



Cała prawda o



Od chwili, gdy w grudniu 1995 r. Microsoft rozpoczął swoją ekspansję w kierunku Internetu, przed nazwami wielu nowych technologii tej firmy pojawiło się magiczne słowo „Active”: Active Desktop, Active Server, Active Platform i wreszcie ActiveX. W niniejszym artykule odpowiemy na pytanie, czym są te nowe, „aktywne” elementy.

Hasło „Activate the Internet” zostało stworzone przede wszystkim po to, aby zwiększyć wysiłek projektantów i programistów Microsoftu w wielkim pościgu za ówczesnym internetowym liderem – Netscape Communications. Dlatego też w celach czysto marketingowych oprogramowaniu stworzonemu przez giganta z Redmond, przeznaczonemu do współpracy z Siecią dodano przedrostek „Active”.

ActiveX jest to cały szereg technologii opracowanych po to, by umożliwić współpracę aplikacji Windows poprzez Internet i Intranet. Każdy, kto z pomocą programu *Internet Explorer 3.0* odczytuje stronę WWW zawierającą elementy aktywne używa tej technologii, nie zdając sobie nawet z tego sprawy. To właśnie kontrolki ActiveX powodują, że dokumenty HTML „ożywają”.

Jednym z przykładów zastosowania technologii ActiveX może być wirtualny odpowiednik Stadionu Olimpijskiego w Monachium. Aby się z nim zapoznać, użytkownik, posługując się przeglądarką *Internet Explorer 3.0* (lub nowszą), powinien odczytać stronę <http://www.sport.de/sidi.html>. Po połączeniu się z tym serwerem nastąpi automatyczne – poprzez Sieć – ściągnięcie, zainstalowanie i uruchomienie dodatkowych, niezbędnych do odbycia wirtualnej wycieczki składników. Wszystko to stanie się bez żadnych dodatkowych działań ze strony użytkownika i nie będzie on musiał samodzielnie instalować żadnych rozszerzeń przeglądarki WWW. Użytkownicy *Netscape Navigатора* będą mieli z tą stroną trochę więcej „zachodu” – muszą sami ściągnąć i zainstalować

odpowiednie „wtyczki” (plug-ins), a dopiero potem kontynuować przeglądanie zawartości tego serwisu WWW.

Jak to działa

Fundamentem ActiveX jest znany już od kilku lat Component Object Model (COM). Jest to opracowany przez Microsoft standard, który określa warunki i metody współpracy różnych modułów aplikacji uruchomionych na pojedynczym komputerze (wersja sieciowa nosi nazwę Distributed COM – DCOM). COM został wprowadzony oficjalnie wraz z OLE 2.0 i stał się niebawem fundamentem systemu Windows. Technologia ActiveX obejmuje: ActiveX Controls, ActiveX Documents, Active Scripting, ActiveX Conferencing i Active Server (patrz słowniczek poniżej).

Programista nie musi wprawdzie znać szczegółów standardu Component Object Model, przydatna jest jednak znajomość jego głównych założeń. COM określa jak – poprzez zdefiniowane interfejsy – mogą porozumiewać się ze sobą poszczególne moduły programu, istniejące w formie plików .EXE, .DLL i .OCX. Interfejs taki nie jest niczym innym, jak swoją listą funkcji, zmiennych lub stałych.

Aby umożliwić komunikację wszystkich modułów, każdy obiekt COM dysponuje specjalnym interfejsem *IUnknown*. Udośćpnia on trzy funkcje; jedna z nich nazywa się *IQueryInterface* i zwraca wskaźnik do podanego (jako parametr) interfejsu. Gdyby moduł A chciał wywołać funkcję udostępnianą przez składnik B, musi uzyskać wskaźnik do niej. W tym celu obiekt A wywołuje funkcję *IQueryInterface* i tym samym pyta obiekt B, czy udostępnia on żadaną funkcję. Jeśli tak, to poprzez zwrócony wskaźnik jest ona wywoływana. Dzięki mechanizmowi DCOM, nie ma żadnego znaczenia, czy obiekt B znajduje się na tym samym, czy też na dowolnym innym komputerze w Sieci.

