

Window Programming

Visual C++ MFC Programming

Lecture 13

김예진

Dept. of Game Software

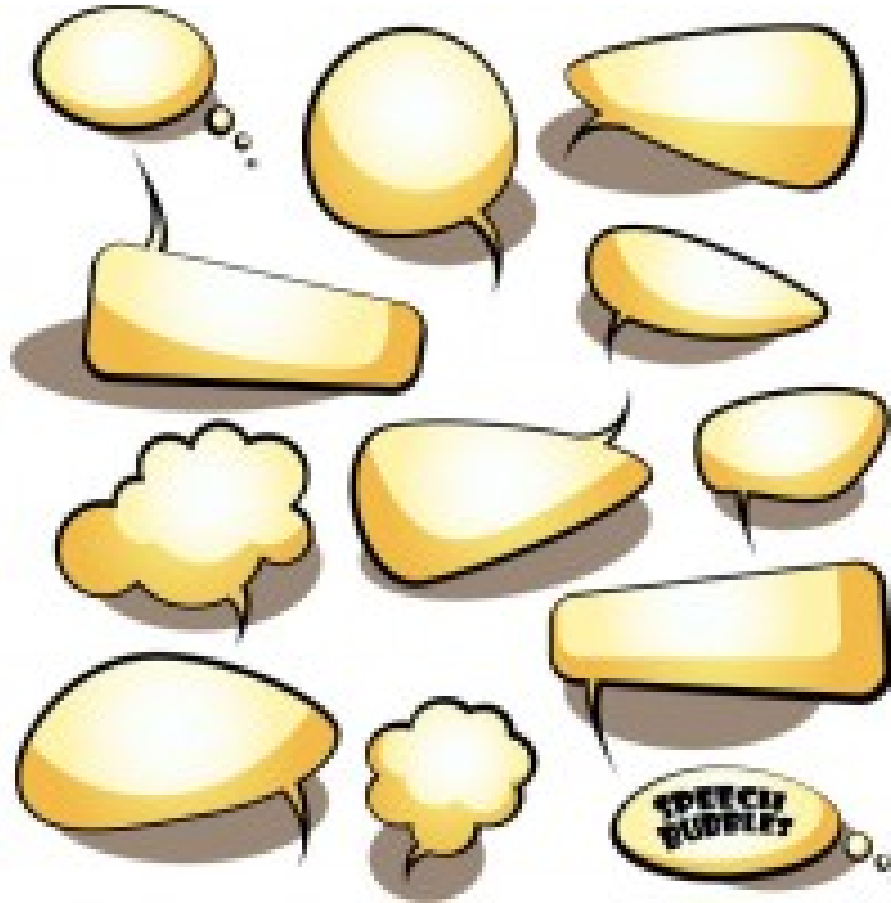
Notices

- 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) → Avg: 5.97
- 05/31: HW 4 (Due: 06/08)
- 06/14: Final
 - 5문제, ~75 min., 강의록 1~13
- 06/15: 보강
 - 501: 09:00~11:00
 - 502: 14:00~16:00

Plan: 대화상자 (Dialog Box)

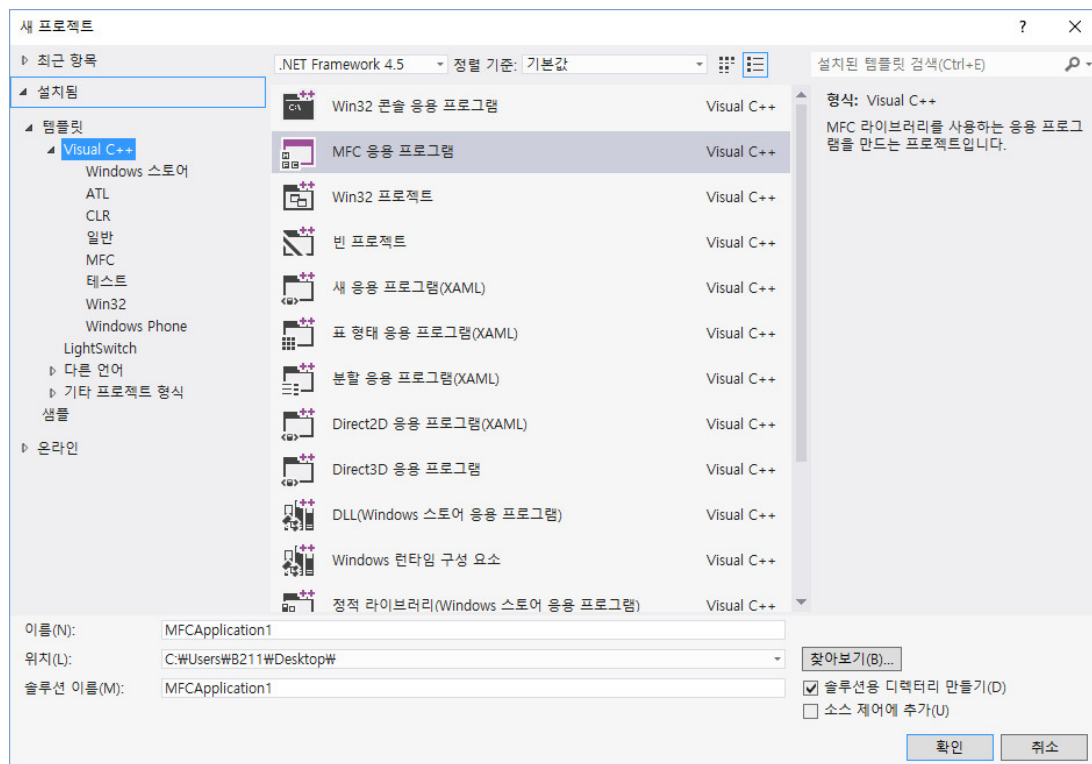
- 모드형 대화상자 (Modal Dialog Box)
 - 연습: 모드형 대화상자
- 비모드형 대화상자 (Modaless Dialog Box)
 - 연습: 비모드형 대화상자

대화상자 (Dialog Box)



