

Wydawnictwo Helion ul. Chopina 6 44-100 Gliwice tel. (32)230-98-63 e-mail: helion@helion.pl



Visual Basic .Net. Szybki start

Autor: Harold Davis Tłumaczenie: Krzysztof Jurczyk, Tomasz Walczak ISBN: 83-7361-737-X Tytuł oryginału: Visual Basic .Net for Windows Visual Quickstart Guide Format: B5, stron: 600



Kurs tworzenia aplikacji WWW w języku Visual Basic .NET

Visual Basic od dawna cieszy się uznaniem wśród informatyków. Według ostatnich badań, jest wykorzystywany przez ponad 5 milionów programistów z całego świata. Visual Basic jest niezwykle łatwy do opanowania, a jednocześnie posiada ogromne możliwości i elastyczność. Jego najnowsza wersja - Visual Basic .NET otwiera przed twórcami oprogramowania nowe horyzonty związane z platformą .NET – szybki i wydajny dostęp do baz danych, stosowanie języka XML oraz możliwość tworzenia aplikacji typu Web Services to tylko niektóre z nich. Język VB .NET jest potężnym, eleganckim i w pełni obiektowym nowoczesnym językiem programowania, w którym jednak została zachowana prostota, która zawsze była cechą charakterystyczną języków z rodziny Basic.

"Visual Basic .NET. Szybki start" to książka dla tych, którzy chcą poznać możliwości najnowszej wersji Visual Basica bez konieczności przebijania się przez tysiące linii przykładowych kodów i setki stron opisów. Każde z omawianych zagadnień przedstawione jest w postaci sekwencji czynności opatrzonej ilustracjami. Nie znajdziesz tu rozważań teoretycznych – poznasz tajniki środowiska .NET, obiektów i programowania poprzez rozwiązywanie przykładowych zagadnień.

- Środowisko Visual Studio .NET
- Projektowanie aplikacji Web Services
- · Klasy i obiekty
- Tworzenie formularzy Windows
- Okna komunikatów
- Praca z formularzami MDI
- Obsługa wyjątków i usuwanie błędów
- Stosowanie komponentów
- Dodawanie menu do aplikacji
- Bazy danych oraz technologie ADO.NET i XML

Aplikacje internetowe ASP.NET

Poznaj język Visual Basic szybko i bez problemów. Dołącz do ogromnej społeczności programistów VB.NET

Spis treści

	Wprowadzenie	9
Rozdział 1.	Poznajemy Visual Studio .NET	15
	Strona startowa	16
	Solucie projekty i moduły	
	Narzędzia środowiska Visual Studio	
Rozdział 2.	Tworzenie usług Web Service	43
	Poznajemy usługi Web Service	44
	Tworzenie usługi Web Service	47
	Testowanie usług Web Service	
	Dodawanie metod Web	53
	Testowanie metody Web	54
≀ozdział 3.	Praca z klasami	55
	Poznajemy obiekty i klasy	56
	Dodawanie klasy do usługi Web Service	
	Dodawanie klasy do aplikacji Windows	
	Składowe klasy	
	Zasięg	
	Przeciążanie metody	
	Konstruktory klasy	
	Dziedziezenie wizuelne	
Rozdział 4.	Interfejsy klasy	101
	Abstrakcyjne klasy bazowe	
	Interfejsy klasy	111
	Tworzenie własnego interfejsu klasy	118
Rozdział 5.	Formularze Windows	125
	Klasa Form	
	Właściwości formularza	
	Zmiana kształtu formularza	151

	Praca z egzemplarzami formularza	154
	Uruchamianie aplikacji z Sub Main	167
Rozdział 6.	Używanie usług Web Service	173
	Tworzenie interfejsu użytkownika	174
	Sprawdzanie poprawności danych wprowadzanych przez użytkownika	177
	Dodawanie odnośnika do usługi Web Service	180
	Używanie usługi	182
Rozdział 7.	Okna komunikatu i instrukcja If	187
	Klasa MessageBox	188
	Tworzenie interfejsu użytkownika	208
	Stosowanie instrukcji If	214
	Podłączanie formularza	215
Rozdział 8.	Praca z formularzami MDI	225
	Używanie formularza nadrzędnego	227
	Używanie formularzy potomnych	230
	Używanie tablic	237
	Klasy pojemnikowe Collection	245
Rozdział 9.	Wyjątki i usuwanie błędów	247
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków	247
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try	247 248 249
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków	247 248 249 261
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception	247 248 249 261 264
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions	247 248 249 261 264 265
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions Poznajemy różne rodzaje błędów	247 248 249 261 264 265 267
Rozdział 9.	Wyjątki i usuwanie błędówStrukturalna obsługa wyjątkówUżywanie instrukcji TryGenerowanie wyjątkówPraca z obiektami ExceptionUżywanie okna dialogowego ExceptionsPoznajemy różne rodzaje błędówKorzystanie z metod obiektu Debug	247 248 269 261 264 265 267 270
Rozdział 9.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions Poznajemy różne rodzaje błędów Korzystanie z metod obiektu Debug Używanie poleceń Step i Watch	247 248 269 261 264 265 267 270 273
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędówStrukturalna obsługa wyjątkówUżywanie instrukcji TryGenerowanie wyjątkówPraca z obiektami ExceptionUżywanie okna dialogowego ExceptionsUżywanie okna dialogowego ExceptionsPoznajemy różne rodzaje błędówKorzystanie z metod obiektu DebugUżywanie poleceń Step i WatchKontrolki umożliwiające interakcję z użytkownikiem	247 248 269 261 264 265 267 270 273 277
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędówStrukturalna obsługa wyjątkówUżywanie instrukcji TryGenerowanie wyjątkówPraca z obiektami ExceptionUżywanie okna dialogowego ExceptionsUżywanie okna dialogowego ExceptionsPoznajemy różne rodzaje błędówKorzystanie z metod obiektu DebugUżywanie poleceń Step i WatchKontrolki umożliwiające interakcję z użytkownikiemDostosowanie paska narzędzi do własnych wymagań	247 248 269 261 265 267 270 273 277 278
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędówStrukturalna obsługa wyjątkówUżywanie instrukcji TryGenerowanie wyjątkówPraca z obiektami ExceptionUżywanie okna dialogowego ExceptionsUżywanie okna dialogowego ExceptionsPoznajemy różne rodzaje błędówKorzystanie z metod obiektu DebugUżywanie poleceń Step i WatchKontrolki umożliwiające interakcję z użytkownikiemDostosowanie paska narzędzi do własnych wymagańPolecenia menu Format	247 248 249 261 264 265 267 270 273 277 278 284
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions Voznajemy różne rodzaje błędów Korzystanie z metod obiektu Debug Używanie poleceń Step i Watch Kontrolki umożliwiające interakcję z użytkownikiem Dostosowanie paska narzędzi do własnych wymagań Polecenia menu Format Praca z grupami kontrolek	247 248 269 261 265 265 267 270 273 277 278 284 288
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions Voznajemy różne rodzaje błędów Korzystanie z metod obiektu Debug Używanie poleceń Step i Watch Kontrolki umożliwiające interakcję z użytkownikiem Dostosowanie paska narzędzi do własnych wymagań Polecenia menu Format Praca z grupami kontrolek Praca z kontrolką paska narzędzi ToolBar	247 248 249 261 264 265 267 270 273 277 278 284 284 284 294
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions Poznajemy różne rodzaje błędów Korzystanie z metod obiektu Debug Używanie poleceń Step i Watch Kontrolki umożliwiające interakcję z użytkownikiem Dostosowanie paska narzędzi do własnych wymagań Polecenia menu Format Praca z grupami kontrolek Praca z kontrolką paska narzędzi ToolBar Użycie kontrolki ToolTip	247 248 249 261 265 267 270 273 277 278 284 284 288 294 284 294
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions Poznajemy różne rodzaje błędów Korzystanie z metod obiektu Debug Używanie poleceń Step i Watch Kontrolki umożliwiające interakcję z użytkownikiem Dostosowanie paska narzędzi do własnych wymagań Polecenia menu Format Praca z grupami kontrolek Praca z kontrolką paska narzędzi ToolBar Użycie kontrolki ToolTip Obsługa zakładek za pomocą kontrolki TabControl	247 248 249 261 265 267 270 273 277 278 278 284 288 294 305
Rozdział 9. Rozdział 10.	Wyjątki i usuwanie błędów Strukturalna obsługa wyjątków Używanie instrukcji Try. Generowanie wyjątków Praca z obiektami Exception Używanie okna dialogowego Exceptions. Poznajemy różne rodzaje błędów Korzystanie z metod obiektu Debug Używanie poleceń Step i Watch. Kontrolki umożliwiające interakcję z użytkownikiem Dostosowanie paska narzędzi do własnych wymagań Polecenia menu Format Praca z grupami kontrolek. Praca z kontrolką paska narzędzi ToolBar Użycie kontrolki ToolTip. Obsługa zakładek za pomocą kontrolki TabControl Korzystanie z kontrolki TrackBar	247 248 249 261 264 265 267 270 273 277 278 284 284 294 303 305 311

