



Visual C++에 온 것을 환영한다. 21장에 걸쳐, Windows 운영체제에서 실행되는 진보된 응용 프로그램을 만들기 위해 Microsoft 사가 제시한 C++ 개발 환경과 Microsoft 사가 새로 발표한 .NET 플랫폼에서 응용 프로그램을 만들기 위한 간략한 소개 정도를 배울 것이다. Microsoft 개발자들은 Visual C++ 이면에 세계적인 C++ 컴파일러와 개발 환경을 만들고, 개발자들이 Windows 응용 프로그램을 쉽고 빠르게 제작할 수 있도록 이전에 본 적이 없는 도구 모음을 제공하려고 결정하였다. 첫 번째 버전 이후로, Microsoft는 Visual C++의 일부인 도구들을 Windows 응용 프로그램에 좀더 쉽게 개발할 수 있도록 지속적으로 개량하였다. Microsoft 사가 새 기술들을 Windows 플랫폼에 소개할 때마다, 응용 프로그램에 쉽게 적용할 수 있도록 그 도구들을 Visual C++ 모음에 포함시켜 나갔다.

C++을 처음 접한다고 겁내지 않아도 된다. Visual C++ 도구를 이용하여 응용 프로그램을 만드는 방법을 배우는 동안 C++ 프로그래밍 언어를 배울 수 있도록 충분히 쉽게 설명하려고 노력하였다. 이 책에서 C++ 언어 자체에 대한 완벽한 내용을 모두 설명할 수는 없지만, 책 전반에 걸쳐 언어에 대한 이해가 필요할 때마다 설명하려고 노력하였다.

C++ 부가 설명 : 특정 프로그래밍 도움말

이 책을 보면서, 예제 코드에 있는 C++ 프로그래밍 언어의 다양한 측면을 설명하기 위한 특별히 독립된 절을 볼 수 있을 것이다. 이미 C++ 프로그래밍 언어에 익숙하고, 단지 이 책을 통해 Visual C++ 도구에 대해서만 배우고 싶다면, 이와 같은 절들은 넘어가도 좋다.

MFC 노트 : 클래스의 빠른 이해

이 책의 이전 판인 “초보자를 위한 Visual C++ 6 21일 완성”을 봤었다면, 많은 부분이 재작성되었다는 것을 알 수 있을 것이다. 새 버전에서는 Visual C++로 응용 프로그램을 만드는데 사용하는 다양한 도구와 리소스들을 소개하는 것에 그치지 않고, 각 기능들이 덮고 있는 다양한 옵션들에 관해 더 자세히 설명하려고 노력하였다.

어떤 기능을 구현하는데 사용하는 클래스들에 더 깊게 들어가기 위해 별도의 소단원을 마련하기도 하였다. C++ 부가 설명 절과 유사하게, 이들 MFC 클래스 라이브러리에 대해 다룬 절들을 넘어간 다음, 나중에 Visual C++의 하부 구조를 좀더 깊게 파고 들어갈 준비가 된 다음에 돌아와서 봐도 된다. 이런 식으로, 처음 21일 동안 공부한 다음, 이 책의 나머지 부분을 살펴보면 된다.

이 책은 어떻게 구성되었을까?

이 책은 각 7일 동안의 내용이 한 주로 묶여져 있다. 하지만, 주 단위로 묶여져 있다 하더라도 주제가 이런 식으로 묶여져 있지는 않다.

각 일(day)의 마지막에는 간단한 퀴즈와 배운 주제에 관해 더 경험을 쌓을 수 있도록 한, 두 개의 연습문제를 준비했다. 하지만, 너무 걱정하지는 않아도 된다. 퀴즈에 대해 답이 필요하거나 길잡이가 필요할 경우에는 부록 A, “퀴즈 및 연습문제 해답”을 참고하면 된다.

첫째 주에는 Visual C++로 응용 프로그램을 만드는데 필요한 기본을 소개하고 있다. 응용 프로그램 윈도우를 디자인하는데 사용하는 디자이너의 사용법과, Windows 응용 프로그램 개발자에게 제공되는 다양한 컨트롤들을 사용하는 방법을 배울 것이다. 또한, Visual C++ 개발 환경과 여러 도구들에 대해서도 살펴볼 것이다.

둘째 주를 시작하면서, 점점 더 프로그래밍의 깊은 주제를 살펴볼 것이다. 여전히 Visual C++ 도구들을 사용해서 응용 프로그램을 만들겠지만, 프로그래밍 코드가 조금씩 붙어날 것이다. 또한, 그래픽 출력과 SDI/MDI 응용 프로그램 작성 등 심도 있는 주제를 살펴보기 시작할 것이다. 둘째 주의 마지막에는 데이터베이스도 다뤄볼 것이다.

셋째 주에는 ActiveX 컨트롤을 작성하고 사용하는 방법을 배울 것이다. 또한, 동시에 여러 개의 작업을 수행하는 멀티태스킹 응용 프로그램을 만드는 방법도 배울 것이다. 마지막으로, Microsoft의 새 .NET 플랫폼에서 Visual C++을 사용하는 기본적인 방법들을 배우고, 응용 프로그램을 디자인하고 프로그래밍하는 방법을 배울 것이다.

셋째 주를 마친 후에는, Visual C++로 Windows 프로그래밍할 준비가 된 것이다. 대부분 Windows 응용 프로그램을 만드는데 필요한 기술과 노하우를 가지게 된 것이다.

이 책에서 사용한 규칙들

이 책을 읽으면서, 다루고 있는 주제를 좀더 쉽게 이해하도록 도와주는 여러 가지 규칙을 볼 수 있을 것이다.

이 책의 모든 소스 코드는 일정 크기의 폰트로 보여진다(예로, [리스트 0.1]을 보자). 만들 응용 프로그램의 모든 소스 코드와 다양한 함수들의 사용 방법을 설명하는데도 적용된다. 적당한 값으로 대체해야 할 구문 변수는 일정 크기의 이탤릭체로 보여진다. 이미 다른 코드가 있는 함수에 새 코드를 추가하거나 변경할 때에는 굵은 글씨의 텍스트로 만들고 특별한 입력 아이콘으로 위치를 표시하였다.

입력 리스트 0.1 예제 코드

```
void main()
{
    // 존재하는 코드에 약간의 코드를 추가하거나 변경할 때에는
    // 그 부분을 굵은 글씨로 표시하여 알아보기 쉽게 할 것이다.
}
```

다음과 같은 표시를 사용해서 특정 정보를 표시하였다.

분석 이 아이콘은 조금 전의 코드를 설명하는 텍스트임을 표시한다.



새로운 용어

이 아이콘은 새 용어를 정의하고 설명하였음을 표시한다. 정의된 용어는 *이탤릭* 형식으로 보여진다.

 **Note**

설명하고 있는 주제나 중요한 포인트에 대해 좀더 설명할 때 사용한다. 또한, 문제를 피하는데 도움을 주거나 설명한 것을 사용할 때 고려해야 하는 정보들을 제공하기도 한다.

 **Tip**

어떤 작업을 대체할 수 있는 좀더 쉬운 방법을 제안하거나 팁 등을 조언할 때 사용한다.

 **주의**

잠재적인 문제에 대해 알려주어, 나중에 후회하지 않도록 경고할 때 사용한다.

이 책이 어떻고, 어떻고..를 읽으려고 이 책을 구입하지는 않았을 것이다. Windows 응용 프로그램을 만들기 위해 Visual C++를 배우려고 이 책을 샀을 터이니, 어서 페이지를 넘겨 프로그래밍을 시작해 보도록 하자.



Week 1

시작하면서

Visual C++의 세계에 온 여러분을 환영한다. 이제 3주, 즉 21장에 걸쳐서 Visual C++라는 매우 유연하고 완벽한 프로그래밍 도구를 사용해서 다양한 응용 프로그램을 만드는 방법을 공부하게 될 것이다. 각 장마다 서로 다른 영역에 대해 공부한 주제를 여러분의 응용 프로그램에 녹여 넣는 방법을 배우며, 장이 끝날 무렵에는 여러분이 직접 만든 샘플 응용 프로그램을 하나씩 가지게 될 것이다. 물론, 스스로 하는 것보다 새로운 기술을 배우는 가장 효과적인 방법은 없다는 사실은 자명하다. 이 책을 접을 때까지 직접 같이 작업하면서 독파해 가기 바란다.

1일째에서의 과정은 Visual C++를 통한 응용 프로그램 제작의 기초에 초점을 두었다. 간단한 응용 프로그램을 만들어 가면서 Visual C++의 개발 환경에 익숙해질 수 있도록 도와주는 것으로 시작된다.

2일째는 Visual C++로 만든 응용 프로그램을 디버깅하는 도구와 기술을 배울 것이다. 사용자에게 응용 프로그램을 주기 전에 응용 프로그램에 잠재된 버그들을 드러내는데 도움을 주는 특별한 매크로를 추가하는 방법을 배울 것이다. 또한, Visual Studio .NET 통합 개발 환경(Integrated Development Environment, IDE)의 통합 디버거를 사용하는 방법과 코드 상의 잘못된 문제를 찾는 방법을 배울 것이다.



Note

왜 어떤 것도 배우지 않은 상태에서 문제를 디버깅하는 방법을 배워야 할까? 디버깅하고 코드를 단계별로 살펴보는 것은 여러분의 코드가 어떻게 동작하는지 더 잘 이해할 수 있게 해준다. 또한, 응용 프로그램에 한, 두 줄의 코드를 바꾸었을 때, 버그가 살며시 숨어 들어가게 되면 이 버그들이 어디에 있는지 정확하게 찾아내는 방법을 익혀야 한다.

3일째는 Visual C++를 통한 응용 프로그램 제작의 조금 세심한 부분으로 조금 더 들어가게 된다. Windows 응용 프로그램에서 사용되는 표준 컨트롤의 일반적인 사항과, 이 컨트롤들 응용 프로그램 윈도우에 놓고 속성을 설정하며, 조작하는 방법을 공부하게 될 것이다.

