IT FACULTY

ASSEMBLIES

ROLE OF .NET ASSEMBLIES

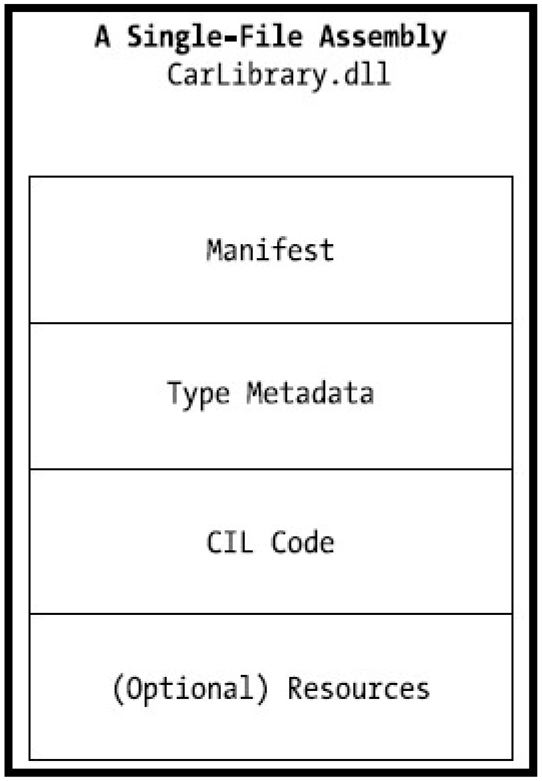
- Assemblies Promote Code Reuse;
- Assemblies Are Versionable Units;
- Assemblies Are Self-Describing;
- Assemblies Are Configurable.

FORMAT OF A .NET ASSEMBLY

- A .NET assembly (*.dll or *.exe) consists of the following elements:
  - A Win32 file header
  - A CLR file header
  - CIL code
  - Type metadata
  - An assembly manifest
  - Optional embedded resources

FORMAT OF A .NET ASSEMBLY

- Single-File Assemblies:
  - one-to-one correspondence between the (logical) assembly and the underlying (physical) binary.

