

Grupa: UZUPEŁNIENIA

Notepad++ 4.1.2

edytor dla programistów
plik kurs_skrypt02.vbs – materiały
dodatkowe do bieżącego odcinka
Artykuł z pierwszym odcinkiem kursu
z PC Format 8/2007

Odczytywanie z rejestru

Skrypty to potężne narzędzie nie tylko dla administratorów Windows. Dzięki nim możesz łatwo modyfikować dowolne ustawienia systemu. Tym razem poznasz sposoby modyfikacji rejestru Windows.

W poprzednim odcinku naszego kursu zaprezentowaliśmy Windows Script Host – wbudowany w system Windows mechanizm interpretujący skrypty. Pokazaliśmy także, jakich narzędzi programistycznych – linii komend i zwykłego Notatnika Windows – może używać początkujący skrypter (pierwszy odcinek kursu znajdziesz w wersji elektronicznej na płycie dołączonej do bieżącego wydania PC Formatu). Tym razem na początek proponujemy niewielką zmianę. Dla wygody pisania skryptów warto zamienić systemowy Notatnik na program Notepad++ 4.1.2., który zamieszczamy na naszym CD.

Pierwszy prosty skrypt powinien mieć już za sobą. Wiesz również, jaka powinna być struktura skryptu, ale ów pierwszy skrypt był na tyle prosty, że nie znalazło się w nim miejsce na wszystkie odpowiednie elementy struktury. Teraz to zmienimy. Zajmiemy się tworzeniem pełnowartościowego skryptu VBScript, za którego pomocą będziesz mógł odczytywać wartości z rejestru Windows. Do tego celu posłużysz się zestawem zmiennych.

Zmienne w skrypcie

Czym są zmienne? W kontekście programowania są one czymś w rodzaju skrótów odwołujących się do frag-

mentu pamięci komputera, w którym są przechowywane informacje potrzebne programowi i zmieniane w trakcie działania skryptu. Aby skorzystać ze zmiennej w skrypcie, wystarczy ją zadeklarować.

Zmienne VBScript i typy danych

W językach programowania różni się typy zmiennych. Ich podział może być różny w zależności od tego, czy zmienne dzielimy ze względu na zakres ważności (widoczności zmiennej w kodzie programu; np. zmienna lokalna lub zmienna globalna), ze względu na sposób ich inicjowania (wypełnienia fragmen-

tu pamięci odpowiednią wartością; zmienna statyczna lub dynamiczna), czy ze względu na typ danych, jakie reprezentują. W zależności od języka programowania rozróżnia się wiele typów danych. Język VBScript dopuszcza tylko jeden typ – **Variant**. Typ ten może przechowywać wiele różnych rodzajów informacji (liczby całkowite, tekst itd.). Tym samym funkcje VBScript zwracają także jeden typ danych – właśnie **Variant**. W jaki sposób jeden typ danych przekłada się na różnego typu informacje przechowywane w zmiennych bądź zwracane przez funkcje skryptu? Otóż typ **Variant** zawiera szereg podtypów, ale na obecnym etapie naszego kursu nie będziemy się nimi jeszcze zajmować.

Oceń lub skomentuj na **PC Format ocena**
➔ www.pcformat.pl/ocena

➔ Nagłówek skryptu i deklaracja zmiennych

Na pierwszym etapie tworzenia skryptu powstaje nagłówek skryptu. Co prawda nie jest on niezbędny do działania skryptu (przynajmniej w większości wypadków), ale warto, by każdy tworzony skrypt zawierał nagłówek. Nagłówek czyni cały skrypt czytelniejszym, gdyż od razu widać, jakie zmienne są w skrypcie zadeklarowane, a także, w jaki sposób skrypt jest uruchamiany.

1 Uruchom program Notepad++ i wpisz listing 1 lub z naszego krążka dołączonego do bieżącego wydania wczytaj plik kurs_skrypt02.vbs.