Zasada działania komponentu Timer 332 Wyświetlanie bieżącego czasu 337 Animacja duszka 342 Rozdzioł 12. System menu 353 Dodawanie mazw elementów menu 361 Doskonalenie systemu menu 364 Obsługa zdarzeń dotyczących menu 369 Modyfikacja menu za pomocą kodu aplikacji 377 Dodawanie menu za pomocą kodu aplikacji 388 Rozdzioł 13. Kontrolki Common Dialog 401 Przykładowa aplikacja. 403 Okno dialogowe Czcionka 408 Okno dialogowe Kolor 411 Okna dialogowe File 414 Korzystanie z obiektu FileStream 423 Drukowanie 428 Rozdzioł 14. Przeglądarki obiektów Object Browser 437 Otwieranie przeglądarki obiektów 438 Interfejs przeglądarki obiektów 448 Przestrzenie nazw 451 Waźniejsze przestrzenie nazw 451 Waźniejsze przestrzenie nazw 451 Praca z XML 463 Praca z z bazami danych 453 Rozdzioł 15. XML, zbiory danych i ADO.NET 513 </th <th>Rozdział 11.</th> <th>Komponent Timer</th> <th>331</th>	Rozdział 11.	Komponent Timer	331
Wyświetlanie bieżącego czasu 337 Animacja duszka 342 Rozdzioł 12. System menu 353 Dodawanie menu 354 Definiowanie nazw elementów menu 364 Obsługa zdarzeń dotyczących menu 369 Modyfikacja menu za pomocą kodu aplikacji 385 Implementacja menu za pomocą kodu aplikacji 385 Implementacja menu w aplikacjach MDI 388 Rozdzioł 13. Kontrolki Common Dialog 401 Przykładowa aplikacja 403 Okno dialogowe Czcionka 408 Okno dialogowe Kolor 411 Okna dialogowe Fole 414 Kozzystanie z obiektu FileStream 423 Drukowanie 428 Rozdzioł 14. Przeglądarka obiektów Object Browser 437 Otwieranie przeglądarki obiektów 438 Interfejs przeglądarki obiektów 448 Przestrzenie nazw 451 Ważniejsze przestrzenie nazw 451 Ważniejsze przestrzenie nazw.NET 453 Rozdzioł 15. XML, zbiory danych i ADO.NET 461 Praca z XML 463 Praca z bazami danych 483 </td <td></td> <td>Zasada działania komponentu Timer</td> <td></td>		Zasada działania komponentu Timer	
Animacja duszka 342 Rozdział 12. System menu 353 Dodawanie mazw elementów menu 361 Doskonalenie systemu menu 364 Obstuga zdarzeń dotyczących menu 369 Modyfikacja menu za pomocą kodu aplikacji 377 Dodawanie menu za pomocą kodu aplikacji 388 Rozdział 13. Kontrolki Common Dialog 401 Przykładowa aplikacja. 403 Okno dialogowe Czcionka 408 Okno dialogowe Kolor 411 Okna dialogowe File 414 Korzystanie z obiektu FileStream 423 Drukowanie 428 Rozdział 14. Przeglądarki obiektów Object Browser 437 Otwieranie przeglądarki obiektów 438 Interfejs przeglądarki obiektów 438 Interfejs przeglądarki obiektów 438 Rozdział 15. XML, zbiory danych i ADO.NET 461 Praca z XML 463 Praca z bazami danych 483 Rozdział 16. Aplikacji internetowe ASP.NET 513 Podstawy ASP.NET 513 Podstawy ASP.NET 514 <t< td=""><td></td><td>Wyświetlanie bieżącego czasu</td><td></td></t<>		Wyświetlanie bieżącego czasu	
Rozdzioł 12. System menu 353 Dodawanie menu 354 Definiowanie nazw elementów menu 361 Doskonalenie systemu menu 364 Obsługa zdarzeń dotyczących menu 369 Modyfikacja menu za pomocą kodu aplikacji 377 Dodawanie menu za pomocą kodu aplikacji 385 Implementacja menu w aplikacjach MDI. 388 Rozdzioł 13. Kontrolki Common Dialog 401 Przykładowa aplikacja 403 Okno dialogowe Czcionka 408 Okno dialogowe Kolor 411 Okna dialogowe Kolor 411 Okna dialogowe Kolor 414 Korzystanie z obiektu FileStream 423 Drukowanie 428 Rozdzioł 14. Przeglądarka obiektów Object Browser 437 Otwieranie przeglądarki obiektów 444 Pakiety 448 Przestrzenie nazw 451 Ważniejsze przestrzenie nazw .NET 458 Rozdzioł 15. XML, zbiory danych i ADO.NET 461 Praca z XML 463 Praca z bazami danych 483 Rozdzioł 16.		Animacja duszka	
Dodawanie menu354Definiowanie nazw elementów menu361Doskonalenie systemu menu364Obsługa zdarzeń dotyczących menu369Modyfikacja menu za pomocą kodu aplikacji377Dodawanie menu za pomocą kodu aplikacji385Implementacja menu w apikacjach MDI388Rozdzioł 13.Kontrolki Common Dialog401Przykładowa aplikacja403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzioł 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów448Przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw463Praca z XML463Praca z bazami danych483Rozdzioł 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET513Podstawy ASP.NET513Porze z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560	Rozdział 12.	System menu	353
Definiowanie nazw elementów menu361Doskonalenie systemu menu364Obsługa zdarzeń dotyczących menu369Modyfikacja menu za pomocą kodu aplikacji377Dodawanie menu za pomocą kodu aplikacji385Implementacja menu w aplikacjach MDI388Rozdzich 13. Kontrolki Common Dialog401Przykładowa aplikacja403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzich 14. Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw .NET451Ważniejsze przestrzenie nazw .NET461Praca z XML463Praca z bazami danych483Rozdzich 16. Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z systemu pomocy online558Korzystanie z systemu pomocy online558		Dodawanie menu	
Doskonalenie systemu menu364Obsługa zdarzeń dotyczących menu369Modyfikacja menu za pomocą kodu aplikacji377Dodawanie menu za pomocą kodu aplikacji385Implementacja menu w aplikacjach MDI388Rozdział 13.Kontrolki Common Dialog401Przykładowa aplikacja403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream428Rozdział 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów448Przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw463Praca z XML463Praca z XML463Praca z XML463Praca z z bazami danych451Ważnieji internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z pomocy dynamicznej560		Definiowanie nazw elementów menu	
Obsługa zdarzeń dotyczących menu369Modyfikacja menu za pomocą kodu aplikacji377Dodawanie menu za pomocą kodu aplikacji385Implementacja menu w aplikacjach MDI388Rozdzich 13.Kontrolki Common Dialog401Przykładowa aplikacja403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzich 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów448Przestrzenie nazw448Przestrzenie nazw451Ważniejsze przestrzenie nazw453Rozdzich 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z XML463Praca z kontrolkami515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowe ASP.NET518Dodstawy ASP.NET519Praca z kontrolkami530Dodotek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Doskonalenie systemu menu	
Modyfikacja menu za pomocą kodu aplikacji377Dodawanie menu za pomocą kodu aplikacji385Implementacja menu w aplikacjach MDI388Rozdzicł 13. Kontrolki Common Dialog401Przykładowa aplikacja403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzicł 14. Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw .NET458Rozdzicł 15. XML, zbiory danych i ADO.NET461Praca z XML463Praca z kontrolkami danych483Rozdzicł 16. Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z pomocy dynamicznej560		Obsługa zdarzeń dotyczących menu	