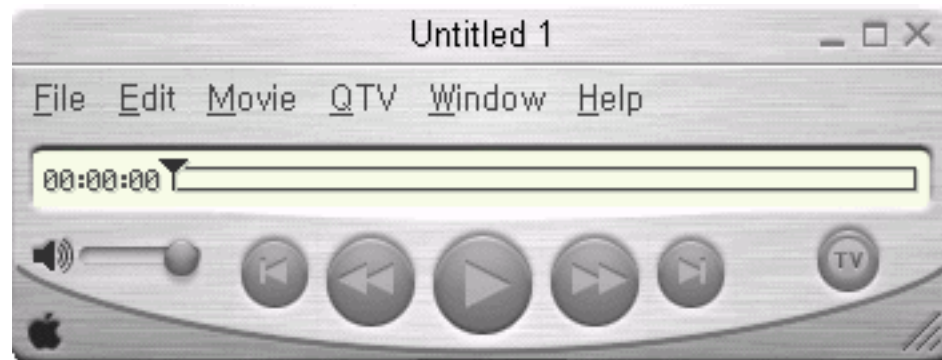
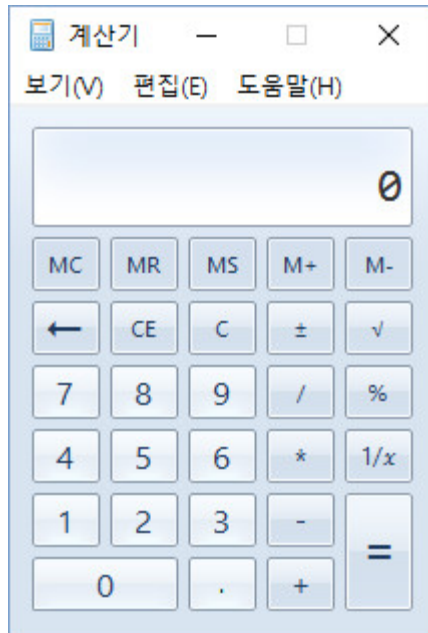
대화상자 (Dialog Box)?

- 대화상자
 - 다양한 컨트롤을 포함하고 있는 일종의 윈도우
 - 사용자로부터 입력을 받거나 정보를 출력



대화상자 기반 응용 프로그램?

- 대화상자 기반 응용 프로그램
 - 대화상자가 메인 윈도우 역할을 하는 응용 프로그램



오늘의 주제

- 대화상자를 program의 필요에 따라 화면에 나타낼 수 있는 방법
- 대화상자에 입력한 사용자의 입력값들을 얻어오거나, program에서 대화상자에 값을 전달해 주는 방법

2가지 종류의 대화 상자

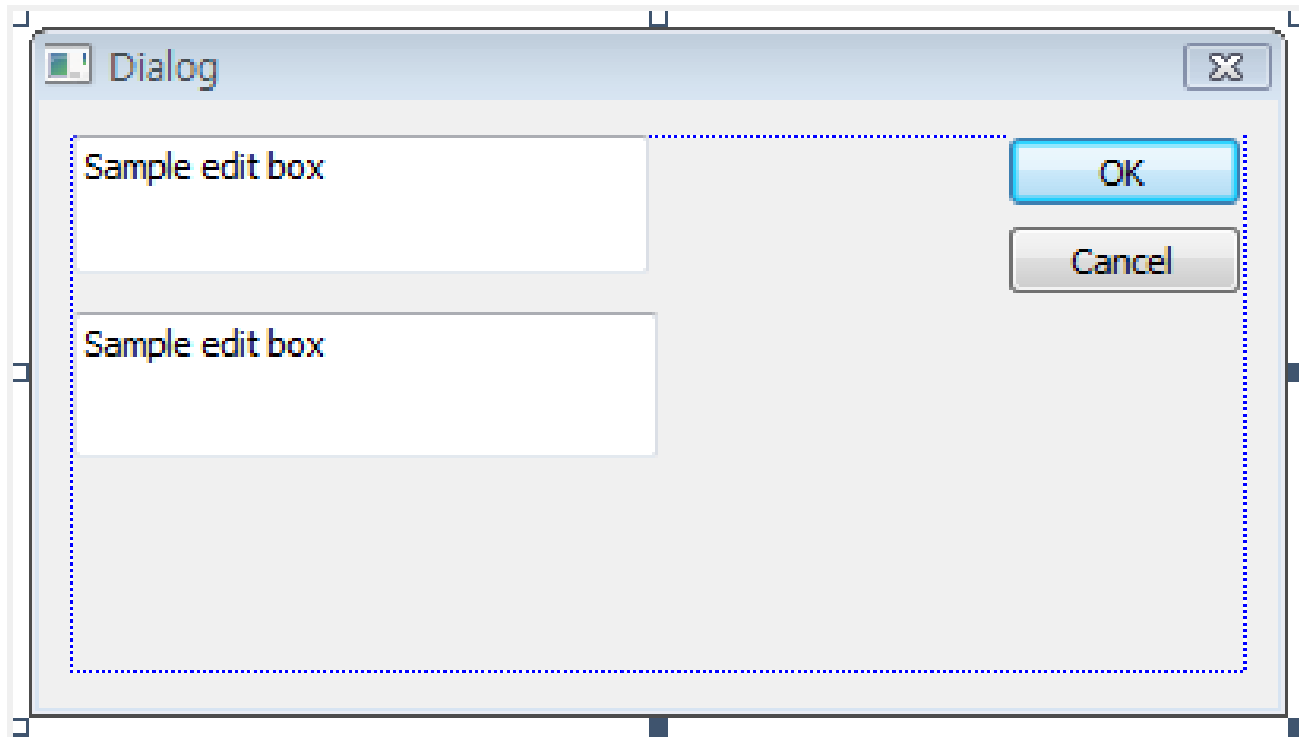
- 모드형 대화상자 (Modal Dialog Box)
 - 대화상자를 닫지 않으면 응용 프로그램이 더 이상 진행할 수 없음
- 비 모드형 대화상자 (Modeless Dialog Box)
 - 대화상자를 닫지 않더라도 응용 프로그램이 계속 진행할 수 있음

대화상자 템플릿 (1/3)

- 대화상자 템플릿
 - 대화상자는 운영체제 내부적으로 대화상자 템플릿(Dialog Box Template)이라는 형태로 존재
 - 대화상자 자체와 포함된 컨트롤에 대한 모든 정보를 가지고 있는 이진(binary) 데이터
 - 리소스로 다자인하여 실행 파일에 포함
- 대화상자 템플릿 작성
 - 비주얼 C++ 리소스 편집기나 텍스트 편집기를 이용하여 리소스 스크립트(*.RC) 작성
 - 컴파일 과정에서 리소스 컴파일러가 이진 파일(*.RES)로 생성
 - 링크 과정에서 실행 파일에 포함

