RegConnectRegistry() nawiązuje połączenie z predefiniowanym kluczem rejestru w innym komputerze.

Składnia:
LONG RegConnectRegistry(LPTSTR IpMachineName, HKEY hKey,
PHKEY phkResult)

Parametry:

lpMachineName	Wskaźnik do ciągu znaków zawierającego nazwę zdalnego komputera (na przykład "\\Komputer").
hKey	Jeden z predefiniowanych uchwytów: HKEY_LOCAL_MACHINE, HKEY_USERS
phkResult	Adres bufora na zwracany uchwyt.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
W celu zamknięcia klucza zwróconego przez funkcję RegConnectRegistry użyj funkcji RegCloseKey.

- `RegCreateKey()`

Funkcja RegCreateKey tworzy nowy klucz lub otwiera klucz, który już istnieje. Programy Win32 powinny zamiast niej używać funkcji RegCreateKeyEx.

Składnia:
LONG RegCreateKey(HKEY hKey, LPCTSTR lpSubKey, PHKEY phkResult)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Wskaźnik do ciągu znaków określających nazwę klucza, który zostanie stworzony lub otwarty wewnątrz klucza hKey.
phkResult	Adres bufora na zwracany uchwyt.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
	kod błędu Wywołanie funkcji nie powiodło się.

Opis:
Klucz identyfikowany przez hKey musi być wcześniej otwarty z prawem dostępu KEY_ CREATE_SUB_KEY. Jeśli jako wartość parametru lpSubKey podaliśmy wartość NULL, funkcja otworzy i zwróci uchwyt do klucza identyfikowanego przez hKey.

- `RegCreateKeyEx()`

Funkcja RegCreateKeyEx tworzy nowy klucz lub otwiera klucz, który już istnieje.

