

انتشار نوع‌ها (Types) به یک ماژول

در این قسمت به نحوه‌ی تبدیل سورس به یک فایل قابل انتشار می‌پردازیم. کد زیر را به عنوان مثال در نظر بگیرید:

```
public sealed class Program {
    public static void Main() {
        System.Console.WriteLine("Hi");
    }
}
```

این کد یک ارجاع به نام کنسول دارد که این ارجاع، داخل فایلی به نام mscorlib.dll قرار دارد. پس برنامه‌ی ما نوعی را دارد، که آن نوع توسط شرکت دیگری پیاده سازی شده است. برای ساخت برنامه‌ی کد بالا، کدها را داخل فایلی با نام program.cs قرار داده و با دستور زیر در خط فرمان آن را کامپایل می‌کنیم:

```
csc.exe /out:Program.exe /t:exe /r:mscorlib.dll Program.cs
```

کد بالا با سوئیچ اول می‌گوید که فایلی را با نام program.exe درست کن و با سوئیچ دوم می‌گوید که این برنامه از نوع کنسول هست.

موقعیکه کامپایلر فایل سورس را مورد بررسی قرار می‌دهد، متوجه متد writeline می‌گردد؛ ولی از آنجاکه این نوع توسط شما ایجاد نشده است و یک نوع خارجی است، شما باید یک مجموعه از ارجاعات را به کامپایلر داده تا آن نوع را در آن‌ها بیابد. ارائه این ارجاعات به کامپایلر توسط سوئیچ /r که در خط بالا استفاده شده است، صورت می‌گیرد.

mscorlib یک فایل سورس است که شامل همه‌ی نوع‌ها، از قبیل int, string, byte و خیلی از نوع‌های دیگر می‌شود. از آنجائیکه استفاده‌ی از این نوع‌ها به طور مکرر توسط برنامه نویس‌ها صورت می‌گیرد، کامپایلر به طور خودکار این کتابخانه را به لیست ارجاعات اضافه می‌کند. به بیان دیگر خط بالا به شکل زیر هم قابل اجراست:

```
csc.exe /out:Program.exe /t:exe Program.cs
```

به علاوه از آنجایی که بقیه‌ی سوئیچ‌ها هم مقدار پیش فرضی را دارند، خط زیر هم معادل خطوط بالاست:

```
csc.exe Program.cs
```

اگر به هر دلیلی دوست ندارید که سمت mscorlib ارجاعی صورت بگیرید، می‌توانید از دستور زیر استفاده کنید:

```
/nostdlib
```

مایکروسافت موقعی از این سوئیچ بالا استفاده کرده است که خواسته است خود mscorlib را بسازد. با اضافه کردن این سوئیچ، کد مثال که حاوی شیء یا نوع کنسول است به خطا برخورد خورد چون تعریف آن در mscorlib صورت گرفته و شما با سوئیچ بالا دسترسی به آن را ممنوع اعلام کرده‌اید و از آنجاکه این نوع تعریف نشده، برنامه از کامپایل بازخواهد ماند.

```
csc.exe /out:Program.exe /t:exe /nostdlib Program.cs
```

بیا باید نگاه دقیقتری به فایل program.exe ساخته شده بیندازیم؛ دقیقا این فایل چه نوع فایلی است؟ برای بسیاری از مبتدیان، این یک فایل اجرایی است که در هر دو ماشین 32 و 64 بیتی قابل اجراست. ویندوز از سه نوع برنامه پشتیبانی می‌کند: CUI یا برنامه‌های تحت کنسول، برنامه‌هایی با رابط گرافیکی GUI و برنامه‌های مخصوص windows store که سوئیچ‌های آن به شرح زیر است:

```
//CUI
/t:exe

//GUI
/t:winexe

//Windows store App
/t:appcontainerexe
```

قبل از اینکه بحث را در مورد سوئیچ‌ها به پایان برسانیم، اجازه دهید در مورد فایل‌های پاسخگو یا response file صحبت کنیم. یک فایل پاسخگو، فایلی است که شامل مجموعه‌ای از سوئیچ‌های خط فرمان می‌شود. موقعی که csc.exe اجرا می‌شود، به فایل پاسخگویی که شما به آن معرفی کرده‌اید مراجعه کرده و فرمان را با سوئیچ‌های داخل آن اجرا می‌کند. معرفی یک فایل پاسخگو به کامپایلر توسط علامت @ و سپس نام فایل صورت می‌گیرد و در این فایل هر خط، شامل یک سوئیچ است. مثلا فایل پاسخگوی response.rsp شامل سوئیچ‌های زیر است:

```
/out:MyProject.exe
/target:winexe
```

و برای در نظر گرفتن این سوئیچ‌ها فایل پاسخگو را به کامپایلر معرفی می‌کنیم:

```
csc.exe @MyProject.rsp CodeFile1.cs CodeFile2.cs
```

این فایل خیلی کار شما را راحت می‌کند و نمی‌گذارد در هر بار کامپایل، مرتب سوئیچ‌های آن را وارد کنید و کیفیت کار را بالا می‌برد. همچنین می‌توانید چندین فایل پاسخگو داشته باشید و هر کدام شامل سوئیچ‌های مختلفی تا اگر خواستید تنظیمات کامپایل را تغییر دهید، به راحتی تنها نام فایل پاسخگو را تغییر دهید. همچنین کامپایلر سی شارپ از چندین فایل پاسخگو هم پشتیبانی می‌کند و می‌توانید هر تعداد فایل پاسخگویی را به آن معرفی کنید. در صورتیکه فایل را با نام csc.rsp نامگذاری کرده باشید، نیازی به معرفی آن نیست چرا که کامپایلر در صورت وجود، آن را به طور خودکار خواهد خواند و به این فایل global response file یا فایل پاسخگوی عمومی گویند.