Dodawanie menu za pomocą kodu aplikacji385Implementacja menu w aplikacjach MDI388Rozdzicł 13.Kontrolki Common Dialog401Przykładowa aplikacja403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzicł 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów448Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw .NET458Rozdzicł 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z z bazami danych483Rozdzicł 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z pomocy dynamicznej560		Modyfikacja menu za pomocą kodu aplikacji	
Implementacja menu w aplikacjach MDI.388Rozdzicł 13.Kontrolki Common Dialog401Przykładowa aplikacja.403Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzicł 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów448Przestrzenie nazw441Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw451Praca z XML463Praca z XML463Praca z XML463Praca z kontroltwe ASP.NET513Podstawy ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z pomocy dynamicznej560		Dodawanie menu za pomocą kodu aplikacji	
Rozdzicł 13. Kontrolki Common Dialog 401 Przykładowa aplikacja. 403 Okno dialogowe Czcionka 408 Okno dialogowe Kolor 411 Okna dialogowe File 414 Korzystanie z obiektu FileStream 423 Drukowanie 428 Rozdzicł 14. Przeglądarka obiektów Object Browser 437 Otwieranie przeglądarki obiektów 438 Interfejs przeglądarki obiektów 442 Pakiety 448 Przestrzenie nazw 451 Ważniejsze przestrzenie nazw. 451 Ważniejsze przestrzenie nazw. 463 Praca z XML 463 Praca z bazami danych 483 Rozdzicł 16. Aplikacje internetowe ASP.NET 513 Podstawy ASP.NET 513 Podstawy ASP.NET 515 Warunki niezbędne do tworzenia aplikacji ASP.NET 518 Tworzenie aplikacji internetowych ASP.NET 519 Praca z kontrolkami 530 Dodatek A Pomoc w środowisku Visual Studio 557 Korzystanie z systemu pomocy online 558 Korzystanie z pomocy dynamicznej 560		Implementacja menu w aplikacjach MDI	
Przykładowa aplikacja	Rozdział 13.	Kontrolki Common Dialog	401
Okno dialogowe Czcionka408Okno dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdział 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw .NET458Rozdział 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Przykładowa aplikacja	
Okno dialogowe Kolor411Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdział 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw451Praca z XML463Praca z bazami danych461Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Okno dialogowe Czcionka	
Okna dialogowe File414Korzystanie z obiektu FileStream423Drukowanie428Rozdzicił 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw451Rozdzicił 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdzicił 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Okno dialogowe Kolor	411
Korzystanie z obiektu FileStream423 DrukowanieDrukowanie428Rozdział 14.Przeglądarka obiektów Object Browser437 Otwieranie przeglądarki obiektówOtwieranie przeglądarki obiektów438 Interfejs przeglądarki obiektów442 PakietyPakiety448 Przestrzenie nazw451 Waźniejsze przestrzenie nazw451 Waźniejsze przestrzenie nazw458Rozdział 15.XML, zbiory danych i ADO.NET461 Praca z XML463 Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513 Podstawy ASP.NET513 Sis Warunki niezbędne do tworzenia aplikacji ASP.NET518 Sis Sis Dodatek APomoc w środowisku Visual Studio557 Korzystanie z systemu pomocy online558 Korzystanie z pomocy dynamicznej560		Okna dialogowe File	414
Drukowanie428Rozdział 14.Przeglądarka obiektów Object Browser437Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw458Rozdział 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Korzystanie z obiektu FileStream	
Rozdzici 14.Przeglądarka obiektów Object Browser437 Otwieranie przeglądarki obiektówOtwieranie przeglądarki obiektów438 Interfejs przeglądarki obiektów442 PakietyPakiety448 Przestrzenie nazw451 Ważniejsze przestrzenie nazw .NET458Rozdzici 15.XML, zbiory danych i ADO.NET461 Praca z XML463 Praca z bazami danych483Rozdzici 16.Aplikacje internetowe ASP.NET513 Podstawy ASP.NET515 Warunki niezbędne do tworzenia aplikacji ASP.NET516 S10Dodatek APomoc w środowisku Visual Studio557 Korzystanie z systemu pomocy online558 Korzystanie z pomocy dynamicznej560		Drukowanie	
Otwieranie przeglądarki obiektów438Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw .NET458Rozdział 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560	Rozdział 14.	Przeglądarka obiektów Object Browser	437
Interfejs przeglądarki obiektów442Pakiety448Przestrzenie nazw451Ważniejsze przestrzenie nazw451Ważniejsze przestrzenie nazw458Rozdział 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Otwieranie przeglądarki obiektów	
Pakiety		Interfejs przeglądarki obiektów	
Przestrzenie nazw451Ważniejsze przestrzenie nazw .NET458Rozdział 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Pakiety	
Ważniejsze przestrzenie nazw .NET		Przestrzenie nazw	451
Rozdzicł 15.XML, zbiory danych i ADO.NET461Praca z XML463Praca z bazami danych483Rozdzicł 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Ważniejsze przestrzenie nazw .NET	
Praca z XML463Praca z bazami danych483Rozdzicł 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560	Rozdział 15.	XML, zbiory danych i ADO.NET	461
Praca z bazami danych483Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Praca z XML	
Rozdział 16.Aplikacje internetowe ASP.NET513Podstawy ASP.NET515Warunki niezbędne do tworzenia aplikacji ASP.NET518Tworzenie aplikacji internetowych ASP.NET519Praca z kontrolkami530Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online558Korzystanie z pomocy dynamicznej560		Praca z bazami danych	
Podstawy ASP.NET 515 Warunki niezbędne do tworzenia aplikacji ASP.NET 518 Tworzenie aplikacji internetowych ASP.NET 519 Praca z kontrolkami 530 Dodatek A Pomoc w środowisku Visual Studio 557 Korzystanie z systemu pomocy online 558 Korzystanie z pomocy dynamicznej 560	Rozdział 16.	Aplikacie internetowe ASP.NET	513
Warunki niezbędne do tworzenia aplikacji ASP.NET 518 Tworzenie aplikacji internetowych ASP.NET 519 Praca z kontrolkami 530 Dodatek A Pomoc w środowisku Visual Studio 557 Korzystanie z systemu pomocy online 558 Korzystanie z pomocy dynamicznej 560		Podstawy ASP.NET	
Tworzenie aplikacji internetowych ASP.NET 519 Praca z kontrolkami 530 Dodatek A Pomoc w środowisku Visual Studio 557 Korzystanie z systemu pomocy online 558 Korzystanie z pomocy dynamicznej 560		Warunki niezbędne do tworzenia aplikacji ASP.NET	
Praca z kontrolkami		Tworzenie aplikacji internetowych ASP.NET	
Dodatek APomoc w środowisku Visual Studio557Korzystanie z systemu pomocy online		Praca z kontrolkami	530
Korzystanie z systemu pomocy online	Dodatek A	Pomoc w środowisku Visual Studio	557
Korzystanie z pomocy dynamicznej		Korzystanie z systemu pomocy online	
		Korzystanie z pomocy dynamicznej	

