

The compiler for a .NET language takes a source code file and produces an output file called an assembly. The code in an assembly is not native machine code but CIL. When a .dll or .exe file has been created using any .NET language compiler, that binary file is known as assembly. As assembly contains CIL code, this is conceptually simple as Java byte code. In that it is not compiled to platform-specific instruction until necessary. In addition, CIL also contains metadata that describes characteristics of every type within the binary.

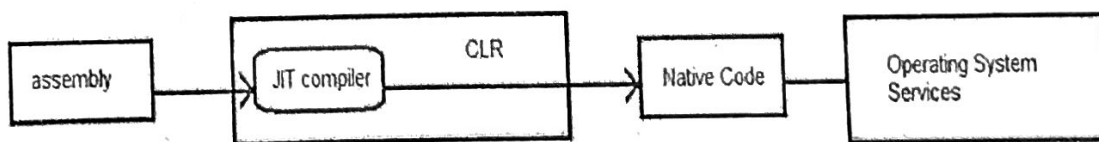
CIL, type metadata, and assemblies themselves are also described using metadata which is termed as manifest. The manifest contains information about the current version of the assembly, the culture information, and all externally referenced assemblies that are required for execution.

Compiling to Native Code and Execution

The program is not compiled to native machine code until it is called to run. CLR manages program's execution at runtime and performs the following steps:

- It checks the security characteristics of the assembly.
- It allocates space in memory.
- It sends the CIL to just-in-time (JIT) compiler which compiles it to native code.

The JIT compiler the CIL code in the assembly and only when needed and it's then cached. The code that is not called during execution is not compiled to native code. And the code that is called need only to compile once.



Compilation occurs at runtime.

Once CIL is compiled to native code, CLR manages it at runtime and perform tasks such as checking parameters, checking array bounds, and exception handling etc.