

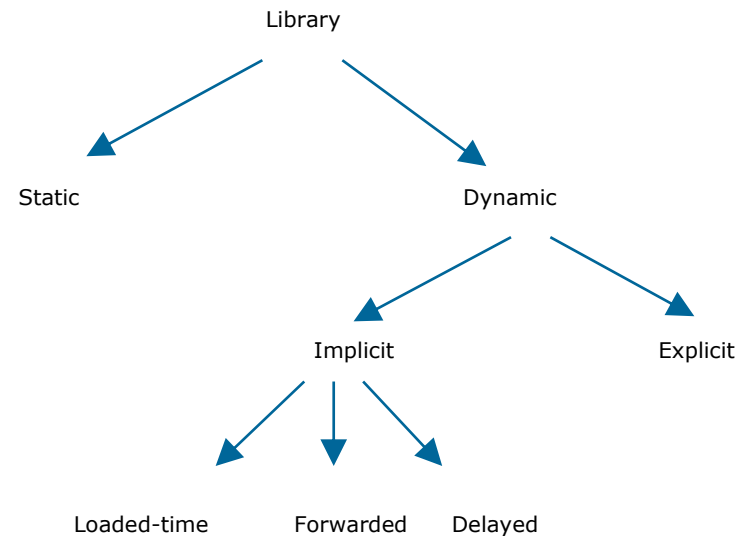
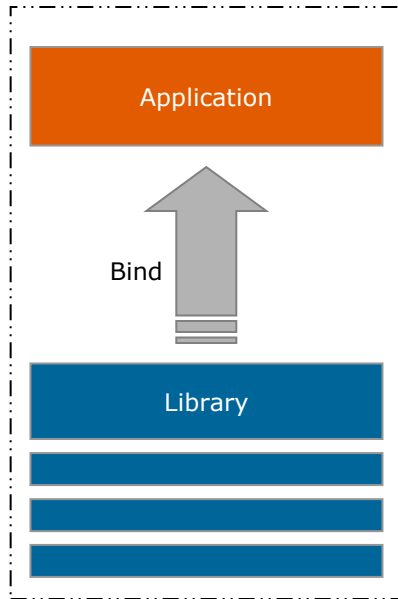
Introduction

- Cornerstone of Windows
- Reuse components
- Enable plug-in architecture
- Simplify project development
- Reduce system consumption
- Implement localization
- Resolve platform differences
- Save testing/validation time