Składnia:

```
LONG RegCreateKeyEx(HKEY hKey,
                    LPCTSTR lpSubKey,
                    DWORD Reserved,
                    LPTSTR lpCIass,
                    DWORD dwOptions,
                    REGSAM samDesired,
                    LPSECURITY_ATTRIBUTES
                    lpSecurityAttributes, PHKEY phkResult,
                    LPDWORD lpdwDisposItion)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Wskaźnik do ciągu znaków określających nazwę klucza, który zostanie stworzony lub otwarty wewnątrz klucza hKey.
Reserved	Zarezerwowane; musi zawierać wartość zero.
lpCIass	Wskaźnik do ciągu znaków, który określa nazwę klasy klucza.
dwOptions	Opcje klucza
samDesired	Opcje dostępu do klucza
lpSecurity	Adres struktury SECURITY_ATTRIBUTES.
phkResult	Adres bufora na zwracany uchwyt.
lpdwDisposition	Adres bufora na zwracaną wartość dyspozycji

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:

Klucz identyfikowany przez hKey musi być wcześniej otwarty z prawem dostępu KEY_ CREATE_SUB_KEY. Jeśli jako wartość parametru lpSubKey podamy wartość NULL, funkcja otworzy i zwróci uchwyt do klucza identyfikowanego przez hKey.

Wartości parametru dwOptions:

REG_OPTION_VOLATILE	Parametr ignorowany w Windows 95. W Windows NT ta wartość wskazuje, że zawartość klucza nie zostanie zapisana na dysk.
REG_OPTION_NON_VOLATILE	Zawartość klucza zostanie zapisana na dysk.

Wartości parametru samDesired:

KEY_ ALL _ACCESS	Kombinacja uprawnień KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY, KEY_CREATE_SUB_KEY, KEY_CREATE_LINK oraz KEY_SET_VALUE.
KEY_CREATE_LINK	Zezwolenie na tworzenie symbolicznych łączy.
KEY_CREATE_SUB_KEY	Zezwolenie na tworzenie podkluczy.
KEY_ENUMERATE_SUB_KEYS	Zezwolenie na wyliczanie podkluczy.
KEY_EXECUTE	Zezwolenie na odczyt.
KEY_NOTIFY	Zezwolenie na uaktywnienie powiadamiania.
KEY_QUERY_VALUE	Zezwolenie na odczyt podkluczy.
KEY_READ	Kombinacja uprawnień KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS , KEY NOTIFY.
KEY_SET_VALUE	Zezwolenie na ustawianie danych.
KEY_WRITE	Kombinacja uprawnień KEY_SET_VALUE oraz KEY_CREATE_SUB_KEY.

Wartości parametru lpdwDisposition:

REG_CREATED_NEW_KEY	Został stworzony nowy klucz.
REG_OPENED_EXISTING_KEY	Został otwarty już istniejący klucz.

- `RegDeleteKey()`

Funkcja RegDeleteKey usuwa klucz z rejestru.

Składnia:
LONG RegDeleteKey(HKEY hKey, LPCTSTR lpSubKey}

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Wskaźnik do ciągu znaków określających nazwę podklucza, który ma zostać usunięty.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Funkcja RegDeleteKey usuwa dany klucz z rejestru, łącznie z wszystkimi zawartymi w nim wartościami. Funkcja nie może usunąć klucza, który zawiera podklucze. Klucz musi być otwarty przy pomocy funkcji RegCreateKeyEx lub RegOpenKeyEx.

- `RegDeleteValue()`

Funkcja RegDeleteValue usuwa wartość ze wskazanego klucza.

Składnia:
LONG RegDeleteValue(HKEY hKey, LPCTSTR lpValueName}

Parametry:

HKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY USERS
lpSubKey	Wskaźnik do ciągu znaków określających nazwę wartości, która ma zostać usunięta.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Klucz identyfikowany przez hKey musi być wcześniej otwarty z prawem dostępu KEY_SET_VALUE.

- `RegEnumKey()`

Funkcja RegEnumKey wylicza podklucze otwartego klucza. Przy każdym wywołaniu zwraca nazwę kolejnego podklucza. Programy Win32 zamiast tej funkcji powinny korzystać z funkcji RegEnumKeyEx.

Składnia:
LONG RegEnumKey(HKEY hKey, DWORD dwlndex, LPTSTR lpValueName,
DWORD cbName)
Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących

	zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
dwlndex	Indeks podklucza, który ma zostać zwrócony (liczone od 0).
lpName	Bufor na nazwę podklucza.
cbName	Rozmiar bufora na nazwę podklucza.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_NO_MORE_ITEMS	Pobrano wszystkie podklucze.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Podczas pierwszego wywołania tej funkcji parametr dwlndex powinien mieć wartość 0. Następnie, przy każdym kolejnym wywołaniu funkcji, należy zwiększać wartość tego parametru o l, aż funkcja zwróci wartość ERROR_NO_MORE_ITEMS. Klucz identyfikowany przez hKey musi być otwarty z ustawionym uprawnieniem KEY_ENUMERATE_SUB_KEYS. W tym celu należy użyć funkcji RegCreateKeyEx lub RegOpenKeyEx. Należy pamiętać, że maksymalny wymagany rozmiar bufora na nazwę to MAX_PATH + l.

- `RegEnumKeyEx()`

Funkcja RegEnumKeyEx wylicza podklucze otwartego klucza. Przy każdym wywołaniu zwraca nazwę kolejnego podklucza.

Składnia:

```
LONG RegEnumKeyEx(HKEY hKey, DWORD dwlndex,
                  LPTSTR lpName, LPDWORD lpcbName,
                  LPDWORD lpReserved, LPTSTR lpCIass,
                  LPDWORD lpcbCIass, PFILETIME
          lpftLastWriteTime)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS.
dwlndex	Indeks podklucza, który ma zostać zwrócony (liczone od 0).
lpName	Bufor na nazwę podklucza.
lpcbName	Bufor na rozmiar nazwy podklucza.
lpReserved	Musi mieć wartość NULL.
lpCIass	Bufor na nazwę klasy podklucza.
lpcbCIass	Bufor na rozmiar nazwy klasy podklucza.
lpftLastWriteTime	Czas ostatniego zapisu do podklucza.

Zwracane wartości:

ERROR SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_NO_MORE_ITEMS	Pobrano wszystkie podklucze.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Podczas pierwszego wywołania tej funkcji parametr dwlndex powinien mieć wartość 0. Następnie, przy każdym kolejnym wywołaniu funkcji, należy zwiększać wartość tego parametru o l, aż funkcja zwróci wartość ERROR_NO_MORE_ITEMS. Klucz identyfikowany przez hKey musi być otwarty z ustawionym uprawnieniem KEY ENUMERATE_SUB_KEYS. Musisz użyć w tym celu funkcji RegCreateKeyEx lub RegOpenKeyEx.

