

Window Programming

Visual C++ MFC Programming

Lecture 12

김예진

Dept. of Game Software

Notices

- 03/20: HW 1 (Due: 03/26) → Avg: 8.85
- 04/05: HW 2 (Due: 04/13) → Avg: 6.90
- 04/19: Midterm → Avg: 3.70
 - 5문제, ~75 min., 강의록 1~8
- 05/10: HW 3 (Due: 05/17)

Plan: 컨트롤 (Controls)

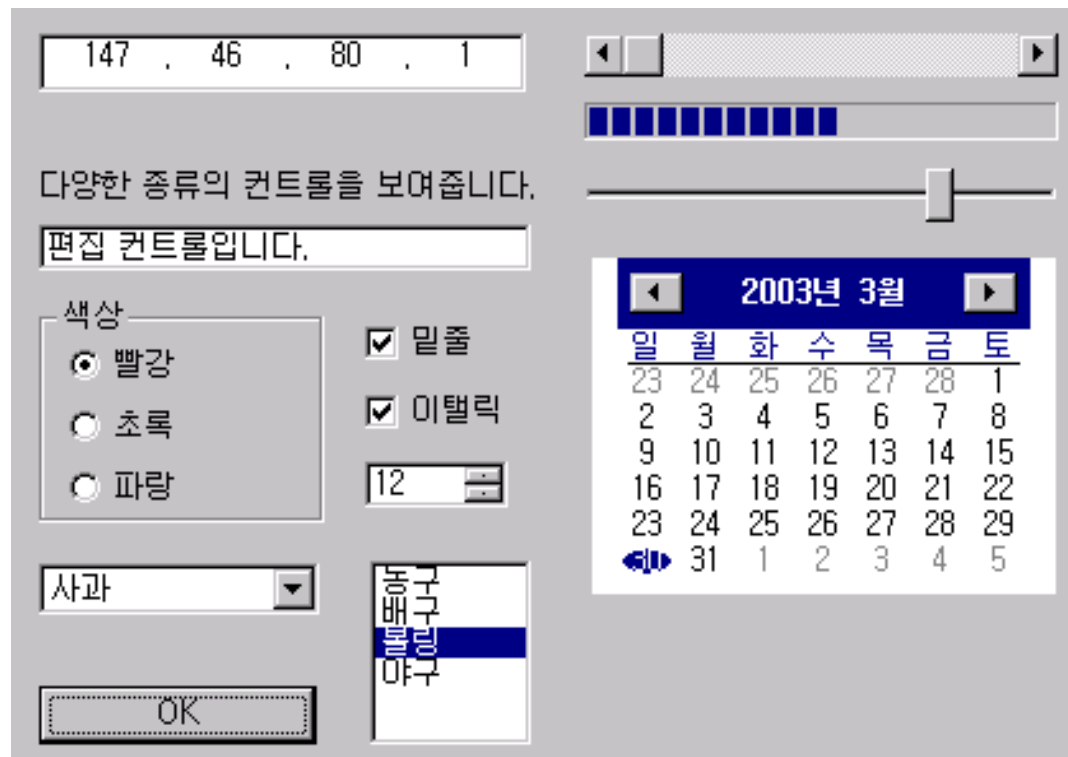
- 컨트롤 (Controls) 활용
 - 연습: Push 버튼 만들기
 - 연습: Check 박스 만들기
 - 연습: Radio 버튼 사용
 - 연습: 대화상자를 통한 다양한 버튼 생성
- 더 많은 컨트롤 (More Controls)
 - 연습: 정적 컨트롤 붙이기
 - 연습: 편집 컨트롤 붙이기
 - 연습: 리스트 박스 붙이기
 - 연습: 편집 컨트롤과 스크롤 바 사용
 - 연습: 스크롤 바로 원의 위치 조정

컨트롤 (Controls)



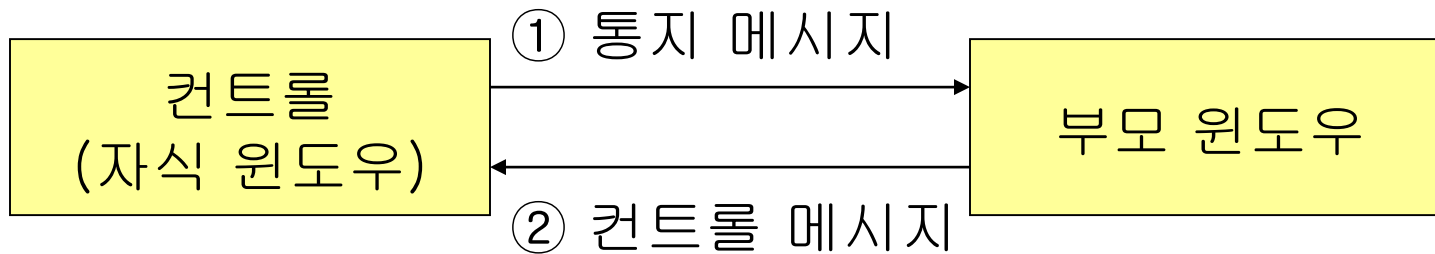
개요 (1/4)

- 컨트롤
 - 표준화된 형태와 특성을 가진 윈도우
 - 사용자에게 입력을 받거나 정보를 보여줌



개요 (2/4)

- 컨트롤과 부모 윈도우



- 통지 메시지 (Notification Message)
 - 컨트롤의 상태가 변화되었음을 알림
 - 메모리 부족 등으로 인한 오류를 알림
- 컨트롤 메시지 (Control Message)
 - 컨트롤의 상태를 알아내거나 변경

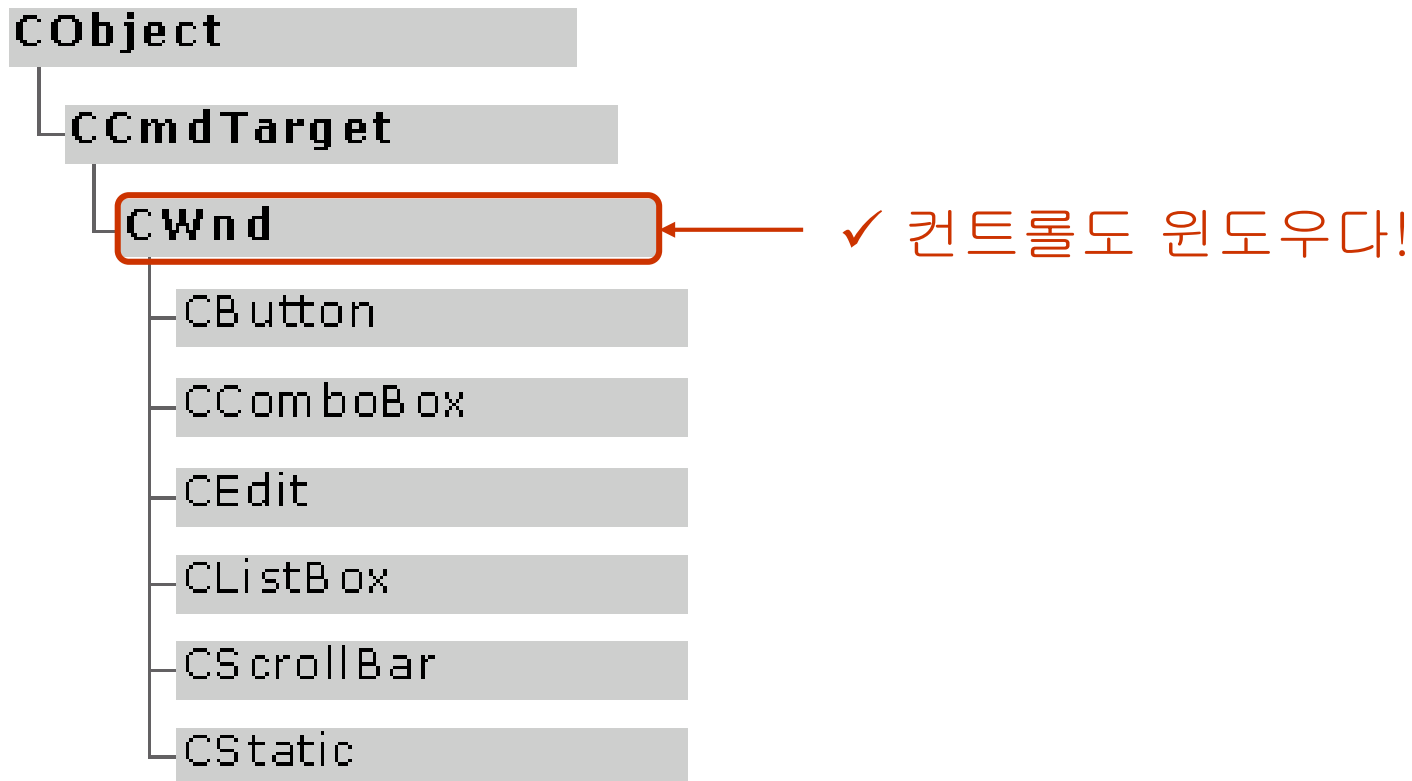
개요 (3/4)

- MFC 컨트롤 클래스

컨트롤	MFC 클래스	컨트롤 버튼
버튼 컨트롤	CButton	   
정적 컨트롤	CStatic	 
편집 컨트롤	CEdit	
리스트 박스 컨트롤	CListBox	
콤보 박스 컨트롤	CComboBox	
스크롤 바 컨트롤	CScrollBar	 

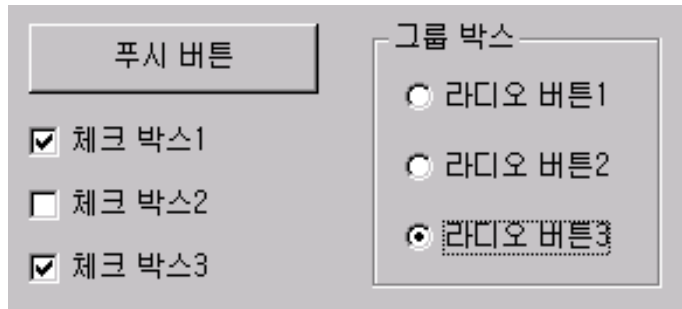
개요 (4/4)

- MFC 클래스 계층도

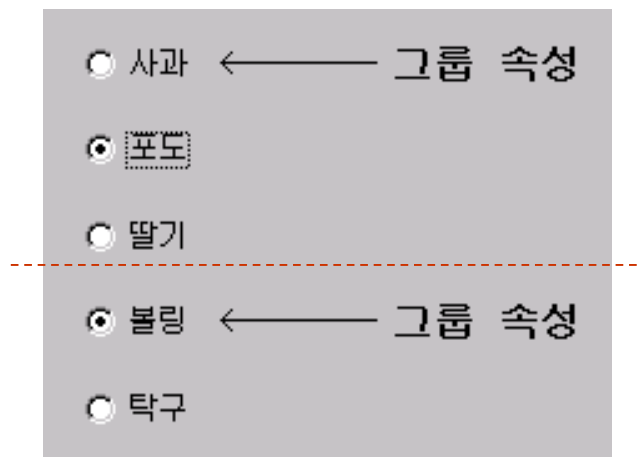


버튼 컨트롤

- Button Control 종류



- Radio Button 그룹



버튼 컨트롤 생성 방법

- 컨트롤 생성 방법

① 일반 윈도우에서 코드로 직접 생성

② 대화상자 리소스를 통해 생성

버튼 컨트롤: 버튼 붙이기 (1/2)

- 일반 윈도우에서 코드로 직접 생성:
 1. 버튼 객체를 멤버 변수로 추가
 - Ex) CChildView의 header file에 CButton 객체 추가

```
// ChildView.h의 CChildView의 내부에...  
CButton m_button; // C++ 객체 추가
```

2. CButton의 Create(..) 멤버 함수를 호출하여 생성
 - Ex) CChildView::OnCreate() 내부에 추가

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

버튼 컨트롤: 버튼 붙이기 (2/2)

- CButton::Create() 함수

```
BOOL CButton::Create (Caption, Style, rect, pParentWindow, ID);
```

- Caption: 캡션 문자열
- Style: 일반 윈도우 스타일 + 버튼 컨트롤 스타일
- rect: 컨트롤 크기와 위치
- pParentWnd: 부모 윈도우
- UINT nID: 컨트롤 ID (ex, 101, 102...)