Windows Dynamic-Link Libraries

Libraries - Types

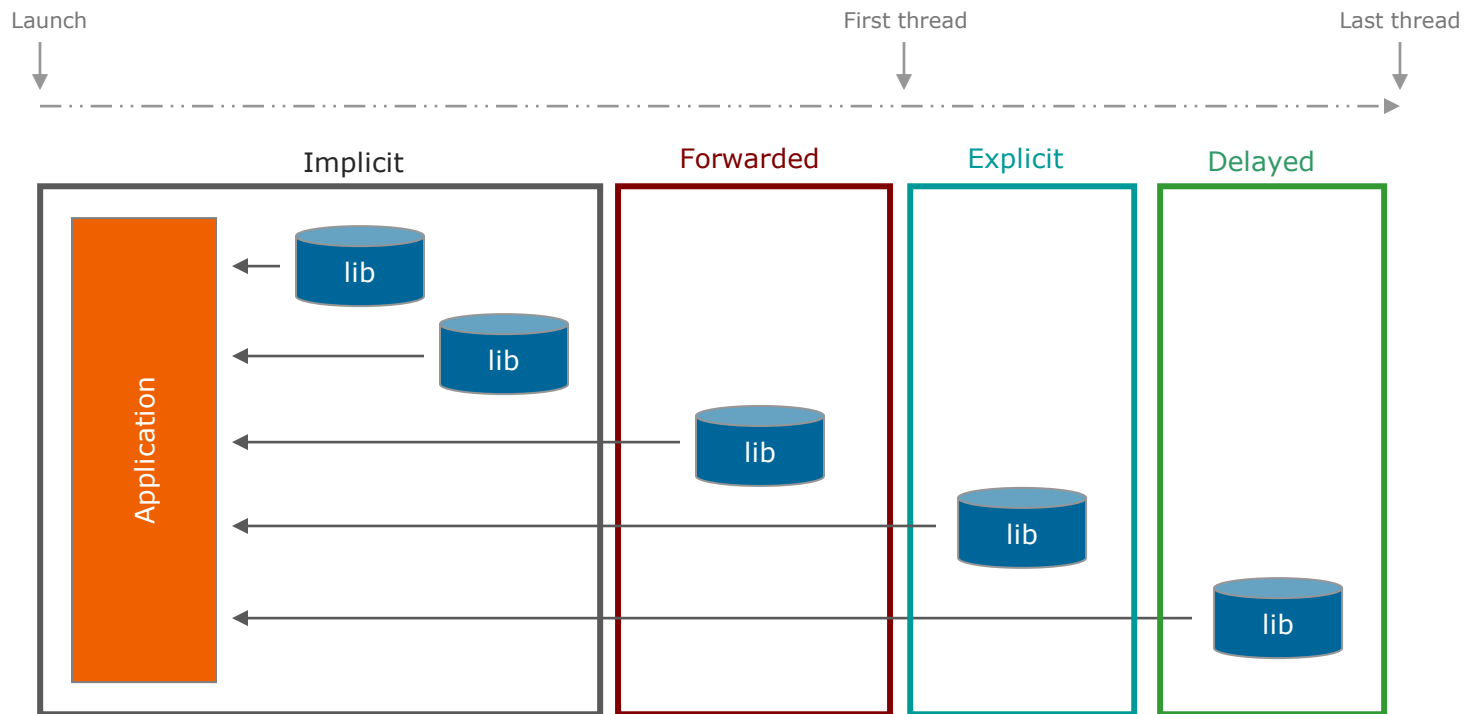
- Different types of libraries exist with different characteristics



Windows Dynamic-Link Libraries

Binding Types

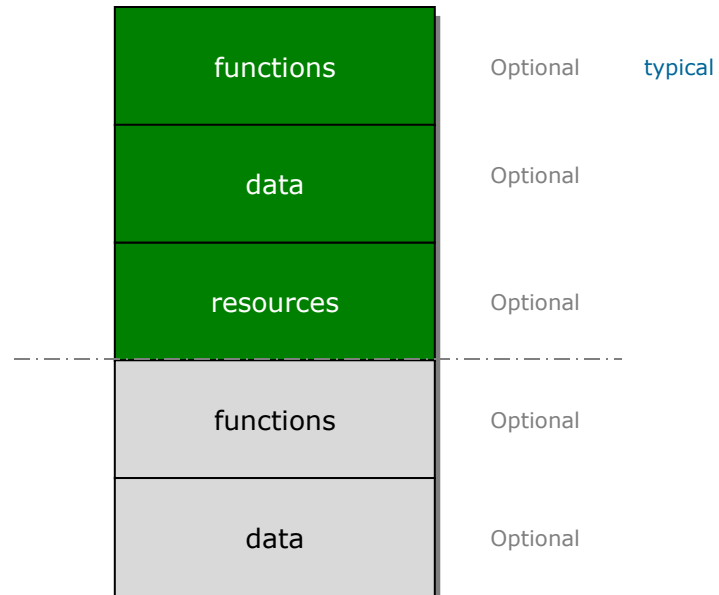
- Different binding types during a process's life-time



