

Microsoft

# .NET Framework 1.1

**.NET är kärnan i Microsofts framtida operativsystem och webbplattform. .NET 1.0 lanserades våren 2002 och i april 2003 kom så version 1.1 . Microsofts utvecklingsverktyg för .NET är .NET Framework SDK och Visual Studio .NET. Borland lanserar först C#Builder för .NET och därefter Delphi och C++Builder för .NET.**

.NET är en ny plattform med tonvikt på kommunikation och samverkan mellan applikationer inom samma enhet (t ex en PC, server, webbserver), mellan applikationer i olika enheter (PC, servrar, mobila enheter, etc) i lokala nätverk och mellan applikationer och tjänster över Internet. .NET innehåller ett modernt komponentbaserat klassbibliotek, vilket gör att .NET-applikationer inte behöver prata direkt med de äldre programmeringsgränssnitten för Windows utan kan skrivas direkt för den portabla .NET-plattformen.

.NET-applikationer har stöd för "remoting" för att skapa täta, snabba, binära kommunikationskanaler eller "lösare förbindelser" mellan webbtjänster och konsumenter/klienter via standardprotokollet SOAP. Data på SOAP-format specificeras med XML och transporteras vanligen via HTTP, dvs samma protokoll som används av webbservrar och webbläsare.

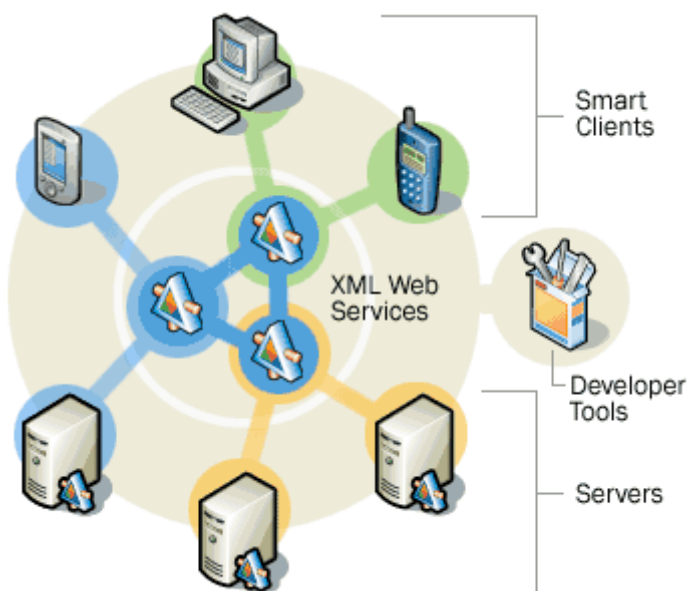
.NET-applikationer kan vara konsolapplikationer, Windows-applikationer med grafiska gränssnitt, serverbaserade webbapplikationer uppbyggda med webbformulär (web forms) för presentation i vanliga webbläsare, serverbaserade webbtjänster, applikationer för mobila enheter, m m.

.NET-applikationer kan idag köras under Windows 98, Me, NT, 2000, XP, nya Windows Server 2003, Pocket PC 2002, Windows CE .NET, etc.

## .NET Framework

I .NET Framework finns den kärna av objektorienterade programbibliotek, som är fundamentet för alla .NET-baserade applikationer. .NET

Framework finns förinstallerat i Microsofts senaste



operativsystem, men kan också laddas ner (drygt 25 MB) och installeras under andra Windows-versioner (dock ej Windows 95). Alla .NET-baserade applikationer, oavsett vilket programspråk de är skrivna i, kompileras till en gemensam processoroberoende kod, "MSIL", som är De använder .NET Framework, som också finns i en kompaktversion för XP Embedded och CE .NET.

.NET Framework har två huvudbeståndsdelar – "Common Language Runtime", CLR, som är kärnan i .NET Framework, samt .NET Framework Class Library.

## CLR – Common Language Runtime

CLR sköter "markettjänsten" i .NET-applikationer, som minneshantering (med automatisk skräphantering – tack för det!), trådhantering, kompilering (!), fjärrkoppling/remoting, garanterar strikt typsäkerhet (via "Common Type System", CTS), m m. CLR tilldelar koden strikta åtkomsträttigheter då det gäller filhantering, m m, vilket

avsevärt höjer säkerheten i applikationer.

