

E L S Ő F E J E Z E T

Bevezetés

A .NET-fejlesztők igazán izgalmas időket élnek, a Visual Studio 2010 (VS2010) és a .NET 4.0 keretrendszer meglehetősen sok változást hozott. Ám aggodalomra semmi ok: ez a könyv bevezeti az érdeklődőket az újdonságokba, és röviden áttekinti a fontosabb, a témához kapcsolódó kiadások, például az ASP.NET MVC, a Silverlight és a WCF Data Services újdonságait. Ez a kiadás nagyon sok érdekességet tartalmaz, és az új szolgáltatások (a variancia és a párhuzamosság kivételével) könnyen elsajátíthatók.

Verziók

A Visual Studio 2010 öt fő verziója áll a rendelkezésünkre:

- Express,
- Professional (799 USD),
- Professional MSDN-nel (1199 USD újonnan vagy 799 USD-megújítással),
- Premium MSDN-nel (5469 USD újonnan vagy 2299 USD-megújítással),
- Ultimate MSDN-nel (11924 USD újonnan vagy 3841 USD-megújítással).

Ezek a kiadások az ingyenes Azure számítási időt is biztosítják (16. fejezet).

Valószínűleg a Professional verzió felel majd meg a legtöbb fejlesztő igényeinek, de a lehetséges hiányosságokra rávilágít a Premium és az Ultimate kiadások további funkcionálisát összefoglaló *1.1. táblázat*. A szolgáltatások kimerítő összehasonlítása a következő webhelyen található: www.microsoft.com/visualstudio/en-us/products/2010/default.mspix.

1.1. táblázat. *A bővített kiadások szolgáltatásainak egyszerűsített összehasonlítása*

Szolgáltatás	Premium	Ultimate
Kódlefedettség	X	X
Felhasználói felület tesztelése kódból	X	X
Webes teljesítmény tesztelése és terheléstesztelés		X
Adatbázis-telepítés, -változás, -kezelés, egységtesztelés és tesztdatok generálása	X	X
Statikus kódelemzés	X	X
Kódmetrikák	X	X
Profilok	X	X
Intellitrace (historikus hibakereső)		X
Tesztkezelés		X

Miről is szól a Visual Studio 2010 és a .NET 4.0?

A Visual Studio 2010 és a .NET 4.0 a .NET-fejlesztés következő korszakát alapozza meg, továbbá néhány korábbi hiányosságot is pótol. Ezeket az újdonosságokat négy fő cím alá sorolhatjuk be:

- hatékonyság,
- a meglévő technológiák tökéletesebbé válása,
- kiterjeszhetőség,
- az aktuális trendek befolyásolása.

Tegyünk egy rövid kitérőt ezekben a témakörökben.

Hatékonyság

Az egyik első dolog, amelyet az új Visual Studióval kapcsolatban észre kell vennünk, az új WPF-alapú IDE. Ez az IDE biztosít néhány olyan remek szolgáltatást, amelyek korábban csak az olyan beépülő modulokban

álltak a rendelkezésre, mint az R# és a Refactor (ne feledjük, hogy már létezik az R# Visual Studio 2010 verziója). Az IDE újdonságai közé tartozik a téglalap alakú kijelöléses dobozválasztás, a kódrészletek, az osztálycsontk létrehozása, a híváshierarchia és a gyorskeresés; ezekről a szolgáltatásokról a 2. fejezetben lesz szó.

Néhány nagyszerű nyelvi fejlesztés letisztultabbá teszi a forráskódot, ilyenek például az opcionális és a nevesített paraméterek, a dinamikus funkcionalitás és a variancia változásai; ezeket a 3. fejezetben vizsgáljuk meg. Ezek az újdonságok a COM-fejlesztők munkáját is megkönnyítik, nekik, őszintén szólva, minden segítségre szükségük van.

A meglevő technológiák tökéletesebbé válása

Több .NET-alapú technológia, például az ASP.NET, már régóta rendelkezésre áll, és ebben a kiadásban sem változott nagymértékben. A Microsoft megoldotta az ASP.NET korábbi hiányosságait, és bevezetett néhány hasznos finomhangolási lehetőséget is; ezekkel a 10. fejezetben ismerkedhetünk meg.

