

آشنایی با

MICROSOFT .NET FRAMEWORK

تألیف: احسان خیابانی فرد

بسم الله الرحمن الرحيم

پیامبر اکرم (ص):

اگر مومنی، پیش از مرگ، برکه ای از علم از خود بر جای گذاشته باشد، همین یک برک روز قیامت میان او و آتش جهنم مانع می شود و خداوند در ازای هر حرفی که بر آن برکه نوشته شده باشد، شری هفت برابر بزرگتر از وسعت کل دنیا به او عطا می کند.

پیشگفتار :

عصر حاضر ، عصر فناوری و اطلاعات است و برنامه نویسی و تولید نرم افزار های کاربردی یک صنعت سودآور در این عصر است.

لزوم آشنایی با تکنولوژی های مرتبط با این صنعت برای برنامه نویسان ، مرا بر آن داشت که این مقاله را تهیه نمایم.

در تهیه این مقاله بنده با استفاده از تمامی تجربیات عملی خودم در زمینه تولید نرم افزار های تجاری با C# و مطالعه دقیق و چندباره مقالات و کتابهایی در زمینه NET. تمامی سعی خود را برای علمی و دقیق بودن این مقاله نموده ام.

در پایان از ارائه پیشنهادات و انتقادات و همچنین ذکر خطاهای احتمالی این مقاله پیشاپیش تشکر و قدردانی می نمایم.

به امید آنکه این خدمت مقبول جامعه انفورماتیکی کشور قرار گیرد.

احسان خیابانی فرد .

تقدیم به استاد عزیزم مهندس محمد مادی اردکانی

فهرست

صفحه	عنوان
۲	۱. مقدمه
۳	۲. .NET Framework چیست؟
۳	۳. Common Language Runtime چیست؟
۳	۳.۱ تعریف CLR
۴	۳.۲ CLR و فواید آن
۴	۳.۲.۱ امنیت
۴	۳.۲.۲ CTS
۴	۳.۲.۳ مستقل از پلتفرم
۴	۴. ASP.NET چیست؟
۵	۵. ADO.NET چیست؟
۶	۶. Windows Presentation Foundation چیست؟
۷	۶.۱ WPF و DirectX
۸	۷. Windows Communications Foundation چیست؟
۹	۷.۱ ویژگی های WCF
۹	۷.۱.۱ گرایش به سرویس
۹	۷.۱.۲ Interoperability
۹	۷.۱.۳ الگوهای پیام چندگانه
۱۰	۷.۱.۴ Service Metadata
۱۰	۷.۱.۵ Data Contracts
۱۰	۷.۱.۶ امنیت
۱۰	۷.۱.۷ Transport ها و Encoding های چندگانه
۱۱	۷.۱.۸ پیام های قابل اعتماد و به صف شده
۱۱	۷.۱.۹ پیام های مدت دار
۱۱	۷.۱.۱۰ تراکنش ها
۱۱	۷.۱.۱۱ Ajax و ساپورت Rest
۱۱	۷.۱.۱۲ توسعه پذیری
۱۱	۸. WF (Windows Workflow Foundation) چیست؟
۱۲	۹. Windows CardSpace چیست؟
۱۲	۱۰. LINQ چیست؟
۱۳	۱۱. ADO.NET Entity Framework چیست؟
۱۳	۱۲. معرفی Task Parallel Library و Parallel LINQ
۱۴	۱۲.۱ مثال عملی Task Parallel Library
۱۵	۱۲.۲ مثال عملی Parallel LINQ
۱۶	۱۳ خلاصه
۱۶	منابع

فهرست شکلها

صفحه	عنوان
۳	۲.۱ قالب کلی .NER Framework
۴	۳.۱ قالب کلی CLR
۱۴	۱۲.۱ فضای نام کتابخانه TPL
۱۴	۱۲.۲ اجرای یک حلقه بدون پردازش موازی
۱۴	۱۲.۳ خروجی یک حلقه بدون پردازش موازی
۱۵	۱۲.۴ اجرای یک حلقه با TPL
۱۵	۱۲.۵ اجرای حلقه ساده تر با TPL
۱۵	۱۲.۶ خروجی یک حلقه با TPL
۱۵	۱۲.۷ اجرای یک کوثری بدون پردازش موازی
۱۵	۱۲.۸ اجرای یک کوثری با پردازش موازی

۱. مقدمه:

سالها کار برنامه نویسی کاری پر زحمت ، هزینه بر و زمانبری بود ، تولید یک نرم افزار کاربردی زمان زیادی را صرف می کرد و توان زیادی را از برنامه نویسان می برد و نهایتا لازمه کارایی آن تست های فراوانی بود که صحت عملکرد آن را تا حدودی تضمین کند.

بعلاوه برنامه نویسان مجبور بودن تا تمام اقدامات گرافیکی و طراحی توابع و ... را خود انجام بدهند به عنوان مثال زمان بسیاری صرف طراحی یک منوی ساده یا یک دکمه صرف می شد و در نهایت نیز نرم افزاری تولید می شد که ممکن بود در یک کامپیوتر دیگر بدرستی اجرا نشود.

مایکروسافت همواره به دنبال تسهیل کار برنامه نویسی بوده ، با ارائه ویژوال استدیو ۲۰۰۰ تا حدودی کار برنامه نویسی تسهیل پیدا کرد اما این پایان کار نبود ، مایکروسافت با الگوگیری از ماشین مجازی جاوا ، چارچوب .NET را ارائه کرد که شامل تمامی ابزارهای لازم ، زبان های برنامه نویسی و امکانات بسیار دیگری است و قابلیت این را دارد که تمامی نرم افزار های تولید شده توسط این چارچوب را در تمامی کامپیوترهای که .NET بروی آنها نصب شده است را با صحت کامل اجرا کند.

