

쉽게 배우는 MFC 윈도우 프로그래밍

Visual C++ 2019

2015·2017 버전 사용 가능



8장. 표준 컨트롤

목차

01 컨트롤 기초

02 고급 컨트롤 기법

개요

■ 컨트롤

- 표준화된 형태와 특성을 가진 윈도우
- 사용자의 입력을 받거나 정보를 보여줌

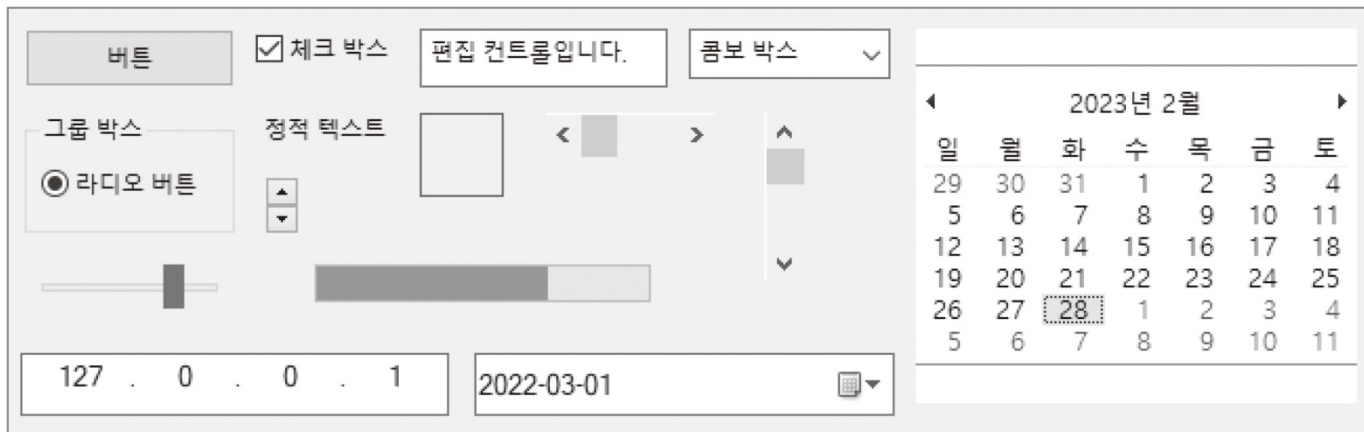


그림 8-1 다양한 종류의 컨트롤

개요

■ 컨트롤과 부모 윈도우의 통신

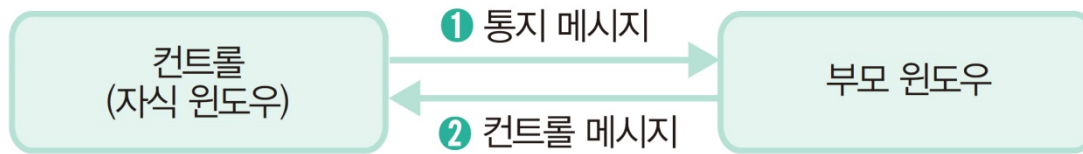


그림 8-2 컨트롤과 부모 윈도우의 통신

- 통지 메시지(Notification Message)
 - 컨트롤이 부모 윈도우에 보내는 메시지
 - 컨트롤의 상태가 변화되었음을 알림
 - 메모리 부족 등으로 인한 오류를 알림
- 컨트롤 메시지
 - 부모 윈도우가 컨트롤에 보내는 메시지
 - 컨트롤의 상태를 알아내거나 변경함
 - 컨트롤에 따라 보낼 수 있는 컨트롤 메시지의 종류가 다름

개요

표 8-1 MFC 컨트롤 클래스

컨트롤	MFC 클래스	도구 상자 아이콘
버튼	CButton	 (푸시 버튼),  (체크 박스),  (라디오 버튼),  (그룹 박스)
스태틱	CStatic	 (텍스트),  (프레임/직사각형/아이콘/비트맵/메타파일)
에디트	CEdit	 (에디트)
리스트 박스	CListBox	 (리스트 박스)
콤보 박스	CComboBox	 (콤보 박스)
스크롤바	CScrollBar	 (수평 스크롤바),  (수직 스크롤바)

개요

■ MFC 클래스 계층도

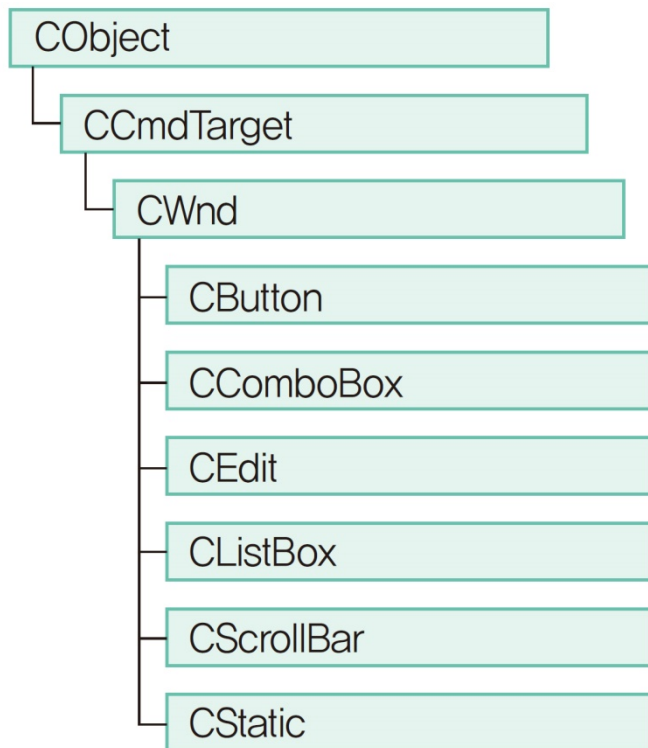


그림 8-3 MFC 클래스 계층도

버튼 컨트롤

■ 버튼 컨트롤 종류

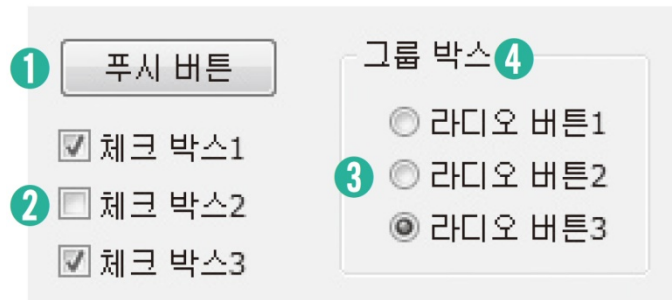


그림 8-4 버튼 컨트롤의 종류

■ 라디오 버튼 그룹

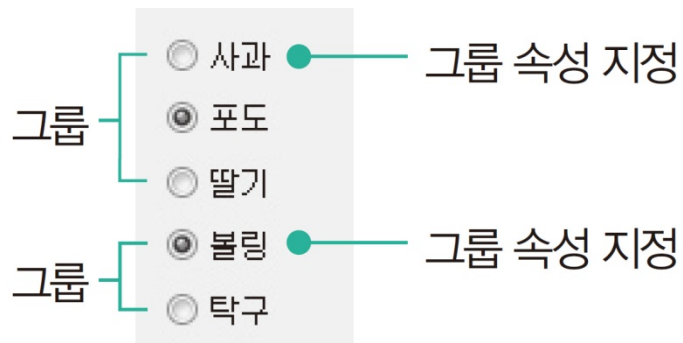


그림 8-5 라디오 버튼 그룹

버튼 컨트롤

■ 컨트롤 생성 방법

- 코드를 실행하여 만들기
- 대화상자 리소스에 포함되어 만들기

■ 컨트롤 생성 방법 1

```
CButton m_button; // C++ 버튼 객체 생성  
m_button.Create(_T("누르세요"), WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,  
CRect(100, 100, 200, 130), this, 101); // 푸시 버튼 생성
```

버튼 컨트롤

■ CButton::Create() 함수

```
BOOL CButton::Create(LPCTSTR lpszCaption, DWORD dwStyle, const RECT& rect,  
                    CWnd* pParentWnd, UINT nID);
```

- lpszCaption - 컨트롤에 표시할 문자열
- dwStyle - 컨트롤 스타일(윈도우 스타일 + 버튼 스타일)
- rect - 컨트롤 크기와 위치
- pParentWnd - 부모 윈도우 지정
- UINT nID - 컨트롤 ID

버튼 컨트롤

표 8-2 버튼 스타일(일부)

스타일	의미
BS_PUSHBUTTON	푸시 버튼
BS_DEFPUSHBUTTON	디폴트 푸시 버튼 (일반 윈도우에서는 차이가 없으나, 대화 상자에서 사용하면 <code>Enter</code> 를 누를 때 눌러진다.)
BS_CHECKBOX	체크 박스
BS_AUTOCHECKBOX	자동 체크 박스 (클릭하면 자동으로 체크 표시가 On/Off 된다.)
BS_3STATE	3상태 체크 박스
BS_AUTO3STATE	자동 3상태 체크 박스 (클릭하면 자동으로 체크 표시가 On/Grayed/Off 된다.)
BS_RADIOBUTTON	라디오 버튼
BS_AUTORADIOBUTTON	자동 라디오 버튼 (클릭하면 자동으로 선택과 선택 해제가 이루어진다.)
BS_GROUPBOX	그룹 박스

버튼 컨트롤

■ 통지 메시지 처리하기 ⇨ 메시지 핸들러 작성

```
ON_BN_CLICKED(101, OnButtonClicked)
...
void CButton1View::OnButtonClicked()
{
    MessageBox(_T("버튼을 눌렀습니다.));
}
```

■ 컨트롤 메시지 보내기 ⇨ 멤버 함수 호출

```
m_button.SetCheck(BST_CHECKED);
```



그림 8-6 컨트롤과 부모 윈도우의 통신

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

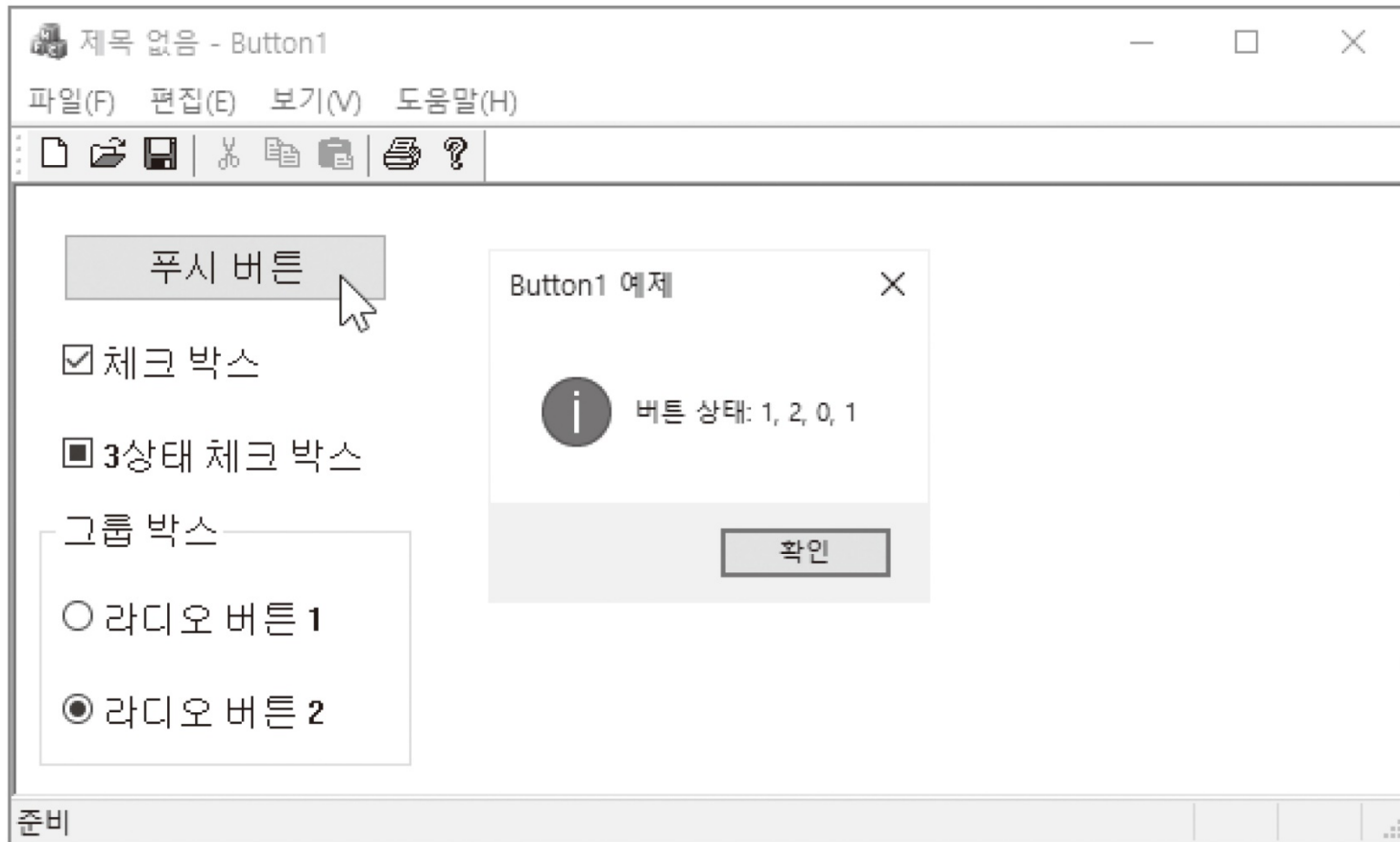


그림 8-7 실행 결과

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

```
class CButton1View : public CView
{
protected: // serialization에서만 만들어집니다.
    CButton1View();
    DECLARE_DYNCREATE(CButton1View)
```

// 특성입니다.

```
public:
    CButton1Doc* GetDocument() const;
    CButton m_pushbutton;
    CButton m_checkbox;
    CButton m_3state;
    CButton m_radio1;
    CButton m_radio2;
    CButton m_groupbox;
```

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

```
int CButton1View::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CView::OnCreate(lpCreateStruct) == -1)
        return -1;

    // 버튼 컨트롤을 생성한다.
    m_pushbutton.Create(_T("푸시 버튼"),
        WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,
        CRect(20, 20, 160, 50), this, 101);
    m_checkbox.Create(_T("체크 박스"),
        WS_CHILD|WS_VISIBLE|BS_AUTOCHECKBOX,
        CRect(20, 60, 160, 90), this, 102);
    m_3state.Create(_T("3상태 체크 박스"),
        WS_CHILD|WS_VISIBLE|BS_AUTO3STATE,
        CRect(20, 100, 160, 130), this, 103);
    m_radio1.Create(_T("라디오 버튼 1"),
        WS_CHILD|WS_VISIBLE|WS_GROUP|BS_AUTORADIOBUTTON,
        CRect(20, 170, 160, 200), this, 104);
    m_radio2.Create(_T("라디오 버튼 2"),
        WS_CHILD|WS_VISIBLE|BS_AUTORADIOBUTTON,
        CRect(20, 210, 160, 240), this, 105);
```

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

```
m_groupbox.Create(_T("그룹 박스"),  
    WS_CHILD|WS_VISIBLE|BS_GROUPBOX,  
    CRect(10, 140, 170, 250), this, 106);
```

```
// 버튼 컨트롤을 초기화한다.
```

```
m_checkbox.SetCheck(1);  
m_3state.SetCheck(2);  
m_radio2.SetCheck(1);
```

```
return 0;
```

```
}
```

