<http://support.microsoft.com/kb/310674>

**HOWTO: Managed Visual C++ 프로젝트에 대한 참조 추가**

이 문서에서는 Visual C++ .NET 프로젝트의 다른 어셈블리에 대한 참조를 추가하는 프로세스를 설명합니다. C# .NET 같은 다른 .NET 언어에서는 **참조 추가** 대화 상자를 통해 참조를 추가할 수 있습니다. 하지만 Managed C++ 응용 프로그램에는 이 대화 상자를 사용할 수 없습니다. Managed C++ 응용 프로그램에서 참조를 보다 쉽게 사용할 수 있는 팁이 몇 가지 있습니다. 

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/310674#top)위로 가기](http://support.microsoft.com/kb/310674#top)

**.NET 참조**

.NET 참조는 공유 어셈블리를 가리킵니다. 예를 들어, System.Windows.Forms.dll 어셈블리는 Windows Forms 클래스에 액세스하기 위한 표준 어셈블리입니다. Managed C++ 응용 프로그램에서 이 어셈블리를 사용하려면 아래에 나와 있는 것처럼 #using 전처리기 지시문을 사용하여 이 어셈블리를 참조해야 합니다.

#using <System.Windows.Forms.dll>

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/310674#top)위로 가기](http://support.microsoft.com/kb/310674#top)

**COM 참조**

Managed C++ 응용 프로그램에서 COM 개체를 사용하는 것은 디자인 결정 사항과 관련이 있습니다. 디자인과 관련하여 결정할 수 있는 사항 중 하나는 관리되는 클래스 대신 관리되지 않는 COM 코드를 사용하는 것입니다. 예를 들어 기존의 #import 솔루션을 사용하기로 결정할 수 있습니다. COM Interop를 사용하는 데 문제가 있는 경우에는 이렇게 하는 것이 좋습니다.   
  
또 다른 선택은 COM 개체가 포함된 interop 어셈블리를 사용하는 것입니다. 이 방법은 C# .NET 및 Visual Basic .NET 같은 기타 .NET 언어에만 사용할 수 있습니다. COM 개체에 대한 interop 어셈블리를 만들려면 TLBIMP.exe 도구("참조" 절 참조)를 사용하십시오. 예들 들어 관리되는 응용 프로그램에서 Microsoft Internet Explorer를 자동화하려면 다음과 같이 하십시오.

1. 명령 프롬프트를 엽니다.
2. Windows System 폴더로 이동합니다.
3. 다음 명령을 입력합니다.

**tlbimp shdocvw.dll /out:Interop.shdocvw.dll**

1. Interop.shdocvw.dll을 사용자 프로젝트 폴더로 이동합니다.

이렇게 하면 Shdocvw.dll에서 COM 개체에 대한 interop 어셈블리가 만들어집니다. 만들어진 파일, Interop.shdocvw.dll을 #using 지시문에서 사용할 수 있습니다. 그러면 이를 관리 구성 요소로 취급할 수 있습니다. 이러한 동적 연결 라이브러리(DLL)를 출력 폴더로 자동으로 복사하기 위한 지침은 이 문서의 "빌드 후 이벤트 사용" 절을 참조하십시오.   
  
**참고:** TLBIMP.exe를 인식하기 위해서는 Visual C++ .NET 환경 변수를 설정해야 합니다. 환경 변수를 설정하지 않은 경우에는 먼저 ./VC7/BIN/VCVARS32.bat을 실행해야 합니다. 

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/310674#top)위로 가기](http://support.microsoft.com/kb/310674#top)

**프로젝트 참조**

프로젝트 참조는 다른 프로젝트가 만든 어셈블리에 대한 참조를 말합니다. 이러한 어셈블리를 참조하려면 #using 지시문을 사용합니다. 하지만, 이러한 어셈블리는 공유되지 않으므로 컴파일러가 이들을 찾을 수 있도록 조치를 취해야 합니다. 이렇게 하는 방법은 두 가지입니다.

* 프로젝트 폴더로 어셈블리를 복사하는 방법 -또는-
* -또는- 어셈블리를 찾을 수 있도록 프로젝트 설정을 변경하는 방법
  1. 해당 프로젝트의 **속성 페이지** 대화 상자를 엽니다.
  2. **C/C++** 폴더를 누릅니다.
  3. **일반** 속성 페이지를 누릅니다.
  4. 대상 어셈블리가 포함되어 있는 폴더를 가리키도록 **Resolve #using References** 속성을 수정합니다.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/310674#top)위로 가기](http://support.microsoft.com/kb/310674#top)

**빌드 후 이벤트 사용**

개인 어셈블리는 자신을 사용할 실행 파일과 같은 폴더에 있어야 합니다. C# .NET이나 Visual Basic .NET에서 참조를 추가하면 개인 어셈블리는 출력 폴더로 자동으로 복사됩니다. Managed C++ 응용 프로그램에서도 빌드 후 이벤트(post-build event)를 사용하면 이러한 단계를 자동화할 수 있습니다. "TestApp"라는 Managed C++ 응용 프로그램의 프로젝트 폴더에 "MYLIB.dll" 어셈블리가 있는 시나리오를 예로 들어 설명합니다. 다음 단계를 수행하면 이 DLL을 TestApp 프로젝트의 출력 폴더로 복사하는 빌드 후 이벤트가 설정됩니다.

1. Managed C++ 프로젝트의 **속성 페이지** 대화 상자를 엽니다.
2. **빌드 이벤트** 폴더를 누릅니다.
3. **빌드 후 이벤트** 속성 페이지를 누릅니다.
4. 다음 명령의 **Command Line** 속성을 수정합니다.

**copy $(*<ProjectDir>*)mylib.dll $(*<TargetDir>*)**

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/310674#top)위로 가기](http://support.microsoft.com/kb/310674#top)