4일째에서는 마우스와 키보드 이벤트를 잡아내어 처리하는 방법을 공부한다. 마우스 커서가 응용 프로그램 공간 내의 어느 위치에 있는지와 사용자가 어떤 키를 눌렀는지를 탐지할 수 있는 노하우를 얻게 될 것이다.

5일째에서는 Visual C++ 응용 프로그램에서 타이머를 사용하는 방법을 배운다. 동시에 두 개 이상의 타이머를 사용하고 각각을 구분하는 방법이 여기서 다루어진다.

6일째에서는 여러분의 응용 프로그램에 윈도우를 하나 더 추가해서 사용자로부터의 입력 정보를 받아 오도록 해본다. 일단, 이미 준비되어 있어서 꾸미지 않고도 사용할 수 있는 내장 대화 상자를 사용하여 사용자에게서 간단한 정보를 받아내 본 다음, 좀더 세밀한 정보를 얻기 위해 여러분만의 커스텀 대화 상자를 만들어 보도록 하자.

7일째는 메뉴에 관해 다룬 장이다. 여러분이 응용 프로그램에 추가한 메뉴에서 적합한 함수를 호출하는 방법을 배우게 될 것이다.

이것으로 첫 번째 주의 공부는 끝을 맺는다. 첫째 주가 끝날 때쯤, 7일에 걸쳐서 배운 것을 꼭 확인해보고 응용 프로그램을 만들 때 어떤 것이 필요했었는지 한번 돌아켜볼 기회가 있을 것이다. 부산 그만 떨고 이제 공부를 시작하자.

7 일째

Visual C++ 개발 환경과 첫 예제 프로그램 만들기

“초보자를 위한 Visual C++ .NET 21일 완성”에 온 것을 환영한다. 3주간에 걸쳐, Microsoft의 Visual C++ .NET을 이용하여 폭넓고 다양한 응용 프로그램을 만들어 볼 것이다. 가장 중요한 것은, 여러분 스스로 이런 형식의 응용 프로그램을 만들어 봄으로써 익히는 것이다. 이 책을 읽어가면서 조금씩 늘어가는 자신의 실력을 느낄 수 있을 것이다. 자, 이제 시작하자!

Note

먼저, Visual C++ 프로그래밍 언어는 Visual Studio IDE(Integrated Development Environment, 통합 개발 환경)와 다르다는 것을 이해해야 한다. 도구와 편집기에 대해 언급하는 것은 Visual Studio IDE에 관한 것이다. Visual Studio 전체 패키지가 아니라, Visual C++만을 구입하였다더라도 Visual Studio IDE에서 작업할 수 있다. Visual Studio IDE는 Visual Studio에 포함된 모든 프로그래밍 언어에서 사용하는 개발 환경과 도구 모음이다. 간단히 말해, 앞으로 21일에 걸쳐 Visual Studio 도구와 편집기를 이용해 작업하게 된다. 그러나, Visual Studio 도구들을 이용해 키보드를 두들겨 작성한 프로그래밍 코드는 Visual C++ 프로그래밍 언어이다.

이 장에서는 Visual C++의 개발 환경과 응용 프로그램을 만들기 위해 같이 제공된 여러 가지 도구에 대해 먼저 알아보도록 하겠다. Visual C++가 제공하는 개발 도구들은 사실 1장 분량으로 공부하기에는 턱없이 부족할 정도로 엄청나고 방대하지만, Visual C++를 가지고 응용 프로그램을 만들 때 우선적으로 가장 많이 필요한 것을 중심으로 살펴보도록 하자. 하여튼, 이 장에서는 다음 것들을 가뿐하게 공부하고 지나가자.

- Visual C++ 개발 환경이 가진 주요한 것들
- 응용 프로그램 마법사를 이용하여 응용 프로그램의 기본 골격을 구축하는 과정
- Visual Basic이나 파워빌더(PowerBuilder), 델파이(Delphi)에서 봐왔던 대화 상자 윈도우의 시각적 편집을 가능하게 해주는 대화 상자 페인터를 사용하는 방법
- 속성 페인(Pane)을 이용하여 응용 프로그램 윈도우에 함수를 추가하는 방법

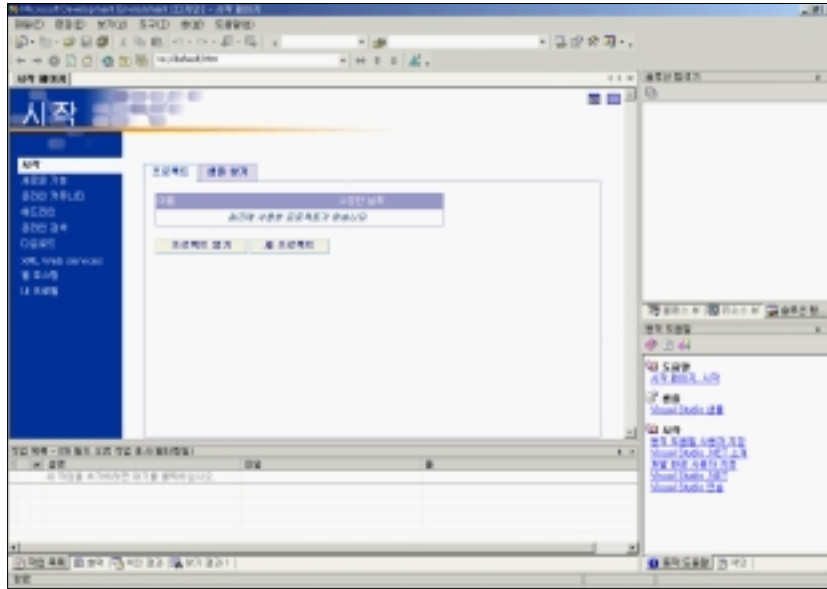
Visual C++ 개발 환경 살펴보기

Visual C++의 개발 환경을 살펴보기 전에, 일단 여러분의 시스템에 있는 Visual C++를 실행시켜서 각 부분이 어떻게 되어 있고, 어떻게 여러분 취향대로 다시 배열시킬 수 있을지를 알아보는 것이 급선무이다.

Visual Studio(Microsoft 비주얼 개발 환경을 말한다)를 시작하면, [그림 1.1]과 같은 윈도우를 보게 될 것이다. 이 윈도우의 각 부분이 디벨로퍼 스튜디오를 구성하는 나름대로의 기능을 가지고 있으며, 마음에 들지 않으면 위치를 변경하여 모습을 바꿀 수도 있다.

그림 1.1

Visual C++가 열렸을 때의 화면



Note

역자주 이 책의 원서는 베타 버전을 기준으로 만들어졌으나, 역자가 번역할 당시 Visual Studio .NET 정식 버전이 발표되었다. 이 책에 사용된 그림은 Visual Studio .NET 정식 버전으로 작업하였으므로 일부 그림과 내용이 원서와 다를 수 있으나 한국어를 제 1 모국어로 삼고 있는 자랑스런 우리 배달의 독자들은 전혀 신경쓸 필요가 없겠다.

주의

시작 페이지를 열거나, Visual Studio 환경의 어떤 함수는 여러분의 컴퓨터에서 스크립트를 만들고 실행한다. Visual Studio .NET 또는 Visual C++ .NET이 설치된 컴퓨터에 백신 프로그램이 있다면 이러한 스크립트를 실행할 때 잠재적으로 바이러스가 있다는 경고가 나타날 수 있다.

솔루션 탐색기

Visual C++을 처음으로 실행하면 IDE의 오른쪽에 꽤 넓은 자리를 꿰어차고 앉은 녀석이 하나 있다. 솔루션 탐색기라고 부르는 이 영역은 다양한 조각과 개발 프로젝트를 조정하는 중심적인 역할을

한다. 솔루션 탐색기는 이전 버전의 Visual Basic, Visual InterDev, 그리고 Visual J++에 있었던 프로젝트 탐색기와 유사하다. 이 그룹에 있는 패인(Pane)은 3가지 다른 방법으로 응용 프로그램의 일부를 보여준다.

- **클래스 패인**은 소스 코드를 C++ 클래스 수준에서 살펴보고 조작할 수 있게 한다.
- **리소스 패인**은 응용 프로그램에서 사용하는 여러 가지 리소스(대화 상자, 아이콘, 메뉴 등)를 찾아서 편집할 수 있도록 한다.
- **솔루션 탐색기라고 표시된 패인**은 응용 프로그램에서 만드는데 필요한 모든 파일을 찾아볼 수 있게 한다. 하나의 솔루션으로 여러 가지의 프로젝트가 있을 때 각각을 볼 수 있게 한다. 솔루션 탐색기를 통해 프로젝트 사이를 탐색할 수 있다.



Note

처음에 이 영역에 있던 패인들의 모음은 단지 처음 배치를 말한 것이다. 초기 3개의 뷰에 있는 어떤 것이든 지 드래그하여 IDE의 다른 부분으로 옮길 수 있고, 이 그룹으로 더할 수도 있다.

출력 영역

출력 영역은 Visual C++를 처음 시작했을 때에는 나타나지 않은 상태이며, 응용 프로그램을 컴파일하고 나면 디벨로퍼 스튜디오의 아래에 생기고 여러분이 닫을 때까지 없어지지 않는다. 출력 영역은 디벨로퍼 스튜디오가 여러분에게 주었으면 하는 정보를 표시하는 역할을 한다. 즉, 여기서 컴파일러의 진행 과정과 경고 메시지, 또는 에러 메시지 등을 볼 수 있으며, Visual C++ 디버거를 실행시켰을 경우 각각의 값을 가진 실행중의 모든 변수도 볼 수 있다. 출력 영역을 닫은 후에도 Visual C++가 여러분에게 무언가를 알려주어야 할 것이 있으면 다시 열린다.

