Metadata describes the contents in an assembly, whereas the manifest describes the assembly itself, providing the logical attributes shared by all the modules and all components in the assembly.
The manifest contains the assembly name, version number, locale and an optional strong name that uniquely identifying the assembly.
This manifest information is used by the CLR. The manifest also contains the security demands to verify this assembly.
It also contains the names and hashes of all the files that make up the assembly.
The .NET assembly manifest contains a cryptographic hash of different modules in the assembly. And when the assembly is loaded, the CLR recalculates the hash of the modules at hand, and compares it with the embedded hash.
If the hash generated at runtime is different from that found in the manifest, .NET refuses to load the assembly and throws an exception.

- The manifest is also generated automatically by the high-level compiler from the source files of all modules in the assembly.
- Manifest is embedded to only one physical file and only once since it is common for all the modules in an assembly, whereas the metadata needs to be embedded all the modules.
- The .NET CLR compatible compilers must generate the manifest and it should be in the standard format.
- Using the manifest .NET captures information about other referenced assemblies.
- This ensures version compatibility, and the assembly gets to interact with the exact trusted set of other assemblies it expects.
- The manifest contains every referenced assembly name, its public key (if a strong name is available), its version number, and locale. While running this assembly, .NET guarantees that only these specific assemblies are used, and that only compatible versions are loaded.

ıe Role of the Assembly Manifest

- The assembly also contains metadata that describes the assembly itself which is termed as manifest.
- Among other details all external assemblies are required by the current assembly to perform correctly, the assembly version number, copyright information etc.
- Compiler is responsible to generate assembly's manifest. Here is an example of the manifest generated by the compiler when compiling the Example.cs.