버튼 컨트롤의 Style

- 버튼 컨트롤 스타일 - 일부

버튼 컨트롤 스타일	의미
BS_PUSHBUTTON	푸시 버튼
BS_CHECKBOX	체크 박스
BS_AUTOCHECKBOX	자동 체크 박스: 클릭하면 자동으로 체크 표시가 On/Off
BS_3STATE	3상태 체크 박스
BS_AUTO3STATE	자동 3상태 체크 박스: 클릭하면 자동으로 체크 표시가 On/Grayed/Off
BS_RADIOBUTTON	라디오 버튼
BS_AUTORADIOBUTTON	자동 라디오 버튼: 클릭하면 자동으로 선택과 선택 해제가 이뤄짐과 선택 해제가 이루어짐
BS_GROUPBOX	그룹 박스

버튼 컨트롤의 동작 (1/2)

- 버튼이 눌렸다면? → 버튼이 눌렸다는 메시지 발생



- 통지 메시지 처리하기 ⇨ 메시지 핸들러 작성
 1. 메시지 맵에 메시지 추가
 2. 메시지 핸들러 함수구현

버튼 컨트롤의 동작 (2/2)

1. Message Map에 추가

`ON_BN_CLICKED`(ID, 함수 이름)

Ex) `BEGIN_MESSAGE_MAP(CChildView, CWnd)`

...

`ON_BN_CLICKED`(101, `OnButtonClicked`) // 메시지맵

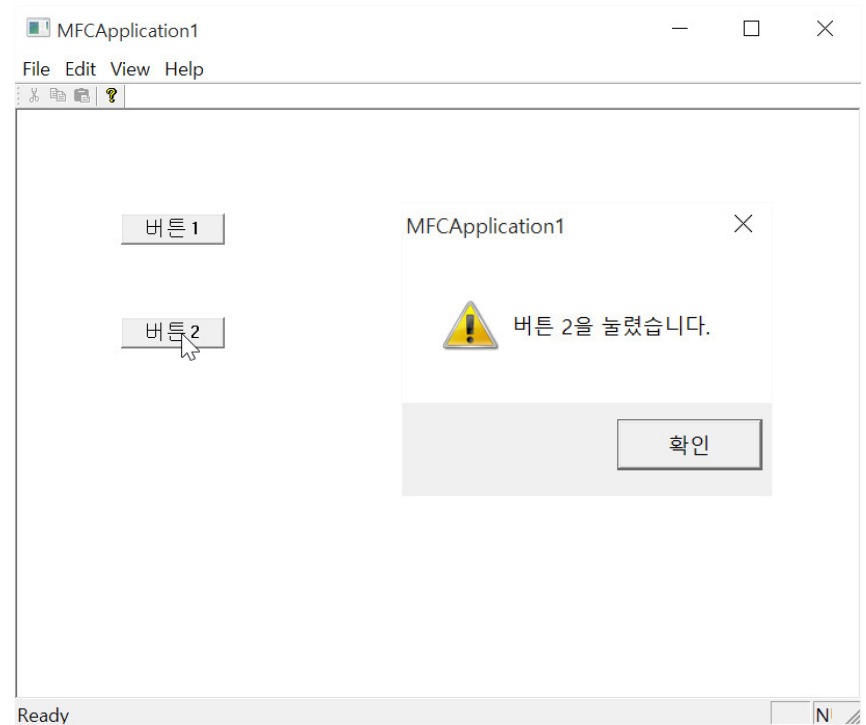
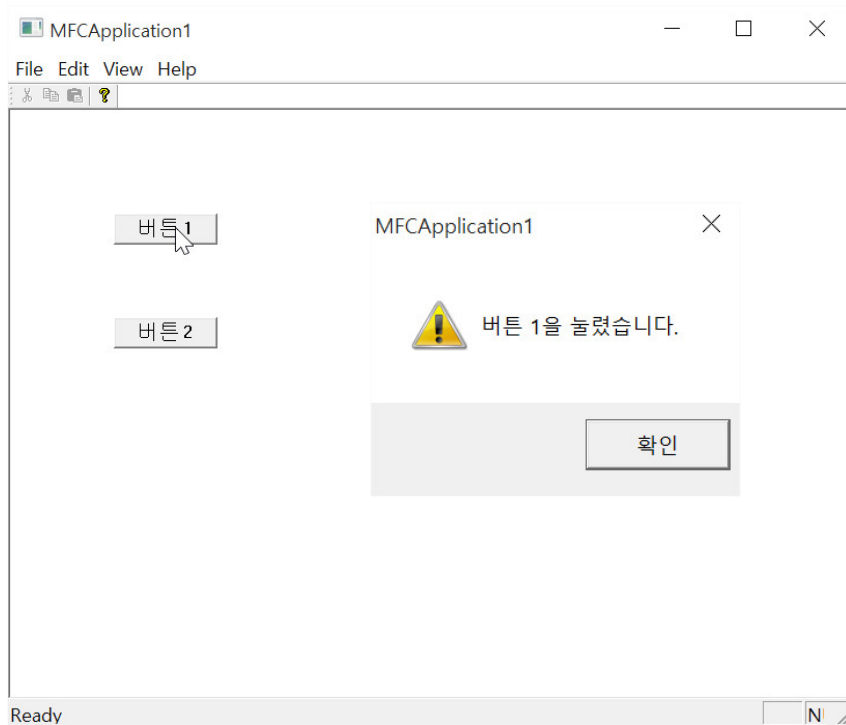
`END_MESSAGE_MAP()`

2. 핸들러 함수 추가

```
void CChildView::OnButtonClicked( ) // 메시지 핸들러
{
    AfxMessageBox(_T("버튼을 눌렀습니다."));
}
```

연습: Push 버튼 만들기

- 다음과 같은 Push 버튼 2개를 만든 후 어떤 버튼이 눌러졌는지 화면에 표시함

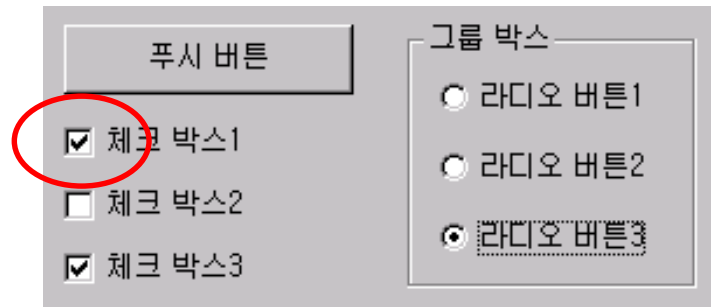


버튼 컨트롤의 정보 변경 (1/2)

- 버튼의 상태를 변경하고 (알고) 싶다면?
→ 컨트롤 메시지 활용



- 컨트롤(버튼)의 현재 상태를 변화 시킴
- 컨트롤(버튼)에게 현재의 상태를 물어봄



버튼 컨트롤의 정보 변경 (2/2)

- 컨트롤 메시지 보내기 ⇨ **멤버 함수 호출**

1. 버튼의 체크상태 변환

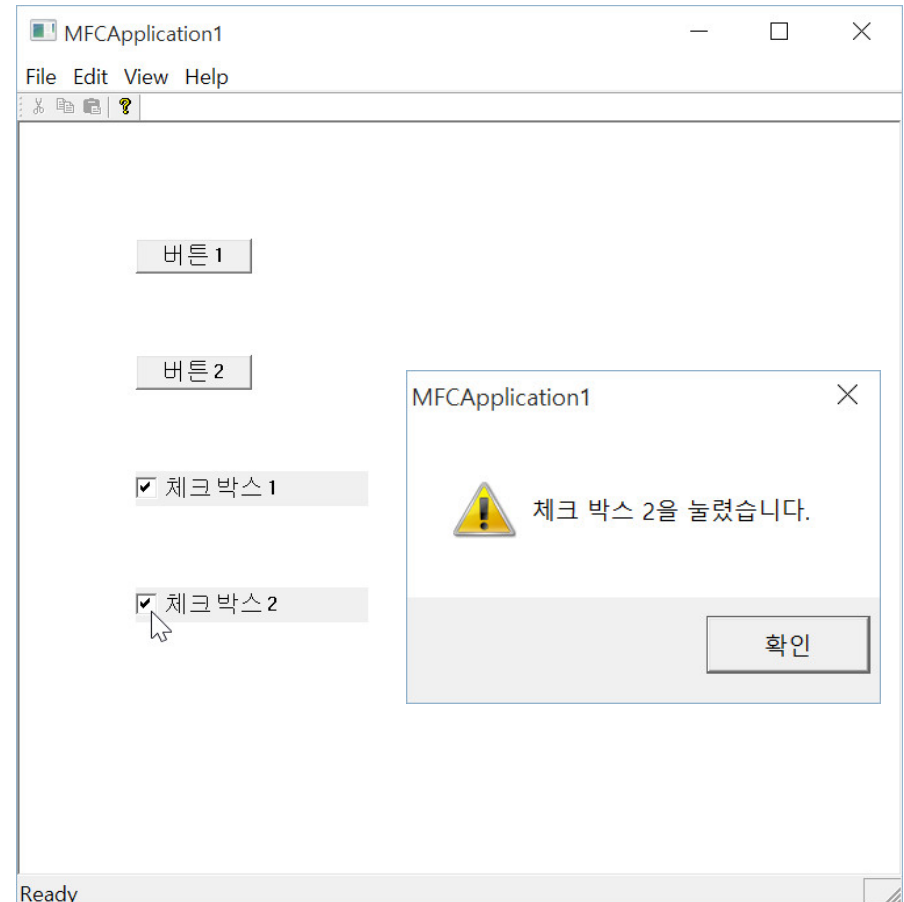
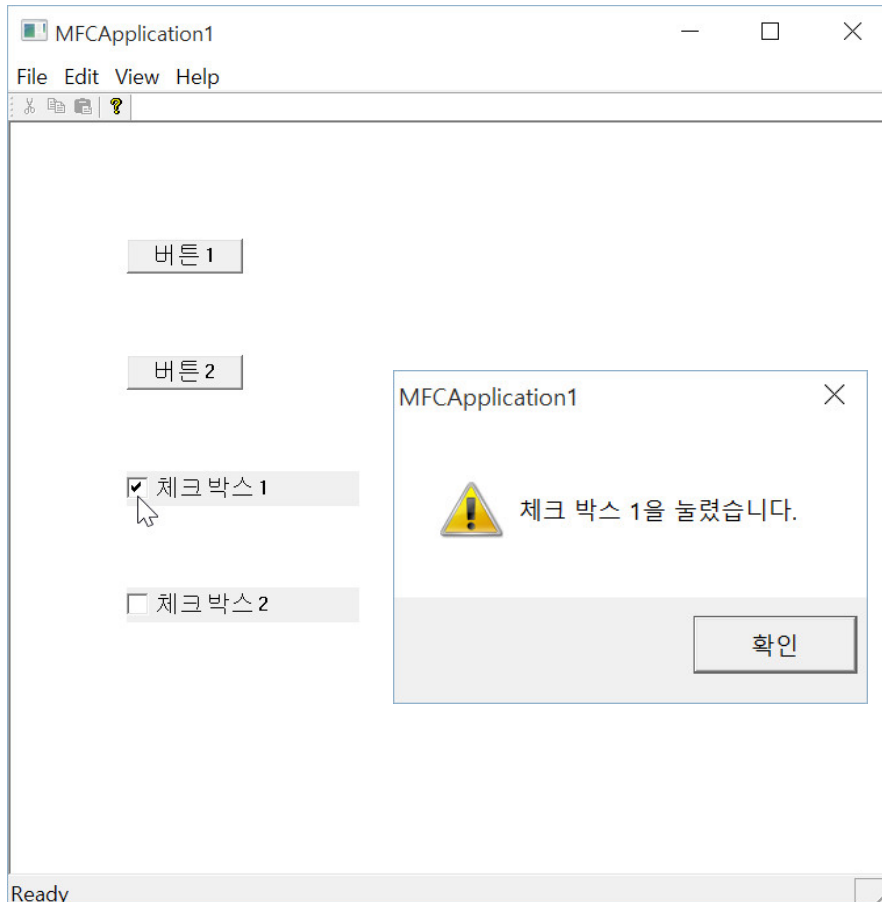
```
m_button.SetCheck(BST_CHECKED);  
m_button.SetCheck(BST_UNCHECKED);
```