2 Wiersze od 1 do 5 to właśnie nagłówek (ang. header) skryptu. Mimo pozorowanej komplikacji tak naprawdę zawarte są tutaj tylko trzy polecenia.

listing 1

```

1 Option Explicit
2 On Error Resume Next
3 Dim ShellObject
4 Dim regActiveComputerName, regComputerName, regHostname
5 Dim ActiveComputerName, ComputerName, Hostname
6
7 regActiveComputerName = "HKLM\SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName\ComputerName"
8 regComputerName = "HKLM\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName\ComputerName"
9 regHostname = "HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Hostname"
10
11 Set ShellObject = CreateObject("WScript.Shell")
12 ActiveComputerName = ShellObject.RegRead(regActiveComputerName)
13 ComputerName = ShellObject.RegRead(regComputerName)
14 Hostname = ShellObject.RegRead(regHostname)
15
16 WScript.Echo "Nazwa aktywnego komputera to: " & ActiveComputerName
17 WScript.Echo "Nazwa komputera to: " & ComputerName
18 WScript.Echo "Nazwa hosta to: " & Hostname

```

skrypt, że ma uznawać za zmienne tylko takie wyrażenia, które zostały wcześniej zadeklarowane. Użycie wyrażenia **Option Explicit** w nagłówku nie jest konieczne, ale gdyby zo-

stało pominięte, wtedy interpreter VBScriptu podczas wykonywania kolejnych linii skryptu uznawałby za zmienne każde wyrażenie, którego nie rozumie, tzn. które nie jest poleceniem języka VBScript.

3 Pierwszy wiersz zaprezentowanego listingu 1 zawiera wyrażenie, które informuje

4 Kolejny wiersz 2 to przykład wyrażenia umieszczanego w nagłówku skryptu, ste-

rującego jego wykonaniem. W tym konkretnym przypadku wyrażenie **On Error Resume Next** informuje komputer, jak ma postąpić, gdy podczas wykonania skryptu napotka jakikolwiek błąd. Jeśli tak się zdarzy, błędna linia zostanie zignorowana, a skrypt będzie uruchamiany dalej od kolejnej linii. Tego typu praktyka jest określana mianem obsługi błędów (ang. error handling). Inna sprawa, że w przypadku, gdy dopiero tworzysz skrypt, zalecanym rozwiązaniem jest nie wpisywanie tego wyrażenia, gdyż „ukrywa” ono możliwe błędy, jakie mogą się pojawić podczas działania skryptu. Zwrot **On Error Resume Next** należy umieszczać w nagłówku dopiero wtedy, kiedy skrypt jest już ukończony i poprawnie działa.

➔ Przypisanie wartości zmiennym

Kolejny fragment skryptu to część, w której zadeklarowanym wcześniej zmiennym przypisujesz potrzebne wartości. Ten fragment skryptu to tzw. część referencyjna. O ile nagłówki skryptu można teoretycznie pominąć, o tyle przypisanie wartości zmiennym jest niezbędne.

1 Zakładamy, że program Notepad++ masz już uruchomiony i wpisałeś już listing 1 lub utworzyłeś plik `kurs_skrypt02.vbs`.

2 W linii 7 zmiennej `regActiveComputerName` przypisujesz odpowiedni klucz rejestru, w którym jest przechowywana nazwa aktywnego komputera.

5 Następane trzy wiersze (od 3 do 5) to deklaracje zmiennych. Zauważ, że instrukcją informującą komputer o tym, że ma do czynienia ze zmienną, jest **Dim**. Nazwy zmiennych mogą być dowolne. Jednak w celu zapewnienia czytelności warto wpisywać nazwy odzwierciedlające dane, które mają być reprezentowane przez te zmienne. W naszym skrypcie nazwy zmiennych są po prostu nazwami kluczy rejestru, z których będziesz za pomocą skryptu odczytywał dane. Wyjątkiem jest zmienna o nazwie `ShellObject`, którą zadeklarowano w celu późniejszego jej wykorzystania do utworzenia obiektu, za którego pośrednictwem odczytasz wybrane klucze rejestru. Deklaracje zmiennych kończą część nagłówkową skryptu.

3 W kolejnych dwóch liniach, 8 i 9, w analogiczny sposób przypisywane są wartości (klucze rejestru) kolejnym dwóm zmiennym.

4 Warto przy okazji zauważyć, że te trzy zmienne pełnią w tworzonego skrypcie rolę stałych, gdyż ich wartość nie ulegnie zmianie w wyniku działania samego skryptu. W tym przypadku masz zatem do czynienia ze zmiennymi statycznymi. A mówiąc prościej, z czymś w rodzaju skrótów do bardzo długich informacji. W dalszej części skryptu łatwiej będzie się odwoływać do czytelnej nazwy zmiennej niż do długiego ciągu definiującego dany klucz rejestru.