2008 vége felé a Microsoft bejelentette, hogy a Visual Studio következő verzióiban megtalálható lesz a népszerű JavaScript-könyvtár, a jQuery. Noha nem kimondottan .NET-újdonság, a jQuery rendkívül hasznos keretrendszer, amelyet egészen biztosan használunk majd a webes alkalmazásainkban. Erre a 12. fejezetben térünk vissza. A 11. fejezetben megvizsgáljuk a Microsoft saját Ajax könyvtárainak a fejlesztéseit, amelyek a klienszkriptekkel megkönnyítik az adathoz kötést.

A Windows Workflow Foundation (WF) és a Windows Communication Foundation (WCF) integrációja ebben a kiadásban még szorosabbá vált. A WF radikális átdolgozáson esett át, és tökéletesebb tervezővel, új tevékenységek bevezetésével, valamint könnyebb testreszabással nyugtázza le a fejlesztőket (lásd a 6. fejezetben). A WCF használata egyszerűbb, új szolgáltatásfelfedezési funkcionalitást biztosít; erről a 7. fejezetben lesz szó.

A WPF remek újdonságokkal bővült, például kifinomultabb tervezővel, multitouchot támogató API-val és Windows 7 tálca támogatásával, ezekkel a 15. fejezetben ismerkedünk meg. A 15. fejezetben továbbá röviden bepillantunk a Silverlight 3-as verziójába is. Annak ellenére, hogy a Silverlight nem .NET 4.0-s technológia, rendkívül fontos kiadás. Ha valaki korábban még soha nem használta a Silverlight technológiát, akkor érdemes elolvasnia a 14. fejezetet, amely egy rövid bevezetést tartalmaz.

Az Entity Framework sok kritikát kapott az első megjelenésekor, a Microsoft-csapat pedig megpróbált a kritikák egy részére reagálni a .NET 4.0-s verziójában. A 8. fejezetben kiderül, hogy sikerrel jártak-e.

Kiterjeszthetőség

A Visual Studio 2010 rugalmas és bővíthető. Az IDE nagy része a WPF segítségével készült, és az MEF (Managed Extensibility Framework – felügyelt bővíthetőségi keretrendszer) felhasználásával testre szabható. Az IDE-testreszabást és a MEF-technológiát a 2. fejezetben vizsgáljuk meg.

Az aktuális trendek befolyásolása

A szoftverek nem légyeres térben készülnek, és néhány trend kétségtelenül rányomta a bélyegét a Visual Studio 2010 alkalmazására és a Microsoft termékvonálára.

Többmagos eltolódás

Különböző fizikai korlátok miatt a processzorgyártók egyre inkább a többmagos processzorok előállítására összpontosítanak. A több szálon futó programok írása nem könnyű feladat, ám aggodalomra nincs ok: a Visual Studio 2010 és a .NET 4.0 fantasztikus párhuzamosítási konstrukciókat és hibakeresési lehetőségeket biztosít; ezekkel az 5. fejezet foglalkozik.

Egységtesztelés és a tesztvezérelt fejlesztés

Az egységtesztelés és a tesztekre épülő fejlesztés egyre népszerűbbé válik a szoftverfejlesztésben. A Visual Studio 2010 több olyan IDE-fejlesztést tartalmaz, amelyek elősegítik ezeket a stratégiákat. Erre a 2. fejezetben térünk vissza. Azok a kapcsolódó újdonságok, amelyek még érdeklődésre tarthatnak számot, az új dinamikus szolgáltatások, a DLR (lásd a 3. fejezetben) és az ASP.NET MVC (lásd a 13. fejezetben). Igaz, hogy az ASP.NET MVC nem .NET 4.0-s technológia, ám fontos közbenső kiadásról van szó.

