

في عالم البرمجة على نظام Windows، مفهوم الـ **Compilation** كان دائماً جزءاً أساسياً من دورة تطوير البرمجيات. لكن مع ظهور الـ **.NET Framework**، حصل تغيير كبير في الطريقة التي نتعامل بها التطبيقات مع الكود. في المقال ده، هنشرح بشكل مبسط الفرق بين الـ **Compilation** قبل ظهور الـ **.NET Framework**، وبعد ما بقى جزء أساسي من تطوير التطبيقات.

Before .NET Framework (Pre-.NET Era):

زمان، لما كنا نكتب كود بلغة زي ++C أو VB6، كنا بنحتاج نعمل **Compilation** مباشر للكود إلى **Native Code** (كود يفهمه المعالج مباشرة). وده معناه:

- الكود بيتحول بالكامل إلى ملفات تنفيذية زي **exe** أو **dll**.
- الملفات دي بتشتغل مباشرة على نظام التشغيل من غير أي وسيط.
- الكود بيكون **optimized** للمعالج اللي اتكتب عشانه، بس في المقابل:
 - صعب نقل التطبيق من جهاز للتاني من غير مشاكل.
 - التوافق بين الإصدارات والأنظمة بيكون محدود.

مع .NET Framework:

لما ظهر الـ **.NET Framework**، ما استخدمش نفس فكرة الـ **Compilation** اللي كانت مستخدمة قبل كده. بدل ما الكود يتحول مباشرة لـ **Native Code**، حصلت مرحلة جديدة في النص:

1. Compilation to IL (Intermediate Language):

- الكود بيتحول الأول إلى **IL code**، وده عبارة عن كود متوسط مش مرتبط بأي معالج.
- الملف الناتج بيكون بصيغة **exe** أو **dll**، بس جواه مش **Native Code**... جواه **IL**.

2. JIT Compilation (Just-In-Time):

- أول ما التطبيق يشتغل، **CLR (Common Language Runtime)** بيبدأ يحول **IL** إلى **Native Code** في وقت التشغيل.
- العملية دي اسمها **Just-In-Time Compilation**.
- كل جزء من الكود بيتحول أول مرة يتنفذ فيها، مش كله مرة واحدة.

💡 طيب إيه فائدة التغيير ده؟

- المرونة والتوافق: الكود المكتوب بلغة زي C# ممكن يشتغل على أي جهاز فيه CLR.
- إدارة الذاكرة: مع وجود Garbage Collector، مش بنقلق على إدارة الذاكرة يدويًا.
- أمان أكثر: CLR بيقدم مستويات حماية إضافية زي Code Access Security.
- سهولة الصيانة والتحديث: IL أسهل في الـ Debugging وإعادة الترجمة.

✅ الخلاصة:

ظهور **NET Framework** غيّر شكل عملية الـ **Compilation** بالكامل. بدل ما نترجم الكود مباشرة للمعالج، بقينا بنعتمد على فكرة **IL + JIT Compilation**، وده وفر مرونة، أمان، وسهولة في التطوير والصيانة.

لو كنت بتشتغل على **NET**، مهم جدًا تكون فاهم إزاي الـ **Compilation** بيشتغل، لأنه بيأثر بشكل مباشر على أداء وتوافق التطبيق بتاعك.

لو المقال مفيد، شاركه مع زمابلك وخلينا نساعد بعض نفهم التكنولوجيا بشكل أوضح 😊
ولو عندك أي أسئلة أو محتوى تحب أكتبه، ابعثلي في الكومنت أو المسج.