편집기 영역

디벨로퍼 스튜디오 환경의 오른쪽은 편집기 영역이 자리잡고 있는데, Visual C++를 사용하면서 필요한 모든 편집을 모두 여기서 할 수 있다. 즉, C++ 소스 코드를 편집할 수 있는 코드 편집기와 대화 상자를 그릴 수 있는 대화 상자 편집기, 심지어 아이콘까지 그릴 수 있는 편집기도 이 영역에서 사용한다. 디벨로퍼 스튜디오를 기본적으로 덮고 있는 것이 편집기 영역이지만, 메뉴, 도구 모음 등과 자리를 같이 쓰기도 한다.

메뉴 바

Visual C++를 처음 실행시켰을 때 메뉴 바 아래에 여러 가지의 도구 모음이 나타난다. Visual C++에는 다른 많은 도구 모음이 있다. 이것을 여러분의 환경에 맞게 편집하거나 새로 도구 모음을 만들 수 있다. 아무튼, 초기에 열린 도구 모음은 다음과 같다.

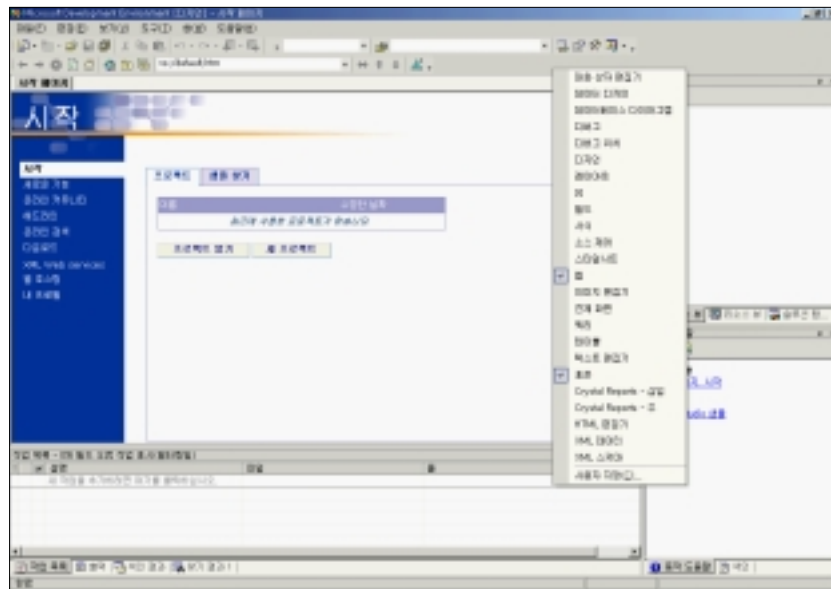
- 가장 빈번하게 사용하는 명령, 즉 파일 열기와 저장하기, 잘라붙이기와 복사하기 등을 가지고 있는 표준(Standard) 도구 모음
- 편집기 영역에서 수행하는 활동에 어울리는 기능을 담은 도구 모음. 예를 들어, 처음에 Visual C++를 열면 Visual C++ 문서를 살펴볼 수 있는 도구 모음을 볼 수 있을 것이다. C++ 코드의 편집을 시작하면 코드를 편집하는 동안 자주 사용하는 기능을 담은 도구 모음이 나타난다. 대화 상자 윈도우의 디자인을 시작하면 대화 상자에 컨트롤을 배치하고 크기를 조절하는 기능을 담은 도구 모음이 나타난다.

디벨로퍼 스튜디오 환경을 조정하기

디벨로퍼 스튜디오가 가지고 있는 도구 모음을 배치하는 방법은 두 가지이다. 첫 번째는 도구 모음 영역에서 마우스 오른쪽 클릭을 하는 것으로, [그림 1.2]와 같은 팝업 문맥 메뉴가 나타나면서 여러 가지 도구 모음을 표시하거나 숨길 수 있다.

그림 1.2

메뉴에서 도구 모음을 표시하거나 숨길 수 있다.



두 번째는 도구 모음을 재배치할 때 사용하는 것으로, 마우스로 도구 모음과 패인의 왼쪽 끝에 있는 바를 잡고 움직이면 된다. 이 상태에서 도킹된 위치로부터 떼어내면 떠 있는 도구 모음으로 만들 수 있으며, 디벨로퍼 스튜디오의 다른 테두리에 도킹할 수도 있다. 도구 모음이 도킹된 상태에서도 원하는 곳으로 드래그해서 위치를 조정하면 된다.

패인을 움직이는 것은 약간 복잡해졌다. 새로운 Visual Studio IDE에서는 어느 패인에 있는 탭이든 지 분리하여 독립된 패인으로 만들 수 있다. 또는 패인끼리 새로운 조합으로 만들 수 있다. 두 개 또는 그 이상의 패인을 하나의 그룹으로 만들면 어느 패인을 보여줄지 정하기 위해 여러 개의 탭들을 가지게 된다.



Note

속성, 출력 영역의 위쪽에 있는 바를 이용하여 전체를 디벨로퍼 스튜디오 환경 주위로 드래그할 수 있다. 이렇게 하면, 이 윈도우들은 디벨로퍼 스튜디오에 도킹된다.

이제 프로젝트를 처음으로 만들어 보자

여러분이 Visual C++로 만들 첫 번째 응용 프로그램은 [그림 1.4]에서 보듯이, 두 개의 버튼이 붙어있는 썬렁한 응용 프로그램이다. 첫 번째 버튼은 사용자에게 [그림 1.5]와 같은 환영 메시지를 띄워주는 역할을 하며, 두 번째 버튼은 응용 프로그램을 닫는다. 멍하니 있지 말고, 다음의 단계를 차근차근 따라가야 한다.

1. 새 프로젝트의 워크스페이스를 만든다.
2. 응용 프로그램 마법사를 사용해서 응용 프로그램의 골격을 만든다.
3. 응용 프로그램 마법사가 자동으로 만들어 준 대화 상자를 재조정해서 원하는 모양으로 만든다.
4. 사용자에게 환영 메시지를 띄우는 C++ 코드를 추가한다.
5. 응용 프로그램에 새 아이콘을 만들어 붙인다.

그림 1.3

여러분의 첫 번째
Visual C++ 응용 프로그램

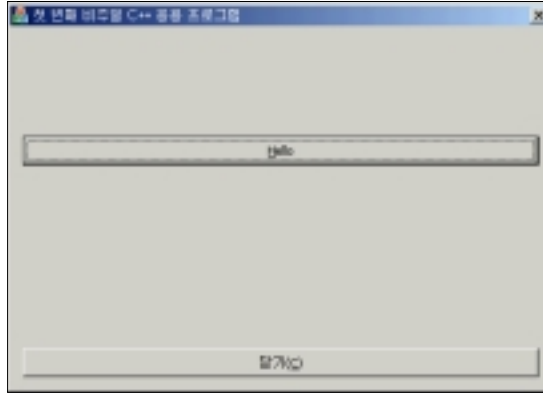
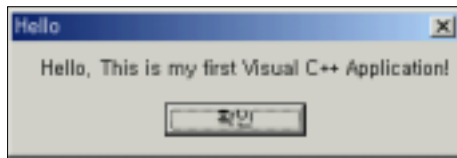


그림 1.4

사용자가 첫 번째 버튼을
클릭하면 간단한 환영
메시지가 나타난다.



프로젝트 워크스페이스 만들기

응용 프로그램 개발 프로젝트라면, 모두 Visual C++에서 각자의 프로젝트 워크스페이스를 가져야 한다. 이 워크스페이스는 응용 프로그램을 구성하는 소스 코드가 어느 디렉토리에 위치해 있고, 여러 가지 설정 정보를 담은 파일이 어느 디렉토리에 위치해 있는지를 디벨로퍼 스튜디오에게 알려주는 역할을 한다. 하여간 다음의 단계를 따라서 새 프로젝트 워크스페이스를 만들기로 하자.

1. VS .NET의 시작 페이지에 있는 새 프로젝트를 클릭한다(페이지의 왼편에 시작 탭을 선택해야 버튼이 보인다). (그림 1.5)와 같은 새 프로젝트 마법사가 나타난다.



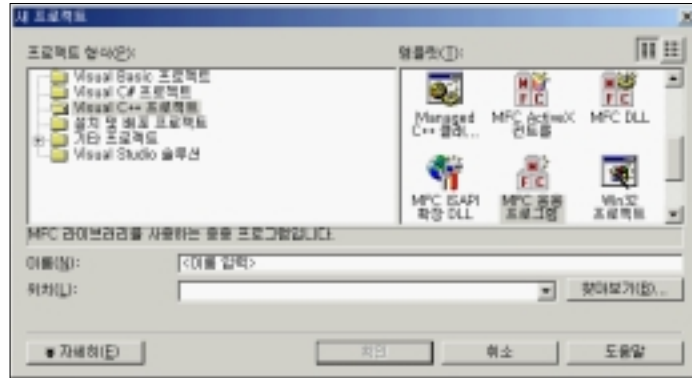
Tip

시작 페이지를 달았다면, VS .NET의 메뉴에서 보기 | 웹 브라우저 | 브라우저 표시를 차례대로 선택한다.

2. 새 프로젝트 마법사에서 프로젝트 형식 트리의 [Visual C++ 프로젝트]를 선택한다. 그리고, 대화 상자 오른편의 템플릿 영역에서 [MFC 응용 프로그램]을 선택한다.

그림 1.5

새 프로젝트 마법사



3. 여러분의 프로젝트 이름을 이름 필드에 써넣는다. 멋진 이름이 생각나지 않으면 **Hello**가 적당할 것이다.
4. [확인]을 클릭하면 두 가지 일이 발생되는데, 즉 프로젝트의 디렉토리가 만들어지고(Location 필드에 설정된 대로 만들어지지만, 응용 프로그램 마법사가 끝나기 전에는 만들어지지 않는다), 응용 프로그램 마법사가 시작된다.