Windows Dynamic-Link Libraries

Components

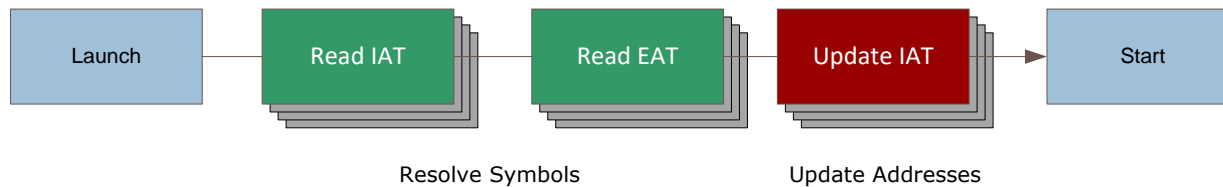
- Some components can be made public



Windows Dynamic-Link Libraries

Implicit Linking

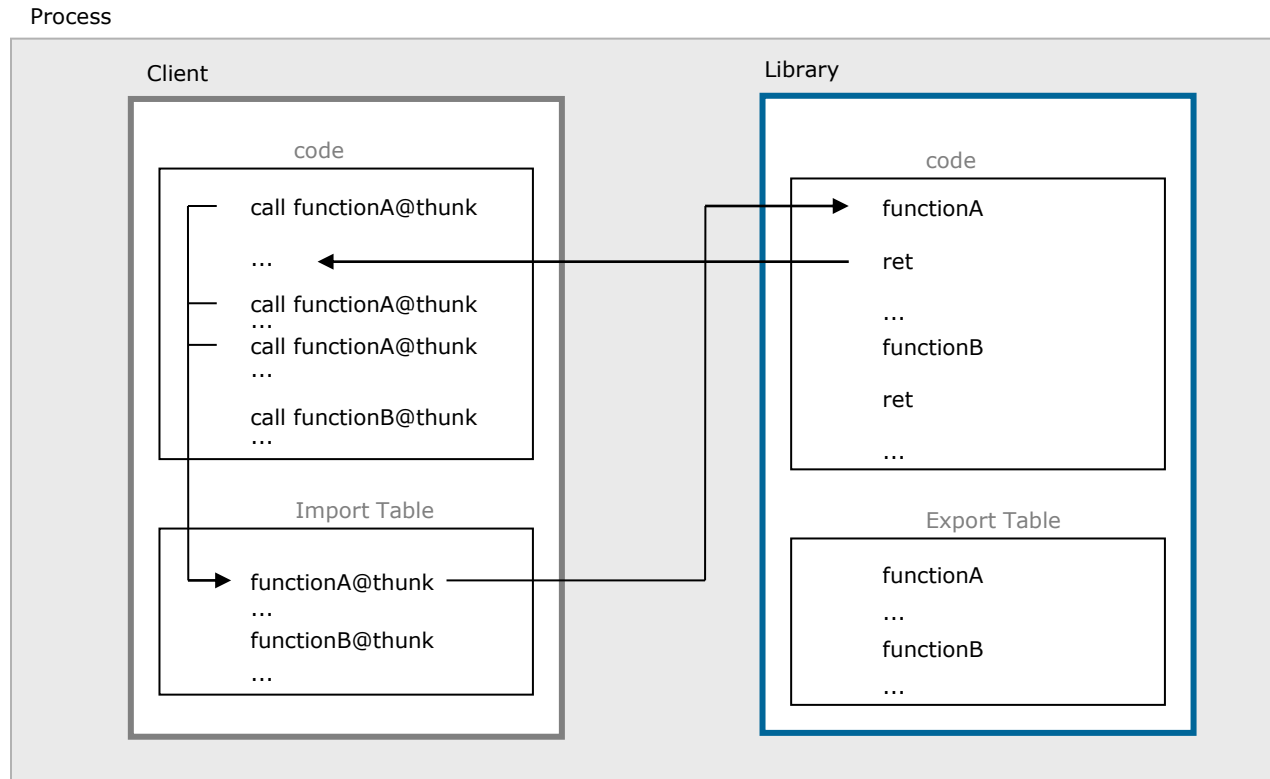
- Most common case
- Dependencies created during development
- Binding occurs when starting the client application



