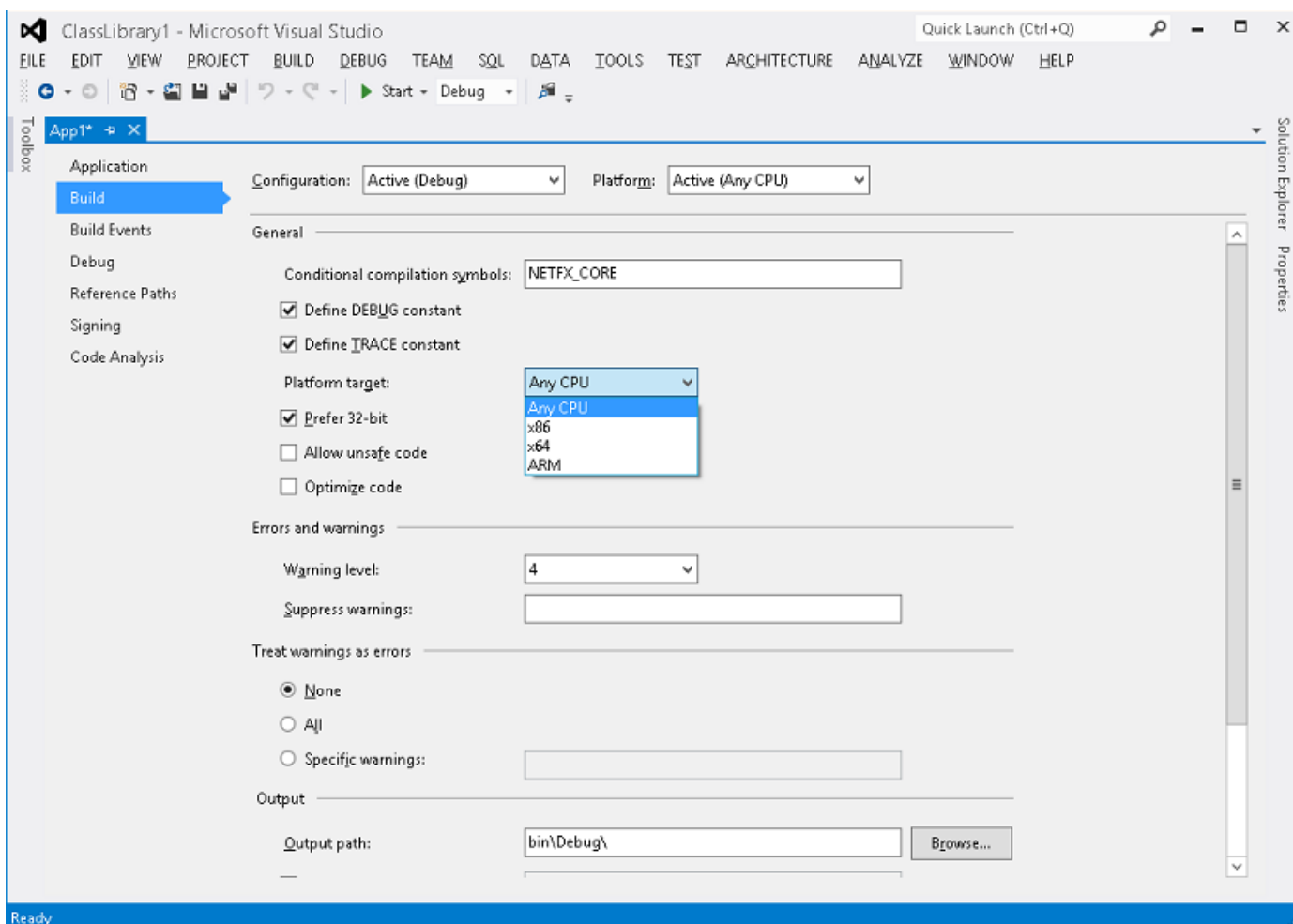


در [قسمت قبلی](#) با اسمبلی‌ها تا حدی آشنا شدیم. امروز می‌خواهیم یاد بگیریم که چگونه اسمبلی‌ها در حافظه بارگذاری می‌شوند. همانطور که می‌دانید CLR مسئول اجرای کدهای داخل اسمبلی‌هاست. به همین دلیل یک نسخه‌ی دات نت فریم ورک هم باید در ماشین مقصد نصب باشد. به همین منظور مایکروسافت بسته‌های توزیع شونده‌ی دات نت فریمورک را فراهم کرده تا به سادگی بر روی سیستم مشتری نصب شوند و بعضی از ویندوزها نیز نسخه‌های متفاوتی از دات نت فریم ورک را شامل می‌شوند. برای اینکه مطمئن شوید که آیا دات نت فریم ورک نصب شده است، می‌توانید در شاخه‌ی system32 سیستم، وجود فایل MSCorEE.dll را بررسی نمایید. البته بر روی یک سیستم می‌تواند نسخه‌های مختلفی از یک دات نت فریم ورک نصب باشد. برای آگاهی از اینکه چه نسخه‌هایی بر روی سیستم نصب است باید مسیرهای زیر را مورد بررسی قرار دهید:

```
%SystemRoot%\Microsoft.NET\Framework
%SystemRoot%\Microsoft.NET\Framework64
```

بسته‌ی دات نت فریمورک شامل ابزار خط فرمانی به نام [CLRVer.exe](#) می‌شود که همه‌ی نسخه‌های نصب شده را نشان می‌دهد. این ابزار با سوییچ all می‌تواند نشان دهد که چه پروسه‌هایی در حال حاضر دارند از یک نسخه‌ی خاص استفاده می‌کنند. یا اینکه ID یک پروسه را به آن داده و نسخه‌ی در حال استفاده را بیابیم.