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

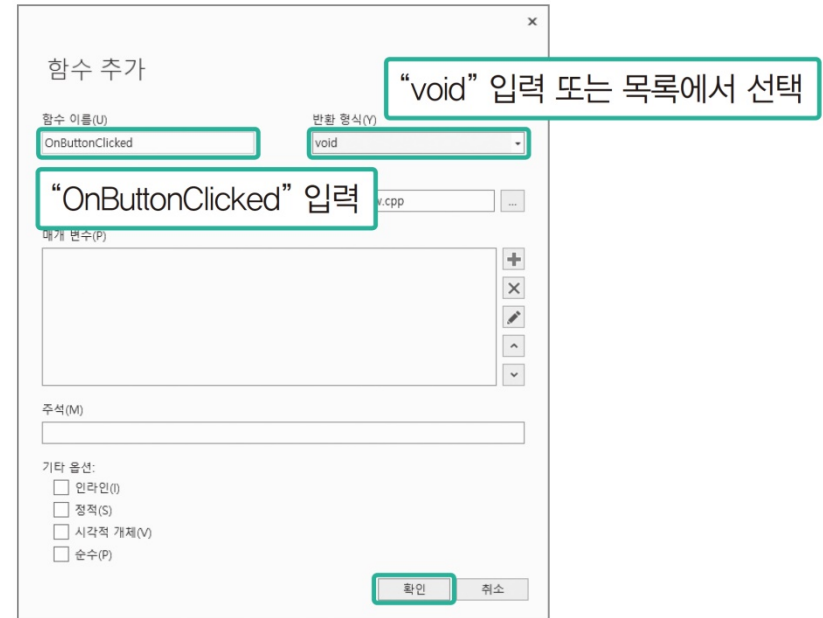
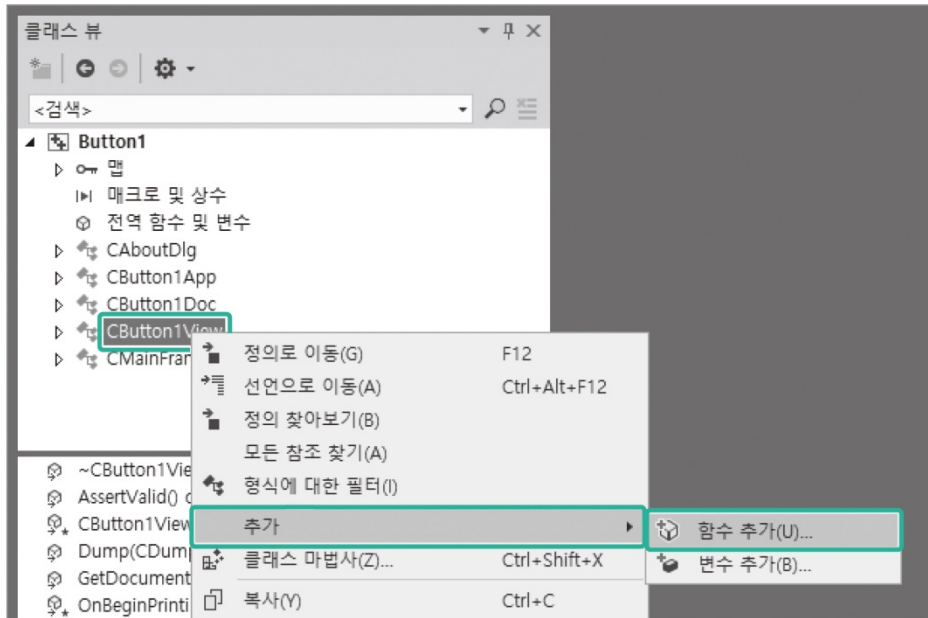


그림 8-8 함수 추가

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

```
void CButton1View::OnButtonClicked(void)
{
    int state_checkbox = m_checkbox.GetCheck();
    int state_3state = m_3state.GetCheck();
    int state_radio1 = m_radio1.GetCheck();
    int state_radio2 = m_radio2.GetCheck();

    CString str;
    str.Format(_T("버튼 상태: %d, %d, %d, %d"),
        state_checkbox, state_3state, state_radio1, state_radio2);
    MessageBox(str, _T("Button1 예제"), MB_ICONINFORMATION);
}
```

[실습 8-1] 코드를 실행하여 다양한 버튼 만들기

```
BEGIN_MESSAGE_MAP(CButton1View, CView)
    // 표준 인쇄 명령입니다.
    ON_COMMAND(ID_FILE_PRINT, &CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, &CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, &CView::OnFilePrintPreview)
    ON_WM_CREATE()
    ON_BN_CLICKED(101, OnButtonClicked)
END_MESSAGE_MAP()
```

버튼 컨트롤

■ 컨트롤 생성 방법 2

- 대화상자 리소스에 컨트롤 추가
 - 대화상자가 생성될 때 컨트롤도 자동으로 생성
- 컨트롤 변수 생성
 - 컨트롤 자체를 나타내는 변수(=컨트롤 변수)를 생성하고 이를 이용하여 컨트롤을 조작

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기

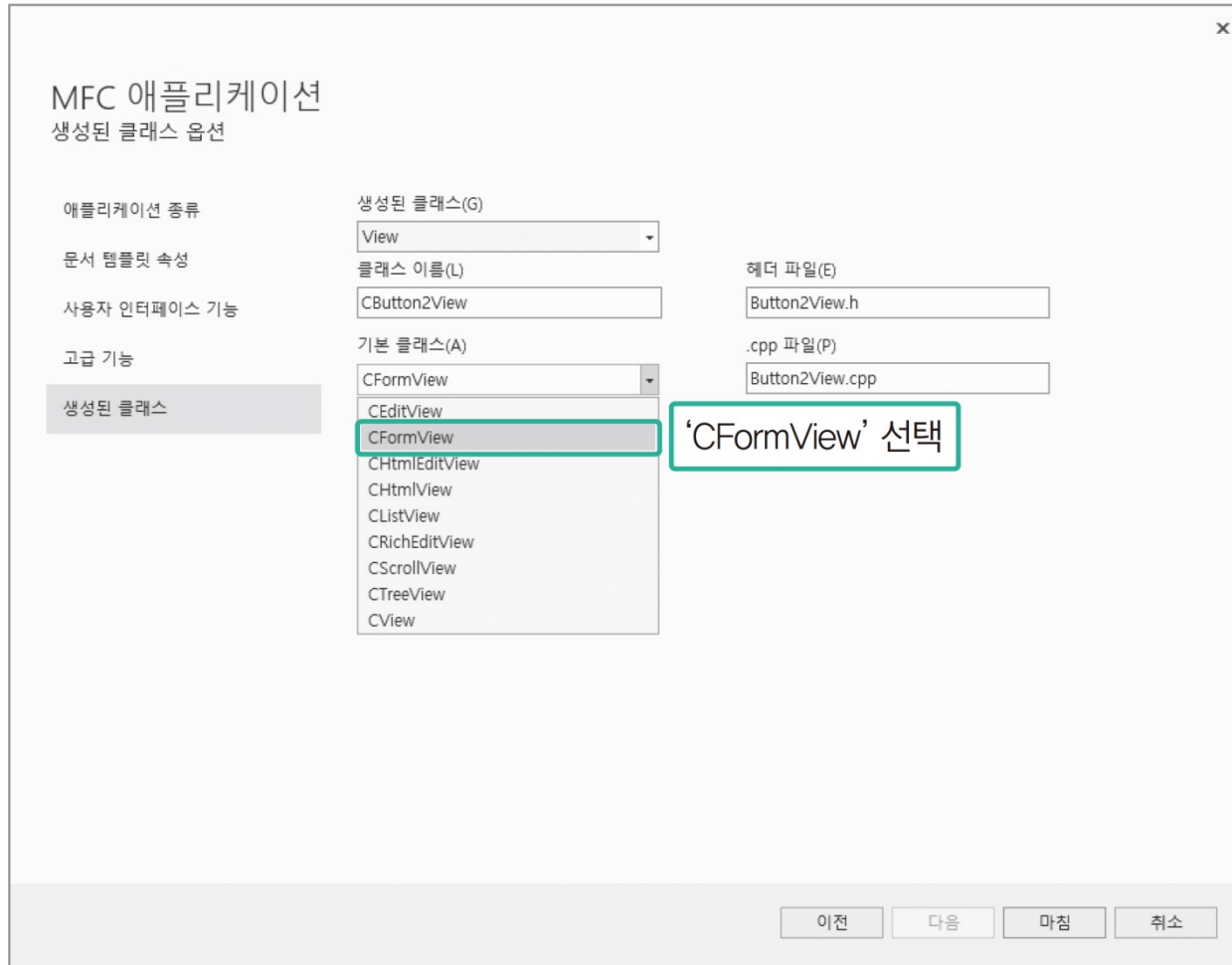


그림 8-9 응용 프로그램 마법사 마지막 단계

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기

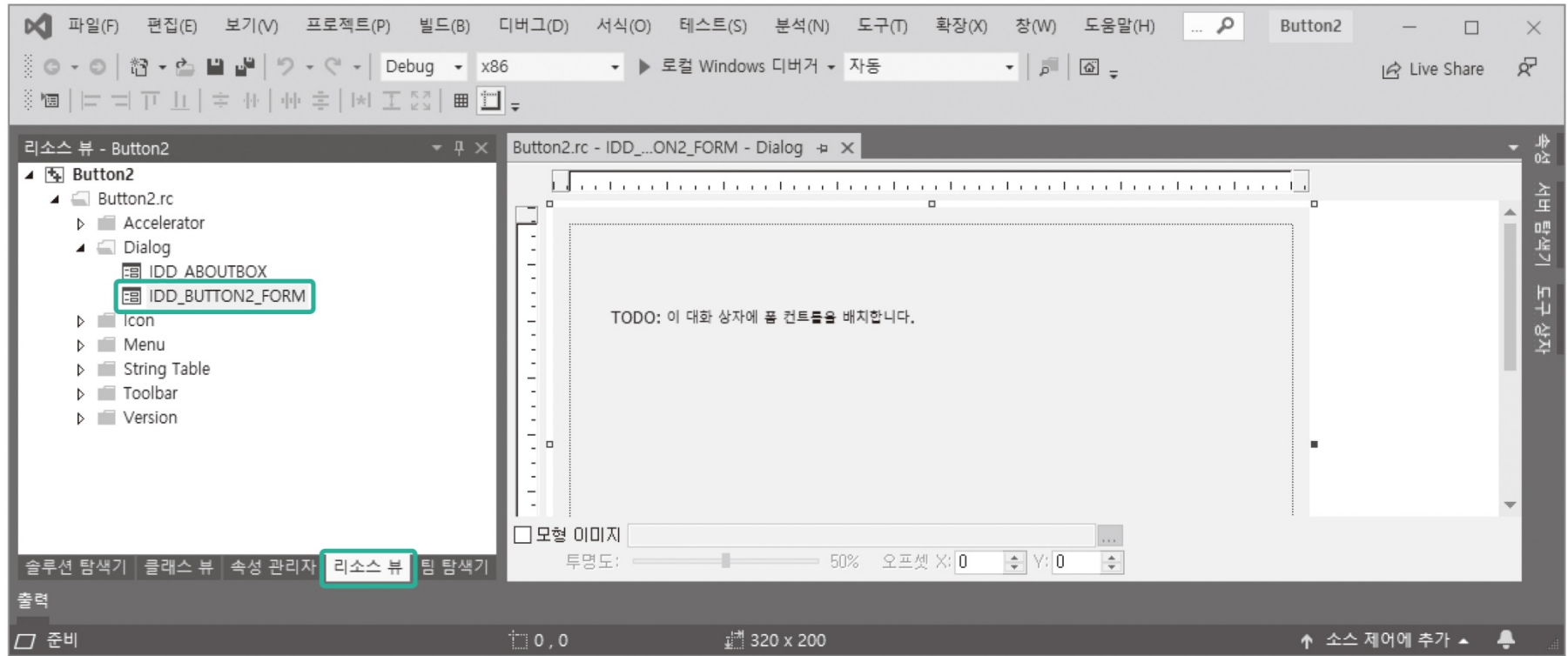


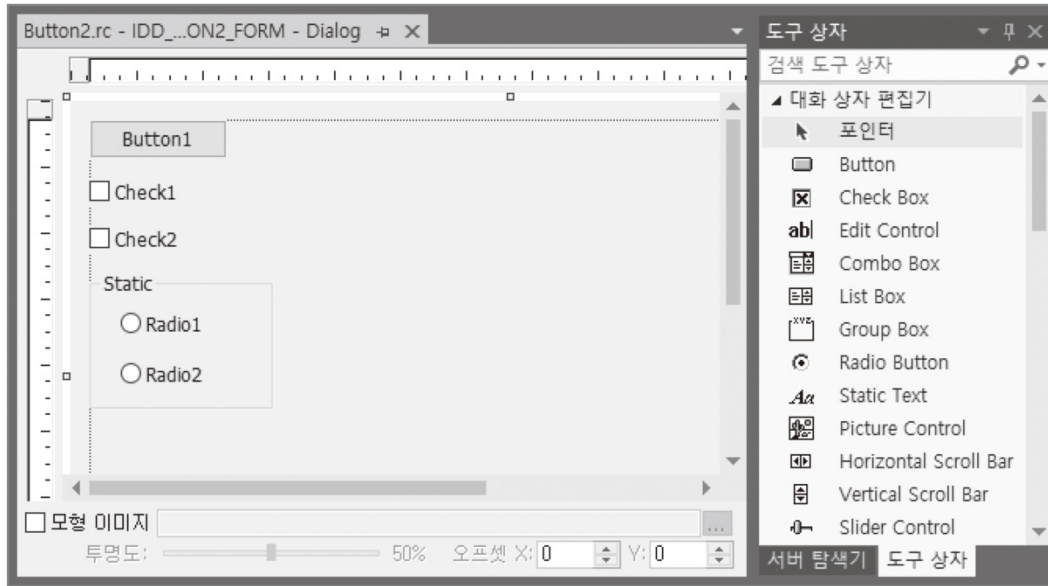
그림 8-10 대화 상자 디자인 화면

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기



그림 8-11 실행 결과

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기



도구 상자에서 선택할 대상	자동 설정되는 캡션 문자열
Button	Button1
Check Box	Check1, Check2
Group Box	Static
Radio Button	Radio1, Radio2