Aby system operacyjny był „świadomy” dostępności poszczególnych modułów zarówno każdy obiekt COM, jak i wszystkie interfejsy muszą zostać zarejestrowane w systemie operacyjnym, za pomocą jednoznacznego 128-bitowego numeru identyfikacyjnego GUID (General Unique Identifier). Dopiero po takiej rejestracji pozostałe moduły mogą się odwoływać do nowych składników systemu operacyjnego. Siła COM tkwi w tym, że nowe interfejsy rejestrować można niezależnie od korzystających z nich modułów (klientów). Każdy

► 169

Słowniczek

Active Desktop – internetowo „aktywny” Pulpit (będący specyficzną przeglądarką WWW); możliwe jest umieszczanie na nim nie tylko ikon reprezentujących aplikacje, ale także obiektów ActiveX i stron WWW. Idea ta zostanie urzeczywistniona w *Internet Explorerze 4.0*

Active Platform – zestaw technologii umożliwiających tworzenie aplikacji sieciowych przy użyciu popularnych narzędzi programistycznych

Active Scripting – technika umożliwiająca umieszczanie na stronach WWW wykonywalnych skryptów napisanych w języku Visual Basic i JScript – microsoftowej implementacji języka JavaScript

Active Server – koncepcja bazująca na *Internet Information Server 3.0*, który udostępnia nie tylko statyczne

i dynamiczne strony HTML, ale także jest serwerem aplikacji. Pozwala to na uruchamianie przez zdalnych użytkowników programów na serwerze

ActiveX – pojęcie zbiorcze dla szeregu technologii, takich jak ActiveX Controls, ActiveX Documents i Active Scripting, stanowiących swoistą bazę dla Active Desktop

ActiveX Conferencing – umożliwia wielu użytkownikom jednoczesny dostęp do dokumentów przez Internet. Przykładem zastosowania tej techniki jest *Microsoft NetMeeting*

ActiveX Documents – mechanizm pozwalający na modyfikację dokumentów poza środowiskiem ich macierzystych aplikacji; dzięki tej technice można np. edytować arkusz Excela bezpośrednio w przeglądarce WWW



interfejs można także w dowolnej chwili wzbogacić nowymi elementami, bez konieczności modyfikowania współpracujących z nim modułów.

Gdy programista chce poszerzyć właściwości już istniejącego obiektu COM, dodaje do niego nowy zestaw funkcji (interfejs). O ich dostępności inne moduły mogą się dowiedzieć za pośrednictwem metody *IUnknown/IQueryInterface*. Dodanie nowego interfejsu (lub rozszerzenie istniejącego) nie wpływa na współpracę innych modułów z tym obiektem.

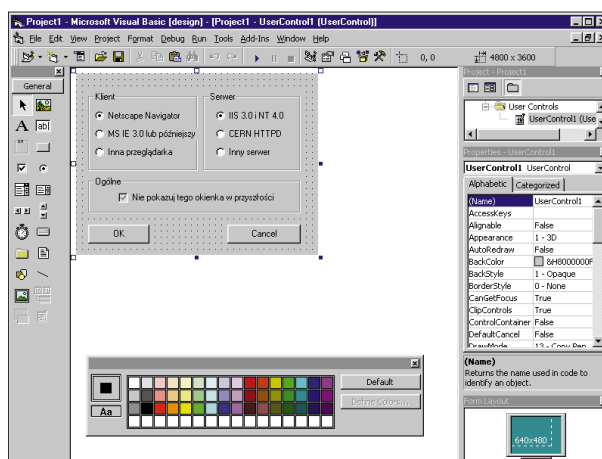
Konkurent: Java

Niektórzy mówią, że Microsoft stworzył ActiveX po to, aby przeciwstawić go wywołanej przez Sun i Netscape euforii Javy. W rzeczywistości jednak porównanie takie nie jest proste. Przede wszystkim, ActiveX i Javy wprost konfrontować ze sobą nie można. ActiveX to szkielet pewnej technologii, Java natomiast jest językiem programowania, zbiorem funkcji API i koncepcją systemu operacyjnego, bazującego na maszynie wirtualnej, niezależnej od systemu operacyjnego.

