

Internet Information Services (IIS) 7.0 Resource Kit

Resource Kit

*Mike Volodarsky, Olga Londer, Brett Hill,
Bernard Cheah, Steve Schofield, Carlos
Aguilar Mares i Kurt Meyer
oraz Microsoft IIS Team*

Internet Information Services (IIS) 7.0 Resource Kit
Edycja polska Microsoft Press
Original English language edition © 2008 by Microsoft Press
Tytuł oryginału: Internet Information Services (IIS) 7.0 Resource Kit

Polish edition by APN PROMISE Sp. z o.o. Warszawa 2009

APN PROMISE Sp. z o.o., biuro: ul. Kryniczna 2, 03-934 Warszawa
tel. (022) 355-16-00; fax (022) 355-16-99
e-mail: mspress@promise.pl

Wszystkie prawa zastrzeżone. Żadna część niniejszej książki nie może być powielana ani rozpowszechniana w jakiegokolwiek formie i w jakikolwiek sposób (elektroniczny, mechaniczny), włącznie z fotokopiowaniem, nagrywaniem na taśmy lub przy użyciu innych systemów bez pisemnej zgody wydawcy.

Microsoft, Microsoft Press, Active Directory, ActiveX, Internet Explorer, MSDN, Outlook, SQL Server, Visual Basic, Visual Studio, Windows, Windows Media, Windows Server oraz Windows Vista są zarejestrowanymi znakami towarowymi Microsoft Corporation.

Wszystkie inne nazwy handlowe i towarowe występujące w niniejszej publikacji mogą być znakami towarowymi zastrzeżonymi lub nazwami zastrzeżonymi odpowiednich firm odośnych właścicieli.

Przykłady firm, produktów, osób i wydarzeń opisane w niniejszej książce są fikcyjne i nie odnoszą się do żadnych konkretnych firm, produktów, osób i wydarzeń. Ewentualne podobieństwo do jakiegokolwiek rzeczywistej firmy, organizacji, produktu, nazwy domeny, adresu poczty elektronicznej, logo, osoby, miejsca lub zdarzenia jest przypadkowe i niezamierzone.

APN PROMISE Sp. z o.o. dołożyła wszelkich starań, aby zapewnić najwyższą jakość tej publikacji. Jednakże nikomu nie udziela się rękojmi ani gwarancji.
APN PROMISE Sp. z o.o. nie jest w żadnym wypadku odpowiedzialna za jakiegokolwiek szkody będące następstwem korzystania z informacji zawartych w niniejszej publikacji, nawet jeśli APN PROMISE została powiadomiona o możliwości wystąpienia szkód.

ISBN: 978-83-7541-034-1

Przekład: Leszek Biolik, Maria Chaniewska, Jakub Niedźwiedz
Andrzej Bańkowski, Arkadiusz Czajkowski
Redakcja: Miłosław Kotyński, Marek Włodarz
Korekta: Ewa Swędrowska, Anna Wojdanowicz
Skład i łamanie: MAWart Marek Włodarz

Spis treści

Podziękowania	xv
Wstęp	xvii
Co nowego w IIS 7.0?.....	xvii
Przegląd książki.....	xviii
Konwencje używane w książce.....	xix
Korzystanie z obrazu dysku CD.....	xx
Dodatkowa treść online.....	xxi

Część I Podstawy

1 Wprowadzenie do IIS 7.0	3
Przegląd IIS 7.0.....	4
Nowości w IIS 7.0.....	5
Rdzeń serwera sieci Web.....	6
Konfiguracja.....	8
Narzędzia administracyjne.....	10
Diagnostyka.....	13
Windows Process Activation Service.....	14
Kompatybilność aplikacji.....	14
Podstawowe zadania administracyjne.....	15
Tworzenie witryny sieci Web.....	15
Tworzenie aplikacji.....	17
Tworzenie katalogu wirtualnego.....	18
Tworzenie puli aplikacji.....	20
Przypisywanie aplikacji do puli aplikacji.....	21
Właściwości IIS 7.0 w Windows Server 2008 i Windows Vista.....	22
Podsumowanie.....	25
Dodatkowe zasoby.....	26
2 Architektura IIS 7.0	27
Przegląd architektury IIS 7.0.....	28
Rdzenne komponenty IIS 7.0.....	31
HTTP.sys.....	31
World Wide Web Publishing Service.....	33
Windows Process Activation Service.....	34
Magazyn konfiguracji.....	35
Proces roboczy.....	37
Przetwarzanie żądań w puli aplikacji.....	39
Tryb klasyczny.....	40

Tryb zintegrowany z .NET	43
Zakres modułu	47
Określanie kolejności modułu	48
Przetwarzanie żądań innych niż HTTP	49
Podsumowanie	51
Dodatkowe zasoby	52
3 Podstawy struktury modularnej	53
Koncepcje	53
Idee	54
Typy modułów	54
Moduły i konfiguracja	55
Główne zalety	57
Bezpieczeństwo	57
Wydajność	59
Rozszerzalność	59
Wbudowane moduły	60
Podsumowanie	61
Dodatkowe zasoby	61
4 System konfiguracyjny	63
Przegląd systemu konfiguracyjnego	64
Hierarchia plików konfiguracyjnych	65
Składnia pliku konfiguracyjnego	70
System konfiguracyjny IIS 7.0 a metabaza IIS 6.0	77
Systemy konfiguracyjne IIS 7.0 i .NET	79
Edycja konfiguracji	81
Wybór miejsca na konfigurację	82
Określanie konfiguracji	83
Błędy konfiguracyjne	86
Zarządzanie konfiguracją	90
Tworzenie kopii zapasowej konfiguracji	90
Korzystanie z historii konfiguracji	91
Eksport i import konfiguracji	92
Delegowanie konfiguracji	93
Udostępnianie konfiguracji pomiędzy serwerami	103
Podsumowanie	109
Dodatkowe zasoby	110
 Część II Uruchomienie	
5 Instalowanie IIS 7.0	113
Planowanie instalacji	113
Scenariusze instalacyjne dla IIS 7.0	115
Sposoby instalacji IIS 7.0	127
Korzystanie z Server Manager	127

Korzystanie z Package Manager	129
Korzystanie z ServerManagerCMD	129
Pliki instalacji nienadzorowanej	132
System instalacji Sysprep/New	134
Instalacje automatyczne	135
Instalacja Windows Server 2008 dla funkcji opcjonalnych	135
Po instalacji	136
Foldery i zawartość	136
Rejestr	137
Usługi	138
Sprawdzanie	138
Rozwiązywanie problemów z instalacją	139
Dzienniki zdarzeń	139
Dziennik IIS 7.0	139
Inne związane opcje rejestrowania	140
Usuwanie IIS 7.0	141
Interfejs użytkownika w Windows Server 2008 i Windows Vista	141
Metoda wiersza polecenia	143
Podsumowanie	144
Dodatkowe zasoby	144

Część III Administracja

6 Korzystanie z konsoli IIS Manager	147
Przegląd funkcji programu IIS Manager	148
Uruchamianie programu IIS Manager	150
Interfejs użytkownika programu IIS Manager	151
Pasek narzędzi nawigacji	153
Okienko Połączenia	153
Obszar roboczy	156
Okienko Akcje	169
Działanie funkcji	170
Funkcja mapowania modułu	170
Gdzie konfiguracja zostaje zapisana	173
Zakres funkcji	176
Dostosowywanie i możliwości rozszerzania funkcji programu IIS 7.0 Manager	177
Administracja zdalna	180
Podsumowanie	181
Dodatkowe zasoby	182
7 Korzystanie z narzędzi wiersza polecenia	183
Korzystanie z narzędzi zarządzających w wierszu polecenia	183
Appcmd.exe	185
Początek pracy z Appcmd	186
Składnia Appcmd	187
Obsługiwane obiekty	189
Uzyskiwanie pomocy	190

Zrozumienie danych wyjściowych Appcmd	194
Parametry ogólne	196
Korzystanie z operatorów zakresu	197
Unikanie typowych pułapek Appcmd	198
Korzystanie z podstawowych czynności: List, Add, Set, Delete	199
Korzystanie z polecenia List do wypisywania i wyszukiwania obiektów	199
Korzystanie z czynności Add do tworzenia obiektów	200
Korzystanie z czynności Set do zmiany istniejących obiektów	201
Korzystanie z czynności Delete do usuwania obiektów	202
Praca z konfiguracją	203
Przeglądanie konfiguracji przy pomocy polecenia List Config	204
Ustawianie konfiguracji przy pomocy polecenia Set Config	205
Zarządzanie delegowaniem konfiguracji	209
Zarządzanie kopiami zapasowymi konfiguracji	209
Praca z aplikacjami, katalogami wirtualnymi i pulami aplikacji	210
Praca z modułami serwera Web	211
Badanie działających procesów roboczych i żądań	211
Wypisywanie listy działających procesów roboczych IIS	212
Wypisywanie listy aktualnie wykonywanych żądań	212
Praca ze śledzeniem niepomysłnych żądań	214
Włączanie śledzenia niepomysłnych żądań	214
Tworzenie reguł śledzenia niepomysłnych żądań	214
Przeszukiwanie dzienników śledzenia niepomysłnych żądań	216
Microsoft.Web.Administration	219
Tworzenie witryn przy pomocy MWA	219
Tworzenie pul aplikacji przy pomocy MWA	220
Ustawianie konfiguracji	221
Windows PowerShell, a usługi IIS 7.0	222
Dostawca WMI	223
Obiekty COM IIS 7.0 Configuration	224
Podsumowanie	224
Dodatkowe zasoby	225
8 Administracja zdalna	227
IIS Manager	228
Web Management Service	228
Instalacja	229
Konfiguracja usług WMS	230
Zarządzanie administracją zdalną	238
Używanie administracji zdalnej	248
Rozwiązywanie problemów	251
Rejestrowanie	254
Podsumowanie	256
Dodatkowe zasoby	256

9 Zarządzanie witrynami sieci Web	257
Witryny, aplikacje, katalogi wirtualne i pule aplikacji sieci Web	257
Witryny sieci Web	258
Aplikacje	261
Katalogi wirtualne	262
Pule aplikacji	263
Zadania administracyjne	264
Dodawanie nowej witryny sieci Web	264
Konfigurowanie powiązań witryny sieci Web	268
Ograniczanie wykorzystania witryny sieci Web	271
Konfigurowanie rejestrowania i śledzenia niepomyślnych żądań witryny sieci Web	273
Uruchamianie i zatrzymywanie witryn sieci Web	274
Zarządzanie katalogami wirtualnymi	275
Dodawanie nowego katalogu wirtualnego	275
Konfigurowanie katalogów wirtualnych	277
Wyszukiwanie katalogów wirtualnych	280
Zarządzanie zawartością zdalną	282
Konfigurowanie aplikacji tak, by korzystała z zawartości zdalnej	283
Wybór modelu zabezpieczeń do uzyskiwania dostępu do zawartości zdalnej	283
Konfigurowanie poświadczeń ustalonych dla uzyskania dostępu do zawartości zdalnej	285
Przydzielanie dostępu do zawartości zdalnej	286
Podsumowanie	287
Dodatkowe zasoby	288
10 Zarządzanie aplikacjami i pulami aplikacji	289
Zarządzanie aplikacjami sieci Web	289
Tworzenie aplikacji sieci Web	290
Tworzenie listy aplikacji sieci Web	295
Zarządzanie pulami aplikacji	297
Wskazówki dotyczące puli aplikacji	298
Dodawanie nowej puli aplikacji	300
Zarządzanie tożsamościami puli aplikacji	303
Zaawansowana konfiguracja puli aplikacji	308
Zarządzanie procesami roboczymi i żadaniami	314
Monitorowanie procesów roboczych i żądań	314
Podsumowanie	318
Dodatkowe zasoby	319
11 Utrzymywanie środowisk rozwijania aplikacji	321
IIS jako platforma rozwijania aplikacji	321
Dodawanie wsparcia dla platform aplikacyjnych	323
Obsługiwane platformy aplikacyjne	324
Utrzymywanie aplikacji ASP.NET	325
Zrozumienie zintegrowanego i klasycznego trybu ASP.NET	326

Uruchamianie wielu wersji ASP.NET obok siebie.....	328
Instalowanie ASP.NET.....	330
Wdrażanie aplikacji ASP.NET.....	332
Dodatkowe uwarunkowania dotyczące wdrażania.....	338
Utrzymywanie aplikacji ASP.....	340
Instalowanie ASP.....	340
Wdrażanie aplikacji ASP.....	341
Dodatkowe uwarunkowania dotyczące wdrażania.....	342
Utrzymywanie aplikacji PHP.....	343
Wdrażanie aplikacji PHP.....	344
Dodatkowe uwarunkowania dotyczące wdrażania.....	348
Techniki dołączania platform aplikacyjnych.....	351
Włączanie obsługi nowych rozszerzeń plików statycznych.....	352
Wdrażanie platform opartych na modułach macierzystych IIS 7.0.....	354
Wdrażanie platform w oparciu o programy obsługi ASP.NET.....	354
Wdrażanie platform w oparciu o rozszerzenia ISAPI.....	355
Wdrażanie platform wykorzystujących FastCGI.....	356
Wdrażanie platform wykorzystujących CGI.....	360
Podsumowanie.....	362
Dodatkowe zasoby.....	363
12 Zarządzanie modułami serwera Web.....	365
Rozszerzenia w IIS 7.0.....	365
Rzut oka na architekturę rozszerzeń IIS 7.0.....	366
Zarządzanie rozszerzeniami.....	368
Rozszerzenia środowiska uruchomieniowego serwera Web.....	370
Czym jest moduł?.....	370
Instalowanie modułów.....	375
Typowe zadania zarządzania modułami.....	387
Korzystanie z konsoli IIS Manager do instalowania i zarządzania modułami.....	395
Korzystanie z konsoli IIS Manager do tworzenia i zarządzania mapowaniami obsługi.....	398
Korzystanie z Appcmd do instalowania i zarządzania modułami.....	402
Tworzenie i zarządzanie mapowaniami obsługi.....	407
Zabezpieczanie modułów serwera Web.....	409
Podsumowanie.....	421
Dodatkowe zasoby.....	421
13 Zarządzanie rozszerzeniami konfiguracji i interfejsu użytkownika.....	423
Przegląd stosu administracyjnego.....	423
Zarządzanie rozszerzeniami konfiguracji.....	425
Schemat sekcji konfiguracyjnych.....	427
Deklarowanie sekcji konfiguracyjnych.....	429
Instalowanie nowych sekcji konfiguracyjnych.....	432
Zarządzanie rozszerzeniami administracyjnymi.....	438
Jak działają rozszerzenia administracyjne.....	440

Instalowanie rozszerzeń administracyjnych.....	441
Zabezpieczanie rozszerzeń administracyjnych.....	441
Zarządzanie rozszerzeniami konsoli IIS Manager	442
Jak działają rozszerzenia konsoli IIS Manager	443
Instalowanie rozszerzeń konsoli IIS Manager	445
Zabezpieczanie rozszerzeń konsoli IIS Manager.....	445
Podsumowanie.....	448
Dodatkowe zasoby.....	448
14 Implementowanie strategii zabezpieczeń.....	449
Zmiany funkcji zabezpieczeń wprowadzone w usługach IIS 7.0.....	450
Zmniejszenie możliwych obszarów ataków.....	452
Zmniejszanie obszarów zagrożeń aplikacji.....	462
Konfigurowanie aplikacji zgodnie z zasadą minimalnych uprawnień	468
Używanie tożsamości puli aplikacji o niskich uprawnieniach.....	468
Konfigurowanie uprawnień systemu plików NTFS, by przydzielany był minimalny dostęp	471
Zmniejszenie zaufania aplikacji ASP.NET	473
Izolowanie aplikacji	475
Implementowanie kontroli dostępu.....	477
Ograniczenia adresów IP i domen	478
Filtrowanie żądań.....	480
Autoryzacja.....	486
Autoryzacja bazująca na liście ACL systemu NTFS	487
Autoryzacja URL	488
Uwierzytelnienie.....	493
Uwierzytelnienie anonimowe	495
Uwierzytelnienie podstawowe	497
Uwierzytelnienie szyfrowane	499
Uwierzytelnienie systemu Windows	501
Uwierzytelnienie certyfikatu klienta	505
Uwierzytelnienie certyfikatu klienta w usługach IIS	507
Uwierzytelnienie UNC	513
Działanie delegowania uwierzytelnienia	514
Zabezpieczanie komunikacji za pomocą protokołu SSL (Secure Socket Layer).....	516
Konfigurowanie protokołu SSL	516
Wymaganie używania protokołu SSL.....	518
Certyfikaty klienckie.....	519
Zabezpieczanie konfiguracji.....	520
Ograniczanie dostępu do konfiguracji	521
Zabezpieczanie ważnych konfiguracji.....	526
Kontrolowanie delegowania konfiguracji	531
Podsumowanie.....	536
Dodatkowe zasoby.....	537

Część IV Rozwiązywanie problemów i wydajność

15 Rejestrowanie	541
Co nowego?	541
IIS Manager	542
Schemat rejestrowania bazujący na XML	543
Scentralizowane opcje konfiguracji rejestrowania	544
Opcje konfiguracji SiteDefaults	544
Wyłączanie opcji konfiguracji rejestrowania HTTP	545
Domyślna lokalizacja pliku dziennika	545
Domyślne kodowanie UTF-8	546
Nowe kody stanu	546
Usługa zarządzania	546
Formaty plików dzienników, które nie zostały zmienione	547
Scentralizowane rejestrowanie	547
Scentralizowany format rejestrowania W3C	547
Scentralizowany binarny format rejestrowania	548
Rejestrowanie zdalne	548
Konfigurowanie rejestrowania zdalnego przy użyciu programu IIS Manager	548
Konfigurowanie rejestrowania zdalnego przy użyciu polecenia Appcmd	551
Rejestrowanie zdalne przy użyciu usługi FTP 7.0 Publishing Service	551
Rejestrowanie niestandardowe	552
Konfigurowanie rejestrowania IIS	554
IIS Manager	554
Appcmd	557
Zaawansowane opcje narzędzia Appcmd	559
Rejestrowanie usługi HTTP.sys	563
Rejestrowanie aplikacji	564
Rejestrowanie odtwarzania procesu	564
ASP	565
ASP.NET	565
Zdarzenia IIS	565
Opcje kompresji folderów	566
Analizy zarejestrowanych informacji przy użyciu narzędzia Log Parser	567
Podsumowanie	568
Dodatkowe zasoby	568
16 Śledzenie i rozwiązywanie problemów	569
Śledzenie i diagnostyka problemów	570
Instalowanie modułu Failed Request Tracing	570
Włączanie i konfigurowanie funkcji FRT	571
Odczytywanie dzienników funkcji FRT	578
Integracja funkcji śledzenia i platformy ASP.NET	582
Analiza dotycząca wydajności	583
Rozwiązywanie problemów	585
Stosowanie metodologii	585

Korzystanie z narzędzi i programów narzędziowych	586
Rozwiązywanie problemów protokołu HTTP	600
Rozwiązywanie specyficznych kwestii	608
Nie został zainstalowany zestaw IIS 6.0 Administration Tools	608
Nie zostały włączone dyrektywy SSL	608
Nieoczekiwane odtwarzanie	608
Awarie	608
Nie można połączyć się z witryną sieci Web	609
Błędy dotyczące uwierzytelniania	609
Długi czas odpowiedzi lub zawieszenie serwera	609
Podsumowanie	609
Dodatkowe zasoby	610
17 Dostrajanie wydajności	611
Zachowanie proporcji pomiędzy bezpieczeństwem a wydajnością	612
W jaki sposób ocenić koszt?	612
Uwierzytelnienie	618
SSL	618
Wpływ ograniczonych zasobów	619
Procesor	620
Co powoduje obciążenie CPU?	620
Tłumienie	620
Liczniki związane z monitorowaniem CPU	621
Wpływ ograniczeń	623
Środki zaradcze	623
Pamięć	624
Co jest powodem zajmowania pamięci?	625
Licznik związane z monitorowaniem pamięci	625
Wpływ ograniczeń	627
Środki zaradcze	628
Dyski twarde	628
Jakie są przyczyny przeciążeń dysków twardych?	628
Liczniki związane z monitorowaniem dysku twardego	629
Wpływ ograniczeń	630
Środki zaradcze	630
Sieć	631
Co powoduje przeciążenie sieci?	631
Liczniki związane z monitorowaniem sieci	632
Wpływ ograniczeń	632
Środki zaradcze	633
Liczniki na poziomie aplikacji	634
Porównanie trybu 64-bitowego z trybem 32-bitowym	642
Dostrajanie wydajności	643
Poziom serwera	644

IIS 644	
Optymalizacja związana z typem obciążenia.....	644
Narzędzia po stronie serwera.....	646
Aplikacja	656
Monitorowanie wydajności.....	657
WCAT.....	658
Reliability And Performance Monitor.....	658
FRT	658
Event Viewer.....	659
System Center Operations Manager 2007.....	659
Skalowalność	659
Projekt.....	660
Zwiększanie czy zmniejszanie	660
Podsumowanie.....	663
Dodatkowe zasoby.....	664

Część V Dodatki

A Kody stanu HTTP w usługach IIS 7.0	667
B Komunikaty o błędach w usługach IIS 7.0	675
Błędy HTTP w usługach IIS 7.0	676
Konfiguracja <httpErrors>	677
Pomocnicze kody stanu	678
Przykład kodu pomocniczego.....	679
Błędy niestandardowe specyficzne dla języka.....	679
Opcje błędów niestandardowych	680
Wykonanie adresu URL	680
Przekierowanie żądania	681
C Moduły usług IIS 7.0	683
Moduły macierzyste	683
Moduły zarządzane	692
D Kolejność modułów	697
E Ustawienia domyślne, progi i limity czasu w usługach IIS 7.0	701
ASP.NET	701
IIS 709	
Management (Zarządzanie)	731
Ustawienia domyślne pul aplikacji.....	733
F Usługi IIS 7.0 a 64-bitowe wersje systemu Windows	737
Windows Server 2008 x64	737

Konfigurowanie 32-bitowej aplikacji w 64-bitowym systemie Microsoft Windows.....	738
G Funkcje programu IIS Manager i odniesienia do systemów konfiguracji.....	741
ASP.NET	741
IIS 742	
Management (Zarządzanie)	744
H Mapowania metabazy IIS 6.0 do usług IIS 7.0.....	745
I Udostępniana obsługa IIS 7.0	757
Implementacja kontroli liczby procesów	757
Przy użyciu wiersza polecenia	758
Zmiany konfiguracji	759
Włączanie dynamicznego progu bezczynności	759
Przy użyciu wiersza polecenia	762
Zmiany konfiguracji	762
J Typowe zadania administracyjne przy użyciu programu IIS Manager.....	763
Indeks.....	769
Informacje o autorach	797
Wymagania systemowe.....	800

Podziękowania

Książka, którą trzymacie w rękach, jest wynikiem wspólnego wysiłku wielu osób.

Chcemy zacząć od podziękowania Billowi Staplesowi, Mai-Lan Tomsen Bukovec i całemu zespołowi produkcyjnemu IIS za ich wsparcie. Wielu z nas pracowało w zespole produktu IIS i wiemy z pierwszej ręki, że po prostu nie moglibyśmy napisać tej książki bez nieocenionej pomocy wszystkich członków zespołu.

Po drugie, jesteśmy bardzo wdzięczni Martinowi DelRe z Microsoft Press za wizję i ciężką pracę pozwalającą na powstanie tego projektu i na zapewnienie jego pomyślnego zakończenia oraz za nigdy niekończące się wsparcie i zaangażowanie.

Powołanie do życia takiej książki, jak ta, wymaga wielu osób i wiele pracy. Szczególnie chcielibyśmy podziękować kilku ludziom, bez których nie byłoby tej książki. Brett Hill zaczął ten projekt i pracował nad nim do czasu ukończenia. Specjalne podziękowania należą się Mike'owi Volodarsky'emu, którego pasja, by zapewnić książce jak największą jakość i kompletność, spowodowała, że stał się wiodącym autorem. Kurt Meyer pomógł wielu menedżerom projektu koordynować pisanie i sprawił, że nie ominięto żadnych ważnych części projektu.

Wielu naszych kolegów z zespołu produktu IIS ma znaczący wkład w treść książki. Faktycznie każdy rozdział został przejrany przez przynajmniej jednego członka zespołu produkcyjnego. Inni ludzie z zespołu produktu napisali notki „wprost ze źródła”, które zostały zamieszczone w różnych miejscach książki, dając Czytelnikowi unikalny wgląd w projekt i tworzenie IIS 7.0. Chcielibyśmy wyrazić szczerze podziękowanie następującym członkom zespołu produktu IIS, którzy pracowali z nami nad książką, wymienionych w kolejności alfabetycznej według imion: Anil Ruia, Bill Staples, Edmund Chou, Eric Deily, Fabio Yeon, Jaroslav Dunajsky, Kanwaljeet Singla, Nazim Lala, Michael Brown, Thomas Marquardt, Tobin Titus, Ulad Malashanka i Wade Hilmo.

Chcemy także podziękować Tito Leverette za jego wskazówki i wkład do rozdziału 17 „Dostrajanie wydajności”.

Wiele innych zespołów w Microsoft zapewniło techniczne przeglądy i dzieliło się z nami swoim doświadczeniem i wnikliwością. W szczególności, jesteśmy wdzięczni Tomowi Hawthorn z zespołu Windows Performance, a także Georgowi Holman i całemu zespołowi Microsoft.com Operations. Nick McCollum z Quixtar Inc. pomógł nam także, dokonując przeglądów technicznych i sugestii w rozdziałach 5, 15 i 17.

Następnie chcemy podziękować naszemu wybitnemu zespołowi redakcyjnemu. W szczególności chcielibyśmy wyrazić wdzięczność redaktorom projektu, Karen Szall i Victorii Thulman z Microsoft Press za ich profesjonalizm, pomoc, wspaniałą pracę redakcyjną, a najbardziej za ich cierpliwość. Bob Hogan i Bob Dean przeprowadzili techniczne przeglądy książki, zapewniając, że całość jest spójna i łatwa do zrozumienia. Jean Findley z Custom Editorial Productions, Inc. wykonała wspaniałą pracę zarządzając produkcją książki przy napiętym harmonogramie.

Ponadto chcielibyśmy podziękować Susan Chory i Isaacowi Roybal za pomoc w powstaniu tego projektu. Jesteśmy także wdzięczni Simonowi Brown i Arvindra Sehmi za ich poparcie dla tej pracy.

Dziękujemy wszystkim!

Pozdrowienia,

Zespół autorów: Mike, Olga, Brett, Bernard, Steve, Carlos i Kurt

Wstęp

Witamy Czytelników *Internet Information Services (IIS) 7.0 Resource Kit*. Ta książka jest szczegółowym technicznym źródłem informacji na temat planowania, wdrażania i obsługi Microsoft Internet Information Services (IIS) 7.0, będącego następną generacją platformy serwera sieci Web firmy Microsoft. Chociaż książka ta jest przeznaczona przede wszystkim dla profesjonalistów IT, którzy mieli doświadczenie z poprzednimi wersjami IIS, każdy, kto chce się nauczyć, jak wdrażać i operować IIS 7.0, uzna ją za bardzo wartościową.

