

# クロスプラットフォーム P/Invoke コードの作成

2022/01/29 •

## ライブラリ名のバリエーション

クロスプラットフォームの P/Invoke コードをさらに単純にするために、正規の共有ライブラリ拡張機能 ( `.dll`、`.so` または `.dylib` ) が、ランタイムによってネイティブライブラリ名に追加されます。また、Linux と macOS では、ランタイムによって `lib` の付加が試みられます。これらのライブラリ名のバリエーションは、アンマネージド ライブラリを読み込む API (`DllImportAttribute` など) を使用すると、自動的に検索されます。

### NOTE

ライブラリ名の絶対パス ( `/usr/lib/libc` など) はそのままの状態扱われ、バリエーションは検索されません。

P/Invoke を使用する次の例を考えてみましょう。

```
[DllImport("nativdep")]
static extern int ExportedFunction();
```

Windows で実行している場合、DLL は次の順序で検索されます。

1. `nativdep`
2. `nativdep.dll` (ライブラリ名の末尾が `.dll` または `.exe` ではない場合)

Linux または macOS で実行されている場合、ランタイムによって、`lib` の付加、および正規の共有ライブラリ拡張機能の追加が試みられます。これらの OS では、ライブラリ名のバリエーションは次の順序で試行されます。

1. `nativdep.so` / `nativdep.dylib`
2. `libnativdep.so` / `libnativdep.dylib`<sup>1</sup>
3. `nativdep`
4. `libnativdep`<sup>1</sup>

Linux では、ライブラリ名の末尾が `.so`、またはライブラリ名に `.so` (末尾の `.` に注意) が含まれている場合に、検索の順序が異なります。次の例を確認してください。

```
[DllImport("nativdep.so.6")]
static extern int ExportedFunction();
```

この場合、ライブラリ名のバリエーションは、次の順序で試みられます。

1. `nativdep.so.6`
2. `libnativdep.so.6`<sup>1</sup>
3. `nativdep.so.6.so`
4. `libnativdep.so.6.so`<sup>1</sup>

<sup>1</sup> パスは、ライブラリ名にディレクトリの区切り文字 ( `/` ) が含まれていない場合にのみ確認されます。

## カスタム インポート リゾルバー

より複雑なシナリオでは、実行時での DLL のインポートを解決する、[SetDllImportResolver](#) を使用できます。次の例では、CPU がサポートする場合に、`nativdep` が `nativdep_avx2` に解決されます。

#### TIP

この機能は、.NET 5 以降、および .NET Core 3.1 以降でのみ使用できます。

```
using System;
using System.Reflection;
using System.Runtime.InteropServices;

namespace PInvokeSamples
{
    public static class Program
    {
        [DllImport("nativdep")]
        private static extern int ExportedFunction();

        public static void Main(string[] args)
        {
            // Register the import resolver before calling the imported function.
            // Only one import resolver can be set for a given assembly.
            NativeLibrary.SetDllImportResolver(Assembly.GetExecutingAssembly(), DllImportResolver);

            int value = ExportedFunction();
            Console.WriteLine(value);
        }

        private static IntPtr DllImportResolver(string libraryName, Assembly assembly, DllImportSearchPath? searchPath)
        {
            if (libraryName == "nativdep")
            {
                // On systems with AVX2 support, load a different library.
                if (System.Runtime.Intrinsics.X86.Avx2.IsSupported)
                {
                    return NativeLibrary.Load("nativdep_avx2", assembly, searchPath);
                }

                // Otherwise, fallback to default import resolver.
                return IntPtr.Zero;
            }
        }
    }
}
```

## 関連項目

- プラットフォーム呼び出し (P/Invoke)