در ادامه به بررسی جزئیات .NET Framework و ویژگی های آن خواهیم پرداخت.

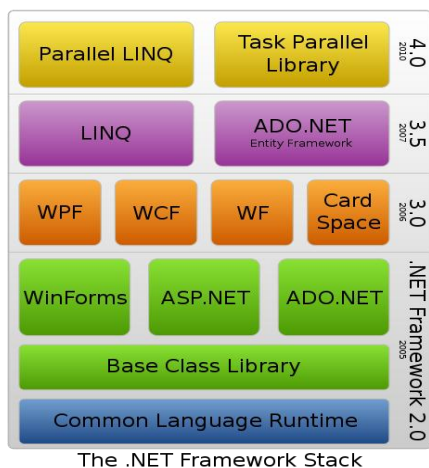
۲. NET Framework چیست ؟

چارچوب .NET یک فناوری نرم افزاری است که بروی بسیاری از نسخه های سیستم عامل مایکروسافت ویندوز قابل اجرا است و به سیستم عامل های دیگر از جمله لینوکس و مکینتاش نیز وارد شده است .

این چارچوب شامل زبان های برنامه نویسی است که سی شارپ و ویژوال بیسیک ، از مهمترین آنها می باشد.

مجموعه ای از کتابخانه های بسیار غنی جهت کمک به سهولت توسعه نرم افزار در این چارچوب قرار گرفته اند که در قالب بخشهای عمده ای همچون فناوری های ASP.NET و ADO.NET و بسیاری فناوری های خاص دیگر ارائه می شوند که تعداد آنها در نسخه های اخیر بیشتر همگام با اهداف خاص مایکروسافت شده است.

در حال حاضر (اواخر سال ۲۰۱۱) چهار نسخه از .NET وجود دارد که شکل ۲.۱ قالب کلی .NET را نشان می دهد.



شکل ۲.۱

.NET Framework ۴.۰

.NET Framework ۳.۵

.NET Framework ۳.۰

.NET Framework ۲.۰

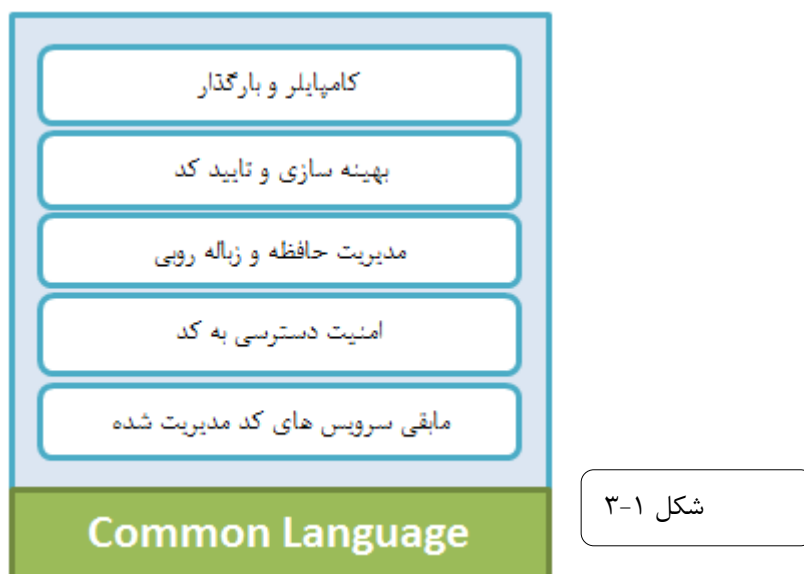
۳. Common Language Runtime (CLR) چیست ؟

۳.۱ تعریف :

سیستم کامپایل برنامه های تحت .NET بدین گونه است که برنامه های نوشته شده توسط .NET ابتدا به زبان میانی مایکروسافت ترجمه شده و سپس توسط CLR اجرا می شوند ، یعنی در واقع CLR برنامه را اجرا می کند که محیطی یکدست را برای ساخت همه جور برنامه با .NET را فراهم می کند و همچنین بخاطر قابلیت های چند زبانه بودن محیط .NET برنامه نویسان C و VB بر راحتی می توانند کدهای نوشته شده توسط دیگری را مورد استفاده قرار دهند و می توان گفت که در محیط .NET بر راحتی و با حداقل تغییرات یک پروژه را با چند زبان نوشت و در نهایت با هم ترکیب کرد.

CLR کدها را در زمان اجرا مدیریت می کند مانند مدیریت حافظه و thread ها و همچنین مدیریت سازگاری ، به همین دلیل به کدهایی که برای اجرا در CLR نوشته شده اند Managed Code و در غیر اینصورت Unmanaged Code می گویند.

شکل ۳-۱ CLR را نشان می دهد.



۳.۲ CLR و فواید آن :

۳.۲.۱ امنیت : هر Managed component درجه ای از مجوز ها را داراست بدین مفهوم که آیا این Component می تواند به فایلی یا رجیستری دسترسی داشته باشد یا خیر و موارد دیگر.

۳.۲.۲ Common Type System (CTS) : کامپایلر های مختلفی که برای زبان های NET طراحی شده اند با CTS مطابقت دارند ، این امر در سازگاری زبان های برنامه نویسی تحت NET با یکدیگر تاثیر بسزایی دارد.^۱

۳.۲.۳ مستقل از پلتفرم (Platform Independence) : زمانی که شما با استفاده از یکی از زبان های تحت NET یک فایل EXE یا DLL تولید می کنید ، این فایل در فرمت Executable نیست و در فرمت MSIL می باشد.