2. 버튼의 체크상태 확인

```
if (m_button.GetCheck() == BST_CHECKED)  
    AfxMessageBox(_T("버튼이 체크되었습니다"));
```

연습: Check 박스 만들기

- 다음과 같은 Check 박스 2개를 만든 후 어떤 박스를 체크했는지 화면에 표시함



버튼 컨트롤 생성 방법

- 컨트롤 생성 방법

- ① 일반 윈도우에서 코드로 직접 생성

- ② 대화상자 리소스를 통해 생성

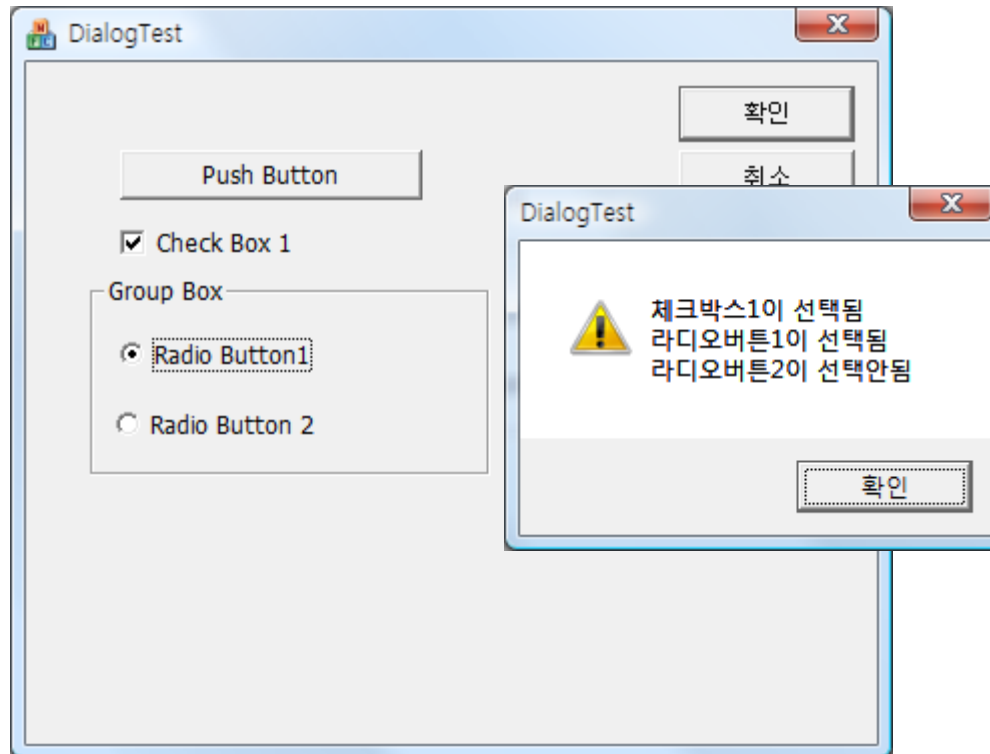
- 컨트롤의 위치와 크기를 시각적으로 디자인 가능

버튼 컨트롤 생성 방법

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

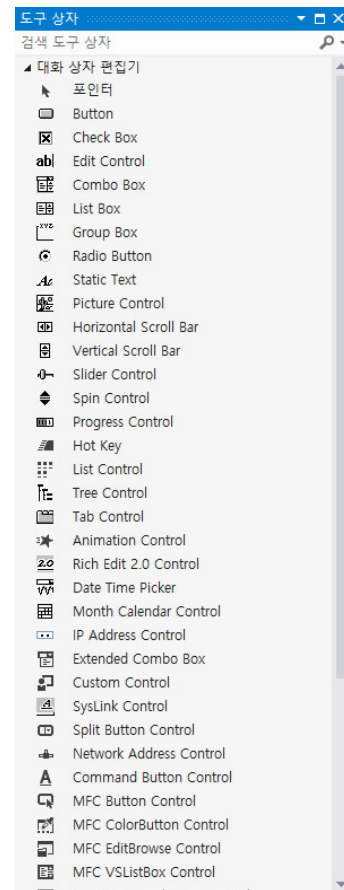
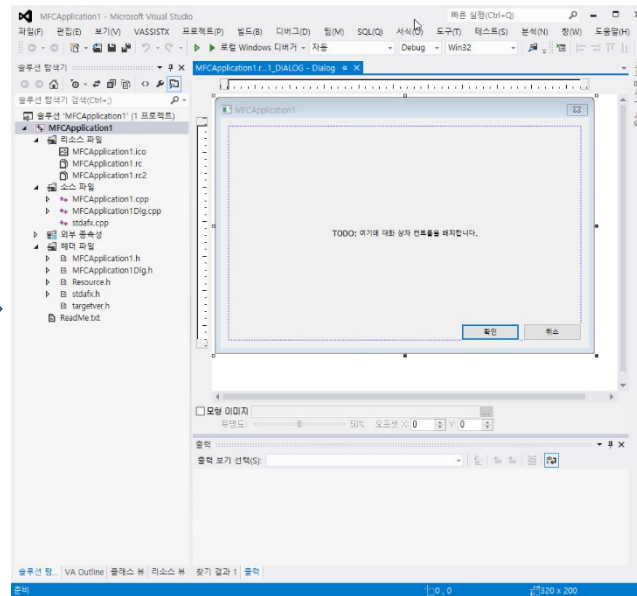
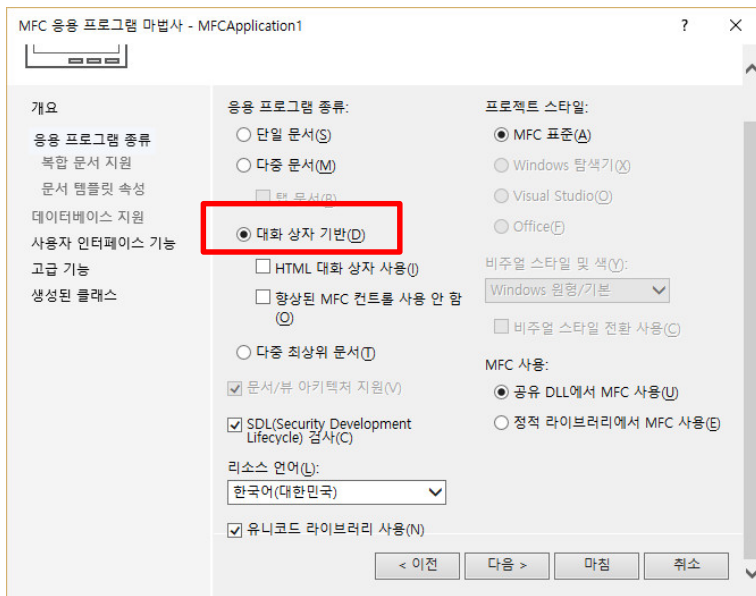
연습: 대화상자를 통한 다양한 버튼 생성

- 다음과 같이 대화상자에 다양한 버튼을 생성
- Push Button을 누르면 선택된 버튼 상태를 표시



연습: 대화상자를 통한 다양한 버튼 생성

- 대화 상자 기반을 선택하여 프로젝트 생성
- 대화 상자의 편집기를 사용하여 컨트롤 추가



버튼 컨트롤 변수 추가 (1/2)

- 해당 컨트롤에 마우스 오른 버튼 → 변수추가선택

멤버 변수 추가 마법사 - FormViewTest

멤버 변수 추가 마법사 시작
클래스, 구조체 또는 공용 구조체에 멤버 변수를 추가합니다.

액세스(A):
public

변수 형식(V):
CButton

변수 이름(N):

컨트롤 변수(O) ☒

컨트롤 ID(I):
IDC_BUTTON1

범주(T):
Control

컨트롤 형식(Y):
BUTTON

최소값(U):

최대 문자 수(X):

최대값(E):

.h 파일(F):

.cpp 파일(P):

주석(// 표시 필요 없음)(M):

마침 취소 도움말

ActiveX 컨트롤에 기반한 변수인 경우 마법사에서 래퍼 클래스를 만듭니다. 클래스 선언을 삽입할 헤더 파일을 입력합니다.