	Korzystanie z okna zawartości Contents	
	Korzystanie z okna Index	
	Korzystanie z okna Search	
	Filtrowanie	566
Dodatek B	Słowa kluczowe Visual Basic .NET	569
Dodatek C	Typy i konwersja typów	571
	Typy Visual Basic .NET	
	Opcja kompilatora Option Strict	
	Podstawy silnego typowania	
	Korzystanie z konwersji niejawnej	
	Korzystanie z funkcji CType	
	Korzystanie z dołączanych funkcji konwersji typów	
	Stosowanie innych metod konwersji	
	Skorowidz	585

Tworzenie usług Web Service

Teraz, kiedy wiesz już, jak poruszać się w środowisku Visual Studio .NET, pora, abyś stworzył pierwszy program Visual Studio .NET.

Jedną z najbardziej spektakularnych cech platformy .NET jest łatwe i szybkie tworzenie usług Web Service. Dlatego też pierwszym programem, jaki utworzysz, będzie opisana w tym rozdziale usługa *ASP.NET Web Services*.

Poznajemy usługi Web Service

Czym jest usługa Web Service? W najprostszej postaci usługa Web Service dostarcza metody, które mogą być wywołane przez sieć za pomocą standardowych protokołów, takich jak HTTP (ang. *Hypertext Transfer Protocol*) oraz XML (ang. *eXtensible Markup Language*). *Metoda* jest to funkcja lub procedura, która związana jest z danym obiektem. Innymi słowy, wywołanie metody powoduje, że wykonywane jest określone dla niej działanie.

Usługi Web Service są o tyle ciekawe, że pozwalają tworzyć programy, które są całkowicie międzyplatformowe. Programy te mogą być uruchamiane poprzez sieć internetową niezależnie od użytych do ich tworzenia narzędzi oraz bez względu na system, na jakim mają działać.