그림 8-12 대화 상자 디자인 – 컨트롤 추가 및 배치

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기

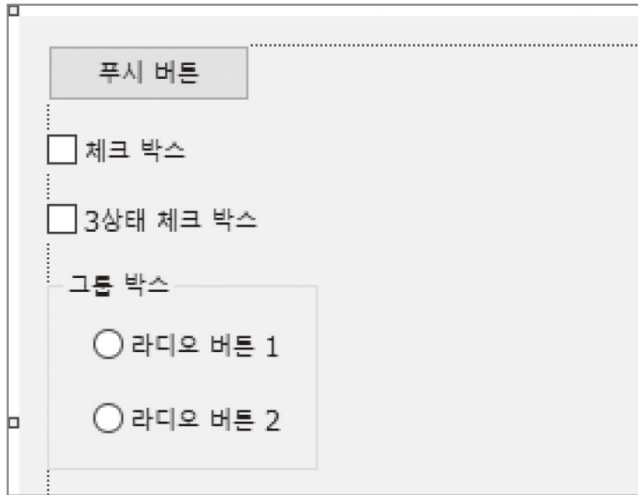
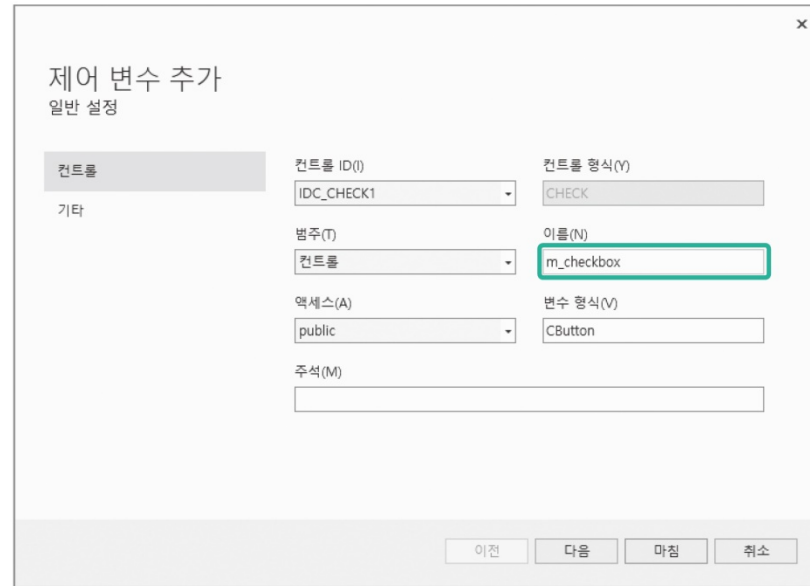
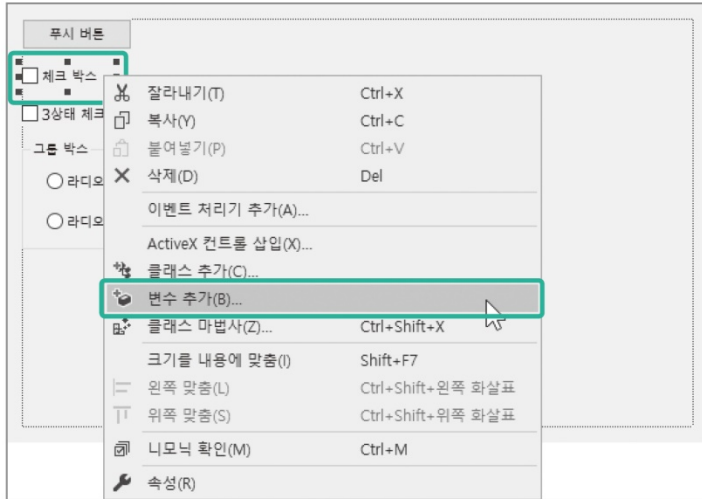


그림 8-13 대화 상자 디자인 – 컨트롤 속성 설정

ID	Caption	그 밖의 변경 사항
IDC_BUTTON1	푸시 버튼	
IDC_CHECK1	체크 박스	
IDC_CHECK2	3상태 체크 박스	'Tri-state' 속성을 True로 변경한다.
IDC_STATIC	그룹 박스	
IDC_RADIO1	라디오 버튼 1	'Group' 속성을 True로 변경한다.
IDC_RADIO2	라디오 버튼 2	

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기



컨트롤 ID	변수 이름
IDC_CHECK1	m_checkbox
IDC_CHECK2	m_3state
IDC_RADIO1	m_radio1
IDC_RADIO2	m_radio2

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기

```
void CButton2View::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    // 버튼 컨트롤을 초기화한다.
    m_checkbox.SetCheck(1);
    m_3state.SetCheck(2);
    m_radio2.SetCheck(1);
}
```

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기

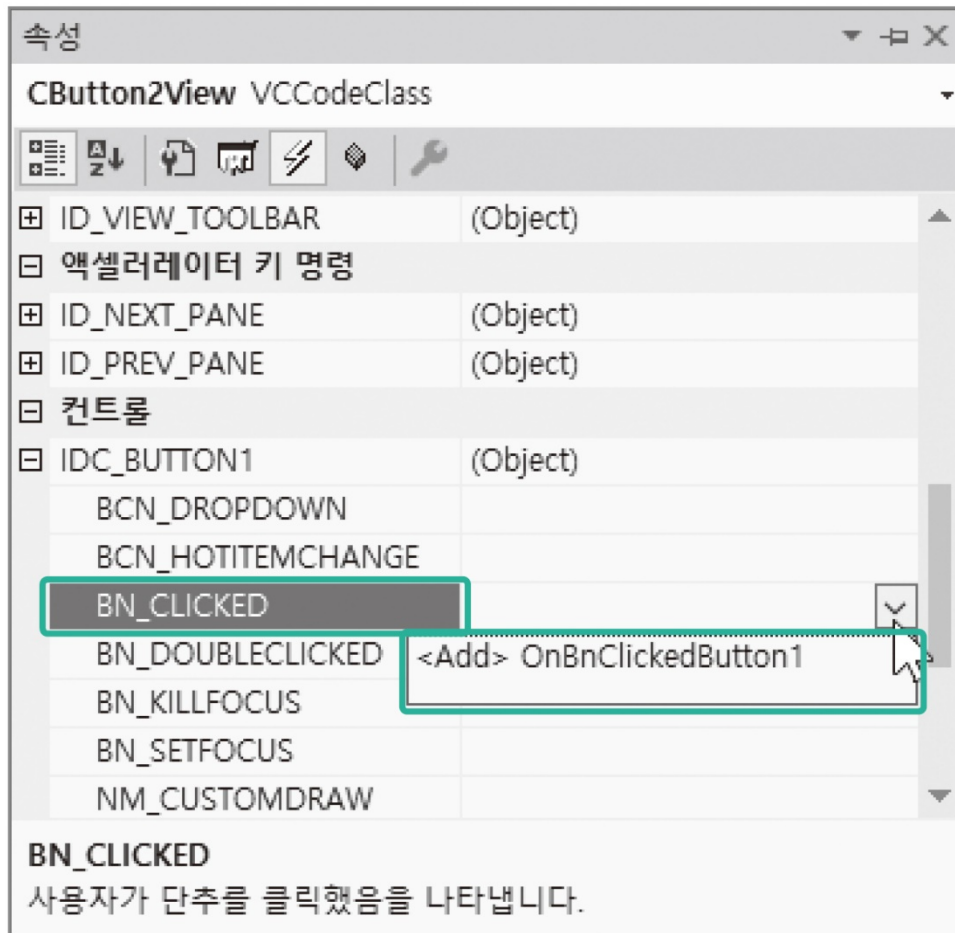


그림 8-15 통지 메시지 핸들러 추가

[실습 8-2] 대화 상자 리소스에 포함하여 다양한 버튼 만들기

```
void CButton2View::OnBnClickedButton1()
{
    int state_checkbox = m_checkbox.GetCheck();
    int state_3state = m_3state.GetCheck();
    int state_radio1 = m_radio1.GetCheck();
    int state_radio2 = m_radio2.GetCheck();

    CString str;
    str.Format(_T("버튼 상태: %d, %d, %d, %d"),
        state_checkbox, state_3state, state_radio1, state_radio2);
    MessageBox(str, _T("Button2 예제"), MB_ICONINFORMATION);
}
```


스태틱 컨트롤

■ 스태틱 컨트롤 종류

- 텍스트
- 프레임(색이 채워지지 않은 직사각형)
- 직사각형(색이 채워진 직사각형)
- 아이콘
- 비트맵
- 확장 메타파일(Enhanced Metafile)

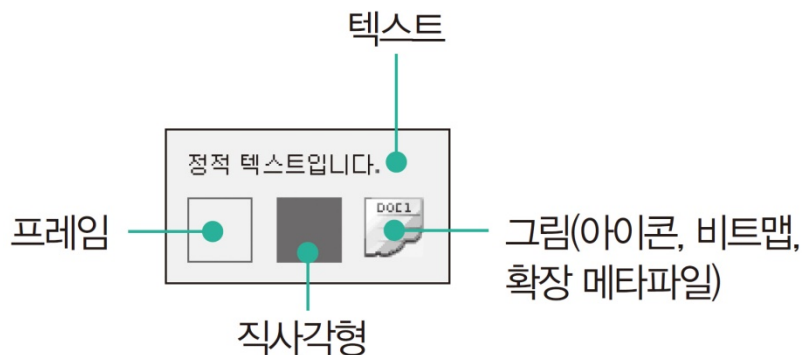


그림 8-16 스태틱 컨트롤

스태틱 컨트롤

■ 스태틱 컨트롤 통지 메시지

- 일반적으로 통지 메시지를 발생시키지 않음
- 단, SS_NOTIFY 스타일을 설정하면 통지 메시지 발생
 - STN_CLICKED(클릭), STN_DBLCLK(더블 클릭), STN_DISABLE(비활성화), STN_ENABLE(활성화)

[실습 8-3] 스테틱 컨트롤 활용하기

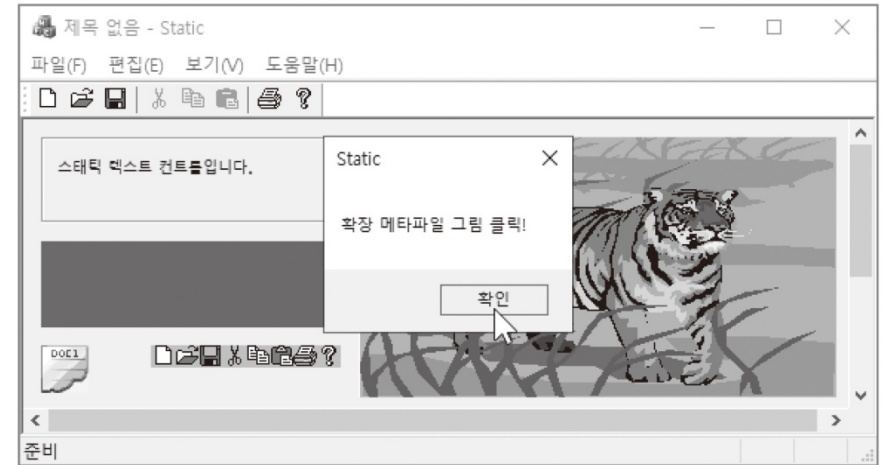


그림 8-17 실행 결과

[실습 8-3] 스택 컨트롤 활용하기

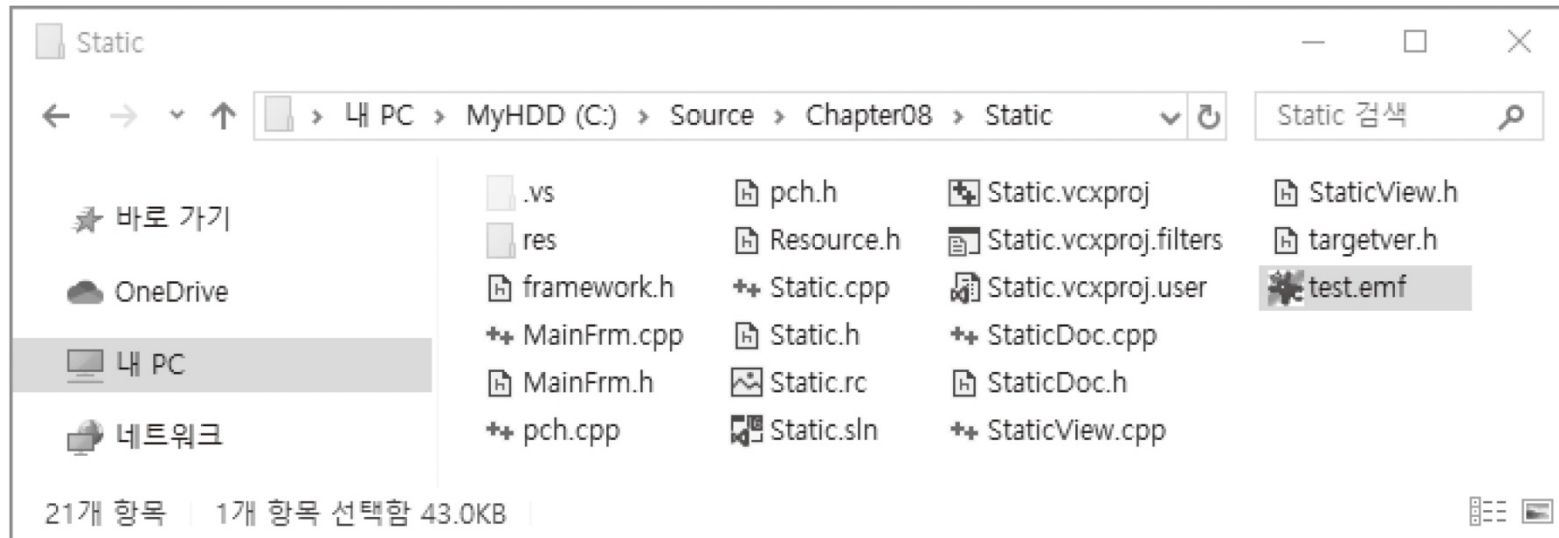
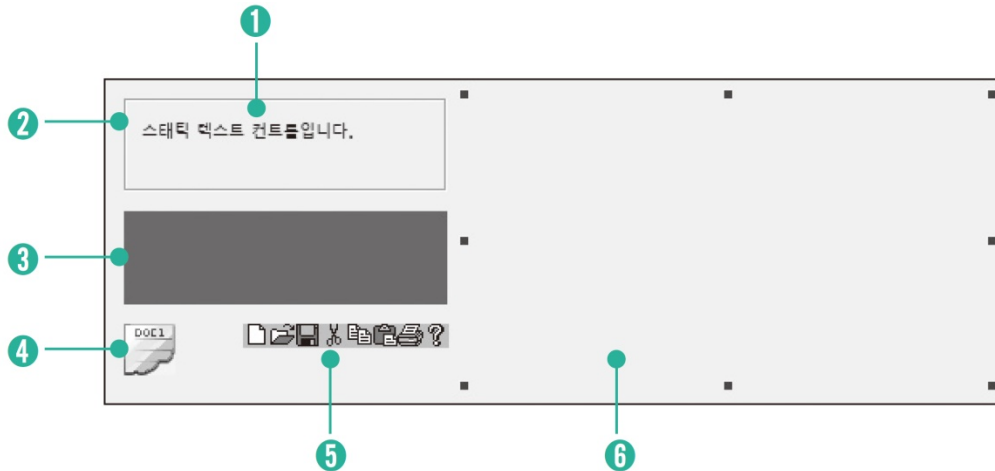