변수 이름 지정

버튼 컨트롤 변수 추가 (2/2)

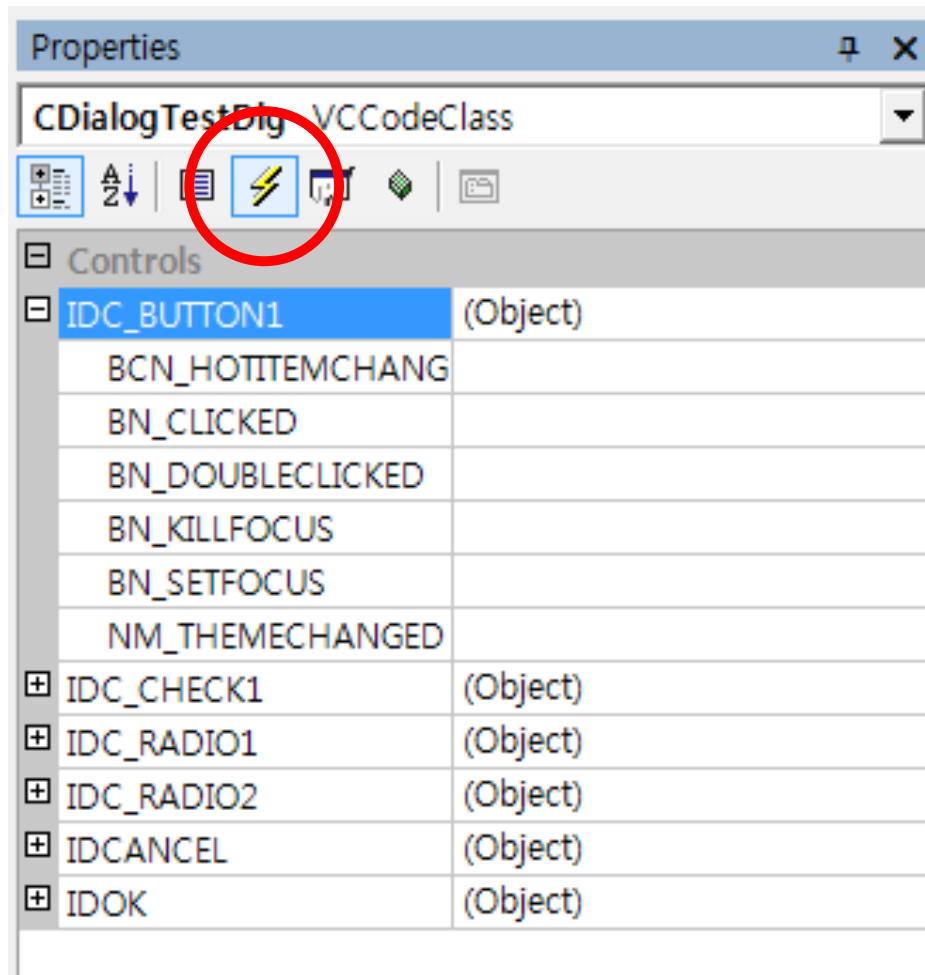
- 컨트롤 변수 생성 (cont'd)

```
// 헤더 파일
class CTestDlg::public CDiglog
{
    ...
    CButton m_button;
}

// 구현 파일
void CTestDlg::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CExButtonView)
    DDX_Control(pDX, IDC_BUTTON1, m_button);
    //}}AFX_DATA_MAP
}
```

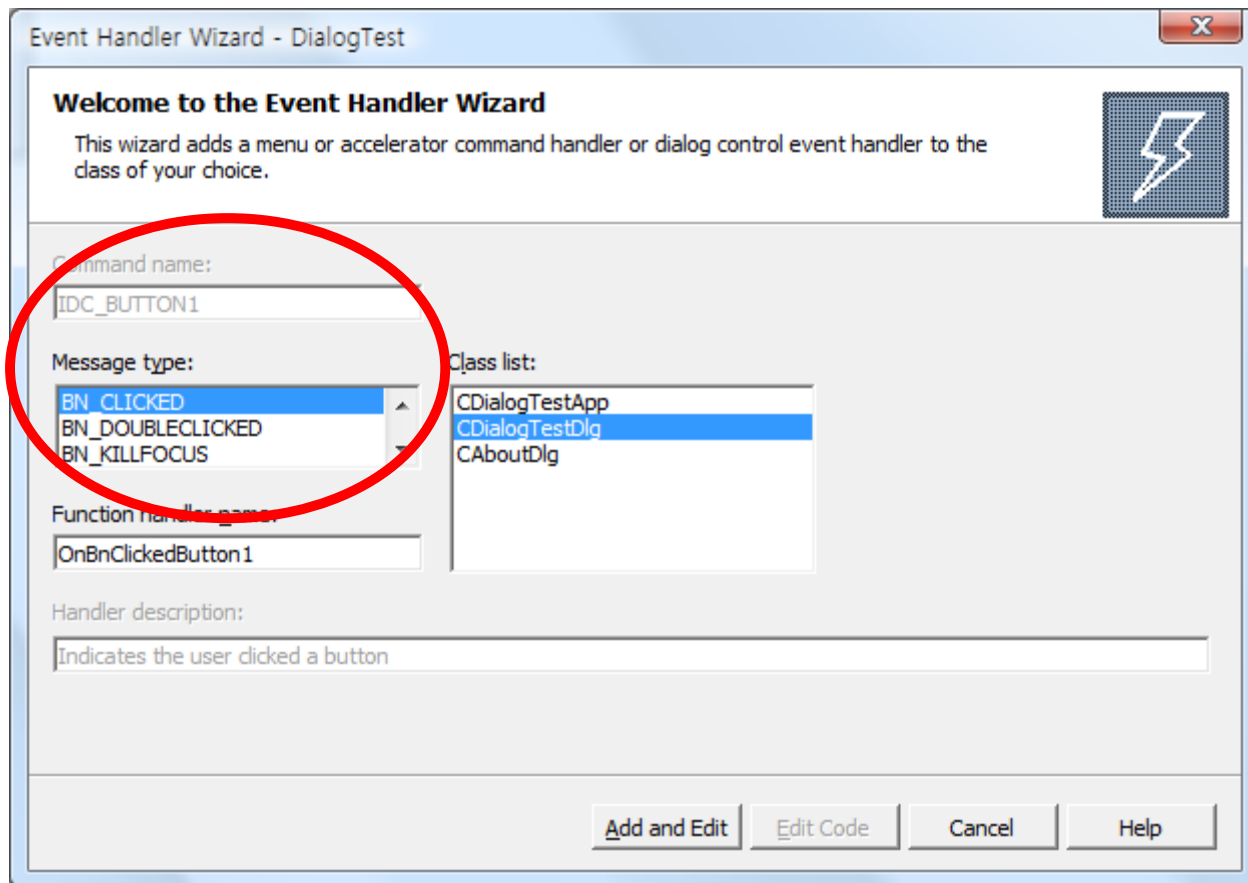
버튼 컨트롤 메시지 핸들러 추가 방법 1

- 대화창의 속성창을 통해 이벤트 핸들러 추가



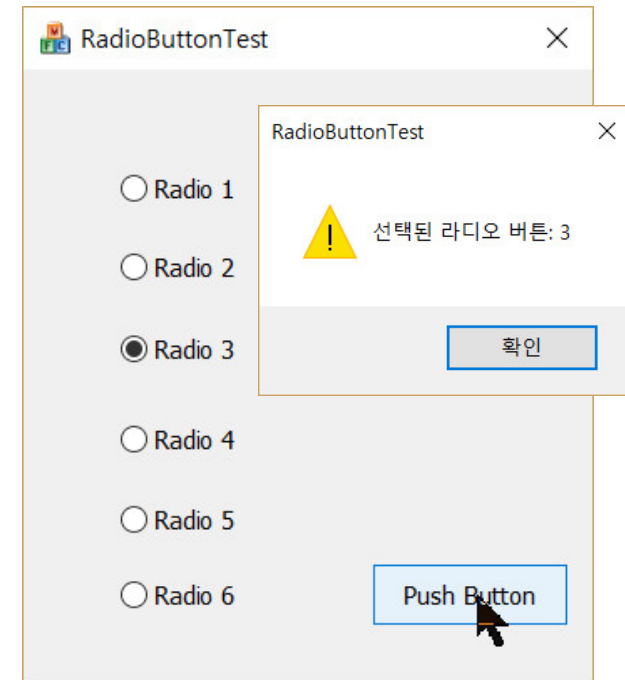
버튼 컨트롤 메시지 핸들러 추가 방법 2

- 해당 컨트롤에 마우스 오른 버튼 → 이벤트핸들러 추가선택



연습: Radio 버튼 사용

- 다음과 같이 다수의 라디오 버튼을 배치하고, Push Button을 누르면 선택된 라디오 버튼 번호 표시
 - 일일이 핸들러를 추가하기 번거러움
 - 한꺼번에 핸들러를 추가
- ON_COMMAND_RANGE 사용



ON_COMMAND_RANGE (IDC_RADIO1, IDC_RADIO6, OnRadioClick)

시작ID 번호

끝ID 번호

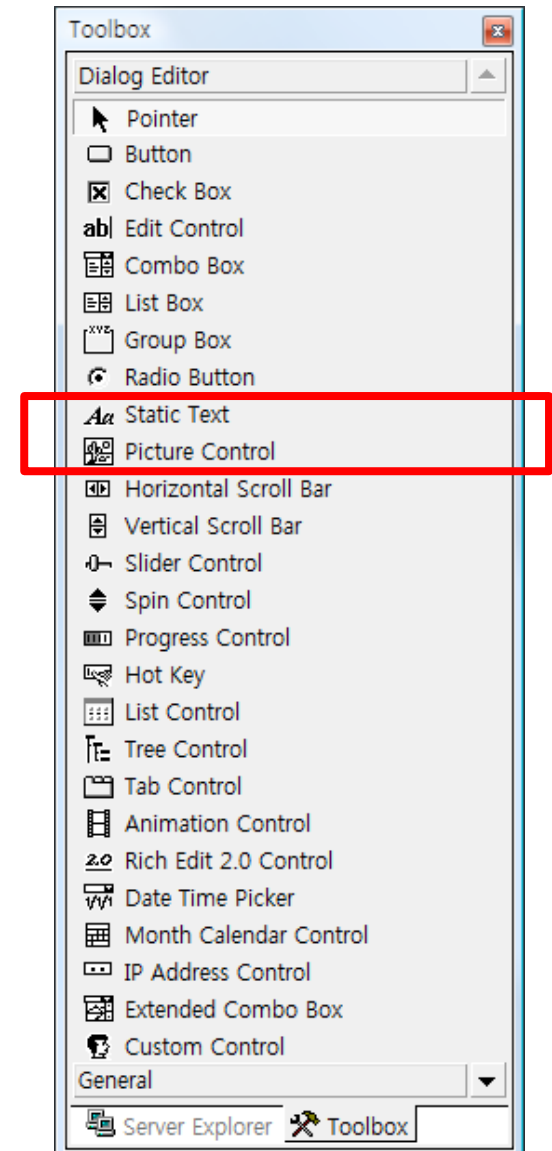
호출함수

더 많은 컨트롤 (More Controls)



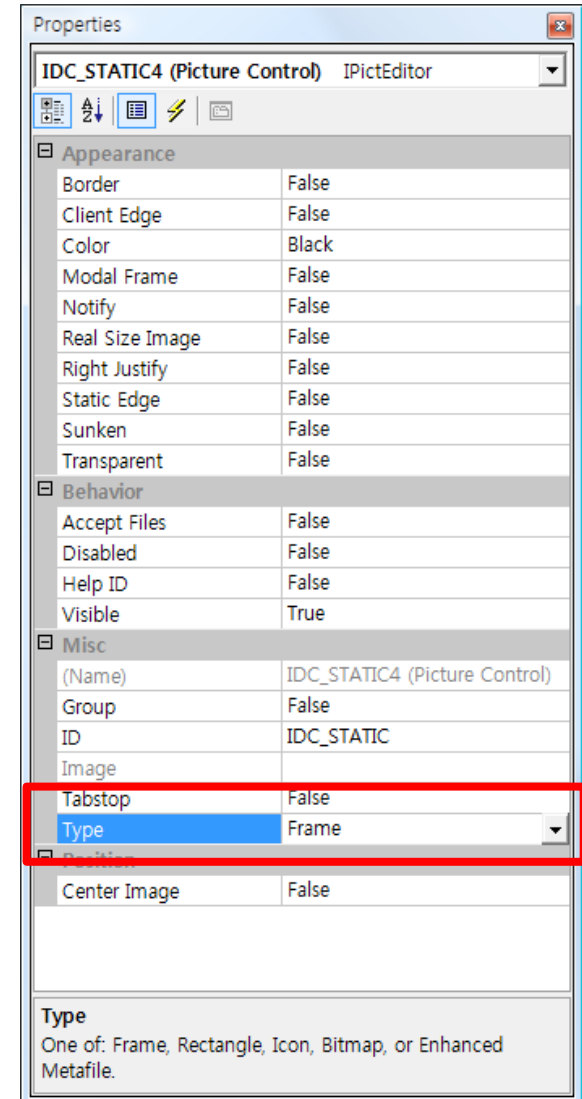
정적 컨트롤 (1/3)

