

Just-In-Time (JIT) Compiler in C#

1. The JIT compiler translates the MSIL code of an assembly to native code and uses the CPU architecture of the target machine to execute a .NET application.
2. It also stores the resulting native code so that it is accessible for subsequent calls.
3. If a code executing on a target machine calls a non-native method, the JIT compiler converts the MSIL of that method into native code.
4. The JIT compiler also enforces type-safety in the runtime environment of the .NET Framework.
5. It checks for the values that are passed to parameters of any method.
6. The following are the various types of JIT compilation in .NET:
 - Pre - JIT.
 - Econo - JIT.
 - Normal - JIT.

Pre - JIT

1. In Pre-JIT compilation, complete source code is converted into native code in a single cycle (i.e. compiles the entire code into native code in one stretch)
2. This is done at the time of application deployment.
3. In .Net it is called "Ngen.exe"

Econo - JIT

1. In Econo-JIT compilation, the compiler compiles only those methods that are called at run time.
2. After execution of this method the compiled methods are removed from memory.

Normal - JIT

1. In Normal-JIT compilation, the compiler compiles only those methods that are called at run time.
2. After executing this method, compiled methods are stored in a memory cache.
3. Now further calls to compiled methods will execute the methods from the memory cache.