Można natomiast porównywać ze sobą kontrolki ActiveX oraz aplety Javy. Jednym z ich zadań jest ożywienie statycznych stron HTML i rozszerzenie możliwości WWW. Wystarczy obejrzeć dostępne w Internecie strony, aby przekonać się do ich nowej, pełnej życia formy. Wszystko to można stosunkowo prosto zaprogramować w Javie i uruchamiać niezależnie od używanego systemu operacyjnego.

Aplet Javy nie są programami wykonywalnymi, to znaczy, że procesor nie jest w stanie ich tak po prostu uruchomić. Zawierają one jedynie specjalny kod pośredni, zbiór rozkazów zrozumiałych tylko dla tzw. wirtualnej maszyny Javy (JVM – Java Virtual Machine), która jest elementem przeglądarki WWW. Wirtualna maszyna Javy jest swoistym interfejsem pomiędzy apletem, a systemem operacyjnym. Aplet „widzi” ją zawsze tak samo, niezależnie od systemu operacyjnego, na którym uruchomiona jest przeglądarka WWW. JVM zajmuje się natomiast tłumaczeniem kolejnych rozkazów apletu na kod zrozumiały dla procesora. Aby działać szybciej, aplety bywają często przekształcane przez specjalne kompilatory (Just-in-Time) bezpośrednio na kod maszynowy określonego procesora.

Aplety uchodzą za bezpieczne, ponieważ w trakcie kompilacji (przekształcania kodu źródłowego na binarny), a także uruchomienia cały szereg zabezpieczeń kontroluje, aby nie wykonały one żadnych niedozwolonych operacji (np. modyfikacji na dysku użytkownika). Aplet Javy



**Visual Basic 5 Control
Creation Edition:**
tworzenie elementów
sterujących ActiveX staje
się dziecinnie proste.
Komputerowe asystenty
pomogą użytkownikom,
którzy nie dysponują
wiedzą programistyczną

załadowany z Sieci praktycznie nie może ani uzyskać dostępu do plików na lokalnym dysku, ani uruchomić innych programów. Wynika to z faktu, że JVM zgodna ze specyfikacją nie dysponuje funkcjami dostępu do dysków. Co prawda znane są przykłady niebezpiecznych apletów, jednak do ich stworzenia potrzebna jest gruntowna znajomość mechanizmów bezpieczeństwa wykorzystywanych przez JVM.

W przypadku kontrolki ActiveX rzecz wygląda inaczej. Z punktu widzenia „surfera” realizują one te same zadania (i często wyglądają podobnie), jednak od apletów Javy różnią się zasadniczo. ActiveX Controls wywodzą się z 16-bitowych VBX Controls – samodzielnych, pełniących bardzo różne funkcje „klocków”, z których buduje się aplikacje (standardowe okienka dialogowe, ramki wyświetlające obrazki itp). Moduły VBX pojawiły się w 1991 r. wraz z językiem *Visual Basic 1.0*. Wprowadzenie *Accessa 2.0* i *Visual Basic 4.0* spowodowało zastąpienie ich przez OCX Controls. Aby podkreślić fakt, że współpracują one nie tylko z Visual Basicem, ale praktycznie z każdą inną aplikacją Windows (wykorzystując do tego celu mechanizm OLE), Microsoft przemianował je na „OLE Controls”.

W pewnym momencie gigant z Redmond zmuszony był jednak stwierdzić, że nie posiada niczego porównywalnego z wielce obiecującymi apletami Javy. W celu rozwiązania tego problemu postanowiono sięgnąć do tego, co w przeszłości tak dobrze się sprawdziło – do OLE Controls. Te jednak miały jedną wadę – były o wiele za duże na to, by przy liniach o małej przepustowości dostatecznie szybko migrować w Internecie.

Co więc zrobiono? Z klasycznych modułów OLE usunięto wszelki zbędny balast głównie w postaci niepotrzebnych (rzadko używanych) interfejsów i ActiveX Controls były gotowe. Obiekt

ActiveX musi dysponować jedynie interfejsem *IUnknown* i potrafić się zarejestrować. Cała reszta to swego rodzaju „miniprogram”, który zapisany w postaci plików .OCX spotkać można często w „okienkowych” podkatalogach **SYSTEM** lub **OCACHE**. Ostatnio wszystkie obiekty OLE ewoluują w stronę ActiveX Controls – każdy element sterujący OLE/OCX jest z definicji elementem ActiveX, lecz odwrotna relacja nie zawsze jest prawdziwa.