۴ . ASP.NET چیست ؟

امکان دیگری که در دات نت ارائه شده ASP.NET است که امکان طراحی سایت پویا را به کاربران ارائه می دهد ، با کمک ASP.NET با آسودگی می توان دست به طراحی سایت ها و پرتال های عظیم در حد یک بانک یا وزارتخانه زد ، همه جوری امکانی پیش بینی شده و فقط باید شما منطق سایت خود را پیاده سازی نمایید .

در ابتدا ممکن است کار با ASP.NET کمی مشکل به نظر برسد اما اگر با نرم افزار Dreamweaver و سایت های ایستا آشنایی داشته باشید از کار با ASP.NET لذت خواهید برد.

^۱ منظور از سازگاری این است که با استفاده از هر زبان برنامه نویسی تحت NET می توانیم به سایر component های نوشته شده توسط این زبان ها دسترسی پیدا کنیم.

ASP.NET به عنوان یک فناوری سمت سرور طراحی شده و تمام کدهای ASP.NET روی سرور اجرا می شود ، هنگامی که اجرا به اتمام رسید ، کاربر یک صفحه معمولی HTML را دریافت می کند که می توان آن را در هر مرورگری مشاهده کرد.

مایکروسافت با وجود ۳.۵ ASP.NET در نظر دارد که موفقیت خود را با تصحیح و تکمیل ASP.NET تداوم بخشد.

به طور کلی چهار نسخه اصلی از ASP.NET وجود دارد :

ASP.NET ۱.۰

ASP.NET ۱.۱

ASP.NET ۲.۰

ASP.NET ۳.۵

اما ۳.۰ ASP.NET چی شده ؟ این نسخه اصلاً وجود ندارد !!! مایکروسافت از نام ۳.۰ NET برای ارائه مجموعه ای از فناوری های جدید استفاده کرد . فناوری های مذکور عبارتند از WF ، WCF ، WPF که در ادامه مورد بررسی قرار می گیرد.

۵ . ADO.NET چیست ؟

در ویژوال استدیو ۲۰۰۰ تکنیک ADO برای اتصال به بانک های اطلاعاتی ارائه شده بود که در NET. این امکان با تغییرات شگرف و با عنوان ADO.NET ارائه شده است ، ADO.NET یک تکنیک است که برنامه های کاربردی دات نت اعم از دسک تاپ و تحت وب از آن برای تعامل با یک بانک اطلاعاتی استفاده می کنند .

ADO.NET بر قابلیت های مجموعه کوچکی از کلاس ها تکیه دارد این کلاس ها به دو گروه تقسیم می شوند : آن هایی که برای شمول و مدیریت داده ها می روند و آن هایی که برای اتصال به یک منبع داده خاص به کار می روند

مانند : DateReader , connection , command , DataRow , DataTable , Datasets

بطور کلی دات نت شامل چهار فراهم کننده زیر است :

- فراهم کننده SQL Server: بهینه ترین دسترسی به یک بانک اطلاعاتی SQL Server نسخه ۷ به بالا را فراهم می کند.
- فراهم کننده OLE DB: دسترسی به هر منبع داده ای را که دارای یک درایور OLE DB باشد ، فراهم می کند.

- فراهم کننده Oracle: بهینه ترین دسترسی به یک بانک اطلاعاتی Oracle نسخه ۸i و بالاتر را فراهم می کند.

- فراهم کننده ODBC: دسترسی به هر منبع داده ای را که دارای یک درایور ODBC (Open Database Connection) باشد را فراهم می کند .

بدین ترتیب می توان به راحتی و بدون نگرانی از ADO.NET برای برقراری ارتباط با بانک اطلاعاتی و اعمال تغییرات در آن ، در تمامی برنامه کاربردی تحت دات نت استفاده کرد .

برای یادگیری کاربردی این تکنیک بنده کتاب برنامه نویسی بانک اطلاعاتی با C# ، تالیف مهندس رمضان عباس نژاد^۱ را پیشنهاد می کنم.

۶ . WPF چیست ؟

WPF سر آغاز سه کلمه Windows Presentation Foundation می باشد. هر کسی که تا به حال در محیط های گرافیکی و یا به اصلاح برنامه نویسان، محیط های ویژوال، برنامه نویسی کرده باشد، یقیناً با مفاهیم Windows Application ها که گاهی به صورت مخفف WinApp نیز نامیده می شوند، آشنا می باشد. این نوع برنامه نویسی همزمان با ورود سیستم عامل های ویندوز در دنیای کامپیوتر شروع شد و روز به روز با به وجود آمدن زبان های متفاوت جایگاه محبوبتری نزد برنامه نویسان پیدا کرد، همانطور که می دانید، Windows Application ها، از API های سیستم عامل مربوطه (که اکثراً ویندوز XP نیز می باشد) برای ترسیم عناصر گرافیکی یا همان عناصر ویژوال، استفاده می کنند. به عنوان مثال برای ترسیم انواع دکمه ها، فرم ها و بسیاری از عناصر دیگری که با آن ها آشنا هستید، از توابع API ویندوز کمک گرفته می شود. همین مسئله باعث ایجاد محدودیت برای برنامه نویسان در ایجاد کنترل های سفارشی با ظاهر دلخواه خود شده بود. اگر چه با ابزار های گرافیکی که در دات نت فریم ورک ۲.۰ نیز وجود داشت، می توانستیم تا حد خوبی اقدام به ایجاد کنترل های مورد دلخواه خود را بکنیم، اما این موضوع نیاز به دانستن اطلاعات زیاد در مورد ایجاد کنترل های سفارشی و همچنین نوشتن گاه کد های بسیار زیادی جهت ایجاد کنترل مورد نظر می بود. این به آن دلیل بود که قالب و اساس اولیه کنترل ها بسته بود و نمی توانستید به راحتی کنترل ها را شخصی سازی نمایید. در بهترین حالت، یک برنامه نویس ماهر میتوانست با ارث بری از کلاس Control اقدام به ایجاد یک کنترل جدید با ظاهر و امکانات مورد نظر خود بکند. به عنوان مثال طراحی یک منوی سفارشی بیش از ۱۰۰۰ خط کد احتیاج داشت ، شاید ۱۰۰۰ خط، برای یه برنامه نویس بسیار ناچیز باشد. ولی چنانچه بخواهید تمامی کنترل های برنامه های خود را، خودتان طراحی کنید، می بینید که زمان زیادی از وقت شما صرف نوشتن کد ها می گردد، این مسئله زمانی نمود بیشتری پیدا میکند که بخواهید، اکثر جنبه های یک کنترل را در کنترل سفارشی خود قرار دهید. به عنوان مثال به دلیل قرار گرفتن حالت های مختلف گرادیان بر روی منو، استفاده از امکانات قبلی مانند ترسیم متن آیتم به صورت اتوماتیک توسط خود منو و یا ترسیم کلید های میان بر آیتم و از بین می رود و تمامی این موارد بایستی با کد و توسط شما ایجاد گردد. درست است که می توان از کنترل های ایجاد شده توسط خودتان