Számítási felhő

A *számítási felhő* (Cloud Computing) megnyerheti a 2009-es év felkapott kifejezése díját. A költségcsökkentés és az infrastruktúrakezelés egyszerűsítésének egyre népszerűbb módjává növi ki magát. A Windows Azure a Microsoft belépője erre a területre; a technológia képességeit, valamint felhasználási lehetőségeit a 16. fejezet ismerteti. Ha az MSDN-nel vásároljuk meg a Visual Studio 2010-et, akkor ingyenes Azure számítási időt is kapunk hozzá.

Mit gondolnak mások a .NET 4.0-ról?

Olyan cégekkel és fejlesztőkkel készült interjúkat is beletettem a könyvbe, akik ezekkel az új technológiákkal fejleszthettek, így bepillantást nyerhettek a lehetséges problémákba és lehetőségekbe.

Több szakértőt megkérdeztem, hogy mi keltette fel érdeklődésüket a Visual Studio 2010-ben és a .NET 4.0-ban.

Mike Ormond (Microsoft-evangelista)

blogs.msdn.com/mikeormond/

„Hogy mi keltette fel az érdeklődésemet? Többtonnányi termelékenység javítás található a Visual Studio 2010-ben, például a Visual Web Developer kódrészletei. És végre ASP.NET-, HTML- és JavaScript-kódrészletek. A referenciakiemelés és a típusok generálása két további roppant izgalmas újonság csakúgy, mint az Intellisense-fejlesztések (mostantól nem kell emlékeznünk a pontos névre, és a tökéletesített JavaScript-támogatás döbbenetes).

Időről időre belekontárkodom Office-fejlesztésekbe, és a C# használata felé hajlok, így megelégedéssel tölt el, hogy a nyelv felhasználja az opcionális és nevesített paramétereket csakúgy, mint az új dinamikus típust, amely nagyon megkönnyíti a COM-együtműködést. A C# nyelv segítségével készülő Office-fejlesztés mostantól álomszerű lesz.

A webes újonságok szempontjából lenyűgöznek a Web Forms új szolgáltatásai. Sok szolgáltatás apróságnak tűnhet, de a platform kulcsfontosságú fejlesztései között van egy vadonatúj, amely nem más, mint az

ASP.NET MVC. A választás mindig remek dolog, és az ASP.NET most két nagyszerű választási lehetőséget biztosít. Tegyük hozzá ezeket a Microsoft Ajax-fejlesztések, például a kliensoldali sablonok/adatkötések és a jQuery-integráció keverékéhez, és lehengető webes alkalmazási platform lesz az eredmény. Természetesen rendelkezésünkre áll a Web Deployment Tool, valamint a web.config átalakítások, ezekkel automatikusan elkészíthetjük, és a telepítéshez csomagolhatjuk az alkalmazásainkat.”

Eric Nelson (Microsoft-evangelista)

blogs.msdn.com/ericnel/

„Számomra a .NET 4.0 fordulópontot jelent az adatbázis-alkalmazások jövőbeli fejlesztésében. A .NET 4.0 révén nagyszerű ORM- (Object Relational Mapping - objektumrelációs leképezés) technológia áll a rendelkezésünkre az ADO.NET Entity Framework 4.0-ban, és ez nagymértékben egyszerűsíti az RDBMS használatával kapcsolatos erőfeszítéseket a .NET-alkalmazásokban. Igazán lenyűgözött, hogy a termékfejlesztő csoport odafigyelt a fejlesztői közösségnek az Entity Framework kezdeti kiadására vonatkozó visszajelzéseire, és a 4. verzióban kiemelkedő új funkcionális szállítást szállítanak.”

Craig Murphy (MVP és fejlesztői közösségszervező)

www.craigmurphy.com/blog/

„Időről időre a számítástechnikai szektor paradigmaváltáson megy keresztül. A programozók sok olyan váltást megélték már, amelyek közül több a tervezési módszerek változásainak vagy a megközelítési módok módosulásának az eredménye volt. A procedurális programozásról az objektumorientált programozásra való áttérés a paradigmaváltást példázza. Manapság, amikor a funkcionális programozás a fővonalbeli Visual Studio-termék részévé válik, a programozók újabb váltásra készülhetnek fel.