대화상자 템플릿 (2/3)

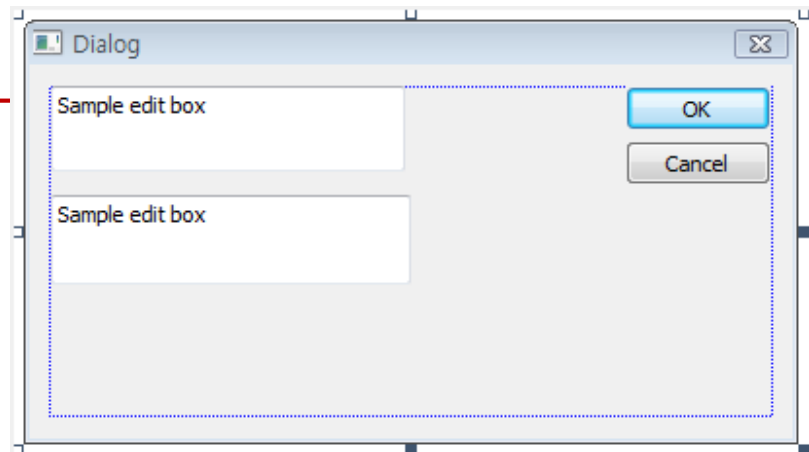
- 리소스 스크립트 작성



대화상자 템플릿 (3/3)

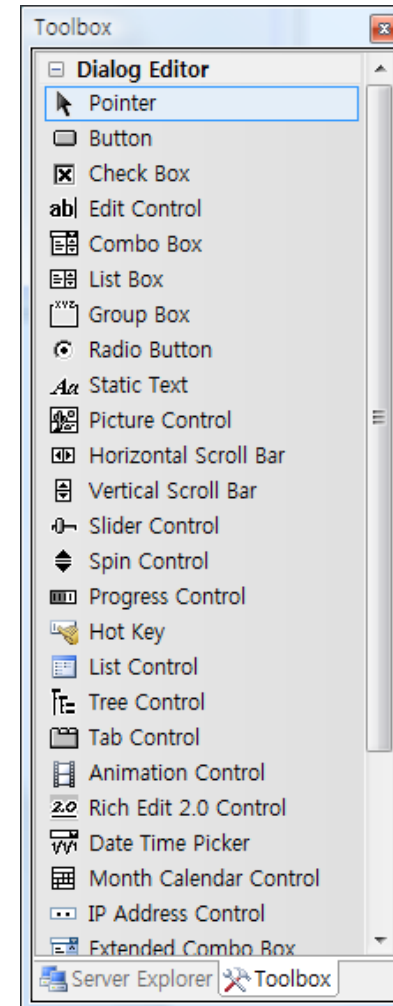
- 리소스 스크립트 작성 (cont'd)

```
IDD_DIALOG1 DIALOG DISCARDABLE 0, 0, 187, 98
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Dialog"
FONT 10, "System"
BEGIN
    DEFPUSHBUTTON "OK", IDOK, 130, 7, 50, 14
    PUSHBUTTON "Cancel", IDCANCEL, 130, 24, 50, 14
    EDITTEXT IDC_EDIT1, 7, 7, 117, 14, ES_AUTOHSCROLL
    EDITTEXT IDC_EDIT2, 7, 24, 117, 14, ES_AUTOHSCROLL
END
```



대화상자 편집기 (1/4)

- 컨트롤 추가와 삭제
 - 도구상자 (Toolbox) 툴바 이용

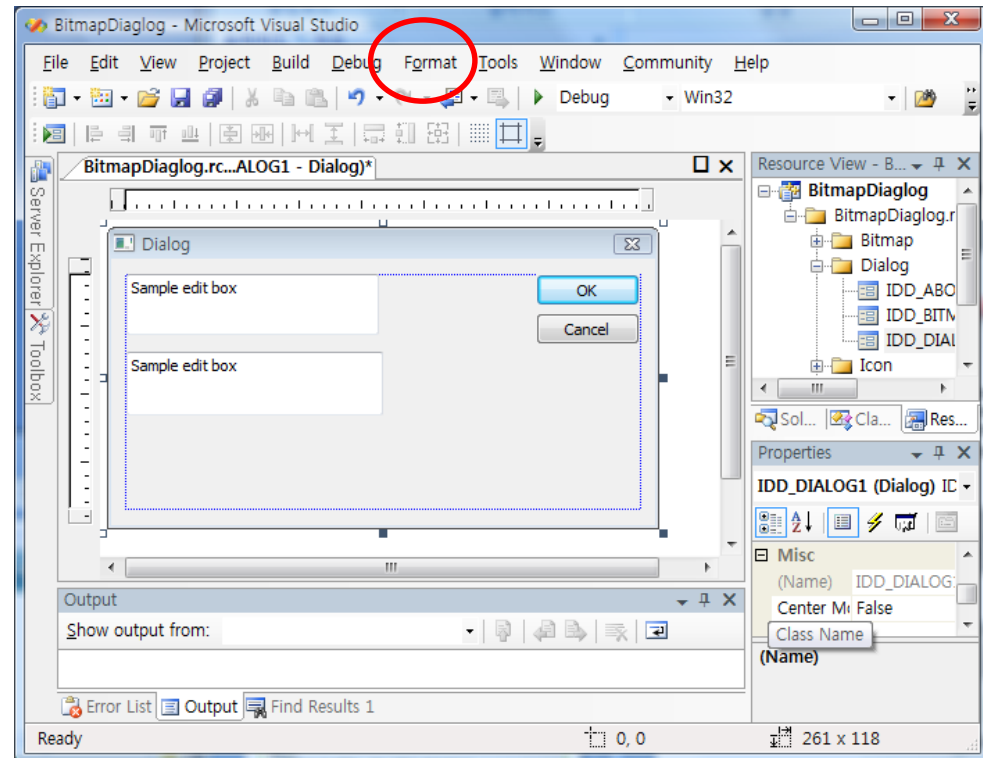


대화상자 편집기 (2/4)