^۱ آموزش گام به گام برنامه نویسی بانک اطلاعاتی با C# (مرجع کامل) تالیف : مهندس رمضان عباس نژادورزی انتشارات فن آوری نوین

به کرات و در برنامه های مختلف استفاده کنید و لی تجربه نشان داده است که گاهی نیز مجبور به ایجاد کنترل دیگری می شوید. این به این دلیل نیست که شما الزاما کنترل قبلی خود را خوب طراحی نکرده اید. عوامل زیادی می توانند باعث بروز این مورد شوند که پرداختن به آن ها از حوصله این بحث خارج است.

حال که تا حدودی با مشکلات برنامه نویسی های WinApp به روش جاری شدید، تکنولوژی WPF به روشی دیگر عمل می کند. در واقع علاوه بر اینکه این تکنولوژی همچنان دارای کنترل های سابق که آن ها را می شناسید، می باشد، می تواند دسترسی به بیشتر جنبه های کنترل ها را برای شما فراهم کند. در واقع قدرت WPF در این است که اساس و پایه هر کنترلی مانند برنامه نویسی قبل، بسته نیست و این شما هستید که به WPF خواهید گفت که متن روی کنترل را به چه صورتی طراحی کنید. یا پس زمینه کنترل یا کناره های آن را به آن صورتی که شما می گنید طراحی کند. به همین منظر نیز دارای ابزارهای بسیار زیادی جهت کار برای طراحی کنترل های شما مهیا می کند. ابزارهایی مانند قلم مو های گرادیان با تعداد رنگ های نامحدود، انواع ابزار های گرافیکی برای ترسیم شکل دلخواه شما، امکان ایجاد افکت های بسیار زیبا و متنوع بر روی هر قسمتی از کنترل که بخواهید، وجود افکت های از پیش تعریف شده، امکان طراحی های ۲ بعدی و نیز ۳ بعدی، امکان ایجاد انیمیشن و ...

۶.۱ WPF و DirectX

پایه و اساس WPF بر Directx استوار می باشد. این موضوع سبب می شود که بتوان از بسیاری از جنبه های گرافیکی بدون ایجاد سربار اضافی بر روی برنامه بهره برد و در واقع برنامه هایی با ظاهر هایی بسازید که ساختن آن ها با برنامه نویسی های پیشین یا غیر ممکن و یا متحمل کار بسیار زیادی بوده است. اگرچه نقطه قوت این تکنولوژی اعمال گرافیکی، انیمیشن و .. می باشد، ولی این بدان معنی نیست که نمی توان با WPF اقدام به ایجاد فرم ها و کنترل های سابق نمود. این تکنولوژی به شما امکان استفاده از کنترل های پیشین را می دهد و همچنین برنامه نویسی WinApp را به همان شکلی که می شناسید، برای شما مقدور می سازد. علاوه بر این موارد، WPF امکان کار با اسناد متنی، کنترل کردن بر روی نحوه Print شدن آن ها و ... را برای شما مهیا می سازد.

نکته دیگری که در مورد WPF باید بدان اشاره کرد، امکان برنامه نویسی به شیوه ای است که شاید تاکنون امثال آن را یا ندیده اید و یا بسیار کم دیده اید و آن هم برنامه نویسی بر اساس عناصری در WPF می باشد که به آن ها Page می گویند. این نوع برنامه نویسی را می توان به نوعی شبیه سازی برنامه های وب نامگذاری کرد. این نوع برنامه نویسی WPF Browser Application نام دارد، توسط این مدل برنامه نویسی، می توانید اسمبلی های ایجاد شده را در مرورگر وب نظیر Internet Explorer بدون پیغام های امنیتی که معمولا در صفحات وب وجود دارند، نمایش دهید.