- 정적 컨트롤 종류
 - 텍스트
 - 프레임(색으로 채워지지 않은 사각형),
사각형(색으로 채워진 사각형),
아이콘, 비트맵,
향상된 메타파일(Enhanced Metafile)



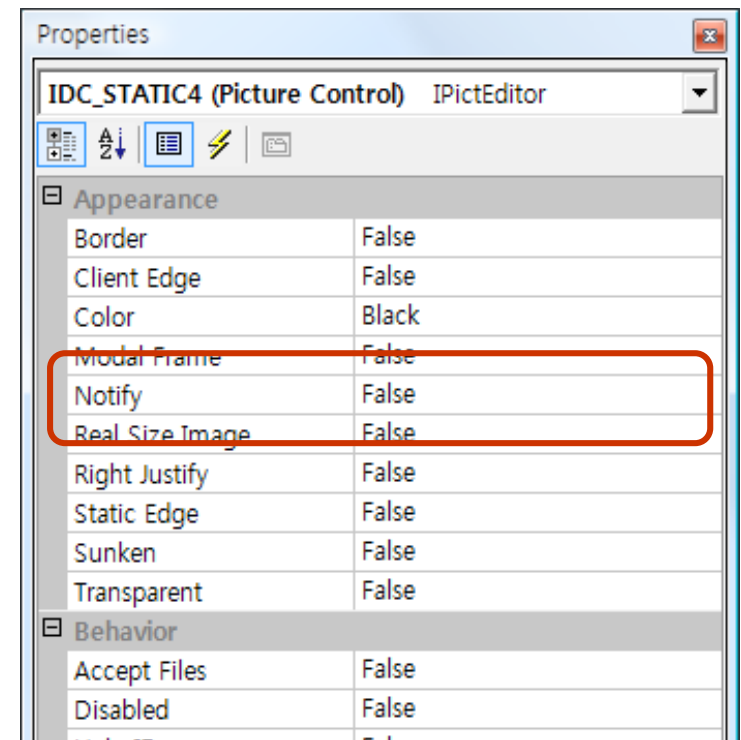
정적 컨트롤 (2/3)

- Picture Control 사용하기
 - Type을 통해 종류 결정
 - Frame, Icon, Bitmap, EMF...



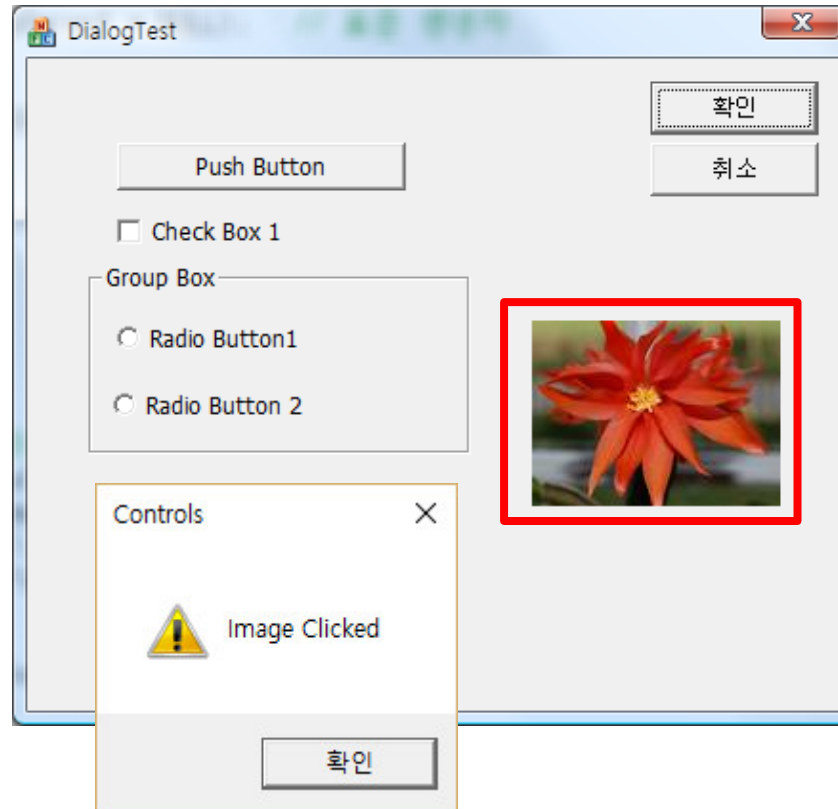
정적 컨트롤 (3/3)

- 그림을 버튼처럼?
- SS_NOTIFY 스타일을 설정하면 통지 메시지 발생
 - STN_CLICKED(클릭), STN_DBLCLK(더블 클릭), STN_DISABLE(비 활성화), STN_ENABLE(활성화)



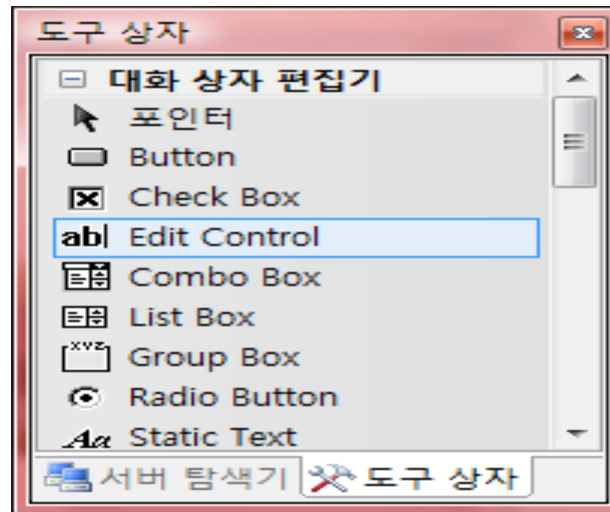
연습: 정적 컨트롤 붙이기

- 아래와 같이 정적 컨트롤에 그림 붙이기
- 그림 클릭시 메시지 출력

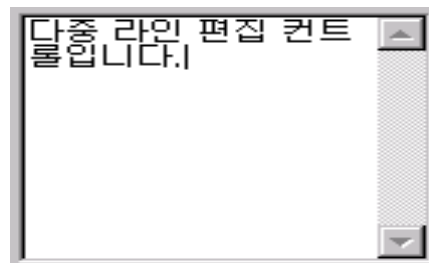


편집 컨트롤 (1/6)

- 편집 컨트롤

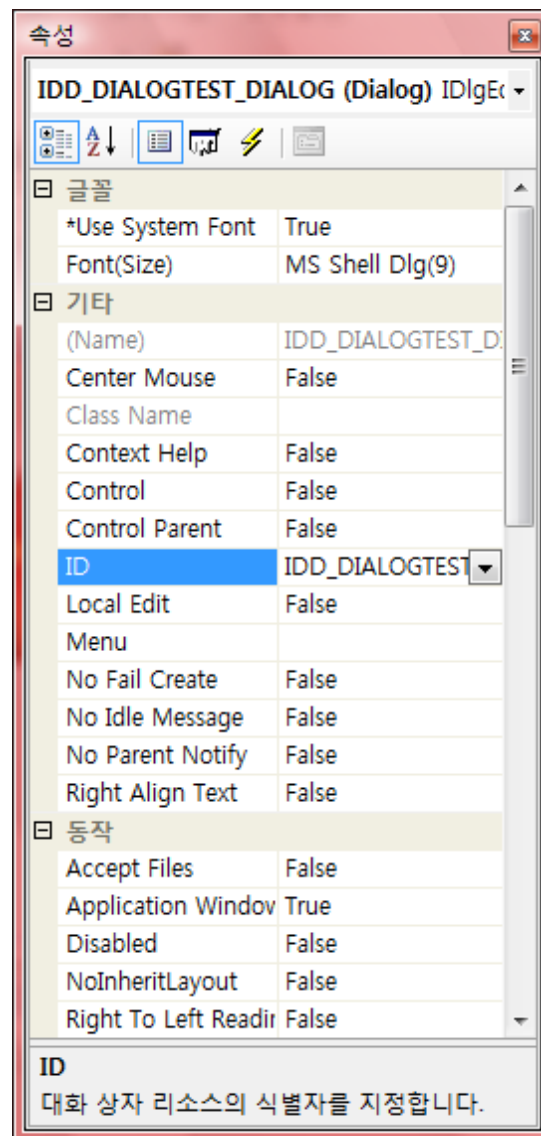


- 편집 컨트롤 종류



편집 컨트롤 (1/6)

- 속성 대화상자



편집 컨트롤 (2/6)

- 편집 컨트롤 스타일

편집 컨트롤 스타일	의미	속성 대화상자의 항목
ES_AUTOHSCROLL	줄 끝에 도달하면 자동으로 수평 스크롤	Auto HScroll
ES_AUTOVSCROLL	줄 끝에 도달하면 자동으로 수직 스크롤	Auto VScroll
ES_CENTER	텍스트를 가운데 정렬	Align text: Centered
ES_LEFT	텍스트를 왼쪽 정렬	Align text: Left
ES_LOWERCASE	입력된 모든 문자를 소문자로 변환	Lowercase
ES_MULTILINE	다중 라인 편집 컨트롤임을 나타냄	Multiline
ES_NOHIDESEL	컨트롤이 키보드 포커스를 잃더라도 선택된 텍스트가 계속 반전된 상태로 남아있도록 함	No hide selection
ES_NUMBER	숫자만 입력할 수 있다.	Number

편집 컨트롤 (3/6)

- 편집 컨트롤 스타일 (cont'd)

편집 컨트롤 스타일	의미	속성 대화상자의 항목
ES_OEMCONVERT	입력된 텍스트가 OEM 문자셋으로 변환되어 편집 컨트롤 내부에 저장	OEM convert
ES_PASSWORD	단일 라인 편집 컨트롤에만 사용할 수 있으며 입력된 문자를 *로 표시	Password
ES_READONLY	텍스트를 읽기만 가능하다.	Read-only
ES_RIGHT	텍스트를 오른쪽 정렬한다.	Align text: Right
ES_UPPERCASE	입력된 모든 문자를 대문자로 변환한다.	Uppercase
ES_WANTRETURN	다중 라인 편집 컨트롤에만 사용할 수 있으며 이 스타일을 지정하지 않으면 대화상자에서 Enter 키를 이용한 줄바꿈이 되지 않는다. Ctrl+Enter 키를 이용하면 이 스타일과 무관하게 줄바꿈을 할 수 있다.	Want return

편집 컨트롤 (4/6)

- 편집 컨트롤 통지 메시지

통지 메시지	의미
EN_CHANGE	사용자가 컨트롤의 내용을 변경하면 화면에 컨트롤을 다시 그리는데 그 후에 이 메시지가 발생한다.
EN_ERRSPACE	메모리가 부족하다.
EN_HSCROLL	사용자가 편집 컨트롤의 수평 스크롤 바를 클릭하였다.
EN_KILLFOCUS	키보드 포커스를 잃었다.
EN_MAXTEXT	더 이상 문자를 입력할 수 없다. CEdit::SetLimitText 함수로 문자 개수를 제한한 경우나 ES_AUTOHSCROLL, ES_AUTOVSCROLL 등의 스타일을 지정하지 않은 상태에서 줄 끝까지 입력한 경우에 발생한다.
EN_SETFOCUS	키보드 포커스를 얻었다.
EN_UPDATE	사용자가 컨트롤의 내용을 변경하면 화면에 컨트롤을 다시 그리는데 그 직전에 이 메시지가 발생한다.
EN_VSCROLL	사용자가 편집 컨트롤의 수직 스크롤 바를 클릭하였다.