Inną charakterystyczną cechą usług Web Service jest to, że dostarczają pełnej informacji o wykonywanych działaniach. Oznacza to, że można sprawdzić, jakie są nazwy metod, których dana usługa dostarcza, jakie argumenty przyjmują poszczególne metody oraz jakie wartości zwracają. Jeśli więc wie się, że dana usługa Web Service istnieje, to można z niej korzystać w tworzonym programie

Celem, dla którego utworzono usługi Web Service, jest dostarczenie komponentów o specyficznych cechach. Komponenty te mogą się ze sobą komunikować niezależnie od języka programowania, w którym zostały napisane oraz bez względu na system operacyjny, na którym są uruchamiane. Metody tych komponentów mogą być używane przez inne programy. Aby użyć usługi Web Service, program wysyła żądanie za pomocą protokołu HTTP, i aby posłużyć się konkretną metodą, wywołuje HTTP GET, HTTP POST lub — najczęściej — używa protokołu SOAP (ang. Simple Object Access Protocol). Dzięki platformie .NET programista nie musi wgłębiać się w szczegóły kodu protokołu SOAP. Ponieważ jednak usługi Web Service korzystają z żądań protokołu SOAP, warto zapoznać się z istotą jego działania.

We wcześniejszym fragmencie znalazła się informacja, że usługi Web Service dostarczają uniwersalnego mechanizmu, który informuje o używanych metodach. W celu implementacji tego mechanizmu platforma .NET automatycznie tworzy dokument w języku WSDL (ang. *Web Services Description Language*), który może zostać odczytany przez aplikację lub człowieka. Dokument ten pozwala zrozumieć, jak działa dana usługa Web Service. Platforma .NET uwalnia programistę od obowiązku tworzenia dokumentów WSDL, warto jednak wiedzieć, że taki dokument istnieje i do czego służy.

Usługi Web Service korzystają także ze standardu UDDI (ang. Universal Description, Discovery and Integration). Standard ten jest czymś w rodzaju Panoramy Firm dla usług Web Service. Jeśli chcesz odnaleźć usługę Web Service lub jeśli chcesz, aby dana usługa mogła zostać odnaleziona, musisz skorzystać ze standardu UDDI.

W następnych paragrafach przedstawione zostaną podstawowe cechy trzech głównych komponentów usług Web Service — SOAP, WSDL oraz UDDI. Podane także zostaną niezbędne informacje, potrzebne do utworzenia zaawansowanej usługi Web Service za pomocą platformy .NET.

SOAP

Standard SOAP, kontrolowany przez konsorcjum W3C (ang. *World Wide Web Consortium*), jest oparty na standardzie XML. Składa się z trzech części:

- Otoczki (ang. *envelope*), która definiuje reguły opisywania wiadomości i ich przetwarzania.
- Zestawu reguł kodowania (ang. encoding rules) struktur danych definiowanych w aplikacjach.
- Konwencji (ang. convention) reprezentowania zdalnych wywołań procedur i ich odpowiedzi.

Jeśli myślisz, że ręczne tworzenie usług Web Service, które będą zgodne ze standardem SOAP, jest skomplikowane i żmudne, to masz całkowitą rację. Na szczęście w trakcie tworzenia usług Web Service w postaci projektu ASP.NET w środowisku Visual Studio nie trzeba wpisywać kodu. Jeśli jednak ktoś chce dowiedzieć się więcej o szczegółach specyfikacji standardu SOAP, może zapoznać się z nimi na stronie *http://www.w3.org/TR/SOAP/*.

WSDL

Standard WSDL używany jest do tworzenia dokumentów, które opisują metody obsługiwane przez usługę Web Service, argumenty przyjmowane przez te metody oraz zwracane przez nie wartości. Innymi słowy dokument WSDL przekazuje programowi lub programiście wszystkie niezbędne informacje, potrzebne do skorzystania z usługi.

Ponieważ dokumenty WSDL odczytywane są zwykle przez komputery, a nie przez ludzi, muszą być formułowane w sposób, który pozwoli na ich przetwarzanie przez oprogramowanie. Dlatego też dokumenty WSDL tworzone są w oparciu o język XML, który służy do opisu formatu innych dokumentów (w tym także i XML). Przy tworzeniu usługi Web Service za pomocą projektu ASP.NET w środowisku Visual Studio nie trzeba się martwić o tworzenie dokumentu WSDL. Zajmuje się tym środowisko Visual Studio w momencie uruchamiania projektu ASP.NET z usługą Web Service. Więcej informacji o standardzie WSDL można znaleźć na stronie *http://www.w3.org/TR/WSDL*.

Jeśli będziesz chciał utworzyć program VB .NET, który używa usług Web Service, ale ich nie udostępnia, wtedy środowisko Visual Studio utworzy usługę proxy w oparciu o dokument WSDL dostarczony przez usługę Web Service. Klasa proxy zawiera kod, który pozwala na użycie metod usług Web Service w kodzie języka Visual Basic. Szczegółowy opis znajduje się w rozdziale 6. "Używanie usług Web Services".

UDDI

Jeśli w lesie upadło drzewo, ale nikt tego nie widział, to czy rzeczywiście upadło? Ta sama zagadka mogłaby odnosić się do usług Web Service. W tym przypadku istnieje jednak standard UDDI, który przekazuje informacje o istniejących usługach.

Jeśli na przykład dysponujesz usługą Web Service, której mogą używać inne osoby, możesz skorzystać z UDDI, aby umieścić usługę na liście. Potencjalni użytkownicy usług mogą ją tam znaleźć. W podobny sposób i Ty możesz znaleźć potrzebną usługę.

W przypadku standardów SOAP i WSDL środowisko .NET sprawia, że wystarczy zrozumieć tylko ogólny sens ich działania. Z kolei standard UDDI wymaga działania programisty już na wczesnych etapach pracy.

Aby znaleźć usługę Web Service:

1. Na stronie startowej środowiska Visual Studio na zakładce *Online Resources* kliknij odnośnik *XML Web Services* (rysunek 2.1).

Pojawi się zakładka Find a Service.

- 2. Upewnij się, że zaznaczony jest przycisk opcji UDDI Production Environment.
- **3.** Wybierz kategorię z listy rozwijanej *Category*, na przykład *Financial*.
- **4.** W polu tekstowym *Search For* wpisz szukane pojęcie lub zostaw pole puste, jeśli chcesz zobaczyć wszystkie dostępne w danej kategorii usługi.
- 5. Kliknij przycisk Go.

Zostanie wyświetlona lista dostępnych serwisów *Web Services* wraz z ich opisami.