Czytelnik znajdzie wewnątrz pogłębione informacje na temat udoskonaleń wprowadzonych w IIS 7.0 i podległych koncepcjach architektonicznych. Pozwolą one lepiej zrozumieć zasady związane z wdrażaniem i zarządzaniem serwerami sieci Web IIS 7.0 i nauczyć się technik wykorzystywania nowych właściwości i możliwości IIS 7.0. Czytelnik zapozna się także z zaawansowanymi zagadnieniami, w tym z wdrażaniem modularnych serwerów sieci Web, konfigurowaniem witryn i aplikacji tej sieci oraz poprawianiem zabezpieczeń, niezawodności i wydajności serwera sieci Web. W książce zawarto także wiele uwag sporządzonych przez członków zespołu produktu IIS, które dostarczają głębokiego wglądu w sposób działania IIS 7.0, i opisano zarówno najlepsze praktyki zarządzania platformą serwera sieci Web, jak i nieocenione wskazówki służące rozwiązywaniu problemów. Dołączona do książki płyta CD obejmuje dodatkowe narzędzia i dokumentację, których można używać do zarządzania i rozwiązywania problemów dotyczących serwerów sieci Web IIS 7.0.

Co nowego w IIS 7.0?

IIS 7.0 został całkowicie przebudowany w celu dostarczenia modularnej i rozszerzalnej platformy serwera sieci Web, tworząc podstawy dla lekkich, zużywających mało zasobów serwerów tej sieci. Nowa, rozszerzalna architektura pozwala na całkowite dostosowanie serwera sieci Web, ale można wybrać tylko wymagane funkcje IIS i w miarę potrzeb dodawać lub zastępować je nowymi właściwościami, które zapewniają nową, bogatą rozszerzalność interfejsów programowania aplikacji (API). Ponadto serwer sieci Web pozwala na użycie nowego, rozproszonego systemu konfiguracyjnego i narzędzi zarządzania, które upraszczają wdrażanie i zarządzanie. Rdzenny zbiór właściwości IIS 7.0 dodatkowo podnosi niezawodność zorientowanej na zabezpieczenia architektury istniejącej już w poprzedniej wersji, IIS 6.0, i dodaje kolejne ulepszenia do wzmacniania niezawodności i bezpieczeństwa platformy serwera sieci Web. IIS 7.0 zapewnia także szersze wsparcie dla środowisk aplikacji obejmujących lepszą integrację z ASP.NET i wbudowane wsparcie dla środowisk zgodnych z FastCGI.

Wśród wielu udoskonaleń IIS 7.0 dostarcza między innymi:

- **Modularną architekturę serwera sieci Web** W przeciwieństwie do swych poprzedników, IIS 7.0 jest kompletnie modularnym serwerem sieci Web, zawierającym ponad 40 komponentów, które administrator może instalować pojedynczo, by utworzyć mało obciążające zasoby wdrożenia serwera, które odgrywają specyficzną rolę w topologii aplikacji. Co więcej – nowa architektura rozszerzeń pozwala dowolnym, wbudowanym właściwościom modularnym na zastępowanie przez dostosowane implementacje dostarczane przez Microsoft i firmy trzecie.

- **Rozszerzalność .NET przez integrację z ASP.NET** Nowe możliwości integracyjne ASP.NET pozwalają na wdrażanie właściwości IIS 7.0 wraz z ASP.NET i .NET Framework, redukując koszty użycia i utrzymania dla własnych rozwiązań serwera sieci Web. W tym trybie można wykorzystać istniejące usługi ASP.NET do wzmocnienia dowolnych technologii aplikacji, nawet tych, które nie są wdrażane z myślą o ASP.NET. Możliwości te pozwalają aplikacjom sieci Web korzystającym z IIS 7.0 na dalsze dostosowywania serwera tej sieci do swych potrzeb – bez ponoszenia wyższych kosztów wdrażania związanych z poprzednio używanym Internet Server Application Programming Interface (ISAPI).
- **Udoskonalone wsparcie środowiska aplikacji** Oprócz poprawionej integracji ASP.NET w celu rozszerzania serwera sieci Web IIS 7.0 dostarcza więcej opcji do hostingu innych środowisk aplikacji. Obejmuje to wbudowane wsparcie dla protokołu FastCGI używanego przez wiele środowisk aplikacji open source, takich jak PHP Hypertext Preprocessor (PHP), które w ten sposób mogą być bezpiecznie umieszczane w środowisku Windows.
- **Rozproszony system konfiguracyjny ze wsparciem delegacji** IIS 7.0 zastępuje scentralizowany magazyn konfiguracyjny, zwany metabazą, nowym systemem konfiguracyjnym opartym na rozproszonej hierarchii plików XML, która pozwala aplikacjom na kontrolowanie własnej konfiguracji. Nowy system konfiguracyjny pozwala na uproszczone wdrożenie aplikacji bez nadmiarowości wymaganego zaangażowania administracyjnego i dostarcza podstaw do bardziej elastycznego zarządzania konfiguracją serwera sieci Web.
- **Poprawione narzędzia zarządzania** IIS 7.0 oferuje wiele narzędzi zarządzania, które wykorzystują nowy system konfiguracyjny do dostarczania bardziej elastycznego i prostszego zarządzania konfiguracją serwera sieci Web. Obejmuje to całkiem nowe, zorientowane na zadania narzędzie IIS Manager, które oferuje delegowane zarządzanie, nowe narzędzie do zarządzania z wiersza polecenia (Appcmd) i wiele interfejsów API dla zarządzania konfiguracją serwera sieci Web za pomocą skryptów, Windows Management Instrumentation (WMI) i programów .NET Framework.
- **Udoskonalona diagnostyka i rozwiązywanie problemów** IIS 7.0 dostarcza właściwości diagnostycznych do pomocy w diagnozowaniu błędów serwera sieci Web i rozwiązywania trudnych do skopiowania warunków za pomocą infrastruktury Failed Request Tracing. Diagnostyczne właściwości śledzenia są integrowane z aplikacjami ASP.NET w celu zapewniania całościowej diagnostyki aplikacji sieci Web.

Przegląd książki

Książkę podzielono na cztery części obejmujące następujące tematy:

- **Część I: Podstawy** Zawiera przegląd właściwości IIS 7.0, opisuje udoskonalenia wprowadzone w IIS 7.0 i wprowadza rdzeń architektury serwera sieci Web.
- **Część II: Wdrożenie** Wyjaśnia modularną architekturę instalacji do wdrażania IIS 7.0 i przedstawia procedury instalowania IIS 7.0 dla typowych obciążeń serwera sieci Web.

- **Część III: Administracja** Opisuje kluczowe koncepcje zarządzania IIS 7.0 oraz to, jak wykonywać zadania zarządzania, używając narzędzi zarządzania dostarczanych przez IIS 7.0.
- **Część IV: Rozwiązywanie problemów i wydajność** Opisuje, jak używać infrastruktury rejestrowania i śledzenia do zapewniania sprawnej pracy serwera sieci Web i do rozwiązywania problemów w razie występowania błędów, a także monitorowanie i udoskonalanie wydajności serwera sieci Web.

Dodano także wiele uzupełnień na różne tematy oraz słownik pojęć.

Konwencje używane w książce

W tej książce do podkreślenia specjalnych właściwości lub zastosowań używane są następujące konwencje:

Pomoc dla Czytelnika

W celu wskazania użytecznych szczegółów w tej książce używane są następujące hasła:

Hasło	Znaczenie
Uwaga	Podkreśla wagę specyficznej koncepcji lub specjalny przypadek, którego nie da się zastosować do wszystkich sytuacji.
Ważne	Podkreśla wagę informacji, których nie należy przeoczyć.
Ostrzeżenie	Ostrzega, że podjęcie lub uniknięcie pewnego działania może spowodować poważne problemy dla użytkowników, systemów, integralności danych itp.
Na dysku CD	Na załączonym dysku CD znajdują się powiązane z treścią książki skrypty, narzędzia, szablony lub pomoce umożliwiające wykonanie zadań opisanych w tekście.

Uwagi

W tej książce zamieszczono następujące uwagi mające na celu dostarczenie dodatkowych wskazówek i porad związanych z różnymi właściwościami IIS 7.0.

Uwaga	Znaczenie
Ze źródła	Napisane przez ekspertów z Microsoft, aby dostarczyć wnikliwych informacji na temat działania IIS 7.0, najlepszych sposobów zarządzania IIS 7.0 i wskazówek dotyczących rozwiązywania problemów.
Jak to działa?	Opis właściwości IIS 7.0 i sposób ich działania.

Przykłady wiersza polecenia

W celu ukazania przykładów wiersza polecenia użyto następujących konwencji stylistycznych:

Styl	Znaczenie
Czcionka pogrubiona	Używana w celu wskazania informacji wprowadzanych przez użytkownika (znaki, które się wpisuje dokładnie tak, jak pokazano).
<i>Kursywa</i>	Do wskazania zmiennych, dla których trzeba podać konkretną wartość (na przykład: file_name może wskazywać dowolną prawidłową nazwę pliku).
Czcionka o stałej szerokości	Używana dla przykładów kodu i wyników wiersza polecenia.
%SystemRoot%	Składnia zapisu zmiennych środowiskowych.

Korzystanie z obrazu dysku CD

Obraz dysku CD towarzyszącego książce jest dostępny na stronie wydawcy przy opisie książki w zakładce Dodatkowe informacje, pod adresem:

<http://www.książki.promise.pl/asp/produkt.aspx?pid=49842>

Na podstawie tego obrazu można wykonać fizyczny dysk CD lub zainstalować go jako napęd wirtualny. Dysk ten zawiera:

- **Książka elektroniczna** Pełen tekst książki w formacie PDF (w języku angielskim).
- **Skrypty** Skrypty pomagające automatyzować zadania IIS.
- **Narzędzia** Łączy do narzędzi dla IIS, Windows PowerShell itp., których można używać od razu.
- **Informacje o produkcie** Łączy do informacji na temat właściwości i możliwości IIA NS Windows Server® 2008 i innych produktów podane, aby pomóc zmodyfikować Windows Server 2008 w firmie.
- **Zasoby** Łączy do przewodników, zasobów technicznych, webcastów, for i innych zasobów pomagających rozwiązywać problemy z IIS, Windows Server 2008 i innymi produktami, a jednocześnie ich używać.
- **Przykładowe rozdziały** Przegląd rozdziałów z piętnastu książek na temat Windows Server 2008 w formacie PDF.

Dodatkowa treść online

Nowy lub uaktualniony materiał uzupełniający tę książkę zostanie udostępniony online na witrynie sieci Web Microsoft Press Online Windows Server and Client. Będzie oparty na finalnym systemie Windows Server 2008 i obejmie aktualizację książki, artykuły, łącza do dołączonej treści, erratę, przykładowe rozdziały itp. Wspomniana witryna sieci Web będzie wkrótce dostępna pod adresem: <http://www.microsoft.com/learning/books/online/serverclient> i będzie okresowo uaktualniana.

Zasady wsparcia Resource Kit

Dołożyliśmy wszelkich starań, aby zapewnić ścisłość tej książki i dołączonej płyty CD. Microsoft Press udostępnia poprawki w języku angielskim dla tej książki przez sieć Web pod adresem:

<http://www.microsoft.com/learning/support/search.asp>.

Komentarze i pytania dotyczące tej książki lub zawartości dołączonej płyty oraz pytania, na które Czytelnik nie znalazł odpowiedzi w Knowledge Base, proszę wysłać do Microsoft Press, używając jednej z następujących metod:

E-mail:

rkinput@microsoft.com

Poczta tradycyjna:

Microsoft Press

Attn: Microsoft Internet Information Services 7.0 Resource Kit, Editor

One Microsoft Way

Redmond, WA 98052-6399

Wsparcie dla tego produktu nie jest oferowane przez podane powyżej adresy pocztowe. Informacje te znajdują się na witrynie sieci Web Microsoft Product Support pod adresem: <http://support.microsoft.com>.

Część I

Podstawy

W tej części:

Rozdział 1.	Wprowadzenie do IIS 7.0	3
Rozdział 2.	Architektura IIS 7.0.	27
Rozdział 3.	Podstawy struktury modularnej.	53
Rozdział 4.	System konfiguracyjny	63

Wprowadzenie do IIS 7.0

W tym rozdziale:

Przegląd IIS 7.0	4
Nowości w IIS 7.0	5
Podstawowe zadania administracyjne	15
Właściwości IIS 7.0 w Windows Server 2008 i Windows Vista.	22
Podsumowanie.	25
Dodatkowe zasoby	26

Microsoft Internet Information Services (Internetowe usługi informacyjne) 7.0 – w skrócie IIS 7.0 – dołączone do Windows Server 2008 to serwer sieci Web, który zapewnia bezpieczną, łatwą w zarządzaniu platformę do tworzenia i wiarygodnego udostępniania aplikacji i usług sieci Web. IIS 7.0 został całkowicie przeprojektowany i pod wieloma względami przewyższa poprzednie wersje. Dzięki nowej, modularnej i rozszerzalnej architekturze IIS 7.0 sprawia, że tworzenie, wdrażanie, konfigurowanie i zarządzanie aplikacjami WWW i infrastrukturą jest wydajniejsze niż kiedykolwiek poprzednio.

W uproszczeniu – IIS 7.0 jest najbardziej rozbudowaną platformą serwera sieci Web ze wszystkich wydanych przez Microsoft. Udostępnia szereg nowych możliwości usprawniających tworzenie, wdrażanie i zarządzanie aplikacjami i usługami sieci Web. Modularny projekt IIS 7.0 daje administratorom pełną kontrolę nad funkcjonalnością serwerów WWW, zapewniając rozszerzalną architekturę, która pozwala administratorom i deweloperom tworzyć dostosowane i specjalistyczne serwery sieci Web. Nowe możliwości administracyjne i rozproszony, bazujący na XML system konfiguracji sprawia, że wdrażanie i zarządzanie aplikacjami sieci Web na IIS 7.0 jest prostsze i wydajniejsze niż na dowolnym innym serwerze sieci Web. W dodatku nowe możliwości diagnostyki i rozwiązywania problemów w IIS 7.0 pozwalają zarówno administratorom, jak i deweloperom minimalizować potencjalny czas wyłączenia.

W tym rozdziale skupimy się na głównych, nowych właściwościach i funkcjach IIS 7.0 oraz na ich zaletach w porównaniu z poprzednimi wersjami IIS. Przyjrzymy się także podstawowym zadaniom administracyjnym i omówimy różnice dostępności funkcji IIS 7.0 w Windows Server 2008 i Windows Vista.

Przegląd IIS 7.0

IIS 7.0 dostarcza właściwości i funkcji, które pozwalają administratorom na wiarygodne i efektywne zarządzanie infrastrukturami sieci Web, zaś deweloperom na szybkie tworzenie aplikacji i usług sieci Web, a firmom hostingowym na efektywne kosztowo, skalowalne i wiarygodne wprowadzanie serwerów sieci Web dla szerokiego zakresu klientów.

Administratorom IIS 7.0 zapewnia bezpieczną, wiarygodną i łatwą do zarządzania platformę serwera sieci Web. Dostosowywalna instalacja IIS 7.0 zapewnia, że mogą one minimalizować powierzchnię ataku, a także wymagania dotyczące wprowadzania aktualizacji i ślady pamięci w infrastrukturze sieci Web. Model procesu IIS 7.0 sprawia, że witryny WWW i aplikacje są bezpieczniejsze przez automatyczne ich izolowanie, domyślnie zapewniają konfigurację piaskownic (sandbox) i unikalną identyfikację procesu.

IIS 7.0 redukuje złożoność zarządzania, zapewniając zbiór narzędzi, które czynią administrację infrastrukturami sieci Web bardziej efektywną. IIS Manager ma nową, opartą na zadaniach i skupioną na funkcjach konsolę, która do zadań administracyjnych dostarcza intuicyjnego interfejsu użytkownika. W dodatku do konsoli zarządzania usługami IIS, czyli IIS Manager, istnieje nowe narzędzie administracji z wiersza polecenia – dostawca WMI (Windows Management Instrumentation) i interfejs programowania aplikacji (API) .NET.

IIS 7.0 wspiera uproszczone zarządzanie farmami sieci Web, gdzie konfiguracja serwera tej sieci może być przechowywana wspólnie z kodem aplikacji sieci Web i jej treścią na scentralizowanym serwerze plików i może być współdzielona przez wiodące (front-end) serwery sieci Web na farmie.

IIS 7.0 pozwala administratorom na bezpieczne delegowanie kontroli administracyjnej dla witryn i aplikacji do programistów i właścicieli treści bez udzielania im administracyjnych przywilejów na serwerze, redukując w ten sposób nakład administracyjny i koszt własności. Używając IIS Manager z Windows Vista, Windows XP, Windows Server 2003 lub Windows Server 2008, deweloperzy i właściciele treści mogą zarządzać i witrynami, i aplikacjami, zdalnie będąc połączonymi z serwerem przez HTTPS z dowolnej lokalizacji.

W dodatku nowe możliwości rozwiązywania problemów i diagnostyki w IIS 7.0 pozwalają administratorom redukować czas wyłączenia serwera sieci Web.

Deweloperom IIS 7.0 zapewnia elastyczną, bardziej rozszerzalną platformę serwera sieci Web do tworzenia i wdrażania aplikacji na Windows Server 2008 i Windows Vista. Deweloperzy mogą budować aplikacje na IIS 7.0 z wykorzystaniem tego środowiska sieci Web, które wybiorą, w tym ASP.NET, klasycznego ASP, PHP, PERL, ColdFusion, Ruby i wielu innych.

IIS 7.0 umożliwiającą ogromną rozszerzalność. Mają całkowicie komponentową architekturę, w tym ponad 40 włączanych modułów, zbudowanych na szczycie publicznie rozszerzalnych interfejsów API. Deweloperzy mogą tworzyć nowe moduły lub zastępować je w natywnym lub zarządzanym kodzie, rozszerzać konfigurację IIS i budować rozszerzenia IIS Manager, które są z łatwością włączane do konsoli zarządzania.

IIS 7.0 mają bazujący na rozproszonych plikach system konfiguracji, który umożliwia przechowywanie ustawień IIS w plikach web.config wraz z ustawieniami ASP.NET. Ten zunifikowany system konfiguracyjny upraszcza tworzenie prekonfigurowanych aplikacji i umożliwia ich wdrażanie za pomocą xcopy do serwerów IIS 7.0.

W dodatku nowe możliwości diagnostyczne, w tym dostęp do informacji czasu wykonania i automatycznego śledzenia nieudanych żądań, pomagają deweloperom rozwiązywać problemy szybciej i minimalizować czas wyłączenia witryny WWW.

Hosterom IIS 7.0 zapewniają efektywną pod względem kosztów i bardziej skalowalną platformę serwera sieci Web w celu dostarczania wiarygodnego hostingu tej sieci dużemu zbiorowi klientów. Ponadto zmniejszają koszty przez dostarczenie nowej, skalowalnej i współdzielonej architektury, która jest zdolna do hostingu tysięcy witryn sieci Web na pojedynczym serwerze IIS 7.0 bez naruszenia izolacji czy niezawodności.

IIS 7.0 pozwalają hosterom sieci Web obsłużyć większą liczbę klientów dzięki używaniu nowego modułu FastCGI, który jest zdolny do dostarczania szybkiego i wiarygodnego hostingu dla PHP i innych środowisk sieci Web.

Ponadto IIS 7.0 dostarcza serwera FTP (File Transfer Protocol), który umożliwia prowadzącym hosting sieci Web oferowanie klientom w pełni zintegrowanej platformy WWW/FTP o nowoczesnych możliwościach publikacyjnych, takich jak FTP przez SSL (Secure Sockets Layer) i uwierzytelnianie oparte na członkostwie.

Nowości w IIS 7.0

IIS 7.0 został od podstaw całkowicie przeprojektowany i przebudowany. Nowe właściwości i funkcje zapewniają wiele nowych możliwości, które – za pomocą ziarnistej kontroli nad śladami serwera sieci Web – pozwalają administratorom i deweloperom na:

- minimalizowanie ryzyka związanego z aktualizacjami i bezpieczeństwem;
- szybkie implementowanie nowych rozwiązań sieci Web dzięki użyciu rozszerzalnego środowiska;
- szybsze wejście na rynek, przy uproszczonym wdrożeniu i konfiguracji aplikacji;
- redukcję kosztów administracyjnych dzięki bardziej efektywnemu zarządzaniu infrastrukturami sieci Web;
- redukcję czasu wyłączania witryny sieci Web dzięki szybkiemu rozpoznaniu błędnych aplikacji.

Te udoskonalenia stały się możliwe z powodu udoskonaleń IIS 7.0, takich jak:

- modularny, rozszerzalny rdzeń serwera sieci Web;
- zunifikowany system konfiguracji oparty na rozproszonych plikach;
- zintegrowane monitorowanie kondycji i diagnostyka;
- zbiór nowych narzędzi administracyjnych ze wsparciem delegacji.

W dodatku IIS 7.0 oferuje nową usługę WAS (Windows Process Activation Service), która eksponuje model przetwarzania IIS 7.0 do aplikacji i usług opartych na HTTP i innych.

Przyjrzymy się teraz bardziej szczegółowo tym nowościom i ich zaletom w porównaniu z poprzednimi wersjami IIS.

Rdzeń serwera sieci Web

Rdzeń serwera sieci Web IIS 7.0 został całkowicie przeprojektowany i bardzo różni się od IIS 6.0. Jego nowa i w pełni modularna architektura zapewnia dwa fundamentalne udoskonalenia, które stanowią podstawę wielu zalet dotyczących bezpieczeństwa, wydajności, skalowalności, zarządzalności i elastyczności. Tymi dwoma fundamentalnymi udoskonaleniami są modularność i rozszerzalność.

Modularność

W poprzednich wersjach IIS cała funkcjonalność była wbudowana w monolityczny serwer. Nie można było w łatwy sposób rozszerzyć lub zamienić danej funkcjonalności. W IIS 7.0 rdzeń serwera sieci Web ma całkowicie modularną architekturę. Wszystkie funkcjonalności serwera tej sieci są teraz zarządzane jako samodzielne komponenty. Rdzeń serwera sieci Web IIS 7.0 jest podzielony na ponad 40 oddzielnych komponentów, z których każdy implementuje szczególną właściwość lub funkcjonalność. Komponenty te są określane mianem modułów. W zależności od potrzeb moduły można dodawać usuwać i zastępować.

W IIS 7.0 czas działania ASP.NET jest w pełni zintegrowany z rdzeniem serwera sieci Web, zapewniając zunifikowany potok przetwarzania żądań. Kody zarówno natywne, jak i zarządzane są przetwarzane przez ten pojedynczy potok żądań. Wszystkie zdarzenia powiadomień w potoku żądania są eksponowane zarazem w natywnych i zarządzanych modułach. Integracja ta pozwala istniejącym właściwościom ASP.NET (w tym uwierzytelnianiu bazującym na formularzach, członkostwie, stanie sesji i wielu innych) na stosowanie jej do wszystkich typów treści, zapewniając spójne działanie całej aplikacji sieci Web.

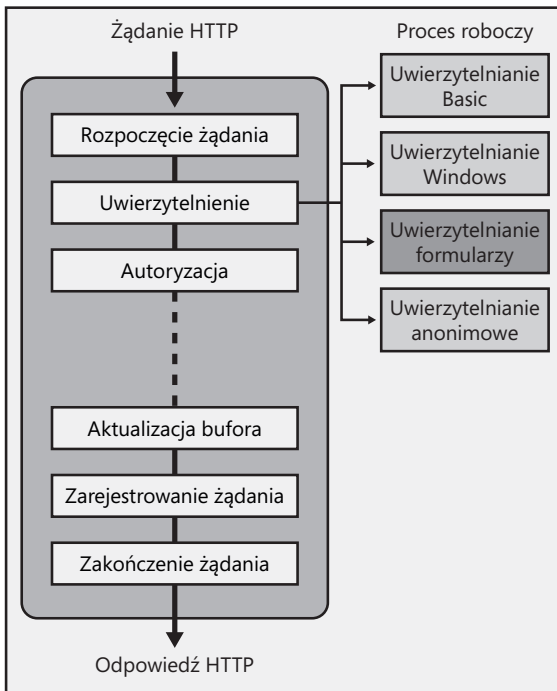
Na rysunku 1-1 pokazano zunifikowany potok przetwarzania żądań – z wieloma stanami na początku i końcu przetwarzania żądania. W stanie Authenticate Request (żądania uwierzytelniania) rysunek 1-1 przedstawia moduły uwierzytelniania, które są dostępne dla wszystkich żądań. Uwierzytelniania zarówno podstawowe (Basic), Windows i anonimowe są modułami natywnymi. Uwierzytelnianie Forms jest modulem zarządzanym. Zarazem natywne, jak i zarządzane moduły uwierzytelniania zapewniają usługi dowolnego typu treści, w tym zarządzanego kodu, kodu natywnego i plików statycznych.



Uwaga Więcej informacji na temat przetwarzania żądań znajduje się w rozdziale 2 „Architektura IIS 7.0”.

Modularność IIS 7.0 pozwala na wykonanie następujących czynności:

- **Zabezpieczenie serwera przez redukcję obszaru powierzchni ataku** Redukcja obszaru powierzchni ataku jest jednym z głównych kroków do zapewnienia bezpiecznego systemu. W IIS 7.0 funkcje serwera sieci Web, które nie są wymagane, mogą być bez szkody usunięte, co nie ma wpływu na funkcjonowanie aplikacji, redukując w ten sposób obszar powierzchni ataku.
- **Zwiększenie wydajności i redukcja śladu pamięci** Gdy usunie się niepotrzebne funkcje serwera sieci Web, wykorzystanie pamięci serwera jest redukowane. W dodatku ilość kodu, który wykonuje się przy każdym żądaniu, zostanie zredukowana, prowadząc do zwiększonej wydajności.



Rysunek 1-1 Zintegrowane przetwarzanie żądań IIS 7.0.

- **Budowa własnych i wyspecjalizowanych serwerów** Wybranie pewnego zbioru funkcjonalności serwera i usunięcie tych, które nie są wymagane, pozwala na budowanie własnych serwerów zoptymalizowanych do wykonywania pewnej funkcji, takiej jak buforowanie brzegowe (edge caching) lub równoważenie obciążenia.



Uwaga Więcej informacji na temat modularności serwera znajduje się w rozdziale 3 „Podstawy struktury modularnej”.

Rozszerzalność

Modularna architektura IIS 7.0 pozwala na budowanie komponentów serwera, które rozszerzają lub zastępują istniejącą funkcjonalność i dodają wartość do aplikacji WWW umieszczonych na serwerze IIS.