- `RegEnumValue()`

Funkcja RegEnumValue wylicza wartości zawarte w kluczu rejestru.

Składnia:

```
LONG RegEnumValue( HKEY hKey,
                   DWORD dwlndex,
                   LPTSTR lpValueName,
                   LPDWORD lpcbValueName,
                   LPDWORD lpReserved,
                   LPTSTR lpType,
```

LPBYTE lpData,
LPDWORD lpcbData)

Parametry:

hKey	Uchwyt otwartego klucza.
dwIndex	Indeks wartości, która ma zostać zwrócona.
lpValueName	Bufor na nazwę wartości.
lpcbValueName	Bufor na rozmiar nazwy wartości.
lpReserved	Musi mieć wartość 0.
lpType	Bufor na typ wartości.
lpData	Bufor na dane wartości.
lpcbData	Bufor na rozmiar danych wartości.

Zwracane wartości:

ERROR SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_NO_MORE_ITEMS	Pobrano wszystkie wartości.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Podczas pierwszego wywołania tej funkcji parametr dwlndex powinien mieć wartość 0. Następnie, przy każdym kolejnym wywołaniu funkcji, należy zwiększać wartość tego parametru o l, aż funkcja zwróci wartość ERROR_NO_MORE_ITEMS. Klucz identyfikowany przez hKey musi być otwarty z ustawionym uprawnieniem KEY_QUERY_VALUE. Aby wyznaczyć maksymalny rozmiar bufora na dane, należy użyć funkcji RegQueryInfoKey.

Wartości parametru lpType:

REG BINARY	Dane binarne
REG_DWORD	Liczba 32-bitowa
REG_DWORD_LITTLE_ENDIAN	Liczba 32-bitowa w formacie little-endian
REG_DWORD_BIG_ENDIAN	Liczba 32-bitowa w formacie big-endian
REG_EXPAND_SZ	Ciąg znaków zakończony znakiem Null, zawierający odwołanie do zmiennej środowiskowej
REG_LINK	Symboliczne łącze Unicode
REG_MULTI_SZ	Tablica ciągów znaków oddzielonych znakami Null
REG_NONE	Nieokreślony typ danych
REG_RESOURCE_LIST	Lista zasobów sterowników urządzeń

REG_SZ	Ciąg znaków zakończony znakiem Null

- `RegFlushKey()`

Funkcja RegFlushKey powoduje natychmiastowy zapis zmian do rejestru.

Składnia:
LONG RegFlushKey(HKEY hKey)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Zbyt częste wywoływanie tej funkcji może spowolnić działanie aplikacji, stosuje się ją wtedy gdy istnieje konieczność natychmiastowego zapisu do rejestru.

- `RegGetKeySecurity()`

Funkcja RegGetKeySecurity zwraca kopię deskryptora ochrony otwartego klucza rejestru.

Składnia:
LONG RegGetKeySecurity(HKEY hKey,
SECURITY_INFORMATION SecurityInformation,
PSECURITY_DESCRIPTOR pSecurityDescriptor,
LPDWORD lpcbSecurityDescriptor)

Parametry:

hKey	Uchwyt otwartego klucza
SecurityInformation	Wykaz informacji o ochronie, które chcemy pobrać
pSecurityDescriptor	Adres bufora na deskryptor ochrony
lpcbSecurityDescriptor	Adres bufora na rozmiar deskryptora ochrony

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_INSUFFICIENT_BUFFER	Bufor jest zbyt mały.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli rozmiar bufora wskazywanego przez pSecurityDecriptor jest zbyt mały, funkcja zwraca wartość ERROR_INSUFICIENT_BUFFER, a w buforze IpcbSecurityDescriptor umieszcza wymaganą ilość bajtów. Wywołujący proces musi posiadać uprawnienie READ_CONTROL (KEY_READ, KEY_ WRITE, KEY_EXECUTE) lub być właścicielem klucza. W celu odczytania systemowej listy ACL proces musi posiadać także przywilej SE_SECURITY_NAME.

- `RegLoadKey()`

Funkcja RegLoadKey importuje do rejestru informacje zawarte w ulu.