응용 프로그램 마법사를 사용해서 응용 프로그램의 골격 만들기

드디어 나타난 응용 프로그램 마법사는 여러분에게 몇 단계에 걸쳐서 어떤 형식의 응용 프로그램에 어떤 기능을 넣고 싶은지를 꼬치꼬치 물어본다. 여러분이 입력한 정보를 바탕으로 응용 프로그램의 골격을 만들어 내며, 바로 컴파일해서 실행해 볼 수 있다. 이제 여러분은 이 골격을 바탕으로 원하는 응용 프로그램을 만들면 되는 것이다. 다음의 단계를 따르자.

1. 마법사의 왼쪽에 응용 프로그램 종류를 선택한다. 여기에서 여러분이 만들기 원하는 응용 프로그램의 몇 가지 옵션을 제공한다. 오른쪽에서 “대화 상자 기반” 옵션을 선택한다.
2. 왼쪽에서 “사용자 인터페이스 기능”을 선택한다. 여기에서는 메인 응용 프로그램의 모양을 선택할 수 있다. 아래 필드에 적혀 있는 프로젝트 이름(Hello)을 지우고 다른 이름으로 쓸 수 있는데, 이 이름이 주 응용 프로그램 윈도우에 나타나게 된다. 우리는 “첫 번째 비주얼 C++ 응용 프로그램”이라고 쓰자.

그림 1.6

응용 프로그램 형식을 정한다.

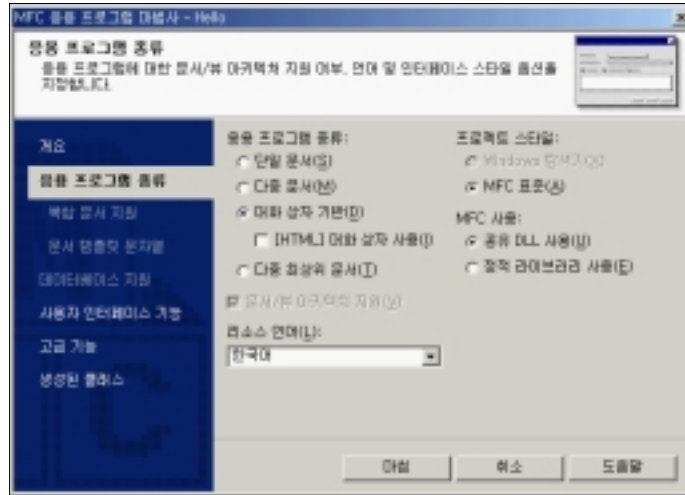
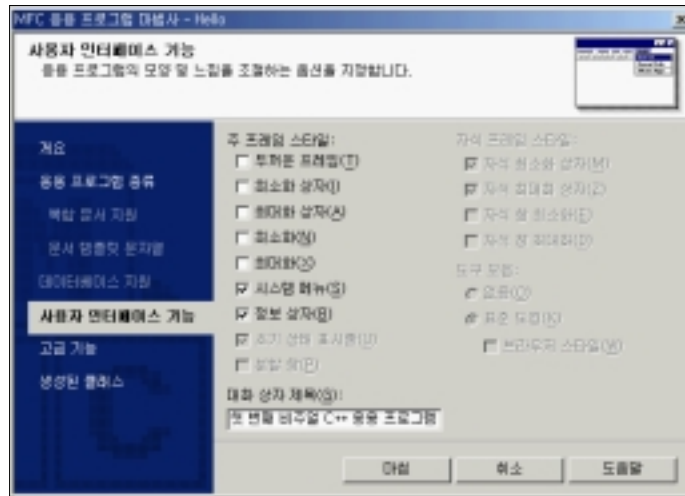


그림 1.7

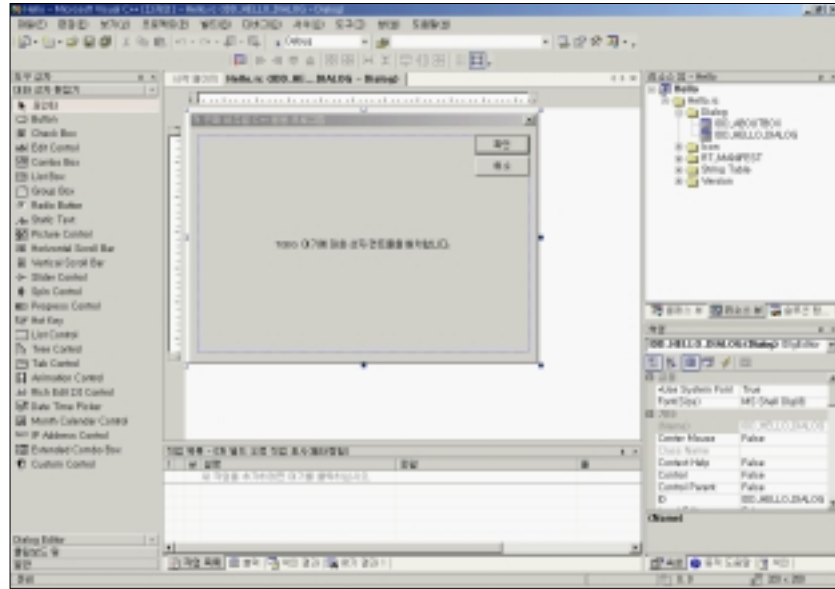
응용 프로그램의 제목을 정한다.



3. [마침]을 클릭하면 MFC 응용 프로그램 마법사가 응용 프로그램 골격을 만든다.
4. 응용 프로그램 마법사가 응용 프로그램 골격을 모두 만들어낸 이후에는 디벨로퍼 스튜디오 환경으로 되돌아 온 여러분을 느낄 수 있을 것이다. 또한, [그림 1.8]과 같이 프로젝트 워크스페이스 구역에 리소스 구조가 트리 형식으로 나와 있는 것도 새로 볼 수 있을 것이다. 편집기 영역에 메인 대화 상자가 나와 있을 수도 있다.
5. 빌드 | Hello 빌드를 선택해서 응용 프로그램을 컴파일하자.

그림 1.8

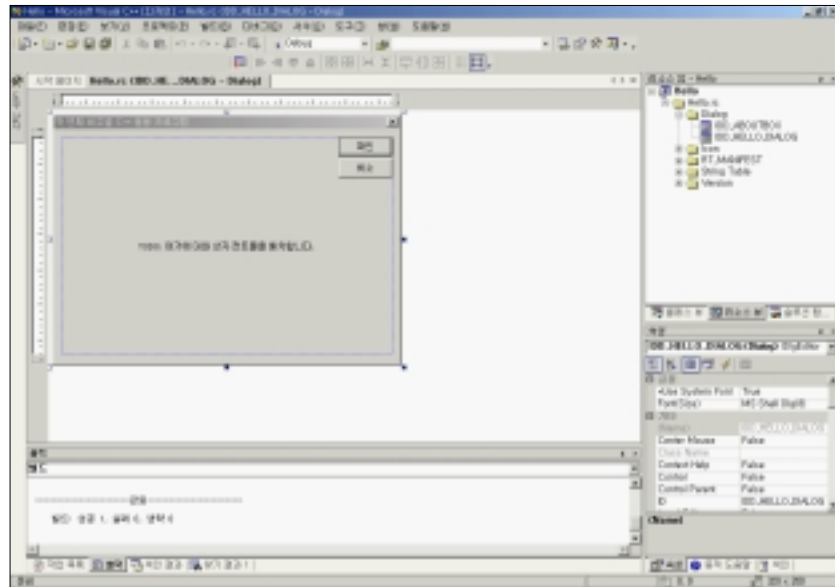
프로젝트의 클래스 정보를 트리뷰 형태로 가지고 있는 워크스페이스



6. Visual C++ 컴파일러가 여러분의 응용 프로그램을 컴파일해 갈 때 출력 영역을 잘 보기 바란다. 컴파일 진행 상황과 메시지가 출력될 것이다. 응용 프로그램이 모두 만들어지면, 출력 영역에 “에러나 경고가 없습니다”라는 메시지가 나타난다((그림 1.9) 참조).

그림 1.9

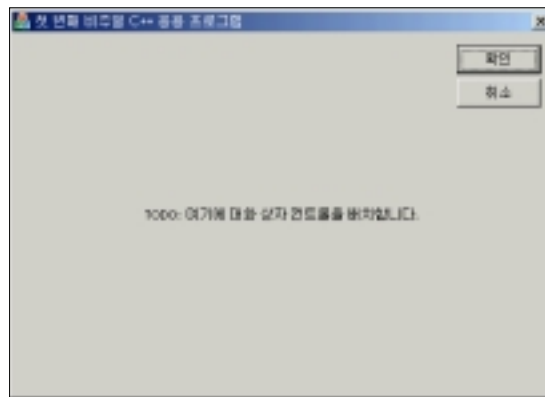
출력 윈도우에는 어떠한 컴파일러 에러라도 나타난다.



7. 디버그 | 시작을 선택해서 응용 프로그램을 실행시키자.
8. 강렬한 하드디스크의 움직임과 함께 TODO 메시지와 [확인], [취소] 버튼이 달린 대화 상자가 나타난다(그림 1.10 참조). 응용 프로그램 자체 이름 대신에, 이전에 응용 프로그램 마법사에서 입력한 제목이 타이틀 바에 보일 것이다. 두 버튼 중 하나만 클릭하면 응용 프로그램이 끝날 것이다.