Rdzeń serwera sieci Web zawiera nowe Win32 API do budowy rdzennych modułów serwera. Można dodać nowe komponenty, aby rozszerzyć lub zastąpić istniejące właściwości serwera sieci Web swoimi własnymi rozszerzeniami rdzenia serwera sieci Web zbudowanymi za pomocą nowego API lub też rozszerzeniami firm trzecich.

Moduły rdzenia serwera sieci Web są nowe i jednocześnie potężniejsze od zastępowanych filtrów i rozszerzeń Internet Server Application Programming Interface (ISAPI), chociaż te filtry i rozszerzenia są nadal wspierane w IIS 7.0. Nowy model rozszerzalności C++ w IIS 7.0 korzysta z uproszczonego, obiektowego API promującego pisanie wydajnego kodu serwera w celu usunięcia problemów, które poprzednio pojawiały się podczas programowania ISAPI.

Co więcej – IIS 7.0 zawiera także dla rozszerzeń rdzenia serwera WWW wsparcie za pomocą .NET Framework. IIS 7.0 zintegrował istniejące IHttpModule API dla ASP.NET, pozwalając własnym modułom kodu zarządzanego na dostęp do wszystkich zdarzeń w toku żądania dla wszystkich żądań.

Integracja ASP.NET z IIS 7.0 pozwala na szybkie tworzenie modułów serwera z wykorzystaniem możliwości ASP.NET i .NET Framework zamiast korzystania z niskiego poziomu IIS C++ API. Zarządzane moduły ASP.NET mogą w pełni rozszerzać serwer oraz obsługiwać żądania dla wszystkich typów treści, takich jak na przykład ASP, CGI (Common Gateway Interface) i pliki statyczne.

Korzystając z ASP.NET lub natywnej rozszerzalności C++, deweloperzy mogą budować rozwiązania, które dodają wartość do wszystkich komponentów aplikacji, takich jak własne schematy uwierzytelniania, monitorowanie i rejestrowanie, filtrowanie zabezpieczeń, równoważenie obciążeń, przekierowywanie treści i zarządzanie stanem.



Uwaga Więcej informacji na temat rozszerzalności rdzenia serwera sieci Web znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

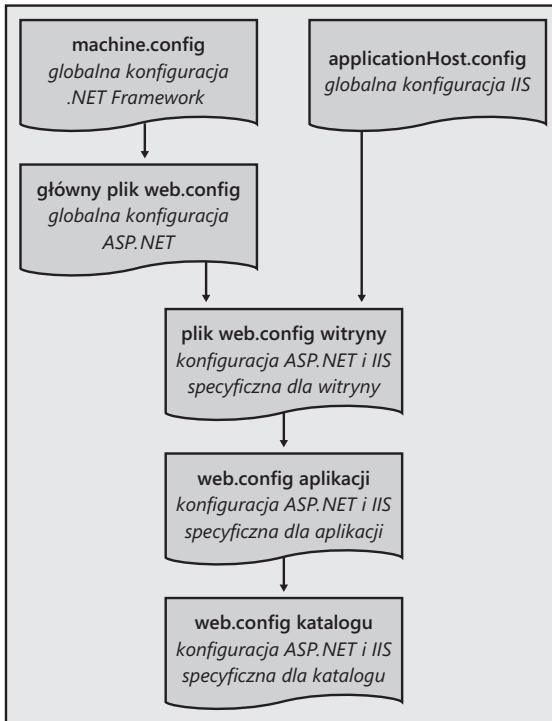
Konfiguracja

Wczesne wersje serwera IIS miały niewiele przechowywanych w rejestrze ustawień konfiguracyjnych. IIS 5.0 wprowadziły przechowywanie binarne, nazywane metabazą, dla zarządzania konfiguracją opartą na URL. W IIS 6.0 binarna metabaza została zastąpiona metabazą opartą o XML do przechowywania danych konfiguracyjnych. IIS 7.0 wprowadzają oparty o rozproszone pliki XML system konfiguracji, który pozwala administratorom na określanie ustawień dla IIS i jego funkcjonalności w plikach testowych XML przechowywanych z kodem i treścią. Pliki XML przechowują ustawienia konfiguracyjne dla całej platformy serwera sieci Web – w tym IIS, ASP.NET – i innych komponentów. Pliki przechowują ustawienia na poziomach serwera, witryny i aplikacji i mogą być opcjonalnie umieszczone na poziomie katalogów treści razem z treścią sieci Web, umożliwiając oddelegowanie zarządzania.

Ponieważ witryna sieci Web i ustawienia aplikacji nie są już dłużej przypisane do scentralizowanego miejsca przechowywania konfiguracji na lokalnej maszynie – jak w poprzednich wersjach IIS – ten rozproszony, oparty na plikach system konfiguracji znacznie upraszcza wdrażanie aplikacji za pomocą xcopy, gdzie konfiguracja jest kopiowana razem z kodem aplikacji i treścią. Ten system konfiguracyjny pozwala w dodatku na współdzielenie konfiguracji dla witryny lub aplikacji na farmie sieci Web.

Konfiguracja IIS 7.0 bazuje na magazynie konfiguracyjnym .NET Framework. Ten wspólny format pozwala ustawieniom konfiguracyjnym IIS na przechowywanie ASP.NET wraz z konfiguracją w hierarchii plików web.config, dostarczając jeden magazyn konfiguracyjny dla wszystkich ustawień konfiguracyjnych platformy sieci Web, które są dostępne przez wspólny zbiór API i zarazem przechowywane w spójnym formacie.

Ta rozproszona hierarcha konfiguracji obejmuje globalne, dotyczące komputera, pliki konfiguracyjne .NET Framework, machine.config i główny (root) web.config, globalny plik konfiguracji IIS applicationHost.config, a także rozproszone pliki web.config, zlokalizowane w witrynach sieci Web, aplikacjach i katalogach, jak pokazano na rysunku 1-2.



Rysunek 1-2 Rozproszony oparty o pliki magazyn konfiguracji

Globalne ustawienia .NET Framework dla komputera skonfigurowanego jako serwer są przechowywane w pliku `machine.config` zlokalizowanym w folderze `%SystemRoot%\Microsoft .NETFramework\<version>\config`. Natomiast globalne ustawienia ASP.NET dla serwera sieci Web są przechowywane w głównym pliku `web.config` zlokalizowanym w tym samym folderze na komputerze skonfigurowanym jako serwer.

IIS 7.0 przechowuje globalną konfigurację w pliku `applicationHost.config` zlokalizowanym w folderze `%SystemRoot%\System32\Inetsrv\Config`. `ApplicationHost.config` zawiera dwie główne części konfiguracyjne: `<system.applicationHost>` i `<system.webServer>`.

W sekcji `<system.applicationHost>` znajdują się ustawienia dla witryny, aplikacji, katalogu wirtualnego i puli aplikacji. Sekcja `<system.webServer>` zawiera konfiguracje wszystkich innych ustawień, w tym globalnych ustawień domyślnych sieci Web.

Konfiguracja specyficzna dla URL jest przechowywana w `applicationHost.config` przy pomocy znaczników `<location>`. IIS 7.0 odczytuje i zapisuje tę konfigurację na serwerze wraz z konfiguracją ASP.NET w hierarchii plików `web.config` dla witryn, aplikacji i katalogów treści.

Rysunek 1-3 ukazuje strukturę pliku `web.config` dla witryny i dziedziczenie tego pliku z globalnych plików konfiguracyjnych.

Administrator serwera może delegować różne poziomy hierarchii konfiguracji dla innych użytkowników, takich jak administrator witryny lub deweloper aplikacji. Domyślnie dostęp do zapisu ustawień konfiguracyjnych jest ograniczony tylko do administratora serwera. Może on delegować zarządzanie specyficznymi ustawieniami konfiguracyjnymi użytkownikom bez uprawnień administratora na komputerze skonfigurowanym jako serwer.

- Narzędzie wiersza polecenia Appcmd.exe zastępuje skrypty wiersza polecenia IIS 6.0. Zapewnia dostęp z wiersza polecenia do hierarchii plików konfiguracyjnych i innych ustawień serwera.
- Interfejs Microsoft.Web.Administration zapewnia pewne typowane zarządzanie API w celu dostępu zarządzalnego kodu do konfiguracji i innych ustawień serwera.
- Nowy dostawca WMI oferuje dostęp skryptowy do wszystkich konfiguracji IIS i ASP.NET. Starszy dostawca IIS 6.0 WMI jest nadal dostępny w celu kompatybilności wstecznej z istniejącymi skryptami.

Można także wykorzystać Windows PowerShell do potężnego dostępu skryptowego do rozproszonej hierarchii konfiguracyjnej.



Uwaga Więcej informacji na temat korzystania z PowerShell do zarządzania IIS 7.0 znajduje się w rozdziale 7 „Korzystanie z narzędzi wiersza polecenia”.

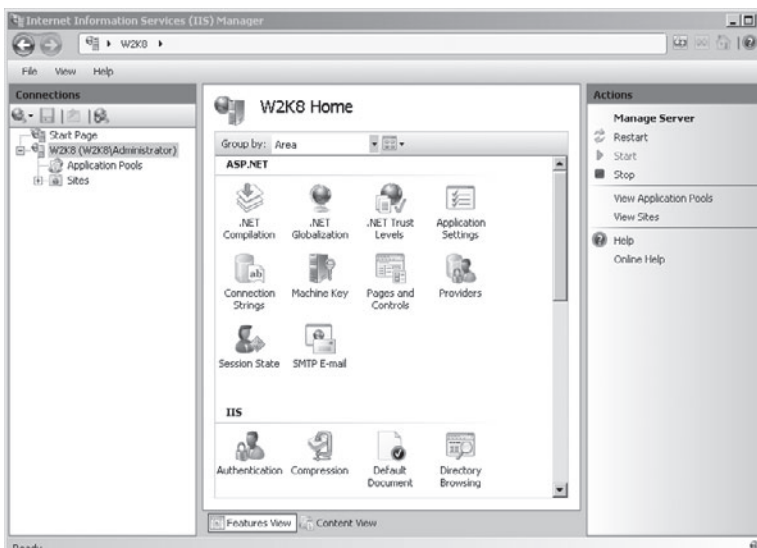
Również przystawka IIS 6.0 MMC jest dostarczana z Windows Server 2008, aby wspierać administrację zdalną i administrować witrynami FTP.

Wszystkie nowe narzędzia administracyjne w pełni wspierają nową, rozproszoną konfigurację IIS 7.0, a przy tym wszystkie pozwalają na delegację dostępu do konfiguracji poszczególnych witryn i aplikacji do użytkowników, którzy nie mają uprawnień administracyjnych na serwerze.



Uwaga Można oddzielnie instalować narzędzia administracyjne i komponenty serwera sieci Web.

Rysunek 1-4 ukazuje nowy interfejs użytkownika IIS Manager, który wyglądem przypomina przeglądarkę, i pasek adresu podobny do znajdującego się w Windows Explorer.



Rysunek 1-4 IIS Manager UI

Główna część okna IIS Manager jest podzielona na trzy obszary:

- Panel Connections (Połączenia) po lewej stronie okna IIS Manager pozwala na łączenie się z serwerami, witrynami i aplikacjami. Połączenia te są wyświetlane w postaci drzewa.
- Centralny obszar stanowiący przestrzeń roboczą jest zlokalizowany w środku okna IIS Manager. Przestrzeń robocza ma dwa widoki: Features View (Widok funkcji) i Content View (Widok zawartości).
 - Features View pozwala widzieć i konfigurować funkcjonalności aktualnie wybranej ścieżki konfiguracyjnej. Typowo każda funkcjonalność IIS Manager odpowiada sekcji konfiguracyjnej, która kontroluje odpowiednią właściwość serwera sieci Web.
 - Content View dostarcza widoku tylko do odczytu treści odpowiadającej aktualnie wybranej ścieżce konfiguracyjnej. W Content View, jeśli wybierze się węzeł drzewa w panelu drzewa Connections, jego treść jest wymieniana w przestrzeni roboczej.
- Panel Actions (Akcje) jest zlokalizowany po prawej stronie IIS Manager. Elementy panelu Actions dotyczą zadań i są specyficzne dla kontekstu.

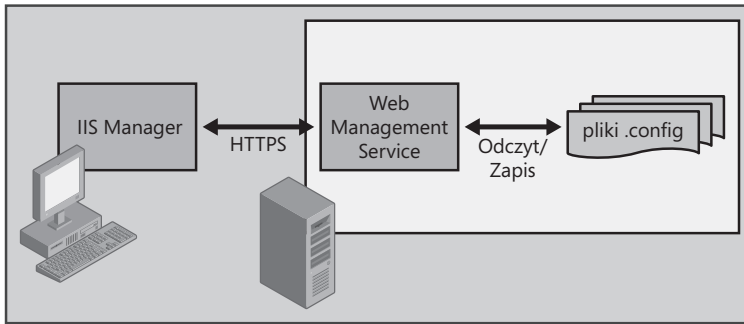
Tak jak w przypadku innych narzędzi administracyjnych delegacja zarządzania jest jedną z najważniejszych możliwości IIS Manager. Dzięki temu użytkownicy usług hostingu mogą uruchamiać IIS na swoich stacjach roboczych i łączyć się zdalnie, aby zarządzać witrynami i aplikacjami na serwerze, na którym są one umieszczone, bez posiadania dostępu administracyjnego do tego serwera. Do identyfikacji użytkowników IIS Manager może używać poświadczeń Windows, a także alternatywnych magazynów poświadczeń. Poświadczenia IIS Manager są szczególnie użyteczne w scenariuszach, w których nie chcemy tworzyć kont Windows dla wszystkich użytkowników zdalnych lub gdy poświadczenia są już umieszczone w systemie uwierzytelniania innym niż Windows i chcemy je zachować w pojedynczym magazynie.

IIS Manager wspiera zdalną administrację przez współdziałające z zaporami połączenia HTTPS, zezwalając na łatwą lokalną i zdalną administrację, nie wymagając otwierania na zaporze portów DCOM (Distributed Component Object Model) ani żadnych innych portów administracyjnych. W IIS 6.0 zdalne korzystanie z konsoli zarządzania odbywało się poprzez MMC i było zawsze włączone. Inaczej jest w IIS 7.0, gdzie zarządzanie zdalne przez IIS Manager jest domyślnie wyłączone i musi być włączane jawnie. W celu administracji zdalnej IIS 7.0 usługa WMSvs (Web Management Service) musi być zainstalowana na serwerze, natomiast zdalne połączenia do tej usługi muszą być włączone. WMSvc jest usługą Windows, która zapewnia możliwość zdalnego zarządzania witrynami i aplikacjami IIS 7.0 przy pomocy IIS Manager. Zdalną architekturę IIS Manager pokazano na rysunku 1-5.

IIS Manager w IIS 7.0 może być dostosowywany i rozszerzany. Ma własny plik konfiguracyjny, `administration.config`, który pozwala na dodawanie własnej funkcjonalności. Dowolne wtyczki administracyjne są po dodaniu integrowane z narzędziem i pojawiają się później wraz z właściwościami IIS i ASP.NET.



Uwaga Więcej informacji na temat IIS Manager znajduje się w rozdziale 6 „Korzystanie z konsoli IIS Manager”, natomiast dalsze informacje na temat `Appcmd.exe`, WMI i Microsoft Web Administration API znajduje się w rozdziale 7.



Rysunek 1-5 Zdalna architektura IIS Manager

Diagnostyka

IIS 7.0 wprowadza główne udoskonalenia w diagnostyce i w rozwiązywaniu problemów z witrynami sieci Web i aplikacjami. Pozwala na ich szybsze rozwiązywanie i jednocześnie na minimalizację czasu wyłączenia witryny sieci Web przez nowe możliwości diagnostyczne, w tym dostęp do informacji o czasie działania i automatyczne śledzenie błędnych żądań. Zmiany w diagnostyce i w rozwiązywaniu problemów w IIS 7.0 pozwalają w czasie rzeczywistym widzieć wszystkie żądania, które są wykonywane na serwerze, a także automatycznie przechwytywać błędy, dysponując szczegółowym dziennikiem śledzenia.

Dostęp do informacji o czasie wykonania

IIS 7.0 zawiera nowy interfejs programowania aplikacji RSCA (Runtime State and Control API) dostarczający w czasie rzeczywistym informacji na temat puli aplikacji, procesów roboczych, witryn, domen aplikacji i działających żądań.

RSCA ma służyć administratorom do dogłębnego przyglądania się bieżącemu stanowi działających obiektów, w tym aktualnych procesów roboczych i ich aktualnie wykonywanych żądań, a także umożliwić administratorom użycie tego samego API do kontroli nad tymi obiektami. RSCA pozwala osiągnąć szczegółowe dane czasu wykonania, które były poprzednio niedostępne.

Te informacje są eksponowane przez natywne COM (Component Object Model) API. Samo API jest opakowane i eksponowane przez nowego dostawcę IIS 7.0 WMI, Microsoft. Web.Administration API, narzędzie zarządzania wiersza polecenia Appcmd.exe i IIS Manager.

Na przykład korzystając z IIS Manager, administratorzy mogą osiągnąć informacje o czasie wykonania, bądź o tym, jakie żądania są aktualnie wykonywane, jak długo będą działać, które adresy URL wywołują, jaki klient je wywoła i jaki jest status tych adresów.

Śledzenie błędnych żądań

IIS 7.0 udostępnia szczegółowe zdarzenia śledzenia na ścieżce żądań i odpowiedzi, pozwalając na śledzenie żądań na drodze do IIS poprzez potok przetwarzania żądań IIS do dowolnych, istniejących kodów poziomu strony, a następnie z powrotem do odpowiedzi. Te szczegółowe zdarzenia śledzenia pozwalają nie tylko poznać ścieżkę żądania i dowolne

informacje o błędach, jakie powstały w wyniku żądania, ale także czas trwania i inne informacje diagnostyczne, towarzyszące rozpoznawaniu wszystkich typów błędów i sytuacji, gdy system przestanie odpowiadać.

Takie problemy, jak słaba wydajność dla pewnych żądań, błędy związane z uwierzytelnianiem dla innych żądań lub błąd serwera 500, mogą być często trudne do rozwiązania, dopóki nie przechwyci się śladu problemu w czasie jego wystąpienia. Wtedy pomocne może być śledzenie błędnych żądań. Z założenia śledzenie ma buforować ślady zdarzeń dla żądań i zapisywać je na dysku w dzienniku śledzenia, jeśli żądanie nie zostanie obsłużone pomyślnie. Można skonfigurować IIS 7.0, aby automatycznie przechwytywał pełne dzienniki śledzenia w formacie XML dla dowolnego zadanego żądania bazując na upływającym czasie i kodach błędów odpowiedzi.

Możliwości diagnostyczne w IIS 7.0 są rozszerzalne, a nowe zdarzenia śledzenia mogą być wprowadzane do własnych modułów.



Uwaga Więcej informacji na temat diagnostyki i rozwiązywania problemów znajdują się w rozdziale 16 „Śledzenie i rozwiązywanie problemów”.

Windows Process Activation Service

IIS 7.0 zapewnia nową i niezależną od protokołu usługę Windows Process Activation Service (WAS), która jest rozszerzonym i uogólnionym następcą Windows Activation Service w IIS 6.0. Model aktywacji procesów HTTP został wprowadzony z pulami aplikacji w IIS 6.0. Ta usługa została rozszerzona w IIS 7.0, aby stała się dostępna nie tylko dla aplikacji sieci Web. Ma ona zdolność odbierania żądań lub komunikatów przez dowolny inny protokół i wspiera aktywację poprzez włączanie arbitralnych odbiorników protokołów. Poza niezależnością od protokołu WAS zapewnia wszystkim typom aplikacji, które są aktywowane przez komunikaty, inteligentne zarządzanie zasobami, aktywację procesu na żądanie, monitorowanie kondycji, wykrywanie błędów i odzyskiwanie. WCF (Windows Communication Foundation) jest dostarczane z adapterami protokołów, które mogą podnosić możliwości WAS. Korzystając z tych możliwości, można znacznie poprawić niezawodność, a także wykorzystanie zasobów usług WCF.



Uwaga Więcej informacji na temat WAS i wsparcia protokołów innych niż HTTP w IIS 7.0 znajduje się w rozdziale 2 „Architektura IIS 7.0”.

Kompatybilność aplikacji

IIS 7.0 są kompatybilne z poprzednimi wersjami IIS. Większość istniejących aplikacji ASP, ASP.NET 1.1 i ASP.NET 2.0 powinna działać na IIS 7.0 bez zmian kodu, wykorzystując wsparcie kompatybilnego ISAPI.

Działają nadal również wszystkie rozszerzenia ISAPI i większość filtrów ISAPI. Jednak filtry ISAPI, które używają powiadamiania READ RAW DATA nie są wspierane w IIS 7.0.

Istniejącym interfejsom ADSI (Active Directory Service Interfaces) i skryptom WMI IIS 7.0 dostarcza analogicznych właściwości z poprzednimi wersjami, pozwalając tym

skryptom na korzystanie ze starszych interfejsów konfiguracyjnych z wykorzystaniem warstwy Metabase Compatibility.



Uwaga Więcej informacji na temat kompatybilności aplikacji znajduje się w rozdziale 11 „Utrzymywanie środowisk rozwijania aplikacji”.

Podstawowe zadania administracyjne

Serwer WWW do rozpoczęcia udostępnienia treści musi mieć podstawową konfigurację: witrynę, aplikację, katalog wirtualny i pulę aplikacji. IIS 7.0 zapewniają domyślną konfigurację, która obejmuje Default Web Site z główną aplikacją zamapowaną do fizycznego katalogu `%SystemDrive%\Inetpub\Wwwroot` i z domyślną pulą aplikacji nazwaną DefaultAppPool, do której dana aplikacja należy.

Jednak może pojawić się potrzeba, aby utworzyć własną witrynę, dodać do niej aplikację, następnie wirtualny katalog do aplikacji, utworzyć nową pulę aplikacji i przypisać do niej aplikację. Następne podrozdziały opisują, jak wykonać te podstawowe zadania administracyjne za pomocą IIS Manager.



Uwaga Informacje na temat wykonywania innych, częstych zadań administracyjnych znajdują się w dodatku J „Realizacja typowych zadań administracyjnych przy użyciu IIS Manager”.

IIS Manager uruchamia się za pomocą skrótu Internet Information Services (IIS) Manager (Menedżer internetowych usług informacyjnych (IIS)) z grupy programów Administrative Tools (Narzędzia administracyjne).

Tworzenie witryny sieci Web

Witryna jest kontenerem dla aplikacji i katalogów wirtualnych. Każda witryna może być dostępna przez jedno unikalne powiązanie lub więcej. Powiązanie (binding) obejmuje protokół powiązania i informacje powiązania. Protokół powiązania definiuje protokół, przez który odbywa się komunikacja pomiędzy serwerem IIS 7.0 a klientem sieci Web, takim jak przeglądarka. Informacje powiązania definiują informacje, które służą do dostępu do tej witryny. Na przykład protokołem powiązania dla witryny sieci Web może być HTTP lub HTTPS, a informacje powiązania są kombinacją adresu IP, portu i opcjonalnie nagłówka hosta.

Aby utworzyć witrynę sieci Web przy pomocy IIS Manager, należy:

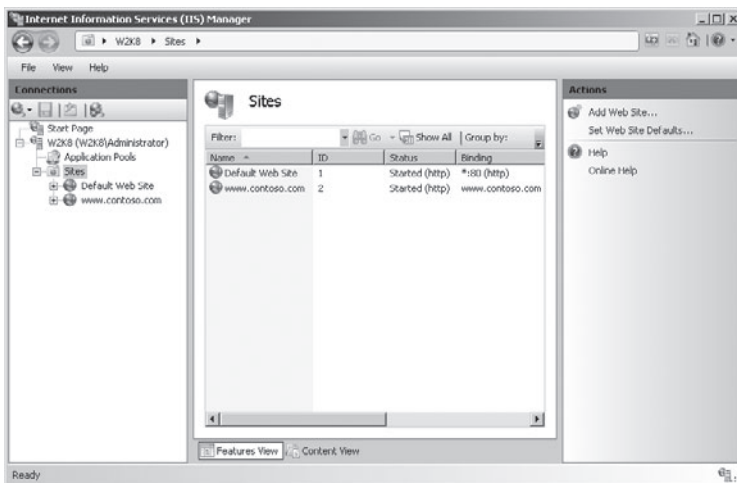
1. W panelu Connections, rozwinąć węzeł serwera, kliknąć prawym przyciskiem myszy węzeł Sites (Witryny), a następnie Add Web Site (Dodaj witrynę sieci Web). Pojawi się okno dialogowe Add Web Site (Dodawanie witryny sieci Web).
2. W polu Site Name (Nazwa witryny) wpisać nazwę witryny sieci Web, na przykład **www.contoso.com**.

The screenshot shows the 'Add Web Site' dialog box. It contains the following fields and controls:

- Site name:** A text input field.
- Application pool:** A dropdown menu with a 'Select...' button next to it.
- Content Directory:**
 - Physical path:** A text input field with a browse button (...).
 - Pass-through authentication:** A section containing 'Connect as...' and 'Test Settings...' buttons.
- Binding:**
 - Type:** A dropdown menu showing 'http'.
 - IP address:** A dropdown menu showing 'All Unassigned'.
 - Port:** A text input field showing '80'.
 - Host name:** A text input field with an example: 'Example: www.contoso.com or marketing.contoso.com'.
- Start Web site immediately:** A checked checkbox.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

3. Jeśli chcemy przypisać inną pulę aplikacji niż wymieniona w polu Application Pool (Pula aplikacji), należy kliknąć Select (Wybierz). Następnie w oknie dialogowym Select Application Pool (Wybieranie puli aplikacji) wybrać pulę aplikacji z listy rozwijanej Application Pool (Pula aplikacji) i kliknąć OK.
4. W polu Physical Path (Ścieżka fizyczna) trzeba wpisać fizyczną ścieżkę do folderu witryny WWW lub nawigować do niego, korzystając z przycisku przeglądania (...). Jeżeli wprowadzona ścieżka fizyczna wskazuje zdalny udział, należy kliknąć Connect As (Połącz jako) i określić wymagane poświadczenie. Jeśli nie jest wymagane poświadczenie dostępu do ścieżki, trzeba wybrać opcję Application User (Pass-Thru Authentication) (Użytkownik aplikacji (uwierzytelnianie przekazywane)) w oknie dialogowym Connect As.
5. Opcjonalnie: kliknąć Test Settings (Testuj ustawienie), aby zweryfikować określone ustawienia.
6. By skonfigurować wymagane powiązania dla nowej witryny, należy:
 - ❑ Jeżeli korzysta się z HTTPS w celu dostępu do witryny Web, należy zmienić protokół z HTTP na HTTPS na liście rozwijanej Type (Typ).
 - ❑ Jeżeli ma się dedykowany statyczny adres IP witryny, należy wpisać go w polu IP Address (Adres IP). Jeżeli nie ma się statycznego adresu IP witryny, trzeba zostawić domyślną wartość All Unassigned (Wszystkie nieprzypisane).
 - ❑ Jeśli witryna będzie korzystać z innego numeru portu niż domyślny numer portu 80, należy go wpisać w polu Port (Port).
 - ❑ Jeżeli witryna będzie używać nagłówka hosta, należy wpisać nazwę nagłówka hosta dla witryny w polu Host Name (Nazwa hosta). Na przykład można wpisać **www.contoso.com**.
7. Jeśli chcemy, aby witryna sieci Web była dostępna natychmiast, należy zaznaczyć pole wyboru Start Web Site Immediately (Natychmiast uruchom witrynę sieci Web).