به طور کلی برنامه های ویندوزی از دو امکان، توابع User۳۲ و GDI/GDI+ برای ترسیم عناصر گرافیکی استفاده می کنند که User۳۲ امکان ترسیم عناصر ویزوال را با ظاهر عادی مهیا می کند. عناصری مانند فرم ها، دکمه ها و ... و GDI/GDI+ امکانات گسترده تری را جهت ایجاد برخی اعمال گرافیکی مانند ایجاد گرادیان ها و

... را مهیا می کنند. شرکت مایکروسافت به دلیل محدودیت هایی که در هر یک از دویخش فوق، وجود داشت، اقدام به ایجاد کتابخانه سطح بالایی به نام DirectX کرد. (حرف X می تواند جایگزین کلماتی مانند Sound و .. شود). این ابزار که امروزه نیز از آن استفاده های زیادی میشود، (از جمله در ایجاد بازی های سه بعدی و ...) با بهره گیری از توان کارت های گرافیکی با بهره بری بالا، حداکثر توان آن را برای ایجاد گرافیک های قوی به کار می برد. اما با قدرت زیاد این کتابخانه، به دلیل برقراری ارتباط مشکل با آن و نیاز به کد نویسی های زیاد، این ابزار بیشتر در تهیه بازی ها و برنامه های گرافیکی مورد استفاده قرار گرفت و جایگاه زیادی در توسعه برنامه های تجاری پیدا نکرد. تکنولوژی WPF تمامی این مشکلات را مرتفع کرد و در واقع کاربر را از درگیر کردن نوشتن کدهای زیاد و گاه طاقت فرسا به صورت مستقیم در DirectX، رهایی داد. WPF از تمامی قدرت DirectX جهت ایجاد گرافیک های ۲ بعدی، ۳ بعدی، ایجاد انیمیشن ها، استفاده می کند. همچنین ابزار های بسیاری را جهت طراحی کردن در اختیار شما قرار می دهد. علاوه بر این DirectX به جهت اینکه به خوبی با مفاهیم Gradient، Texture و ... تطبیق پیدا می کند، دارای سرعت بالاتری نسبت GDI و GDI+ می باشد. به این دلیل که این تکنولوژی ها برای رندر کردن از روش پیکسلی و الگوریتم های آن که اصطلاحاً Pixel By Pixel Instruction گفته می شود، استفاده می کنند.

یکی دیگر از مشکلاتی که کار کردن با DirectX به صورت مستقیم وجود داشت (دارد) به دلیل نوع بهینه سازی و نحوه رندر کردن اشکال توسط کارت های ویدیویی متفاوت بود، که با WPF این مشکل نیز مرتفع شده است. یکی از مهمترین اهداف WPF استفاده از GPU به جای CPU جهت انجام روتین های پیچیده گرافیکی می باشد که این امر باعث آزاد بودن CPU بوده که میتواند به پردازش های دیگر در سیستم رسیدگی کند

۷. Windows Communication Foundation چیست؟

Windows Communication Foundation یا (WCF)، یک قالب کاری برای ساختن برنامه های سرویس گرا است. با استفاده از WCF می توانید داده ها را به صورت پیام هایی از یک درگاه سرویس به درگاهی دیگر ارسال کنید. یک درگاه سرویس (service endpoint) می تواند بخشی از سرویس همیشه در دسترس باشد که توسط IIS، میزبانی می شود، یا می تواند سرویسی باشد که توسط یک برنامه میزبانی می شود. یک درگاه می تواند client یک سرویس باشد که داده ها را از درگاه سرویس request می کند. پیام ها می توانند به سادگی یک کاراکتر واحد یا یک کلمه باشند که به صورت XML ارسال شده است. یا به پیچیدگی یک رشته از داده های binary. چند سناریوی نمونه شامل موارد زیر می شود:

- یک سرویس امن برای مبادلات تجاری
- سرویسی که داده های کنونی را برای دیگران مهیا می کند، از قبیل گزارش ترافیک یا دیگر سرویس های مانیتورینگ.

- یک سرویس چت که به دو نفر اجازه می دهد با هم در ارتباط باشند یا داده ها را به هم رد و بدل کنند.
- یک برنامه dashboard که داده های یک یا چندین برنامه را جمع آوری می کند و در یک presentation منطقی نمایش می دهد.
- Expose کردن یک workflow پیاده سازی شده با استفاده از Windows Workflow Foundation به صورت یک سرویس WCF.
- یک برنامه Silverlight برای جمع آوری آخرین فیدهای داده ها.

در حالی که ایجاد چنین برنامه هایی قبل از وجود WCF ممکن بود، WCF توسعه درگاه ها را از قبل آسانتر می کند. به طور خلاصه، WCF جهت ارائه روشی قابل مدیریت برای ایجاد سرویس های وب و client های سرویس وب، طراحی شده است.

۷.۱ ویژگی های WCF

WCF شامل مجموعه ویژگی های زیر است.

۷.۱.۱ گرایش به سرویس

یکی از نتایج استفاده از استانداردهای WS این است که WCF شما را قادر به ایجاد برنامه های سرویس گرا می کند. معماری های سرویس گرا (SOA)، به معنای وابستگی به وب سرویس ها جهت ارسال و دریافت داده ها است. این سرویس ها مزیت loosely-coupled بودن بجای hard-coded بودن از برنامه ای به برنامه دیگر هستند. یک رابطه loosely-coupled بدین معناست که هر client ایجاد شده در هر platform می تواند تا وقتی که contract های لازم برآورده می شوند، به هر سرویسی وصل شود.

۷.۱.۲ Interoperability

WCF، استانداردهای صنعتی مدرن را برای interoperability سرویس های وب پیاده می کند.

۷.۱.۳ الگوهای پیام چندگانه

پیام ها با استفاده یکی از چندین الگو مبادله می شوند. یکی از رایج ترین الگوها، الگوی request/reply است، که یک درگاه داده ها را از یک درگاه دیگر request می کند، و آن درگاه پاسخ می دهد. الگوهای دیگری نیز وجود دارند، مانند یک پیام یک طرفه (one-way) که در آن یک درگاه واحد پیامی را بدون درخواست پاسخ ارسال می کند. الگویی پیچیده تر، الگوی تبادل دوگانه است که دو درگاه یک connection برقرار می کنند و داده ها را ردوبدل می کنند، مانند یک برنامه پیام نگاری سریع.