A hardver fejlődése azt jelentette, hogy a belépőszintű hordozható számítógépek két- vagy többmagos processzorral működnek. Ez a hardvertechnológia-váltás pozitív következményekkel jár. A váltás a programozók gondolkodásában is változásokhoz vezet, nemcsak az alkalmazásaikról, de az alkalmazástervezésről alkotott elképzeléseikre is hatással van. Már nem

jelent kihívást egyetlen processzort többszálás munkára bírni. Két vagy több maggal rendelkező processzorral dolgozni, amelynek magjai egységszerű munka elvégzésére képesek – ez az új kihívás.

Olyan alkalmazások létrehozása, amelyek a processzor meghatározott magjait fogják munkára, a holtpontok, a versenyfeltételek, a skálázhatóság, a kizárás és a determinizmus kihívásai elé állítják a programozókat. Mindezekon felül az olyan alkalmazások, amelyek több magon osztják szét a terhelést, gyorsabban reagálnak, és hatékonyabbak.

A Parallel Extensions és a többmagos technológia megismerése egy másik paradigmaváltást jelent a programozók számára. Ez a váltás azonban nem kizárólag a programozókat érinti: a mélyre nyúló pozitív hatások az alkalmazási teljesítmény és a felhasználói élmény területére mutatnak. Egy helyesen megtervezett alkalmazás, amely kihasználja a többmagos technológiát, a végfelhasználóban az érzékenyebb reagálás érzését keltheti: ebben a helyzetben mindenki nyer.

A Visual Studio és a .NET 4.0 szilárd keretrendszert biztosít olyan .NET-alkalmazások készítéséhez, amelyek a többmagos technológiát veszik célba. 2007-ben a Microsoft a Parallel Extensionst bocsátotta a programozók rendelkezésére, amely támogatta a Paralell LINQ-et (PLINQ) és a Task Parallel Library segítségével a feladat párhuzamosítását. A programozóknak meg kell tanulniuk a .NET 4.0-t; ez a könyv kiváló alapokat biztosít annak az elsajátításához, hogy a Visual Studio 2010 segítségével hogyan lehet a .NET 4.0 szolgáltatáskészletét, beleértve a Parallel Extensionst, kihasználni.”

Phil Whinstanley (ASP.NET-MVP és szerző)

weblogs.asp.net/Plip/

„Az ASP.NET 4.0 az utólagos előrelátás előnyeivel rendelkezik. A Web Forms és az MVC renderelési motorjainak integrálásával a fejlesztők szabadon kifejezhetik magukat, és eközben kihasználhatják az alapul szolgáló ASP.NET platform előnyeit is. Segédkezet nyújt a fejlesztőknek a sokoldalú és hatékony ASP.NET-alkalmazások készítéséhez, a Visual Studio 2010 pedig továbbfejlesztette a megközelítését, és azokra a területekre összpontosít, amelyek a fejlesztők számára fontosak: a sebességre, a hatékonyságra és a könnyű használhatóságra.”

Dave Sussman (MVP és szerző)

www.ipona.com

„Az egyik dolog, amelyet a legjobban szeretek a .NET és a Visual Studio programokban, az a fejlesztői csapatuk és a nyíltságuk, amellyel a fejlesztői közösség felé fordulnak. Természetes, hogy bizonyos dolgokat titokban tartanak, és privát béta-kiadásokat készítenek, de munkájuk nagy része nyilvános, nyitottak a visszajelzésekre, és így a terméket a fejlesztők igényeinek megfelelően tökéletesítik.

