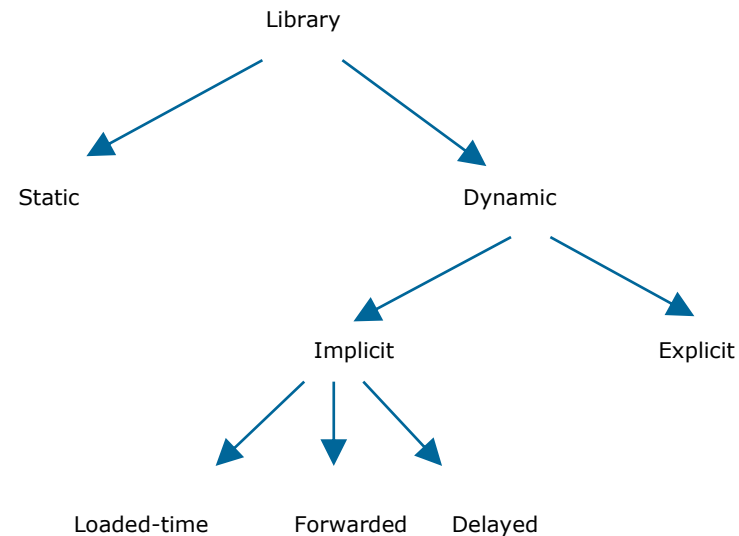
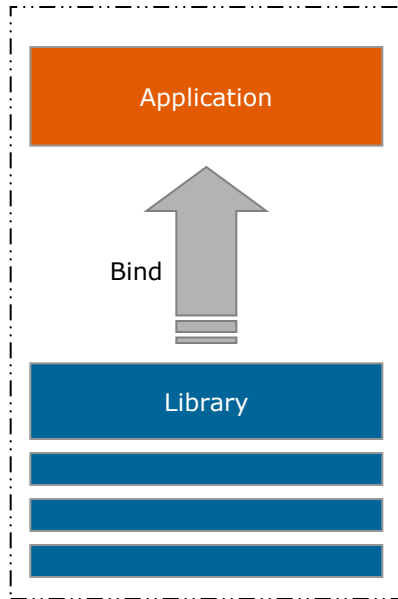


Introduction

- Cornerstone of Windows
- Reuse components
- Enable plugging mechanism
- Simplify project development
- Reduce system consumption
- Support localization
- Resolve platform differences
- Save testing/validation time

Libraries - Types

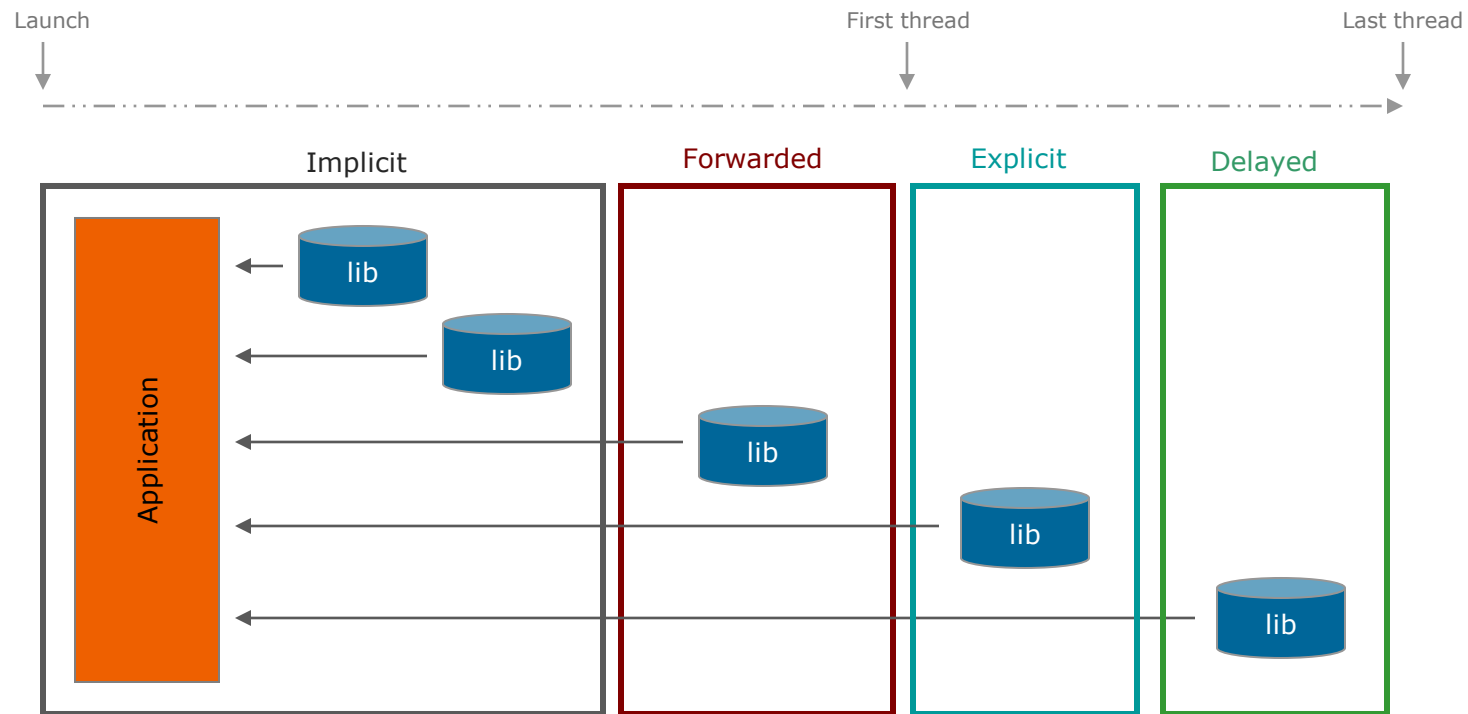
- Different types of libraries exist with different characteristics