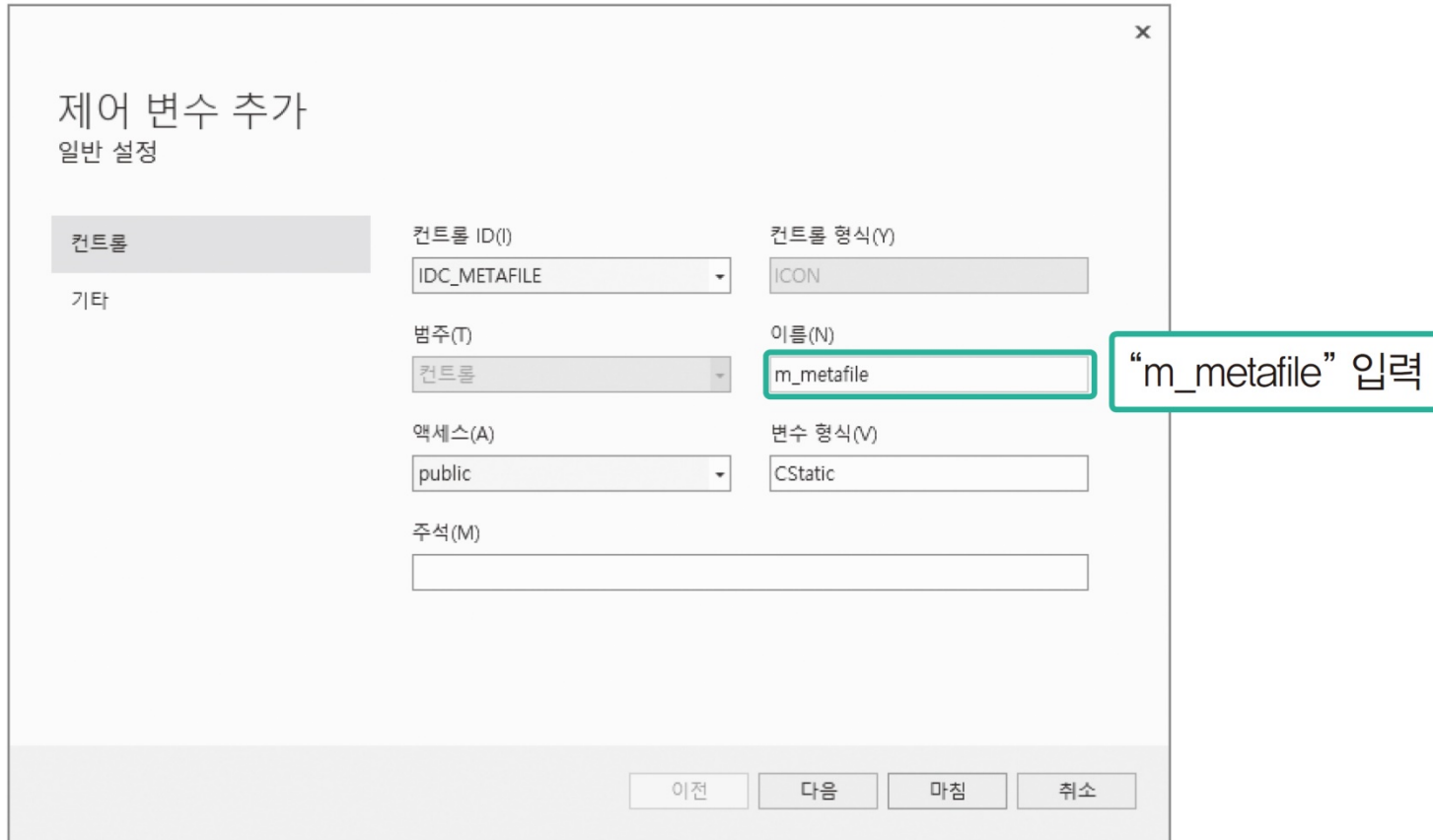
그림 8-18 확장 메타파일 준비

[실습 8-3] 스택 컨트롤 활용하기



번호	변경 사항
①	'Caption'을 "스택 텍스트 컨트롤입니다."로 변경한다.
②	'Color'를 Etched로 변경한다.
③	'Type'을 Rectangle로 변경한다.
④	'Type'을 Icon으로 변경하고 'Image'로 IDR_StaticTYPE을 선택한다.
⑤	'Type'을 Bitmap으로 변경하고 'Image'로 IDR_MAINFRAME을 선택한다.
⑥	'ID'를 IDC_METAFILE로, 'Type'을 Enhanced Metafile로, 'Notify'를 True로 변경한다.

[실습 8-3] 스택 컨트롤 활용하기



제어 변수 추가
일반 설정

컨트롤
기타

컨트롤 ID(I)
IDC_METAFILE

컨트롤 형식(Y)
ICON

범주(T)
컨트롤

이름(N)
m_metafile

액세스(A)
public

변수 형식(V)
CStatic

주석(M)

이전 다음 마침 취소

“m_metafile” 입력

그림 8-20 컨트롤 변수 생성

[실습 8-3] 스테틱 컨트롤 활용하기

```
void CStaticView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();
    m_metafile.SetEnhMetaFile(_T("test.emf"));
}
```

[실습 8-3] 스태틱 컨트롤 활용하기

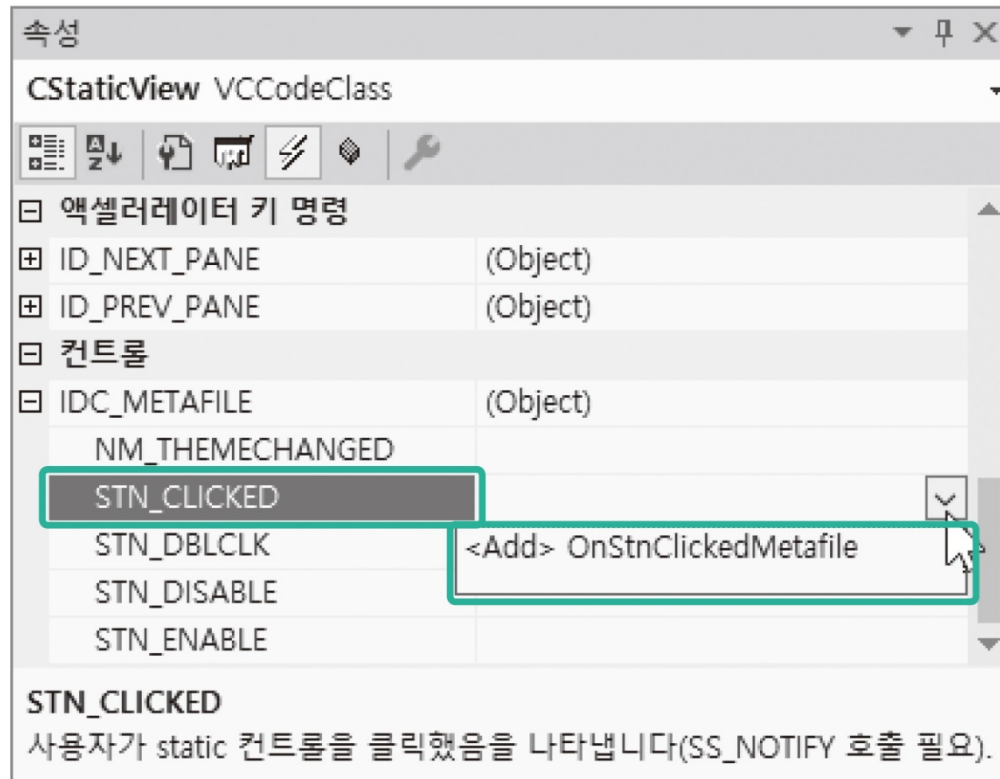


그림 8-21 통지 메시지 핸들러 추가

에디트 컨트롤

■ 에디트 컨트롤 종류

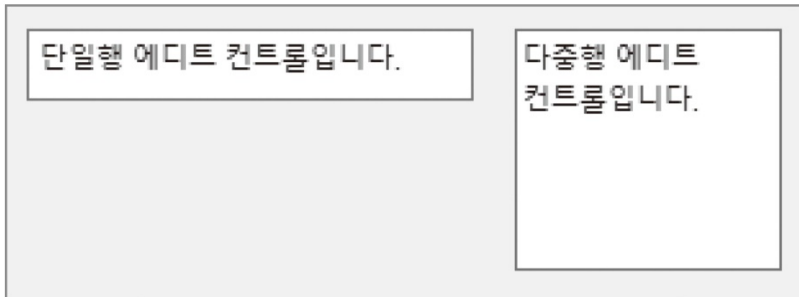


그림 8-22 에디트 컨트롤

에디트 컨트롤

표 8-3 에디트 컨트롤 스타일

스타일/속성 창 항목	기능
ES_AUTOHSCROLL/Auto HScroll	줄 끝에 도달하면 자동으로 수평 스크롤 한다.
ES_AUTOVSCROLL/Auto VScroll	줄 끝에 도달하면 자동으로 수직 스크롤 한다.
ES_CENTER/Align Text: Centered	텍스트를 가운데 정렬한다.
ES_LEFT/Align Text: Left	텍스트를 왼쪽 정렬한다.
ES_LOWERCASE/Lowercase	입력된 모든 문자를 소문자로 변환한다.
ES_MULTILINE/Multiline	다중행 에디트 컨트롤로 동작한다.
ES_NOHIDESEL/No Hide Selection	컨트롤이 키보드 포커스를 잃어도 선택된 텍스트가 반전 상태를 계속 유지한다.
ES_NUMBER/Number	숫자만 입력할 수 있다.
ES_PASSWORD/Password	입력된 글자를 *로 표시한다. (단일행 에디트 컨트롤에만 사용 가능)

에디트 컨트롤

ES_READONLY/Read Only	텍스트를 읽기만 할 수 있다.
ES_RIGHT/Align Text: Right	텍스트를 오른쪽 정렬한다.
ES_UPPERCASE/Uppercase	입력된 모든 문자를 대문자로 변환한다.
ES_WANTRETURN/Want Return	<p>이 스타일을 지정하지 않으면 대화 상자에서 <code>Enter</code>를 이용한 줄 바꿈이 되지 않는다. <code>Ctrl</code> + <code>Enter</code>를 이용하면 이 스타일과 무관하게 줄 바꿈을 할 수 있다.</p> <p>(다중행 에디트 컨트롤에만 사용 가능)</p>

에디트 컨트롤

표 8-4 에디트 컨트롤 통지 메시지

통지 메시지	의미
EN_CHANGE	사용자가 내용을 변경하면 화면에 컨트롤을 다시 그리는데, 그 후에 이 메시지가 발생한다.
EN_ERRSPACE	메모리가 부족하다.
EN_HSCROLL	사용자가 에디트 컨트롤의 수평 스크롤바를 클릭했다.
EN_KILLFOCUS	키보드 포커스를 잃었다.
EN_MAXTEXT	더는 문자를 입력할 수 없다. CEdit::SetLimitText() 함수로 문자의 개수를 제한한 경우나 ES_AUTOHSCROLL, ES_AUTOVSCROLL 스타일을 지정하지 않은 상태에서 줄 끝까지 입력할 때 발생한다.
EN_SETFOCUS	키보드 포커스를 얻었다.
EN_UPDATE	사용자가 내용을 변경하면 화면에 컨트롤을 다시 그리는데, 그 전에 이 메시지가 발생한다.
EN_VSCROLL	사용자가 에디트 컨트롤의 수직 스크롤바를 클릭했다.

에디트 컨트롤

■ 텍스트를 변경하거나 입력된 텍스트를 알아내기

```
m_edit.SetWindowText(_T("초기 값")); // 에디트 컨트롤의 텍스트를 변경한다.  
CString str;  
m_edit.GetWindowText(str); // 에디트 컨트롤에 입력된 텍스트를 얻는다.
```

■ 입력 가능한 텍스트의 길이를 제한하기

```
m_edit.SetLimitText(10); // 열 글자로 제한한다.
```

■ 각종 편집 작업 - 삭제, 잘라내기, 복사, 붙여넣기, 실행 취소

```
m_edit.Clear(); // 현재 선택된 텍스트를 삭제한다.  
m_edit.Cut(); // 텍스트를 잘라내서 클립보드에 저장한다.  
m_edit.Copy(); // 텍스트를 복사해서 클립보드에 저장한다.  
m_edit.Paste(); // 클립보드에 저장된 내용을 붙여넣는다.  
m_edit.Undo(); // 이전에 실행한 동작을 취소한다.
```

에디트 컨트롤

■ 텍스트 선택과 치환하기

- ❶ `m_edit.SetSel(5, 7);` // 여섯 번째 글자부터 시작해서 글자 두 개가 선택된다.
- ❷ `m_edit.ReplaceSel(_T("ABC"));` // 현재 선택된 부분을 새로운 텍스트로 치환한다.



그림 8-23 텍스트 선택과 치환 예

```
int len = m_edit.GetWindowTextLength();  
m_edit.SetSel(len, len);  
m_edit.ReplaceSel(_T("새로 추가한 텍스트"));
```

[실습 8-4] 에디트 컨트롤 활용하기

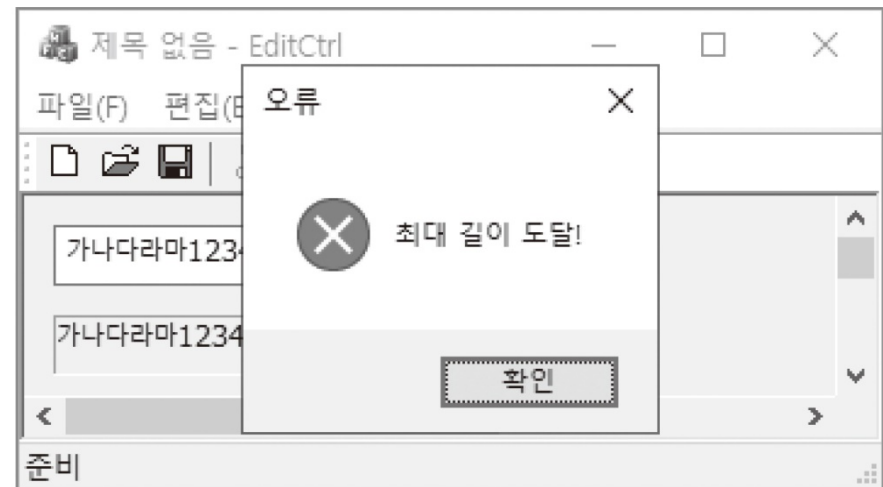
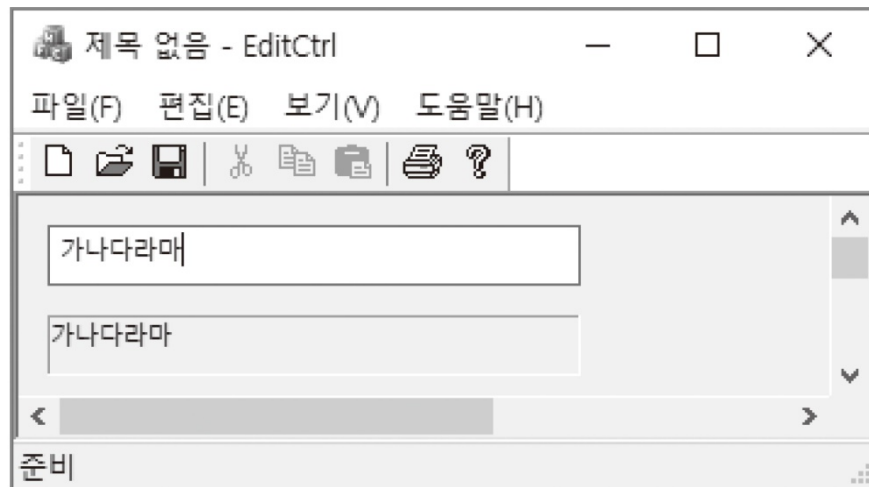


그림 8-24 실행 결과

[실습 8-4] 에디트 컨트롤 활용하기

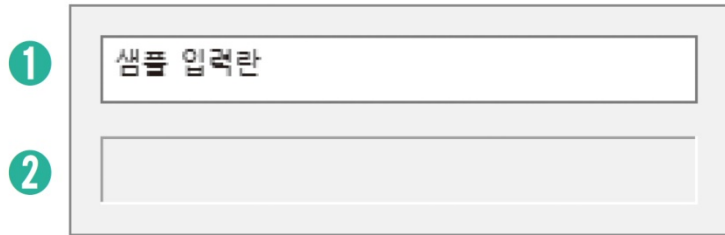


그림 8-25 대화 상자 디자인

번호	변경 사항
①	'ID'를 IDC_MYEDIT로 변경한다.
②	'ID'를 IDC_MYSTATIC으로 변경한다. 'Caption'을 삭제하고 'Sunken'을 True로 설정한다.