ActiveX Controls mogą robić to wszystko, co są w stanie zrobić programy wykonywalne: sięgać do plików zapisanych na dysku użytkownika, wywoływać funkcje API czy tworzyć połączenia internetowe. Bez zmiany kodu z tych samych kontrolki ActiveX można budować zarówno całe aplikacje, jak i wzbogacać nimi strony WWW. Co więcej, technologia ActiveX pozwala uruchomić program nie tylko na maszynie użytkownika (klienta), ale także na serwerze. Działające na różnych komputerach moduły mogą się ze sobą porozumiewać, wymieniać dane itd. Można sobie zatem wyobrazić coś na wzór np. serwera MS Office, z którego odczytujemy stronę WWW zawierającą kontrolkę ActiveX, de facto będącą edytorem tekstu MS Word. Pozwala to na edycję dokumentów bezpośrednio w przeglądarce WWW, bez konieczności instalowania całego pakietu biurowego. Komunikacja pomiędzy serwerem a klientem odbywa się przy wykorzystaniu modelu DCOM.

Gdy element ActiveX zostanie odczytany wraz ze stroną WWW, ma – w przeciwieństwie do apletu Javy – pełny dostęp do zasobów lokalnego komputera. Właśnie bezpieczeństwo jest podstawową różnicą pomiędzy apletem, a elementem ActiveX. Obydwa mogą zrobić wiele, z tym tylko, że aplet praktycznie nie może zrobić nic złego.

By jednak zaoferować użytkownikom choćby minimalny stopień bezpieczeństwa ► 17

Niebezpieczeństwa ze strony ActiveX

Problemy z bezpieczeństwem kontrolki ActiveX i ogólnie Internet Explorera zgłaszane były już od dawna. Microsoft uparcie odpowiadał jednak dyplomatycznym „być może jest to niebezpieczne, ale pokażcie nam jakiś udokumentowany przypadek tego typu”. W końcu jednak stało się.

Pod koniec stycznia br. w programie „PlusMinus” niemieckiej telewizji publicznej MDR można było zobaczyć prezentację hamburskiej grupy hackerów znanej pod nazwą Chaos Computer Club (<http://berlin.ccc.de/>). Pokazali oni stworzony przez siebie obiekt ActiveX, za pomocą którego dokonali przelewu z jednego konta bankowego na drugie bez znajomości niezbędnego przy tego typu operacjach tajnego numeru PIN (Personal Identification Number). Ich produkt „działa” wykorzystując popularną w Niemczech aplikację do zarządzania finansami osobistymi *Quicken* firmy Intuit. Obiekt dokonuje modyfikacji w środowisku tego programu, które powodują, że przy następnym (po odczytaniu i uruchomieniu kontrolki ActiveX) podłączeniu użytkownika do sieci bankowej, automatycznie dokonywany jest przelew. Opisana kontrolka nie posiada podpisu elektronicznego, w związku z czym użytkownik musi zgodzić się na załadowanie jej z Sieci. Jednak nieuwaga nieświadomego zagrożenia użytkownika (i wyłączenie w Internet Explorerze pytań o zezwolenie na załadowanie kontrolki) mogą spowodować jednorazowe jej odczytanie i zainstalowanie w systemie, co w przyszłości spowodować będzie straty finansowe. Oczywiście ta konkretna kontrolka

stanowi zagrożenie „tylko” dla ponad 9 milionów użytkowników systemu *Quicken*, ale ukazuje niebezpieczeństwa jakie może spowodować nieświadome korzystanie z technologii ActiveX.

Innym przykładem zagrożenia czyhającego ze strony technologii ActiveX jest dostępny dla wszystkich użytkowników Sieci *Exploder*. Kontrolka ta dostępna jest pod adresem <http://www.halcyon.com/mclain/ActiveX/>, a jej autorem jest Fred McLain. Załadowanie strony zawierającej ten element powoduje, że jeśli w ciągu 10 sekund użytkownik nie kliknie go myszką, nastąpi zamknięcie systemu Windows (bez pytania



o potwierdzenie). Co szczególnie ciekawe – *Exploder* początkowo

posiadał sygnaturę firmy VeriSign, co powodowało, że mógł on być traktowany przez użytkowników jako „godny zaufania”. Ponieważ jednak został stworzony jako demonstracja „niebezpiecznego” działania kontrolki VeriSign nakazał autorowi usunięcie z niej elektronicznego podpisu, dającego pozory bezpieczeństwa.