Windows Dynamic-Link Libraries

Implicit Linking

- Invoking methods

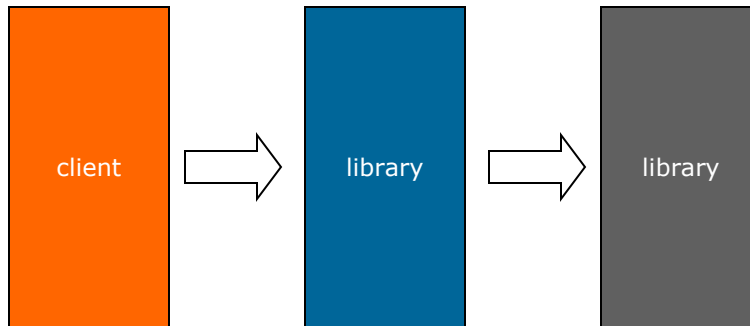


Explicit Linking

- 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

Windows Dynamic-Link Libraries

Entry Point

- Function implemented as a callback
 - Is optional
 - Is case sensitive
 - Is informational
 - TLS initialization

pestudio-9.62 (professional) - Malware Initial Assessment - www.winator.com | c:\temp\1b6a92ddd17950bc9fd3b80eb9730f53

file settings about

c:\temp\1b6a92ddd17950bc9fd3b80eb9730f53

- ... virustotal (offline)
- ... indicators (n/a)
- ... footprints (wait...)
- ... groups (count > 14)
- ... mitre (technique > 10)
- ... dos-header (size > 64 bytes)
- ... dos-stub (size > 80 bytes)
- ... rich-header (tools > Visual Studio 2005)
- ... file-header (dll > 64-bit)
- optional-header (subsystem > GUI)**
- ... directories (count > 4)
- ... sections (count > 6)
- ... libraries (group > execution)
- ... imports (flag > 35)
- ... exports (name > foo)
- ... thread-local-storage (n/a)
-NET (n/a)
- ... resources (signature > manifest)
- ... strings (technique > 10)
- ... debug (n/a)
- ... manifest (level > aslnvoker)
- ... version (n/a)
- ... certificate (n/a)
- ... overlay (entropy > zero)

property	value	detail
general		
subsystem	0x0002	GUI
magic	0x020B	PE+
file-checksum	0x00017D13	0x00017E12 (expected)
entry-point > location (file-offset)	0x00003A64	section[.text]
base-of-code > location (file-offset)	0x00001000	section[.text]
size-of-code	0x00008000	32768 bytes
size-of-initialized-data	0x00004400	17408 bytes
size-of-uninitialized-data	0x00000000	0 bytes
size-of-image	0x00011000	69632 bytes
size-of-Headers	0x00000400	1024 bytes
size-of-stack-reserve	0x00100000	1048576 bytes
size-of-stack-commit	0x00001000	4096 bytes
size-of-heap-reserve	0x00100000	1048576 bytes
size-of-heap-commit	0x00001000	4096 bytes
section-alignment	0x00001000	4096 bytes
file-alignment	0x00000200	512 bytes
directories > count	0x00000010	16
LoaderFlags	0x00000000	0x00000000
Win32VersionValue	0x00000000	0x00000000
image-base	0x00000000180000000	0x00000000180000000
linker > version	9.0	Microsoft Linker 9.0
os > version	5.2	Windows Server 2003 R2

sha256 > 087A137229B9E63DEA45EEDFE2A841BCCEE79A0A649A451B54C6CA7488CA835AC size > 51455 bytes entropy > 5.94 type > dynamic-link-library