Składnia:
LONG RegLoadKey (HKEY hKey, LPCTSTR lpSubKey, LPCTSTR lpFile)

Parametry:

hKey	Uchwyt zwrócony przez funkcję RegConnectRegistry lub jeden z predefiniowanych uchwytów: HKEY_LOCAL_MACHINE, HKEY USERS
lpSubKey	Adres ciągu znaków zawierającego nazwę klucza, który zostanie stworzony wewnątrz hKey.
lpFile	Adres nazwy pliku zawierającego ul.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Plik zawierający ul musi zostać utworzony w wyniku wywołania funkcji RegSaveKey. Proces wywołujący musi mieć uprawnienie SE_RESTORE_NAME.

- `RegNotifyChangeKeyValue()`

Funkcja RegNotifyChangeKeyValue informuje, czy klucz lub któryś z jego podkluczy został zmieniony.

Składnia:

```
LONG RegNotifyChangeKeyValue( HKEY hKey,
                              BOOL bWatchSubtree,
                              DWORD dwNotifyFilter,
                              HANDLE hEvent,
                              BOOL fAsynchronous )
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
bWatchSubtree	Jeśli parametr ma wartość TRUE, monitorowane są zmiany wewnątrz całej gałęzi; w przeciwnym wypadku monitorowane są jedynie zmiany wewnątrz samego klucza.
dwNotifyFilter	Zawiera znaczniki, określające co powinno być zgłaszane: REG_NOTIFY_CHANGE_ANEM, REG_NOTIFY_CHANGE_ATTRIBUTES, REG_NOTIFY_CHANGE_LAST_SET, REG_NOTIFY_CHANGE_SECURITY
hEvent	Identyfikuje zdarzenie zgłaszane, gdy parametr fAsynchronous ma wartość TRUE.
fAsynchronous	Jeśli parametr ma wartość TRUE, zmiany są sygnalizowane poprzez zgłoszenie zdarzenia

hEvent, w przeciwnym wypadku funkcja nie zwraca sterowania, aż do momentu dokonania zmiany.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_INVALID_HANDLE	Klucz znajduje się w zdalnym komputerze.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis
Funkcja nie działa z uchwytami kluczy otwartych w innych komputerach.

- `RegOpenKey()`

Funkcja RegOpenKey otwiera istniejący klucz. Programy Win32 powinny zamiast niej używać funkcji RegOpenKeyEx.

Składnia:

```
LONG RegOpenKey(HKEY hKey,
                LPCTSTR lpSubKey,
                PHKEY phkResult)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Wskaźnik do ciągu znaków określających nazwę klucza, który zostanie otwarty wewnątrz klucza hKey.
phkResult	Adres bufora na zwracany uchwyt.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:

Jeśli klucz nie istnieje, funkcja RegOpenKey nie tworzy nowego klucza. Jeśli jako wartość parametru lpSubKey podasz wartość NULL, funkcja otworzy i zwróci uchwyt do klucza identyfikowanego przez hKey.

- `RegOpenKeyEx()`

Funkcja RegOpenKeyEx otwiera istniejący klucz.

Składnia:
```
LONG RegOpenKeyEx( HKEY hKey,
                   LPCTSTR lpSubKey,
                   DWORD ulOptions,
                   REGSAM samDesired,
                   PHKEY phkResult)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENTJJSER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Wskaźnik do ciągu znaków określających nazwę klucza, który zostanie otwarty wewnątrz klucza hKey.
ulOptions	Musi mieć wartość 0.
samDesired	Opcje dostępu do klucza (patrz tabela A.5)
phkResult	Adres bufora na zwracany uchwyt.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli klucz nie istnieje, funkcja nie tworzy nowego klucza.

Wartości parametru samDesired:

KEY_ALL_ACCESS	Kombinacja uprawnień KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS,

	KEY_NOTIFY, KEY_CREATE_SUB_KEY, KEY_CREATE_LINK oraz KEY_SET_VALUE.
KEY_CREATE_LINK	Zezwolenie na tworzenie symbolicznych łączy.
KEY_CREATE_SUB_KEY	Zezwolenie na tworzenie podkluczy.
KEY_ENUMERATE_SUB_KEYS	Zezwolenie na wyliczanie podkluczy.
KEY_EXECUTE	Zezwolenie na odczyt.
KEY_NOTlFY	Zezwolenie na uaktywnienie powiadamiania.
KEY_QUERY_VALUE	Zezwolenie na odczyt podkluczy.
KEY_READ	Kombinacja uprawnień : KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS oraz KEY_NOTIFY.
KEY_SET_VALUE	Zezwolenie na ustawianie danych.
KEY_WRITE	Kombinacja uprawnień: KEY_SET_VALUE oraz KEY_CREATE_SUB_KEY.

- `RegOueryInfoKey()`

Funkcja RegOueryInfoKey zwraca informacje o kluczu.

Składnia:

```
LONG RegQueryInfoKey(HKEY hKey,
                     LPTSTR lpCIass,
                     LPDWORD lpcbCIass,
                     LPDWORD lpReserved,
                     LPDWORD lpcbSubKeys,
                     LPDWORD lpcbMaxSubKeyLen,
                     LPDWORD lpcbMaxClassLen,
                     LPDWORD lpcbValues,
                     LPDWORD lpcbMaxValueNameLen,
                     LPDWORD lpcbMaxValueLen,
                     LPDWORD lpcbSecurityDescriptor,
                     PFILETIME lpftLastWriteTime)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, HKEY_USERS
lpCIass	Adres bufora na zwracaną nazwę klasy
lpcbCIass	Adres bufora na zwracaną długość nazwy klasy
lpReserved	Musi mieć wartość NULL
lpcbSubKeys	Adres bufora na zwracaną liczbę podkluczy
lpcbMaxSubKeyLen	Adres bufora na zwracaną długość najdłuższej nazwy podklucza
lpcbMaxClassLen	Adres bufora na zwracaną długość najdłuższej nazwy klasy
lpcbValues	Adres bufora na zwracaną liczbę wartości
lpcbMaxValueNameLen	Adres bufora na zwracaną długość najdłuższej nazwy wartości
lpcbMaxValueLen	Adres bufora na zwracany rozmiar najobszerniejszych danych wartości
lpcbSecurity	Adres bufora na zwracaną długość deskryptora ochrony Descriptor
lpftLastWriteTime	Wskaźnik do struktury FILETIME.

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Klucz identyfikowany przez hKey musi być otwarty z ustawionym uprawnieniem KEY_ QUERY_VALUE. To uprawnienie jest zawarte także w stałej KEY_READ.

- `RegQueryMultipleValues()`

Funkcja RegQueryMultipleValues zwraca dane oraz informacje o typie wartości zawartych w kluczu.

Składnia:
LONG RegQueryMultipleValues(HKEY hKey,
PYALENT val_list,
DWORD num_vals,
LPTSTR lpValueBuf,
LPDWORD idwTotsize)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
val_list	Adres tablicy struktur VALENT
num_ vals	Ilość elementów tablicy wskazywanej przez val_list
lpValueBuf	Adres na bufor danych każdej wartości
ldwTotsize	Rozmiar bufora wskazywanego przez lpValueBuf (w bajtach)

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
ERRORCANTREAD	Nie można odczytać tego klucza
ERROR_MORE_DATA	Bufor lpValuebuf jest zbyt mały
ERROR_TRANSFER_TOO_LONG	Dane mają rozmiar ponad l MB
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Ta funkcja działa także w przypadku kluczy otwartych w innym komputerze.

- `RegQueryValue()`

Funkcja RegQueryValue zwraca dane domyślnej wartości klucza.

Składnia:
LONG RegQueryValue(HKEY hKey,
LPCTSTR lpSubKey,
LPTSTR lpYalue,
PLONG lpcbValue)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Adres nazwy pobieranego podklucza
lpValue	Bufor na zwracaną daną domyślnej wartości podklucza
lpcbValue	Bufor na zwracany rozmiar danej

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_MORE_DATA	Bufor lpValue jest zbyt mały
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli bufor jest zbyt mały, funkcja zwraca wartość ERROR_MORE_DATA. Jeśli wskaźnik IpValue ma wartość NULL, funkcja zwraca wartość ERROR_SUCCES, a w buforze lpcbValue umieszcza rozmiar danej. Klucz identyfikowany przez hKey musi być otwarty z ustawionym uprawnieniem KEY_QUERY_VALUE.