Kogo nie przekonują zacytowane tu przykłady niech spróbuje odczytać stronę <http://www.thur.de/home/steffen/rename.html>. Zamieszczona na niej kontrolka ActiveX powoduje przeniesienie zawartości (DOS-owa

komenda MOVE) katalogu **C:\WIN95** do katalogu **C:\WINDOWS.ACTIVEX**. Ze względu na szczególne właściwości niektórych plików systemowych (unmovable) odtworzenie stanu pierwotnego (poprzez przeniesienie ww. katalogu na swoje miejsce) nie jest możliwe. System Windows 95 trzeba będzie ponownie zainstalować.

Analizując sposób działania obiektów ActiveX można stwierdzić, że ze względu na praktycznie nieograniczone możliwości mogą stanowić potencjalne zagrożenie dla użytkowników komputerów. Pamiętać jednak należy o koncepcji komputerów sieciowych i o korzystaniu z komercyjnych aplikacji przez Internet. Wszystko to wymaga pełnego dostępu oprogramowania sieciowego do zasobów lokalnego komputera użytkownika. Być może zatem przyszłość ActiveX opiera się na sukcesie rozpoczętej przez Microsoft kampanii informacyjnej, której celem jest uświadomienie użytkownikom potencjalnych niebezpieczeństw wynikających z korzystania z obiektów ActiveX niewiadomego pochodzenia. Jeśli uda się wpoić użytkownikom myśl Corneliusa Willisa (jednego z menedżerów Microsoftu) „Nie bierz cukierków od nieznajomych”, w najbliższej przyszłości technologia ta może okazać się jedną z tych, które zrewolucjonizują Internet.

W Sieci znaleźć można jeszcze wiele przykładów kontrolki ActiveX. Zaskakują swoją widowiskowością, aktywnością i wszechstronnymi możliwościami. Nastrój euforii psuje jednak świadomość, że mogą one zrobić z naszym komputerem wszystko. Wszystko co dobre, ale także wszystko co złe. I to właściwie bez naszej wiedzy.

Microsoft wprowadził technologię Authenticode. Programiści tworzący kontrolki muszą zaopatrzyć się w organizację rejestrującą – amerykańskiej firmie VeriSign – w specjalny podpis elektroniczny (sygnaturę) w postaci fragmentu kodu dołączonego do ich produktu, który kwalifikuje go jako godny zaufania. Zwracając się o tego typu sygnaturę podpisują oni swoiste zapewnienie, że ich dzieło nie jest niebezpieczne dla komputera użytkownika. Przy odczytywaniu strony WWW zawierającej kontrolkę można na podstawie takiej sygnatury zdecydować, czy decydujemy się na ładowanie i uruchomienie danego elementu ActiveX, czy też nie. Metoda ta gwarantuje jednak tylko to, że odczytywana kontrolka jest autorstwa danej firmy, w żaden sposób nie zapewnia jednak, że element jest bezpieczny. Prawdopodobieństwo by programista wirusów (lub innych szkodliwych programów) zarejestrował „niszczyielską” kontrolkę jest co prawda niewielkie, ale zawsze

istnieje. Bezpieczeństwo elementów ActiveX opiera się zatem wyłącznie na zaufaniu użytkownika do ich producenta, a nie na ich specyficznej budowie. Microsoft zapowiada zainicjowanie kampanii informacyjnej, której celem jest uświadomienie użytkownikom zasad funkcjonowania kontrolki ActiveX umieszczanych na stronach WWW i wynikających z tego potencjalnych zagrożeń. Ponieważ jednak coraz więcej organizacji (np. US Navy) decyduje się przyjąć MS Internet Explorera jako domyślną przeglądarkę WWW, sprawy związane z zapewnieniem bezpieczeństwa stają się bardzo pilne.