Windows Dynamic-Link Libraries

Binding Types

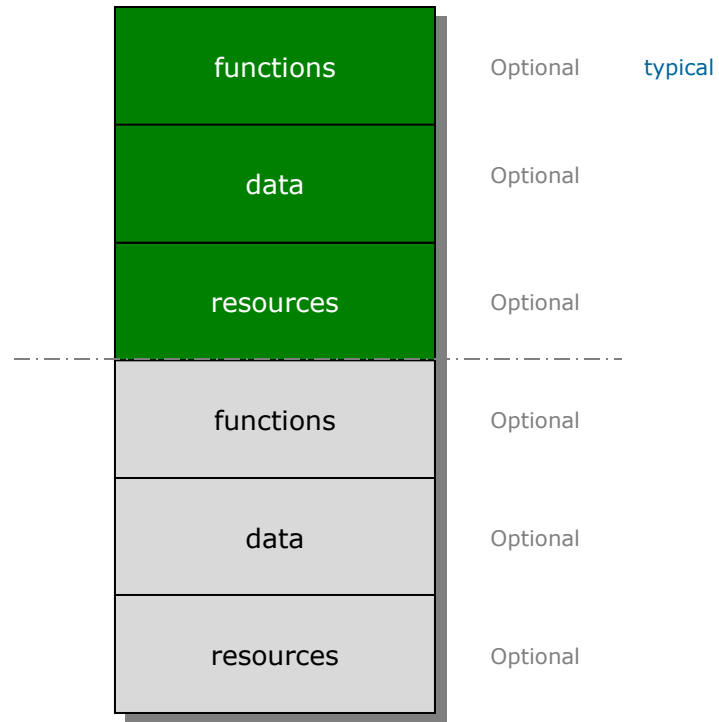
- Different binding types during a process's life-time