[실습 8-4] 에디트 컨트롤 활용하기

제어 변수 추가
일반 설정

컨트롤
기타

컨트롤 ID(I)
IDC_MYEDIT

컨트롤 형식(Y)
EDIT

범주(T)
컨트롤

이름(N)
m_edit

액세스(A)
public

변수 형식(V)
CEdit

주석(M)

이전 다음 마침 취소

그림 8-26 컨트롤 변수 생성

[실습 8-4] 에디트 컨트롤 활용하기

```
void CEditCtrlView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();
    m_edit.SetLimitText(20);
}
```

[실습 8-4] 에디트 컨트롤 활용하기

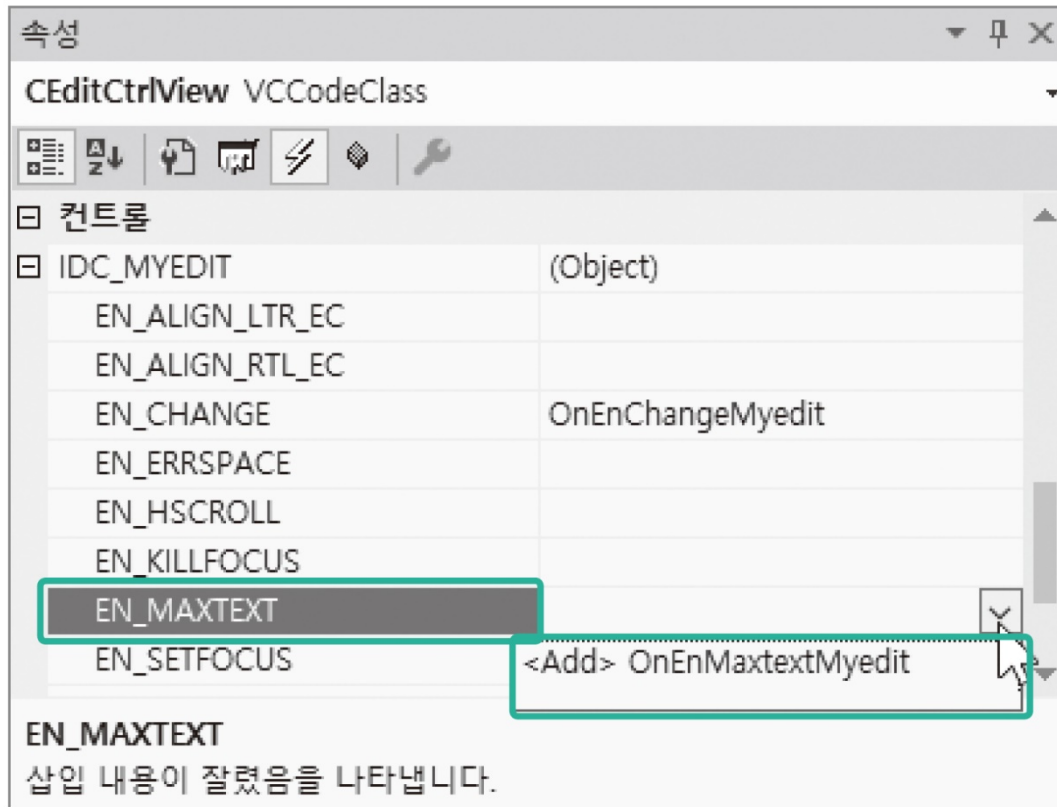


그림 8-27 통지 메시지 핸들러 생성

[실습 8-4] 에디트 컨트롤 활용하기

```
void CEditCtrlView::OnEnChangeMyedit()
{
    CString str;
    m_edit.GetWindowText(str);
    m_static.SetWindowText(str);
}
```

```
Void CEditCtrlView::OnEnMaxtextMyedit()
{
    MessageBox(_T("최대 길이 도달!"), _T("오류"), MB_ICONERROR);
}
```

리스트 박스 컨트롤

■ 리스트 박스 컨트롤 종류



그림 8-28 리스트 박스 컨트롤

리스트 박스 컨트롤

표 8-5 리스트 박스 컨트롤 스타일

스타일/속성 창 항목	기능
LBS_DISABLENOSCROLL/ Disable No Scroll	표시할 항목의 개수가 적어도 수직 스크롤바가 사라지지 않는다.
LBS_EXTENDEDSEL/ Selection: Extended	<code>Shift</code> , <code>Ctrl</code> 과 마우스 클릭을 이용한 다중 선택이 가능하다.
LBS_HASSTRINGS/ Has Strings	LBS_OWNERDRAW* 스타일을 지정하지 않았을 때의 기본 스타일로, 컨트롤이 문자열을 저장하고 관리한다.
LBS_MULTICOLUMN/ Multicolumn	여러 열 ^{Column} 로 구성된 리스트 박스를 생성하며, 항목이 많으면 수평 스크롤이 가능하다.
LBS_MULTIPLESEL/ Selection: Multiple	마우스 클릭을 이용한 다중 선택이 가능하다.
LBS_NODATA/No Data	항목 데이터를 컨트롤이 아닌 부모 윈도우가 유지하며, 필요할 때마다 부모 윈도우가 직접 그린다. 항목이 1,000개 이상일 경우에 사용을 권장한다.

리스트 박스 컨트롤

LBS_NOINTEGRALHEIGHT/ No Integral Height	리스트 박스의 높이가 응용 프로그램이 지정한 크기로 유지된다. 이 스타일을 지정하면 항목 일부가 잘려서 보이는 경우가 생길 수 있다.
LBS_NOREDRAW/No Redraw	항목이 변해도 리스트 박스 컨트롤을 다시 그리지 않는다.
LBS_NOSEL/Selection: None	항목을 볼 수 있지만 선택할 수는 없다.
LBS_NOTIFY/Notify	사용자가 항목을 클릭하거나 더블 클릭하면 부모 윈도우에 통지 메시지를 보낸다.
LBS_OWNERDRAWFIXED/ Owner Draw: Fixed	부모 윈도우가 리스트 박스 항목을 직접 그리되 항목의 높이가 일정한 경우이다.
LBS_OWNERDRAWVARIABLE/ Owner Draw: Variable	부모 윈도우가 리스트 박스 항목을 직접 그리되 항목의 높이가 일정하지 않은 경우이다.
LBS_SORT/Sort	항목이 문자열인 경우 정렬하여 표시한다.
LBS_STANDARD/Notify, Sort, Border, Vertical Scrollbar	LBS_NOTIFY, LBS_SORT, WS_VSCROLL, WS_BORDER 스타일의 조합이다.

리스트 박스 컨트롤

LBS_USETABSTOPS/ Use Tabstops	이 스타일을 지정하면 항목 문자열에 포함된 탭 문자("\t")를 제대로 처리할 수 있다.
LBS_WANTKEYBOARDINPUT/ Want Key Input	리스트 박스 컨트롤이 키보드 포커스를 가진 상태에서 사용자가 키를 누르면 부모 윈도우가 이를 감지하여 특별한 처리를 할 수 있다.

리스트 박스 컨트롤

표 8-6 리스트 박스 컨트롤 통지 메시지

통지 메시지	의미
LBN_DBLCLK	사용자가 항목을 더블 클릭했다.
LBN_SELCHANGE	사용자가 선택을 변경했다.
LBN_SELCANCEL	사용자가 선택을 취소했다.
LBN_SETFOCUS	키보드 포커스를 얻었다.
LBN_KILLFOCUS	키보드 포커스를 잃었다.
LBN_ERRSPACE	메모리가 부족하다.

리스트 박스 컨트롤

■ 항목을 추가하거나 삭제하기

```
m_list.AddString(_T("사과")); // 문자열 항목을 추가한다.  
m_list.DeleteString(3); // 네 번째 항목을 삭제한다.
```

■ 항목을 선택된 상태로 만들기

```
/* 단일 선택 리스트 박스 컨트롤인 경우 */  
m_list.SetCurSel(2); // 세 번째 항목을 선택한다.  
  
/* 다중 선택 리스트 박스 컨트롤인 경우 */  
m_list.SetSel(2); // 세 번째 항목을 선택한다.  
m_list.SetSel(3, FALSE); // 네 번째 항목을 선택 해제한다
```

리스트 박스 컨트롤

■ 선택된 항목을 알아내기

```
/* 단일 선택 리스트 박스 컨트롤인 경우 */  
int nIndex = m_list.GetCurSel();  
if (nIndex != LB_ERR) {  
    CString str;  
    m_list.GetText(nIndex, str);  
}
```

```
/* 다중 선택 리스트 박스 컨트롤인 경우 */  
int nIndex = m_list.GetCaretIndex();  
if (nIndex != LB_ERR) {  
    CString str;  
    m_list.GetText(nIndex, str);  
}
```

[실습 8-5] 리스트 박스 컨트롤 활용하기

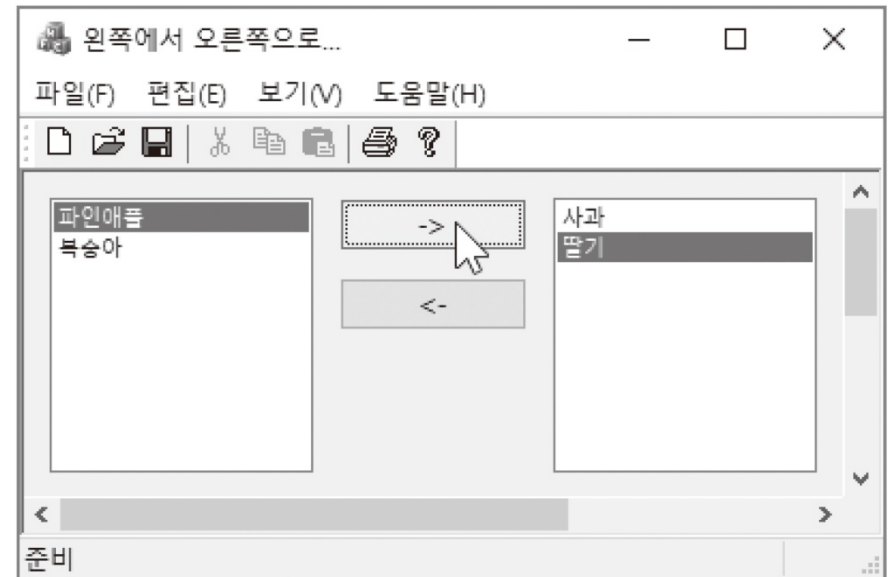
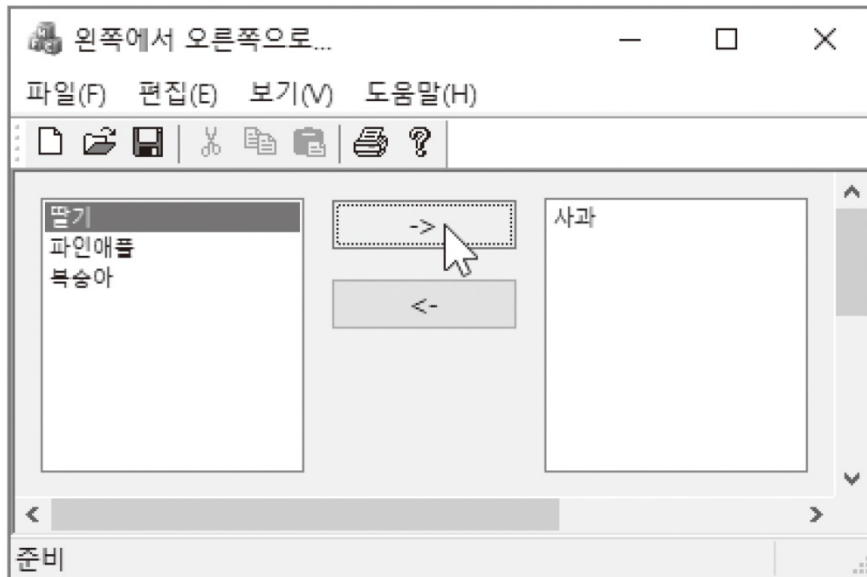


그림 8-29 실행 결과

[실습 8-5] 리스트 박스 컨트롤 활용하기

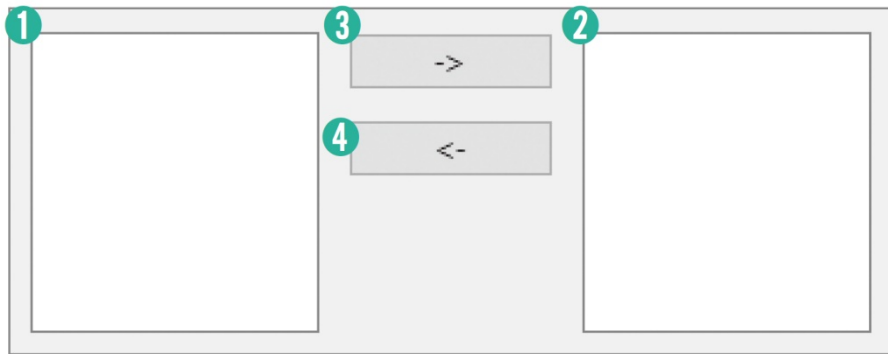


그림 8-30 대화 상자 디자인

번호	변경 사항
1	'Sort'를 False로 변경한다.
2	'Sort'를 False로 변경한다.
3	'ID'를 IDC_RIGHT로, 'Caption'을 "->"로 변경한다.
4	'ID'를 IDC_LEFT로, 'Caption'을 "<-"로 변경한다.

[실습 8-5] 리스트 박스 컨트롤 활용하기

제어 변수 추가
일반 설정

컨트롤
기타

컨트롤 ID(I)
IDC_LIST1

컨트롤 형식(Y)
LISTBOX

범주(T)
컨트롤

이름(N)
m_list1

액세스(A)
public

변수 형식(V)
CListBox

주석(M)

이전 다음 마침 취소

그림 8-31 컨트롤 변수 생성

[실습 8-5] 리스트 박스 컨트롤 활용하기

```
void CListBoxView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    m_list1.AddString(_T("사과"));
    m_list1.AddString(_T("딸기"));
    m_list1.AddString(_T("파인애플"));
    m_list1.AddString(_T("복숭아"));
}
```

[실습 8-5] 리스트 박스 컨트롤 활용하기

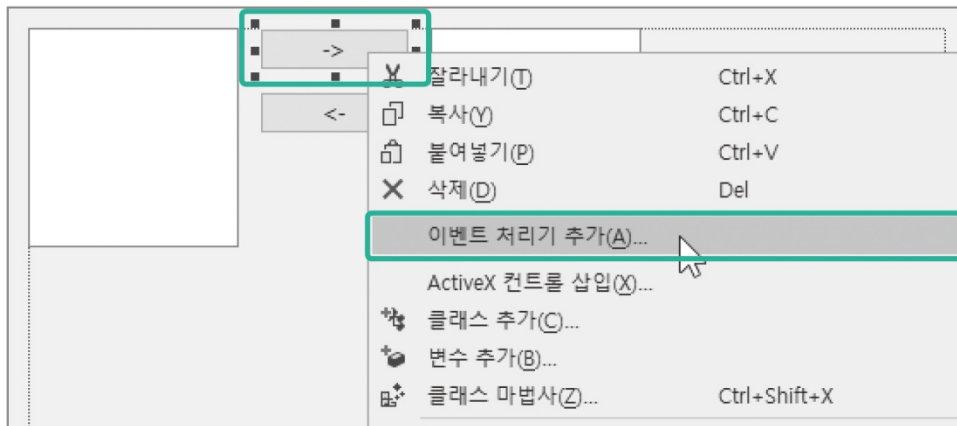


그림 8-32 통지 메시지 핸들러 생성

[실습 8-5] 리스트 박스 컨트롤 활용하기

```
void CListBoxView::OnBnClickedRight()
{
    int nIndex = m_list1.GetCurSel();
    if (nIndex != LB_ERR) {
        CString str;
        m_list1.GetText(nIndex, str);
        m_list1.DeleteString(nIndex);
        m_list1.SetCurSel(nIndex); // 삭제된 다음 위치의 항목이 자동 선택됨
        m_list2.AddString(str);
        AfxGetMainWnd()->SetWindowText(_T("왼쪽에서 오른쪽으로..."));
    }
}
```

[실습 8-5] 리스트 박스 컨트롤 활용하기

```
void CListBoxView::OnBnClickedLeft()
{
    int nIndex = m_list2.GetCurSel();
    if (nIndex != LB_ERR) {
        CString str;
        m_list2.GetText(nIndex, str);
        m_list2.DeleteString(nIndex);
        m_list2.SetCurSel(nIndex); // 삭제된 다음 위치의 항목이 자동 선택됨
        m_list1.AddString(str);
        AfxGetMainWnd()->SetWindowText(_T("오른쪽에서 왼쪽으로..."));
    }
}
```