قبل از اینکه پروسه‌ی بارگیری یک اسمبلی را بررسی کنیم، بهتر است به نسخه‌های 32 و 64 بیتی ویندوز، نگاهی بیندازیم: یک برنامه در حالت عمومی بر روی تمامی نسخه‌ها قابل اجراست و نیازی نیست که توسعه دهنده کار خاصی انجام دهد. ولی اگر توسعه دهنده نیاز داشته باشد که برنامه را محدود به پلتفرم خاصی کند، باید از طریق برگه build در projectProperties در قسمت PlatformTarget معماری پردازنده را انتخاب کند:



موقعیکه گزینه برای روی anyCPU تنظیم شده باشد و تیک گزینه 32-bit perfer را زده باشید، به این معنی است که بر روی هر سیستمی قابل اجراست؛ ولی اجرا به شیوهی 32 بیت اصلاح است. به این معنی که در یک سیستم 64 بیت برنامه را به شکل 32 بیت بالا می‌آورد.

بسته به پلتفرمی که برای توزیع انتخاب می‌کنید، کامپایلر به ساخت اسمبلی‌های با هدرهای P32(+) می‌پردازد. مایکروسافت دو ابزار خط فرمان را به نام‌های [DumpBin.exe](#) و [CoreFlags.exe](#) در راستای آزمایش و بررسی هدرهای تولید شده توسط کامپایلر ارائه کرده است.

موقعی که شما یک فایل اجرایی را اجرا می‌کنید، ابتدا هدرها را خوانده و طبق اطلاعات موجود تصمیم می‌گیرد برنامه به چه شکلی اجرا شود. اگر دارای هدر p32 باشد قابل اجرا بر روی سیستم‌های 32 و 64 بیتی است و اگر PE32+ باشد روی سیستم‌های 64 بیتی قابل اجرا خواهد بود. همچنین به بررسی معماری پردازنده که در قسمت هدر embed شده، پرداخته تا اطمینان کسب کند که با خصوصیات پردازنده مقصد مطابقت می‌کند.

نسخه‌های 64 بیتی ارائه شده توسط مایکروسافت دارای فناوری به نام WOW64 یا Windows On Windows64 هستند که اجازه‌ی اجرای برنامه‌های 32 بیت را روی نسخه‌های 64 بیتی، می‌دهند.

جدول زیر اطلاعاتی را ارائه میکند که در حالت عادی برنامه روی چه سیستم‌هایی ارائه شده است و اگر آن را محدود به نسخه‌های 32 یا 64 بیتی کنیم، نحوه‌ی اجرا آن بر روی سایر پلتفرم‌ها چگونه خواهد بود.

/platform Switch	Resulting Managed Module	x86 Windows	x64 Windows	ARM Windows RT
anycpu (the default)	PE32/agnostic	Runs as a 32-bit application	Runs as a 64-bit application	Runs as a 32-bit application
anycpu32bitpreferred	PE32/agnostic	Runs as a 32-bit application	Runs as a 32-bit application	Runs as a 32-bit application
x86	PE32/x86	Runs as a 32-bit application	Runs as a WoW64 application	Doesn't run
x64	PE32+/x64	Doesn't run	Runs as a 64-bit application	Doesn't run
ARM	PE32/ARM	Doesn't run	Doesn't run	Runs as a 32-bit application

بعد از اینکه هدر مورد آزمایش قرار گرفت و متوجه شد چه نسخه‌ای از آن باید اجرا شود، بر اساس نسخه‌ی انتخابی، یک از نسخه‌های MSCorEE سی و دو بیتی یا 64 بیتی یا ARM را که در شاخه‌ی system32 قرار دارد، در حافظه بارگذاری می‌نماید. در نسخه‌های 64 بیتی ویندوز که نیاز به MSCorEE نسخه‌های 32 بیتی احساس می‌شود، در آدرس زیر قرار گرفته است:

%SystemRoot%\SysWow64

بعد از آن ترد اصلی پروسه، متدی را در MSCorEE صدا خواهد زد که موجب آماده سازی CLR بارگذاری اسمبلی اجرایی EXE در حافظه و صدا زدن مدخل ورودی برنامه یعنی متد Main می‌گردد. به این ترتیب برنامه‌ی مدیریت شده (managed) شما اجرا می‌گردد.