- 컨트롤 배치
 - 대화상자(Dialog Box) 툴바 이용

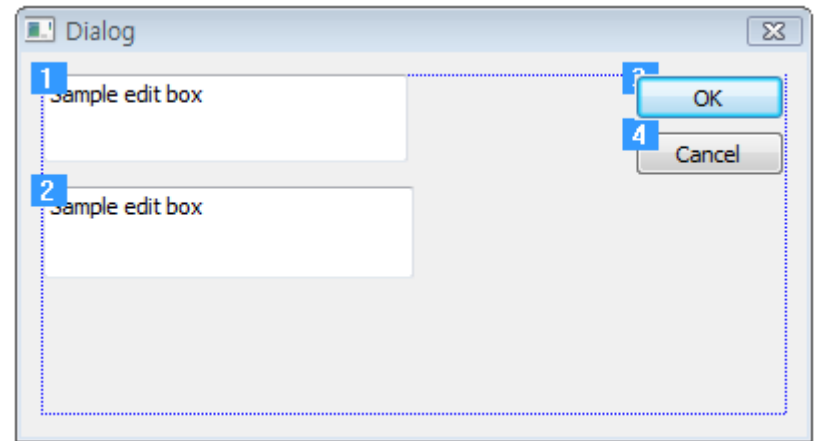
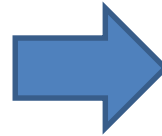
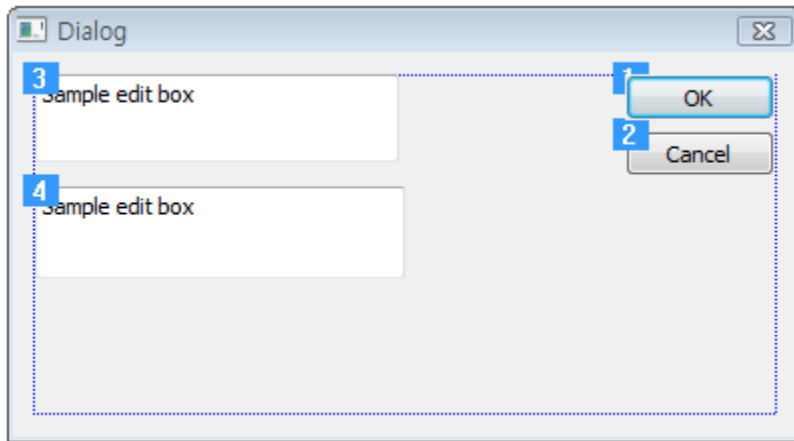


- 서식(Format) 메뉴 이용



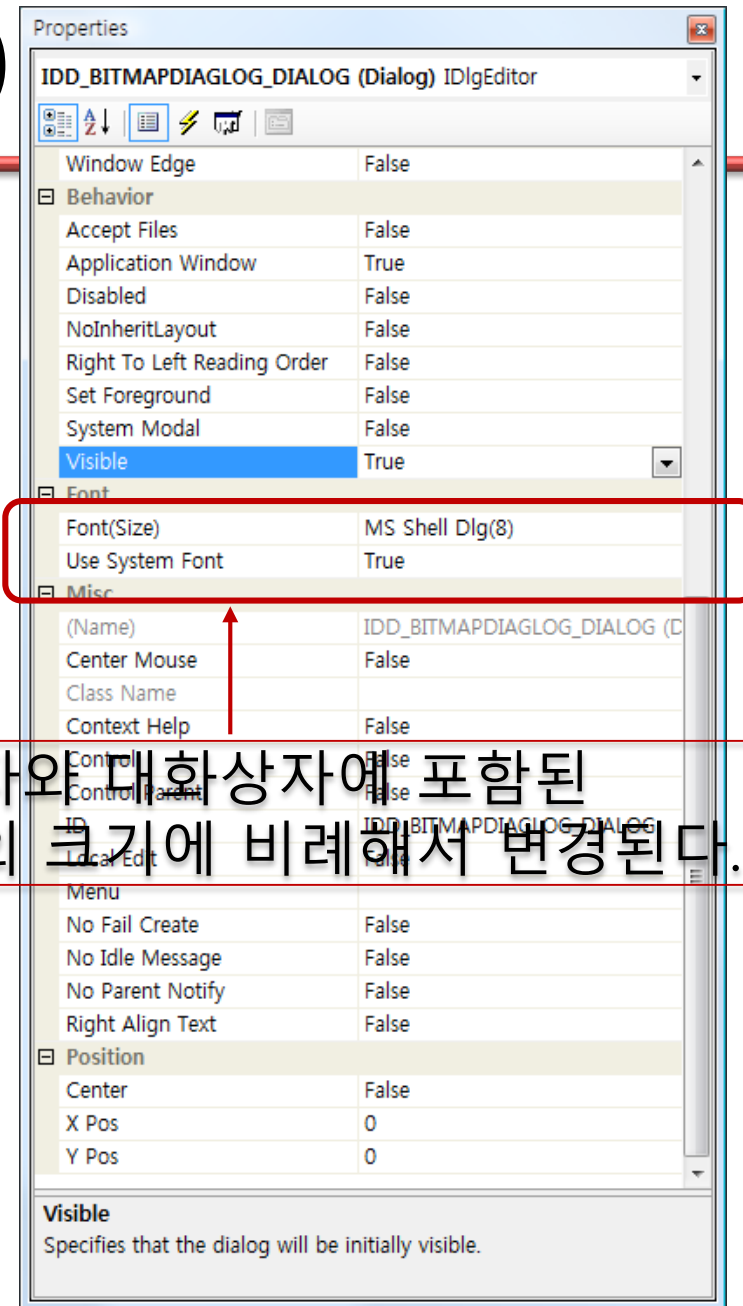
대화상자 편집기 (3/4)

- 탭(Tab) 순서
 - 탭 키를 눌렀을 때 키보드 포커스가 이동하는 순서
 - [서식]→[탭 순서] 메뉴를 이용하여 변경



대화상자 편집기 (4/4)