8. Kliknąć OK. Nowa witryna sieci Web została utworzona i pojawiła się w panelu Connections.



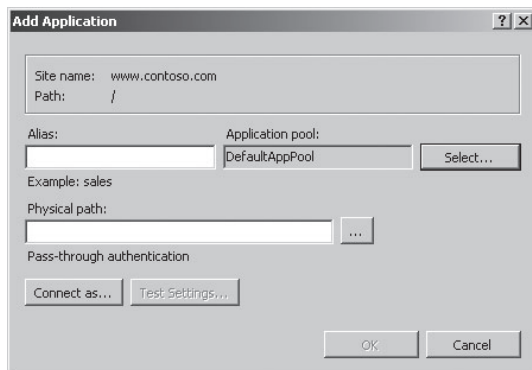
Tworzenie aplikacji

Aplikacja jest grupą plików dostarczającą treść lub usługi przez protokoły, takie jak HTTP. Gdy aplikacja jest tworzona, jej ścieżka staje się częścią URL.

Witryna może zawierać wiele aplikacji, w tym domyślną aplikację witryny nazywaną aplikacją główną (root). Poza przynależnością do witryny aplikacja przynależy również do puli aplikacji, która izoluje ją od aplikacji w innych pulach na serwerze.

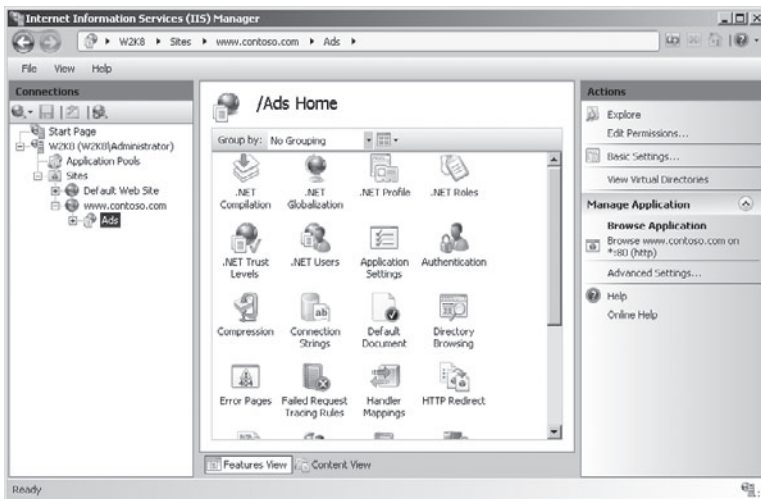
Aby utworzyć aplikację za pomocą IIS Manager, należy:

1. W panelu Connections kliknąć prawym przyciskiem myszy witrynę, w której ma działać nowa aplikacja. Następnie wybrać Add Application (Dodaj aplikację). Pojawi się okno dialogowe Add Application (Dodawanie aplikacji).



2. W polu Alias (Alias) wpisać wartość URL aplikacji, taki jak **Ads**. Wartość ta służy do tego, by uzyskać dostęp do tej aplikacji w URL.

3. Jeżeli chcemy przypisać inną pulę aplikacji niż wymieniona w polu Application Pool (Pula aplikacji), trzeba kliknąć Select (Wybierz). Następnie w oknie dialogowym Select Application Pool (Wybieranie puli aplikacji) wybrać pulę aplikacji z listy rozwijanej Application Pool (Pula aplikacji) i kliknąć OK.
4. W polu Physical Path (Ścieżka fizyczna) należy wpisać fizyczną ścieżkę do folderu witryny sieci Web lub nawigować do tego folderu za pomocą przycisku przeglądania (...). Jeśli fizyczna ścieżka, którą się wpisało, wskazuje udział zdalny, trzeba kliknąć Connect As (Połącz jako) i podać wymagane poświadczenia. Jeśli żadne poświadczenia nie są potrzebne w celu uzyskania dostępu do ścieżki, należy wybrać opcję Application User (Pass-Thru Authentication) (Użytkownik aplikacji (uwierzytelnianie przekazywane)) w oknie dialogowym Connect As (Łączenie jako).
5. Opcjonalnie: kliknąć Test Settings (Testuj ustawienia), aby zweryfikować określone ustawienia.
6. Kliknąć OK. Nowa aplikacja została utworzona i pojawiła się w panelu Connections.



Tworzenie katalogu wirtualnego

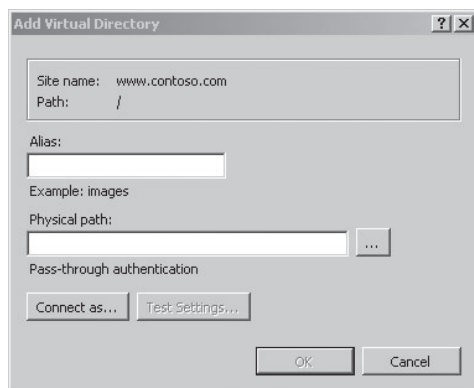
Katalog wirtualny to nazwa (zwana także ścieżką), która wchodzi w skład adresu URL i jest mapowana na katalog fizyczny znajdujący się na lokalnym lub zdalnym serwerze. Żądania odwołujące się do takiego katalogu kierowane są do zawartości odpowiedniego katalogu fizycznego, takiej jak strona sieci Web lub lista zawartości.

Aplikacja może zawierać wiele wirtualnych katalogów. Każda aplikacja musi mieć główny (root) katalog wirtualny, który mapuje aplikację do katalogu fizycznego zawierającego treść aplikacji.

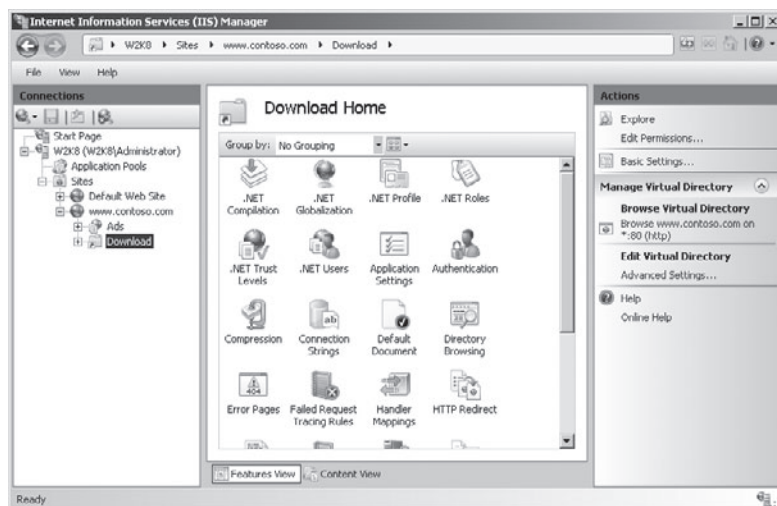
Aby utworzyć katalog wirtualny za pomocą IIS Manager, należy:

1. W panelu Connections trzeba kliknąć prawym przyciskiem myszy witrynę, w której ma się pojawić katalog wirtualny. Następnie należy wybrać Add Virtual Directory

(Dodaj katalog wirtualny). Pojawi się okno dialogowe Add Virtual Directory (Dodawanie katalogu wirtualnego).



2. W polu Alias wpisać wartość URL katalogu wirtualnego, taką jak **Download**. Służy ona do dostępu do katalogu w URL.
3. W polu Physical Path (Ścieżka fizyczna) wpisać fizyczną ścieżkę do folderu witryny WWW lub nawigować do tego folderu za pomocą przycisku przeglądania (...). Jeżeli fizyczna ścieżka, którą się wprowadziło, wskazuje udział zdalny, należy kliknąć Connect As (Połącz jako) i określić wymagane poświadczenie. Jeśli żadne poświadczenia nie są wymagane w celu uzyskania dostępu do tej ścieżki, trzeba wybrać opcję Application User (Pass-Thru Authentication) (Użytkownik aplikacji (uwierzytelnianie przekazywane)) w oknie dialogowym Connect As (Łączenie jak).
4. Opcjonalnie: kliknąć Test Settings (Testuj ustawienia), aby zweryfikować określone ustawienia.
5. Kliknąć OK. Nowy katalog wirtualny został utworzony i pojawił się w panelu Connections.



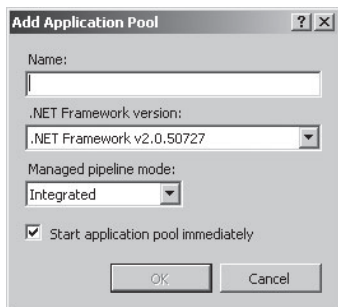
Tworzenie puli aplikacji

Pula aplikacji jest grupą jednej lub wielu aplikacji, które obsługują proces roboczy lub zbiór procesów roboczych. Pule aplikacji określają granice dla tych aplikacji, które zawierają, zapewniając izolację pomiędzy aplikacjami działającymi w różnych pulach.

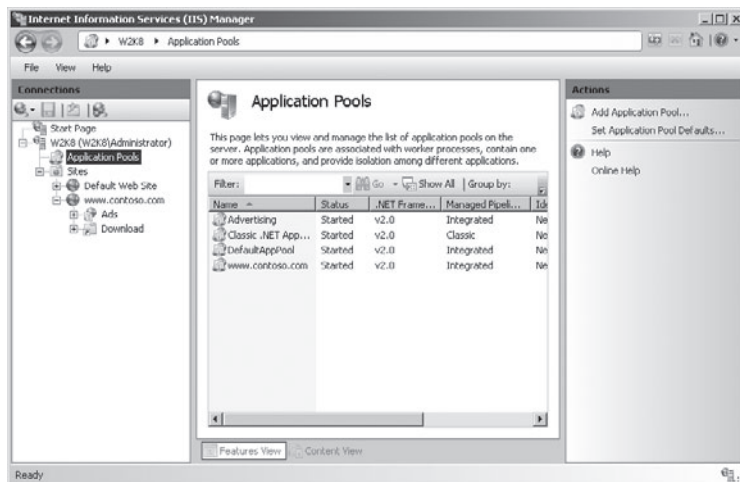
W IIS 7.0 żądania ASP.NET w pulach aplikacji mogą być wykonywane w jeden z dwóch zarządzanych trybów potokowych: zintegrowanego lub klasycznego. W trybie zintegrowanym serwer używa zunifikowanego, czyli zintegrowanego potoku, do przetwarzania żądania. W trybie klasycznym serwer przetwarza żądania ASP.NET, korzystając z dwóch różnych potoków IIS i ASP.NET w taki sam sposób, w jaki aplikacje działały w IIS 6.0.

Aby utworzyć pulę aplikacji za pomocą IIS Manager, należy:

1. W panelu Connections rozwinąć węzeł serwera i kliknąć prawym przyciskiem myszy węzeł Application Pools (Pule aplikacji). Wybrać Add Application Pool (Dodaj pulę aplikacji). Pojawi się okno dialogowe Add Application Pool (Dodawanie puli aplikacji).



2. W polu Name (Nazwa) wpisać nazwę puli aplikacji, na przykład **Advertising**.
3. Z listy rozwijanej .NET Framework Version (Wersja architektury .NET Framework) wybrać wersję .NET Framework wymaganą przez zarządzane aplikacje, moduły i programy obsługi. Jeśli aplikacje, które uruchamia się w tej puli, nie wymagają .NET Framework, trzeba wybrać No Managed Code (Bez kodu zarządzanego).
4. Z listy rozwijanej Managed Pipeline Mode (Zarządzany tryb potokowy) wybrać jedną z następujących opcji:
 - ☐ **Integrated (zintegrowany)** Należy wybrać ten tryb, jeżeli chcemy użyć zintegrowanego potoku żądań IIS i ASP.NET. Jest to tryb domyślny.
 - ☐ **Classic (klasyczny)** Należy wybrać ten tryb, jeżeli chcemy zastosować oddzielne potoki żądań IIS i ASP.NET.
5. Domyślnie zaznaczone jest pole wyboru Start Application Pool Immediately (Uruchom pulę aplikacji natychmiast). Jeżeli nie chcemy uruchamiać tej puli, należy wyczyścić pole.
6. Kliknąć OK. Nowa pula aplikacji zostanie utworzona i pojawi się na liście Application Pools.

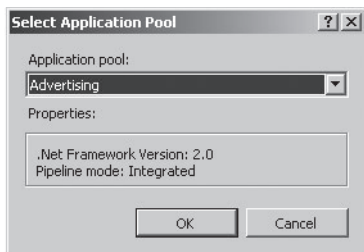


Przypisywanie aplikacji do puli aplikacji

Można przypisać aplikację do puli aplikacji, jeśli chcemy izolować tę aplikację od innych, działających na serwerze. Można przypisać wiele aplikacji do tej samej puli, jeśli wszystkie aplikacje korzystają z tych samych ustawień konfiguracyjnych czasu wykonania, na przykład ustawień procesu roboczego lub wersji ASP.NET.

Aby przypisać aplikację do puli aplikacji za pomocą IIS Manager, należy:

1. W panelu Connections, kliknąć prawym przyciskiem myszy aplikację, którą chcemy przypisać do innej puli aplikacji, wybrać Manage Application (Zarządzaj aplikacją), a następnie kliknąć Advanced Settings (Ustawienia zaawansowane).
2. Na stronie Advanced Settings (Ustawienia zaawansowane) wybrać Application Pool (Pula aplikacji) i kliknąć przycisk przeglądania. Pojawi się okno dialogowe Select Application Pool (Wybieranie puli aplikacji).



3. Wybrać pulę aplikacji, w której ma działać aplikacja.
4. Kliknąć OK. Aplikacja zostanie przypisana do puli aplikacji.

Właściwości IIS 7.0 w Windows Server 2008 i Windows Vista

IIS 7.0 są częścią Windows Server 2008 i Windows Vista. Jednak dostępność właściwości IIS 7.0 jest różna w Windows Server 2008 i w różnych edycjach Windows Vista.

Windows Server 2008 obejmuje wszystkie właściwości IIS 7.0. IIS 7.0 są dostępne we wszystkich wersjach Windows Server 2008. Nie ma różnicy funkcjonalności pomiędzy edycjami. IIS 7.0 są dostępne na platformach 32- i 64-bitowych.

IIS 7.0 są wspierane w instalacjach Server Core systemu Windows Server 2008. IIS 7.0 na Server Core stanowią serwer sieci Web osadzony na minimalnym, lekkim serwerowym systemie operacyjnym, mającym mniejsze wymagania pojemności dysku, mniejsze wykorzystanie pamięci, zredukowaną powierzchnię ataku i mniejsze potrzeby serwisowe. Instalacja IIS 7.0 na Windows Server 2008 Server Core różni się od zwykłej instalacji Windows Server 2008 IIS 7.0. Na Server Core nie ma powłoki Windows ani .NET Framework. Dlatego nie jest dostępny IIS Manager, więc nie można uruchamiać modułów, programów obsługi ani aplikacji ASP.NET na Server Core. Można jednak uruchamiać ASP, PHP, CGI i inne niezarządzane kody aplikacji na instalacjach Server Core IIS 7.0.



Uwaga Więcej informacji na temat instalacji IIS 7.0 na Server Core znajduje się w rozdziale 5 „Instalowanie IIS 7.0”.

W edycjach Windows Vista IIS 7.0 zapewniają deweloperom internetowym platformę sieci Web do budowania i testowania aplikacji tej sieci dla IIS 7.0, a także włączają infrastrukturę aktywacji procedur i zarządzania dla aplikacji Microsoft WCF (Windows Communication Foundation). Infrastruktura ta jest zapewniajana przez Windows Process Activation Service.

Właściwości IIS 7.0 dostępne w instalacjach Windows Vista zależą od edycji Windows Vista w następujący sposób:

- W edycjach Windows Vista Starter i Home komponenty IIS 7.0 oferują jedynie wsparcie infrastruktury dla WCF, ale nie zapewniają serwera sieci Web wspierającego treść statyczną, klasyczne ASP czy ASP.NET.
- W edycji Windows Vista Home Premium dostępna jest większość funkcjonalności IIS 7.0 Web Server wymaganych do tworzenia witryn sieci Web. Jednak nie są dostępne serwer FTP, zaawansowane uwierzytelnianie, autoryzacja sieci Web, czy zdalna administracja.
- W edycjach Windows Vista Business, Enterprise i Ultimate dostępne są wszystkie właściwości IIS 7.0 z wyjątkiem administracji zdalnej.

Tabela 1-1 wymienia listę dostępnych funkcjonalności w Windows Server 2008 i edycjach Windows Vista. Właściwości pogrupowano w następujące kategorie:

- ogólne właściwości HTTP,
- właściwości tworzenia aplikacji,
- właściwości kondycji i diagnostyki,
- właściwości bezpieczeństwa,

- właściwości wydajności,
- narzędzia zarządzania,
- Windows Process Activation Service,
- właściwości usługi publikowania FTP (File Transfer Protocol),
- limity równoległych połączeń.

W każdej kategorii dostępność właściwości zostały opisane w sposób następujący:

- **Domyślna** Właściwość jest dostępna domyślnie podczas instalacji IIS 7.0. Można nie instalować tej właściwości, jeśli nie jest potrzebna.
- **Dostępna** Właściwość ta jest dostępna, ale domyślnie nie jest zaznaczona, gdy instaluje się IIS 7.0. Można ją zainstalować, jeśli się jej potrzebuje.
- **Niedostępna** Ta właściwość jest niedostępna i nie może zostać zainstalowana wraz z IIS 7.0.

Tabela 1-1 Właściwości IIS Windows Server 2008 i Windows Vista

Nazwa właściwości	Edycje Windows Server 2008	Edycje Windows Vista		
		Ultimate, Business i Enterprise	Home Premium	Home Basic i Starter
Ogólne właściwości HTTP				
Treść statyczna	Domyślna	Domyślna	Domyślna	Niedostępna
Domyślny dokument	Domyślna	Domyślna	Domyślna	Niedostępna
Przeglądanie katalogu	Domyślna	Domyślna	Domyślna	Niedostępna
Błędy HTTP	Domyślna	Domyślna	Domyślna	Domyślna
Przekierowywanie HTTP	Domyślna	Domyślna	Domyślna	Domyślna
Właściwości tworzenia aplikacji				
ASP.NET	Dostępna	Dostępna	Dostępna	Niedostępna
Rozszerzenia .NET	Domyślna	Domyślna	Domyślna	Domyślna
ASP	Dostępna	Dostępna	Dostępna	Niedostępna
CGI	Dostępna	Dostępna	Dostępna	Niedostępna
Rozszerzenia ISAPI	Dostępna	Dostępna	Dostępna	Niedostępna
Filtry ISAPI	Dostępna	Dostępna	Dostępna	Niedostępna
Server-Side Includes	Dostępna	Dostępna	Dostępna	Niedostępna
Właściwości kondycji i diagnostyki				
Rejestrowanie HTTP	Domyślna	Domyślna	Domyślna	Domyślna
Narzędzia rejestrowania	Domyślna	Domyślna	Domyślna	Domyślna
Monitor żądań	Domyślna	Domyślna	Domyślna	Domyślna
Śledzenie	Domyślna	Domyślna	Domyślna	Domyślna
Własne logowanie	Dostępna	Dostępna	Dostępna	Niedostępna

Ciąg dalszy na stronie następnej

Tabela 1-1 Właściwości IIS Windows Server 2008 i Windows Vista

Nazwa właściwości	Edycje Windows Server 2008	Edycje Windows Vista		
		Ultimate, Business i Enterprise	Home Premium	Home Basic i Starter
Logowanie ODBC	Dostępna	Dostępna	Niedostępna	Niedostępna
Właściwości zabezpieczeń				
Podstawowe uwierzytelnianie	Dostępna	Dostępna	Dostępna	Niedostępna
Uwierzytelnianie Windows	Dostępna	Dostępna	Niedostępna	Niedostępna
Uwierzytelnianie Digest	Dostępna	Dostępna	Niedostępna	Niedostępna
Uwierzytelnianie przez mapowanie certyfikatów klienckich	Dostępna	Dostępna	Niedostępna	Niedostępna
Uwierzytelnienie przez mapowanie certyfikatów klienckich IIS	Dostępna	Dostępna	Niedostępna	Niedostępna
Autoryzacja URL	Dostępna	Dostępna	Dostępna	Dostępna
Filtrowanie żądań	Dostępna	Dostępna	Dostępna	Dostępna
Ograniczenia IP i domeny	Dostępna	Dostępna	Dostępna	Dostępna
Właściwości wydajności				
Kompresja treści statycznej	Domyślna	Domyślna	Domyślna	Domyślna
Kompresja treści dynamicznej	Dostępna	Dostępna	Dostępna	Dostępna
Narzędzia zarządzania				
Menedżer internetowych usług informacyjnych (IIS Manager)	Domyślna	Domyślna	Domyślna	Niedostępna
Skrypty i narzędzia zarządzania IIS	Dostępna	Dostępna	Dostępna	Dostępna
Usługa zarządzania	Dostępna	Dostępna	Dostępna	Niedostępna
Kompatybilność zarządzania IIS 6.0	Dostępna	Dostępna	Dostępna	Dostępna
Kompatybilność z metabazą IIS	Dostępna	Dostępna	Dostępna	Dostępna
Kompatybilność z IIS 6 WMI	Dostępna	Dostępna	Dostępna	Niedostępna

Tabela 1-1 Właściwości IIS Windows Server 2008 i Windows Vista

Nazwa właściwości	Edycje Windows Server 2008	Edycje Windows Vista		
		Ultimate, Business i Enterprise	Home Premium	Home Basic i Starter
Narzędzia skryptowe IIS 6	Dostępna	Dostępna	Dostępna	Niedostępna
Konsola zarządzania IIS 6	Dostępna	Dostępna	Dostępna	Niedostępna
Właściwości Windows Process Activation Service				
Model procesu	Domyślna	Domyślna	Domyślna	Domyślna
Środowisko.NET	Dostępna	Dostępna	Dostępna	Dostępna
Interfejsy konfiguracyjne API	Dostępna	Dostępna	Dostępna	Dostępna
Właściwości usługi publikowania FTP (File Transfer Protocol)				
Serwer FTP	Dostępna	Dostępna	Niedostępna	Niedostępna
Konsola zarządzania FTP	Dostępna	Dostępna	Niedostępna	Niedostępna
Limity równoległych połączeń				
Limity równoległych połączeń	Bez limitu	10	3	3

Podsumowanie

IIS 7.0 zostały całkowicie przeprojektowane i przebudowane od podstaw. Usługi te mają znaczne zalety w porównaniu z poprzednimi wersjami i sprawiają, że tworzenie, wdrażanie, konfigurowanie i zarządzanie aplikacjami sieci Web i infrastrukturą staje się łatwiejsze i wydajniejsze niż kiedykolwiek wcześniej.

IIS 7.0 dostarczają wiele nowych potężnych właściwości i funkcjonalności opartych na głównych udoskonaleniach, takich jak:

- **Modularność** Architektura IIS 7.0 jest w całości zbudowana z komponentów. Pozwala administratorom na wybór tego, które właściwości mają być zainstalowane i mają działać na serwerze WWW. Dysponując ponad 40 modułami właściwości, które mogą być instalowane niezależnie, administratorzy mogą zredukować potencjalną powierzchnię ataku i zmniejszać wymagania wykorzystania zasobów serwera.
- **Rozszerzalność** Rdzenne właściwości serwera sieci Web IIS 7.0 zostały zbudowane przy użyciu nowego zbioru wszechstronnych, publicznych interfejsów API, których programiści mogą używać do rozszerzania, zastępowania i dodawania funkcjonalności do serwera sieci Web. Interfejsy API są dostępne jako natywne Win32 API, a także zarządzane .NET Framework API. Deweloperzy mogą także rozszerzać konfigurację IIS i budować rozszerzenie IIS Manager, które mogą być w prosty sposób włączane w konsolę zarządzania.

- **Ujednolicony system rozproszonej konfiguracji** IIS 7.0 zapewniają ujednolicony, rozproszony i bazujący na plikach system konfiguracji do przechowywania wszystkich ustawień IIS i ASP.NET w pojedynczym, tekstowym formacie XML w hierarchii plików konfiguracyjnych, gdzie pliki te są przechowywane razem z treścią witryny sieci Web i aplikacji. Ten system konfiguracyjny pozwala na wdrażanie konfiguracji za pomocą xcopy wraz z kodem aplikacji i treścią, a także zapewnia łatwy sposób współdzielenia konfiguracji na farmie sieci Web.
- **Nowe narzędzia administracyjne** IIS 7.0 oferuje zbiór narzędzi administracyjnych, które upraszczają zarządzanie infrastrukturą WWW i pozwalają administratorom na delegowanie kontroli administracyjnej witryn i aplikacji do deweloperów i właścicieli treści. IIS 7.0 obejmują nową konsolę zarządzania GUI – IIS Manager, nowe narzędzie wiersza poleceń – Appcmd.exe, nowego dostawcę WMI służącego do automatyzacji zadań administracyjnych, a również nowe zarządzanie API. Wszystkie te narzędzia zapewniają ujednolicone wsparcie zarządzania razem z ustawieniami IIS i ASP.NET. Administratorzy i deweloperzy mogą także używać do dostępu Windows PowerShell za pomocą skryptów informacji konfiguracyjnych dla całej platformy sieci Web.
- **Zintegrowana diagnostyka** IIS 7.0 pozwalają administratorom i deweloperom na minimalizację czasu wyłączenia dzięki nowym możliwościom diagnostyki i rozwiązywania problemów. Internetowe usługi informacyjne eksponują informacje diagnostyczne czasu wykonania, w tym aktualnie wykonywane żądania. IIS 7.0 mogą być także konfigurowane do automatycznego rejestrowania szczegółowych śladów zdarzeń nieudanych żądań do błędnych witryn i aplikacji sieci Web.

Dodatkowe zasoby

Dodatkowe informacje i narzędzia związane z rozdziałem 1 znajdują się w:

- Więcej informacji na temat przetwarzania żądań IIS 7.0 znajduje się w rozdziale 2 „Architektura IIS 7.0”.
- Więcej informacji na temat modularności znajduje się w rozdziale 3 „Podstawy struktury modularnej”.
- Więcej informacji na temat rozszerzalności IIS 7.0 znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web” i w rozdziale 13 „Zarządzanie rozszerzeniami konfiguracji i interfejsem użytkownika”.
- Więcej informacji na temat ujednoliconego, rozproszonego systemu konfiguracji znajduje się w rozdziale 4 „System konfiguracyjny”.
- Więcej informacji na temat narzędzi administracyjnych znajduje się w rozdziale 6 „Korzystanie z konsoli IIS Manager” i w rozdziale 7 „Korzystanie z narzędzi wiersza poleceń”.
- Więcej informacji na temat funkcji rozwiązywania problemów IIS 7.0 i sposobu ich używania znajduje się w rozdziale 16 „Śledzenie i rozwiązywanie problemów” oraz w rozdziale 17 „Dostrajanie wydajności”.