Wskazówki

- Więcej informacji o stronie startowej środowiska Visual Studio znajduje się w rozdziale 1.
- Aby uzyskać dostęp do większej ilości opcji, które kontrolują szukanie usług, kliknij przycisk Advanced.
- Aby zarejestrować własną usługę Web Service za pomocą UDDI, wybierz zakładkę *Register a Service*, a następnie kliknij odnośnik *Register Your XML Web Service Today*.
- Innym sposobem wyszukania usługi Web Service jest użycie Internet Explorera i odwołanie się do strony http://uddi. microsoft.com/visualstudio (rysunek 2.2)



Rysunek 2.1. Strona XML Web Service służy do odnajdywania istniejących usług Web Service, a także do rejestrowania nowych usług, aby inni użytkownicy mogli z nich korzystać

Edit View Favorites	as Icols Help	R.
Back • 🕥 • 💌	📓 🕼 🔎 Search 👷 Favorites 🜒 Media 🚱 🔗 چ	
🕬 🗃 http://uddi.micros	soft.com/visualstudio/	🛩 🛃 Go 🛛 Links
loosolt		
kuaai		
tool searches for services	s and providers in this UDDI Services site with WSDL described services. Search by name or brown	se through available
egorization schemas.		
Service name:	Search	
Provider name:	Search	
Select a categoriza	ation scheme:	
ntis-gov:sic:1987	uton scheme.	
	1-1	
unspsc-org:unspsc:3-		
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowe	eb:2000	
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowe ntis-gov:naics:1997	eb:2000	
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowe ntis-gov:naics:1997 VS Web Service Sear	eb:2000 Inch Categorization	
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowe ntis-gov:naics:1997 VS Web Service Sear ntis-gov:naics:2002	eb:2000 irch Categorization	
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowe ntis-gov:naics:1997 VS Web Service Sear ntis-gov:naics:2002 unspsc-org:unspsc:vt	eb:2000 wrch Categorization :6:0501	
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowents- nts-gov:naics:1997 VS Web Service Sear ntis-gov:naics:2002 unspsc-org:unspsc:vt ubr-uddi-org:iso-ch:3	eb:2000 crdb Categorization 6.0501 516-2003	
unspsc-org:unspsc:3- unspsc-org:unspsc microsoft-com:geowentis-gov:naics:1997 VS Web Service Sear ntis-gov:naics:2002 unspsc-org:unspsc:vd; ubr-uddi-org:iso-ch:3	46:2000 /6.6551 3136-2003	

Rysunek 2.2. Strona http://uddi.microsoft.com/ visualstudio umożliwiająca wyszukanie usług Web Service



Rysunek 2.3. *Aby utworzyć nowy projekt Web Service, wybierz opcję ASP.NET Web Service z panelu Templates*

Tworzenie usługi Web Service

Przy użyciu języka Visual Basic .NET niezwykle łatwo utworzyć usługę Web Service, więc zacznijmy od razu.

W rozdziale 1. zostało opisane, że programy opierają się na projektach VB .NET. Tak więc rodzaj tworzonego programu zależy całkowicie od rodzaju projektu, który wybierzesz spośród: *Windows Application, ASP.NET Web Application, ASP.NET Web Service* oraz innych.

Aby rozpocząć tworzenie usługi Web Service, należy utworzyć nowy projekt ASP.NET Web Service.

Aby utworzyć nowy projekt ASP.NET Web Service:

- Otwórz okno dialogowe New Project (rysunek 2.3). Możesz to zrobić wybierając opcję New Project z menu File lub klikając przycisk New Project na stronie startowej.
- **2.** Wybierz opcję *Visual Basic Projects* z panelu *Project Types*.
- **3.** Wybierz jako typ projektu *ASP.NET Web Service* z panelu *Templates*.
- W polu tekstowym *Location*, po adresie URL *http://localhost/*, wpisz nazwę projektu.
- 5. Kliknij przycisk OK.

Zostanie utworzony nowy projekt *ASP.NET Web Service*.

Wskazówki

- Adres URL (ang. Uniform Resource Locator) został uzupełniony w czwartym kroku powyższego zadania przy założeniu, że uruchamiasz i testujesz usługi Web Service na lokalnym serwerze IIS (ang. Internet Information Services). Jeśli korzystasz z odległego komputera macierzystego, wpisz odpowiedni adres URL w polu tekstowym Location.
- Jeśli w trakcie tworzenia projektu pojawią się komunikaty o błędzie, możesz znaleźć rozwiązania problemów w ramce "Komunikaty o błędach: tworzenie aplikacji sieciowych", która znajduje się w rozdziale 1.
- Chociaż pole tekstowe Name jest nieaktywne, to tekst, który wpiszesz w polu tekstowym Location jako uzupełnienie adresu http://localhost, zostanie przypisany jako nazwa projektu i pojawi się w polu tekstowym Name.
- Domyślnie http://localhost wskazuje na katalog Inetpub/wwwroot. Tak więc przykładowy projekt o nazwie 02VBVQS01 zostanie umieszczony w katalogu Inetpub/ wwwroot/02VBVQS01. Możesz jednak użyć panelu administracyjnego Internet Information Services, aby umieścić projekt w innym miejscu i utworzyć wirtualny adres URL. Aby uzyskać dostęp do panelu administracyjnego IIS, należy otworzyć grupę Narzędzia administracyjne (Administrative Tools) w Panelu sterowania (Control Panel) komputera.

Otwórz teraz okno *Solution Explorer* (rysunek 2.4) i sprawdź, jakie pliki zostały dodane do utworzonego projektu. Zauważ, że wśród plików znajduje się moduł Web Service, z domyślną nazwą *Service 1. asmx*. Informacje o tym, jak otworzyć okno *Solution Explorer*, znajdują się w rozdziale 1., w części zatytułowanej "Aby otworzyć okno Solution Explorer".



Rysunek 2.4. W oknie Solution Explorer można zauważyć moduł usługi Web Service o rozszerzeniu .asmx





Po utworzeniu nowego projektu powinien się w nim otworzyć moduł Web Service. Dzieje się tak, o ile nie zostaną zmienione domyślne ustawienia. Jeśli projekt nie jest otwarty, wystarczy dwukrotnie kliknąć jego nazwę w oknie *Solution Explorer*.

Projektant usług Web Service może służyć do dodawania komponentów do modułu klasy w trybie graficznym (rysunek 2.5). O ile projektowanie formularzy Windows w trybie graficznym jest sensowne, a większość formularzy Web posiada istotne elementy graficzne, o tyle projektowanie modułów Web Service w ten sposób przeważnie nie jest zasadne. Dzieje się tak, ponieważ usługi Web Service składają się z samego kodu w postaci metod (lub funkcji). Uzywanie narzędzi, które umożliwiają projektowanie graficzne, nie daje żadnych korzyści. Więcej informacji o używaniu projektantów do przyspieszenia procesu tworzenia aplikacji Windows i Web znajduje się w rozdziale 5. "Formularze Windows" oraz w rozdziale 16. "Aplikacje internetowe ASP.NET".