در صورتیکه چندین فایل پاسخگو که به آن فایل‌های محلی local می‌گویند، معرفی کنید که دستورات آن‌ها (سوئیچ) با دستورات داخل csc.rsp مقدار متفاوتی داشته باشند، فایل‌های محلی الویت بالاتری نسبت به فایل global داشته و تنظیمات آن‌ها روی فایل global رونوشت می‌گردند.

موقعی که شما دات نت فریمورک را نصب می‌کنید، فایل csc.rsp را با تنظیمات پیش فرض در مسیر زیر نصب می‌کند:

```
%SystemRoot%\
Microsoft.NET\Framework(64)\vX.X.X
```

حروف x نمایانگر نسخه‌ی دات نت فریمورکی هست که شما نصب کرده‌اید. آخرین ورژن از این فایل در زمان نگارش کتاب، شامل سوئیچ‌های زیر بوده است.

```
# This file contains command-line options that the C#
# command line compiler (CSC) will process as part
# of every compilation, unless the "/noconfig" option
# is specified.
# Reference the common Framework libraries
/r:Accessibility.dll
/r:Microsoft.CSharp.dll
```

```

/r:System.Configuration.dll
/r:System.Configuration.Install.dll
/r:System.Core.dll
/r:System.Data.dll
/r:System.Data.DataSetExtensions.dll
/r:System.Data.Linq.dll
/r:System.Data.OracleClient.dll
/r:System.Deployment.dll
/r:System.Design.dll
/r:System.DirectoryServices.dll
/r:System.dll
/r:System.Drawing.Design.dll
/r:System.Drawing.dll
/r:System.EnterpriseServices.dll
/r:System.Management.dll
/r:System.Messaging.dll
/r:System.Runtime.Remoting.dll
/r:System.Runtime.Serialization.dll
/r:System.Runtime.Serialization.Formatters.Soap.dll
/r:System.Security.dll
/r:System.ServiceModel.dll
/r:System.ServiceModel.Web.dll
/r:System.ServiceProcess.dll
/r:System.Transactions.dll
/r:System.Web.dll
/r:System.Web.Extensions.Design.dll
/r:System.Web.Extensions.dll
/r:System.Web.Mobile.dll
/r:System.Web.RegularExpressions.dll
/r:System.Web.Services.dll
/r:System.Windows.Forms.dll
/r:System.Workflow.Activities.dll
/r:System.Workflow.ComponentModel.dll
/r:System.Workflow.Runtime.dll
/r:System.Xml.dll
/r:System.Xml.Linq.dll

```

این فایل حاوی بسیاری از ارجاعات اسمبلی‌هایی است که بیشتر توسط توسعه دهندگان مورد استفاده قرار می‌گیرد و در صورتیکه برنامه‌ی شما به این اسمبلی‌ها محدود می‌گردد، لازم نیست که این اسمبلی‌ها را به کامپایلر معرفی کنید.

البته ارجاع کردن به این اسمبلی‌ها تا حد کمی باعث کند شدن صورت کامپایل می‌شوند؛ ولی تاثیری بر فایل نهایی و نحوه‌ی اجرای آن نمی‌گذارند.

نکته: در صورتی که قصد ارجاعی را دارید، می‌توانید آدرس مستقیم اسمبلی را هم ذکر کنید. ولی اگر تنها به نام اسمبلی اکتفا کنید، مسیرهای زیر جهت یافتن اسمبلی بررسی خواهند شد:

دایرکتوری برنامه

دایرکتوری که شامل فایل csc.exe می‌شود. که خود فایل mscorlib از همانجا خوانده می‌شود و مسیر آن شبیه مسیر زیر است:

```
%SystemRoot%\Microsoft.NET\Framework\v4.0.#####
```

هر دایرکتوری که توسط سوئیچ /lib مشخص شده باشد.

هر دایرکتوری که توسط [متغیر محلی lib](#) مشخص شده باشد.

استفاده از سوئیچ /noconfig هم باعث می‌شود که فایل‌های پاسخگوی از هر نوعی، چه عمومی و چه محلی، مورد استفاده قرار نگیرند. همچنین شما مجاز هستید که فایل csc.rsp را هم تغییر دهید؛ ولی این نکته را فراموش نکنید، در صورتی که برنامه‌ی شما به سیستمی دیگر منتقل شود، تنظیمات این فایل در آنجا متفاوت خواهد بود و بهتر هست یک فایل محلی را که همراه خودش هست استفاده کنید.

در قسمت بعدی نگاه دیگری بر متادیتا خواهیم داشت.