Architektura IIS 7.0

W tym rozdziale:

Przegląd architektury IIS 7.0	28
Rdzenne komponenty IIS 7.0.	31
Przetwarzanie żądań w puli aplikacji	39
Przetwarzanie żądań innych niż HTTP.	49
Podsumowanie.	51
Dodatkowe zasoby	52

W tym rozdziale zajmiemy się architekturą przetwarzania od początku do końca żądań w IIS 7.0. IIS 7.0 zostały całkowicie przeprojektowane w porównaniu z poprzednimi wersjami tego programu. Jak napisaliśmy w poprzednim rozdziale, najważniejszymi innowacjami architektury są:

- **Modularność** Rdzenna funkcjonalność serwera sieci Web IIS 7.0 jest zaimplementowana w postaci ponad 40 wbudowanych natywnych i zarządzanych modułów. Z powodu integracji czasu wykonania IIS i ASP.NET zarówno natywne, jak i zarządzane moduły mogą obsługiwać żądania dowolnego typu. W dalszej części tego rozdziału przyjrzymy się zintegrowanemu potokowi żądań IIS 7.0.
- **Rozszerzalność** IIS 7.0 są w pełni rozszerzalne i zawierają zbiór publicznych interfejsów API pozwalających deweloperom na rozszerzanie ich właściwości i funkcjonalności. Rdzenne moduły serwera sieci Web w IIS 7.0 zostały zbudowane przy użyciu nowych publicznych API, które są dostępne jako natywne Win32 API, a także zarządzane .NET API. W dodatku oprócz rozszerzania serwera sieci Web deweloperzy mogą także rozszerzyć konfigurację IIS i zbudować rozszerzenia dla IIS Manager. Pełna rozszerzalność IIS 7.0 jest dokładnie omówiona w rozdziale 12 „Zarządzanie modułami serwera sieci Web” i w rozdziale 13 „Zarządzanie rozszerzeniami konfiguracji i interfejsu użytkownika”.
- **System konfiguracyjny** W IIS 7 konfiguracja IIS i ASP.NET jest ujednolicona. IIS 7.0 przechowują wszystkie ustawienia IIS i ASP.NET razem w plikach tekstowych opartych na XML, które tworzą rozproszoną hierarchię konfiguracji, nazywaną magazynem konfiguracyjnym. Hierarchia ta zastępuje starszy magazyn konfiguracyjny, metabazę. W dalszej części rozdziału przyjrzymy się sposobowi używania tego magazynu w przetwarzaniu żądań. Co więcej – hierarcha konfiguracyjna IIS 7.0, schemat i ustawienia są opisane szczegółowo w rozdziale 4 „System konfiguracyjny”.

- **Stos administracyjny** IIS 7.0 dostarcza zbioru narzędzi administracyjnych, które razem wspierają zarządzanie ustawieniami IIS i ASP.NET, a także pozwalają na delegację kontroli administracyjnej do użytkowników bez dostępu administracyjnego do serwera. IIS 7.0 obejmuje nową konsolę zarządzania GUI – IIS Manager, nowe narzędzie wiersza polecenia – Appcmd.exe, nowego dostawcę WMI do automatyzacji zadań administracyjnych i nowe zarządzane API. Te narzędzia administracyjne są szczegółowo omówione w rozdziale 6 „Korzystanie z konsoli IIS Manager” i w rozdziale 7 „Korzystanie z narzędzi wiersza polecenia”.
- **Diagnostyka i rozwiązywanie problemów** Możliwości diagnostyki i rozwiązywania problemów w IIS 7.0 pozwalają administratorom i deweloperom na zwiększenie wydajności i minimalizację czasu wyłączenia. Co więcej – ponieważ IIS 7.0 eksponuje model procesu, który nie jest oparty o protokół HTTP aplikacji i usług, możliwości diagnostyczne są także dostępne dla tych aplikacji i usług. W następnej części rozdziału przyjrzymy się przetwarzaniu niezwiązanemu z HTTP w IIS 7.0. Diagnostykę i rozwiązywanie problemów IIS 7.0 opisano prócz tego dokładnie w rozdziale 16 „Śledzenie i rozwiązywanie problemów”.

Aby w pełni zrozumieć implikacje tych i innych zmian architektury, należy omówić, jak IIS 7.0 przetwarza żądania. Treścią tego rozdziału jest architektura przetwarzania żądań w IIS 7.0. Zaczniemy od rdzennych komponentów IIS 7.0 i ich roli w przetwarzaniu żądań HTTP. Następnie przyjrzymy się, jak żądanie jest wykonywane, skupiając się na zintegrowanym potoku przetwarzania żądań i rdzennej modularności serwera sieci Web. W końcu opiszemy przetwarzania żądań niezwiązanych z HTTP w IIS 7.0.

Przegląd architektury IIS 7.0

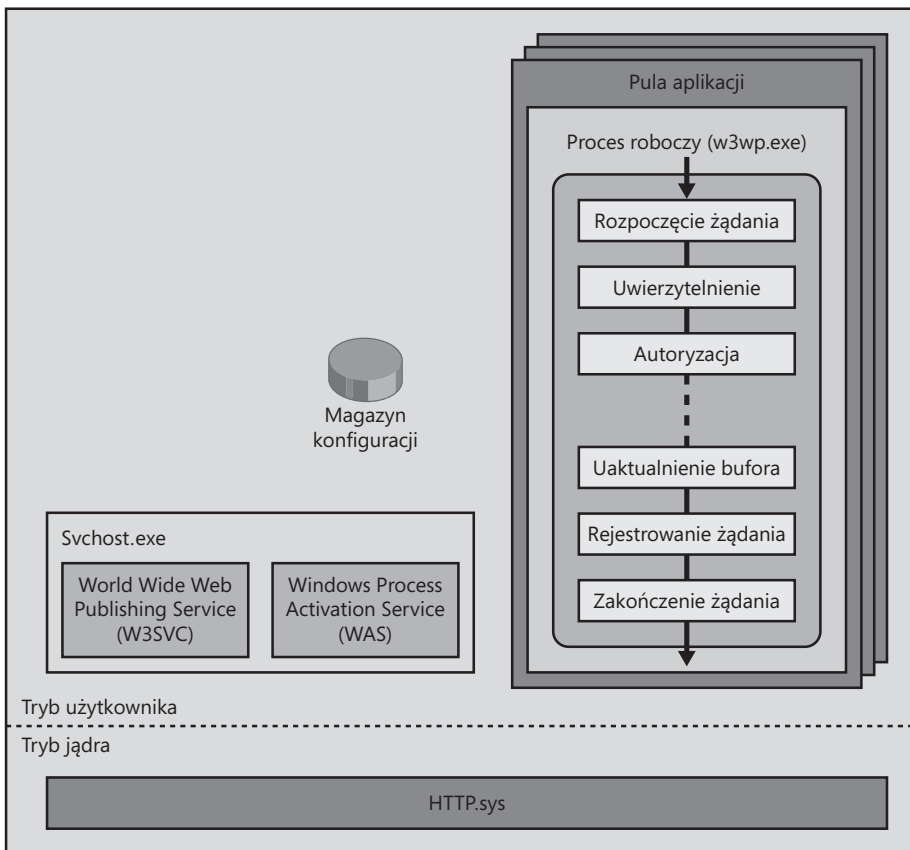
IIS 7.0 zawiera wiele rdzennych komponentów, które pracują razem, aby przetwarzać żądania klienckie HTTP. Każdy komponent ma różne możliwości związane z przetwarzaniem żądań, takie jak nasłuchiwanie na żądania dokonywane do serwera, aktywacje, zarządzanie procesami i wykonywanie żądań. Rysunek 2-1 ukazuje architekturę i rdzenne komponenty IIS 7.0.

Rdzennymi komponentami pokazanymi na rysunku 2-1 są:

- **Stos protokołu HTTP (HTTP.sys)** HTTP.sys to odbiornik (listener) protokołu trybu jądra, który nasłuchuje żądań HTTP i HTTPS.
- **World Wide Web Service Publishing Service (W3SVC)** W3SVC jest adapterem odbiornika (listener – adapter) HTTP. Jest właścicielem komunikacji z HTTP.sys oraz Windows Process Activation Service i dostarcza informacji konfiguracyjnych do HTTP.sys.
- **Windows Process Activation Service (usługa WAS, nazywana też WPAS)** WAS zapewnia zarządzanie procesami roboczymi. Uruchamia, zatrzymuje i czyści pule aplikacji i monitoruje kondycję procesów roboczych podczas ich działania. Dodatkowo pozyskuje informacje konfiguracyjne z magazynu konfiguracyjnego.
- **Magazyn konfiguracyjny** Magazyn konfiguracyjny jest rozproszoną hierarchią bazującą na plikach XML, która przechowuje ustawienia IIS i ASP.NET. Informacje

konfiguracyjne IIS są przechowywane w pliku globalnej konfiguracji IIS application-Host.config zlokalizowanym na szczycie hierarchii. Także pliki konfiguracyjne global.NET Framework, machine.config, a również główny web.config są zlokalizowane na szczycie hierarchii.

- **Proces roboczy w3wp.exe** W3wp.exe jest długoterminowo działającym procesem, który przetwarza żądania i generuje odpowiedzi. Żądania są wykonywane w ramach procesu roboczego. Wiele takich procesów może działać równolegle. Proces roboczy może wykonywać żądanie na jeden z dwóch sposobów – w trybie zintegrowanym z .NET, używając zintegrowanego potoku przetwarzania żądań IIS i ASP.NET, lub w trybie klasycznym, gdzie przetwarzanie żądań IIS i ASP.NET nie jest zintegrowane jak w IIS 6.0. Te tryby są opisane w podrozdziale zatytułowanym "Przetwarzanie żądań w puli aplikacji" w dalszej części rozdziału.



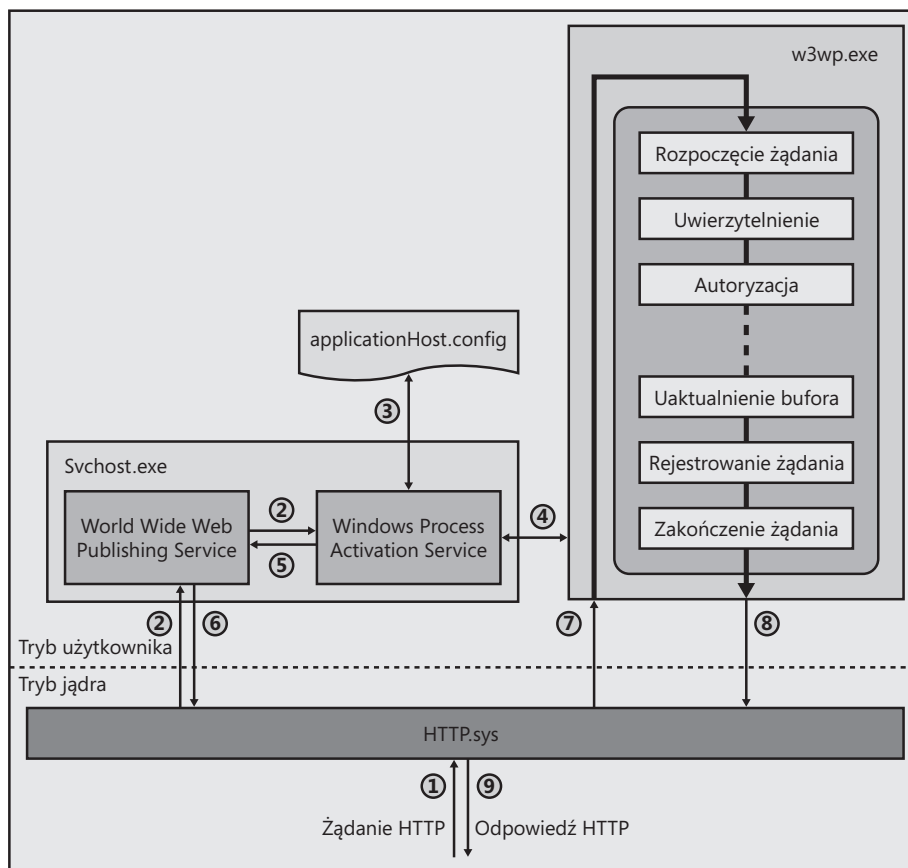
Rysunek 2-1 Architektura IIS 7.0

Rdzenne komponenty IIS 7.0 dokonują głównych funkcji w przetwarzaniu żądań HTTP. Przed przyjrzeniem się rolom, jakie rdzenne komponenty IIS 7.0 odgrywają w przetwarzaniu żądań, trzeba zrozumieć, jak serwer wyznacza, który proces roboczy powinien wykonać konkretne żądanie.

Gdy żądanie HTTP przybywa do serwera z klienta, ścieżka w żądanym URL jest przetwarzana, aby wyznaczyć, której witryny i aplikacji dotyczy dane żądanie. Każda aplikacja działa w puli aplikacji, którą obsługuje jeden lub więcej procesów roboczych

Gdy IIS 7.0 odbiera żądanie dla aplikacji, IIS mapuje żądania do procesu roboczego dla tej puli, do której dana aplikacja należy. Jeżeli jest to pierwsze zadanie dla puli aplikacji, proces roboczy jest uruchamiany, a funkcjonalność serwera ładowana do procesu. Następne żądanie jest przekazywane do procesu roboczego, który wykonuje żądanie, a wynikowe żądanie HTTP zostaje zwrócone do klienta.

Rysunek 2-2 ukazuje przetwarzanie żądania HTTP od początku do końca i interakcję pomiędzy komponentami IIS 7.0.



Rysunek 2-2 Przetwarzanie żądania HTTP w IIS 7.0

W IIS 7.0 przetwarzanie żądań HTTP składa się z następujących kroków, pokazanych na rysunku 2-2:

1. Żądanie HTTP z przeglądarki klienta przybywa do serwera. HTTP.sys przejmuje je.
2. HTTP.sys sprawdza, czy ma informacje konfiguracyjne dla aplikacji, do której żądanie jest przesyłane.

- ❑ jeżeli HTTP.sys ma informacje konfiguracyjne, przekazuje żądanie do odpowiedniego procesu roboczego (patrz krok 7),
 - ❑ jeżeli HTTP.sys nie ma informacji konfiguracyjnych, kontaktuje się z usługą W3SVC, która przekazuje informacje żądania do WAS.
3. WAS pozyskuje informacje konfiguracyjne z globalnego pliku konfiguracyjnego IIS – applicationHost.config.
 4. WAS sprawdza proces roboczy w puli aplikacji, do której kierowane jest żądanie. Jeżeli nie ma procesu roboczego, WAS rozpoczyna go dla tej puli aplikacji.
 5. WAS przekazuje konfigurację, w tym pulę aplikacji i ustawienie konfiguracyjne aplikacji do W3SVC.
 6. W3SVC korzysta z konfiguracji otrzymanej z WAS do konfiguracji i aktualizacji HTTP.sys.
 7. HTTP.sys przekazuje żądanie do procesu roboczego.
 8. Proces roboczy zaczyna potok przetwarzania żądania w celu wykonania żądania. Potok przetwarzania żądania jest uporządkowaną listą komponentów, które dokonują specyficznych zadań do przetwarzania żądania. Na końcu tego przetwarzania odpowiedź zostaje wygenerowana i przekazana do HTTP.sys. Opiszemy potok przetwarzania żądania w podrozdziale zatytułowanym "Przetwarzanie żądań w puli aplikacji" w dalszej części rozdziału.
 9. HTTP.sys wysyła odpowiedź do klienta.

Rdzenne komponenty IIS 7.0

W tym podrozdziale przyjrzymy się rdzennym komponentom IIS 7.0 oraz ich roli w aktywacji procesu i przetwarzaniu żądań.

HTTP.sys

HTTP.sys jest odbiornikiem (listener) protokołu, który nasłuchuje żądań HTTP i HTTPS. HTTP.sys został wprowadzony w IIS 6.0 jako specyficzny dla HTTP odbiornik protokołu dla żądań HTTP. W IIS 7.0 HTTP.sys wspiera także SSL (Secure Sockets Layer), które w IIS 6.0 obsługiwał proces Lsass.exe.

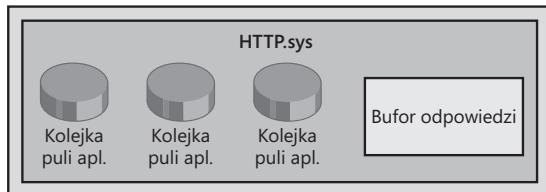
HTTP.sys jest sterownikiem urządzenia w trybie jądra dla stosu protokołu HTTP. Jest częścią podsystemu sieciowego systemów operacyjnych Windows. Zaczynając od IIS 6.0, ten sterownik trybu jądra zastąpił Windows Sockets API (Winsock) będący komponentem trybu użytkownika, którego poprzednie wersje IIS używały do odbierania żądań HTTP i wysyłania odpowiedzi HTTP.

Gdy przeglądarka klienta żąda strony sieci Web z witryny na serwerze IIS 7.0, HTTP.sys przechwytuje żądanie na powiązaniu strony na serwerze i przekazuje je do procesu roboczego w celu przetworzenia. Po przetworzeniu żądania HTTP.sys zwraca odpowiedź do przeglądarki klienta.

Poza przechwytywaniem i zwracaniem żądań HTTP HTTP.sys wykonuje także następujące zadania:

- wstępne przetwarzanie i filtrowanie bezpieczeństwa dla przychodzących żądań HTTP,
- kolejkowanie żądań HTTP dla puli aplikacji,
- buforowanie wychodzących odpowiedzi HTTP.

Rysunek 2-3 ukazuje kolejki żądań HTTP.sys i bufor odpowiedzi.



Rysunek 2-3 Kolejki żądań i bufor odpowiedzi HTTP

Odbiornik HTTP z kolejką żądań i buforem odpowiedzi opartym na jądrze redukuje nakład w kontekście przełączania się do trybu użytkownika i daje w wyniku poprawę wydajności w sposób następujący:

- **Kolejkowanie żądań trybu jądra** Żądania powodują mniejszy nakład na przełączanie kontekstu, ponieważ jądro przekazuje żądania bezpośrednio do właściwego procesu roboczego. Jeśli żaden proces roboczy nie jest dostępny, aby zaakceptować żądanie, kolejka żądań trybu jądra przetrzymuje je do czasu przechwycenia przez proces roboczy.
- **Buforowanie trybu jądra** Żądania dla buforowanych odpowiedzi są obsługiwane bez przełączania się do trybu użytkownika.

HTTP.sys utrzymuje kolejkę żądań dla każdego procesu roboczego. Wysyła odbierane żądania HTTP do kolejki do procesu roboczego, jaki obsługuje pulę aplikacji, w której jest umieszczona żądana aplikacja. Dla każdej aplikacji HTTP.sys utrzymuje tabelę routingu przestrzeni nazw URI z jednym wpisem. Dane tabeli routingu są wykorzystywane do wyznaczenia, która pula aplikacji odpowiada na żądania z danej części przestrzeni nazw. Każda kolejka żądań odpowiada jednej puli aplikacji. Pula aplikacji odpowiada jednej kolejce żądań w HTTP.sys i jednemu lub wielu procesom roboczym.

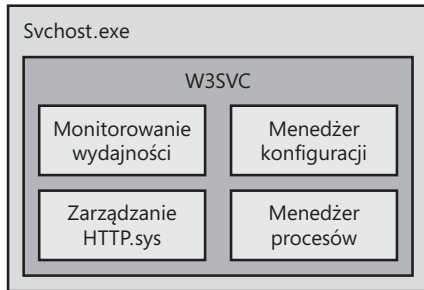
Jeśli błędna aplikacja powoduje usterkę procesu roboczego, usługa nie jest przerywana, a usterka jest nie wykrywana przez użytkownika końcowego, ponieważ jądro kolejkuje żądania, podczas gdy usługa WAS uruchamia nowy proces roboczy dla tej puli aplikacji. Gdy usługa WAS identyfikuje źle działający proces roboczy, uruchamia nowy, jeśli wyróżniające się żądania oczekują na obsłudze. Chociaż czasowe zakłócenie występuje w przetwarzaniu żądań trybu użytkownika, użytkownik nie doznaje usterki, ponieważ połączenia TCP/IP są utrzymywane, a żądania kontynuują kolejkowanie i przetwarzanie. Jedynie te żądania, które działają w procesie roboczym podczas usterki, są widoczne dla użytkowników w postaci statusu błędu. Żądania, które nie były jeszcze przetwarzane, będą kierowane do nowego procesu roboczego.

HTTP.sys nie przetwarza otrzymywanych żądań w inny sposób, niż otrzymując zapamiętaną odpowiedź z bufora wewnętrznego. Dlatego żaden specyficzny dla aplikacji kod nie jest nigdy ładowany w trybie jądra, ale przetwarzany wewnątrz procesu roboczego, który działa w trybie użytkownika. W efekcie błędy w kodzie aplikacji nie wpływają na jądro i nie prowadzą do awarii systemu.

World Wide Web Publishing Service

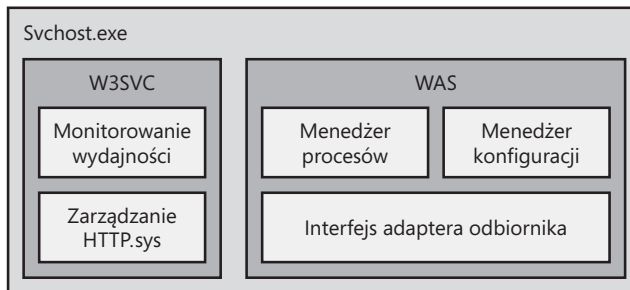
W porównaniu z IIS 6.0 znacznie zmieniono w IIS 7.0 usługę World Wide Web Publishing Service (W3SVC).

W IIS 6.0 usługa W3SVC była odpowiedzialna za zarządzanie HTTP.sys, zarządzanie konfiguracją, procesem i monitorowaniem wydajności, co pokazano na rysunku 2-4.



Rysunek 2-4 W3SVC w IIS 6.0

W IIS 7.0 ta funkcjonalność jest podzielona pomiędzy dwie usługi: W3SVC i nową w IIS 7.0 usługę WAS. Działają one jako LocalSystem w tym samym procesie Svchost.exe i współdzielą te same binaria. W3SVC i WAS w IIS 7.0 pokazano na rysunku 2-5.



Rysunek 2-5 W3SVC i WAS w IIS 7.0

W IIS 7.0 W3SVC działa jako adapter odbiornika (listener – adapter) dla odbiornika HTTP, HTTP.sys. Adaptery odbiorników są komponentami, które ustanawiają komunikację pomiędzy WAS a odbiornikami protokołu. WAS zawiera interfejs adaptera odbiornika, który zapewnia komunikację z adapterami odbiorników.

W3SVC jest odpowiedzialna za konfigurowanie HTTP.sys, uaktualnianie HTTP.sys, gdy konfiguracja się zmienia, i za powiadamianie WAS, gdy żądanie wchodzi do kolejki żądań. Dodatkowo W3SVC kontynuuje kolekcjonowanie liczników dla witryn sieci Web. Jednak, nie odczytuje już informacji konfiguracyjnych z magazynu konfiguracji ani nie zarządza pulami aplikacji czy procesami roboczymi. Zamiast tego odpowiedzialność za odczyt konfiguracji i aktywację oraz zarządzanie procesami została wbudowana w WAS.

Zmiany w funkcjonalności usługi W3SVC pomiędzy IIS 7.0 a IIS 6.0 są podsumowane w poniższej liście:

- Zarządzanie konfiguracją:
 - W IIS 6.0 W3SVC odczytuje informacje konfiguracyjne z magazynu informacji IIS 6.0, czyli z metabazy.
 - W IIS 7.0 W3SVC nie odczytuje już informacji konfiguracyjnych z magazynu konfiguracji. Zamiast tego WAS odczytuje konfigurację z magazynu konfiguracji IIS 7.0, applicationHost.config, a następnie przekazuje ją do W3SVC.
- Zarządzanie HTTP.sys:
 - W IIS 6.0 W3SVC konfiguruje i uaktualnia HTTP.sys, korzystając z informacji konfiguracyjnych odczytanych z metabazy.
 - W IIS 7.0 W3SVC konfiguruje i uaktualnia HTTP.sys, używając informacji konfiguracyjnych pobranych z WAS. Adapter odbiornika dla protokołu HTTP, W3SVC, jest właścicielem komunikacji pomiędzy WAS a HTTP.sys.
- Zarządzanie procesem:
 - W IIS 6.0 W3SVC zarządza pulami aplikacji i procesami roboczymi, w tym uruchamianiem, zatrzymywaniem i wznowianiem procesów roboczych. Dodatkowo W3SVC monitoruje kondycję procesów roboczych i wywołuje szybkie wykrywanie usterek, aby nie zezwolić na uruchamianie nowych procesów, jeśli wiele procesów roboczych ulegnie w określonym czasie awarii.
 - W IIS 7.0 W3SVC nie ma już żadnej odpowiedzialności za zarządzanie procesami roboczymi. Została ona przekazana do WAS.
- Monitorowanie wydajności:
 - W IIS 6.0 W3SVC monitoruje wydajność i dostarcza licznika wydajności dla witryn sieci Web i bufora IIS.
 - W IIS 7.0 W3SVC kontynuuje kolekcjonowanie liczników dla witryn sieci Web.



Uwaga Ponieważ liczniki wydajności pozostały częścią W3SVC, są specyficzne dla HTTP i nie dotyczą WAS.

Windows Process Activation Service

Model aktywacji procesu HTTP został wprowadzony w IIS 6.0 z pulami aplikacji. W IIS 7.0 ta usługa została rozszerzona i jest nazywana Windows Process Activation Service (WAS). Jest usługą odpowiedzialną za pozyskiwanie żądań lub komunikatów przez dowolny protokół i wspiera włączaną aktywację arbitralnych odbiorników protokołów.

W IIS 7.0 WAS zarządza konfiguracją puli aplikacji i procesami roboczymi. W3SVC pełni tę funkcję w IIS 6.0. WAS obejmuje następujące komponenty, ukazane wcześniej na rysunku 2-5:

- menedżer konfiguracji, który odczytuje aplikacje i konfigurację puli aplikacji z magazynu konfiguracji,
- menedżer procesu, który mapuje pule aplikacji do istniejących procesów roboczych i jest odpowiedzialny za uruchamianie nowych instancji W3wp.exe do obsługi nowych puli aplikacji w odpowiedzi na żądania aktywacji,

- interfejs adaptera odbiornika, który definiuje, jak zewnętrzne odbiorniki komunikują otrzymywane żądania aktywacji do WAS. Na przykład usługa W3SVC jest właścicielem komunikacji z HTTP.sys i komunikuje żądania aktywacji HTTP do WAS przez interfejs adaptera odbiornika.

