📚 Linking Guide: wxWidgets, Boost.Signals, and SFML

> 📘 **Note**: For wxWidgets linking, please refer to the wxWidgets Installation Manual

🔧 0. VCPKG Environment Setup

Locate VCPKG Installation

1. Open Command Prompt
2. Run:

```
where vcpkg
```

3. Note the path (typically like `C:\dev\vcpkg\vcpkg.exe`)

Set VCPKG_ROOT Environment Variable

1. Press `Windows + S`
2. Type "environment variables"
3. Click "Edit the system environment variables"
4. Click "Environment Variables" button
5. Under "System variables", click "New"
6. Set:
   - Variable name: `VCPKG_ROOT`
   - Variable value: Your vcpkg path (e.g., `C:\dev\vcpkg`) - remove `\vcpkg.exe`
7. Click "OK" on all windows

Verify Environment Variable

```
echo %VCPKG_ROOT%
```

> 🚨 **CRITICAL**: Restart Visual Studio after setting environment variables

📥 1. Library Installation

Dynamic Library

```
# Install Boost.Signals2 (part of Boost)
vcpkg install boost-signals2:x64-windows

# Install SFML
vcpkg install sfml:x64-windows
```

Static Library

```
vcpkg install boost-signals2:x64-windows-static
vcpkg install sfml:x64-windows-static
```

✅ 2. Verify Installation

```
vcpkg list boost-signals2
vcpkg list sfml
```

> ⚠️ **Warning**: Ensure all libraries are listed before proceeding

🔨 3. Project Configuration

Include Directories Setup

> 🚨 **CRITICAL**: Make sure to select "All Configurations" and "All Platforms"!

1. Right-click project in Solution Explorer → Properties
2. At the top of Properties window:
   - Configuration dropdown: Select "All Configurations"
   - Platform dropdown: Select "All Platforms"
3. Navigate to:
   - C/C++ → General → Additional Include Directories
4. Click dropdown arrow → Edit
5. Add new line:

```
$(VCPKG_ROOT)\installed\x64-windows\include
```

> 💡 **Tip**: Verify path by pasting in File Explorer: `%VCPKG_ROOT%\installed\x64-windows\include`

Library Directories Setup

Debug Configuration (x64)

1. Configuration: Debug, Platform: x64
2. Linker → General → Additional Library Directories
3. Add:

```
D:\vcpkg\installed\x64-windows\debug\lib
```

Release Configuration (x64)

1. Configuration: Release, Platform: x64
2. Linker → General → Additional Library Directories
3. Add:

```
$(VCPKG_ROOT)\installed\x64-windows\lib
```

Dependencies Configuration

Debug Configuration (x64)

Add to Linker → Input → Additional Dependencies:

```
sfml-graphics-d.lib
sfml-window-d.lib
sfml-system-d.lib
```

Release Configuration (x64)

Add to Linker → Input → Additional Dependencies:

```
sfml-graphics.lib
sfml-window.lib
sfml-system.lib
```

📁 4. Required DLLs

> 🚨 **CRITICAL**: Copy these files to `Project Folder > x64 > debug`

Look in these folders (replace `D:\vcpkg` with your actual VCPKG installation path):

- `%VCPKG_ROOT%\installed\x64-windows\debug\bin`
- `%VCPKG_ROOT%\installed\x64-windows\debug\lib`

> ⚠️ **Important**: The path may differ if your VCPKG is not installed in the D drive. Use the path where your VCPKG is actually installed. You can verify your VCPKG path by:
>
> 1. Opening Command Prompt
> 2. Running: `echo %VCPKG_ROOT%`

Debug Libraries: (Check both bin and lib folders)

- bz2d.dll
- freetype6d.dll
- libpng16d.dll
- zlibd1.dll
- sfml-audio-d.lib
- sfml-audio-d-2.dll
- sfml-graphics-d.lib
- sfml-graphics-d-2.dll
- sfml-network-d.lib
- sfml-network-d-2.dll
- sfml-system-d.lib
- sfml-system-d-2.dll
- sfml-window-d.lib
- sfml-window-d-2.dll

🔍 5. Common Issues and Solutions

Missing DLLs

- ✅ Verify all required DLLs are in correct output directory
- ✅ Ensure using debug (-d) versions for Debug configuration
- ✅ Check VCPKG_ROOT path setting

Linker Errors

- ✅ Confirm all library paths are correct

- ✅ Verify correct library versions (debug vs. release)
- ✅ Check dependency order in linker settings

Build Configuration Mismatches

- ✅ Use consistent configuration (static/dynamic)
- ✅ Match SFML configuration with project runtime library settings
- ✅ Verify platform (x64) matches across all settings