Kiedy nowy projekt *ASP.NET Web Service* jest już otwarty, należy otworzyć edytor kodu, aby zapoznać się ze szczegółami modułu *Web Service*.

Aby otworzyć edytor kodu:

1. W oknie *Solution Explorer* wybierz moduł *Web Service*, a następnie opcję *Code* z menu *View*.

lub

Kliknij odnośnik *Switch to Code View* w projektancie *Web Service*.

Kiedy analizuje się moduł *Web Service* w edytorze kodu (rysunek 2.6), można zauważyć, że program rozpoczyna się od instrukcji Imports:

Imports System.Web.Services

Service Lasinx. Vu		
Service1	✓ IN (Declarations)	
Imports System.Web.Services		
∃ <webservice(namespace:="<u>http:/</webservice(namespace:="<u>	<pre>//tempuri.org/")> _</pre>	
Public Class Service1		
Inherits System.Web.Servio	ces.WebService	
Heb Services Designer Generat	ted Code	
- WEB SERVICE EXAMPLE		
' The HelloWorld() example	e service returns the string Hello World.	
' To build, uncomment the	following lines then save and build the p	roject.
' To test this web service	e, ensure that the .asmx file is the start	page
' and press F5.		
' <webmethod()> _</webmethod()>		
' Public Function HelloWor	rld() As String	
' Return "Hello World"		
 ' End Function 		
-End Class		

Rysunek 2.6. W edytorze kodu można zauważyć oznaczoną jako komentarz metodę (WebMethod) HelloWorld

Instrukcja ta przesyła do kompilatora informację, że w kodzie mogą być używane klasy oraz elementy klas z przestrzeni nazw System.Web.Services.

Następnie umieszczone są linie kodu, które w rzeczywistości stanowią części jednej instrukcji:

<WebService(Namespace:= _
 "http://tempuri.org")> _
Public Class Service1
Inherits System.Web.Services.WebService

Pierwsza część instrukcji, znacznik <WebService>, informuje środowisko Visual Basic, że kod stanowi usługę Web Service.

W dalszej części instrukcji znajduje się nazwa klasy (Servicel) oraz określenie klasy, z której dziedziczy (System.Web.Services.WebServices). Więcej informacji o dziedziczeniu klas zawiera rozdział 3. "Praca z klasami".

Wskazówka

W następnej linii edytora kodu znajduje się szary znak + (plus), po którym można zobaczyć napis Web Services Designer Generated Code. Po kliknięciu tego znaku kod, który został utworzony automatycznie przez projektanta, rozwinie się i będzie widoczny. Kod ten ułatwia rozmieszczenie komponentów w projektancie usług Web Service.

W dalszej części kodu znajduje się przykładowa usługa Web Service: metoda o nazwie HelloWorld, która zwraca łańcuch tekstowy "Hello World". Na rysunku 2.6 fragment ten jest oznaczony jako komentarz za pomocą pojedynczych apostrofów ('). Po usunięciu symboli komentarza dostępny jest szkielet metody umieszczony w usłudze Web Service.

Kod kończy się instrukcją:

End Class

Kiedy wiadomo już, co znajduje się w module *Web Service* w konfiguracji standardowej, można przystąpić do pracy nad własną usługą. Na początek zmień domyślny adres URI.

Adres Uniform Resource Identifier

Domyślnie każda nowa usługa Web Service w środowisku Visual Studio posiada atrybut *Namespace* w postaci *http://tempuri.org.* Jest to adres URI (ang. *Uniform Resource Identifier*), który powinien wskazywać na lokalizację będącą pod kontrolą programisty. W przeciwieństwie do adresu URL, użytkownik nie musi mieć możliwości dostania się do tej lokalizacji.

Jeśli pozostawisz domyślny adres URI, każde uruchomienie strony testowej usługi Web Service spowoduje, że wyświetli się propozycja zmiany adresu.

Aby zmienić domyślny adres URI:

1. Przypisz adresowi URI w znaczniku WebService nową wartość:

```
<WebService(Namespace:= _
    "http://peachpit.com/webservices")> _
    Public Class Service1
    Inherits System.Web.Services.WebService
```

Następnie dodaj metodę Web do usługi.

Aby dodać metodę Web do usługi:

1. Dodaj znacznik atrybutu <WebMethod>.

Funkcja oznaczona tym znacznikiem będzie metodą Web.

2. Utwórz funkcję.

Funkcja ta jest metodą Web.

Na listingu 2.1 pokazany jest cały kod usługi Web Service wraz z nową metodą Web, która zwraca łańcuch znaków.

Wskazówka

Instrukcja

HelloVQS = "Visual Basic .NET jest super!"

przypisuje łańcuch znaków "Visual Basic .NET jest super!" do funkcji, która staje się wartością zwracaną przez metodę Web.

Listing 2.1. Usługa Web Service z metodą, która zwraca łańcuch znaków



Testowanie usług Web Service

Można łatwo testować usługi Web Service, jak też związane z nimi metody, za pomocą środowiska Visual Studio, które tworzy strony testowe.

Aby przetestować usługę Web Service i jej metody:

1. Uruchom projekt poprzez wybranie opcji *Start* z menu *Debug* lub kliknięcie strzałki *Start*.

Powinna otworzyć się strona testowa (rysunek 2.7).

- Kliknij odnośnik metody Web, który znajduje się na stronie testowej usługi. Otworzy się strona testowa dla konkretnej metody Web (rysunek 2.8).
- 3. Kliknij przycisk Invoke.

Program zwróci stronę w formacie XML, na której znajduje się wartość zwracana przez metodę Web oraz typ danych — w tym przypadku łańcuch znaków String (rysunek 2.9).