Projektowanie elementów aktywnych

Obiekty ActiveX tworzyć można przy pomocy *Visual C++*, *Visual J++* oraz, od niedawna, z pomocą specjalnej wersji (*Control Creation Edition*) *Visual Basic* 5.0, którą można np. bezpłatnie ściągnąć

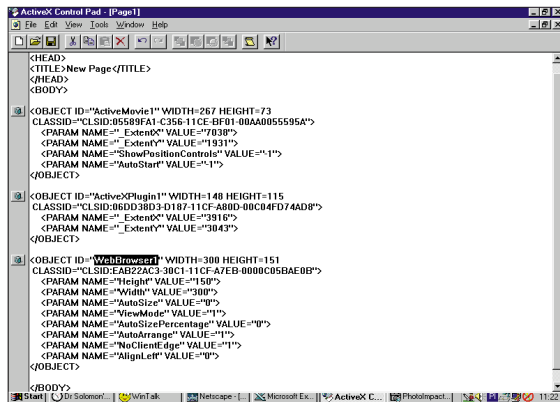
z Internetu. Każdy kto zna Visual Basic na choćby elementarnym poziomie, może teraz w prosty sposób stworzyć własny obiekt ActiveX.

Sposób programowania jest następujący: uruchamiamy Visual Basic 5.0 CCE i tworzymy interfejs graficzny (okienko) naszego elementu ActiveX Control (o ile oczywiście jest ono w ogóle potrzebne). Następnie w procedurach obsługi zdarzeń przypisanych do poszczególnych elementów tego interfejsu opisujemy, jak nasza kontrolka ma się zachowywać, gdy użytkownik na przykład kliknie przycisk. Gdy program w Basicu jest skończony, przekształcamy go w plik .OCX i nasz obiekt ActiveX jest gotów. Nie jest on jednak kompilowany do postaci kodu binarnego, lecz swego rodzaju kodu pośredniego, który przeglądarka WWW potrafi interpretować.

ActiveX Control trzeba jeszcze w odpowiedni sposób umieścić w dokumencie ► 173



HTML. Można tego dokonać np. z pomocą programu *ControlPad* Microsoftu, który jest bezpłatnie dostępny m.in. w Internecie. *ControlPad* nie tylko wstawia do dokumentu HTML niezbędny, wielocyfrowy numer identyfikacyjny kontrolki



ControlPad: za pośrednictwem tego narzędzia każdy użytkownik może bez trudu wbudować w stronę HTML stworzony przez siebie element ActiveX

(CLASSID) i odpowiedni znacznik `<OBJECT>`, lecz oferuje także możliwość powiązania zdarzeń generowanych przez element ActiveX z umieszczonymi na tej stronie skryptami (napisanymi na przykład w języku JScript).

Visual Basic 5.0 CCE dysponuje dwoma atutami: jest bezpłatny, a utworzone przy jego użyciu ActiveX Controls istnieją w formie kodu pośredniego – dokładnie tak jak aplety Javy. Wykonywane są z pomocą specjalnego modułu, określanego jako „wirtualna maszyna Visual Basic” (VBVM – Visual Basic Virtual Machine). Na wszystkich udostępniających go systemach operacyjnych można wykonywać utworzone w Visual Basicu elementy kontrolne ActiveX. Nie jest już tajemnicą, że VBVM będzie stałym składnikiem

następnej wersji Internet Explorera (4.0); wiadomo też, że pojawią się edycje tej przeglądarki WWW dla wielu innych platform. Tak więc programiści będą tworzyć w Visual Basicu aplikacje, które bez ponownego kompilowania będą działać bezbłędnie na innych platformach, takich jak Macintosh, Unix lub nawet dużych systemach mainframe.

No właśnie – będą. W tej chwili obiekty ActiveX w odróżnieniu od apletów Javy i języka HTML są nieprzeznaczone. Jeśli chcemy stworzyć stronę wykorzystującą kontrolki ActiveX i jednocześnie zagwarantować jednakowe jej działanie w większości systemów operacyjnych, musimy włożyć znacznie więcej pracy niż w przypadku Javy. Aplety działają w środowisku wirtual-

nej maszyny Javy, kontrolki ActiveX natomiast wykonują się wprost w środowisku systemu operacyjnego. Są one co prawda dość specyficznymi programami, ale przygotowanymi do pracy (skompilowanymi) dla konkretnego typu procesora i korzystającymi ze standardowych funkcji 32-bitowego środowiska Windows. Dlatego właśnie koniecznym byłoby napisanie kilku wersji tego samego elementu ActiveX dla różnych platform i umieszczenie ich (ze względu na sposób w jaki się tego dokonuje) na oddzielnych, dedykowanych stronach HTML. Jedynym sposobem stworzenia przenośnego ActiveX jest napisanie go w języku Java. Tylko po co pisać potencjalnie niebezpiecznego ActiveX-a skoro przy pomocy tego samego języka i tych samych narzędzi można stworzyć znacznie bezpieczniejszy aplet?