- 대화상자 속성



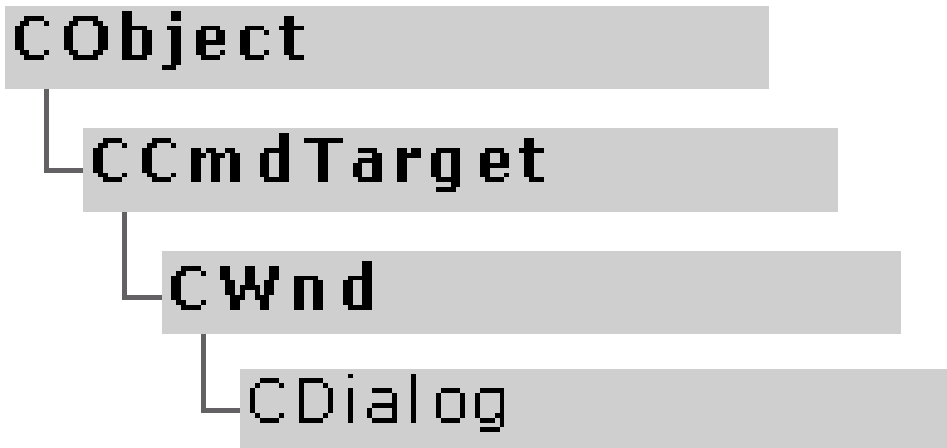
폰트를 바꾸면 대화상자와 대화상자에 포함된 컨트롤의 크기가 폰트의 크기에 비례해서 변경된다.

2가지 종류의 대화 상자

- 모드형 대화상자 (Modal Dialog Box)
 - 대화상자를 닫지 않으면 응용 프로그램이 더 이상 진행할 수 없음
- 비 모드형 대화상자 (Modeless Dialog Box)
 - 대화상자를 닫지 않더라도 응용 프로그램이 계속 진행할 수 있음

모드형 대화상자 (1/4)

- MFC 클래스 계층도



모드형 대화상자 (1/4)

- 모드형 대화상자 작성 및 활용 순서
 - ① 대화상자 리소스 작성
 - 리소스 뷰(Resource View)
 - ② CDialog (파생) 클래스 객체 생성
 - [프로젝트] → [클래스 추가] 메뉴 이용
 - ③ CDialog::DoModal() 함수 호출로 대화상자 표시

모드형 대화상자 (2/4)

- CDialog 클래스 가상 함수

```
virtual BOOL CDialog::OnInitDialog ( );
```

- 호출 시점: WM_INITDIALOG 메시지 발생시
- 용도: 컨트롤을 초기화하거나 키보드 포커스를 변경

```
virtual void CDialog::OnOK ( );
```

- 호출 시점: IDOK 버튼을 누를 때
- 용도: 컨트롤의 값을 읽거나 값의 타당성 여부를 검사한 후 대화상자 종료

모드형 대화상자 (3/4)

- CDialog 클래스 가상 함수 (cont'd)

```
virtual void CDialog::OnCancel ( );
```

- 호출 시점: IDCANCEL 버튼을 누를 때
- 용도: 대화상자 종료

모드형 대화상자 (4/4)

- OnOK(), OnCancel() 함수 내부 구현

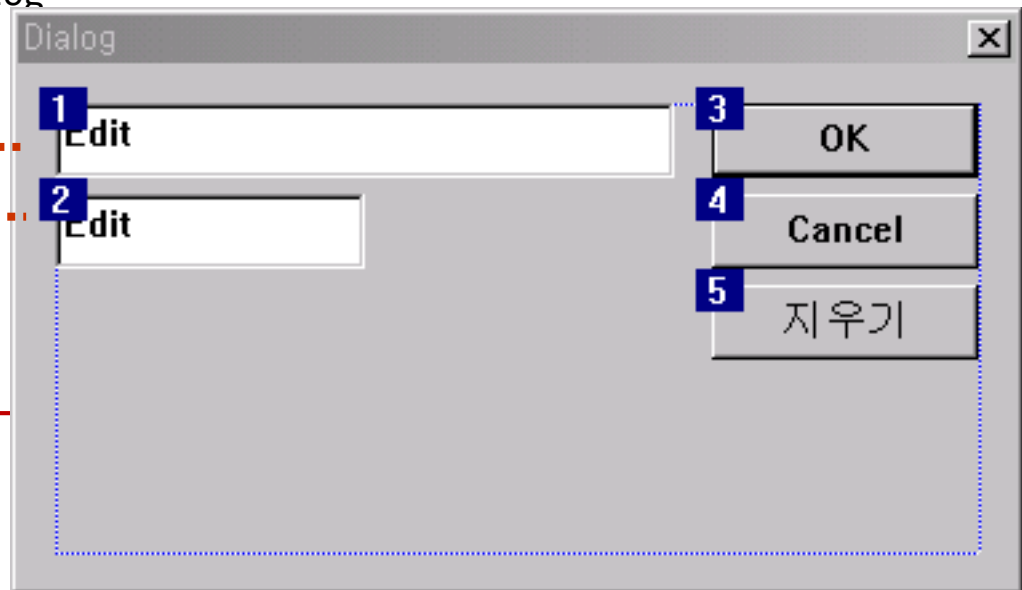
```
void CDialog::OnOK()
{
    UpdateData(TRUE);
    EndDialog(IDOK);
}

void CDialog::OnCancel()
{
    EndDialog(IDCANCEL);
}
```

DDX/DDV (1/8)

- 모드형 대화상자 구현

```
class CMyDialog : public CDialog
{
    ...
    CString m_str;
    int m_color;
    ...
}
```



DDX/DDV (2/8)