그림 1.10

수정되지 않은 응용
프로그램 골격



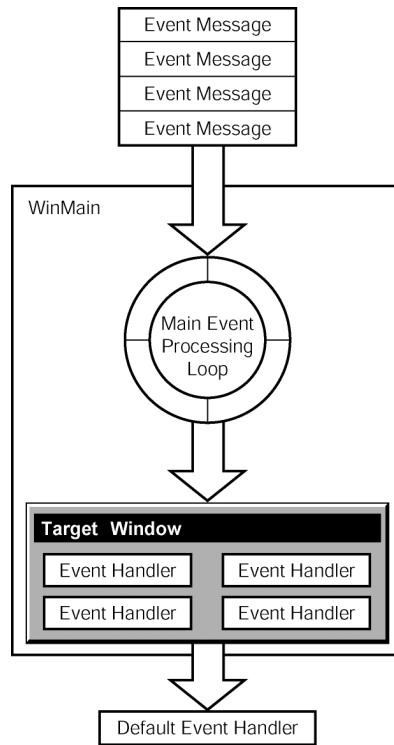
MFC 노트 : Windows 응용 프로그램은 어떻게 동작하는가?

이 섹션 뿐 만 아니라 앞으로도 계속하여 Microsoft Foundation Classes(MFC)에 대하여 살펴볼 것이다. 또한, 어떻게 하여 Windows 응용 프로그램을 빠르게 작성할 수 있는지도 배울 것이다. 이 정보가 이 책에서 필요한 것을 만드는데 꼭 필요하지는 않다. 이 섹션을 읽지 않아도, MFC에 대하여 많이 배우게 되고, 어떻게 MFC를 이용하여 응용 프로그램을 만드는지 알게 된다. 그러나, 표면적인 것 외에 어떻게 동작하는지 이해하고 싶다면 이곳을 꼭 읽어봐야 한다.

Microsoft Foundation Classes는 Windows 운영 시스템과 상호 작용할 수 있도록 많은 함수들을 추상적으로 만든 C++ 클래스의 모음이다. Visual Basic이나 Delphi같은 Windows 응용 프로그램을 만드는 다른 비주얼 프로그래밍 도구를 사용해 본 경험이 있다면, 함수들을 모아서 제공하는 컴포넌트(components)를 가지고 작업을 했을 것이다. Visual Basic의 명령어 버튼(Command Button)은 모양을 조절하고 행동을 정의할 수 있는 다양한 속성을 가지고 있다. MFC에서는 명령어 버튼의 기능을 담고 있는 CButton과 같은 클래스를 가지고 있다.

MFC와 같은 응용 프로그램 프레임워크를 사용하지 않고 C/C++로 Windows 프로그램을 어떻게 만드는지 공부한 적이 있다면, 모든 Windows 프로그램이 WinMain이라는 함수로부터 시작한다는 것은 알고 있을 것이다. 이 함수가 응용 프로그램의 주 윈도우를 만드는 응용 프로그램의 시작점이다. 컨트롤이나 윈도우가 올바르게 진행하도록 메시지를 적당한 이벤트 핸들러 함수에게 전달하기 위해, 계속해서 이벤트 메시지를 받는 루프를 시작한다(그림 1.11 참조). 메시지를 처리할 적당한 이벤트 핸들러가 없다면, 운영 시스템에서 처리하도록 디폴트 이벤트 핸들러(Default Event Handler)에게 전달하면 된다.

그림 1.11
Windows 응용 프로그램의
메시지 처리 과정



MFC는 Windows 응용 프로그램을 만드는데 필요한 이러한 과정의 대부분을 없앴다. 깊이 파고든다면 MFC 소스 코드의 깊은 곳에 있는 WinMain 함수를 찾을 수 있을 것이다. 전에 본 것처럼 워크스페이스 패인의 Class View를 보면, CAboutDlg, CHelloApp, 그리고 CHelloDlg 3개의 클래스가 있다. 이 클래스 중에 두 번째인 CHelloApp는 CWinApp 클래스로부터 파생되었다. 이 클래스는 WinMain의 기능의 많은 부분을 감추고, Windows 응용 프로그램을 만드는데 짐이 되었던 지겨운 일들은 대신해 준다. 또한, CWinApp 클래스는 레지스트리에서 설정 정보를 읽고 쓰거나 명령어

라인의 매개 변수를 처리해 주는 기능들처럼, 어떤 컨트롤이나 개체에는 없는 일반 응용 프로그램의 기능을 가지고 있다.

다른 두 개의 클래스 CAboutDlg와 CHelloDlg는 모두 CDialog 클래스로부터 파생되었다. 이 CDialog 클래스 역시 더 일반적인 목적으로 만들어진 CWnd 클래스로부터 특별히 파생되었다. 이 CWnd 클래스는 윈도우 기능의 많은 부분을 보호하고 있고, CDialog 클래스는 대화 상자 윈도우에 중점을 맞추고 있다. 여러분은 다음 며칠 동안 CDialog 클래스와 그 상위 클래스인 CWnd 클래스에 대해 더 많이 배우게 될 것이다.



새로운 용어

어떤 클래스가 다른 클래스로부터 *파생(derivation)*되었다 함은, 첫 번째 클래스가 두 번째 클래스로부터 상속되었음을 의미한다. 즉, 두 번째 클래스가 첫 번째의 부모가 되는 것이다. 파생된 클래스(상속받은 클래스)는 부모 클래스를 자신의 일부로 가진다. 파생된 클래스에 어떤 기능을 추가하지 않고도 이미 부모 클래스의 모든 기능을 가지고 있는 것이다. 모든 존재하는 기능을 부모 클래스로부터 상속받기 때문에 파생된 클래스는 부모 클래스와 다른 점에 초점을 맞추면 상당량의 코드를 줄일 수 있다.

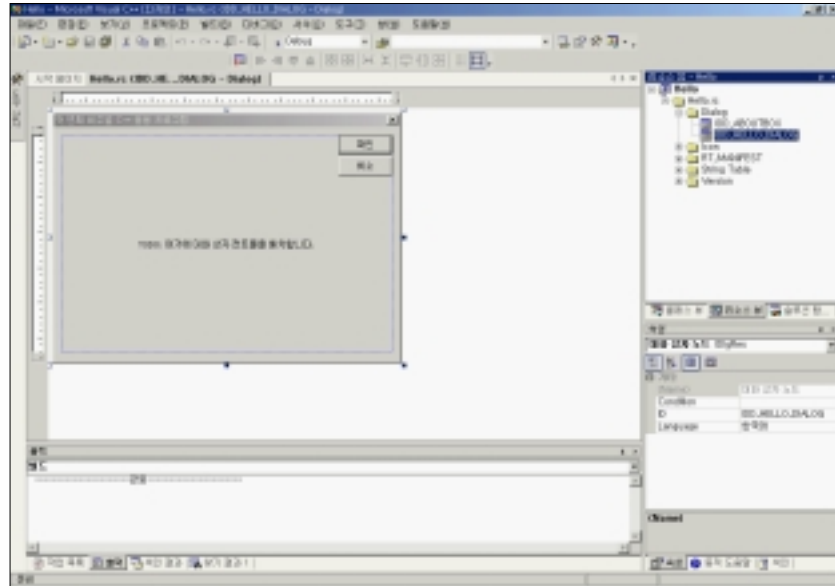
응용 프로그램 윈도우를 직접 꾸미자

현재 만들어진 응용 프로그램 골격이 실행이 안될 염려는 없기 때문에, 이제 할 일은 윈도우 꾸미기에 전력을 다하는 것이다. 메인 대화 상자 윈도우가 편집기 영역에 이미 나타나 있는 경우도 있겠지만, 이후에도 이럴 것이란 법이 절대 없으므로 순차적으로 대화 상자를 찾아가도록 하자. 그리고 나서 대화 상자의 레이아웃을 고치도록 하자.

1. [그림 1.12]와 같이 프로젝트 워크스페이스에서 리소스 뷰 탭을 선택한다.
2. 트리 구조를 확장해서 Dialog 부분이 나타나게 하자. 이 부분을 한번 더 펼치면 IDD_HELLO_DIALOG가 나타나며, 이 항목을 더블 클릭하면 디벨로퍼 스튜디오의 편집기 영역에 대화 상자가 나타난다.
3. 대화 상자에 표시되는 텍스트를 선택하고 Delete 키를 눌러서 지우자.

그림 1.12

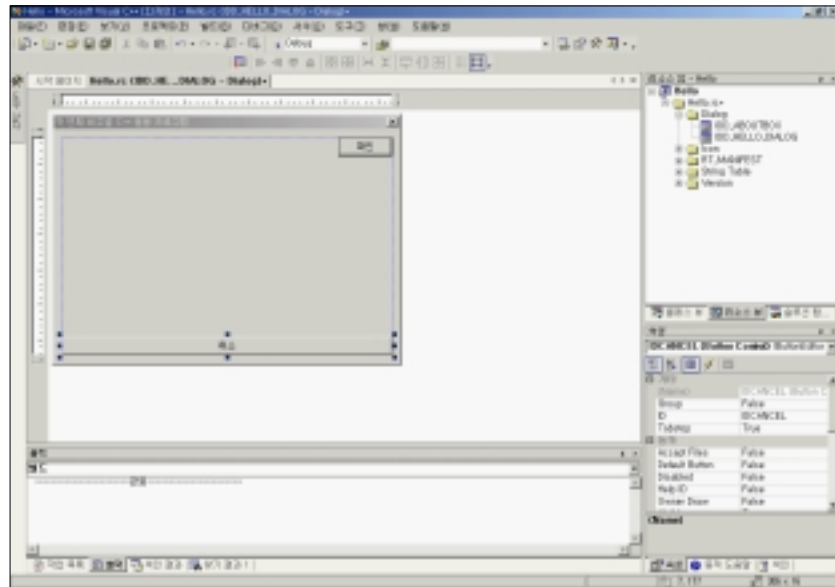
워크스페이스의 리소스 뷰 탭



4. [취소] 버튼을 클릭하고 대화 상자의 아래까지 잡아 끈 다음, 윈도우의 폭을 완전히 차지하도록 크기를 조절한다(그림 1.13) 참조.

