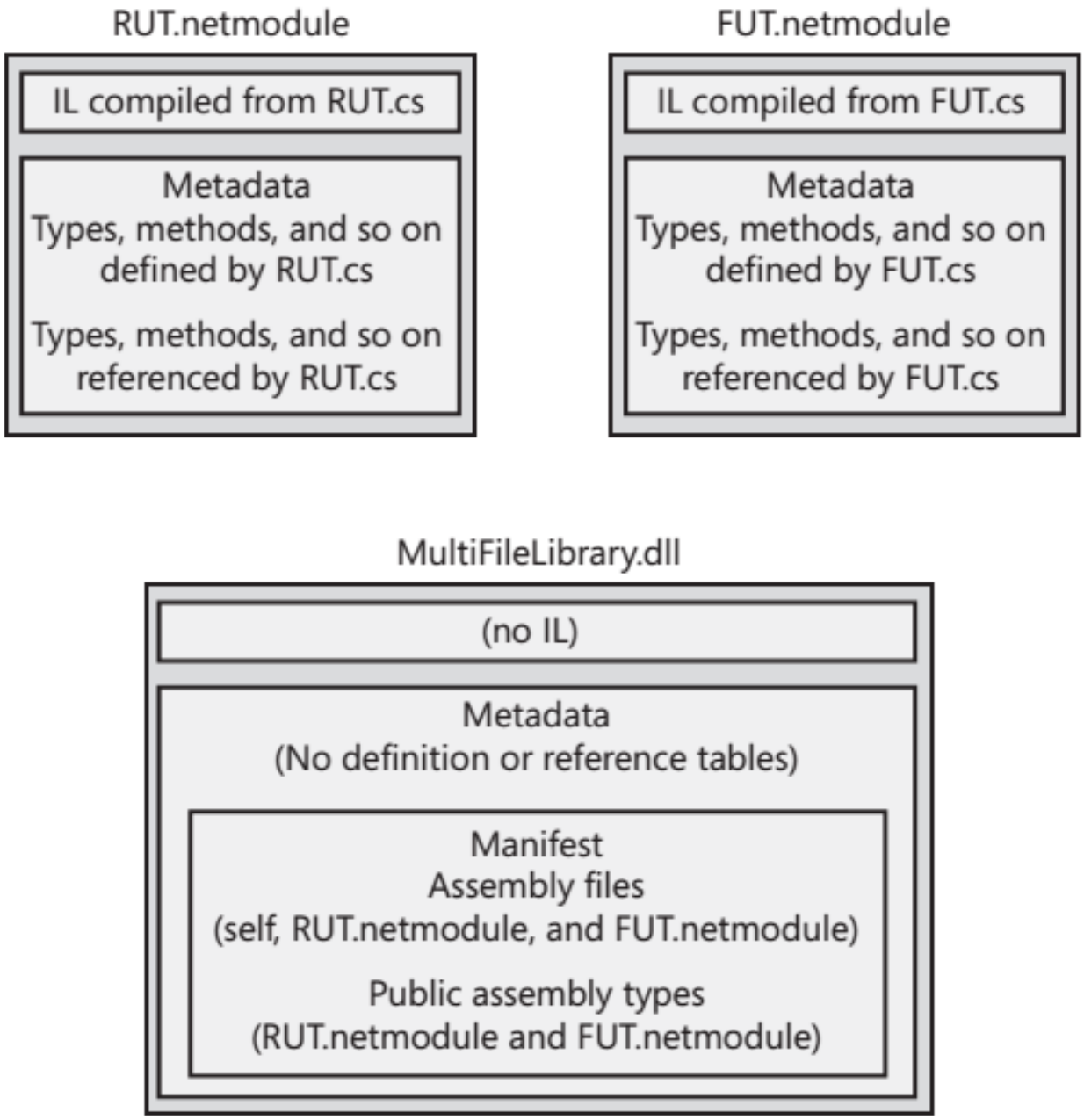
عنوان: **آشنایی با CLR: قسمت پانزدهم**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۵/۳۰ ۱:۵۵
آدرس: www.dotnettips.info
گروه‌ها: CLR, .NET Framework

در قسمت قبلی نحوه‌ی ساخت اسمبلی را یاد گرفتیم. ولی ممکن است که بخواهید اسمبلی را از طریق Assembly Linker یا AL.exe ایجاد کنید. این روش موقعی سودمند است که بخواهید یک اسمبلی از ماژولها از کامپایلرهای مختلف را ایجاد کنید یا اینکه کامپایلر شما مانند کامپایلر سی شارپ از دستور یا سوئیچی مشابه addmodule استفاده نمی‌کند. یا حتی اینکه در زمان کامپایل هنوز اطلاعاتی از نیازمندی‌های اسمبلی‌ها ندارید و به بعد موکول می‌کنید. از AL همچنین می‌توانید در زمینه‌ی ساخت اسمبلی‌های فقط ریسورس هم استفاده کنید که می‌تواند جهت انجام localization به کار رود. AL می‌تواند یک فایل dll یا exe تولید کند که شامل یک فایل manifest بوده که اشاره به ماژول‌های تشکیل دهنده‌اش دارد.