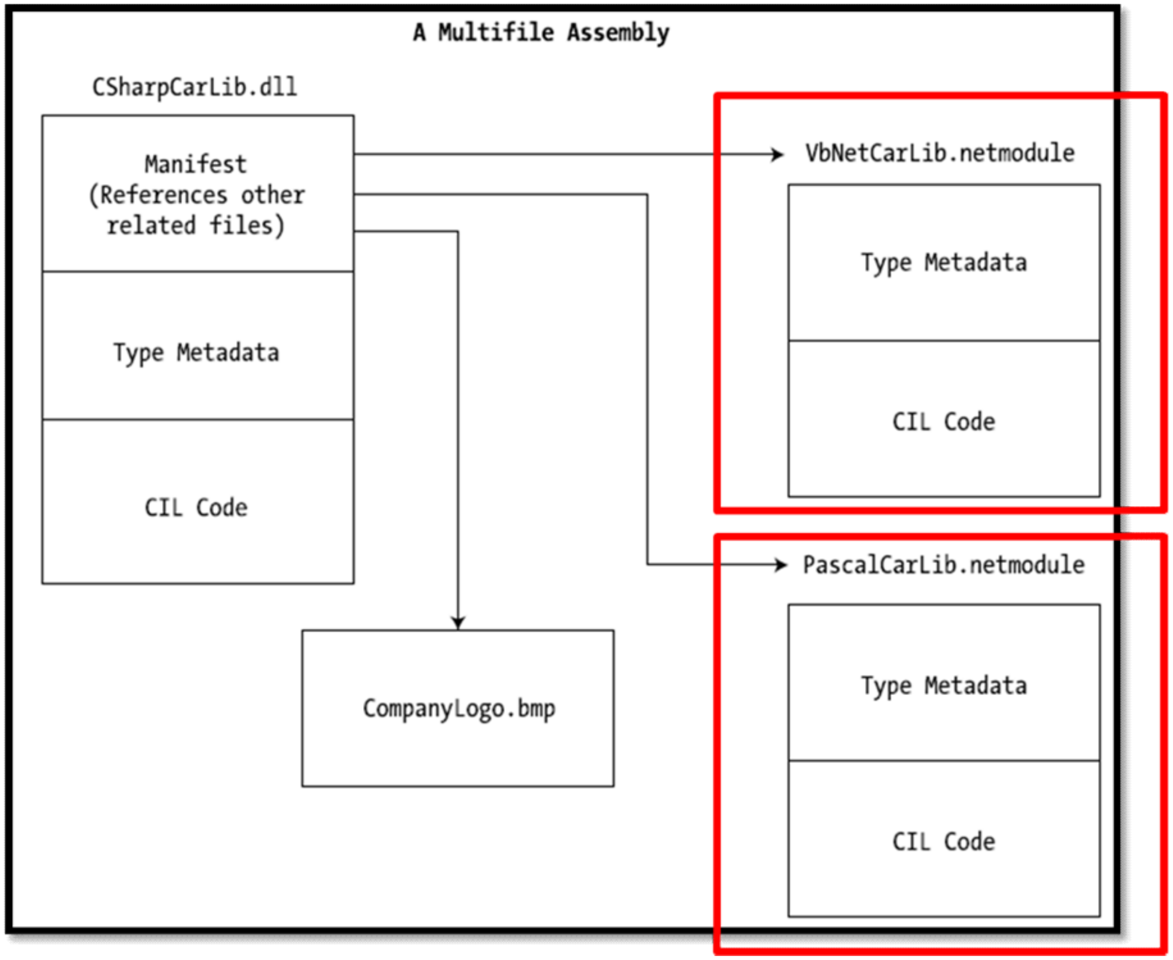


FORMAT OF A .NET ASSEMBLY

- Multi-file Assemblies:
  - A set of .NET *.dlls that are deployed and versioned as a single logic unit.
  - Primary module contains the assembly-level manifest (as well as any necessary CIL code, metadata, header information, and optional resources).
  - The modules are only logically related by information contained in the primary module's manifest.

FORMAT OF A .NET ASSEMBLY

- Multi-file Assemblies:



PRIVATE ASSEMBLIES

- Private assemblies are required to be located within the same directory as the client application (termed the application directory) or a subdirectory thereof.
- The Identity of a Private Assembly
  - The full identity of a private assembly consists of the friendly name and numerical version
  - The friendly name simply is the name of the module that contains the assembly's manifest

PRIVATE ASSEMBLIES

- Configuring Private Assemblies:
  - The CLR will not probe the subdirectory to look for reference assembly unless you supply a configuration file
  - Configuration files must have the same name as the launching application and take a *.config file extension, and they must be deployed in the client's application directory

SHARED ASSEMBLIES

- Like private assembly, a shared assembly is a collection of types and (optional) resources.
- Difference between shared and private assemblies is the fact that **a single copy of a shared assembly can be used by several applications on a single machine**.
- Shared assemblies are installed into the Global Assembly Cache (GAC) - **C:\Windows\Assembly**.
- **Executable assemblies (*.exe)** cannot be installed into the GAC.

SHARED ASSEMBLIES

- Understanding Strong Names:
  - An assembly must be assigned a **strong name** before being deployed to GAC.
  - **Strong name** is used to uniquely identify “the publisher” of a given .NET binary.
- A strong name is composed of a set of related data:
  - The **friendly name** of the assembly
  - The **version number** of the assembly (assigned using the [AssemblyVersion] attribute)
  - The **public key** value (assigned using the [AssemblyKeyFile] attribute)
  - An **optional culture identity** value for localization purposes (assigned using the [AssemblyCulture] attribute)
  - An **embedded digital signature** created using a hash of the assembly’s contents and the private key value