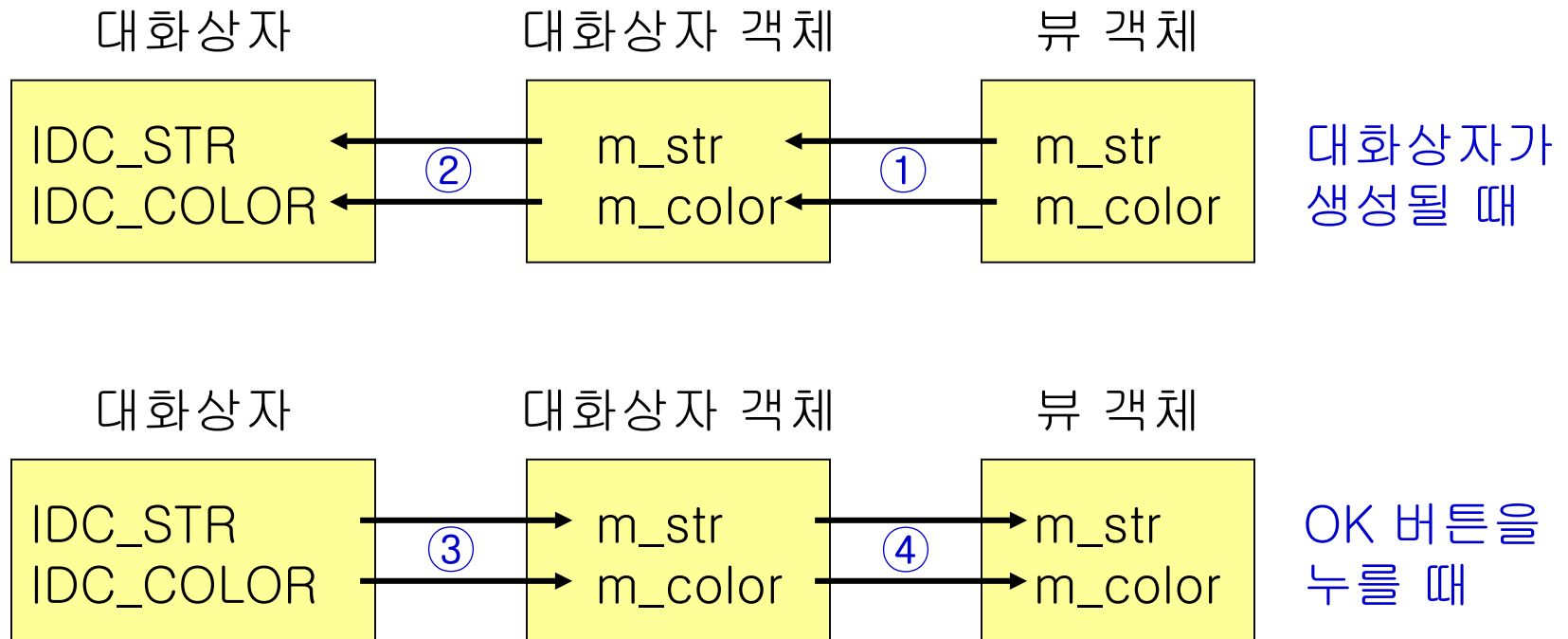
- 모드형 대화상자 구현 (cont'd)

```
BOOL CMyDialog::OnInitDialog()
{
    CDialog::OnInitDialog();
    SetDlgItemText(IDC_STR, m_str);
    SetDlgItemInt(IDC_COLOR, m_color);
    return TRUE;
}

void CMyDialog::OnOK()
{
    GetDlgItemText(IDC_STR, m_str);
    m_color = GetDlgItemInt(IDC_COLOR);
    CDialog::OnOK();
}
```

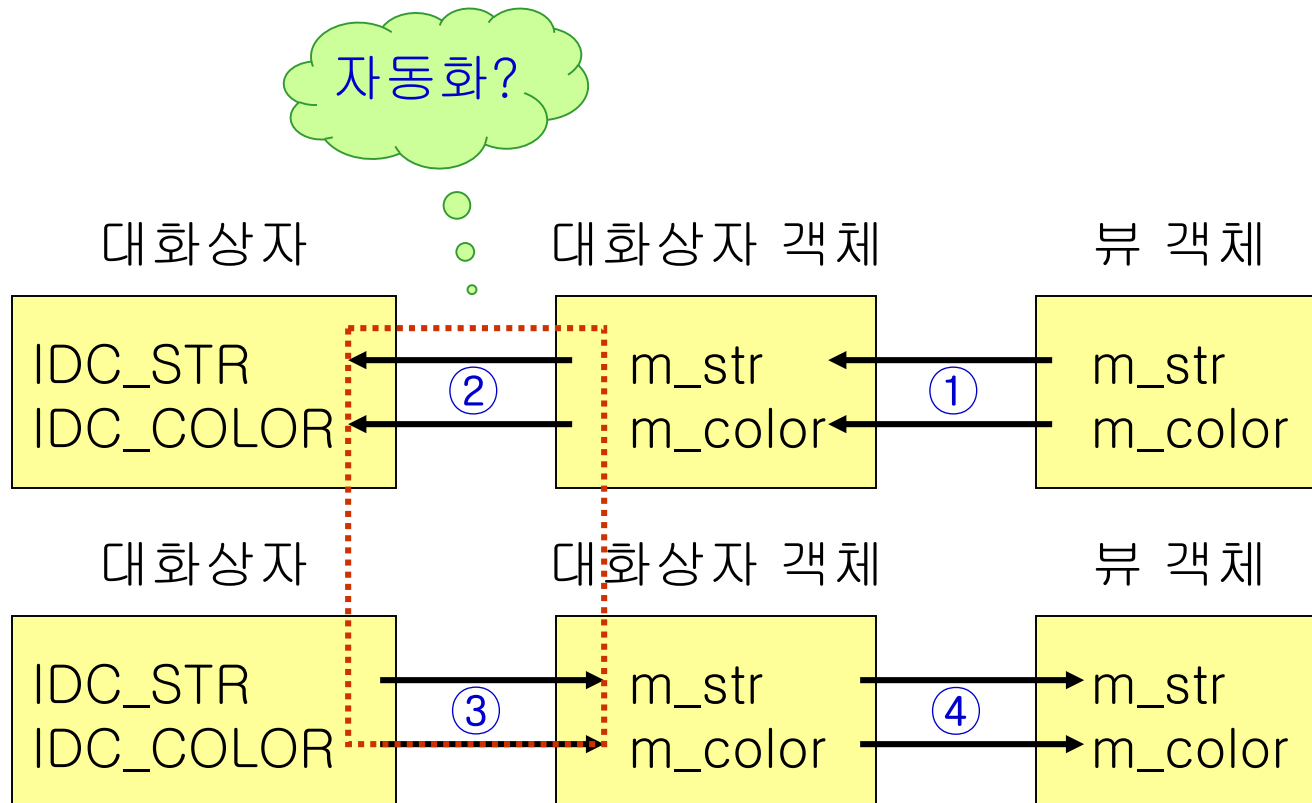
DDX/DDV (3/8)

- 동작 원리



DDX/DDV (4/8)