콤보 박스 컨트롤

■ 콤보 박스 컨트롤 종류



그림 8-33 콤보 박스 컨트롤

콤보 박스 컨트롤

■ 콤보 박스 컨트롤에 초기 데이터 넣기

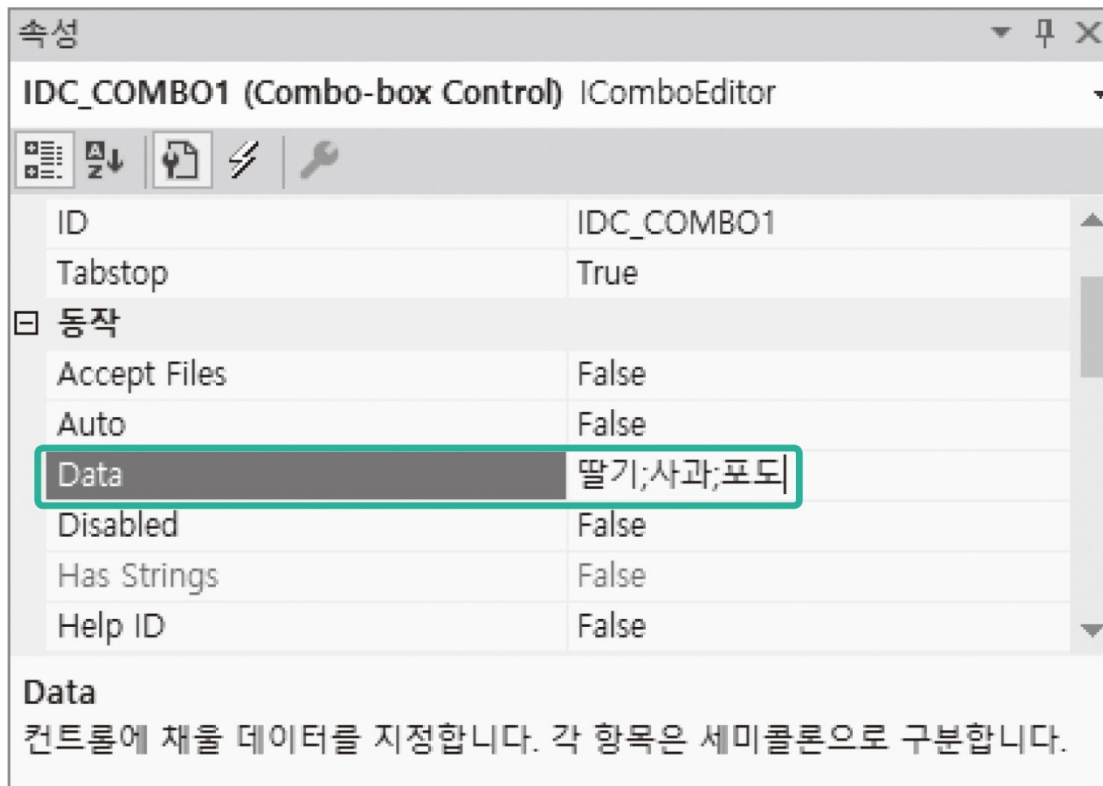


그림 8-34 콤보 박스 컨트롤에 초기 데이터 넣기

콤보 박스 컨트롤

표 8-7 콤보 박스 컨트롤 스타일

스타일/속성 창 항목	기능
CBS_AUTOHSCROLL/Auto	= ES_AUTOHSCROLL
CBS_DISABLENOSCROLL/Disable No Scroll	= LBS_DISABLENOSCROLL
CBS_DROPDOWN/Type: Dropdown	드롭다운 스타일을 지정한다.
CBS_DROPDOWNLIST/Type: Drop List	드롭다운 리스트 스타일을 지정한다.
CBS_HASSTRINGS/Has Strings	= LBS_HASSTRINGS
CBS_LOWERCASE/Lowercase	= ES_LOWERCASE
CBS_NOINTEGRALHEIGHT/No Integral Height	= LBS_NOINTEGRALHEIGHT
CBS_OWNERDRAWFIXED/Owner Draw: Fixed	= LBS_OWNERDRAWFIXED
CBS_OWNERDRAWVARIABLE/Owner Draw: Variable	= LBS_OWNERDRAWVARIABLE

콤보 박스 컨트롤

CBS_SIMPLE/Type: Simple	단순 스타일을 지정한다.
CBS_SORT/Sort	= LBS_SORT
CBS_UPPERCASE/Uppercase	= ES_UPPERCASE

콤보 박스 컨트롤

표 8-8 콤보 박스 컨트롤 통지 메시지

통지 메시지	의미	스타일		
		단순	드롭다운	드롭다운 리스트
CBN_CLOSEUP	리스트 박스가 닫혔다.		●	●
CBN_DBLCLK	사용자가 항목을 더블 클릭했다.		●	
CBN_DROPDOWN	리스트 박스가 열리기 직전이다.		●	●
CBN_EDITCHANGE	= EN_EDITCHANGE	●	●	
CBN_EDITUPDATE	= EN_EDITUPDATE	●	●	
CBN_ERRSPACE	메모리가 부족하다.	●	●	●
CBN_KILLFOCUS	키보드 포커스를 잃었다.	●	●	●
CBN_SELCHANGE	= LBN_SELCHANGE	●	●	●

콤보 박스 컨트롤

CBN_SELENDNCANCEL	= LBN_SELENDNCANCEL		●	●
CBN_SELENDOK	사용자가 항목을 선택했다.	●	●	●
CBN_SETFOCUS	키보드 포커스를 얻었다.	●	●	●

콤보 박스 컨트롤

■ 항목을 추가하거나 삭제하기

```
m_combo.AddString(_T("사과")); // 문자열 항목을 추가한다.  
m_combo.DeleteString(3); // 네 번째 항목을 삭제한다.
```

■ 선택된 항목을 알아내기

```
int nIndex = m_combo.GetCursel();  
if (nIndex != CB_ERR) {  
    CString str;  
    m_combo.GetLBText(nIndex, str);  
}
```

콤보 박스 컨트롤

■ 입력 가능한 텍스트의 길이를 제한하기

```
m_combo.LimitText(10); // 열 글자로 제한한다.
```

■ 각종 편집 작업 - 삭제, 잘라내기, 복사, 붙여넣기

```
m_combo.Clear(); // 현재 선택된 텍스트를 삭제한다.  
m_combo.Cut(); // 텍스트를 잘라내서 클립보드에 저장한다.  
m_combo.Copy(); // 텍스트를 복사해서 클립보드에 저장한다.  
m_combo.Paste(); // 클립보드에 저장된 내용을 붙여넣는다.
```

스크롤바 컨트롤

■ 스크롤바 컨트롤

- 화면의 일정 영역을 스크롤하거나 정해진 범위의 값을 변경하는 용도로 사용

■ 스크롤바의 종류

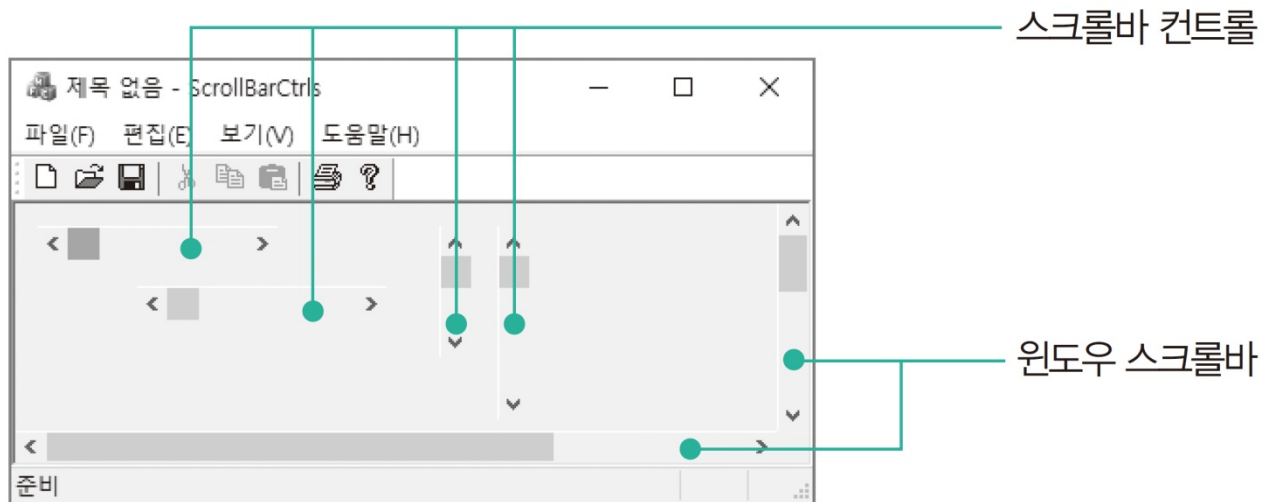
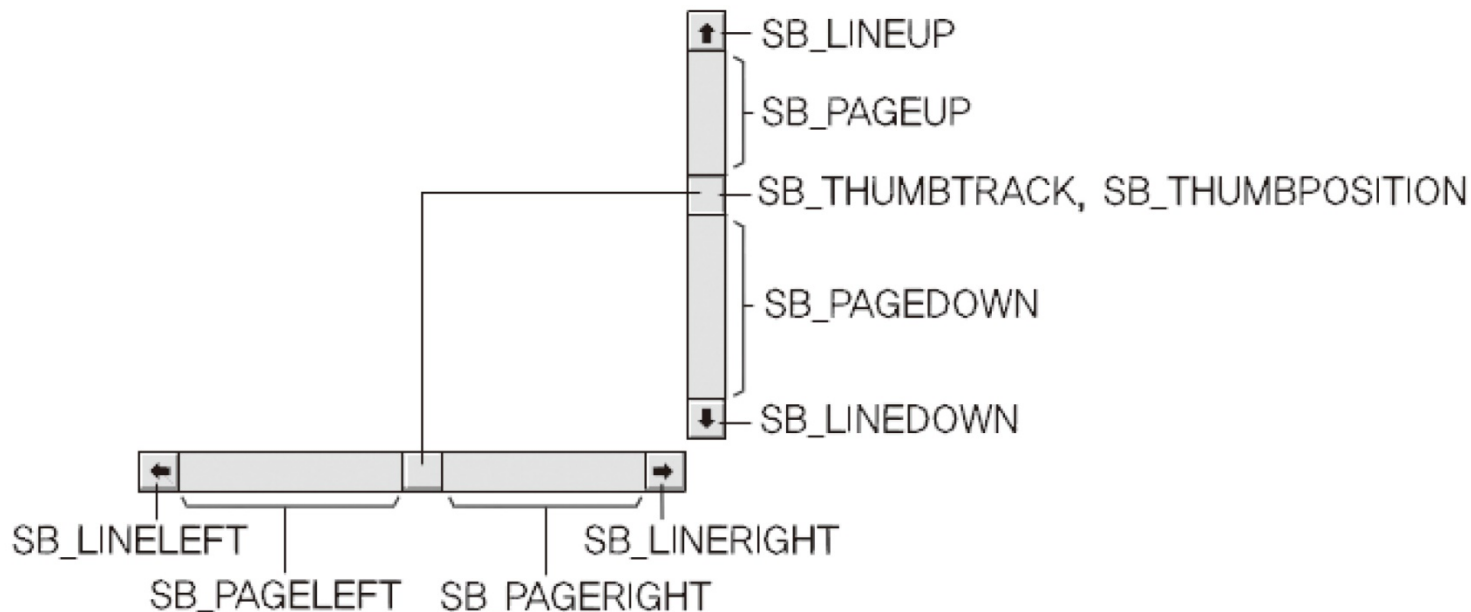


그림 8-35 스크롤바의 종류

스크롤바 컨트롤

■ 스크롤바 코드

- 사용자가 스크롤바를 조작하면 WM_HSCROLL 또는 WM_VSCROLL 메시지 발생
 - 메시지와 더불어 사용자의 구체적인 행위를 알 수 있는 정보가 전달됨 📖 스크롤바 코드(Scroll Bar Code)



스크롤바 컨트롤

표 8-9 스크롤바 코드

스크롤바 코드	응용 프로그램이 해야 할 작업
SB_LINELEFT, SB_LINERIGHT	왼쪽 또는 오른쪽으로 한 줄만큼 스크롤 한다.
SB_PAGELEFT, SB_PAGERIGHT	왼쪽 또는 오른쪽으로 한 페이지만큼 스크롤 한다.
SB_LINEUP, SB_LINEDOWN	위쪽 또는 아래쪽으로 한 줄만큼 스크롤 한다.
SB_PAGEUP, SB_PAGEDOWN	위쪽 또는 아래쪽으로 한 페이지만큼 스크롤 한다.
SB_THUMBTRACK	현재의 스크롤 박스 위치를 고려하여 화면의 출력 내용을 이동시킨다.
SB_THUMBPOSITION	

스크롤바 컨트롤

■ WM_HSCROLL/WM_VSCROLL 메시지 핸들러

```
void OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);
```

- nSBCode
 - 스크롤바 코드(앞의 표 참조)
- nPos
 - 스크롤 박스의 위치(스크롤바 코드가 SB_THUMBTRACK 또는 SB_THUMBPOSITION인 경우에만 의미가 있음)
- pScrollBar
 - CScrollBar 객체(스크롤바 컨트롤에서 발생한 메시지인 경우) 또는 NULL(윈도우 스크롤바에서 발생한 메시지인 경우)

스크롤바 컨트롤

표 8-10 CScrollBar 멤버 함수(일부)

멤버 함수	기능
SetScrollRange()	스크롤 박스 위치의 최솟값과 최댓값을 설정한다.
SetScrollPos()	스크롤 박스의 현재 위치를 설정한다.
GetScrollPos()	스크롤 박스의 현재 위치를 얻는다.