Sok szó esik az MVC-ről, erről a nagyszerű platformról, amely új felhasználókat vezetett be, de engem a Web Forms, az MVC érettebb „társának” a fejlesztései érdekelnek igazán. A Web Formsot a Microsoft továbbra is támogatja, és aktívan fejleszti, és noha a változások nem annyira radikálisak, mint az MVC esetében, mégis messzire mutatnak. A .NET 4.0-s verzióban több ASP.NET-vezérlőelem tiszta HTML-t állít elő (a Menü ennek remek példája), és a kliensoldali azonosítók kézben tartásával egy sokkal megfelelőbb platform áll a rendelkezésre az Ajax-alapú webhelyek elkészítéséhez. Ha ezt olyan szolgáltatásokkal párosítjuk, mint a kliensablonok és a jQuery-támogatás, a platform új életre kel; roppant vonzó számomra az a gondolat, hogy Web Form-oldalakat készíthetünk anélkül, hogy sokat kellene küzdenünk a szabványoknak megfelelő, CSS-szel könnyedén formázható HTML-ért.

A Visual Studióban a legizgalmasabb új szolgáltatás a WPF által biztosított lehetőség. A WPF segítségével készült kódszerkesztővel nemcsak a felhasználói felület frissült, és kapott vonzóbb megjelenést, de nagyszerű lehetőségek nyíltak meg az eszközök számára is, hogy tökéletesíteni lehessen a kódszerkesztési élményeket. Úgy gondolom, most a lehetőségekben a jéghegynek csak a csúcsát látjuk – a szerkesztési támogatás terén számtalan további lehetőség áll a rendelkezésre.”

Matt Lacey (Devevening.co.uk szervező)

blog.mrlacey.co.uk

www.devevening.co.uk

„Sok fejlesztővel kerülök kapcsolatba, és legtöbbjüknek sem ideje, sem hajlandósága nincs arra, hogy új dolgokat tanuljon. Ezért mindig lázba hoznak azok a szolgáltatások, amelyek minimális tanulást és magyarázatot

igényelnek. A nevesített és az opcionális paraméterek két ilyen szolgáltatást képviselnek. Könnyedén megmagyarázhatók, és meg is érthetők. Továbbá drámai módon csökkentik a bizonyos feladatok számára szükséges kód mennyiségét. Minél kevesebb kód szükséges egy feladat végrehajtásához, annál kevesebb programhiba fordulhat elő, és annál több idő marad új szolgáltatások tesztelésére és megvalósítására. Mindenki nyer ezzel.

A másik szolgáltatás, amelyet izgatottan várok, az ASP.NET MVC. Ez két ok miatt különösen fontos. Először is kérdéseket vet fel a tesztelhető kód fontosságával kapcsolatban, ez pedig azt eredményezi, hogy egyre több fejlesztő alkalmaz majd tesztelőeszközöket és módszereket a kód tökéletesítéséhez. Másodsorban a fejlesztőkben így felmerül az a kérdés, hogy erre miért van szükség, és mi a probléma az ASP.NET Web Forms-szal. Az ASP.NET MVC-nél kulcsfontosságú különbség az, hogy HTTP ismerete nélkül sokkal nehezebb fejleszteni. Attól függetlenül, hogy MVC-t vagy Web Formsot használunk-e, előnyösebb, ha minél több webes fejlesztő rendelkezik a lehető legmélyebb HTTP-tudással.

Alex Mackey (a könyv szerzője, MVP)

simpleisbest.co.uk

„Számomra ebben a verzióban a legkiválóbb változások azok az egyszerű újdonságok, amelyet a fejlesztők mindennap használnak majd. Olyan változásokra gondolok, mint az új `String.IsNullOrEmpty()` vagy az `Enum.TryParse()`. Magam is webes fejlesztő vagyok, és látom, hogy az ASP.NET megoldást talált néhány régóta ismert bosszúságra, és a jQuery integrációja kifejezetten okos lépés volt. Bár a többszálúsággal még túl sokat nem kellett foglalkoznom, ám nagyon tetszik az új Task modell, és úgy vélem, sokkal intuitívabban lehet vele dolgozni. Az olyan változásokat, amelynek révén alkalmazásunk anélkül fut gyorsabban, hogy nekünk bármit tennünk kellene, lelkesen üdvözlöm, tehát örömmel látom a GC- és a threadpoolos fejlesztéseket ebben a kiadásban. Tetszik az az új irány, amerre a biztonsági modell halad, ugyanis úgy vélem, hogy a korábbi rendszer túlságosan összetett volt. Számomra az egyik legnagyobb meglepetést az Entity Framework nagyszerűsége okozta, de ennél nem mondok többet, mert a könyv további részében kizárólag a nagyszerű új szolgáltatásokról lesz majd szó.”