- DDX(Dialog Data eXchange)



DDX/DDV (5/8)

- OnInitDialog(), OnOK() 함수 내부 구현

```
BOOL CDialog::OnInitDialog()
{
    ...
    UpdateData(FALSE);           // 변수로부터 컨트롤 값 초기화
    ...
}

void CDialog::OnOK()
{
    ...
    UpdateData(TRUE);           // 컨트롤에 입력된 값을
                                // 변수값에 적용
    ...
}
```

DDX/DDV (6/8)

- CWnd::UpdateData() 함수 내부 구현

```
BOOL CWnd::UpdateData(BOOL bSaveAndValidate)
{
    ...
    CDataExchange dx(this, bSaveAndValidate);
    DoDataExchange(&dx);
    ...
}
```

DDX/DDV (7/8)

- DDX 구현

1. 컨트롤의 값을 저장할 변수 선언
2. DoDataExchange() 함수를 자신의 프로그램에 맞게 재정의
 - DDX_* 매크로 사용

```
void CMyDialog::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CMyDialog)
    DDX_Text(pDX, IDC_STR, m_str);
    DDX_Text(pDX, IDC_COLOR, m_color);
    //}}AFX_DATA_MAP
}
```

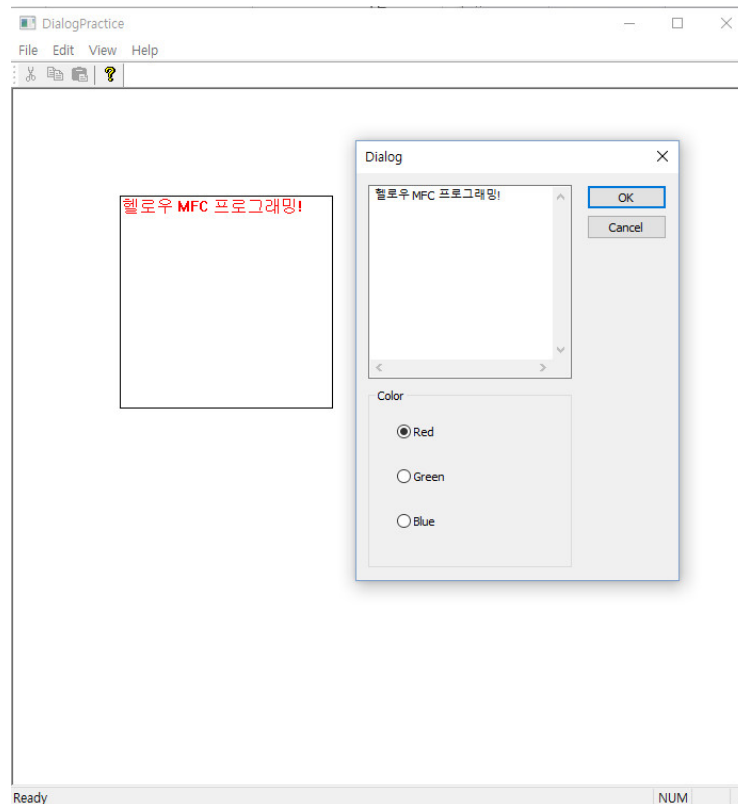
DDX/DDV (8/8)

- DDV(Dialog Data Validation)
 - 대화상자의 컨트롤에 입력한 데이터의 타당성 여부를 자동으로 검사
 - DDV_* 매크로 사용

```
void CMyDialog::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CMyDialog)
    DDX_Text(pDX, IDC_STR, m_str);
    DDV_MaxChars(pDX, m_str, 10);
    DDX_Text(pDX, IDC_COLOR, m_color);
    DDV_MinMaxInt(pDX, m_color, 0, 255);
    //}}AFX_DATA_MAP
}
```

연습: 모드형 대화상자

- 다음과 같이 에디트 박스와 라디오 버튼 컨트롤을 붙여 대화상자를 만들고, 마우스 왼쪽 클릭을 통해 모드형 대화상자로 창의 띄워 이를 이용하여 글을 쓰고 글의 색을 조절할 수 있는 프로그램을 만들어라.



2가지 종류의 대화 상자

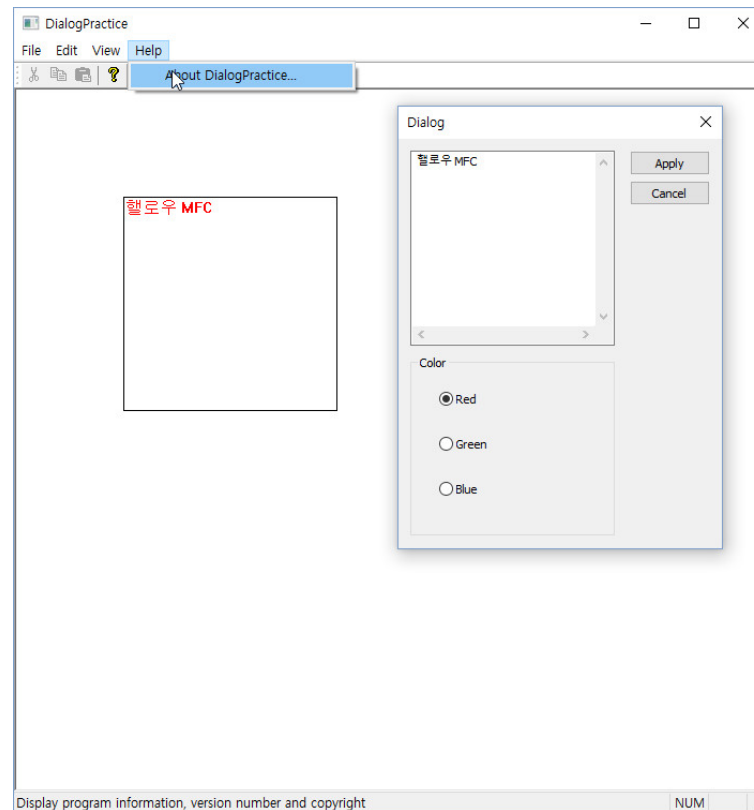
- 모드형 대화상자 (Modal Dialog Box)
 - 대화상자를 닫지 않으면 응용 프로그램이 더 이상 진행할 수 없음
- 비모드형 대화상자 (Modeless Dialog Box)
 - 대화상자를 닫지 않더라도 응용 프로그램이 계속 진행할 수 있음