- `RegQueryValueEx()`

Funkcja RegQueryValueEx zwraca dane wartości klucza.

Składnia:

```
LONG RegQueryValueEx(HKEY hKey,
                     LPTSTR lpValueName,
                     LPDWORD lpReserved,
                     LPDWORD lpType,
                     LPBYTE lpData,
                     LPDWORD lpcbData)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT,

	HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpValueName	Adres nazwy pobieranej wartości
lpReserved	Musi mieć wartość 0
lpType	Typ danych
lpData	Bufor na zwracane dane
lpcbData	Bufor na rozmiar zwracanych danych

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
ERROR_MORE_DATA	Bufor lpData jest zbyt mały
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli bufor jest zbyt mały, funkcja zwraca wartość ERROR_MORE_DATA. Jeśli wskaźnik lpData ma wartość NULL, funkcja zwraca wartość ERROR_SUCCES, a w buforze lpcbDa ta umieszcza rozmiar danej. Klucz identyfikowany przez hKey musi być otwarty z ustawionym uprawnieniem KEY_QUERY_VALUE.

Wartości parametru IpType:

REG_BINARY	Dane binarne
REG_DWORD	Liczba 32-bitowa
REG_DWORD_LITTLE_ENDIAN	Liczba 32-bitowa w formacie little-endian
REG_DWORD_BIG_ENDIAN	Liczba 32-bitowa w formacie big-endian
REG_EXPAND_SZ	Ciąg znaków zakończony znakiem Null, zawierający odwołanie do zmiennej środowiskowej
REG_LINK	Symboliczne łącze Unicode
REG_MULTI_SZ	Tablica ciągów znaków oddzielonych znakami Null
REG_NONE	Nieokreślony typ danych
REG_RESOURCE_LIST	Lista zasobów sterowników urządzeń
REG_SZ	Ciąg znaków zakończony znakiem Null

- `RegSetKeySecurity()`

Funkcja RegSetKeySecurity ustawia atrybuty ochrony otwartego klucza.

Składnia:
LONG RegSetKeySecurity(HKEY hKey,
SECURITY_INFORMATION SecurityInformation,
PSECURITY_DESCRIPTOR pSecurityDescriptor)

Parametry:

hKey	Uchwyt otwartego klucza
SecurityInformation	Określa zawartość deskryptora ochrony
pSecurityDescriptor	Adres atrybutów ochrony

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Ustawienie właściciela lub grupy właścicieli kluczy wymaga, by wywołujący proces posiadał przywilej WRITE_OWNER. Ustawienie dyskretnej listy ACL klucza wymaga, by wywołujący proces posiadał uprawnienie WRITE_DAC lub był właścicielem klucza. Ustawienie systemowej listy ACL klucza wymaga, by wywołujący proces posiadał uprawnienie SE_SECURITY_NAME.

- `RegReplaceKey()`

Funkcja RegReplaceKey zastępuje innym plikiem plik zawierający ul rejestru.

Składnia:
LONG RegReplaceKey(HKEY hKey,
LPCTSTR lpSubKey,
LPCTSTR lpNewFile,
LPCTSTR lpOldFile)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Uchwyt jednego z predefiniowanych głównych kluczy: HKEY_LOCAL_MACHINE, HKEY_USERS
lpNewFile	Nazwa nowego pliku
lpOldFile	Nazwa pliku, który ma zostać zastąpiony

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Proces wywołujący musi posiadać przywilej SE_RESTORE_NAME.

- `RegRestoreKey()`

Funkcja RegRestoreKey importuje do rejestru pozycje z pliku i zastępuje wskazany klucz.

Składnia
LONG RegRestoreKey(HKEY hKey, LPCTSTR IpFile, DWORD dwFlags)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpFile	Nazwa importowanego pliku
dwFlags	Wskazuje, czy klucz jest ulotny: REG_WOLE_HIVE_VOLATILE

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli jakikolwiek podklucz klucza hKey jest otwarty, wywołanie funkcji nie powiedzie się. Proces wywołujący musi posiadać przywilej SE_RESTORE_NAME. Funkcja RegRestoreKey zastępuje całe poddrzewo rejestru nowym poddrzewem importowanym z pliku.