pestudio – www.winator.com

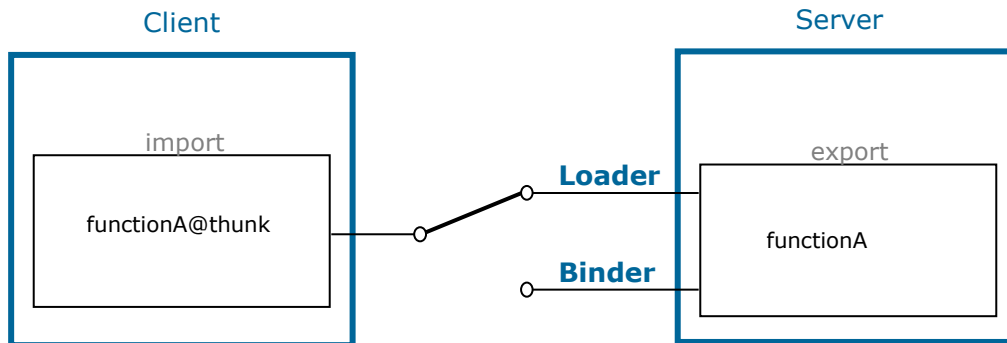
Performance - Rebasing

- 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

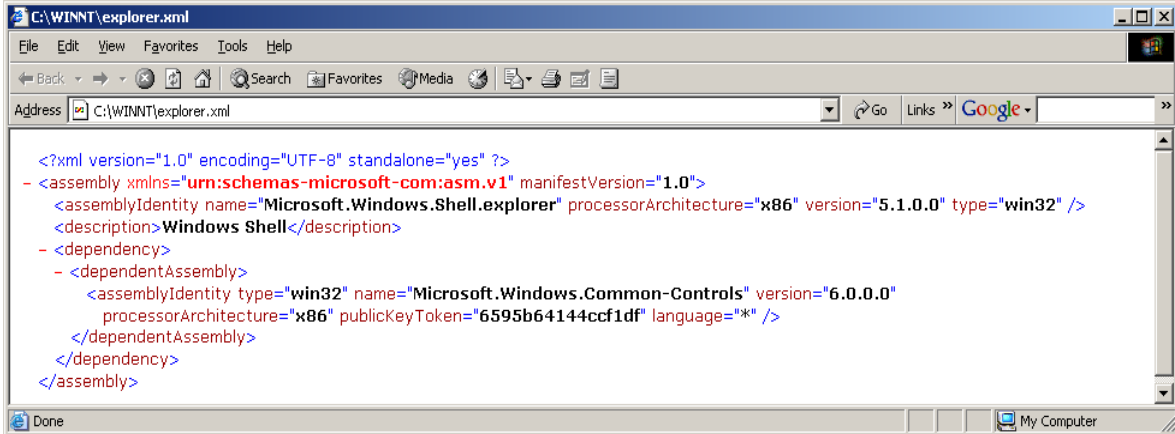
- Simple name-based dependencies
- Installing a product which overwrites a DLL file
- Solutions
 - WFP
 - Redirection
 - Known Directories
 - Known Libraries
 - WinSxS



Windows Dynamic-Link Libraries

Manifest

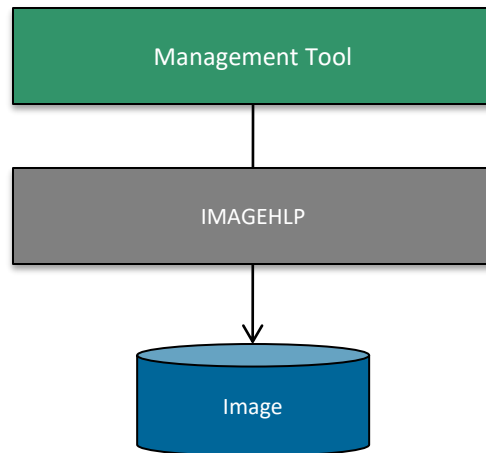
- Allow different versions of the same DLL to exist “side-by-side”
- Typtes
 - Extern
 - Intern
- Assemblies
 - Private
 - Shared



```
<?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



Windows Dynamic-Link Libraries

Difference between executable and DLL

- Executable vs. Dynamic-Link Library

Executable	DLL
IMAGE_FILE_EXECUTABLE (0x2)	IMAGE_FILE_DLL (0x2000)
Entry point is mandatory	Entry point is optional
Usually without exported functions	Often with exported functions
Code is mandatory	Code is optional
Can host and can be hosted	Must be hosted
Own separated address space	Shared address space
Unhandled exception crashes process	Unhandled exception crashes host

Convert a DLL into an Executable

- A DLL can be converted into an Executable (e.g. to ease debugging)
 - Modify PE Characteristic: `IMAGE_FILE_EXECUTABLE` > `IMAGE_FILE_DLL`
 - Modify the existing entry-point to an exported function

References

- Dynamic-Link Library Entry-Point Function
 - <https://docs.microsoft.com/en-us/windows/desktop/DLLs/dynamic-link-library-entry-point-function>
- DllMain entry point
 - <https://docs.microsoft.com/en-us/windows/desktop/DLLs/dllmain>