Service Metadata ۷.۱.۴

WCF، با استفاده از فرمت های معین شده در استانداردهای صنعتی از قبیل WSDL، Schema XML و WS-Policy، انتشار service metadata را ساپورت می کند. می توان از این metadata، برای پیکربندی و generate کردن اتوماتیک client بمنظور دسترسی به سرویس های metadata استفاده کرد. می توان metadata را با استفاده از استاندارد Service Metadata Exchange Web در HTTP یا HTTPS منتشر کرد.

Data Contracts ۷.۱.۵

از آنجاییکه WCF با استفاده از .NET Framework ساخته شده است، متدهای code-friendly برای تامین contractهایی که می خواهید اجرا کنید را نیز در بر می گیرد. یکی از انواع جهانی contractها، data contract است. در واقع، هنگامی که سرویس تان را با استفاده از Visual c# یا Visual Basic کد نویسی می کنید، آسانترین راه برای مدیریت داده ها، ایجاد کلاس هایی است که موجودیت داده ها را با propertyهایی که به موجودیت داده ها تعلق دارند، نمایش می دهند. WCF شامل یک سیستم جامع برای کارکردن با داده ها به روشی آسانتر است. بعد از اینکه کلاس هایی را که داده ها را نمایش می دهند، وارد کردید، سرویس شما به طور اتوماتیک metadata را generate می کند.

امنیت ۷.۱.۶

می توان پیام ها را برای محافظت از حریم خصوصی رمزگذاری کرد و می توان کاربران را ملزم کرد قبل از مجاز شدن به دریافت پیا ها، خودشان را authenticate کنند. امنیت را می توان با استفاده از استانداردهای شناخته شده از قبیل SSL یا WS-Secure Conversation پیاده سازی کرد.

Transport ها و Encoding های چندگانه ۷.۱.۷

پیام ها را می توان با هریک از چندین پروتکل و رمزگذاری transport داخلی ارسال کرد. رایج ترین پروتکل و رمزگذاری، ارسال پیام های SOAP متنی رمزگذاری شده با استفاده از HTTP برای استفاده در WWW است. متناوباً، WCF به شما اجازه ارسال پیام ها در TCP، یا MSMQ را می دهد. این پیام ها را می توان به صورت متن یا با استفاده از یک فرمت binary بهینه شده، رمزگذاری کرد. داده ها binary را می توان با استفاده از استاندارد MTOM به طور موثری ارسال کرد. اگر هیچ یک از transport ها یا encoding های موجود، نیاز شما را برآورده نکنند، می توانید transport یا encoding های دلخواه خودتان را ایجاد کنید.

۷.۱.۸ پیام های قابل اعتماد و به صف شده (queued)

WCF با استفاده از session های پیاده شده روی WS-Reliable Messaging، پیام های قابل اعتماد را ساپورت می کند.

۷.۱.۹ پیام های مدت دار (Durable)

پیام مدت دار، پیامی است که در صورت وقوع اختلال در ارتباط، هرگز از دست نمی رود. پیام ها در الگوی پیام مدت دار، همیشه در database ذخیره می شوند. اگر اختلالی روی دهد، database به شما اجازه می دهد هنگامی که connection به حالت اول بازمی گردد، به تبادل پیام ادامه دهید. همچنین می توانید با استفاده از Foundation Workflow، یک پیام مدت دار ایجاد کنید.

۷.۱.۱۰ تراکنش ها

WCF، تراکنش ها را هم با استفاده از یکی از این سه مدل تراکنش ساپورت می کند: Atomic، Distributed، API، transactions، ها در فضای نامی System. Transactions، و Microsoft Transaction Coordinator.

۷.۱.۱۱ Ajax و ساپورت Rest

REST، نمونه ای از تکنولوژی وب در حال تکامل است. می توان WCF را برای پردازش داده های XML ساده پیکربندی کرد. همچنین می توان WCF را برای ساپورت از فرمت های خاص XML از قبیل ATOM، که یک استاندارد محبوب RSS است، و حتی فرمت های غیر XML از قبیل JavaScript Object Notation گسترش داد.

۷.۱.۱۲ توسعه پذیری (Extensibility)

معماری WCF دارای چندین نکته توسعه پذیری است. اگر نیاز به قابلیت اضافی باشد، نکاتی وجود دارند که به شما اجازه سفارشی کردن رفتار یک سرویس را می دهند.

۸. WF (Windows Workflow Foundation) چیست ؟

برای شناخت این بستر کاملاً جدید باید ابتدا معنی Workflow را بدانید. یعنی توالی انجام دادن فعالیتها در یک ترتیب مشخص. windows Workflow Foundation، امکاناتی برای کار با Workflow ها در اختیار شما قرار می دهد.

یا به بیان دیگر پلت فرمی برای مدل سازی منطق برنامه های کاربردی با استفاده از دیاگرام های شبیه فلوچارت .

۹ . Windows CardSpace چیست ؟

این بستر برای ایجاد و استقرار هویت دیجیتال با یک دیدگاه جدید و امن طراحی شده است.

۱۰ . LINQ چیست ؟

این تکنولوژی ارائه شده توسط دات نت برای برنامه نویسانی که با داده های جدول گونه و آرایه های چند بعدی سر و کار دارند حتما بسیار جذاب است . اگر بخواهیم LINQ را تعریف کنیم اینگونه تعریف می کنیم :

مختصر کلمه Query Language Integrated بوده و یک زبان برای تقاضا از هر گونه مجموعه داده (بانک اطلاعاتی، آرایه ها، Xml و...) می باشد. این زبان بسیار شبیه زبان SQL بوده و با زبان های C# و Vb.NET قابل پیاده سازی است.