نحوه‌ی ساخت اسمبلی با استفاده از ابزار AL :

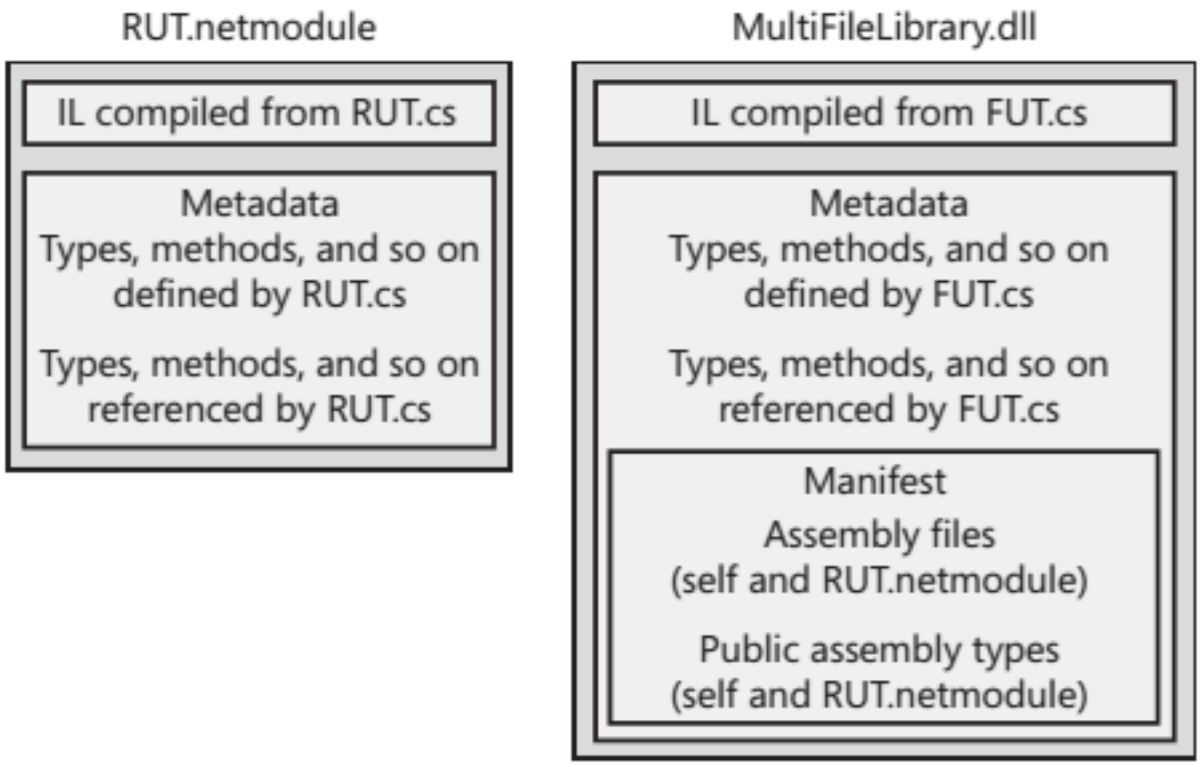
```
csc /t:module RUT.cs
csc /t:module FUT.cs
al /out: MultiFileLibrary.dll /t:library FUT.netmodule RUT.netmodule
```

تصویر زیر نتیجه‌ی دستور بالاست:



در این مثال ما دو ماژول جدا به نام‌های RUT.netmodule و FUT.netmodule را در یک اسمبلی ایجاد کرده‌ایم. داخل این اسمبلی‌ها جدول متادیتا یا بخش IL از ماژول‌ها به چشم نمی‌خورد. به این معنی که کد IL و جداول مربوطه به آن، هر کدام داخل ماژول یا فایل خودش بوده و در اسمبلی کدی وجود ندارد و تنها یک جدول مانیفست جهت شناسایی ماژول‌هایش دارد. شکل بالا گویای اطلاعات داخلی اسمبلی است که می‌توانید با تصویری که در قسمت قبلی درج شده مقایسه کنید.

تصویر قسمت قبلی جهت مقایسه:



در این حالت سه فایل تشکیل شده است که یکی از آن‌ها FUT.netmodule ، MultiFileLibrary.dll و RUT.netmodule است و در استفاده از این ابزار هیچ راهی برای داشتن یک تک فایل وجود ندارد.
این ابزار همچنین می‌تواند فایل‌های GUI, CUI و ... را با سوئیچ‌های زیر هم تولید کند:

```
/t[arget]:exe, /t[arget]:winexe, or /t[arget]:appcontainerexe
```

البته اینکار تا حدی غیر معمول است که یک فایل exe بخواهد کدهای IL ابتدایی را از ماژول‌های جداگانه بخواند. در صورتیکه چنین قصدی را دارید، باید یکی از ماژول‌ها را به عنوان مدخل ورودی Main تعریف کنید تا برنامه از آنجا آغاز به کار کند. نحوه‌ی ساخت یک فایل اجرایی و معرفی ماژول Main به شکل زیر است:

```
csc /t:module /r:MultiFileLibrary.dll Program.cs
al /out:Program.exe /t:exe /main:Program.Main Program.netmodule
```

در اولین خط مانند سابق فایل netmodule تهیه می‌گردد و در خط دوم، داخل اسمبلی قرار می‌گیرد. ولی به علت استفاده از سوئیچ main یک تابع عمومی global به نام __EntryPoint هم تعریف می‌گردد که کد IL آن به شرح زیر است:

```
.method privatescope static void __EntryPoint$PST06000001() cil managed
{
.entrypoint
// Code size 8 (0x8)
.maxstack 8
IL_0000: tail.
IL_0002: call void [.module 'Program.netmodule']Program::Main()
IL_0007: ret
} // end of method 'Global Functions'::__EntryPoint
```

کد بالا یک کد ساده است که می‌گوید داخل فایل Program.netmodule در نوع Program متدی وجود دارد به نام Main که محل آغازین برنامه است. البته این روش ایجاد فایل‌های EXE، بدین شکل توصیه چندانی نمی‌شود و ذکر این مطلب فقط اطلاع از وجود

چنین قابلیتی بود.