연습: 편집 컨트롤 붙이기

- 아래와 같이 편집 컨트롤 붙이기



편집 컨트롤을 제어하려면?

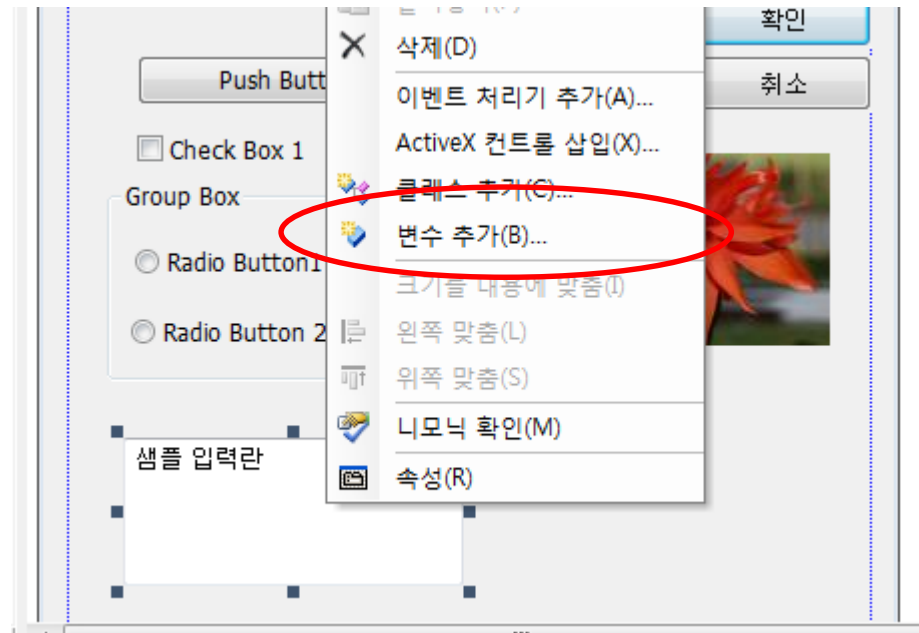
- 텍스트의 상태를 변경하고 (알고) 싶다면?
→ 컨트롤 메시지 활용



- 컨트롤 메시지를 주기 위해서는...
 - CEdit 타입의 변수를 추가하고 컨트롤과 연결
 - 직접 연결 (DoDataExchange 함수 이용)
 - 또는 컨트롤의 변수 추가 기능을 이용하여 자동으로 추가

편집 컨트롤을 제어하려면?

- 컨트롤 메시지를 주기 위해서는...
 - CEdit 타입의 변수를 추가하고 컨트롤과 연결
- 직접 연결 (DoDataExchange 함수 이용)
- 또는 컨트롤의 변수 추가 기능을 이용하여 자동으로 추가



편집 컨트롤 (5/6)

- 텍스트를 변경하거나 입력된 텍스트를 알아내기

```
m_edit.SetWindowText("초기값입니다.");  
CString str;  
m_edit.GetWindowText(str);
```

- 입력 가능한 문자열의 길이를 제한하기

```
m_edit.SetLimitText(10);
```

- 각종 편집 작업

```
m_edit.Clear();  
m_edit.Cut();  
m_edit.Copy();  
m_edit.Paste();  
m_edit.Undo();
```

편집 컨트롤 (6/6)

- 텍스트 선택과 치환하기

- ① `m_edit.SetSel(5, 7);`
- ② `m_edit.ReplaceSel("ABC");`

①을 실행하기 전



0123456789

①을 실행한 후



0123456789

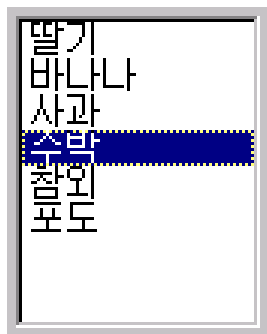
②를 실행한 후



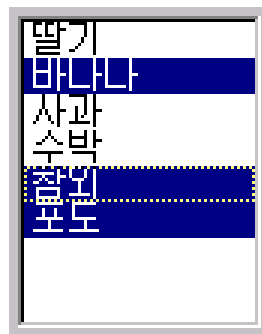
01234ABC789

리스트 박스 컨트롤 (1/8)

- 리스트 박스 컨트롤 종류

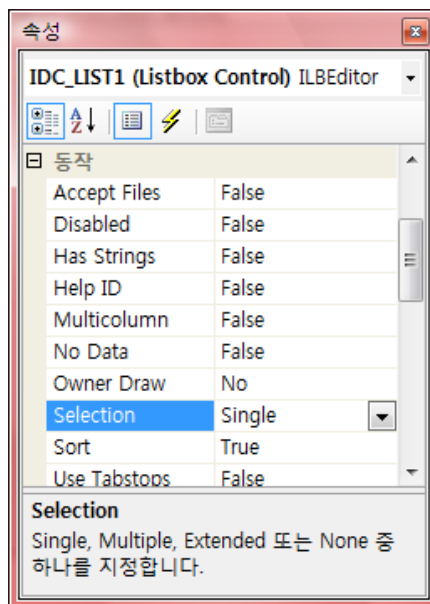


단일 선택
리스트 박스



다중 선택
리스트 박스

- 속성 대화상자



리스트 박스 컨트롤 (2/8)

- 리스트 박스 컨트롤 스타일

리스트 박스 컨트롤 스타일	의미	속성 대화상자 항목
LBS_DISABLENOSCROLL	표시할 항목의 개수가 적은 경우에도 수직 스크롤 바가 사라지지 않는다.	Disable no scroll
LBS_EXTENDEDSEL	SHIFT, CTRL 키와 마우스 클릭을 이용한 다중 선택이 가능하다.	Selection: Extended
LBS_HASSTRINGS	LBS_OWNERDRAW* 스타일을 지정하지 않을 경우의 디폴트 스타일이며 컨트롤이 문자열을 저장 및 관리한다.	Has strings
LBS_MULTICOLUMN	여러 줄(Column)로 구성된 리스트 박스를 생성하며 항목이 많을 경우 수평 스크롤이 가능하다.	Multi-column

리스트 박스 컨트롤 (3/8)

- 리스트 박스 컨트롤 스타일 (cont'd)

리스트 박스 컨트롤 스타일	의미	속성 대화상자 항목
LBS_MULTIPLESEL	마우스 클릭을 이용한 다중 선택이 가능하다.	Selection: Multiple
LBS_NODATA	항목 데이터를 컨트롤이 아닌 부모 윈도우가 유지하며 필요할 때마다 부모 윈도우가 직접 그린다. 항목의 개수가 1000 개 이상일 경우에	없음
LBS_NOINTEGRALHEIGHT	이 스타일을 지정하지 않으면 항목의 일부가 잘려서 보이지 않는 경우가 발생할 수 있다.	No integral height
LBS_NOREDRAW	항목에 변화가 생기더라도 리스트 박스 컨트롤을 다시 그리지 않는다.	No redraw

리스트 박스 컨트롤 (4/8)

- 리스트 박스 컨트롤 스타일 (cont'd)

리스트 박스 컨트롤 스타일	의미	속 성 대 화 상 자 의 항목
LBS_NOSEL	항목을 선택할 수 없다.	Selection: None
LBS_NOTIFY	사용자가 항목을 클릭하거나 더블 클릭하면 부모 윈도우에게 통지 메시지를 보낸다.	Notify
LBS_OWNERDRAWFIXED	부모 윈도우가 리스트 박스 항목을 직접 그리되 항목의 높이가 일정한 경우이다.	Owner draw: Fixed
LBS_OWNERDRAWVARIABLE	부모 윈도우가 리스트 박스 항목을 직접 그리되 항목의 높이가 일정하지 않은 경우이다.	Owner draw: Variable

리스트 박스 컨트롤 (5/8)

- 리스트 박스 컨트롤 스타일 (cont'd)

리스트 박스 컨트롤 스타일	의미	속 성 대 화 상 자 의 항목
LBS_SORT	항 목 이 문 자 열 인 경 우 정 렬 하 여 표시한다.	Sort
LBS_STANDARD	LBS_NOTIFY, LBS_SORT, WS_VSCROLL, WS_BORDER 네 가지 스타일의 조합이다.	Notify, Sort, Border, Vertical scroll
LBS_USETABSTOPS	이 스타일을 지정하면 항목 문자열에 포 함 된 탭 문 자 ('Wt') 를 제 대 로 처리할 수 있다.	Use tabstops
LBS_WANTKEYBOAR DINPUT	리 스토 박스 컨 트롤 이 키 보 드 포커스를 가진 상태에서 사용자가 키를 누르면 부모 윈도우가 이를 감지하여 특별한 처리를 할 수 있다.	Want key input

리스트 박스 컨트롤 (6/8)

- 리스트 박스 컨트롤 통지 메시지

통지 메시지	의미
LBN_DBLCLK	사용자가 항목을 더블 클릭하였다.
LBN_SELCHANGE	사용자가 선택을 변경하였다.
LBN_SELCANCEL	사용자가 선택을 취소하였다.
LBN_SETFOCUS	키보드 포커스를 얻었다.
LBN_KILLFOCUS	키보드 포커스를 잃었다.
LBN_ERRSPACE	메모리가 부족하다.

