عنوان: **آشنایی با CLR: قسمت هفدهم**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۶/۰۱ ۰:۴۰
آدرس: www.dotnettips.info
گروه‌ها: CLR, .NET Framework

در مقاله قبلی در مورد افزودن منابع به اسمبلی صحبت‌هایی کردم که قسمتی از این منابع مربوط به اطلاعات نسخه بندی بود. در این مقاله قصد داریم این مسئله را بازتر کرده و در مورد نحوه‌ی نسخه بندی بیشتر صحبت کنیم.
در مقاله‌ی قبلی وقتی نسخه‌ی یک اسمبلی را مشخص می‌کردیم، از 4 عدد که با نقطه از هم جدا شده بودند، استفاده کردیم که در جدول زیر این 4 نامگذاری را مشاهده می‌کنید:

شماره اصلی Major Number	*شماره جزئی Minor Number*	*شماره ساخت Build Number*	*شماره بازبینی Revision Number*
2	5	719	2

اسمبلی بالا به ورژن یا نسخه‌ی 2.5.719.2 اشاره دارد که دو شماره‌ی اول (2.5) مثل تمامی برنامه‌ها به میزان تغییرات کارکردی یک اسمبلی اشاره دارد و عموم مردم هم نسخه یک نرم افزار را به همین دو عدد میشناسند. عدد سوم به این اشاره دارد که در شرکت، این ورژن از اسمبلی چندبار build شده است و شما باید به ازای هر Build این عدد را افزایش دهید. عدد آخری به این اشاره دارد که در طول روز انتشار، این چندمین Build بوده است. اگر در زمان ارائه‌ی این اسمبلی باگ مهمی در آن یافت شود، با هر بار Build آن در یک روز، باید این عدد افزایش یابد و برای روزهای آتی این مقدار مجددا آغاز می‌شود. مایکروسافت از این سیستم نسخه بندی استفاده می‌کند و بسیار توصیه می‌شود که توسعه دهندگان هم از این روش تبعیت کنند.

در جدول سابق شما متوجه شدید که سه نسخه بندی را می‌توان روی یک اسمبلی اعمال کرد که به شرح زیر است:

AssemblyFileVersion : این شماره نسخه در منابع اطلاعاتی Win32 ذخیره می‌گردد و کاربرد آن تنها جهت نمایش این اطلاعات است و CLR هیچ گونه ارجاع یا استفاده‌ای از آن ندارد. در حالت عادی، شما باید دو شماره اولی را جهت نمایش عمومی مشخص کنید. سپس با هر بار Build کردن، شماره‌های ساخت و بازبینی را هم به همان ترتیب افزایش می‌دهید. حالت ایده‌آل این است که ابزار AL یا CSC به طور خودکار با هر بار Build شدن، با توجه به زمان سیستم، به طور خودکار این دو شماره آخر را مشخص کنند ولی متاسفانه واقعیت این است که چنین کاری صورت نمی‌گیرد. این اعداد جهت نمایش و شناسایی اسمبلی برای اشکال زدایی مشکلات اسمبلی به کار می‌رود.

AssemblyInformationalVersion : این شماره نسخه هم در منابع اطلاعاتی Win32 ذخیره می‌گردد و تنها هدف اطلاعاتی دارد. مجددا اینکه CLR هیچ گونه اهمیتی به آن نمی‌دهد. این شماره نسخه به محصولی اشاره می‌کند که شامل این اسمبلی است.

به عنوان مثال ورژن 2 یک نرم افزار ممکن است شامل چند اسمبلی باشد که ورژن یکی از این اسمبلی‌ها یک است و دلیلش هم این است که این اسمبلی از نسخه‌ی 2 به بعد اضافه شده و در نسخه‌ی یک نرم افزار وجود نداشته است. به همین دلیل در این مدل از نسخه بندی شما دو شماره اول را به نسخه خود نرم افزار مقداردهی کرده و سپس مابقی اعداد را با هر بار پکیج شدن کل نرم افزار با توجه به زمان افزایش می‌دهید.

AssemblyVersion : این شماره نسخه در جدول متادیتای AssemblyDef ذخیره می‌گردد. CLR از این شماره نسخه جهت اتصال نام قوی Strongly Named به اسمبلی استفاده می‌کند (این مورد در فصل سه کتاب توضیح داده شده است). این شماره نسخه بسیار مهم بوده و به عنوان شناسه‌ی یکتا برای اسمبلی استفاده می‌شود.

موقعیکه شما شروع به توسعه‌ی یک اسمبلی می‌کنید، باید هر 4 شماره نسخه را مقداردهی کرده و تا زمانیکه توسعه‌ی نسخه بعدی

آن اسمبلی را آغاز نکرده‌اید، نباید آن را تغییر دهید. علت اصلی آن این است که موقعیکه اسمبلی «الف» با یک نام قوی به اسمبلی «ب» ارجاع می‌کند، نسخه‌ی اسمبلی «ب» در ورودی جدول AssemblyRef اسمبلی «الف» قرار گرفته است. این مورد باعث می‌شود زمانیکه CLR به بارگزاری اسمبلی «ب» احتیاج دارد، دقیقا می‌داند که چه نسخه‌ای با اسمبلی «الف» ساخته و تست شده است . البته این احتمال وجود دارد که CLR نسخه‌ای متفاوت از اسمبلی را با Binding Redirect بار کند که ادامه‌ی این مباحث در فصل سوم دنبال می‌شود.