Przy starcie menedżera konfiguracji w WAS odczytuje informacje z magazynu konfiguracji i przekazuje je do adaptera odbiornika HTTP, W3SVC, który jest odpowiedzialny za komunikację z odbiornikiem HTTP, HTTP.sys. Po otrzymaniu przez W3SVC informacji konfiguracyjnych konfiguruje HTTP.sys i przygotowuje do nasłuchiwania żądań.

Menedżer konfiguracji w WAS pozyskuje z magazynu konfiguracji następujące informacje:

- globalne informacje konfiguracyjne,
- informacje konfiguracyjne protokołu,
- konfigurację puli aplikacji, taką jak informacje o koncie procesu,
- konfigurację witryny, taką jak powiązania i aplikacje,
- konfigurację aplikacji, taką jak włączone protokoły i pule aplikacji, do których należy aplikacja.

Jeśli konfiguracja się zmieni, menedżer konfiguracji w WAS odbiera powiadomienia i kolejno aktualizuje W3SVC nowymi informacjami. Po odebraniu przez W3SVC nowej konfiguracji uaktualnia i konfiguruje HTTP.sys. Na przykład, gdy doda się lub usunie pule aplikacji, menedżer konfiguracji przetwarza zmiany konfiguracyjne i komunikuje je usłudze W3SVC, która uaktualnia HTTP.sys, aby dodać lub usunąć kolejkę puli aplikacji.

Menedżer procesu w WAS jest odpowiedzialny za zarządzania procesami roboczymi, co obejmuje rozpoczynanie procesu roboczego i utrzymywanie informacji na temat działania procesów roboczych. Wyznacza także, kiedy uruchomić proces roboczy, kiedy go wznowić i kiedy ponownie uruchomić proces roboczy, gdy zostanie zablokowany i nie może przetwarzać żadnych żądań więcej.

Gdy HTTP.sys pobiera żądanie klienta, menedżer procesu w WAS sprawdza, czy proces roboczy już działa. Jeżeli pula aplikacji już ma proces roboczy obsługujący żądania, wtedy HTTP.sys przekazuje dane żądanie do procesu roboczego, aby je wykonać. Jeśli w puli aplikacji nie istnieje proces roboczy, menedżer procesu uruchamia nowy, aby HTTP.sys mogło przekazywać do niego żądania do przetwarzania.

WAS wspiera inne protokoły prócz HTTP. Ta sama konfiguracja i model procesu używany dla HTTP jest dostępny dla aplikacji i usług nieużywających HTTP. Przyjrzymy się tej możliwości w podrozdziale zatytułowanym „Przetwarzanie żądań innych niż HTTP” w dalszej części rozdziału.

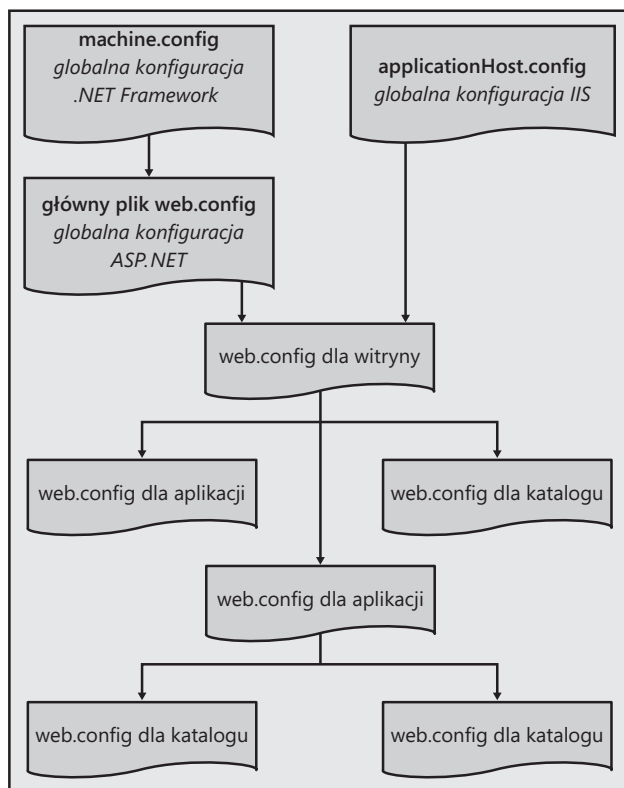
Magazyn konfiguracji

W IIS 6.0 dane konfiguracyjne są przechowywane w opartej na XML metabazie. IIS 7.0 już z niej nie korzysta. Zamiast tego ustawienia konfiguracyjne są przechowywane w rozproszonym, opartym na plikach XML systemie konfiguracyjnym, który łączy ustawienia IIS i ASP.NET.

Rozproszona hierarcha konfiguracyjna obejmuje globalne, dotyczące komputera, pliki konfiguracyjne .NET Framework `machine.config`, główny (root) `web.config`, globalny plik konfiguracyjny IIS `applicationHost.config` i rozproszone pliki konfiguracyjne `web.config` zlokalizowane w witrynach sieci Web, aplikacjach i katalogach, jak pokazano na rysunku 2-6.

Ponieważ pliki konfiguracyjne są przechowywane razem z witryną sieci Web i treścią aplikacji, ten system konfiguracji pozwala na wdrażanie konfiguracji wraz z kodem i treścią aplikacji za pomocą `xcopy`, pozwalając administratorowi serwera na delegację administracji witrynami i aplikacjami do użytkowników bez uprawnień administracyjnych na serwerze i dostarcza także łatwego sposobu współdzielenia konfiguracji na farmie sieci Web.

Globalna, dotycząca całego serwera konfiguracja IIS jest przechowywana w pliku `applicationHost.config` zlokalizowanym w folderze `%SystemRoot%\system32\Inetsrv\Config`. Z tego pliku usługa WAS pozyskuje informacje konfiguracyjne dotyczące puli aplikacji i aplikacji.

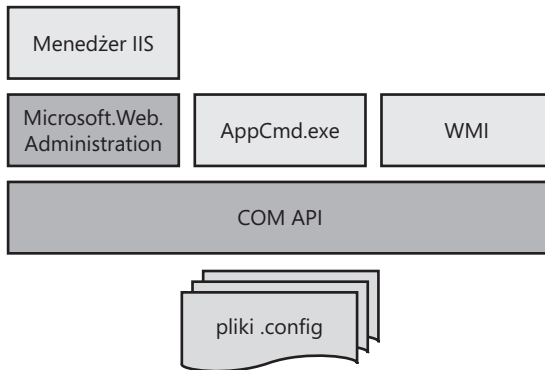


Rysunek 2-6 Rozproszony magazyn informacji IIS 7.0

IIS 7.0 dostarcza wielu narzędzi administracyjnych i interfejsów programowania aplikacji (API), które odczytują i zapisują konfigurację w systemie konfiguracyjnym IIS 7.0. Pliki konfiguracyjne są tekstowymi plikami XML, więc do pracy z plikami konfiguracyjnymi IIS 7.0 można użyć Notatnika, jeśli naprawdę tego chcemy. Jednak nawet dla prostych plików `web.config` jest to działanie podatne na błędy i lepiej go unikać.

W celu uproszczenia zadań administracyjnych IIS 7.0 zapewnia przeprojektowany, oparty na zadaniach, skupiony na właściwościach IIS Manager do zarządzania serwerem sieci Web za pomocą graficznego interfejsu użytkownika (GUI), a także oferuje narzędzie administracyjne wiersza poleceń, Appcmd.exe. W celu dostępu programowego do zarządzania konfiguracją z programów C++ istnieje COM API oraz .NET API, Microsoft.Web.Administration, do programów .NET. Większość właściwości IIS Manager można implementować, używając nowego .NET API. W IIS 7.0 dostarczany jest także dostawca WMI (Windows Management Instrumentation) dla skryptów wraz z przestarzałym dostawcą IIS 6.0 WMI, który jest dostępny do wstecznej kompatybilności z istniejącymi skryptami.

Microsoft.Web.Administration API, Appcmd.exe, i dostawca WMI są nadpisane na szczycie COM API. Ten nowy stos administracyjny pokazano na rysunku 2-7.



Rysunek 2-7 Stos administracyjny IIS 7.0



Uwaga Więcej informacji na temat systemu konfiguracyjnego IIS 7.0 i globalnego pliku applicationHost.config znajduje się w rozdziale 4 „System konfiguracyjny”.

Starszy magazyn konfiguracji, metabaza, nie jest częścią IIS 7.0. Jednak w celu kompatybilności wstecznej IIS 7.0 dostarcza opcjonalną właściwość Metabase Compatibility, która instaluje usługę IIS Administration Service (IISADMIN), odczytującą i zapisującą metabazę w IIS 6.0. Także właściwość Metabase Compatibility instaluje proces Inetinfo.exe, który zawiera usługę IISADMIN. Te dwa komponenty dostarczają warstwę translacji, nazywaną Admin Base Objects (ABO) Mapper. ABO Mapper wspiera przestarzałe interfejsy ABO API do pracy z metabazą, ale przechowuje konfigurację bezpośrednio w plikach konfiguracyjnych IIS 7.0. Jeżeli nie zainstaluje się właściwości Metabase Compatibility, IIS 7.0 nie użyje usługi IISADMIN ani procesu Inetinfo.exe.

Proces roboczy

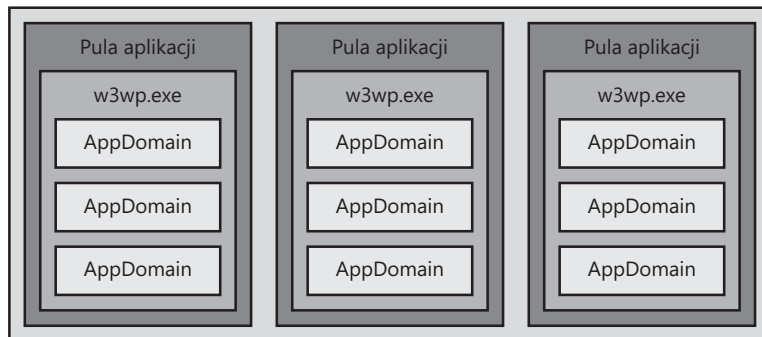
Rolą procesu roboczego jest przetwarzanie żądań. Proces roboczy jest samo zawierającym się, długotrwałym procesem, który działa jako program o nazwie w3wp.exe, w trybie użytkownika.

Każdy proces roboczy zapewnia rdzenną funkcjonalność serwera sieci Web. Odpowiedź jest generowana, a potem zwracana do klienta jako wynik przetwarzania żądań

w procesie roboczym. Każdy proces roboczy używa HTTP.sys do odbierania żądań i wysyłania odpowiedzi.

Jeden proces roboczy obsługuje jedną pulę aplikacji, która grupuje razem jedną lub więcej aplikacji. Z tego powodu można zastosować specyficzne ustawienia konfiguracyjne do grup aplikacji i procesów roboczych, które je obsługują.

Każda aplikacja działa w puli aplikacji. Pula aplikacji może być obsługiwana przez zbiór procesów roboczych. Proces roboczy może obsługiwać tylko jedną pulę. Wiele procesów roboczych, które obsługują różne pule aplikacji, może działać równolegle, co pokazano na rysunku 2-8.



Rysunek 2-8 Izolacja procesu i aplikacji

Granice procesów roboczych oddzielają pule aplikacji, co pozwala kodowi danej aplikacji na działanie w izolowanym środowisku, a z tego powodu poszczególne aplikacje mogą wykonywać się w samo zawierającym się procesie roboczym. Model izolacji procesów został najpierw wprowadzony w IIS 6.0. Zapobiega wpływowi aplikacji działających w jednej puli na aplikacje w innej puli na serwerze, dostarczając piaskownic dla aplikacji.

W procesie roboczym domeny aplikacji dostarczają granic dla aplikacji .NET. Każda aplikacja .NET działa we własnej domenie aplikacji (AppDomain), jak ukazano na rysunku 2-8. Domena aplikacji ładuje kod aplikacji, gdy dana aplikacja uruchamia się. Wirtualne katalogi w ramach aplikacji są obsługiwane przez tę samą AppDomain co aplikacja, do której należą.

Żądania wewnątrz każdego procesu roboczego mogą być wykonywane na jeden z dwóch sposobów: w trybie zintegrowanym z .NET, gdy zarówno żądania IIS, jak i ASP.NET korzystają z tego samego zintegrowanego potoku przetwarzania żądań, lub w trybie klasycznym, gdzie dwa oddzielne potoki służą do przetwarzania IIS i ASP.NET. Można konfigurować pulę aplikacji, aby wyznaczyć, w którym z dwóch trybów należy wykonywać żądania ASP.NET. W następnym podrozdziale skupimy się na architekturze przetwarzania żądań w procesach roboczych obsługujących pule aplikacji.

Ze źródła: Izolacja puli aplikacji w IIS 7.0

Projekt puli aplikacji wprowadzony w IIS 6.0 stał się podstawą włączenia większej izolacji bezpieczeństwa dla wielu aplikacji, a także zwiększenia tolerancji błędów serwera sieci Web. IIS 7.0 kontynuuje wykorzystanie tej koncepcji i już nie wspierają starszego, pochodzącego z IIS 5.0 modelu izolacji. Większość właściwości, które spowodowały sukces puli aplikacji w IIS 6.0 pozostała, w tym zdolność uruchamiania każdej puli aplikacji z różnymi poświadczeniami oraz konfigurowanie inteligentnego monitorowania kondycji i parametrów odnawiania w celu utrzymania niezawodności aplikacji.

IIS 7.0 poszły jeszcze dalej, dostarczając automatycznej izolacji puli aplikacji przez automatycznie generowane identyfikatory SID (Security Identifier) dla puli aplikacji i przez automatyczne izolowanie konfiguracji poziomu serwera, tak że może być ona odczytana tylko przez pulę aplikacji, na którą wpływa. Sprawia to, że konfigurowanie w pełni izolowanych aplikacji sieci Web przez wykorzystanie puli aplikacji jest prostsze niż kiedykolwiek.

Dodatkowo IIS 7.0 w Windows Server 2008 dostarczają wielu usprawnień wydajności znacznie zwiększających liczbę puli aplikacji, które mogą być konfigurowane i aktywne na jednym serwerze sieci Web, zarówno przez mniejsze zużycie zasobów przez proces roboczy, jak i inteligentne właściwości zarządzania tym procesem. Te usprawnienia sprawiają, że dla dostawców hostingu sieci Web łatwiejsze staje się umieszczanie każdej indywidualnej aplikacji w oddzielnej puli w celu osiągnięcia maksymalnego rozdzielenia bezpieczeństwa i błędów dla tych aplikacji.

Podczas projektowania infrastruktury aplikacji sieci Web trzeba upewnić się, czy w pełni wykorzystuje się możliwości dostarczane przez izolację puli aplikacji.

Mike Volodarsky, IIS Core Server Program Manager

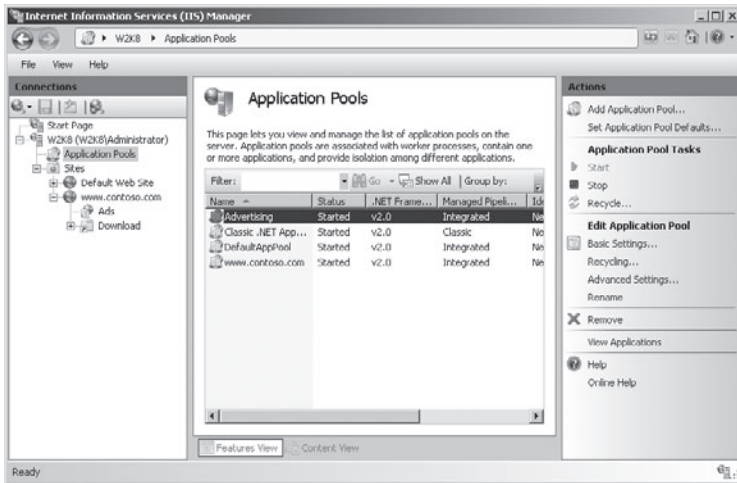
Przetwarzanie żądań w puli aplikacji

W IIS 7.0 dostępne są dwa tryby puli aplikacji: zintegrowany i klasyczny. Gdy skonfiguruje się pulę aplikacji w trybie zintegrowanym, IIS 7.0 przetwarza żądania ASP.NET przy pomocy zintegrowanego potoku przetwarzania żądań IIS i ASP.NET. Jeśli konfiguruje się pulę aplikacji w trybie klasycznym, IIS 7.0 przetwarza żądania ASP.NET, korzystając z oddzielnych potoków przetwarzania żądań IIS i ASP.NET, jak było w IIS 6.0.

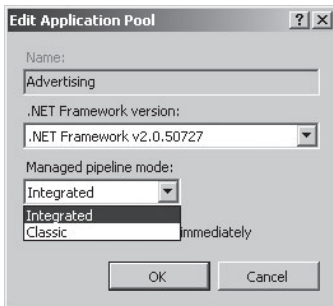
Pule aplikacji skonfigurowane z różnymi trybami mogą działać na tej samej maszynie serwera. Można określić tryb dla danej puli, konfigurując ustawienie Managed Pipeline Mode (Zarządzany tryb potokowy) w IIS Manager.

Aby skonfigurować tryb przetwarzania ASP.NET dla puli aplikacji za pomocą IIS Manager, należy:

1. Rozwinąć węzeł serwera i wybrać węzeł Application Pools w panelu Connections konsoli IIS Manager.
2. Wybrać ze strony Application Pools (Pule aplikacji) pulę aplikacji, którą chcemy konfigurować.



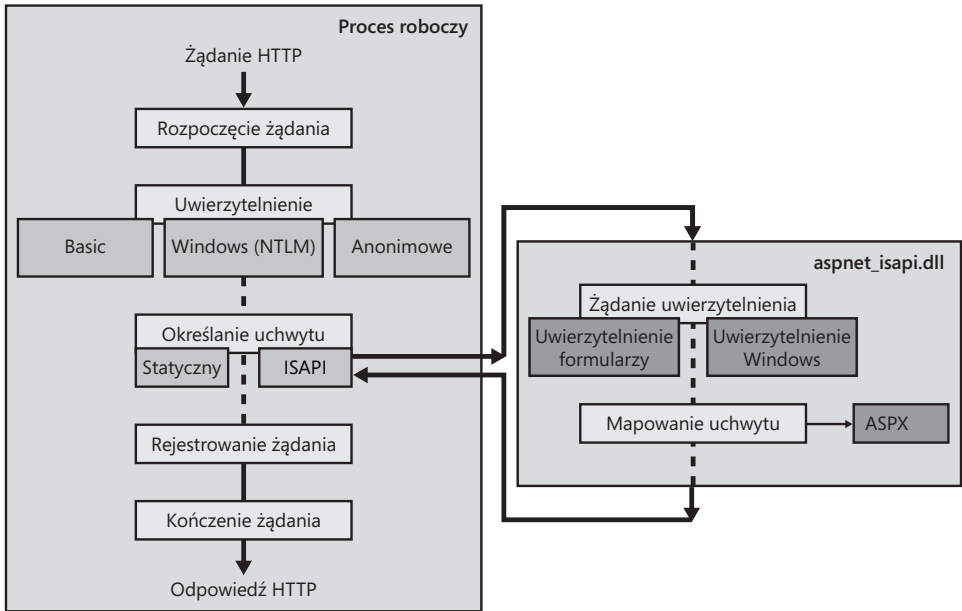
- Wybrać Basic Settings (Ustawienia podstawowe) z panelu Actions (Akcje) pod Edit Application Pool (Edytuj pulę aplikacji).
- Wybrać pożądany tryb (Integrated lub Classic) w oknie dialogowym Edit Application Pool (Edytowanie puli aplikacji) na liście rozwijanej Managed Pipeline Mode i kliknąć OK.



Tryb klasyczny

Tryb klasyczny w IIS 7.0 zapewnia kompatybilność wsteczną z IIS 6.0. Gdy pula aplikacji jest w trybie klasycznym, IIS 7.0 przetwarza żądania ASP.NET, korzystając z dwóch oddzielnych potoków przetwarzania żądań IIS i ASP.NET, tak jak było w IIS 6.0. W celu zrozumienia trybu klasycznego w IIS 7.0 przyjrzyjmy się najpierw, jak żądania ASP.NET są przetwarzane w IIS 6.0.

Rysunek 2-9 ukazuje przetwarzanie żądań ASP.NET w IIS 6.0. W IIS do wersji 6.0 ASP.NET łączy się z serwerem sieci Web jako samodzielne środowisko aplikacji. W tych wersjach ASP.NET jest implementowane jako rozszerzenie IIS ISAPI (Internet Server Application Programming Interface).



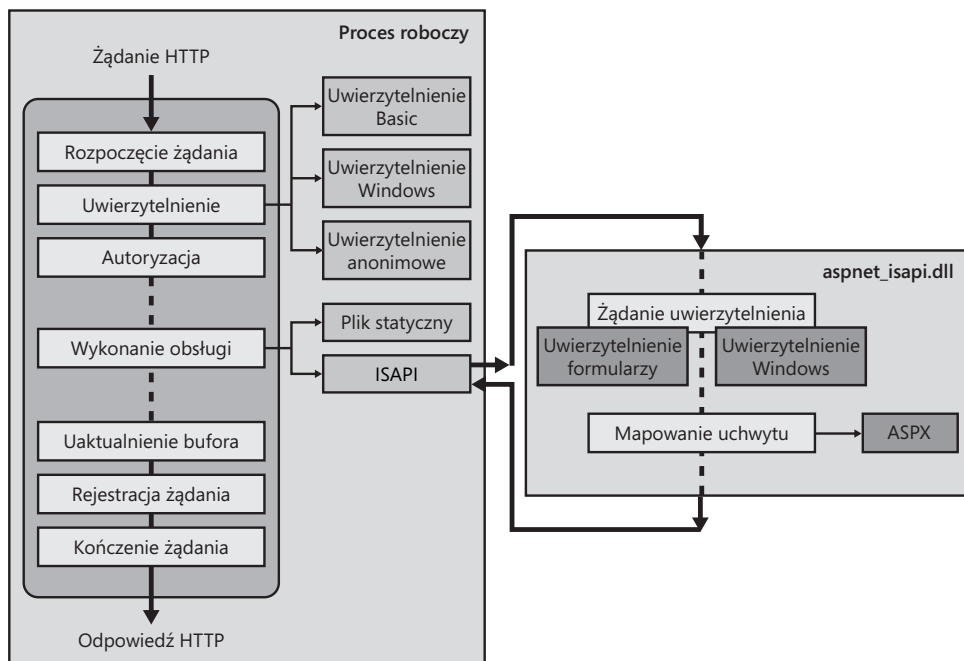
Rysunek 2-9 Przetwarzanie żądań ASP.NET w IIS 6.0

W IIS 6.0 rozszerzenie ASP.NET ISAPI (`aspnet_isapi.dll`) jest odpowiedzialne za przetwarzane typów treści, które są w nim zarejestrowane, takich jak ASPX i ASMX. Oferuje potężne właściwości dla tych żądań, na przykład uwierzytelnianie Forms i buforowanie wyników odpowiedzi. Jednak tylko typy treści zarejestrowane w ASP.NET mogą zyskać na tych usługach. Inne typy treści – w tym strony ASP, pliki statyczne, obrazy i aplikacje CGI (Common Gateway Interface) – nie mają w ogóle dostępu do tych właściwości.

Żądanie do typu treści ASP.NET jest najpierw przetwarzane przez IIS, a następnie przekazywane do `aspnet_isapi.dll`, które utrzymuje aplikację ASP.NET i model przetwarzania żądania, co efektywnie ekspozuje dwa oddzielne potoki serwera, jeden dla natywnych filtrów ISAPI i komponentów rozszerzeń, a drugi dla zarządzanych komponentów aplikacji. Komponenty ASP.NET wykonują się w całości w rozszerzeniu ASP.NET ISAPI i tylko dla żądań mapowanych do ASP.NET w mapie konfiguracji skryptów IIS. Żądania do innej niż ASP.NET treści, takiej jak strony ASP lub pliki statyczne, są przetwarzane przez IIS lub inne rozszerzenia ISAPI i nie są widoczne dla ASP.NET.

W dodatku – nawet dla zasobów ASP.NET – pewne funkcjonalności nie są dostępne z powodu ograniczeń czasu wykonania. Na przykład nie jest możliwe modyfikowanie zbioru wychodzących nagłówków odpowiedzi HTTP, zanim zostaną wysłane do klienta, ponieważ odbywa się to na ścieżce wykonania ASP.NET.

W IIS 7.0 w trybie klasycznym żądania ASP.NET są także przetwarzane przy użyciu rozszerzenia ASP.NET ISAPI, jak pokazano na rysunku 2-10. Rdzeń serwera sieci Web w IIS 7.0 jest w pełni podzielony na komponenty, natomiast w IIS 6.0 jest monolityczny. Jednak żądania ASP.NET w trybie klasycznym są przetwarzane przez `asnet_isapi.dll` w taki sam sposób, jak było w IIS 6.0. Po przetworzeniu żądania w `asnet_isapi.dll` jest ono kierowane z powrotem przez IIS w celu wysłania odpowiedzi.



Rysunek 2-10 Przetwarzanie żądań ASP.NET w trybie klasycznym w IIS 7.0

Tryb klasyczny w IIS 7.0 ma te same główne ograniczenia, jakie istniały w przetwarzaniu ASP.NET w IIS 6.0. Są to:

- usługi dostarczane przez moduły ASP.NET nie są dostępne do żądań innych niż ASP.NET,
- niektóre kroki przetwarzania – na przykład uwierzytelnianie – są duplikowane,
- niektóre ustawienia muszą być zarządzane w dwóch lokalizacjach, takich jak uwierzytelnianie, śledzenie i buforowanie wyjścia,
- aplikacje ASP.NET nie są w stanie wpływać na pewne części przetwarzania żądań IIS, które występują przed i po ścieżce wykonania ASP.NET, z powodu umieszczenia rozszerzenia ASP.NET ISAPI w potoku serwera.

Tryb klasyczny jest dostarczany tylko w celu kompatybilności wstecznej z IIS 6.0. Mówiąc w uproszczeniu, powinno się dodać aplikację do puli aplikacji w trybie klasycznym tylko wtedy, gdy aplikacja nie działa w trybie zintegrowanym.



Uwaga Więcej informacji na temat kompatybilności aplikacji w IIS 7.0 znajduje się w rozdziale 11 „Utrzymywanie środowisk rozwijania aplikacji”.

Tryb zintegrowany z .NET

Jeżeli pula aplikacji jest konfigurowana w trybie zintegrowanym z .NET, można skorzystać z zalety zintegrowanej architektury przetwarzania żądań IIS 7.0 i ASP.NET.

W IIS 7.0 czas wykonania ASP.NET jest zintegrowany z rdzeniem serwera sieci Web. Potoki żądań IIS i ASP.NET są łączone, zapewniając zunifikowany (czyli zintegrowany) potok przetwarzania żądań, który jest eksponowany dla natywnych i zarządzanych modułów.