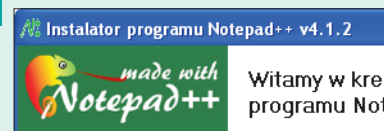
Jak zainstalować program Notepad++ 4.1.2

1 Kliknij dwukrotnie plik `npp4.1.2.Installer.exe` z naszej płyty (lub skorzystaj z menu płyty i wywołaj instalację). Jeżeli pojawi się okno:

Otwórz plik - ostrzeżenie o zabezpieczeniach, kliknij przycisk **Uruchom**.

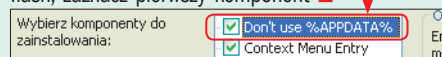
2 W oknie **Installer Language**, z rozwijanej listy wybierz **Polski** i kliknij **OK**.

3 W oknie:



kliknij **Dalej >**, a następnie zatwierdź tekst licencji (program jest darmowy), klikając **Zgadzam się**.

4 Kliknij **Dalej >**. Przejdiesz do etapu wyboru komponentów do zainstalowania. Jeżeli instalujesz Notepad++ na dysku twardym, pozostaw domyślnie zaznaczone komponenty. Jeżeli jednak chcesz mieć ten edytor zainstalowany np. na pamięci flash, zaznacz pierwszy komponent



i dopiero wtedy kliknij **Zainstaluj**.

5 Gdy instalator przekopiuje pliki, kliknij **Zakończ**. Edytor uruchomi się automatycznie.

➔ Część robocza – właściwe działanie skryptu

Do tej pory poznałeś już dwie pierwsze części skryptu, ale z pewnością zauważyłeś, że mimo napisania już kilku linii skrypt wciąż jeszcze nic nie robi. Najwyższy czas zaprząć skrypt do pracy.

1 Przyjrzyj się linii 11 w skrypcie. Zauważ, że zaczyna się ona poleceniem **Set**. Polecenie to, w uproszczeniu, przypisuje zmiennej odwołanie do jakiegoś obiektu. W tym przypadku obiektem będzie **WScript.Shell**, a odpowia-

dającą mu zmienną – **ShellObject**. Po co potrzebny nam obiekt? Z założenia skrypt ma odczytać jakieś informacje z rejestru systemowego. Toteż konieczny jest element, który umożliwi skryptowi **VBScript** odczytywanie tych informacji, a odczyt ten nie będzie możliwy, jeżeli skrypt nie będzie miał „połączenia” z rejestrem.

2 Kolejne trzy linie to właściwy „silnik” skryptu. Wykorzystując kolejne trzy wcześniej

zadeklarowane zmienne, przypisujesz im wartość odczytaną (za pomocą metody `RegRead` obiektu `WScript.Shell` reprezentowanego przez zmienną `ShellObject`) z rejestru, a ściślej z kluczy wskazywanych przez zmienne, którym wartość przypisałeś w części referencyjnej.

3 Skrypt potrafi już odczytywać dane z wybranych (wskazywanych przez zmienne) kluczy rejestru.

➔ Informacje wyjściowe – wynik działania

Gdyby na poprzednich etapach zakończyć pracę, skrypt, owszem, wykonałby swoje działanie, ale użytkownik nie zobaczyłby żadnych wyników.

1 W linii 16 ponownie pojawi się, tym razem już wywołany bezpośrednio (a nie poprzez wcześniej zadeklarowaną zmienną), obiekt **WScript**. Tym razem, w celu wyświetlenia informacji na ekranie, używasz metody `Echo`. Najpierw wyświetlana jest wartość przechowywana przez zmienną `ActiveComputerName` (odczytana z rejestru w linii 12), a następnie – po znaku łącznika – informacja tekstowa zawarta w cudzysłowach.

2 W kolejnych dwóch liniach stosujesz ten sam zabieg w celu wyświetlenia wyników, ale w odniesieniu do pozostałych dwóch zmiennych przechowujących wartości odczytane z rejestru.

3 Skrypt jest gotowy. Jego wykonanie (przypomnienie: jeżeli chcesz wykonać skrypt w oknie wiersza poleceń, użyj polecenia `cscript nazwa_pliku.vbs`; jeśli zaś chcesz zobaczyć wy-

nik w oknie dialogowym, użyj polecenia `wscript nazwa_pliku.vbs`) spowoduje, że na ekranie wyświetlą się następujące po sobie kolejno informacje:

