



Zostań skrypterem!

Dzięki skryptom wiele zadań w Windows można wykonać automatycznie. Korzystasz z funkcji Worda czy prowadzisz obliczenia w Excelu bez potrzeby uruchamiania tych programów w klasyczny sposób. Masz dostęp do ustawień sieciowych, rejestru, każdego pliku oraz foldera i możesz na nich wykonywać niemal dowolne operacje.

Skrypty to, w uproszczeniu, ciągi poleceń wywołujące określone funkcje systemu i zainstalowanych w nim aplikacji w celu realizacji naszych zadań. System Windows jest wyposażony w mechanizm umożliwiający sterowanie systemem za pośrednictwem skryptów. Mowa o programie Windows Script Host (dawniej Windows Scripting Host). Program ten umożliwia interpretację skryptów i wykonuje napisane w nich akcje, a zarazem nie wiąże użytkownika z jakimś jednym wybranym językiem programowania skryptów. Standardowo zainstalowany Windows XP współpracuje – poprzez Windows Script Host – z językami Visual Basic Script (VBScript) oraz JScript. Jednak mechanizm WSH w żaden sposób nie wymusza korzystania z tych dwóch języków. Jeżeli ktoś zna lepiej np. Pythona czy Perlą – języki skryptowe popularne zwłaszcza wśród użytkowników Linuksa, WSH odczyta także kod napisany w tych językach, pod warunkiem instalacji odpowiedniego, wybranego języka. W artykule skupimy się na wykorzystaniu możliwości WSH za pomo-

cą skryptów napisanych w języku VBScript.

Dwa programy

Windows Script Host udostępni użytkownikowi do uruchamiania skryptów dwa programy: CSCRIPT oraz WSCRIPT. Pierwszy to narzędzie umożliwiające uruchamianie skryptów pozbawione własnego interfejsu graficznego, działające z linii komend. Najlepiej przydaje się do wykonywania skryptów, które mają działać właśnie tylko w obrębie linii komend lub całkowicie w tle. Jeżeli chcesz pisać skrypty, które będą wymagały interakcji z użytkownikiem, z pewnością będziesz korzystał z drugiego programu – WSCRIPT – uruchamiającego skrypty w graficznym interfejsie.

Programy interpretowane

Skrypty są niekiedy nazywane programami interpretowanymi. W odróżnieniu od aplikacji kompilowanych, z którymi masz na co dzień do czynienia, interpreter skryptu wykonuje zawarte w nim polecenia linia po linii. Czytelnicy, którzy mieli kiedykolwiek do czynienia

z plikami wsadowymi (pliki z rozszerzeniem .bat) Windows, z pewnością zauważą analogię. Skrypty jednak potrafią więcej od plików wsadowych, głównie dzięki temu, że możliwe jest w nich korzystanie z tzw. obiektów.

Czym jest obiekt?

W kontekście programowania skryptów (a także programowania w ogóle) obiekt to – w uproszczeniu – prosty zbiór funkcji wykonujących podobne zadania. Windows Script Host umożliwia korzystanie z tzw. obiektów COM (Component Object Model). Dzięki obiektom COM twórca skryptu ma dostęp w zasadzie do pełnego interfejsu programowania aplikacji Windows (Windows API – Application Programming Interface), czyli dokładnie tego samego zestawu funkcji, jaki ma do dyspozycji programista tworzący kod kompilowany. Jednak tworzenie skryptów jest o tyle prostsze, że nie musisz obudowywać skryptu wszystkimi składnikami interfejsu i funkcji, w które musi być wyposażona pełnowartościowa, napisana od podstaw aplikacja Windows.

Linia komend i notatnik

Te dwa programy: CMD oraz standardowy Notatnik Windows, to najczęściej używane narzędzia podczas pisania skryptów – przynajmniej w początkowej fazie ich tworzenia. Dlatego już teraz warto, byś ustawił skróty do tych dwóch programów w dogodnym dla siebie miejscu, np. na pulpicie Windows.

Łatwiejsze uruchamianie skryptów

Jeżeli zapoznałeś się z treścią artykułu, wiesz, że aby uruchomić skrypt, należy poprzedzić jego nazwę poleceniem CSCRIPT lub WSCRIPT. Nie jest to konieczne, gdyż Windows przyjmuje domyślnie, że skrypty są uruchamiane za pomocą narzędzia WSCRIPT. Zatem jeżeli wpiszesz samą nazwę skryptu w linii komend, najprawdopodobniej zobaczysz okienko dialogowe z wynikiem działania danego skryptu. Ponieważ większość skryptów łatwiej uruchamiać z linii komend (czyli za pomocą programu CSCRIPT), można tak skonfigurować system, by skrypt był uruchamiany domyślnie za pomocą właśnie CSCRIPT, a nie WSCRIPT. Wystarczy, że w linii komend wpiszesz polecenie:

```
cscript //no logo //h:cscript //s
```

Obiekty COM nie są jedynymi obiektami, z których możesz korzystać, tworząc skrypty z wykorzystaniem Windows Script Hosta. Do twojej dyspozycji pozostają jeszcze wewnętrzne obiekty WSH, jak choćby najczęściej używany obiekt WScript (mimo podobieństwa nazwy mówimy tu o obiekcie, a nie o wcześniej wspomnianym programie WSCRIPT) wywołujący za każdym razem, gdy tylko uruchamiasz skrypt. W trakcie naszego kursu poznasz także wiele innych obiektów.

Struktura skryptu

Zasadniczo każdy skrypt składa się z czterech części: sekcji nagłówka (nie jest on konieczny, ale czyni skrypt czytelniejszym) zawierającego deklaracje zmiennych, sekcji referencji zawierającej przypisanie wartości zmiennym, sekcji roboczej, w której zawarty jest kod jakiegoś działania, które skrypt ma wykonać. Ostatnia sekcja informacji wyjściowych zawiera kod wyświetlający wynik działania skryptu użytkownikowi. Wychodząc z założenia, że najlepszym sposobem nauki jest poznawanie przedmiotu na przykładach, czas spróbować swoich sił w pierwszym skrypcie.

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

➔ Pisanie pierwszego skryptu

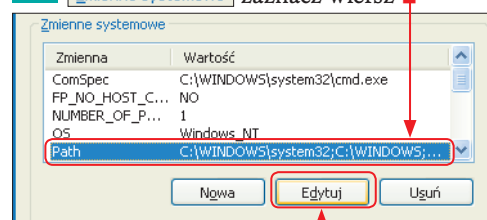
Na początek proponujemy bardzo prosty skrypt, którego wynikiem będzie wyświetlenie nazwy sieciowej twojego komputera. Napisanie pierwszej linii skryptu zacznij jednak od niewielkich przygotowań, dzięki którym pisanie i uruchamianie skryptów będzie łatwiejsze. Przede wszystkim warto zmodyfikować zmienną środowiskową PATH, z której system korzysta podczas przeszukiwania poleceń wpisanych w oknie wiersza polecenia. Skoro będziesz tworzył skrypty, dobrym pomysłem jest globalna modyfikacja tej zmiennej tak, by do uruchomienia skryptu nie było konieczne podawanie całej ścieżki dostępu do foldera, w którym dany skrypt się znajduje.

Ustawianie zmiennej środowiskowej PATH

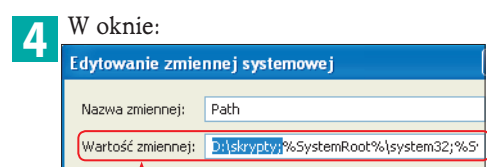
1 Korzystając z Eksploratora Windows lub innego menedżera plików, którego na co dzień używasz, utwórz na dysku twardym folder, w którym będziesz zapisywał tworzone skrypty, i nazwij go np. **Skrypty**.

2 Wciśnij skrót klawiszowy [Windows+Pause Break]. Zobaczysz okno **Właściwości systemu**. Kliknij w nim zakładkę **Zaawansowane**, a następnie przycisk **Zmienne środowiskowe**.

3 W oknie **Zmienne środowiskowe**, w sekcji **Zmienne systemowe** zaznacz wiersz



i kliknij przycisk



w polu dopisz na początku ścieżkę do utworzonego w kroku 1 foldera na skrypty. W naszym przykładzie (na ilustracji) jest to zapis **D:\skrypty**; (pamiętaj, by ścieżkę zakończyć zna-

kiem średnika). Kliknij **OK** i jeszcze dwukrotnie **OK**.

Kod pierwszego skryptu

1 Uruchom Notatnik Windows (wybierając jego ikonę z menu **Start**) i wpisz następujące dwie linie:

```
Set WshNetwork = CreateObject(WScript.Network)
WScript.Echo WshNetwork.ComputerName
```

2 Z menu **Plik** wybierz **Zapisz**. Następnie w oknie **Zapisz jako** przejdź do utworzonego wcześniej foldera dla skryptów. Z listy **Zapisz jako typ** wybierz **Wszystkie pliki**. W polu **Nazwa pliku** wpisz nazwę skryptu, np. **kompname.vbs**. Kliknij przycisk **Zapisz**.

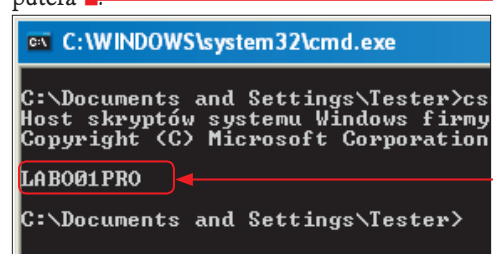
3 Uruchom okno linii komend. Jeżeli wcześniej nie utworzyłeś ikonki skrótu, linię komend możesz wywołać, wybierając z menu **Start** pozycję **Uruchom...**. Następnie wpisz w oknie **Uruchamianie** w polu **Otwórz** polecenie **cmd** i kliknij **OK**.

4 Ponieważ wcześniej zmieniłeś zmienną środowiskową PATH, nie musisz podawać pełnej ścieżki dostępu do pliku skryptu. Aby uruchomić skrypt, w linii poleceń wpisz:

cscript kompname.vbs.