هدف اصلی خلق LINQ ارائه یک زبان یکتا جهت تقاضا، برنامه ریزی و فیلتر کردن اطلاعات ذخیره شده در اشیاء مختلف اعم از پایگاه داده، اشیاء، آرایه ها و از همه مهتر XML است، این زبان قابلیت Debug شدن به صورت Runtime را داراست و بسیار قابل انعطاف است.

مثال زیر یک نمونه بسیار کوچک از کاربرد LINQ است :

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = "";
    int[] x = {100,3,4,5,8,9,2,54,62,93,14,71};

    IEnumerable<int> zoj = from a in x where a % 2 == 0 orderby a select a ;

    foreach (int i in zoj)
    {
        label1.Text += i.ToString() + " ";
    }
}
```

مثال بالا اعداد زوج آرایه را بدست آورده و پس از مرتب سازی در یک برجسب نمایش می دهد.

تقریبا امکان تمام اعمالی که با داده ها می توان انجام داد اعم از گروه کردن ، بزرگترین ، کوچکترین ، میانگین ، مجموع و مرتب سازی و در این تکنولوژی ، وجود دارد.

نهایتا اینکه با کمک تکنولوژی LINQ براحتی می توان از داده هایی مثل آرایه های چند بعدی گزارش هایی را تهیه کرد.

۱۱ . ADO.NET Entity Framework چیست ؟

درواقع Entity Framework تکنولوژی توسعه یافته ADO.Net است که فاصله بین برنامه نویسی شی گرای و بانک اطلاعاتی رابطه ای را پر می کند .این فاصله معمولا تحت عنوان عدم تطابق شناخته می شود. و یک تکنیک برنامه نویسی برای تبدیل ارتباطات در Database به مفاهیم Object Oriented در برنامه نویسی است. در واقع می توان گفت که کلاس ها را به Table ها map می کند. وقتی که شما می خواهید به Database دسترسی پیدا کنید، یا اطلاعاتی را ذخیره کنید، این کارها را مستقیما بر روی اشیاء (Object تان) انجام می دهید. Entity Framework مدل داده ها را به سه بخش تقسیم می کند ۱-مدل مفهومی یا ادراکی ۲-مدل منطقی ۳-مدل فیزیکی

۱۲ . معرفی Parallel LINQ و Task Parallel Library

پردازنده هایی (CPU) که امروزه در سرور ها و رایانه های شخصی مورد استفاده قرار می گیرند، کاملا متفاوت با پردازنده های قدیمی می باشد. پردازنده های جدید اغلب چند هسته ای بوده و قابلیت پردازش اطلاعات را به طور همزمان، موازی و توسط هسته های مختلف را دارند.

برنامه هایی که تاکنون توسط NET Framework نوشته شده اند بیشتر آن ها فقط توسط یکی از هسته های پردازنده مورد پردازش قرار می گیرند. تاکنون جهت اجرای برنامه ها و وظایف (Tasks) به شکل موازی از Thread Pool و به طور کلی Multi-Thread Programming استفاده می شده است. ولی باید توجه داشت که با استفاده از این روش ها هنوز هم پردازش روی یکی از هسته ها انجام می شود و استفاده از Thread Pool به شکل کاملا درستی برنامه ها را به طور همزمان اجرا نمی کند.

تاکنون اگر توسعه گران قصد داشتند اجرای برنامه را با استفاده از هسته های مختلف پردازنده انجام دهند، باید خود را درگیر کدهای سطح پایین و بسیار پیچیده ای می نمودند. اما خوشبختانه با معرفی کتابخانه Task Parallel Library در NET Framework ۴.۰. این عمل به ساده ترین شکل ممکن امکان پذیر شده است.

کتابخانه Task Parallel Library یا به عبارت خلاصه تری TPL به طور کلی در سه حوزه ی زیر مورد استفاده قرار می گیرد

Data Parallelism

۱۲.۱ مثال عملی Task Parallel Library

قبل از اینکه به ادامه مقاله پردازیم، توجه داشته باشید که برای استفاده از امکانات کتابخانه TPL باید فضای نامی زیر را به برنامه اضافه نمایید.

شکل ۱۲.۱

```
using System.Threading.Tasks;
```

یکی از نیازهای بسیار معمول برای برنامه نویسی موازی، پردازش سنگین اطلاعات می باشد. فرض کنید برنامه ی شما حاوی یک حلقه ی For یا ForEach با تعداد تکرارهای بسیار زیاد می باشد و در درون این حلقه نیز یک عمل زمانبر انجام می شود. این مسئله می تواند باعث هنگ نمودن برنامه و یا کاهش راندمان و کارایی برنامه گردد. تا زمانی که تمام تکرارهای این حلقه به پایان نرسیده است این مشکل ادامه خواهد داشت. پس اگر بتوان این پردازش را با استفاده از هسته های مختلف پردازنده و به طور همزمان انجام داد، زمان پردازش به طور قابل توجهی کاهش می یابد و این امر باعث بهبود کارایی نرم افزار خواهد شد.

شکل ۱۲.۲

برای درک بهتر به مثال زیر توجه کنید.

```
Debug.WriteLine("Processor threads states:");
for (int i = 0; i < 10; i++)
{
    Debug.WriteLine(Thread.CurrentThread.ThreadState);
}
```

خروجی حلقه بالا به صورت زیر خواهد بود.