Potok przetwarzania żądań IIS 7.0 jest implementowany przez rdzenny mechanizm serwera sieci Web. Pozwala to wielu niezależnym modułom dostarczać usługi dla tego samego żądania. Wszystkie właściwości serwera sieci Web są implementowane jako samodzielne moduły. Istnieje ponad 40 oddzielnych modułów natywnych i zarządzanych. Każdy z nich implementuje konkretną właściwość lub funkcjonalność serwera sieci Web, taką jak rejestrowanie lub buforowanie wyjścia.



Uwaga Pełna lista wbudowanych modułów IIS 7.0 – zarówno natywnych, jak i zarządzanych – znajduje się w dodatku C „Lista modułów IIS 7.0”.

Moduły natywne są implementowane jako biblioteki DLL (dynamic-link library) oparte na publicznych interfejsach (API) rozszerzeń IIS 7.0 C++. Moduły zarządzane są implementowane jako zarządzane klasy .NET Framework oparte na modelu integracji ASP.NET w IIS 7.0 (IIS 7.0 mają zintegrowany istniejący interfejs IHttpModule API dla ASP.NET). Oba te interfejsy API pozwalają modułom na uczestniczenie w potoku przetwarzania IIS 7.0 i na dostęp do wszystkich zdarzeń dla wszystkich żądań.

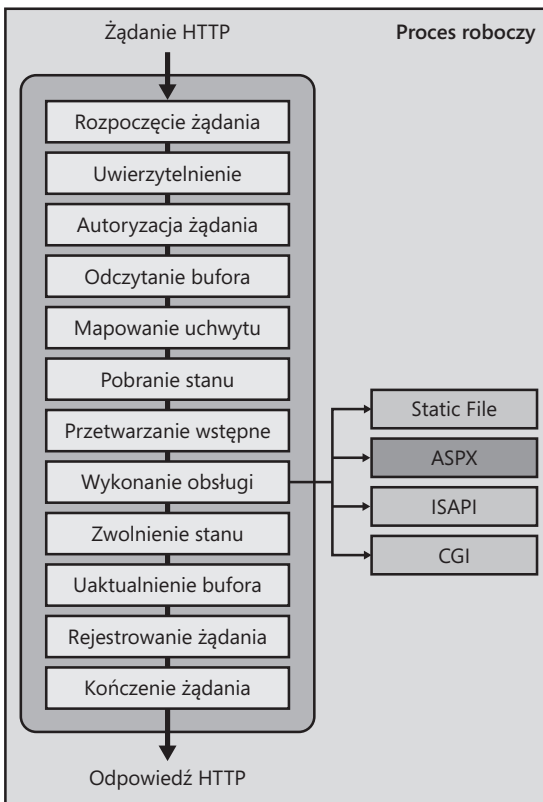
Zintegrowany potok przetwarzania żądań IIS 7.0 pokazano na rysunku 2-11. Potok jest uporządkowaną listą składającą się z natywnych i zarządzanych modułów, które wykonują specyficzne zadania w odpowiedzi na żądania. Gdy proces roboczy w puli aplikacji odbiera żądanie z HTTP.sys, jest ono przekazywane przez uporządkowaną listę faz. Odpowiedź jest generowana i wysyłana z powrotem do HTTP.sys jako wynik przetwarzania.

Każda faza w potoku powoduje zdarzenie. Natywne i zarządzane moduły przeprowadzają subskrypcję do zdarzeń w fazach potoku, które są dla nich odpowiednie. Gdy zdarzenie jest wzbudzone, moduły natywne i zarządzane zasubskrybowane do tego zdarzenia są powiadamiane i wykonują pracę w celu przetwarzania żądania. Model zdarzeń potoku pozwala wielu modułom na wykonywane żądania podczas przetwarzania.

Większość zdarzeń potoków jest przeznaczonych dla specyficznego typu zadania, takiego jak uwierzytelnianie, autoryzacja, buforowanie i rejestrowanie. Podana niżej lista opisuje fazy i odpowiadające im zdarzenia w potoku przetwarzania żądań:

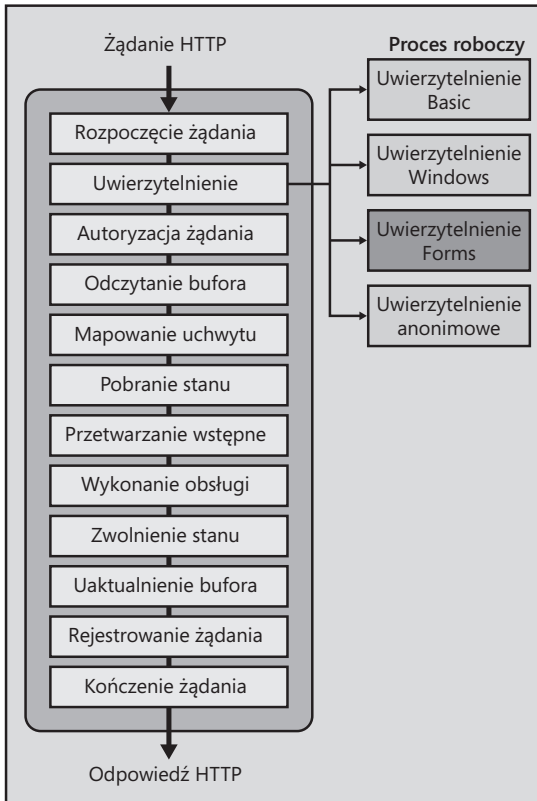
- **Faza rozpoczęcia żądania** Rozpoczyna przetwarzanie. Powstaje zdarzenie *BeginRequest*.
- **Faza uwierzytelnienia** Uwierzytelnia użytkownika żądania. Powstaje zdarzenie *AuthenticateRequest*.
- **Faza autoryzacji żądania** Powstaje tu zdarzenie *AuthorizeRequest*. Ta faza sprawdza dostęp do żądanych zasobów dla uwierzytelnionego użytkownika. Jeżeli dostęp jest zabroniony, żądanie jest odrzucane.
- **Faza sprawdzenia bufora** W tej fazie powstaje zdarzenie *ResolveRequestCache*. W tym czasie sprawdza się, czy odpowiedź na żądanie może być pozyskana z bufora.

- **Faza mapowania uchwytu** Powstaje zdarzenie *MapRequestHandler*. Ta faza wyznacza uchwyt do żądania.
- **Faza odczytania stanu** W tej fazie powstaje zdarzenie *AcquireRequestState*. Ta faza pozyskuje wymagany stan dla żądania.
- **Faza przetwarzania wstępnego** W tej fazie powstaje zdarzenie *PreExecuteRequestHandler*. Sygnalizuje, że mechanizm obsługi ma być wykonany i wykonuje czynności przetwarzania wstępnego, jeśli jest to potrzebne.
- **Faza wykonania obsługi** Powstaje zdarzenie *ExecuteRequestHandler*. Program obsługi wykonuje i generuje odpowiedź.
- **Faza zwolnienia stanu** W tej fazie powstaje zdarzenie *ReleaseRequestState*, które zwalnia stan żądania.
- **Faza aktualizacji bufora** Ta faza uaktualnia bufor. Powstaje zdarzenie *UpdateRequestCache*.
- **Faza rejestrowania** Żądanie jest rejestrowane. Powstaje zdarzenie *LogRequest*.
- **Faza końcowa** W tej fazie powstaje zdarzenie *EndRequest*, które sygnalizuje, że przetwarzanie żądania jest na ukończeniu.



Rysunek 2-11 Potok zintegrowanego przetwarzania IIS 7.0

Moduły, które subskrybują zdarzenia, dostarczają specyficznych usług, odpowiednich dla właściwej fazy w potoku. Na przykład rysunek 2-12 ukazuje wiele natywnych i zarządzanych modułów, które subskrybują do zdarzenia *AuthenticateRequest* w fazie *AuthenticateRequest*, takie jak moduły uwierzytelniania Basic, Windows, ASP.NET Forms i Anonymous. Moduły uwierzytelniania Basic, Windows i Anonymous są modułami natywnymi, natomiast moduł uwierzytelniania Forms jest modulem zarządzanym.



Rysunek 2-12 Natywne i zarządzane moduły w zintegrowanym potoku przetwarzania

Zintegrowany potok .NET ma wiele znaczących zalet nad poprzednimi wersjami IIS. Oto one:

- Pozwolenie usługom dostarczanych przez moduły natywne i zarządzane na stosowanie do wszystkich żądań. Wszystkie typy plików mogą używać funkcjonalności, które w IIS 6.0 są dostępne tylko dla kodu zarządzanego. Na przykład można teraz użyć uwierzytelniania ASP.NET Forms i autoryzacji URL (Uniform Resource Locator) dla plików statycznych, plików ASP, CGI i wszystkich innych typów plików w witrynach i aplikacjach.
- Eliminacja duplikowania wielu funkcjonalności w IIS i ASP.NET. Na przykład, gdy klient żąda pliku zarządzanego, serwer wywołuje odpowiedź IIS, to samo żądanie przechodzi przez proces uwierzytelniania w potoku IIS i w ASP.NET. Inne zuniifikowane funkcjonalności IIS i ASP.NET obejmują autoryzację URL, śledzenie, dowolne błędy i buforowanie wyjścia.

- Zarządzanie wszystkimi modułami w jednej lokalizacji, upraszczające administrację witryną i aplikacją na serwerze.
Zamiast zarządzania pewnymi właściwościami w konfiguracji IIS, a pewnymi w konfiguracji ASP.NET, istnieje jedno miejsce do implementacji, konfiguracji, monitorowania i wspierania właściwości serwera. Na przykład z powodu integracji czasu wykonania IIS i ASP.NET mogą używać tej samej konfiguracji do włączania i określania kolejności modułów serwera, a także konfigurowania mapowania programów obsługi.
- Rozszerzanie IIS zarządzanymi modułami ASP.NET.
IIS 7.0 pozwalają modułom ASP.NET na włączanie się bezpośrednio do potoku serwera, w ten sam sposób, jak modułom napisanym za pomocą natywnego C++ IIS API. Moduły ASP.NET mogą wykonywać się we wszystkich fazach czasu wykonania w potoku przetwarzania żądania, a przy tym mogą być wykonywane w dowolnej kolejności z uwzględnieniem modułów natywnych. Także ASP.NET API zostały rozszerzone, aby pozwolić na większą kontrolę nad przetwarzaniem żądań, niż było poprzednio.



Uwaga Więcej informacji na temat rozszerzania IIS 7.0 znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

Sposób implementacji integracji ASP.NET

Chociaż moduły natywne i zarządzane implementują tę samą koncepcję logiczną, używają dwóch różnych interfejsów API. IIS 7.0 zapewniają specjalny moduł natywny, nazywany Managed Engine, aby umożliwić model zintegrowanego potoku dla modułów natywnych i zarządzanych. Zapewnia on integracyjne opakowania dla modułów ASP.NET, które pozwalają modułom zarządzanym działać tak, jakby były modułami natywnymi i programami obsługi IIS. Działają jako proxy dla powiadomień zdarzeń i propagują wymagany stan żądania do modułów zarządzanych. Wraz z mechanizmem ASP.NET usługi IIS 7.0 stanowią zintegrowany potok i są odpowiedzialne także za czytanie modułów zarządzanych i za konfigurację programów obsługi.

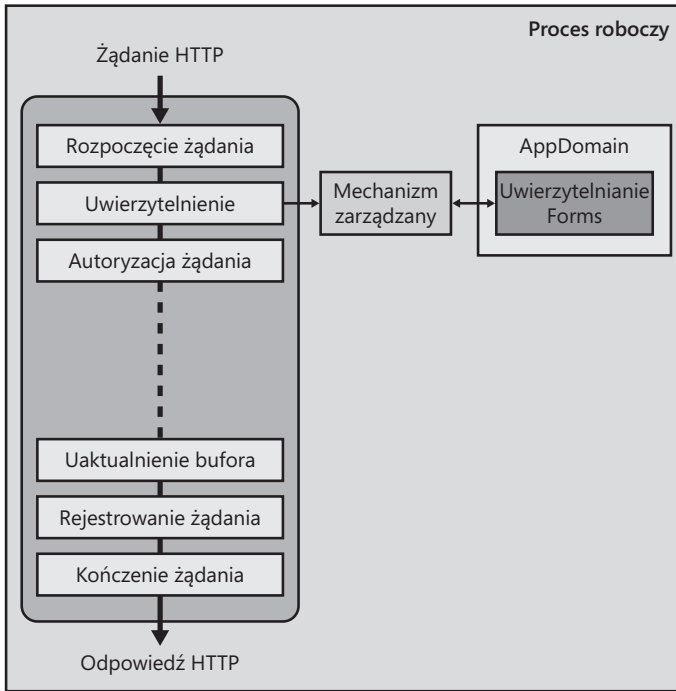
Gdy żądanie wymaga modułu zarządzanego, moduł Managed Engine tworzy AppDomain, gdzie wspomniany moduł zarządzany może wykonywać konieczne przetwarzanie, takie jak uwierzytelnianie użytkownika za pomocą uwierzytelniania Forms. Rysunek 2-13 ukazuje moduł Managed Engine w zarządzanym modulem uwierzytelnianiu Forms, wykonywanym wewnątrz AppDomain.

Wszystkie moduły zarządzane są zależne od modułu Managed Engine i nie mogą być wykonywane bez niego. Moduł Managed Engine musi być zainstalowany i włączony w IIS 7.0, aby działał integrowany potok i aplikacje ASP.NET.

W Windows Server 2008 moduł Managed Engine jest instalowany jako część komponentu Role Service i .NET Extensibility Component, natomiast w Windows Vista jako część komponentu .NET Extensibility.



Uwaga Więcej informacji na temat integracji ASP.NET znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.



Rysunek 2-13 Moduł Managed Engine

Zakres modułu

Moduły mogą być instalowane i włączane na różnych poziomach. Moduły, które są włączane na poziomie serwera, dostarczają domyślnego zbioru właściwości dla wszystkich aplikacji na serwerze. Globalny magazyn konfiguracji IIS – `applicationHost.config` – dostarcza ujednoliconej listy modułów natywnych i zarządzanych. Za każdym razem, gdy WAS aktywuje proces roboczy, pobiera konfigurację z magazynu konfiguracji i proces roboczy ładuje wszystkie globalnie wymienione moduły.

Natywne moduły mogą być instalowane tylko na poziomie serwera, natomiast nie na poziomie aplikacji. Na poziomie aplikacji globalne moduły natywne, które są włączone na poziomie serwera, mogą być usunięte, a te, które są zainstalowane, ale nie włączone globalnie, mogą pozostać włączone dla danej aplikacji.

Zarządzane moduły mogą być dodawane na poziomie serwera, witryny i aplikacji. Specyficzne dla aplikacji moduły są ładowane po pierwszym żądaniu do aplikacji. Moduły zarządzane aplikacji mogą być wdrażane za pomocą `xcopy` razem z innymi plikami aplikacji.

Można zarządzać natywnymi i zarządzanymi modułami za pomocą właściwości `Modules` (Moduły) w IIS Manager.



Uwaga Więcej informacji na temat zarządzania modułami znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

Określanie kolejności modułu

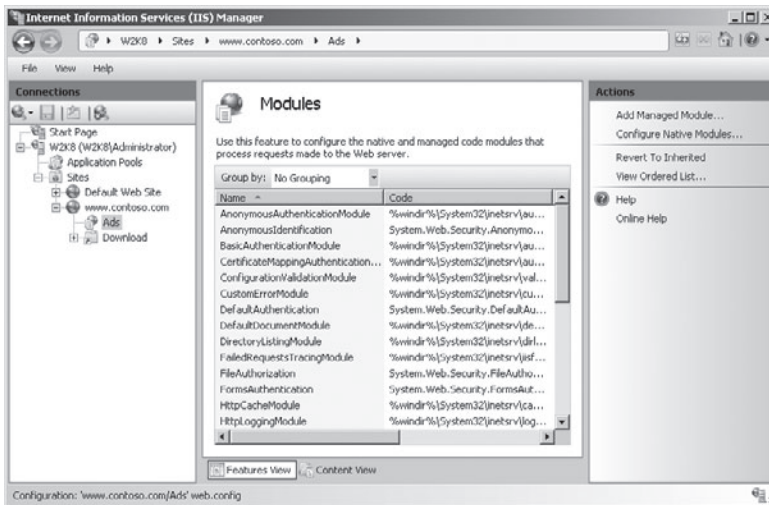
Model potoku zapewnia, że typowe zadania przetwarzania serwera sieci Web są wykonywane w prawidłowej kolejności. Na przykład uwierzytelnienie musi odbywać się przed autoryzacją: uwierzytelnienie użytkownika powiązane z żądaniem w fazie Authenticate Request musi się odbyć przed sprawdzeniem dostępu użytkownika do żadanego zasobu w fazie Authorize Request.

Serwer korzysta z sekwencji listy modułów w sekcji konfiguracyjnej <modules>, aby określić kolejność wykonywania modułów wewnątrz każdej fazy przetwarzania żądania. Większość modułów automatycznie unika problemów kolejności, ponieważ jest wykonywana podczas odpowiedniej fazy. Jednak wiele modułów, które są wykonywane w tej samej fazie, może mieć zależności dotyczące kolejności. Na przykład wbudowane moduły uwierzytelniania działające w fazie Authenticate Request powinny być wykonywane w kolejności od najsilniejszego do najsłabszego, aby żądanie było uwierzytelnione z najsilniejszymi dostępnymi poświadczeniami.

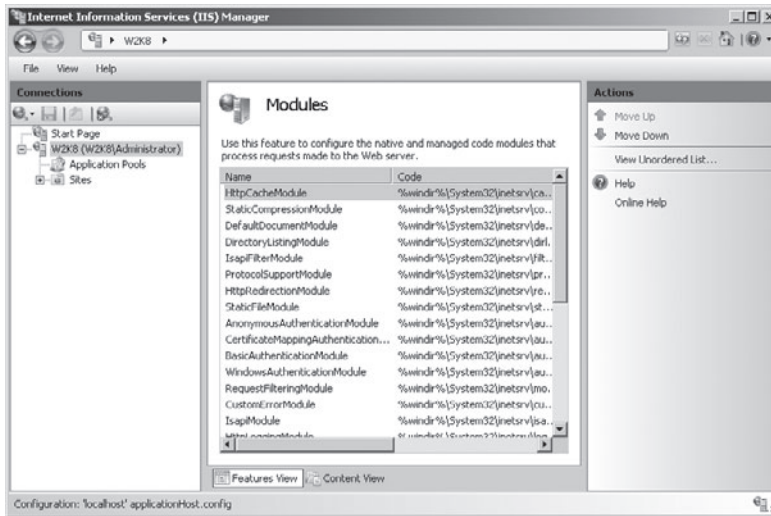
Administrator może kontrolować kolejność modułów, zmieniając kolejność, w jakiej są wymienione w sekcji <modules>, aby zarządzać zależnościami tej kolejności. Może być to dokonane na przykład przy użyciu właściwości Modules w IIS Manager.

Aby wyświetlić i opcjonalnie zmienić uporządkowaną listę modułów na serwerze, należy:

1. Wybrać węzeł serwera z panelu Connections konsoli IIS Manager.
2. Otworzyć właściwość Modules na stronie domowej serwera.



3. Kliknąć View Ordered List (Wyświetl listę uporządkowaną) w panelu Actions.



4. Można zmienić położenie modułu w kolejności przetwarzania, wybierając moduł i używając opcji Move Up (Przenieś w górę) i Move Down (Przenieś w dół) w panelu Action, aby przesunąć go na pożądaną pozycję na liście.



Uwaga Informacje na temat domyślnej kolejności wbudowanych modułów znajdują się w dodatku D „Sekwencja modułów”.

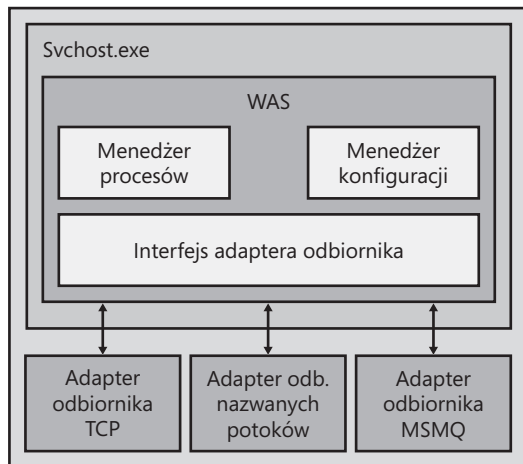
Przetwarzanie żądań innych niż HTTP

W IIS 7.0 WAS wspiera protokoły inne niż HTTP, pozwalając na używanie IIS do udostępniania nieopartych na HTTP aplikacji i usług. Model procesu WAS uogólnia model procesu dla serwera HTTP poprzez usunięcie zależności od HTTP. Ponieważ WAS zarządza konfiguracją puli aplikacji i procesów roboczych w IIS 7.0, ta sama konfiguracja i model procesu, który jest używany dla HTTP, może być użyty dla niezwiązanych z HTTP aplikacji. Wszystkie właściwości zarządzania procesem IIS, takie jak aktywacja na żądanie, monitorowanie kondycji procesu, zarządzalność najwyższej klasy i natychmiastowa ochrona przed usterkami są dostępne dla nieopartych na HTTP aplikacji i usług w IIS 7.0.

Do wspierania usług i aplikacji, które używają protokołów innych niż HTTP i HTTPS, można użyć technologii, takich jak Windows Communication Foundation (WCF). Model procesu WAS pozwala aplikacjom i usługom opartym na WCF na używanie protokołów HTTP i innych w środowisku hostingu, które wspiera aktywację bazującą na komunikatach i oferuje możliwość udostępniania wielkiej liczby aplikacji na pojedynczym komputerze. Windows Communication Foundation jest dostarczana z adapterami protokołów, które mogą wykorzystywać możliwości WAS, zwiększając niezawodność i wykorzystanie zasobów usług WCF.

WAS ma zdolność odbierania żądań lub komunikatów przez dowolny protokół i jednocześnie wspiera włączaną aktywację arbitralnych odbiorników protokołów. Odbiorniki protokołów odbierają specyficzne dla protokołu żądania, wysyłają je do IIS w celu

przetworzenia i zwracają odpowiedzi żądającym. Adapter odbiornika obejmuje funkcjonalność odbiornika protokołu z WCF. Rysunek 2-14 ukazuje WAS z adapterami odbiornika dla protokołów innych niż HTTP.



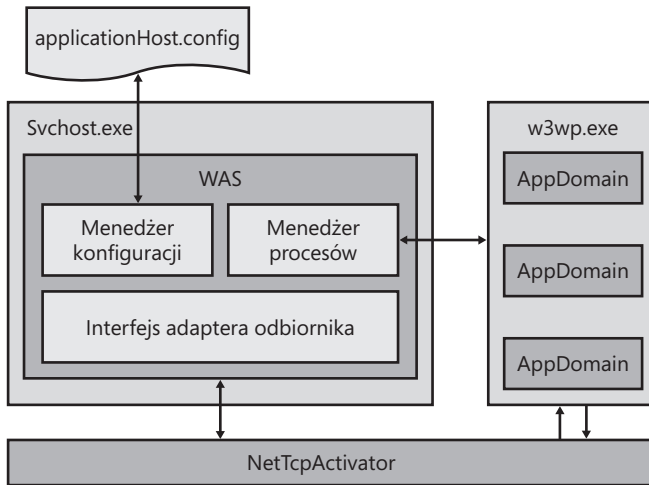
Rysunek 2-14 Wsparcie innego niż HTTP protokołu w WAS

Adaptory protokołów to usługi Windows, które odbierają komunikaty dla specyficznych protokołów sieciowych i komunikują się z WAS w celu kierowania żądań przychodzących do prawidłowego procesu roboczego. Interfejs odbiornika protokołu jest używany do komunikacji z żadaniami aktywacji, które są odbierane przez inne niż HTTP protokoły wspierane. Istnieje wiele niezwiązanych z HTTP adapterów odbiorników, w tym:

- NetTcpActivator dla protokołu TCP,
- NetPipeActivator dla Named Pipes,
- NetMsmqActivator dla Message Queuing (znany także jako MSMQ).

Jeżeli nie jest potrzebna funkcjonalność HTTP, można nawet uruchomić WAS bez W3SVC. Na przykład można zarządzać usługą sieci Web przez adapter odbiornika WCF, taki jak NetTcpActivator, bez uruchamiania W3SVC, jeśli nie trzeba nasłuchiwać żądań HTTP w HTTP.sys.

Globalny magazyn konfiguracji IIS – applicationHost.config – może zawierać konfigurację dla innych niż HTTP protokołów. Na przykład adapter odbiornika TCP – NetTcpActivator – może być konfigurowany na podstawie informacji, jakie WAS odczytuje z magazynu konfiguracji. Po skonfigurowaniu NetTcpActivator, nasłuchuje żądań, które używają protokołu TCP. Gdy adapter odbiornika odbiera żądanie, WAS uruchamia proces roboczy, lecz także adapter odbiornika może przekazać żądanie do przetwarzania. Tę architekturę pokazano na rysunku 2-15.



Rysunek 2-15 Przetwarzanie innych niż HTTP protokołów w IIS 7.0

Ponieważ WAS zarządza procesami dla HTTP i innych protokołów, można uruchamiać aplikację z różnymi protokołami w tej samej puli. Na przykład można udostępniać aplikację przez protokoły HTTP i TCP.

Oprócz niezależności od protokołów model procesu WAS w IIS 7.0 dostarcza wszystkich typów aplikacji aktywowanych komunikatami z inteligentnym zarządzaniem zasobami, aktywacją procesu na żądanie, monitorowaniem stanu oraz automatycznym wykrywaniem awarii i odzyskiwaniem, co pozwala tym aplikacjom na korzystanie z zalet modelu procesu IIS bez wymagania wdrażania pełnej instalacji IIS.



Uwaga Więcej informacji na temat adapterów odbiorników znajduje się w artykule zatytułowanym „WAS Activation Architecture” pod adresem <http://go.microsoft.com/fwlink/?LinkId=88413>.

Podsumowanie

W tym rozdziale przyjrzelśmy się architekturze przetwarzania żądań w IIS 7.0 od początku do końca. IIS 7.0 zawiera wiele rdzennych komponentów, które pracują wspólnie w celu wykonania żądania HTTP. Są to:

- HTTP.sys – odbiornik protokołu HTTP na poziomie jądra,
- World Wide Web Publishing Service (W3SVC) – adapter odbiornika HTTP,
- Windows Process Activation Service (WAS) – usługa, która zapewnia aktywację procesu i zarządzanie nim,
- magazyn konfiguracji – rozproszona hierarchia rozproszonych plików XML, która zawiera zarówno ustawienia IIS, jak i ASP.NET,
- proces roboczy w3wp.exe – samo zawierający proces trybu użytkownika, który uruchamia żądania HTTP i generuje odpowiedzi.