Znając jednak konsekwencję z jaką gigant z Redmond wprowadza nowe standardy powstaje pasjonujące pytanie: kto wygra ten wyścig – ActiveX Controls czy

aplety Javy? Microsoft wcale Javy nie odrzuca – wprost przeciwnie, od niedawna Explorer oferuje bardzo szybką wirtualną maszynę Javy (JVM), która ma wszelkie szanse zostać „okienkowym” standardem. Trzeba jednak pamiętać, że ograniczenia, jakie posiadają aplety Javy zapewniają wprawdzie wysoki poziom bezpieczeństwa, ale znacząco zawężają zakres ich zastosowania. Technologia ActiveX daje znacznie szersze możliwości i jest na tyle interesująca, że twórcy Javy już teraz zastanawiają się nad wprowadzeniem tzw. sygnowanych apletów, o takich samych cechach jak ActiveX Controls. Dlatego pytanie brzmi nie „Java czy ActiveX”, a raczej „model Microsoftu czy też model firm Sun i Netscape”.

oprac. Piotr Wyrzykowski,
Wojciech Wrzaskala (pm)

Uwaga

Na płycie CHIP-CD 6/97 w opcji **Know-how|ActiveX** znaleźć można *Visual Basic 5.0 Control Creation Edition* oraz *ActiveX Control Pad 1.0*. Dokumentację dotyczącą tych narzędzi i dodatkowe informacje na temat technologii ActiveX znajdują się także na płycie CHIP-CD 1/97 w opcji **Software|Test software**.



ActiveX w Internecie

ActiveX SDK:

<http://www.microsoft.com/msdownload/activex.htm>

Visual Basic 5.0 CCE:

<http://www.microsoft.com/vbasic/controls/>

Wszystko o ActiveX:

<http://www.microsoft.com/activex/>

ActiveX Plug-in:

<http://www.ncompasslabs.com/scriptactive/>

Przegląd ActiveX Controls:

<http://www.techweb.com/activexpress/>

<p>MULTIMEDIA VISION</p> <p>Biuro handlowe: 02-673 Warszawa ul. Konstruktorska 4 pok. 101 tel/fax 43-84-81 do 80 w. 230, 231 tel/fax 43-12-01 do 05 w. 230, 231 Zapraszamy w godz. 9.30 - 17.00. e-mail: info@multimedia.com.pl</p>	<p>Active 33.6 wewn. voice PL Motorola Apollo 3400 Lifestyle 28.8 zewn. Motorola Premier 33.6 wewn. USRobotics Courier 33.6 zewn. USRobotics Courier ISDN USRobotics Sportster 33.6 wewn. USRobotics Sportster 33.6 zewn. USRobotics WorldPort 28.8 PC/MILN Zoltrix 14.4 wewn. voice PL Zoltrix 14.4 zewn. voice PL Zoltrix 33.6 wewn. voice PL Zoltrix 33.6 zewn. voice PL ZOOM 28.8 PC/MILN ZOOM 33.6 wewn. PL ZOOM 33.6 zewn. PL ZOOM Comstar 14.4 wewn. PL ZOOM Comstar 33.6 SVD wewn. PL ZyXEL Lite 28.8 zewn. voice ZyXEL Tiny 28.8 ISDN ZyXEL Omni 28.8 wewn. voice</p>	<p>bez VAT z VAT 350,44 zł 430 zł 340,35 zł 420 zł 1000 zł 1220 zł 1122,96 zł 1370 zł 1509,83 zł 1830 zł 373,77 zł 460 zł 672,13 zł 820 zł 853,44 zł 1030 zł 151,54 zł 170 zł 229,51 zł 280 zł 430,33 zł 525 zł 286,87 zł 350 zł 771,31 zł 940 zł 352,46 zł 430 zł 161,21 zł 190 zł 285,88 zł 350 zł 616,89 zł 750 zł 1322,64 zł 1650 zł 1529,67 zł 1910 zł 877,95 zł 1065 zł</p>	<p>The Intelligent Choice in Information Access</p> <p>FAKSMODEMY</p> <p>56 Kbps modem</p> <p>UWAGA!!! Klientom, zawsze dzwoni do nas, aby upewnić się, czy cena w ogłoszeniu jest aktualna (zwykle faktyczna cena jest już niższa).</p>	<p>Internet</p> <p>skanery</p> <table border="1"> <thead> <tr> <th>Model</th> <th>Wzrost</th> <th>Waga</th> </tr> </thead> <tbody> <tr> <td>Musitek Iwain Scan Color 600 (język, kolor)</td> <td>278,60 zł</td> <td>340 zł</td> </tr> <tr> <td>Musitek Iwain Scan Color 800 (język, kolor)</td> <td>180 (3) zł</td> <td>215 zł</td> </tr> <tr> <td>Musitek Paragon Page 630 (język, kolor)</td> <td>508,20 zł</td> <td>620 zł</td> </tr> <tr> <td>Musitek Paragon Page 630 (język, kolor)</td> <td>508,20 zł</td> <td>620 zł</td> </tr> <tr> <td>Musitek Paragon 600 II 111 (język, kolor)</td> <td>860,60 zł</td> <td>1060 zł</td> </tr> <tr> <td>Musitek Paragon 600 II 111 (język, kolor)</td> <td>115,11 zł</td> <td>140 zł</td> </tr> <tr> <td>Musitek Paragon 600 II 111 (język, kolor)</td> <td>1016,39 zł</td> <td>1240 zł</td> </tr> <tr> <td>Musitek Paragon 800 II 5P (język, kolor)</td> <td>1844,96 zł</td> <td>2240 zł</td> </tr> <tr> <td>Musitek Paragon 1700 3P (język, kolor)</td> <td>2254,10 zł</td> <td>2750 zł</td> </tr> <tr> <td>Musitek Paragon 1200 SP Pro (język, kolor)</td> <td>4540,18 zł</td> <td>5540 zł</td> </tr> </tbody> </table>	Model	Wzrost	Waga	Musitek Iwain Scan Color 600 (język, kolor)	278,60 zł	340 zł	Musitek Iwain Scan Color 800 (język, kolor)	180 (3) zł	215 zł	Musitek Paragon Page 630 (język, kolor)	508,20 zł	620 zł	Musitek Paragon Page 630 (język, kolor)	508,20 zł	620 zł	Musitek Paragon 600 II 111 (język, kolor)	860,60 zł	1060 zł	Musitek Paragon 600 II 111 (język, kolor)	115,11 zł	140 zł	Musitek Paragon 600 II 111 (język, kolor)	1016,39 zł	1240 zł	Musitek Paragon 800 II 5P (język, kolor)	1844,96 zł	2240 zł	Musitek Paragon 1700 3P (język, kolor)	2254,10 zł	2750 zł	Musitek Paragon 1200 SP Pro (język, kolor)	4540,18 zł	5540 zł
Model	Wzrost	Waga																																			
Musitek Iwain Scan Color 600 (język, kolor)	278,60 zł	340 zł																																			
Musitek Iwain Scan Color 800 (język, kolor)	180 (3) zł	215 zł																																			
Musitek Paragon Page 630 (język, kolor)	508,20 zł	620 zł																																			
Musitek Paragon Page 630 (język, kolor)	508,20 zł	620 zł																																			
Musitek Paragon 600 II 111 (język, kolor)	860,60 zł	1060 zł																																			
Musitek Paragon 600 II 111 (język, kolor)	115,11 zł	140 zł																																			
Musitek Paragon 600 II 111 (język, kolor)	1016,39 zł	1240 zł																																			
Musitek Paragon 800 II 5P (język, kolor)	1844,96 zł	2240 zł																																			
Musitek Paragon 1700 3P (język, kolor)	2254,10 zł	2750 zł																																			
Musitek Paragon 1200 SP Pro (język, kolor)	4540,18 zł	5540 zł																																			