SHARED ASSEMBLIES

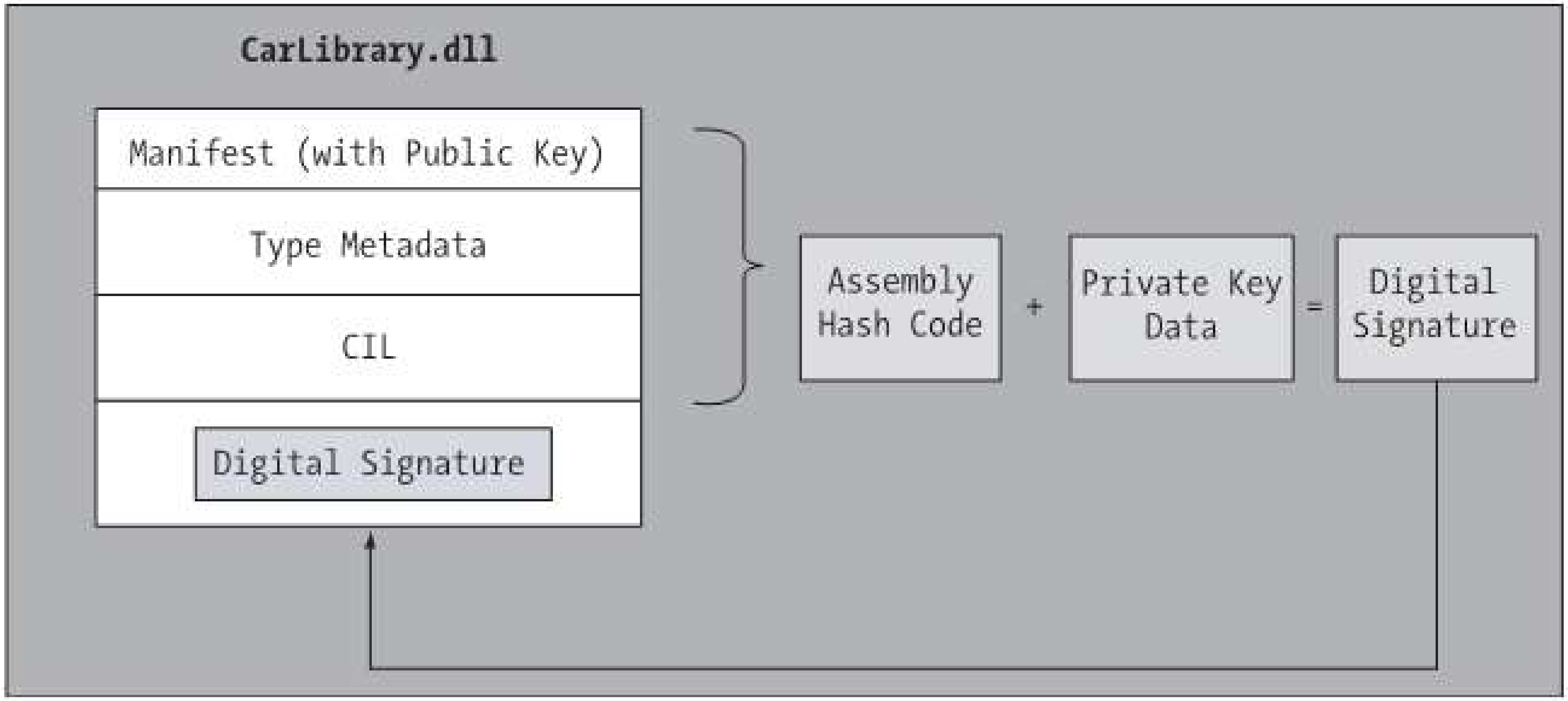


Figure 11-17. *At compile time, a digital signature is generated and embedded into the assembly based in part on public and private key data.*

Thank You !

IT FACULTY