[실습 8-6] 스크롤바 컨트롤 활용하기

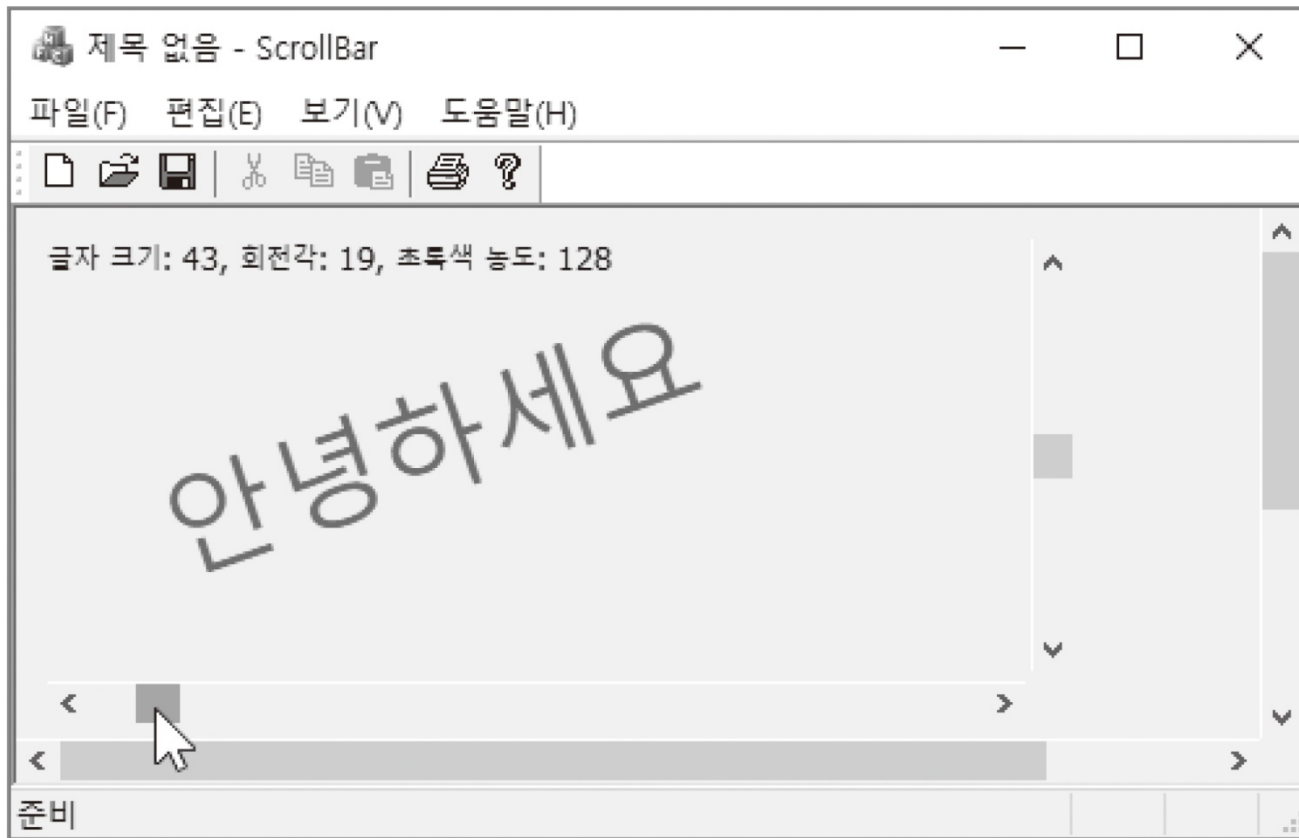


그림 8-37 실행 결과

[실습 8-6] 스크롤바 컨트롤 활용하기



그림 8-38 대화 상자 디자인

번호	변경 사항
①	'ID'를 IDC_STATUS로 변경하고 'Caption'을 삭제한다.
②	'Tabstop'을 True로, 'Align'을 Bottom으로 변경한다.
③	'Tabstop'을 True로, 'Align'을 Right로 변경한다.

[실습 8-6] 스크롤바 컨트롤 활용하기

```
void CScrollBarView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    m_hsb.SetScrollRange(0, 360, FALSE); // 0~360 범위 설정
    m_hsb.SetScrollPos(0);
    m_vsb.SetScrollRange(0, 255, FALSE); // 0~255 범위 설정
    m_vsb.SetScrollPos(128);
}
```

[실습 8-6] 스크롤바 컨트롤 활용하기

```
void CScrollBarView::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar*
pScrollBar)
{
    if (pScrollBar != NULL ) { // 스크롤바 컨트롤에서 발생한 메시지이면... ❶
        if (pScrollBar->GetSafeHwnd() == m_hsb.GetSafeHwnd()) { ❷
            switch (nSBCode) {
                case SB_THUMBTRACK:
                    pScrollBar->SetScrollPos(nPos); ❸
                    break;
            }
            Invalidate(); ❹
        }
    }
    else // 윈도우 스크롤바에서 발생한 메시지이면...
        CFormView::OnHScroll(nSBCode, nPos, pScrollBar);
}
```

[실습 8-6] 스크롤바 컨트롤 활용하기

```
void CScrollBarView::OnVScroll(UINT nSBCode, UINT nPos, CScrollBar*
pScrollBar)
{
    int pos;
    int delta = 0;
    if (pScrollBar != NULL) { // 스크롤바 컨트롤에서 발생한 메시지이면...
        if (pScrollBar->GetSafeHwnd() == m_vsb.GetSafeHwnd()) {
            switch (nSBCode) {
                case SB_THUMBTRACK:
                    pScrollBar->SetScrollPos(nPos);
                    break;
                case SB_LINEUP:
                    delta = -4;
                    break;
                case SB_LINEDOWN:
                    delta = 4;
                    break;
                case SB_PAGEUP:
                    delta = -20;
                    break;
```

[실습 8-6] 스크롤바 컨트롤 활용하기

```
case SB_PAGEDOWN:
    delta = 20;
    break;
}
if (delta != 0) {
    pos = pScrollBar->GetScrollPos();
    pScrollBar->SetScrollPos(pos + delta);
}
Invalidate();
}
}
else // 윈도우 스크롤바에서 발생한 메시지이면...
    CFormView::OnVScroll(nSBCode, nPos, pScrollBar);
}
```

[실습 8-6] 스크롤바 컨트롤 활용하기

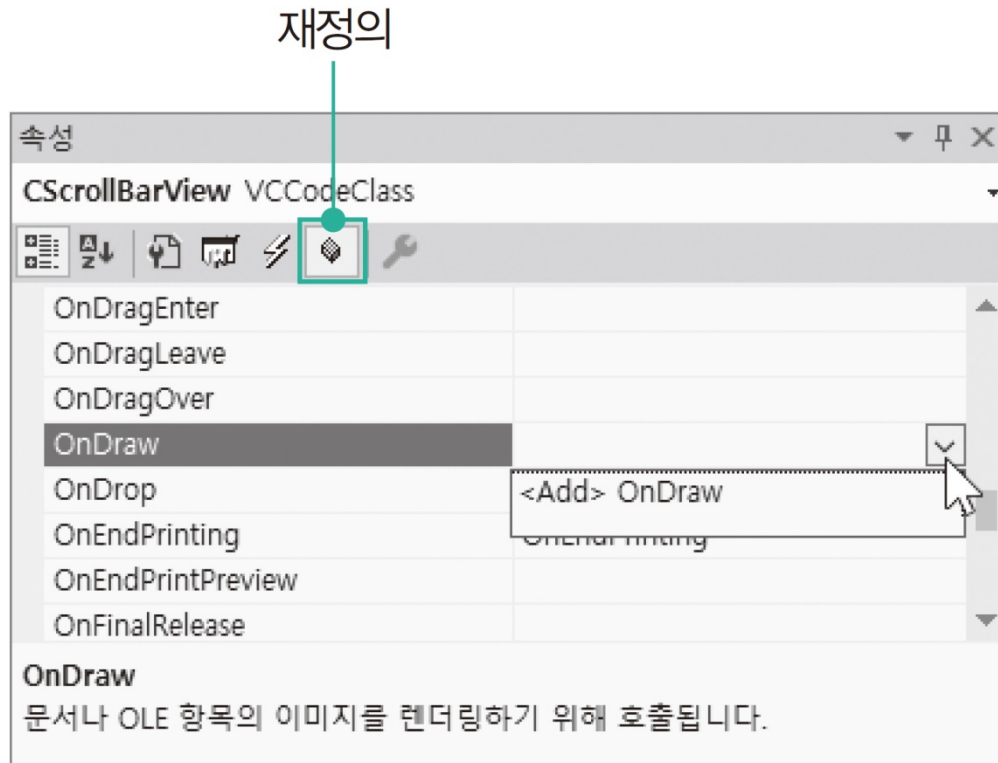


그림 8-40 OnDraw() 함수 추가

[실습 8-6] 스크롤바 컨트롤 활용하기

```
void CScrollBarView::OnDraw(CDC* pDC)
{
    // 스크롤바의 현재 위치를 얻는다.
    int hpos = m_hsb.GetScrollPos();
    int vpos = m_vsb.GetScrollPos();
    Font font;
    LOGFONT lf = { 0 }; // 구조체를 0으로 초기화한다.
    lf.lfHeight = vpos / 3 + 1; // 폰트 크기를 설정한다. vpos==0이면 크기는 1이다.
    lf.lfEscapement = 10 * hpos; // 폰트 회전각을 설정한다.
    font.CreateFontIndirect(&lf); // 폰트를 생성한다.
    pDC->SelectObject(&font);
    pDC->SetTextColor(RGB(0, vpos, 0)); // 폰트 색상을 설정한다.
    pDC->TextOut(50, 100, CString(_T("안녕하세요")));

    CString str;
    str.Format(_T("글자 크기: %d, 회전각: %d, 초록색 농도: %d"),
        lf.lfHeight, hpos, vpos);
    m_status.SetWindowText(str); // 스테틱 텍스트 컨트롤에 출력한다.
}
```

고급 컨트롤 기법

■ 서브클래싱

- 기존 윈도우나 컨트롤의 형태 또는 동작 변경

■ 메시지 반사

- 스스로 통지 메시지를 처리하는 독립적인 컨트롤 클래스를 제작

서브클래싱

■ 서브클래싱

- 윈도우(정확하게는 윈도우 프로시저)로 가는 메시지를 중간에서 붙잡아 처리하는 기법
- 메시지의 종류에 따라, 처리가 끝난 후 원래의 윈도우 프로시저에 전달할 수도 있고 전달하지 않을 수도 있음



그림 8-41 서브클래싱의 원리

서브클래싱

■ 서브클래싱 구현 순서

- 기존의 클래스를 기반으로 새로운 윈도우나 컨트롤 클래스를 생성
- 메시지 핸들러를 추가해서 기능을 재정의
- `CWnd::SubclassWindow()` 또는 `CWnd::SubclassDlgItem()` 함수를 호출

서브클래싱

```
void CMyTestView::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_EDIT1, m_edit1); // 서브클래싱!
}
```

[실습 8-7] 컨트롤 서브클래싱 구현하기

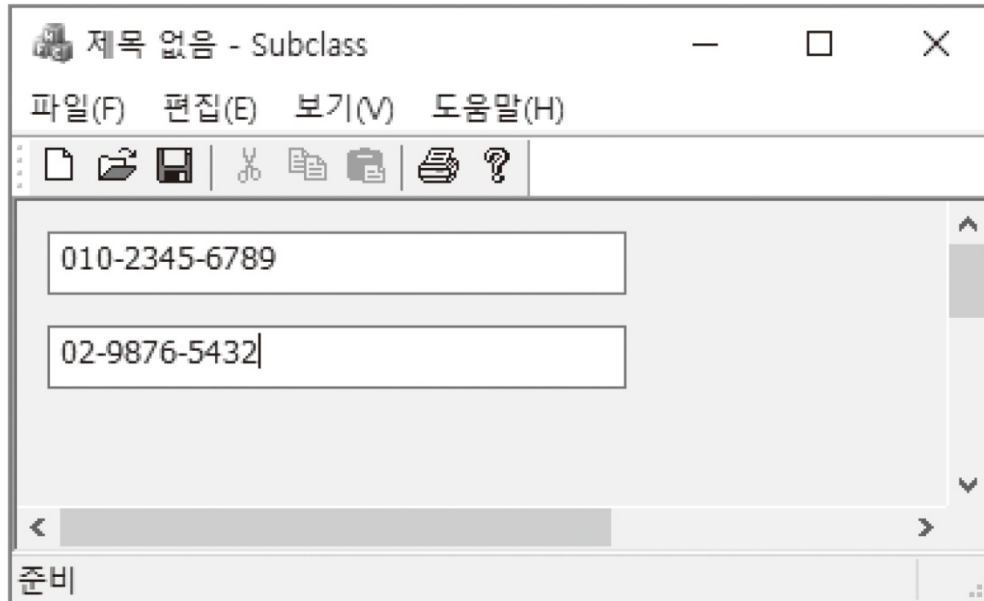


그림 8-42 실행 결과

[실습 8-7] 컨트롤 서브클래싱 구현하기

클래스 마법사

클래스 마법사 시작

프로젝트(P): Subclass 클래스 이름(N): CAboutDlg 클래스 추가(C)...

기본 클래스: CDialogEx

리소스: IDD_ABOUTBOX 클래스 구현(L): Subclass.cpp

MFC 클래스(C)...

명령 메시지 가상 함수 멤버 변수 메서드

검색 명령

개체 ID(B): AFX_ID_PREVIEW_CLOSE AFX_ID_PREVIEW_NEXT AFX_ID_PREVIEW_NUMPAGE AFX_ID_PREVIEW_PREV AFX_ID_PREVIEW_PRINT AFX_ID_PREVIEW_ZOOMIN AFX_ID_PREVIEW_ZOOMOUT AFX_IDB_CHECKLISTBOX_95

메시지(S): COMMAND UPDATE_COMMAND_UI

처리기 추가(A)...

처리기 삭제(D)

코드 편집(E)

멤버 함수(M):

함수 이름	명령 ID	메시지
-------	-------	-----

설명:

확인 취소 적용

[실습 8-7] 컨트롤 서브클래싱 구현하기

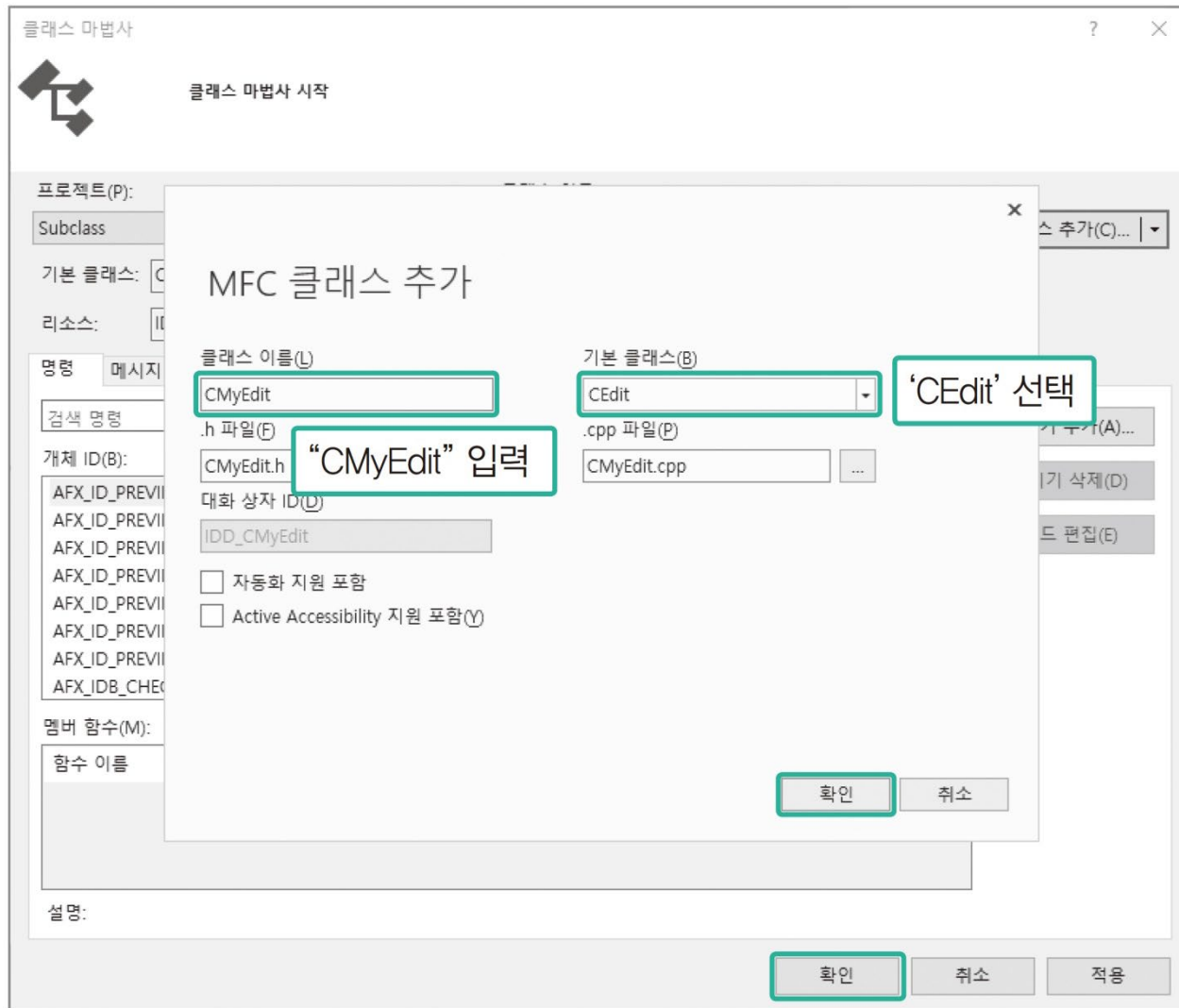


그림 8-43 CMyEdit 클래스 생성

[실습 8-7] 컨트롤 서브클래싱 구현하기

```
void CMyEdit::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    if ((nChar >= _T('0') && nChar <= _T('9'))
        || nChar == _T('-') || nChar == _T('\b'))
    {
        CEdit::OnChar(nChar, nRepCnt, nFlags);
    }
}
```

[실습 8-7] 컨트롤 서브클래싱 구현하기

The diagram shows a light gray rectangular container representing a dialog box. Inside this container, there are two white rectangular input fields stacked vertically. To the left of the top input field is a green circle containing the number '1'. To the left of the bottom input field is a green circle containing the number '2'. Both input fields contain the text '샘플 입력란' (Sample Input Field) in black Korean text.

그림 8-44 대화 상자 디자인

[실습 8-7] 컨트롤 서브클래싱 구현하기

제어 변수 추가
일반 설정

컨트롤
기타

컨트롤 ID(I)
IDC_EDIT1

컨트롤 형식(Y)
EDIT

범주(T)
컨트롤
'IDC_EDIT1'

이름(N)
m_edit1
“m_edit1” 입력

액세스(A)
public

변수 형식(V)
CMyEdit
“CMyEdit” 변경

주석(M)

이전 다음 마침 취소

그림 8-45 컨트롤 변수 생성

[실습 8-7] 컨트롤 서브클래싱 구현하기

```
#pragma once
```

```
#include "CMyEdit.h"
```

```
class CSubclassView : public CFormView  
{
```

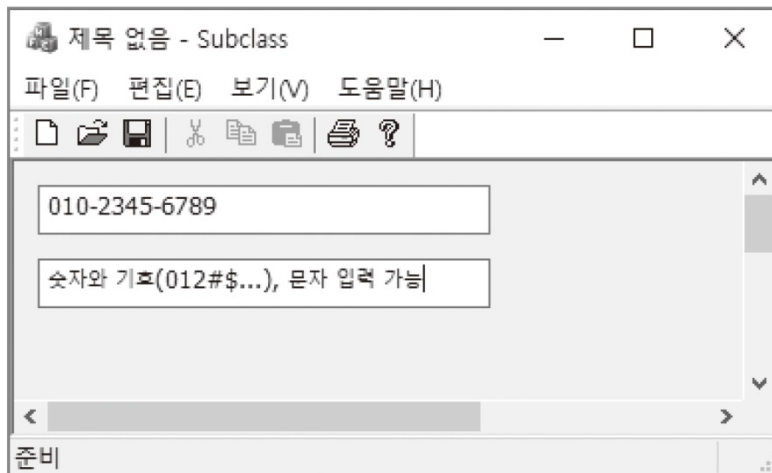


그림 8-46 실행 결과

[실습 8-7] 컨트롤 서브클래싱 구현하기

```
class CSubclassView : public CFormView
{
...
// 특성입니다.
public:
    CSubclassDoc* GetDocument() const;
    CMyEdit m_edit2;
...
}
```

[실습 8-7] 컨트롤 서브클래싱 구현하기

```
void CSubclassView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();
    m_edit2.SubclassDlgItem(IDC_EDIT2, this);
}
```

메시지 반사

■ 메시지 반사

- 부모 윈도우가 처리하지 않는 통지 메시지를 컨트롤 자신이 처리
- 부모 윈도우가 통지 메시지를 처리하지 않는 경우에만 컨트롤이 자신의 통지 메시지 처리 가능

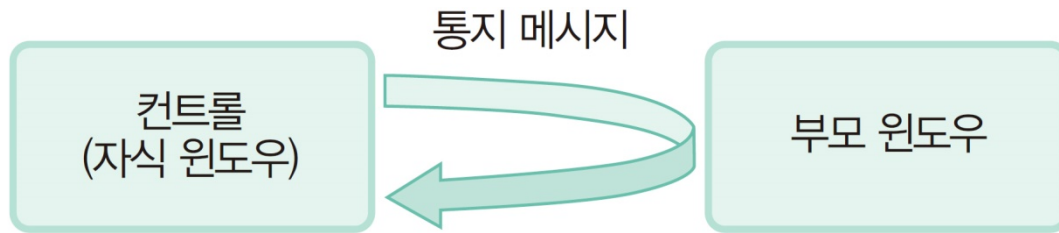


그림 8-47 메시지 반사

메시지 반사

■ 메시지 반사 구현 순서

- 기존 컨트롤 클래스를 상속받아 새로운 클래스를 만들고 자신의 통지 메시지를 처리하는 함수를 작성
- 새로 만든 클래스를 이용하여 컨트롤 생성

[실습 8-8] 메시지 반사 구현하기

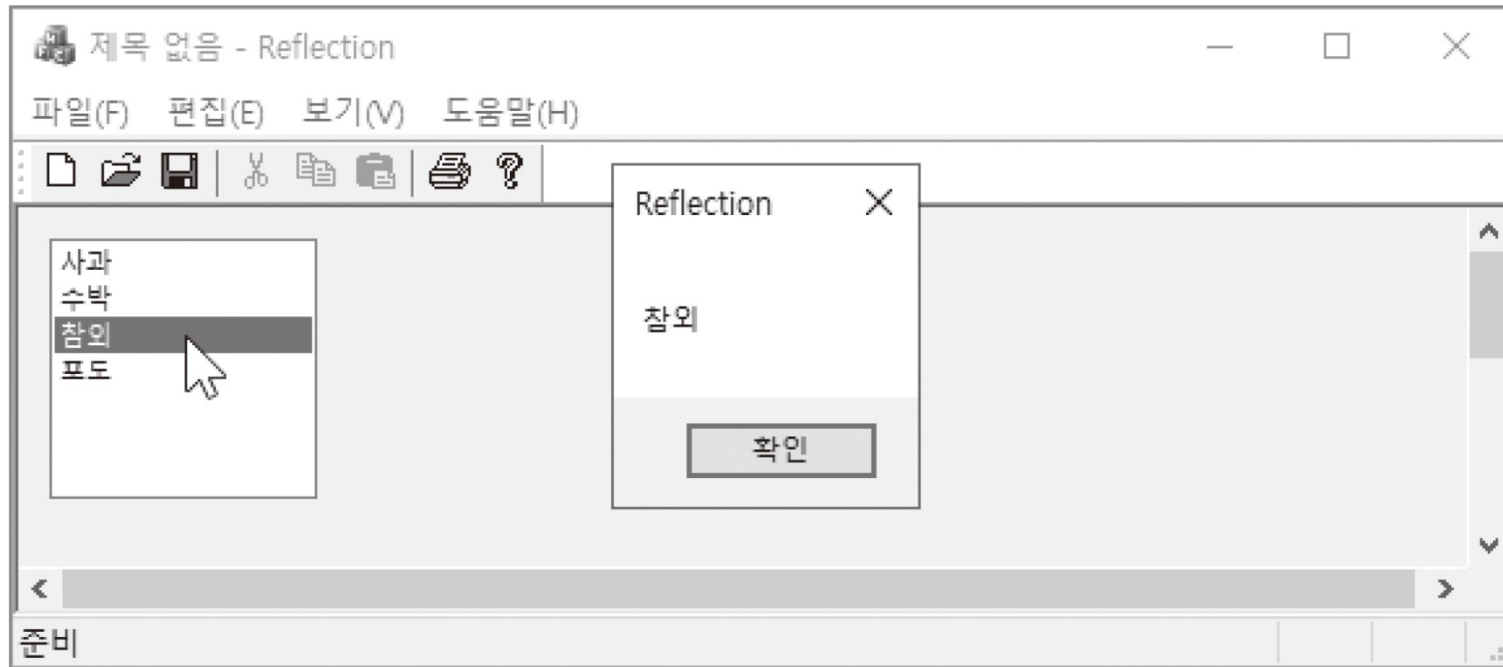
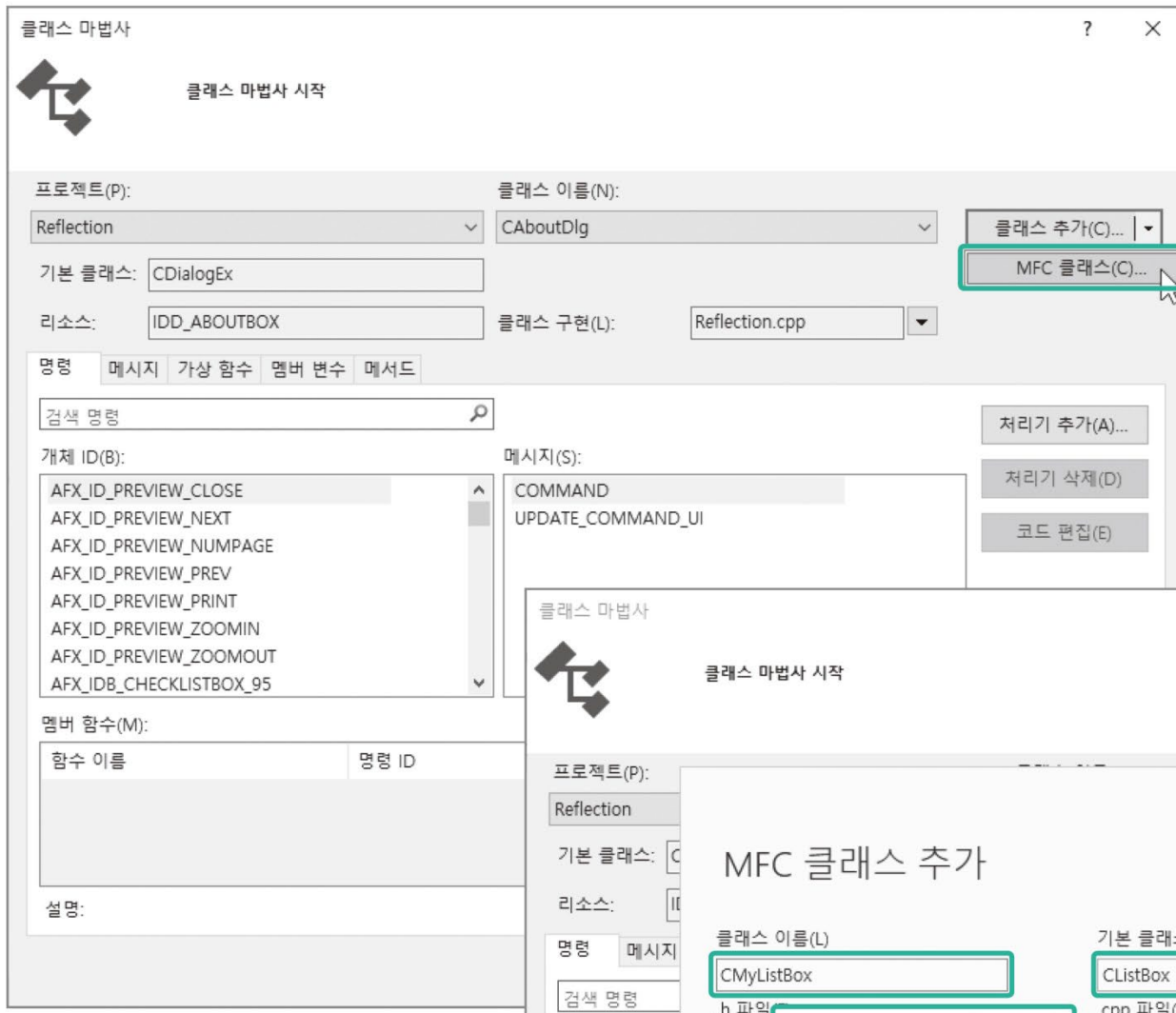


그림 8-48 실행 결과

[실습 8-8] 메시지 반사 구현하기





[실습 8-8] 메시지 반사 구현하기

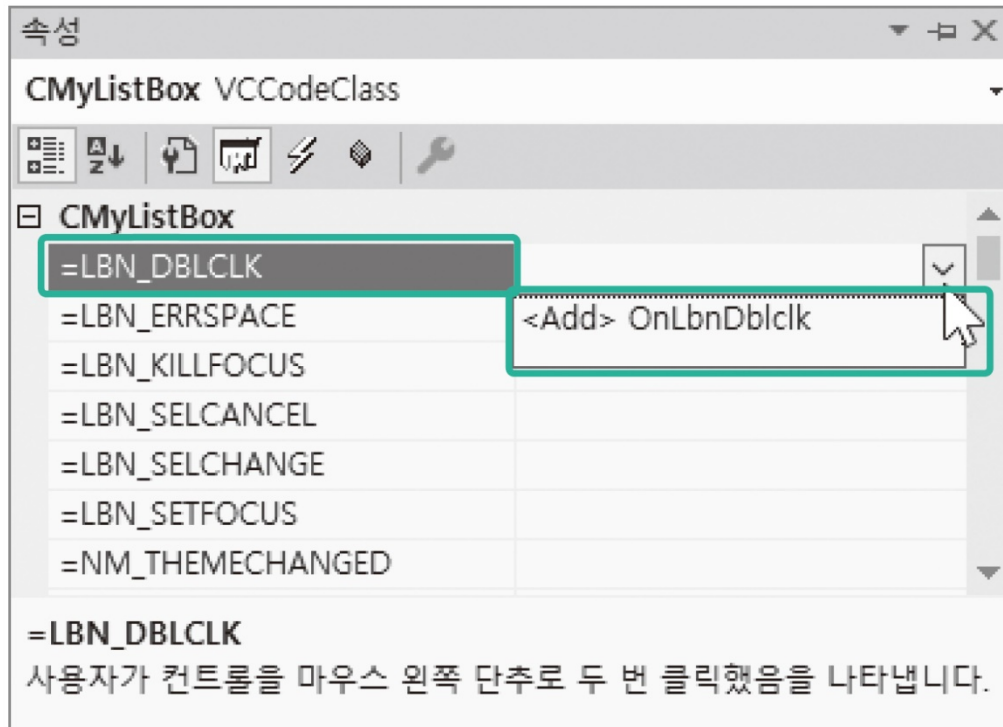


그림 8-50 통지 메시지 핸들러 추가

[실습 8-8] 메시지 반사 구현하기

```
void CMyListBox::OnLbnDblclk()  
{  
    CString str;  
    int nIndex = GetCurSel();  
    GetText(nIndex, str);  
    MessageBox(str);  
}
```

[실습 8-8] 메시지 반사 구현하기



그림 8-51 대화 상자 디자인

[실습 8-8] 메시지 반사 구현하기

제어 변수 추가
일반 설정

컨트롤
기타

컨트롤 ID(I) IDC_LIST1

컨트롤 형식(V) LISTBOX

범주(T) 컨트롤

이름(N) m_list

액세스(A) public

변수 형식(V) CMyListBox

주석(M)

“m_list” 입력

“CMyListBox” 변경

이전 다음 마침 취소

그림 8-52 컨트롤 변수 생성

[실습 8-8] 메시지 반사 구현하기

```
#pragma once
```

```
#include "CMyListBox.h"
```

```
class CReflectionView : public CFormView  
{
```

[실습 8-8] 메시지 반사 구현하기

```
void CReflectionView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    m_list.AddString(_T("사과"));
    m_list.AddString(_T("포도"));
    m_list.AddString(_T("참외"));
    m_list.AddString(_T("수박"));
}
```