그림 1.13

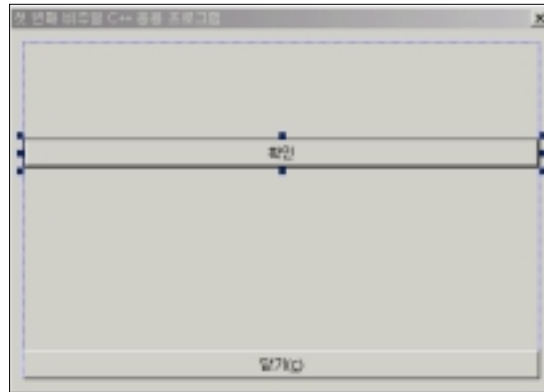
취소 버튼을 위치시킨다.



5. 속성 패인(솔루션 탐색기 패인의 오른쪽 아래)에서 Caption 필드를 "닫기(&C)"로 바꾸자.
6. [확인] 버튼의 위치와 크기를 [그림 1.14]와 비슷하게 만들자.

그림 1.14

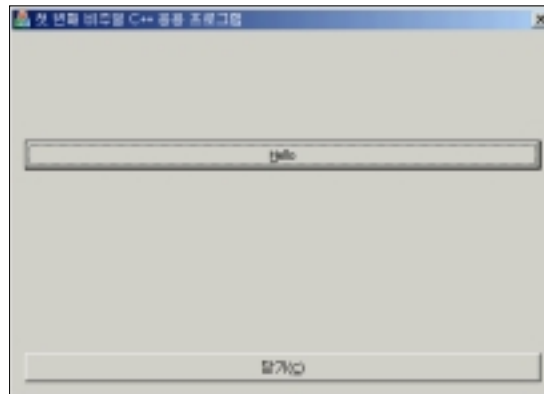
확인 버튼을 위치시킨다.



7. 속성 영역에서 [확인] 버튼의 ID 값을 IDHELLO로 만들고, Caption 필드를 &Hello로 하자.
8. 이제 응용 프로그램을 컴파일하고 실행시키면, [그림 1.15]와 같이 여러분이 바꾼 그대로의 대화 상자가 나타나게 된다.

그림 1.15

다시 꾸민 응용 프로그램을 실행시킨 모습



Note

Windows 응용 프로그램이 처음인 분들을 위해 설명하면, Caption의 앰퍼샌드(& 기호)는 다음에 오는 문자가 버튼, 메뉴 또는 컨트롤의 단축 키라는 것을 의미한다. 단기(&C)에서 C 아래에 밑줄이 있는 것을 볼 수 있을 것이다(어떤 환경에서는 ALT 키를 눌러야 C 밑에 밑줄이 생기기도 한다). 사용자가 이 버튼을 누르기 위해서는 Alt 키와 앰퍼샌드(&) 다음에 오는 문자를 함께 눌러 단축 키를 사용하도록 해준다. 이것은 마우스를 대신하여 버튼(또는 메뉴, 기타 컨트롤)을 누르는 효과를 준다.

Note

방금 실행된 응용 프로그램을 가지고 놀다보면, 단기 버튼을 누르면 여전히 응용 프로그램을 끝내지만 Hello 버튼은 아무리 눌러도 그뿐이다. 이 버튼의 ID를 바꾸었기 때문이다. MFC 응용 프로그램은 각 컨트롤의 ID와 이벤트 메시지에 따라 호출할 함수를 결정하는 매크로를 소스 코드 안에 가지고 있는데, 여러분이 Hello 버튼의 ID를 바꾸었기 때문에 이 매크로가 더 이상 어떤 함수를 부를지 알지 못하는 것이다.

응용 프로그램에 코드 추가하기

대화 상자애다가 코드를 추가하는 데에는 여러 가지 방법이 있을 수 있는데, 가장 많이 쓰이는 방법이 속성 패널을 사용하는 것이다. 속성 패널은 응용 프로그램이 받아 처리할 Windows 메시지와 처리를 위해 사용되는 함수의 테이블을 만들 때 사용하며, 이 때 MFC에서 지정된 매크로가 쓰여 윈도우 컨트롤과 함수를 연결한다. 말은 좀 복잡하지만 사용자 입장에서는 간단하게 할 수 있다. 다음의 단계를 통해서 말이다.

1. Hello 버튼을 눌렀을 때 실행될 동작을 연결해 주려면, 일단 이 버튼을 클릭한 다음 속성 패널에서 컨트롤 이벤트(번개 표시)를 선택한다([그림 1.16] 참조).
2. 메시지 중에서 BN_CLICKED를 선택한다. 이벤트 ID 옆에 영역은 콤보 박스가 될 것이다. 콤보 박스의 화살표를 클릭하면 드롭다운 윈도우에 제안하는 함수의 이름이 표시된다. 제안된 함수의 이름은 OnBnClickedHello일 것이다([그림 1.17] 참조). 드롭다운 윈도우에서 이 이름을 선택하면 OnBnClickedHello 함수가 편집기 영역에 표시된다.

그림 1.16

속성 패널의 컨트롤 이벤트를 선택한다.

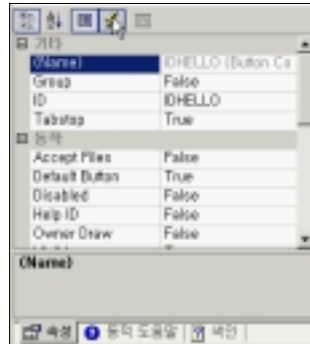
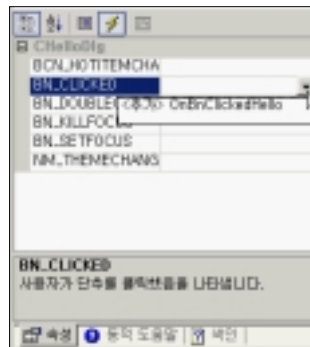


그림 1.17

제안된 이벤트 함수



3. TODO 주석문 뒤에다가 [리스트 1.1]에 나온 대로 코드를 입력하자. 잘 모르겠으면 [그림 1.18]을 참고하도록 하자.

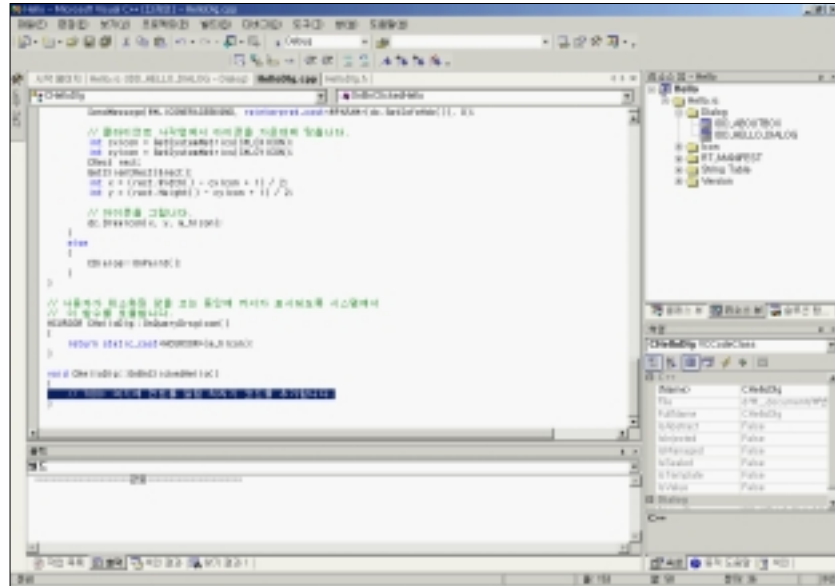


Note

코드를 입력할 때, 편집기는 함수 정의를 보여줄 것이다. 이 기능을 IntelliSense라고 하는데, 여러분이 어떤 함수 또는 개체의 방법/속성을 입력하려고 하는지 추측하고, 가능한 함수의 리스트 중에서 함수, 방법 또는 속성을 선택할 수 있도록 도와준다. 이 기능은 코드를 입력할 때, 에러를 발생하지 않도록 도와준다. 함수 이름은 입력하고 괄호를 열면 함수의 매개 변수를 보여주고 현재 입력할 매개 변수를 강조해서 표시한다. IntelliSense는 여러분이 어떤 함수나 속성이 가능한지 그리고 함수나 속성의 어떤 매개 변수가 가능한지 알려줘서 프로그래밍을 쉽게 하도록 도와준다. 이 기능은 Visual Basic 5에서 처음 소개되었고, 뒤에 Visual C++ 6에 추가되었다.

그림 1.18

[리스트 1.1]을
끼워 넣을 소스 코드 부분



[리스트 1.1] HELLODLG.CPP : OnBnClickedHello 함수

```

void CHelloDlg::OnBnClickedHello(void)
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    // 사용자에겐 hello라고 말한다.
    MessageBox("Hello. This is my first Visual C++ Application!");
}
    
```

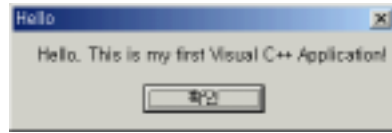
4. 이제 응용 프로그램을 컴파일하고 실행시킨 다음, [Hello] 버튼을 누르면 [그림 1.19]와 같은 메시지 상자가 나타나게 될 것이다.

Note

응용 프로그램을 만들 때 에러를 만나게 되면, [리스트 1.1]에 있는 코드를 정확히 입력했는지 살펴보자. C++는 대소문자를 구분하기 때문에 대문자가 포함된 함수 이름을 정확히 입력해야지 그렇지 않으면 에러가 난다.

그림 1.19

이제 여러분의 응용 프로그램이 환영 메시지를 출력한다.



C++ 부가 설명 : 함수 구조와 문법

이미 C++ 프로그래밍 언어를 알고 있고 Visual C++ 도구를 배우기 위해 이 책을 읽고 있다면 이 섹션은 넘어가도록 하자. 다음날부터 이 섹션은 C++ 프로그래밍 언어와 문법, 그리고 구조를 배우는데 집중할 것이다.