🗿 Service1 Web Service - Microsoft Intern	et Explorer			
Plk Edycja Widok Ulubione Narzędzia Por	moc			R.
🕲 Wstecz 🔹 🕑 🔹 🛋 🏠 🔎) Wyszukaj 🔺 Ulubione	e 🜒 Multimedia	08	• »
Adres 🕘 http://localhost/WebService1/Service1.asm	nx	× E	> Przejdż	Łącza »
Service1 The following operations are supported. For Description.	a formal definition, ple	ase review the <u>Se</u>	ervice	
HellovQS				>
Gotowe		S Lokalny intranet		

Rysunek 2.7. Strona testowa, wygenerowana dla usługi Web Service przez środowisko Visual Studio, zawiera odnośniki do metod, które ta usługa udostępnia

Service1 Web Service - Microsoft Internet Explorer	×
Plk Edycja Widok Ulubione Narzędzia Pomoc	H.
🔇 Wstecz 🔹 🕥 - 🖹 🛃 🏠 🔎 Wyszukaj 📩 Ulubione 🜒 Multimedia 🧐 🎯 -	»
kdres 🕘 http://locahost/WebService1/Service1.asmx?op=HelloVQS 🛛 👻 🔁 Przejdź 🛛 Łącza	»
Service1	~
Click <u>here</u> for a complete list of operations.	
HelloVQS	
Test	
To test the operation using the HTTP POST protocol, click the 'Invoke' button. Invoke	
SOAP	~
🖞 Gotowe 🧐 Lokalny intranet	

Rysunek 2.8. *Aby przetestować metodę Web, kliknij przycisk Invoke*

чік Ебусја	Widok Ului	bione Narzędzi	a Pomoc					Al
G Wstecz	• 🕥 •	× 2 6) 🔎 Wyszukaj	🖈 Ulubione	Multimedia	e é	3- 2 2 2	
dres 🖉 http	p://localhost/W	ebService1/Servic	ce1.asmx/HelloVQS				🗸 ラ Przejdź	Łącza
xml v<br <string< th=""><th>ersion="1.0 xmlns="httj</th><th>" encoding=" p://peachp</th><th>utf-8" ?> it.com/webs</th><th>ervices">Vi</th><th>sual Basic .NE</th><th>T jest :</th><th>super!</th><th>12</th></string<>	ersion="1.0 xmlns=" htt j	" encoding=" p://peachp	utf-8" ?> it.com/webs	ervices">Vi	sual Basic .NE	T jest :	super!	12

Rysunek 2.9. Wartość zwracana przez metodę Web wyświetlana jest w formacie XML

Dodawanie metod Web

Usługa Web Service może zawierać wiele metod Web. Żeby robić coś naprawdę pożytecznego, metoda powinna wykonywać coś więcej, niż tylko zwracanie łańcucha znaków. Stworzysz teraz metodę Web, która może przyjmować różne dane wejściowe. Zacznij od dodania metody Web, która oblicza sumę dwóch liczb całkowitych typu long.

Aby dodać metodę Web, która sumuje dwie liczby całkowite:

1. Dodaj znacznik atrybutu <WebMethod>.

Funkcja oznaczona tym znacznikiem będzie metodą Web.

- 2. Utwórz funkcję o nazwie AddTwoNums, która będzie przyjmować jako argumenty dwie liczby całkowite typu long . Zwracaną wartością będzie liczba tego samego typu.
- **3.** Przypisz wynik dodawania do funkcji, jak przedstawiono to na listingu 2.2.

Wskazówka

Przedstawiony kod nie sprawdza, czy użytkownik rzeczywiście wpisał liczby całkowite. Jeśli użytkownik wprowadzi na stronie testowej tekst zamiast liczby, to program zgłosi wyjątek. W rozdziale 6. "Używanie usług Web Service" znajduje się przykład, jak sprawdzać poprawność wprowadzanych przez użytkownika danych. Aby zrozumieć, jak działają wyjątki, należy zapoznać się z rozdziałem 9. "Wyjątki i usuwanie błędów".

Listing 2.2. Metoda, która dodaje dwie liczby całkowite typu long

Ising

Listing

<

Testowanie metody Web

Nową metodę Web można łatwo testować. Wystarczy uruchomić usługę, która zawiera tę metodę.

Aby przetestować metodę, która dodaje dwie liczby:

1. Uruchom projekt poprzez wybranie opcji *Start* z menu *Debug* lub kliknięcie strzałki *Start*.

Otworzy się strona testowa usługi, na której widać obie metody Web (rysunek 2.10).

2. Na stronie testowej kliknij odnośnik do metody o nazwie *AddTwoNums*.

Otworzy się strona testowa metody *AddTwoNums*, na której znajdują się miejsca, gdzie użytkownik może wpisać wartości argumentów przyjmowanych przez tę metodę (rysunek 2.11).

3. Wpisz wartości dla argumentów num0ne oraz numTwo, a następnie kliknij przycisk *Invoke*.

Program wyświetli stronę w formacie XML. Na stronie tej powinna znaleźć się wartość zwracana przez metodę (suma dwóch wpisanych liczb) oraz typ danych (long) (rysunek 2.12).

Podsumowanie

W rozdziale nauczyłeś się:

- opisywać usługi Web Service, a także związane z nimi standardy, takie jak SOAP, WSDL i UDDI;
- tworzyć projekt usługi Web Service;
- pracować z modułem Web Service w edytorze kodu;
- dodawać metodę do usługi Web Service;
- testować usługi i metody za pomocą strony testowej środowiska Visual Studio;
- dodawać i testować metodę Web, która przyjmuje dane wprowadzane przez użytkownika.



Rysunek 2.10. Nowa metoda Web pojawia się w postaci odnośnika na stronie testowej usługi Web Service

Pik Edycja Widok Ulubione Narzędsia Pomoc W Wsterz 	dź tącze »
Water: ● ● ● ● Wyszukaj	di tacze »
Addes I http://bochhot/WebService/JServiceLasmo?op-AddTwoNluns I I I I I I I I I I I I I I I I I I I	dź tącza »
Service1 Click here for a complete list of operations. AddTwoNums Test To test the operation using the HITP POST protocol, click the 'Invoke' button. Parameter Value numOne: [2]	
Click here for a complete list of operations. AddTwoNums Test To test the operation using the HTTP POST protocol, click the 'Invoke' button. Parameter Value numOne: [23]	
Parameter Value numOne: 23	
numOne: 23	
numTwo: 35	
Invoke	
SOAP	
<	~

Rysunek 2.11. Strona testowa udostępnia pole do wpisywania argumentów przez użytkownika

Plik	Edycja	Widok	Ulubione	Narzędzia	Pomoc					11
G	Wstecz	• 🖸	- ×	2 🟠	🔎 Wyszukaj	tubione	Multimedia	0	8.	, »
Adres	E http	://localho	st/WebSer	vice1/Service:	Lasmx/AddTwoN	lums	~	Przej	dźt	acza »
~ ~	:?xml vi :long xr	ersion=" nins="hi	'1.0" en ttp://p	coding="ut eachpit.c	:f-8" ?> :om/webse	rvices">58				~
										N.

Rysunek 2.12. Wynik działania metody Web na argumentach, które wprowadził użytkownik, wyświetla się w formacie XML