A jövő irányzatai

A szoftverfejlesztők számára nagy kockázat olyan technológiákat megtanulni és támogatni, amelyek gyorsan elavulnak. Természetesen senki nem tudhatja előre, hogy mit hoz a jövő (még akkor sem, ha valaki ezt állítja), ám úgy vélem, hogy érdemes megvizsgálni az aktuális álláshirdetési trendeket, amikor azon gondolkodunk, hogy milyen területekre összpontosítsunk.

Beszéltem Jonathan Keennel, aki egy angliai fejadászügynökség, a Cognitive Group (cognitive-group.co.uk) vezetője, és a .NET-fejlesztésben tapasztalható fejlődési irányvonalakról faggattam. Keen a következőket osztotta meg velem:

- A vállalatok általában megnyirbálják az új projekteket, tehát kevesebb fejlesztői feladatkör vár gazdára.
- A könyv írásának idején (2009 októberében) Londonban a legnépszerűbb és legjobban fizetett szakmai ismeretet a Sharepoint, a Dynamics AX és a Dynamics CRM képviselte.
- Több szerep összpontosít a létező alkalmazások, például a Sharepoint és a Dynamics programcsomag integrációjára.
- Az állásokkal kapcsolatos versengésben nagyobb hangsúlyt kap az, hogy a munkavállalók igyekeznek minél inkább kitűnni. Bizonyítanunk kell a fejlesztés iránti elkötelezettségünket. Részt kell vennünk konferenciákon, a felhasználói csoportokban, és blogot kell vezetni. (És a .NET új kiadásait ismertető Apress könyveket is meg kell vásárolnunk! – A szerző.)
- A C#, az ASP.NET, az SQL Server továbbra is olyan kulcsfontosságú szakértelmet képvisel, amelyet a cégek keresnek.
- Némi érdeklődés mutatkozik az „Agile”- és a SCRUM-jártasság iránt, ám a TDD iránt már egyre csekélyebb.
- A WCF- és a WPF-ismeretek egyre keresettebbek egy fejlesztő önéletrajzában.
- A közsférában rengeteg projekt még mindig a .NET 1.1-re épül.
- Az asztali alkalmazás fejlesztéséhez képest a webes programozás egyre nagyobb hangsúlyt kap.

- A Silverlight és az MVC területén egyelőre nincs hiány. (Több programozó, akivel beszéltem, egyetértett velem abban, hogy a Silverlight az aktuális felkapott terület. – A szerző.)
- Az MVC a fejlesztők megnyeréséhez szükséges csálétek.
- Az olyan technológiák, mint a BizTalk, drasztikusan veszítettek népszerűségükből.

Saját szubjektív véleményem

A WPF határozottan fejlődő területnek tűnik. A .NET-szerződések legnagyobb részét pillanatnyilag a Silverlight teszi ki. Saját álláspiaci tapasztalataim azt mutatják, hogy az ASP.NET népszerűsége töretlen, és a JQuery kötelező szakismeretté növi ki magát. Érdekes lesz megvárni, hogy az ASP.NET MVC divatos technológiává válik, noha gyanítom, hogy több cég már túl sokat fektetett az ASP.NET-be, és az ASP.NET MVC túl nagy gondolkodásmódbeli változást okoz a fejlesztőknek.

Az új párhuzamosítási funkcionalitás lebilincselő, ám az üzleti alkalmazások terén kevésbé fogják kellőképpen kihasználni. A szálkezelésben és a párhuzamosításban járatos fejlesztők mindig magasabb fizetést harcolhatnak ki maguknak, ezért ezen mindenképpen érdemes elgondolkozni.

Zárásként

Minden további szószaporítás helyett térjünk át a következő fejezetre, az új IDE-funkcionalításra.

```
static void Main(string[] args)
{
    dynamic alexBook;
    alexBook.Run();
}
```