- LBN_DBLCLK, LBN_SELCHANGE, LBN_SELCANCEL 통지 메시지는 LBS_NOTIFY 스타일을 설정해야 발생

리스트 박스 컨트롤 (7/8)

- 항목 추가와 삭제

```
m_list.AddString("사과");  
m_list.DeleteString(3);
```

- 항목 선택

```
// 단일 선택 리스트 박스 컨트롤인 경우  
m_list.SetCurSel(2);
```

```
// 다중 선택 리스트 박스 컨트롤인 경우  
m_list.SetSel(2);  
m_list.SetSel(3, FALSE);
```

리스트 박스 컨트롤 (8/8)

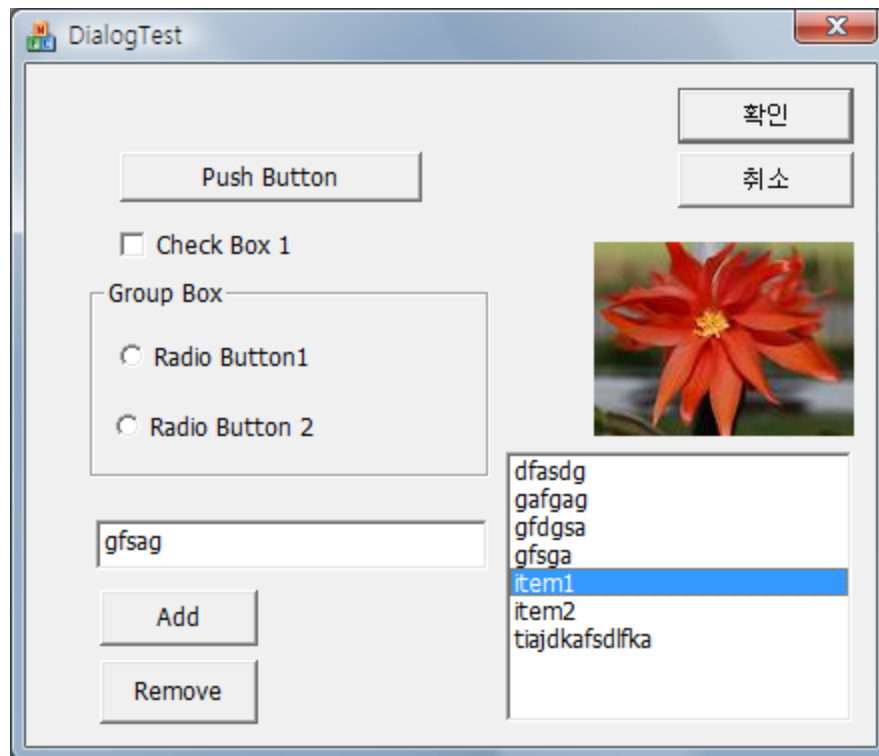
- 선택된 항목 알아내기

```
// 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);
}
```

연습: 리스트 박스 붙이기

- 편집 컨트롤과 리스트 박스 컨트롤을 만들고,
편집 컨트롤을 통해 리스트 박스에 글을 추가하는
프로그램을 만드시오



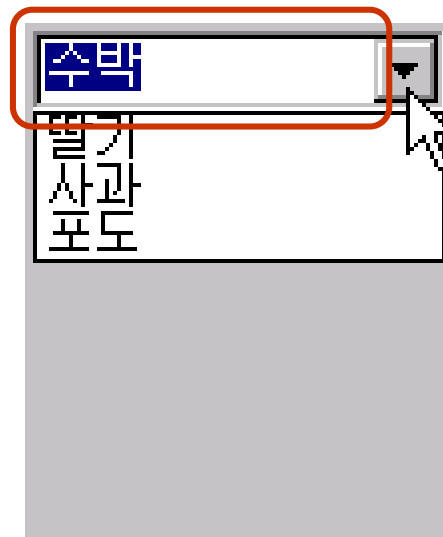
콤보 박스 컨트롤 (1/5)

- 콤보 박스 컨트롤 종류

편집 컨트롤

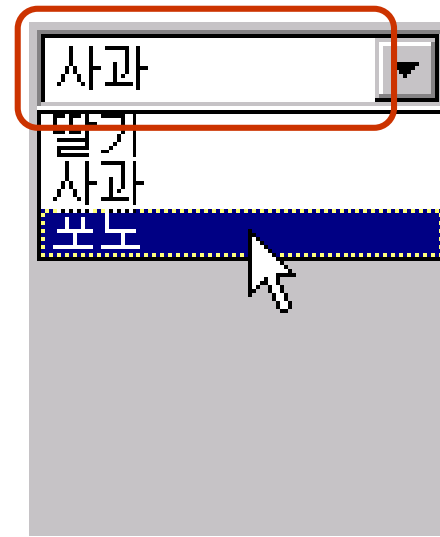


단순
스타일



드롭다운
스타일

정적 컨트롤

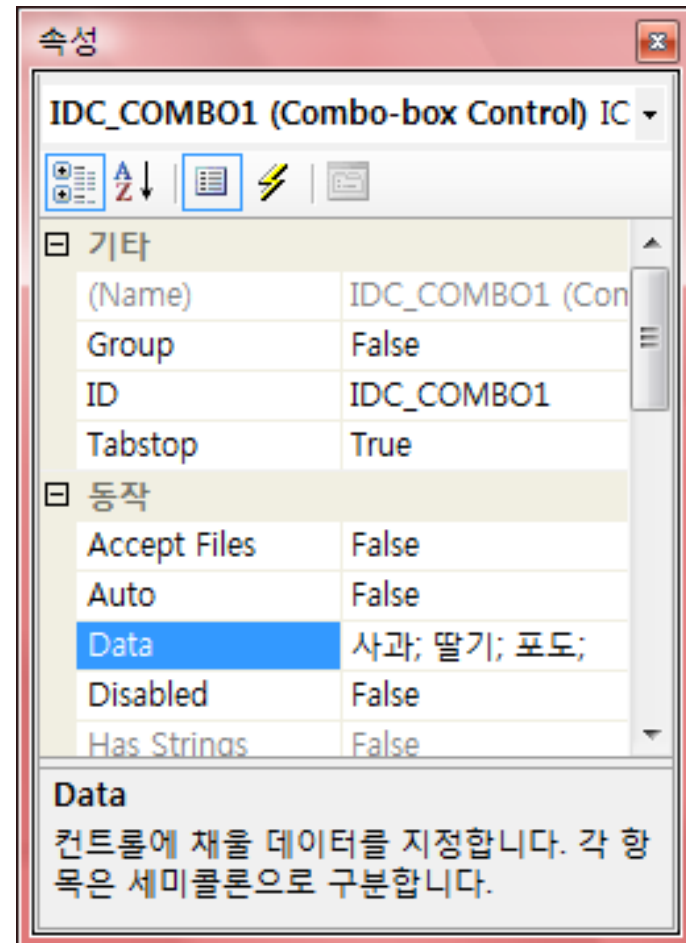
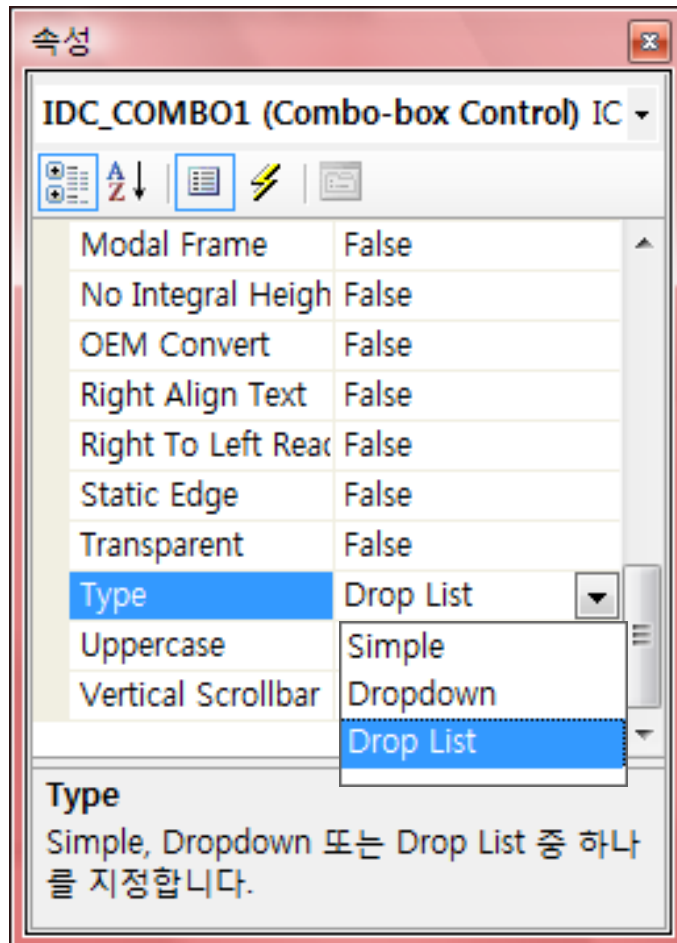


드롭다운 리스트
스타일

리스트
박스
컨트롤

콤보 박스 컨트롤 (2/5)

- 속성 대화상자



콤보 박스 컨트롤 (3/5)

- 콤보 박스 컨트롤 스타일

콤보 박스 컨트롤 스타일	의미	속성 대화상자의 항목
CBS_AUTOHSCROLL	= ES_AUTOHSCROLL	Auto HScroll
CBS_DISABLENOSCROLL	= LBS_DISABLENOSCROLL	Disable no scroll
CBS_DROPDOWN	드롭다운 스타일	Type: Dropdown
CBS_DROPDOWNLIST	드롭다운 리스트 스타일	Type: Drop List
CBS_HASSTRINGS	= LBS_HASSTRINGS	Has strings
CBS_LOWERCASE	= ES_LOWERCASE	Lowercase
CBS_NOINTEGRALHEIGHT	= LBS_NOINTEGRALHEIGHT	No integral height
CBS_OEMCONVERT	= ES_OEMCONVERT	OEM convert
CBS_OWNERDRAWFIXED	= LBS_OWNERDRAWFIXED	Owner draw: Fixed
CBS_OWNERDRAWVARIABLE	= LBS_OWNERDRAWVARIABLE	Owner draw: Variable
CBS_SIMPLE	단순 스타일	Type: Simple
CBS_SORT	= LBS_SORT	Sort
CBS_UPPERCASE	= ES_UPPERCASE	Uppercase

콤보 박스 컨트롤 (4/5)

- 콤보 박스 컨트롤 통지 메시지

통지 메시지	의미	스타일		
		단순	드롭 다운	드롭다운 리스트
CBN_CLOSEUP	리스트 박스가 닫혔다.		•	•
CBN_DBLCLK	사용자가 항목을 더블 클릭하였다.	•		
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	•	•	•
CBN_SELENDCANCEL	= LBN_SELENDCANCEL		•	•
CBN_SELENDOK	사용자가 항목을 선택하였다.	•	•	•
CBN_SETFOCUS	키보드 포커스를 얻었다.	•	•	•

콤보 박스 컨트롤 (5/5)