[리스트 1.1]에 여러분이 작성한 C++ 코드는 살펴볼 것이 여러 가지 있다. 첫 번째 행을 보자.

```
void CHelloDlg::OnBnClickedHello(void)
```

이 행은 Visual C++ 디벨로퍼 스튜디오가 만든 함수 구조의 일부이다. 함수 이름과 호출 문법이 다. 보통, 헤더 파일(파일 확장자를 .h로 표시한다)에 함수를 선언하는데 차후에 보게 될 것이다. 이 코드 리스트는 함수의 구현 부분이다.

행의 첫 번째 단어 void는 반환 값의 데이터형이다. 이곳이나 함수의 매개 변수(행의 끝에 괄호로 둘러싸인다)가 위치하는 곳에 void가 쓰이는데, 이것은 함수의 반환 값이 없거나, 함수로 전달하는 매개 변수가 없다는 것을 의미한다.

행의 두 번째 단어 CHelloDlg는 함수가 멤버로 속해있는 클래스를 표시한다. 다음에 두 개의 콜론 (::)이 오고 함수의 이름이 나온다. 이 요소들 간에 공백은 올 수 있지만 두 개의 콜론 사이에는 공백이 오면 안 된다.

함수 이름 다음에 함수로 전달하는 매개 변수의 리스트가 괄호에 쌓여 오게 된다. 함수가 어떤 매개 변수를 취하지 않더라도 괄호는 항상 사용해야 한다. 이 점이 이전의 Visual Basic과 같은 언어와의 차이점이다. Visual Basic에서 함수가 매개 변수를 취하던가 서브루틴이 반환 값을 갖지 않던가 하는 것은 옵션이었다.

C++ 규칙

C++에서 함수를 호출할 때, 함수에 전달하는 매개 변수가 없을 때에도 함수 이름 다음에 괄호를 항상 포함해야 한다.

리스트의 다음 행은, 함수 몸통의 시작을 알리기 위해 중괄호({})를 연다. 언젠가는 함수 몸통이 끝났음을 뜻하는 닫힌 중괄호(})가 오게 된다. 자바에서도 이와 같이 중괄호를 사용하고 유사하게 Visual Basic이나 Delphi에서는 BEGIN과 END를 사용한다.

C++ 규칙

함수 몸통은 항상 중괄호의 쌍({})으로 둘러싸여 있다.

코드의 다음 행은 디벨로퍼 스튜디오에 의해 마지막으로 생성된 코드로, 두 개의 사선 문자(//)로 시작한다. 사이에 공백이나 탭 등 아무 것도 들어가지 않은 두 개의 사선 문자는 (문자열의 일부가 아니라면) 다음에 오는 모든 것이 주석으로 처리되어 컴파일러가 무시하도록 만든다. 이 문자는 코드에 주석 처리를 하는 두 가지 방법 중에 하나이다. 두 번째 방법은 여러 줄을 한꺼번에 주석 처리할 때 더 유용하다. 주석의 시작에 하나의 사선과 별표(/*)를 사용하고 주석의 끝에 반대로 (*/) 표시하면 된다.

마지막으로, 여러분이 추가한 실제 코드에서 여러 가지를 배울 수 있다.

```
MessageBox("Hello. This is my first Visual C++ Application!");
```

먼저, C++는 대문자를 구분한다. 함수 이름과 변수는 항상 선언된 대로 머리 글자를 대문자로 쓴다. 컴파일러는 다음의 세 가지 함수 이름을 3개의 다른 함수 이름으로 인식한다.

```
MessageBox
messageBox
messagebox
```

코드를 보이는 것처럼 MessageBox로 입력하지 않는다면, 컴파일 할 때 “선언되지 않은 식별자입니다.”라는 에러 메시지를 받는다.

C++ 규칙

C++는 대소문자를 구분하는 언어이다.

함수 이름 다음에 괄호로 둘러싸인 함수 매개 변수가 오고 마지막으로 세미콜론(;)이 온다. 모든 C++ 행은 세미콜론으로 끝난다. 이 법칙의 예외는 if, while, for 등의 연산자를 포함하며, 다음에 여러 행을 담고 있는 중괄호의 쌍이 오거나 세미콜론으로 끝나는 하나의 행이 오는 문장이다. 행의 마지막에 세미콜론을 찍지 않으면, 괄호를 닫기 전에 세미콜론이 빠졌다는 에러 메시지를 받게 된다 (이 에러 메시지는 항상 실제 세미콜론이 빠진 행의 다음 행을 가리킨다). 주석은 코드로 간주하지 않으므로 마지막에 세미콜론을 붙이지 않아도 된다. 그러나, 코드의 행과 같은 행에 주석을 가지고 있으면 주석을 시작하기 전에 코드의 마지막에 세미콜론을 붙여야 한다.

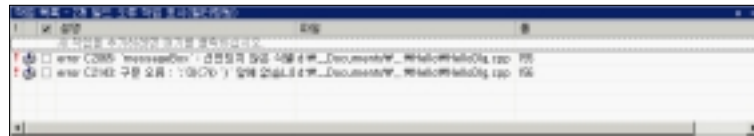
C++ 규칙

실행되는 동작을 포함하는 각 행은 세미콜론으로 끝나야 한다.

위에서 언급한 두 가지 실수(MessageBox 함수의 대문자를 지키지 않고 세미콜론을 마지막에 붙이지 않았을 경우)를 했다면 [그림 1.20]과 같이 컴파일을 할 때, 에러 메시지를 받게 된다. 에러 메시지 중 아무거나 더블 클릭을 하면, 에러가 발생한 곳을 판단해서 코드를 보여주고 표시한다.

그림 1.20

에러 메시지는 코드에서
고쳐야 할 필요가 있는
실수를 보여준다.

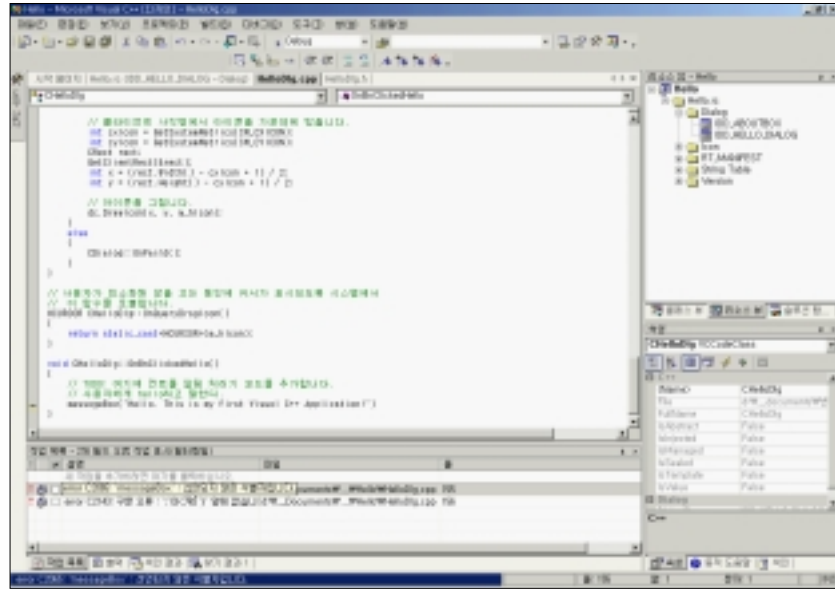
**이제 끝마무리만 남았다**

이제 응용 프로그램이 다 만들어진 상태인데, 프로젝트를 마무리 지을 두어 가지만이 우리를 기다리고 있다.

- 대화 상자의 아이콘 만들기
- 최대화, 최소화 버튼 추가하기

그림 1.21

에러 메시지를 더블 클릭하면 에러가 발생한 코드의 행으로 이동한다.



대화 상자 아이콘 만들기

방금 실행해 본 응용 프로그램 윈도우의 왼쪽 상단을 가만히 보면, M, F, C가 적힌 상자가 쌓여 있는 아이콘임을 알 수 있는데, MFC가 대체 무슨 일을 저질렀기에 그럴까?

MFC는 Microsoft Foundation Classes의 약자로서, 방금 만든 응용 프로그램은 이것을 토대로 한 것이다. 하지만, 여러분이 만든 멋진 응용 프로그램을 M, F, C의 썰렁한 아이콘이 붙어있는 채로 남들에게 보여주거란 참으로 많은 용기를 필요로 할 것이다. 어서 서둘러서 응용 프로그램을 확실히 나타내기 위한 아이콘을 그리도록 하자.

1. 솔루션 영역의 리소스 패인에서 Icon 항목을 펼치고 나서 IDR_MAINFRAME을 선택한다(〔그림 1.22〕 참조). IDE의 편집기 영역에 응용 프로그램 아이콘이 나타날 것이다.
2. 옆에 같이 나타난 그리기 도구를 사용해서 방금 만든 멋진 응용 프로그램을 잘 나타낼 수 있는 그림을 그려보자. 필자는 [그림 1.24]와 같이 그려보았다. 아이콘 근처의 빈 공간에서 마우스 오른쪽 클릭을 하면 보이는 문맥 메뉴에 색상 윈도우 표시를 선택할 수 있다(〔그림 1.23〕 참조).