Każdy proces roboczy obsługuje jedną pulę aplikacji. Może być ona konfigurowana w jednym z dwóch zarządzanych trybów potokowych: zintegrowanym lub klasycznym. W zależności od ustawień konfiguracyjnych żądanie ASP.NET może być wykonywane na jeden z dwóch sposobów w procesie roboczym, który obsługuje pulę aplikacji:

- w trybie zintegrowanym przetwarzanie IIS i ASP.NET jest zunifikowane w zintegrowanym potoku przetwarzania,
- w trybie klasycznym potoki IIS i ASP.NET są oddzielne, jak w IIS 6.0.

Zintegrowany potok przetwarzania dostarcza podstaw modularnej architektury IIS 7.0. Eksponuje żądania dla ponad 40 wbudowanych, samo zawierających, natywnych i zarządzanych modułów, które implementują funkcjonalność serwera sieci Web. Najważniejszymi zaletami zintegrowanego potoku przetwarzania jest:

- pozwolenie usługom dostarczonym przez moduły natywne i zarządzane na zastosowanie do wszystkich typów treści,
- eliminacja duplikowania właściwości w IIS i ASP.NET,
- zarządzanie wszystkimi modułami w jednej lokalizacji, dzięki czemu uproszczona staje się administracja witryną i aplikacją na serwerze,
- rozszerzanie IIS za pomocą zarządzanych modułów ASP.NET.

Oprócz wykonywania żądań HTTP IIS 7.0 wspiera utrzymywanie aplikacji i usług nieopartych na HTTP, które mogą korzystać z tego modelu procesu.

Dodatkowe zasoby

Zawierają dodatkowe informacje związane z tym rozdziałem:

- Więcej informacji na temat systemu konfiguracyjnego IIS 7.0 znajduje się w rozdziale 4 „System konfiguracyjny”.
- Więcej informacji na temat kompatybilności aplikacji znajduje się w rozdziale 11 „Utrzymywanie środowisk rozwijania aplikacji”.
- Więcej informacji na temat zintegrowanego potoku przetwarzania i zarządzania modułami serwera Web, znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.
- Pełną listę modułów natywnych i zarządzanych IIS 7.0 można znaleźć w dodatku C „Lista modułów IIS 7.0”.
- Domyślna kolejność wbudowanych modułów IIS 7.0 znajduje się w dodatku D „Sekwencja modułów”.
- Więcej informacji na temat adapterów odbiorników znajduje się w artykule zatytułowanym „WAS Activation Architecture” pod adresem <http://go.microsoft.com/fwlink/?LinkId=88413>.

Rozdział 3

Podstawy struktury modularnej

W tym rozdziale:

Koncepcje	53
Główne zalety	57
Wbudowane moduły	60
Podsumowanie.	61
Dodatkowe zasoby	61

Co oznacza *rdzeń modularny* Microsoft Internet Information Services (IIS) 7.0? W jaki sposób sprawia, że IIS 7.0 są najpotężniejszym z dotychczasowych serwerów sieci Web firmy Microsoft? I jakie są wbudowane moduły dostarczane z IIS 7.0? Po przeczytaniu tego rozdziału Czytelnik będzie w stanie odpowiedzieć na wszystkie te pytania, a przy tym będzie dysponował logiczną wiedzą na temat nowych koncepcji projektowych IIS 7.0. Przyjrzyjmy się idei komponentowego projektu IIS 7.0, intencjom przyświecającym naprawionej architekturze i zaletom tego projektu. Czytelnik otrzyma także szczegółowe informacje na temat wbudowanych modułów, które są dostarczane z IIS 7.0.

Koncepcje

Jedną z głównych zmian w IIS 7.0 jest ich bazująca na komponentach architektura, która uwzględnia to, co zawierały IIS 6.0 i informacje zwrotne od klientów. IIS 7.0 mają kompletnie przeprojektowaną architekturę – rdzeń serwera sieci Web jest teraz podzielony na poufne komponenty, nazywane modułami. Po raz pierwszy administrator sieci Web może zbudować własny serwer IIS zgodnie z własnymi wymaganiami. Można łatwo dodawać wbudowane moduły, kiedy tylko trzeba, lub jeszcze lepiej – dodawać lub zastępować funkcjonalność modułami własnego projektu, produkowanymi komercyjnie lub dostarczanyymi przez społeczność deweloperów na IIS.net. W ten sposób modularny mechanizm pozwala na osiągnięcie dokładnie tej funkcjonalności, jakiej chcemy od serwera sieci Web, a równocześnie zapewnia elastyczność, tak że można usunąć niechciane moduły, aby lepiej zabezpieczyć serwer sieci Web.

Chociaż głównym punktem modularności w IIS 7.0 jest sam serwer sieci Web, właściwości w całej platformie są implementowane jako moduły. Modularny jest na przykład stos administracyjny. Szczegółowe informacje na temat rozszerzalności serwera sieci Web i stosu administracyjnego IIS 7.0 znajdują się w rozdziale 12 „Zarządzanie modułami serwera sieci Web” i w rozdziale 13 „Zarządzanie rozszerzeniami konfiguracji i interfejsu użytkownika”.

Jak to działa: Projekt modułarny

IIS 7.0 są dostarczane z wieloma różnymi modułami. Każdy z nich jest komponentem (ale nie w znaczeniu COM czyli Component Object Model), który dostarcza usług do potoku przetwarzania żądań HTTP serwera sieci Web. Na przykład StaticFileModule jest modulem, który obsługuje całą treść statyczną, taką jak strony HTML, pliki obrazów itp. Inne moduły dostarczają możliwości dla dynamicznej kompresji, podstawowego uwierzytelnienia i innych właściwości, jakie typowo wiążą się z IIS. Moduły IIS 7.0 są zarządzane poufnie. Mogą być łatwo dodawane lub usuwane z rdzennego mechanizmu przez nowy system konfiguracyjny.

Wewnętrznie rdzeń serwera sieci Web IIS dostarcza potoku przetwarzania żądań dla modułów do wykonywania. Zapewnia także usługi przetwarzania żądań, w których moduły rejestrowane w potoku przetwarzania są wywoływane w celu przetwarzania żądań, bazując na zarejestrowanych powiadomieniach zdarzeń. Administrator nie może niczego kontrolować, zdarzenia są zakodowane do użycia przez moduł. Jest to wykonywane w kodzie wewnątrz modułu. Natomiast można kontrolować, które moduły są ładowane globalnie, a nawet to, które są ładowane dla specyficznej witryny lub aplikacji. Szczegóły na temat kontrolowania i ładowania modułu znajdują się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

Za każdym razem, gdy uruchamiany jest proces roboczy IIS 7.0, odczytuje on plik konfiguracyjny serwera i łączy wszystkie globalnie wymienione moduły. Moduły aplikacji są ładowane do aplikacji w czasie pierwszego żądania. Jest to modułarny projekt i system konfiguracyjny, który sprawia, że łatwiejsze stają się włączanie, usuwanie i zastępowanie modułów w potoku żądania, oferujące rozszerzalność do serwera sieci Web IIS 7.0.

Idee

Moduł przypomina klocek w dziecięcym zestawie LEGO zawierającym klocki w różnych kolorach i kształtach. Gdy połączymy go z dodatkowymi klockami z innych zestawów, możemy budować wiele różnych struktur o różnych kształtach. IIS 7.0 korzystają z tej samej idei do projektu swojego środowiska. Przy użyciu modułów jako bloków budulcowych architektura ta połączona z elastycznym systemem konfiguracyjnym i rozszerzalnym interfejsem użytkownika (UI) pozwala na dodawanie lub usuwanie dowolnych właściwości, aby zbudować serwer, który odpowiada specyficznym potrzebom organizacji. Ten nowy i otwarty projekt jest nowy w firmie Microsoft i otwiera nowe drzwi dla platformy sieci Web.

Typy modułów

IIS 7.0 są dostarczane z około 40 modułami, w tym związanymi z bezpieczeństwem modułami uwierzytelniania i modułami do kompresji treści. Moduły tworzą zbiory właściwości serwera sieci Web, a jej aplikacja składa się z wielu modułów obsługujących żądania. Pod względem roli moduły mogą być kategoryzowane jako dostarczające usługi żądań, takich

jak kompresja i uwierzytelnienie, lub *obsługi żądań*, takich jak dostarczanie plików statycznych, stron ASP.NET itp. Bez względu na role moduły są kluczowymi składnikami IIS 7.0. Deweloperzy mogą tworzyć dwa typy modułów IIS:

- **Moduły zarządzane** Moduł zarządzany jest komponentem .NET Framework opartym o model rozszerzalności ASP.NET. W zintegrowanej architekturze przetwarzania IIS 7.0, usługi aplikacji ASP.NET nie są już ograniczone do żądań dla stron .ASPX ani do innej treści mapowanej do ASP.NET. Moduły zarządzane są włączane bezpośrednio do potoku obsługi żądania serwera sieci Web, co sprawia, że są tak potężne jak moduły budowane z wykorzystaniem warstwy rozszerzalności w IIS 7.0. W celu korzystania z usług dostarczanych przez moduł ASP.NET dla wszystkich żądań aplikacja musi być uruchamiana w puli aplikacji używającej trybu zintegrowanego. Ta integracja jest możliwa dzięki modułowi ManagedEngine, który dostarcza integracji .NET do potoku przetwarzania żądań. Moduły zarządzane są ładowane globalnie tylko wtedy, gdy pula aplikacji jest oznaczona jako zintegrowana. Więcej informacji na temat nowego zintegrowanego trybu przetwarzania znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.
- **Moduły natywne** Moduł natywny to biblioteka Microsoft Windows DLL (dynamic-link library), typowo napisana w języku C++, dostarczająca usług przetwarzania żądań. W IIS 7.0 nowy zbiór natywnych serwerowych (C++) interfejsów programowania aplikacji (API) zastąpił filtry i rozszerzenia ISAPI (Internet Server API) dostarczane przez poprzednie wersje IIS. Nowe API są tworzone w obiektowo zorientowanym modelu i wyposażone w silniejsze interfejsy, które dają więcej kontroli, gdy przystępują do przetwarzania żądań i obsługi odpowiedzi. Deweloperzy zaznajomieni z ISAPI i nowymi modułami natywnymi i interfejsami API będą bardzo zadowoleni z tego, że w porównaniu z poprzednimi wersjami IIS znacznie prościej jest teraz programować za pomocą kodu natywnego.



Uwaga Szczegółowe informacje na temat tego, jak napisać moduły natywne, znajdują się w „How to Build a Native Code IIS7 Module Using C++” pod adresem <http://www.iis.net/go/938>.

Deweloperzy w taki sam sposób mogą zarządzać i konfigurować moduły natywne i zarządzane w IIS 7.0, z wyjątkiem sposobu ich wdrażania. Moduły natywne są instalowane globalnie na serwerze i mogą być włączane lub wyłączane dla każdej aplikacji. Moduły zarządzane mogą być włączane globalnie lub dostarczane przez każdą aplikację. Więcej informacji na temat wdrażania modułów znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

Moduły i konfiguracja

Moduły muszą być zarejestrowane w systemie konfiguracyjnym, aby dostarczać pewnych właściwości lub usług do IIS 7.0. Ten podrozdział omawia związek pomiędzy modułami a różnymi sekcjami w pliku konfiguracyjnym i daje przegląd ustawień modułów w magazynie konfiguracyjnym. Więcej informacji na temat systemu konfiguracyjnego IIS 7.0, który bazuje na XML (Extended Markup Language), znajduje się w rozdziale 4 „System konfiguracyjny”.

Wewnątrz sekcji <system.webServer> pliku konfiguracyjnego ApplicationHost.config (głównego pliku konfiguracyjnego serwera) znajdują się trzy inne, różne sekcje związane z modułami:

- **<globalModules>** Konfigurowalne tylko na poziomie serwera. Ta sekcja definiuje wszystkie natywne moduły kodu, które dostarczają usług dla żądań. Deklaracja modułu w sekcji konfiguracyjnej także określa związany plik DLL dostarczający właściwości modułu. Wszystkie moduły natywne muszą być definiowane lub rejestrowane w tej sekcji przed włączeniem ich do wykorzystania w aplikacji, co zdefiniowano w sekcji <modules>.

```
// Przykład sekcji konfiguracyjnej <globalModules>
<globalModules>
...
<add name="StaticCompressionModule" image="%windir%\...\compstat.dll" />
<add name="DefaultDocumentModule" image="%windir%\...\defdoc.dll" />
<add name="DirectoryListingModule" image="%windir%\...\dirlist.dll" />
...
</globalModules>
```

- **<modules>** Konfigurowalna na poziomie serwera i na poziomie aplikacji. Definiuje moduły włączane dla aplikacji. Chociaż moduły natywne są rejestrowane w sekcji <globalModules>, muszą być włączane w sekcji <modules>, zanim będą mogły dostarczać swoich usług dla żądań do aplikacji. Natomiast moduły kodu zarządzanego mogą być dodawane bezpośrednio do sekcji <modules>. Na przykład można dodać własny zarządzany moduł podstawowego uwierzytelniania do pliku aplikacji Web.config lub utworzyć plik ApplicationHost.config na poziomie serwera.

```
// Przykład sekcji konfiguracyjnej <modules>
<modules>
...
<add name="BasicAuthenticationModule" />
<add name="WindowsAuthenticationModule" />
<add name="OutputCache" type="System.Web.Caching.OutputCacheModule"
  precondition="managedHandler" />
<add name="Session" type="System.Web.SessionState.SessionStateModule"
  precondition="managedHandler" />
...
</modules>
```

- **<handlers>** Konfigurowalna na poziomie serwera, poziomie aplikacji i poziomie URL (Uniform Resource Locator) definiuje, w jaki sposób żądania są obsługiwane. Mapuje także programy obsługi dotyczące URL i słów kluczowych HTTP, określając odpowiedni moduł, który wspiera związany program obsługi (handler). Przetwarzając konfigurację mapowania programu obsługi IIS 7.0, wyznacza, które moduły należy wywołać, gdy przychodzi specyficzne żądanie.

```
// Przykład sekcji konfiguracyjnej <handlers>
<handlers accessPolicy="Script, Read">
...
<add name="ASPClassic" path="*.asp" verb="GET,HEAD,POST"
  modules="IsapiModule" scriptProcessor="...\asp.dll" resourceType="File" />
<add name="SecurityCertificate" path="*.cer" verb="GET,HEAD,POST"
  modules="IsapiModule" scriptProcessor="...\asp.dll" resourceType="File" />
<add name="SSINC-stm" path="*.stm" verb="GET,POST"
  modules="ServerSideIncludeModule" resourceType="File" />
...
</handlers>
```

Główne zalety

Modularna architektura w IIS 7.0 oferuje wiele zalet w porównaniu z poprzednimi wersjami IIS. Ten podrozdział opisuje zalety wypływające z projektu. Dostarcza także scenariuszy ilustrujących, jak administrator sieci Web może korzystać z tych funkcjonalności, budując silny serwer sieci Web.

Bezpieczeństwo

Bezpieczeństwo ma najwyższą wagę, gdy dotyczy dzisiejszych aplikacji sieci Web. IIS 6.0 domyślnie nie jest instalowany, z wyjątkiem edycji Windows Server 2003 Web Server. Domyślna instalacja IIS 6.0 udostępnia tylko treść statyczną. Każda inna funkcjonalność jest wyłączona. IIS 7.0 odzwierciedlają modularną naturę serwera sieci Web, pozwalając użytkownikowi instalować tylko te moduły, które są wymagane dla aplikacji. Binaria, które zawierają inne właściwości, nie są instalowane, ale zamiast tego utrzymywane w chronionym buforze instalacji systemu operacyjnego. Oznacza to, że nie będziemy proszeni o CD ani o wskazanie lokalizacji źródłowej podczas instalacji nowych aktualizacji bądź przy dodawaniu właściwości. Binaria, których się nie używa, nie są ładowane przez procesy robocze IIS. Zamiast tego są umieszczane w kwarantannie, więc nie mogą być dostępne. Gdy wprowadzane są aktualizacje bezpieczeństwa z systemu Microsoft, właściwości, które nie zostały zainstalowane, będą w pełni aktualizowane w buforze instalacji. Może to eliminować potrzebę ponownego zastosowanie pakietów serwisowych, gdy zainstaluje się później nowe właściwości.

Z perspektywy bezpieczeństwa projekt modularny ma dużo korzyści, w tym:

- **zminimalizowana powierzchnia ataku** Dając możliwość instalacji tylko tych komponentów, które są potrzebne, IIS 7.0 bezpośrednio minimalizuje powierzchnię możliwego ataku. Punkty ataku są ograniczone do zainstalowanych komponentów, gdyż tylko dla nich istnieją binaria. Ponieważ tylko zainstalowane komponenty mogą być celem potencjalnych programów typu exploit, jest to najlepsza ochrona. Na przykład razem z domyślną instalacją IIS 7.0 instaluje się około 10 komponentów, aby wspierać wewnętrzne rejestrowanie i zarządzanie IIS, a także obsługiwać żądania treści statycznych.
- **redukcja nakładu na utrzymanie** Projekt modularny nie tylko zapewnia nową wydajność podczas dodawania, usuwania i zastępowania komponentów, ale także dostarcza nowych doświadczeń utrzymania dzięki stosowaniu wybieranych poprawek. Należy

zastosować poprawki lub łatki tylko do komponentów, które są wymagane lub zainstalowane. Komponenty nieużywane lub moduły, które nie zostały zainstalowane, nie wymagają *bezpośredniej* uwagi, przy czym nie jest wymagany żaden czas wyłączenia podczas poprawiania tych komponentów, które nie są zainstalowane. Oznacza to także, że do utrzymania lub aktualizacji potrzeba mniej zadań administracyjnych. Na przykład, jeśli serwer IIS 7.0 korzysta dla swoich aplikacji tylko z uwierzytelniania Windows, tylko łatki modułu uwierzytelniania Windows będą stosowane do tego serwera. Z drugiej strony, jeśli moduł uwierzytelniania Basic jest celem znanego exploita, nie jest wymagane natychmiastowe wprowadzanie poprawek, ponieważ moduł nie jest używany. Zauważmy jednak, że Microsoft zaleca stosowanie wszystkich łatek, aby zapewnić aktualność modułów i właściwości, także tych nieużywanych, na wypadek późniejszej instalacji.



Ważne Microsoft zaleca stosowanie wszystkich łatek do serwera. Podczas naprawy komponentów, które nie są używane, serwer nie jest wyłączany. Jeżeli te komponenty zostaną w końcu zainstalowane, automatycznie będą użyte ostatnie wersje ich binariów i nie będzie trzeba ponownie instalować żadnych poprawek.

Ze źródła: Najbezpieczniejszy serwer sieci Web na świecie

Pierwszy raz, gdy prezentowaliśmy IIS 7.0 szerokiej publiczności, był także moją pierwszą sesją TechEd na TechEd 2005. Pierwsze demo pokazywało możliwość podziału na komponenty IIS 7.0, ukazując to, co żartobliwie nazwaliśmy „najbezpieczniejszym serwerem sieci Web na świecie”.

Jako część tej demonstracji pokazałem, jak edytować konfigurację w pliku ApplicationHost.config, usuwając wszystkie moduły i mapowania programów obsługi. Po zapisaniu pliku IIS automatycznie przechwycił zmiany i ponownie się uruchomił, nie ładując żadnych modułów. Po wykonaniu żądania do domyślnej witryny sieci Web szybko otrzymałem odpowiedź `empty 200` (ta konfiguracja aktualnie zwraca błąd `401 Unauthorized`, ponieważ nie są obecne żadne moduły uwierzytelniania). Serwer nie ma żadnych załadowanych modułów i dlatego nie będzie dokonywał żadnego przetwarzania żądań, ani nie będzie zwracał żadnej treści, stając się w ten sposób naprawdę najbezpieczniejszym serwerem sieci Web na świecie. Później skomentowałem, że chociaż bezpieczny, ten serwer jest także całkiem bezużyteczny i przeszedłem do dodawania z powrotem funkcjonalności, jakiej potrzebowałem dla swojej aplikacji.

Wykonywałem tę demonstrację już wcześniej dla publiczności z firmy Microsoft i byłem oklaskiwany, ale zawsze będę pamiętać reakcję ludzi podczas tej sesji TechEd. Publiczność szalała, czasem nawet przechodząc do owacji na stojąco. Było to głośne uznanie naszych wysiłków, aby dać administratorom możliwość startu od zera, budując serwer z absolutnie minimalnym zbiorem właściwości w celu wytworzenia łatwego do zarządzania serwera sieci Web o najmniejszej możliwej powierzchni.

Mike Volodarsky, IIS7 Core Server Program Manager

- **Ujednolicony model bezpieczeństwa** IIS 7.0 są teraz lepiej zintegrowane z ASP.NET. Posiadanie modułów natywnych IIS 7.0 i modułów zarządzania ASP.NET działających w tym samym potoku żądań ma wiele zalet obejmujących ujednolicenie systemu konfiguracyjnego i modelu bezpieczeństwa IIS i ASP.NET. Gdy patrzymy z perspektywy bezpieczeństwa, zaawansowane usługi bezpieczeństwa ASP.NET mogą być włączone bezpośrednio do potoku głównego przetwarzania żądań IIS i używane razem z właściwościami bezpieczeństwa oferowanymi przez IIS. Mówiąc w skrócie, możliwe jest teraz konfigurowanie razem z IIS 7.0 usług bezpieczeństwa ASP.NET dla innych żądań niż ASP.NET. Na przykład we wcześniejszych wersjach IIS, jeśli aplikacja składała się z zasobów PHP i ASP.NET, uwierzytelnianie ASP.NET Forms mogło być zastosowane tylko do zasobów ASP.NET. Dzięki zintegrowanemu modelowi przetwarzania IIS 7.0 jest teraz możliwe korzystanie z uwierzytelniania Forms dla PHP, ASP.NET, a także z zasobów innych typów, takich jak treści statyczne (HTML, obrazy) i strony ASP.

Wydajność

Wraz z komponentową architekturą IIS 7.0 dostarcza bardzo ziarnistej kontroli, jeśli chodzi o ślady pamięci serwera sieci Web. Moduły są ładowane do pamięci tylko wtedy, gdy są zainstalowane i włączone. Gdy usuwamy niepotrzebne właściwości IIS 7.0, mniej komponentów zostaje załadowanych do potoku przetwarzania – innymi słowy – potrzeba mniej kroków do wypełnienia przychodzących żądań i dlatego ogólna wydajność serwera się zwiększa. Jednocześnie redukując wykorzystanie pamięci dla IIS 7.0, mamy więcej wolnej pamięci, dostępnej dla aplikacji sieci Web i systemu operacyjnego. Na przykład w IIS 6.0 wszyscy dostawcy uwierzytelnienia (Anonymous, Windows, Digest itp.) byli ładowani do procesu roboczego. W IIS 7.0 tylko niezbędne moduły uwierzytelniania są ładowane podczas przetwarzania żądań. Więcej szczegółów na temat usuwania niewymaganych modułów znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

Rozszerzalność

W poprzednich wersjach IIS rozszerzanie lub dodawanie właściwości IIS nie było proste, ponieważ mogło być dokonywane tylko przez programowanie ISAPI z ograniczonym wsparciem API i ograniczonym dostępem do informacji w potoku przetwarzania żądań. W nowym modularnym mechanizmie i ścisłej integracji pomiędzy ASP.NET a IIS rozszerzanie IIS 7.0 jest znacznie prostsze. Moduły IIS 7.0 mogą być tworzone za pomocą natywnego Web Server C++ API lub przy użyciu interfejsów ASP.NET i funkcjonalności .NET Framework. Nie tylko można zadecydować, które właściwości umieścić w serwerze sieci Web, ale także można rozszerzyć ten serwer, dodając własne komponenty w celu dostarczenia specyficznej funkcjonalności.

Na przykład można utworzyć moduł uwierzytelniania podstawowego ASP.NET korzystający z usługi Membership i bazy danych SQL Server zamiast wbudowanej funkcjonalności uwierzytelniania IIS Basic, która działa tylko z kontami Windows. Mówiąc w skrócie, można budować własny serwer, aby dostarczyć zbioru funkcjonalności, jaki jest wymagany przez aplikacje. Można na przykład wdrażać zbiór serwerów IIS 7.0 tylko do celów buforowania lub wdrożyć własny moduł do wykonywania specyficznej funkcji w aplikacji, takiej

jak własna aplikacja ASP.NET, implementująca algorytm równoważenia obciążenia bazujący na wymaganiach klienta. Więcej informacji na temat dostosowywania modułów w IIS 7.0 znajduje się w rozdziale 12 „Zarządzanie modułami serwera sieci Web”.

Wbudowane moduły

Moduły dostarczane z IIS 7.0 są pogrupowane w różnych kategoriach pod względem ról usług, jakich dostarczają. Tabela 3-1 podaje różne kategorie usług i wymienia przykładowe wbudowane moduły w tych kategoriach. Pełna lista modułów znajduje się w dodatku C „Lista modułów IIS 7.0”.

Tabela 3-1 Kategorie modułów

Kategoria	Moduł
Wdrażanie aplikacji	CgiModule (%windir%\system32\inetsrv\cgi.dll) Wspiera programy CGI (Common Gateway Interface).
	FastCgiModule (%windir%\system32\inetsrv\iisfcgi.dll) Wspiera FastCGI dostarczające wysokiej wydajności alternatywy dla starego typu programów opartych na CGI.
	System.Web.SessionState.SessionStateModule (ManagedEngine) Dostarcza zarządzanie stanem sesji, co pozwala na przechowywanie danych specyficznych dla pojedynczego klienta w aplikacji na serwerze.
Kondycja i diagnostyka	FailedRequestsTracingModule (%windir%\system32\inetsrv\iisfrec.dll) Znany powszechnie jako Failed Request Event Buffering (FREB). Wspiera śledzenie błędnych żądań. Definicja i reguły definiujące błędne żądanie mogą być skonfigurowane.
	RequestMonitorModule (%windir%\system32\inetsrv\iisreqs.dll) Implementuje interfejs Run-time State and Control API (RSCA), który pozwala klientom na odpytywanie informacji o czasie wykonania, takich jak aktualnie wykonywane żądanie, stan uruchomienia lub zatrzymania witryny sieci Web lub domeny aktualnie wykonywanych aplikacji.
Właściwości HTTP	ProtocolSupportModule (%windir%\system32\inetsrv\protsup.dll) Implementuje własne i przekierowywane nagłówki odpowiedzi, obsługuje słowa kluczowe HTTP TRACE i OPTIONS oraz wspiera konfigurację podtrzymywania pracy.
Wydajność	TokenCacheModule (%windir%\system32\inetsrv\cachtokn.dll) Buforuje tokeny bezpieczeństwa Windows dla opartych na hasłach schematów (uwierzytelnianie anonimowe, podstawowe i za pomocą certyfikatów klienta IIS).
	System.Web.Caching.OutputCacheModule (ManagedEngine) Definiuje zasady buforowania wyjścia dla strony ASP.NET lub kontrolki użytkownika zawartej na stronie.