Oczywiście zamiast „kompname.vbs” wpisz nazwę, pod jaką zapisałeś powyższy skrypt do pliku.

5 Wynikiem działania skryptu będzie wyświetlona w linii komend nazwa komputera.



➔ Dodatkowe języki i pliki

W artykule wspomnieliśmy, że choć Windows Script Host wbudowany w Windows XP rozpoznaje skrypty napisane w językach VBScript oraz JScript (podobny do JavaScript język opracowany przez Microsoft - jest rozpoznawany nie tylko przez Windows Script Hosta, ale także przez Internet Explorera), to możesz korzystać i poznawać zalety innych języków programowania skryptów. Oto one:

➔ PERL

Język popularny wśród użytkowników Linuksa. Ma rozbudowane funkcje manipulowania tekstem. W internecie można znaleźć mnóstwo gotowych skryptów. Aby skrypty PERL-a zadziałały w Windows, musisz zainstalować np. bezpłatną wersję o nazwie ActivePerl; znajdziesz ją na stronie: www.activestate.com

➔ Python

Język obiektowy, wtyczkę do Windows Script Hosta o nazwie ActivePerl znajdziesz także na: www.activestate.com

UWAGA! Jeżeli wciąż korzystasz z którejś wersji systemu Windows, 98, Me lub NT, to aby móc używać skryptów, musisz doinstalować do systemu mechanizm Windows Script Host. Pobierz go ze strony: <http://www.microsoft.com/downloads/details.aspx?FamilyId=0A8A18F6-249C-4A72-BFCF-FC6A-F26DC390&displaylang=en>

Przed pobraniem musisz przejść procedurę sprawdzenia legalności wersji systemu Windows, której używasz.

6 Krótkie wyjaśnienie kodu tego skryptu.

W pierwszej linii określasz zmienną o nazwie **WshNetwork**, którą przywiązujesz do wbudowanego w WSH obiektu o nazwie **WScript.Network**. W ostatniej linii używasz metody **Echo** obiektu **WScript** do wyświetlenia właściwości **ComputerName** zmiennej **WshNetwork**.

7 Ten sam skrypt możesz wywołać także poleceniem:

wscript kompname.vbs.

Pierwsze modyfikacje

1 Wypróbuj jeszcze jeden manewr. Otwórz napisany poprzednio skrypt do edycji (oczywiście w Notatniku Windows) i między pierwszą a drugą linię wstaw:

WScript.Echo „Nazwa komputera to: „

2 Zapisz plik i wypróbuj jego uruchomienie zarówno w linii komend, jak i w trybie okienkowym. W drugiej linii za pomocą metody **Echo** obiektu **WScript** żadasz od skryptu, by wyświetlił komunikat tekstowy zawarty w cudzysłowach.

3 Jeżeli wywołałeś zmodyfikowany skrypt także za pomocą narzędzia **WSCRIPT**, z pewnością zauważyłeś, że nazwa komputera i komunikat tekstowy wyświetlają się w dwóch oknach. Jak tego uniknąć? To bardzo proste. Użyj znaku łącznika (&) i zmodyfikuj linie 2. i 3. w następującą jedną linię:

WScript.Echo „Nazwa komputera to: „ & WshNetwork.ComputerName

4 Gdy po takiej modyfikacji uruchomisz skrypt, zobaczysz, że komunikat informujący o nazwie komputera pojawi się już w całości w jednym oknie.

➔ Dodatkowe informacje i źródła wiedzy o skryptach w Windows

Jeżeli jesteś poważnie zainteresowany tworzeniem skryptów, to nasz wstępny artykuł z pewnością nie zaspokoili twojej ciekawości. Dlatego oprócz zaplanowanej przez nas ścieżki dydaktycznej, jaką będziemy zmierzać w kolejnych wydaniach PC Formatu, prezentując następne części kursu tworzenia skryptów, proponujemy dodatkową lekturę. Zamieszczamy spis witryn internetowych oraz książek dotyczących „skryptowania” w Windows. Znajdziesz tam informacje, dzięki którym będziesz mógł pogłębić wiedzę, jaką wyniesiesz z lektury PC Formatu.

Przydatne strony WWW:

➔ <http://www.vb4all.pl/>

Polskojęzyczna witryna poświęcona językowi programowania Visual Basic. Część zawartych na stronie treści dotyczy VBScript.

➔ <http://msdn2.microsoft.com/en-us/library/ms950396.aspx>

Witryna (angielskojęzyczna) Microsoft Developer Network zawiera bardzo bogaty dział, który jest poświęcony tylko i wyłącznie skryptom oraz sposobom ich tworzenia zarówno w języku VBScript, jak i w JScript.

➔ <http://www.w3schools.com/vbscript/default.asp>
Na stronie zamieszczonej pod tym adresem znajdziesz interaktywny samouczek, a w nim bardzo pomocne przykładowe skrypty.

Przydatne publikacje:

„Skrypty administracyjne Windows. Czarna Księga”

Autor: Jesse M. Torres; wydawnictwo Helion

„Microsoft VBScript step by step”

Autor: Ed Wilson; Microsoft Press