- 항목 추가와 삭제

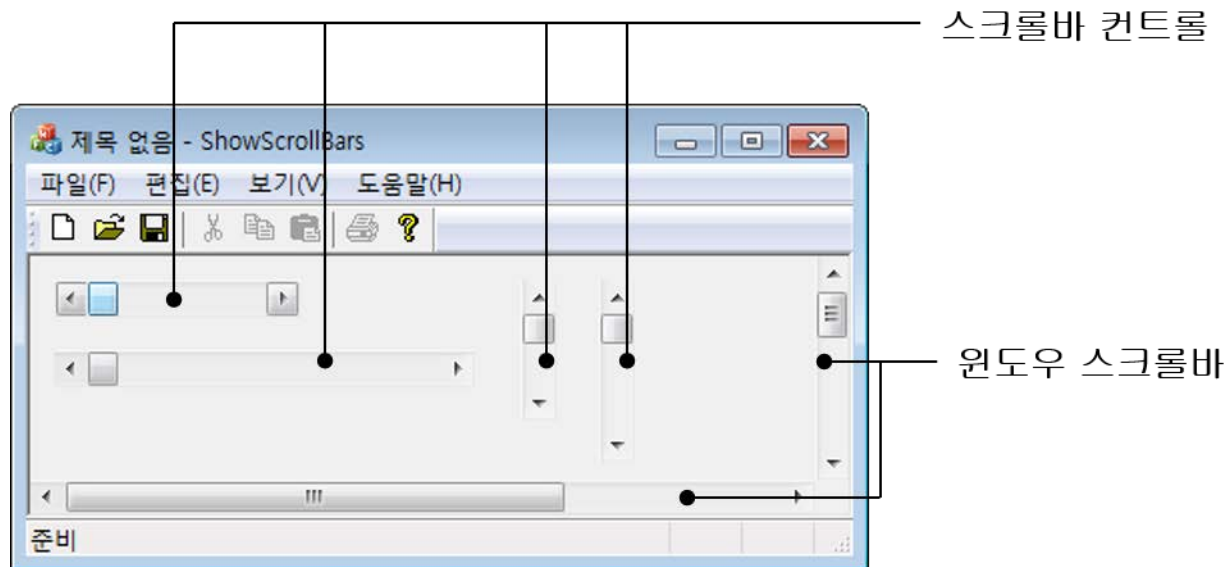
```
m_combo.AddString("사과");  
m_combo.DeleteString(3);
```

- 선택된 항목 알아내기

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

스크롤 바 컨트롤 (1/6)

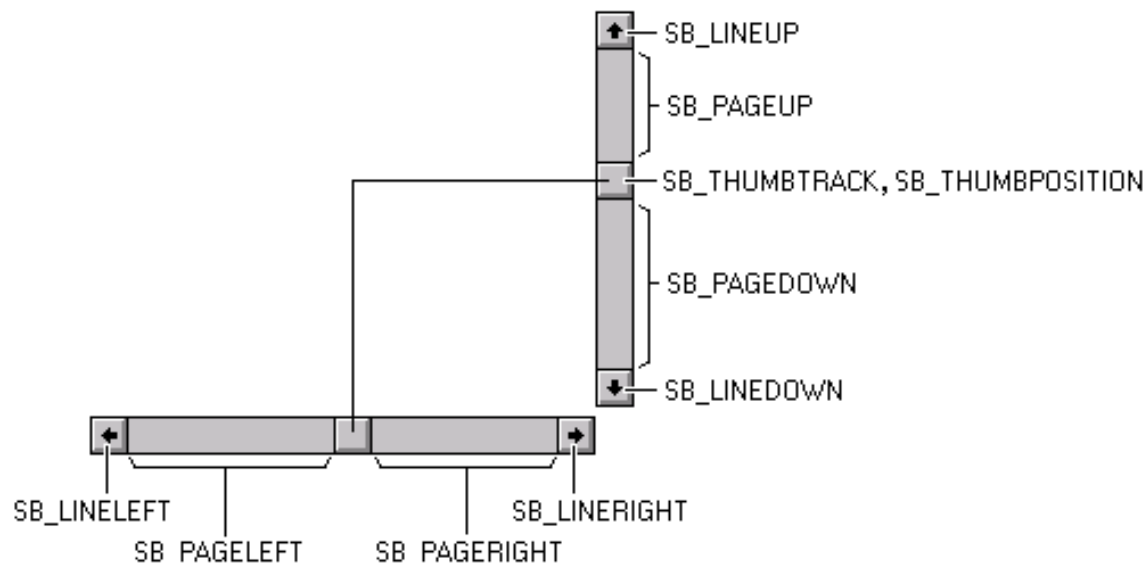
- 스크롤 바 컨트롤
 - 화면의 일정 영역을 스크롤하거나 정해진 범위의 값을 변경시키는 용도로 사용
- 스크롤 바의 종류



스크롤 바 컨트롤 (2/6)

- 스크롤 바 코드

- 사용자가 스크롤 바를 조작하면 WM_HSCROLL 또는 WM_VSCROLL 메시지 발생
- 메시지와 더불어 사용자의 구체적인 행위를 알 수 있는 정보가 전달됨



스크롤 바 컨트롤 (3/6)

- 스크롤 바 코드

스크롤 바 코드	응용 프로그램이 해야 할 작업
SB_LINELEFT, SB_LINERIGHT	왼쪽 또는 오른쪽으로 한 줄만큼 데이터를 스크롤한다.
SB_PAGELEFT, SB_PAGERIGHT	왼쪽 또는 오른쪽으로 한 페이지만큼 데이터를 스크롤한다.
SB_LINEUP, SB_LINEDOWN	위쪽 또는 아래쪽으로 한 줄만큼 데이터를 스크롤한다.
SB_PAGEUP, SB_PAGEDOWN	위쪽 또는 아래쪽으로 한 페이지만큼 데이터를 스크롤한다.
SB_THUMBTRACK	스크롤 박스의 현재 위치로 이동한다. (스크롤 박스를 마우스로 끌고 있을 때 계속 발생)
SB_THUMBPOSITION	스크롤 박스의 현재 위치로 이동한다. (스크롤 박스를 마우스로 끌다가 놓을 때 한 번만 발생)

스크롤 바 컨트롤 (4/6)

- WM_HSCROLL/WM_VSCROLL 메시지 핸들러

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

- nSBCode
 - 스크롤 바 코드
- nPos
 - 스크롤 박스의 위치
 - 스크롤 바 코드가 SB_THUMBPOSITION 또는 SB_THUMBTRACK인 경우에만 사용
- pScrollBar
 - 스크롤 바 컨트롤을 조작하는데 사용할 수 있는 CScrollBar 개체체를 가리키는 포인터
 - 윈도우 스크롤 바에서 발생한 것이면 NULL 값

스크롤 바 컨트롤 (5/6)

- CScrollBar 주요 멤버 함수 (일부)

멤버 함수	역할
SetScrollRange()	스크롤 박스 위치의 최소값과 최대값을 설정한다.
SetScrollPos()	스크롤 박스의 현재 위치를 설정한다.
GetScrollPos()	스크롤 박스의 현재 위치를 얻는다.

스크롤 바 컨트롤 (6/6)

- 발생 메시지: WM_HSCROLL 또는 WM_VSCROLL
- MessageMap 추가: 매크로 함수 사용
 - Scroll Bar는 ID에 상관없이 공통된 매크로 함수 사용

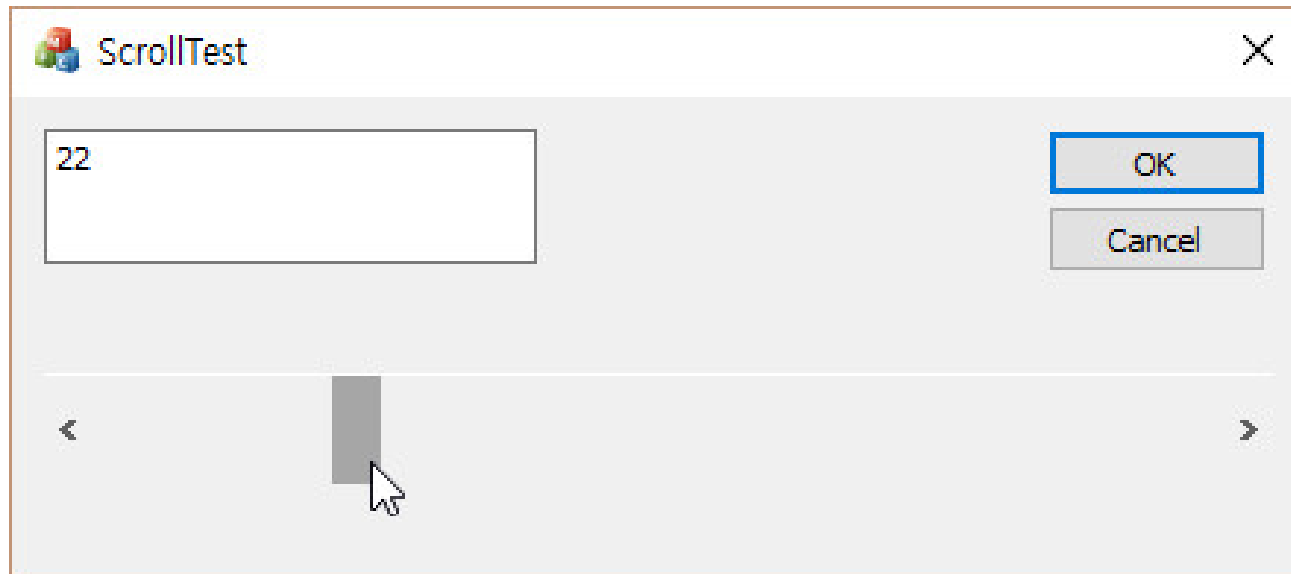
```
ON_WM_HSCROLL() 또는  
ON_WM_VSCROLL()
```

- Event Message Handler 추가

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

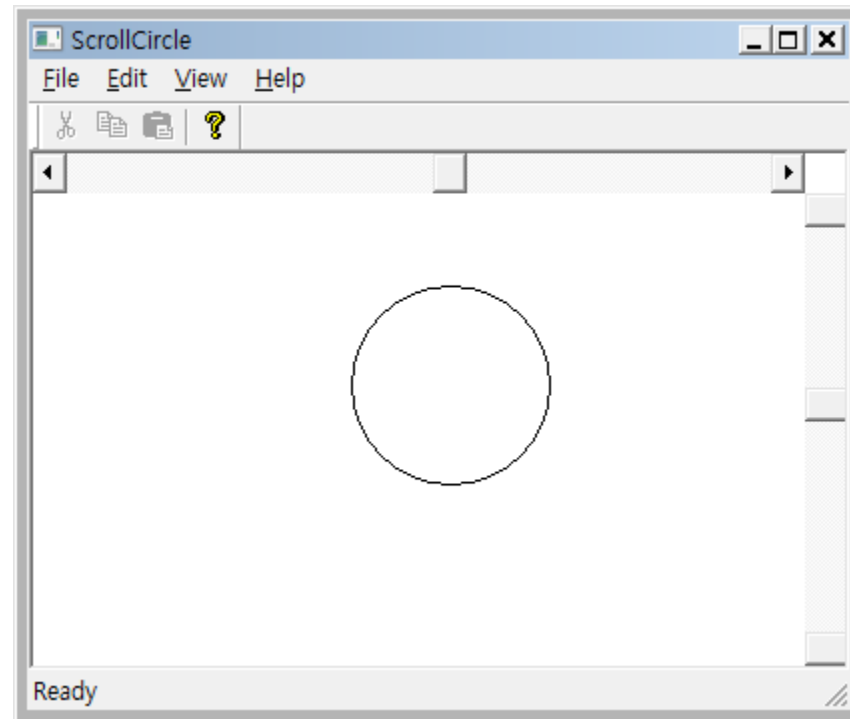
연습: 편집 컨트롤과 스크롤 바 사용

- 다음과 같이 편집 컨트롤과, 스크롤 바를 붙이고, 스크롤 바를 움직이면 편집 컨트롤에 그 값을 표시하는 프로그램을 작성 (대화상자 사용)
 - 스크롤 바의 범위: 0~100



연습: 스크롤 바로 원의 위치 조정

- 다음과 같이 가로, 세로 스크롤 바 각각 1개씩을 붙이고, 스크롤 바를 움직이면 원의 x 와 y 의 좌표가 변하는 프로그램을 작성



Q & A