

عنوان:	MSBuild
نویسنده:	یوسف نژاد
تاریخ:	۲۰:۵۰ ۱۳۹۱/۱۱/۰۸
آدرس:	www.dotnettips.info
برچسب‌ها:	NET, MSBuild.

MSBuild

به عنوان یک تعریف کلی، مایکروسافت بیلد (Microsoft Build)، پلتفرمی برای ساخت اپلیکیشن‌هاست. در این پلتفرم (که با عنوان MSBuild شناخته میشود) کلیه تنظیمات لازم برای تولید و ساخت یک اپلیکیشن درون یک فایل XML ذخیره میشود، که به آن **فایل پروژه** میگویند. ویژوال استودیو نیز از این ابزار برای تولید تمامی اپلیکیشن‌ها استفاده می‌کند، اما MSBuild به ویژوال استودیو وابسته نیست و کاملاً مستقل از آن است.

این ابزار به همراه دات نت فریمورک (البته نسخه کامل آن و نه نسخه‌های سبکتری چون Client Profile) نصب میشود. بنابراین با استفاده از فایل اجرایی این ابزار (msbuild.exe) میتوان فرایند بیلد را برای پروژه و یا سولوشن‌های خود، بدون نیاز به نصب ویژوال استودیو اجرا کرد. استفاده مستقیم از MSBuild در شرایط زیر نیاز میشود:

- ویژوال استودیو در دسترس نباشد.

- نسخه 64 بیتی این ابزار که در ویژوال استودیو در دسترس نیست. البته در بیشتر مواقع این مورد پیش نخواهد آمد مگر اینکه برای فرایند بیلد به حافظه بیشتری نیاز باشد.

- اجرای فرایند بیلد در بیش از یک پراسس (برای رسیدن به سرعت بالاتر). این امکان در تولید پروژه‌های C++ در ویژوال استودیو موجود است. همچنین از نسخه 2012 این امکان برای پروژه‌های C# نیز فراهم شده است.

- سفارشی‌سازی فرایند بیلد

- و ...

همچنین یکی دیگر از بخشهای مهم فرایند تولید اپلیکیشن که همانند ویژوال استودیو از این ابزار بصورت مستقیم استفاده میکند Team Foundation Build است.

با استفاده از خط فرمان این ابزار تنظیمات فراوانی را برای سفارشی سازی عملیات بیلد میتوان انجام داد که شرح آنها بحثی مفصل میطلبد. تنظیمات بسیار دیگری هم در فایل پروژه قابل اعمال است (توضیحات بیشتر در [اینجا](#)). منابع برای مطالعه بیشتر:

[MSBuild Reference](#)

[\(Visual Studio Integration \(MSBuild](#)

[Walkthrough: Using MSBuild](#)

Microsoft Build API

در دات‌نت فریمورک فضای نامی با عنوان Microsoft.Build نیز وجود دارد که امکانات این ابزار را در اختیار برنامه نویسی قرار میدهد. برای استفاده از این کتابخانه باید ارجاعی به اسمبلی آن داد، که به همین نام بوده و به همراه دات‌نت فریمورک نصب میشود. کد زیر نحوه استفاده اولیه از این کتابخانه را نشان میدهد:

```
private static void TestMSBuild(string projectFullPath)
{
    var pc = new ProjectCollection();
    var globalProperties = new Dictionary<string, string>() { { "Configuration", "Debug" }, { "Platform", "AnyCPU" } };
    var buildRequest = new BuildRequestData(projectFullPath, globalProperties, null, new string[] { "Build" }, null);
    var buildResult = BuildManager.DefaultBuildManager.Build(new BuildParameters(pc), buildRequest);
}
```

با اینکه ارائه مقداری غیرنال برای آرگومان globalProperties اجباری است اما پرکردن آن کاملاً اختیاری است، زیرا تمام تنظیمات ممکن را میتوان در خود فایل پروژه ثبت کرد.

برای مطالعه بیشتر منابع زیر پیشنهاد میشود: [Microsoft.Build](#)

[NET 4.0 MSBuild API introduction.](#)

استفاده از msbuild.exe

ابزار msbuild به صورت یک فایل exe در دسترس است و برای استفاده از آن میتوان از خط فرمان ویندوز استفاده کرد. مسیر فایل اجرایی آن (MSBuild.exe) در ریشه مسیر دات نت فریمورک است، بصورت زیر:

نسخه 32 بیتی:

C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe

نسخه 64 بیتی:

C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe

برای استفاده از آن میتوان مسیر فایل پروژه یا سولوشن (فایل با پسوند .csproj یا .vbproj یا .sln) را به آن داد تا سایر عملیات تولید را به صورت خودکار تا آخر به انجام برساند. کاری که عینا در ویژوال استودیو در زمان Build انجام میشود! برای بهره برداری از آن در کد میتوان از کلاس Process استفاده کرد. برای مسیر این فایل هم میتوان از نشانی‌هایی که در بالا معرفی شد استفاده کرد یا برای راحتی و امنیت بیشتر از کلید رجیستری مربوطه که در کد زیر نشان داده شده استفاده کرد:

```
private static void TestMSBuild1(string projectPath)
{
    var regKey = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\MSBuild\ToolsVersions\4.0");
    if (regKey == null) return;
    var msBuildExeFilePath = Path.Combine(regKey.GetValue("MSBuildToolsPath").ToString(), "MSBuild.exe");
    var startInfo = new ProcessStartInfo
    {
        FileName = msBuildExeFilePath,
        Arguments = projectPath,
        WindowStyle = ProcessWindowStyle.Hidden
    };
    var process = Process.Start(startInfo);
    process.WaitForExit();
}
```

بدین ترتیب عملیاتی مشابه عملیات Build در ویژوال استودیو انجام میشود و با توجه به تنظیمات موجود در فایل پروژه، پوشه‌های خروجی (مثلا bin و obj در حالت پیش فرض پروژه‌های ویژوال استودیو) نیز در مسیرهای مربوطه ایجاد میگردد.