Applikationer som direkt arbetar direkt med CLR sägs vara skyddade ("managed"). Till skillnad från Java är skyddad kod dock inte interpreterad utan kompileras av en just-in-time-kompilator (JIT) till maskinkod. Beroende på plattform och processor används anpassade JIT-kompilatorer som översätter till rätt maskinkod.

Kod som inte bygger på CLR sägs vara oskyddad ("unmanaged"). Oskyddad kod används då man behöver utnyttja andra, äldre standarder, som COM-komponenter/ActiveX:er och DLL:er. .NET-applikationer kan därför använda COM-objekt, ActiveX:er och DLL:er, men eftersom sådan kod inte är .NET-baserad, så betraktar .NET den som oskyddad. Man kan också exportera .NET-baserade programbibliotek (sk assemblies) som COM-objekt, så att de kan användas av vanliga Windows-applikationer.

## Gemensamt klassbibliotek

Klassbiblioteket i .NET Framework innehåller allt från utveckling av kommandoradsapplikationer, grafiska .NET-applikationer, webbformulär och XML-baserade webbtjänster m m.

Man kan skriva applikationer, klasser och komponenter i valfritt .NET-anpassat programspråk (Basic, C++, C#, Delphi, etc.), och sedan återanvända denna kod tillsammans med andra .NET-anpassade språk. Här skiljer sig .NET från Java 2-plattformen, där den senare endast accepterar 1 programspråk, nämligen Java.

Delphi för .NET kommer också ha VCL för .NET, ett eget klassbibliotek som underlättar portning av Windows-applikationer till .NET:

## ASP .NET

ASP .NET är en miljö för att kunna arbeta med .NET-baserade webbapplikationer. Med ASP .NET kan man utveckla serverbaserade webbapplikationer (med användargränssnitt byggda med Web Forms) för presentation i vanliga webbläsare. Man kan också utveckla SOAP/XML-baserade webbtjänster och distribuerade objekt baserade på skyddad kod. Som webbserver används då Microsoft Internet Information Server, IIS. Det kommer också alternativa tekniker för att skapa .NET-baserade webbapplikationer som IntraWeb för .NET från ATozed.

## Web Forms

Med Web Forms kan du snabbt och visuellt designa användargränssnittet hos webbapplikationer, som sedan presenteras i vanliga webbläsare som Internet Explorer.

I utvecklingsmiljön designar du webbformulär på samma

sätt som vanliga Windows-formulär. Lägg ut kontroller (Web Controls), knyt händelsekod, etc. Du behöver varken behärska HTML eller skriptspråk för att skapa webbapplikationer, men om du vill kan du integrera egen HTML- och skriptkod.

ASP .NET-baserade applikationer är kompillerade och därför väsentligt snabbare än tidigare skriptbaserade ASP-applikationer.

Exempel på design av ett webbformulär i en ASP .NET-applikation med användning av Visual Studio .NET

## Fjärrkopplingar

.NET har stöd för snabb, binär kommunikation mellan objekt t ex via TCP, eller lösare och mer portabla kopplingar via http-baserad SOAP/XML.

## Programspråk för .NET

Microsoft har implementerat fyra fullvärdiga .NET-språk – **Visual Basic .NET**, **Visual C++ .NET** tillsammans med nya **Visual C# .NET** ("C-sharp") och **Visual J#** ("J-sharp"). Man bör vara medveten om att .NET-versionerna skiljer sig en hel del från tidigare språkversioner, i synnerhet då det gäller Visual

Basic och Visual C++. I Visual Studio .NET finns dock wizards för programuppgradering, som underlättar portning till .NET.

Intressant är nya **C#**, som har valt samma riktning som Java, dvs att utgå från C/C++ syntax, rensa upp gammalt C-"skräp", förenkla (inga multipla arv, istället gränssnitt + klasser), typsäkra (inga undantag från typsäkerhet), samt komponentbaseras. **C#** är det språk den gemensamma runtime-kärnan CLR och klassbiblioteket är utvecklat i.

Microsoft har också ett eget javaliknande programspråk för .NET, **Visual J#**, men som är skräddarsytt för .NET-plattformen och speciellt riktat till tidigare Visual J++ användare.

Kommandoradskompiatorer för bl a C# finns i **.NET Framework SDK**, som kan hämtas fritt från Microsoft.

Från Borland finns **C#Builder** för .NET, som är Borlands utvecklingsverktyg för C#, och på gång är också **Delphi for .NET** och **C++Builder for .NET**. En betaversion av **Delphi for .NET**-kompilatorn och komponentbiblioteket **VCL .NET** får du tillsammans med **Delphi 7**.