- `RegSaveKey()`

Funkcja RegSaveKey zapisuje do pliku zawartość klucza i wszystkich jego podkluczy.

Składnia:
LONG RegSaveKey(HKEY hKey, LPCTSTR lpFile,
LPSECURITY_ATTRIBUTES lpSeurityAttnbutes)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT,HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, HKEY_USERS
lpFile	Nazwa pliku, do którego zostanie zapisana zawartość klucza i jego podkluczy
lpSecuntyAttributes	Deskryptor ochrony nowego pliku

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Proces wywołujący musi posiadać przywilej SE_BACKUP_NAME.

- `RegSetValue()`

Funkcja RegSetValue ustala domyślną wartość klucza rejestru. Programy Win32 powinny zamiast niej korzystać z funkcji RegSetValueEx.

Składnia:
LONG RegSetValue(HKEY hKey,
LPCTSTR lpSubKey,
DWORD dwType,
LPCTSTR lpData,
DWORD cbData)

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Nazwa podklucza
dwType	Musi mieć wartość REG_SZ
lpData	Adres ciągu znaków zakończonego znakiem Null
cbData	Długość ciągu znaków wskazywanego przez IpData

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli klucz określony przez lpSubKey nie istnieje, zostanie on stworzony. Klucz identyfikowany przez hKey musi być otwarty z ustawionym znacznikiem dostępu KEY_SET_VALUE.

- `RegSetValueEx()`

Funkcja RegSetValueEx ustala wartość klucza rejestru.

Składnia:
```
LONG RegSetValueEx(HKEY hKey,
                   LPCTSTR lpValueName,
                   DWORD Reserved,
                   DWORD dwType,
                   CONST BYTE *lpData,
                   DWORD cbData)
```

Parametry:

hKey	Uchwyt otwartego klucza lub jeden z następujących zarezerwowanych uchwytów: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE, HKEY_USERS
Reserved	Musi mieć wartość 0.
dwType	Określa typ danych wartości
lpData	Adres danych do zapisania wartości
cbData	Długość zapisywanych danych

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Jeśli klucz określony przez lpSubKey nie istnieje, zostanie on stworzony. Klucz identyfikowany przez hKey musi być otwarty z ustawionym znacznikiem dostępu KEY_SET_VALUE.

Wartości parametru dwType:

REG_BINARY	Dane binarne
REG_DWORD	Liczba 32-bitowa
REG_DWORD_LITTLE_ENDIAN	Liczba 32-bitowa w formacie little_endian
REG_DWORD_BIG_ENDIAN	Liczba 32-bitowa w formacie big_endian

REG_EXPAND_SZ	Ciąg znaków zakończony znakiem Null, zawierający odwołanie do zmiennej środowiskowej
REG_LINK	Symboliczne łącze Unicode
REG_MULTI_SZ	Tablica ciągów znaków oddzielonych znakami Null
REG_NONE	Nieokreślony typ danych
REG_RESOURCE_LIST	Lista zasobów sterowników urządzeń
REG_SZ	Ciąg znaków zakończony znakiem Null

- `RegUnLoadKey()`

Funkcja RegUnLoadKey zwalnia z rejestru wskazany klucz łącznie z podkluczami.

Składnia:
LONG RegUnLoadKey(HKEY hKey, LPCTSTR lpSubKey)

Parametry:

hKey	Uchwyt zwrócony przez funkcje RegConnectRegistry lub jeden predefiniowanych uchwytów: HKEY_LOCAL_MACHINE, HKEY_USERS
lpSubKey	Nazwa zwalnianego podklucza

Zwracane wartości:

ERROR_SUCCESS	Funkcja poprawnie zakończyła działanie.
kod błędu	Wywołanie funkcji nie powiodło się.

Opis:
Proces wywołujący musi posiadać uprawnienie SE_RESTORE_NAME. Klucz wskazywany przez lpSubKey musi być stworzony przy pomocy funkcji RegLoadKey.

Wnioski