Windows Dynamic-Link Libraries

Components

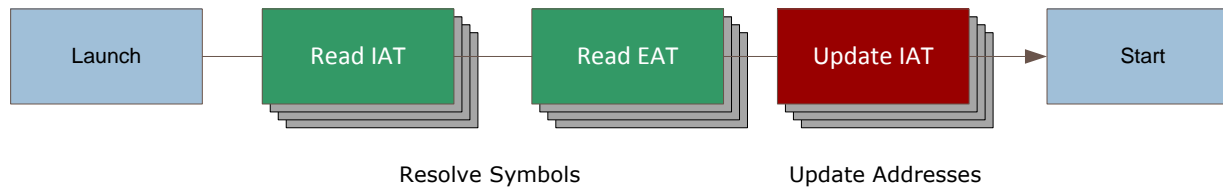
- Types of symbols
- Types of accesses



Windows Dynamic-Link Libraries

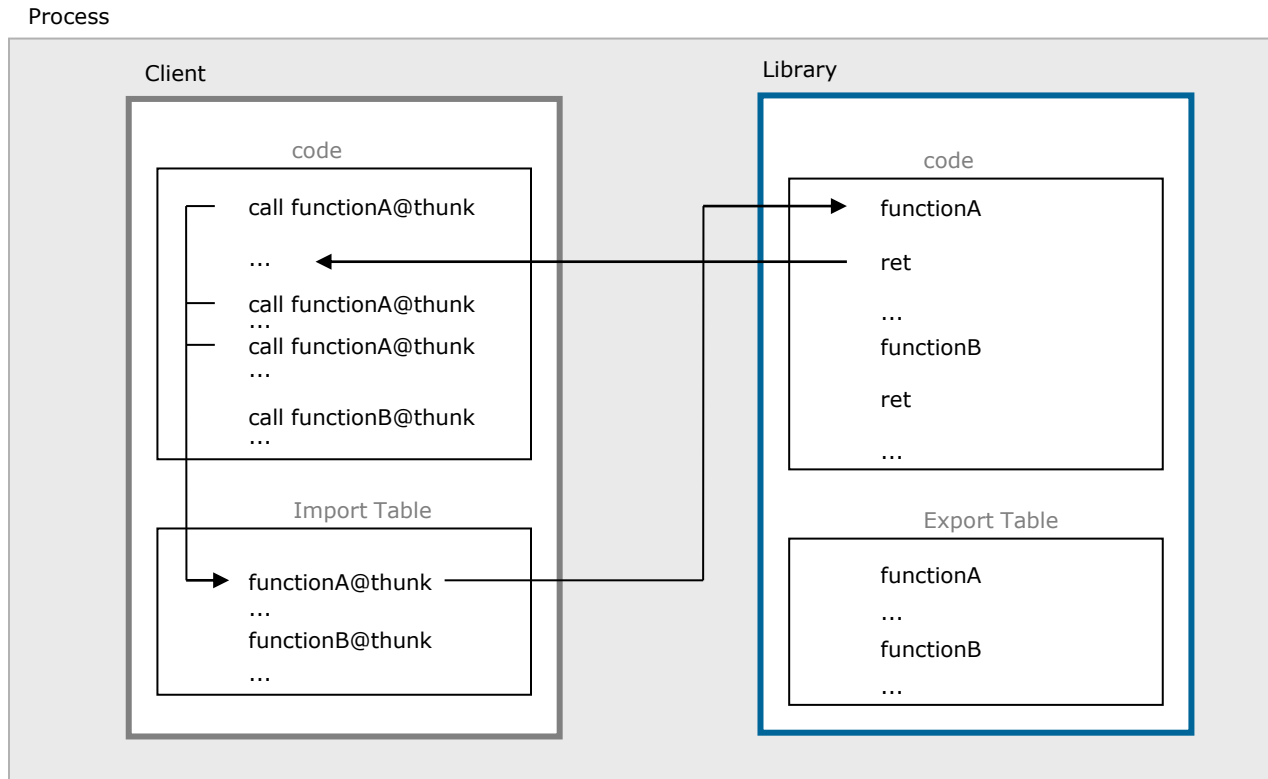
Implicit Linking

- Most common case
- Dependencies created during development
- Binding occurs when starting the client application



Implicit Linking

- Invoking methods

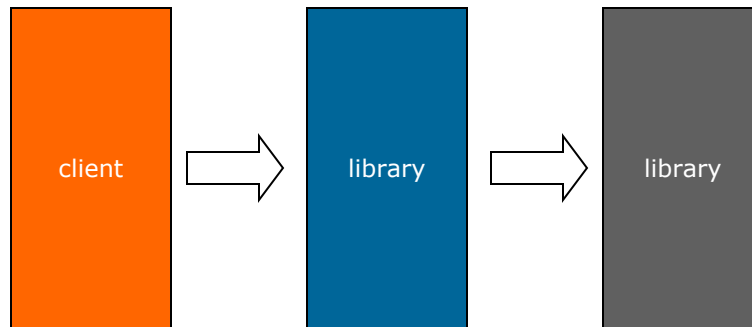


Explicit Linking

- Increase application portability
 - Library NOT found
 - Function is not found
 - Function signature is wrong
- Mechanism
 - LoadLibraryEx(...)
 - GetProcAddress(...)
 - Invoke function