```
Processor threads states:
Running
Running
Running
Running
Running
Running
Running
Running
Running
Running
```

شکل ۱۲.۳

نکته ی مهمی که تصویر بالا بیان می کند، این است که پردازش حلقه فقط توسط ترید (Thread) جاری برنامه انجام می شود و در نتیجه تا زمانی که تمامی تکرارهای حلقه انجام نشده باشد، این ترید مشغول می باشد.

شکل ۱۲.۴

اکنون این حلقه را با استفاده از کتابخانه TPL خواهیم نوشت.

```
Debug.WriteLine("Processor threads states:");

Parallel.For(0, 10, delegate(int i)
{
    Debug.WriteLine(Thread.CurrentThread.ThreadState);
});
```

همانطور که ملاحظه می‌نمایید کلاس Parallel دارای متدی به نام For می‌باشد که امکان پردازش دستورات داخل حلقه به طور موازی بر روی هسته‌های مختلف پردازنده را فراهم می‌آورد. پارامتر اول حلقه، ایندکس شروع حلقه و پارامتر دوم ایندکس پایان حلقه و پارامتر سوم یک Action Delegate می‌باشد که ما در قطعه کد بالا آن را به شکل inline نوشته ایم.

شکل ۱۲.۵

این حلقه را به شکل بسیار ساده‌ای مانند قطعه کد زیر نیز می‌توان نوشت.

```
Debug.WriteLine("Processor threads states:");

Parallel.For(0, 10, i => Debug.WriteLine(Thread.CurrentThread.ThreadState));
```

اما به خروجی این حلقه توجه نمایید.

شکل ۱۲.۶

```
Processor threads states:
Running
Background
Background
Running
Background
Background
Background
Background
Background
Running
Background
```

از شکل بالا به سادگی متوجه می‌شوید که پردازش این حلقه توسط ترید جاری برنامه (Running) و ترید های پس زمینه (Background) انجام شده است. در نتیجه ما بدون اینکه خود را درگیر مدیریت تریدهای مختلف کنیم، توانسته ایم برنامه را بر روی هسته‌های مختلف پردازنده به شکل موازی مورد پردازش قرار دهیم و این سهولت را مدیون کتابخانه TPL می‌باشیم.

۱۲.۲ مثال عملی Parallel LINQ

Parallel LINQ که به شکل اختصار بدان PLINQ نیز گفته می‌شود، جهت اجرای کوئری‌های زبان LINQ بر روی هسته‌های مختلف پردازنده و به شکل موازی استفاده می‌شود.

شکل ۱۲.۷

قطعه کد زیر مثال ساده‌ای از یک کوئری با استفاده از LINQ می‌باشد.

```
var result = from m in customers
              where m.FirstName.StartsWith("mor")
              select m;
```

با استفاده از متد AsParallel می‌توانیم دستور کوئری بالا را بر روی هسته‌های مختلف پردازنده و به شکل موازی انجام دهیم.

شکل ۱۲.۸

```
var result = from m in customers.AsParallel()
              where m.FirstName.StartsWith("mor")
              select m;
```

۱۳. خلاصه :

بدلیل کار سخت و زمانبر برنامه نویسی ، کمپانی مایکروسافت اقدام به عرضه چارچوب NET. در چهار نسخه (اواخر ۲۰۱۱) نمود که شامل مجموعه ای از فایل ها و سرویس های مورد نیاز و زبان های برنامه نویسی بود.

چارچوب NET. شامل موارد زیر می باشد:

- **CLR** : اجرا کننده برنامه های نوشته شده توسط NET. است که ضمن اجرا ، کدها را نیز مدیریت می کند.
- **کتابخانه کلاس NET.** : مخزنی از کلاس ها است که قابلیت های پیش ساخته ای را برای هر نوع کار از خواندن یک فایل XML گرفته تا ارسال یک پیام الکترونیکی ارائه می کند.
- **ASP.NET** : امکان ایجاد وب سایت های پویا و پیشرفته را برای برنامه نویسان مهیا می کند.
- **ADO.NET** : امکان برقراری یک اتصال امن و ایجاد تغییرات و گزارشات از پایگاه داده را ارائه می دهد.
- **WPF** : امکان طراحی برنامه ها با ظاهر و گرافیک فوق العاده را میسر می سازد.
- **WCF** : بستری برای ارتباطات شبکه است و سرویس ها و امکاناتی را برای برنامه نویسی تحت شبکه ارائه می دهد.
- **WF** : امکاناتی را برای کار با Workflow ها در اختیار برنامه نویسان قرار می دهد.
- **Card Space** : این بسته برای ایجاد و استقرار هویت دیجیتال با یک دیدگاه جدید و امن طراحی شده است
- **LINQ** : یک زبان تقاضا از هرگونه مجموعه داده می باشد که شباهت بسیاری به زبان SQL دارد.
- **ADO.NET Entity Framework** : تکنولوژی توسعه یافته ADO.NET است که فاصله بین برنامه نویسی شی گرا و بانک اطلاعاتی رابطه ای را پر می کند.
- **Task Parallel Library و Parallel LINQ** : امکان اجرای موازی پردازش های مختلف و دستورات LINQ را بصورت موازی روی هسته های مختلف پردازشگر مهیا می سازد.

منابع :

۱. چگونه با C# برنامه نویسی کنیم - DEITEL & DEITEL - ترجمه مهندس پروین صفاحی
۲. راهنمای جامع ۳.۵ ASP.NET در ۲۰۰۸ C# - ترجمه مهندس مریم جعفری نژاد
۳. <http://apress.com/book/view/۹۷۸۱۵۹۰۵۹۷۶۰۶>
۴. <http://www.codeproject.com/kb/wpf>
۵. http://en.wikipedia.org/wiki/Windows_Communication_Foundation
۶. <http://msdn۲.microsoft.com/enus/netframework/aa۶۶۳۳۲۴.aspx>