그림 1,22

표준 MFC 아이콘

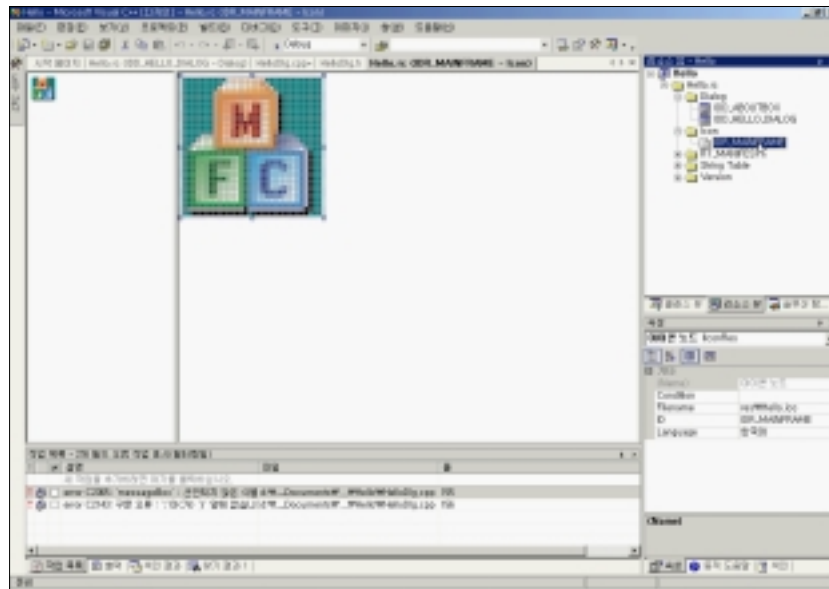


그림 1,23

색상 윈도우를 열기

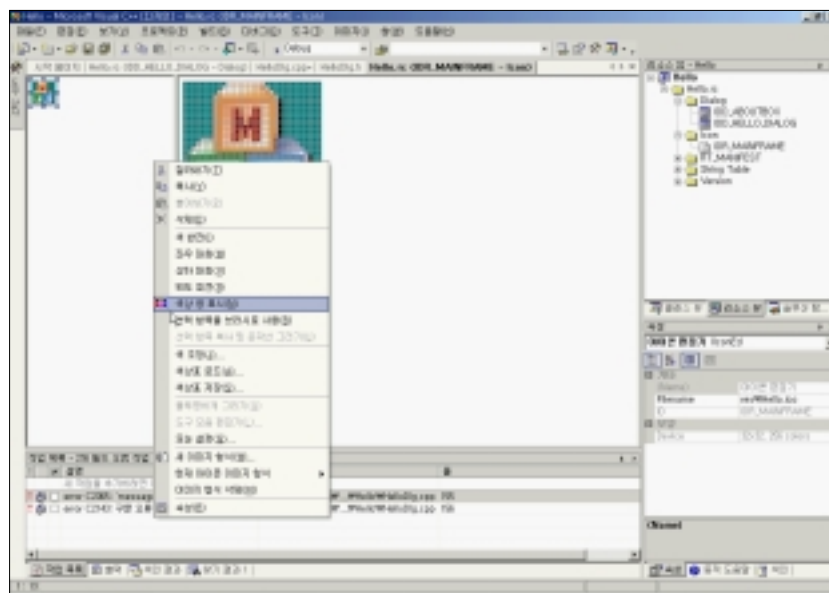
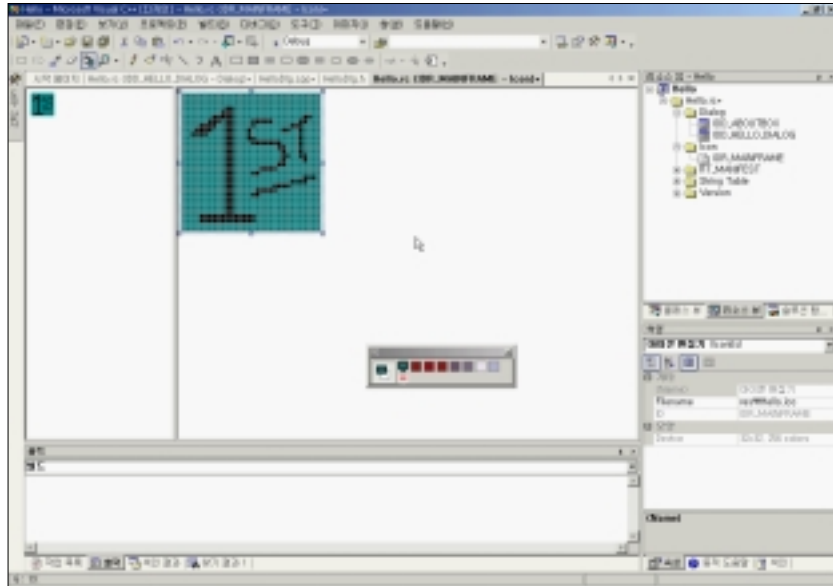


그림 1.24

여러분의 응용 프로그램을 위한 아이콘



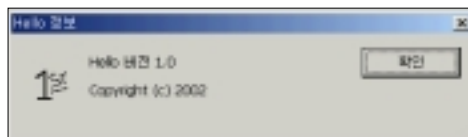
Note

디벨로퍼 스튜디오에서 만들어지는 각 아이콘은 사실 여러 개의 아이콘이다. 아이콘의 크기는 기본적으로 32×32픽셀이다. 이 아이콘은 바탕화면과 윈도우의 About 대화 상자에 나타난다. 16×16픽셀의 아이콘도 응용 프로그램 윈도우의 왼쪽 위에 보여지거나 시작 버튼이 있는 태스크 바에 사용하는 아이콘으로 사용한다. 같은 크기에 다른 칼라 수를 가진 아이콘들은 다양한 환경에 맞춰 다른 칼라 수의 아이콘이 표시된다. 여러분들은 다른 크기와 칼라 수를 가진 아이콘을 중복해서 만들 수 있다. 다른 아이콘을 고치려면 메뉴에서 이미지 | 현재 아이콘 이미지 형식 항목을 선택하면 된다.

- 이제 응용 프로그램을 다시 컴파일하고 실행하면, 왼쪽 상단에 새로 그린 아이콘이 나타났음을 확인할 수 있을 것이다. 이 아이콘을 클릭하고, 드롭다운 메뉴에서 About Hello를 선택해 보자.
- 나타나는 About 대화 상자는 Visual C++가 자동으로 만들어준 것인데, 중간쯤에 여러분이 그린 아이콘이 크게 나왔을 것이다([그림 1.25] 참조).

그림 1.25

About 윈도우



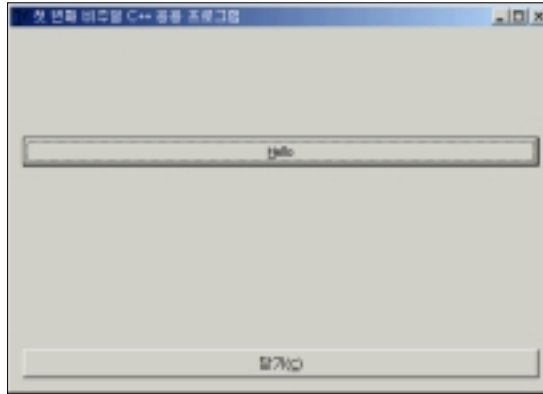
최대화 박스와 최소화 박스 추가하기

대화 상자 윈도우의 최대화 박스와 최소화 박스의 추가는 대화 상자 편집기에서 가능하다. 다음의 단계를 따르자.

1. 윈도우의 크기를 조절하려는 것처럼 대화 상자 윈도우를 자체를 선택한다.
2. 속성 패널에서 Minimize Box와 Maximize Box 속성을 찾아서 둘 다 True로 바꾼다. 다 바뀌었으면 응용 프로그램을 다시 컴파일해서 실행한다. [그림 1.26]에서 보듯이 최소화와 최대화 버튼이 나타났다.

그림 1.26

최소화 버튼과 최대화
버튼이 있는 응용 프로그램
윈도우



요약

이 장에서는 Visual C++를 사용한 응용 프로그램 만들기의 첫 맛을 들여보았다. Visual C++의 디벨로퍼 스튜디오를 구성하고 있는 영역과 각각의 기능에 대해서 잠깐 살펴보고, 디벨로퍼 스튜디오의 환경을 여러분의 입맛에 따라 재조정하는 방법도 간단하게나마 알아보았다. 아울러서, Visual C++ 마법사를 사용해서 응용 프로그램의 골격을 만들고 컨트롤에 연결 동작을 설정해 보는 것으로 이번 장의 멋진 공부를 마쳤다.

Q&A

Q 응용 프로그램의 이름 말고 메시지 박스의 타이틀을 바꾸려면 어떻게 하나요?

A 기본적으로, 메시지 박스 윈도우는 응용 프로그램의 이름을 메시지 박스의 타이틀로 사용하게 되어 있습니다. 이것을 바꾸려면 MessageBox 함수에다가 두 번째 매개 변수를 덧붙여 주면 되죠. 이 함수의 첫 번째 매개 변수는 표시될 메시지이고, 두 번째 매개 변수는 윈도우 타이틀에 표시될 텍스트입니다. 그러니까, 이를테면 OnHello에다가 다음과 같이 추가합니다.

```
// 사용자에게 hello라고 말한다.
MessageBox("Hello. This is my first Visual C++ Application!",
           "My First Application");
```

Q About 대화 상자에 속해 있는 회사 이름이나 저작권 같은 것도 쓸 수 있을까요?

A 가능합니다. About 대화 상자는 프로젝트 워크스페이스 구역의 리소스 뷰에서 찾을 수 있습니다. IDD_ABOUTBOX란 이름으로 되어 있습니다. 이것을 대화 상자 편집기에서 연 다음, 고치면 됩니다.

실습해 보기

“실습해 보기” 절에서는 배운 것을 확인하는 퀴즈와 이를 활용해서 응용력을 높이기 위한 연습문제를 풀어볼 기회를 가질 수 있게 될 것이다. 퀴즈와 연습문제의 답은 부록 A, “퀴즈 및 연습문제 해답”에 있지만, 스스로 해보기 바란다.

퀴즈

1. 버튼의 캡션은 어떻게 바꾸는가?
2. Visual C++ 응용 프로그램 마법사를 가지고 무엇을 할 수 있는가?
3. 버튼에 어떠한 동작(함수)을 연결하려면 어떻게 하는가?

연습문제

여러분의 응용 프로그램이 가지고 있는 About 윈도우에 윈도우를 하나 더 추가하자. 이 버튼을 눌렀을 때 첫 번째 윈도우에서의 그 버튼과 다른 메시지를 띄우도록 해보자.