Forwarded Library

- Delegate a call to another function of another library
- Mechanism



Delay Loaded Library

- Hybrid between implicit and explicit linking
- Reduce application loading time
- Avoid loading rarely used DLLs
- Declared during development

Entry Point

- Function implemented as a callback
 - Proceed to global initialization duties
 - Proceed to TLS initialization duties
 - Optional
 - Case sensitive
 - Informational

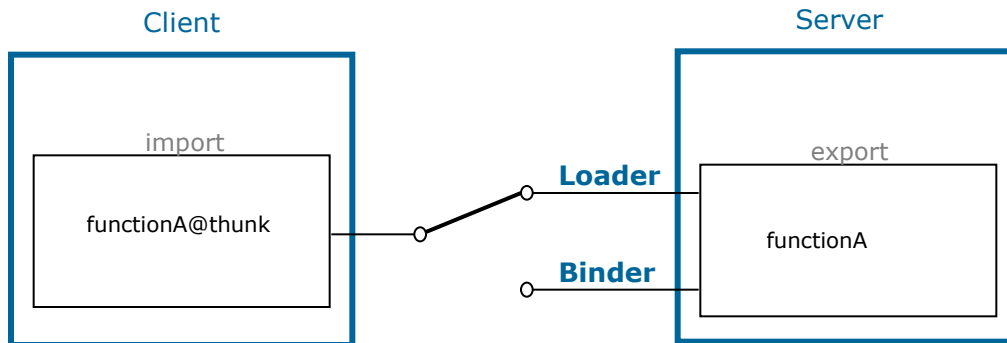
Performance - Rebasing

- Every module has a preferred base address
- Addresses conflict when loading several components
- Used at the end of the build cycle

Windows Dynamic-Link Libraries

Performance - Binding

- Loader resolves the addresses of the imported symbols
- Bind the application during the installation process
- Application must have been previously rebased



Windows Dynamic-Link Libraries

Issues

- Simple name-based dependencies
- Installing a product which overwrites a DLL file
- Solutions
 - WFP
 - Redirection
 - Known Directories
 - Known Libraries
 - WinSxS



Windows Dynamic-Link Libraries

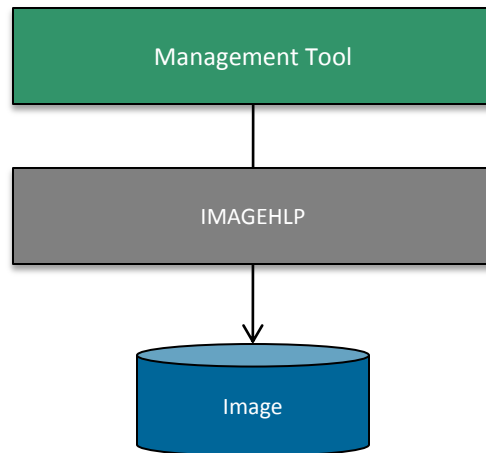
Manifest

- Allow different versions of the same DLL to exist “side-by-side”
- Typtes
 - Extern
 - Intern
- Assemblies
 - Private
 - Shared

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <assemblyIdentity name="Microsoft.Windows.Shell.explorer" processorArchitecture="x86" version="5.1.0.0" type="win32" />
  <description>Windows Shell</description>
  - <dependency>
    - <dependentAssembly>
      <assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls" version="6.0.0.0"
        processorArchitecture="x86" publicKeyToken="6595b64144ccf1df" language="*" />
    </dependentAssembly>
  </dependency>
</assembly>
```

Management

- Access the (some) parts of an image
 - Update the version
 - Manage the certificate
 - Edit the executable image



Thank you

- Questions?