Rejestr jest niewątpliwie najważniejszym składnikiem systemu operacyjnego Windows. Dzięki niemu poprawiono i ulepszono integrację aplikacji z systemem, scentralizowano dane o aplikacjach i ustawieniach systemowych, umożliwiono kontrolę nad wersjami instalowanych składników. Niestety nie ma nic za darmo. Umieszenie tych danych w kilku plikach stwarza potencjalne zagrożenie – w razie uszkodzenia tych zbiorów cały system może nie startować lub zachowywać się co najmniej dziwnie. Pomimo systemowych zabezpieczeń przed taką sytuacją nie można wykluczyć całkowicie takiej sytuacji. Sprawę pogarsza fakt że, Microsoft nie kładzie na ten problem należytego nacisku, ilu użytkowników słyszało o rejestrze poza obiegowymi opiniami ? Polityka producenta z pod znaku Windows jest nieco zagmatwana. Dostarcza narzędzia do ingerencji w rejestr systemowy a zarazem nie prowadzi informacji serwisowych na jego temat. Większość użytkowników gdy ma problemy z systemem po prostu instaluje go ponownie, a czasem by przywrócić Windows do stanu poprzedniego wystarczy kilka zmian w rejestrze, po prostu trzeba sobie z tego zdawać sprawę. O ile ponowna instalacja systemu na stacji roboczej czy komputerze do użytku domowego nie jest wielkim problemem to instalowanie ponownie systemu na serwerze np. z powodu pogubienia skojarzeń plików mija się z celem. Takie praktyki się niestety zdarzają.

Po zapoznaniu się z tą pracą przynajmniej część spraw na temat rejestru systemowego Windows powinna być trochę jaśniejsza. Należy jeszcze raz podkreślić że, rejestr nie jest zbiorem wiedzy niedostępnej dla przeciętnego użytkownika, to tylko kwestia zapoznania się z jego budową. To prawda że, zawiera wiele niuansów, ale nie należy się tym przejmować wiedza rośnie w miarę jej pogłębiania. Nie zachęca się natomiast do bezmyślnego zmieniania i eksperymentowania na rejestrze, bo na sto takich przypadków dziewięćdziesiąt dziewięć prowadzi do katastrofalnych skutków. Na początek do wnoszenia zmian należy korzystać z gotowych narzędzi, jednym z nich jest napisany na potrzeby tej pracy *"Tipser.exe"* który pozwala na automatyczne zmiany niektórych ustawień systemowych, takich jak kolory, ikony, kursory i wiele innych. Jako edytor rejestru można wykorzystać natomiast *"Skaner.exe"* który w odróżnieniu od swojego odpowiednika systemowego pozwala cofnąć krytyczne czynności na rejestrze. Innym programem do wnoszenia zmian jest *"WinAdmin.exe"* zarządzający skojarzeniami plików. Jest też wiele innych programów tego typu dostępnych na rynku, zarówno shareware jak i wersji komercyjnych. Nie sposób wszystkich tu przytoczyć zwłaszcza że, rynek oprogramowania tego typu rośnie z miesiąca na miesiąc.

Podsumowując , nie bójmy się rejestru, po zapoznaniu się z nim możemy mieć wiele korzyści, jest on w końcu sercem wiedzy o systemie, więc najważniejszą częścią z punktu widzenia zarówno programistów jaki administratorów.

Literatura

1 Barkakati Naba "Biblia Turbo C++" LT&P
Warszawa 1995
2 Honeycutt Jerry "Rejestry Windows 95/NT 4 - Czarna księga" HELION
Warszawa 1998
3 Karmański Janusz "Praktyczny kurs programowania w Windows 95" HELION Warszawa 1997
4 Reisdroph Kent "C++ Builder 3.0" Helion
Warszawa 1998
5 Tidrow Rob "Rejestr Windows 95" RM
Warszawa 1996
6 Magazyny komputerowe :
"PC WORLD KOMPUTER"
11/98 artykuł "Windows 95/98 na miarę potrzeb"
6/99 artykuł "40 sposobów na Windows"
"Chip"
4/99 artykuł "Alternatywne powłoki Windows"
7 Artykuły w wydaniu elektronicznym:
"reg95.hlp" Imaginations Unlimited ©1996
"regEntry.hlp" Microsoft ©1985-1996

Dodatki

1. Instrukcja instalacji i obsługi pakietu Windows Administrator
2. Płyta CD-ROM z wersją instalacyjną pakietu oraz kodami źródłowymi
3. Wersja elektroniczna pracy dyplomowej i instrukcji obsługi