비 모드형 대화상자

- 같은점:
 - Resource를 만들고 Class를 만드는 방법은 모드형과 동일
- 차이점:
 - `CDialog::DoModal()` 함수 대신 `CDialog::Create()` 함수를 이용하여 생성
 - 대화상자를 닫을 때 `CDialog::EndDialog()` 함수 대신 `CWnd::DestroyWindow()` 함수를 호출
 - 모드형 대화상자 객체는 대개 스택(Stack)에 생성하지만 비 모드형 대화상자 객체는 힙(Heap)에 생성

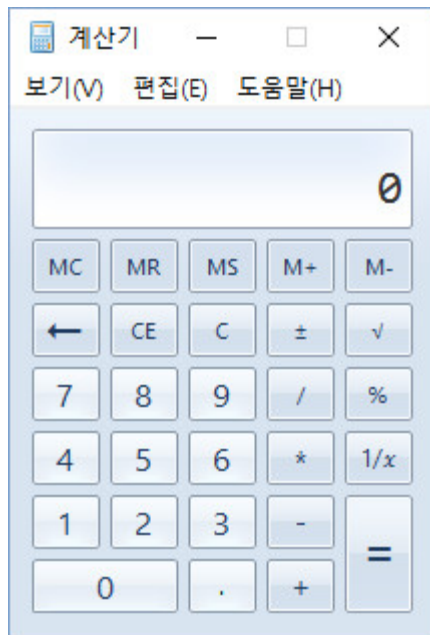
연습: 비모드형 대화상자

- 다음과 같이 에디트 박스와 라디오 버튼 컨트롤을 붙여 대화상자를 만들고, 마우스 오른쪽 클릭을 통해 비모드형 대화상자로 창의 띄워 이를 이용하여 글을 쓰고 글의 색을 조절할 수 있는 프로그램을 만들어라.



대화상자 기반 응용 프로그램 (1/2)

- 대화상자 기반 응용 프로그램
 - 대화상자가 메인 윈도우 역할을 하는 응용 프로그램



대화상자 기반 응용 프로그램 (2/2)

- InitInstance() 함수

```
BOOL CSimpleCalcApp::InitInstance()
{
    CSimpleCalcDlg dlg;
    m_pMainWnd = &dlg;
    int nResponse = dlg.DoModal();
    if (nResponse == IDOK) {

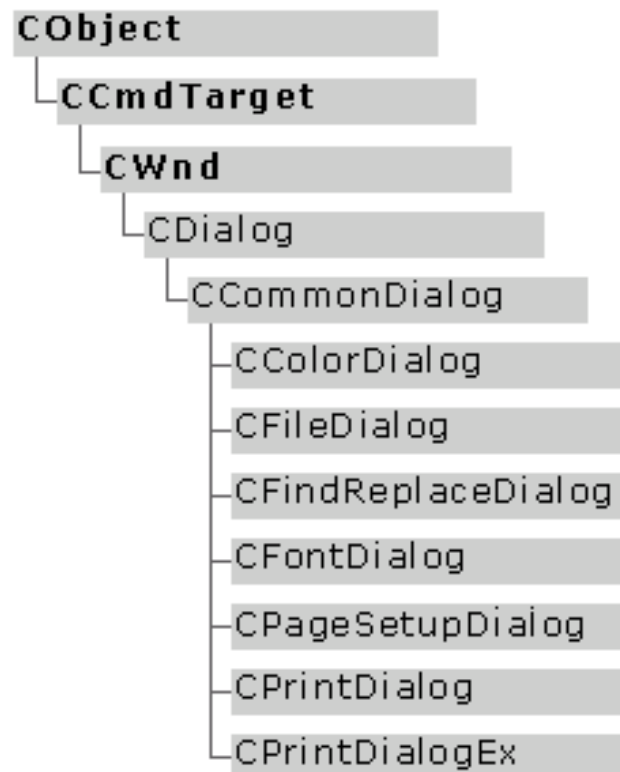
    }
    else if (nResponse == IDCANCEL) {

    }
    return FALSE;
}
```

공통 대화상자 (1/10)

- 공통 대화상자:
 - 응용 프로그램에서 흔히 필요한 대화상자

- MFC 클래스 계층도



공통 대화상자 (2/10)

- MFC 공통 대화상자 클래스

MFC 클래스	용도	API 함수
CColorDialog	색상 선택	ChooseColor
CFileDialog	파일 열기 또는 저장	GetOpenFileName, GetSaveFileName
CFindReplaceDialog	찾기 또는 바꾸기	FindText, ReplaceText
CFontDialog	폰트 선택	ChooseFont
CPageSetupDialog	페이지 설정(페이지 크기, 방향, 페이지 여백 등)	PageSetupDlg
CPrintDialog	인쇄 설정(프린터, 인쇄 범위 등)	PrintDlg
CPrintDialogEx (>MFC 7.0)	인쇄 설정(프린터, 인쇄 범위 등)	PrintDlgEx (>윈도우 2000)

공통 대화상자 (3/10)

- CColorDialog

```
CColorDialog dlg;  
dlg.DoModal();  
COLORREF color = dlg.GetColor();
```



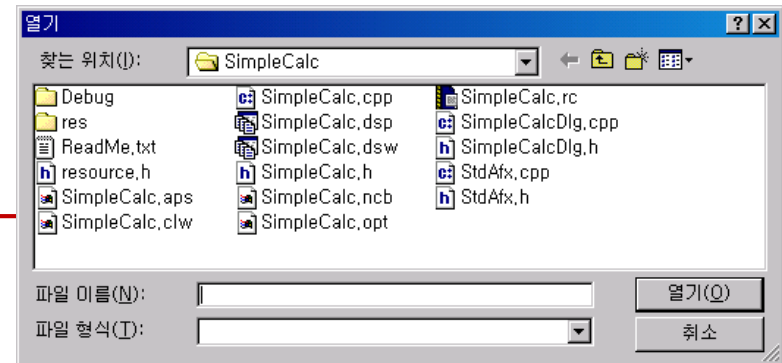
```
CColorDialog dlg(RED, CC_FULLOPEN);  
dlg.DoModal();  
COLORREF color = dlg.GetColor();
```



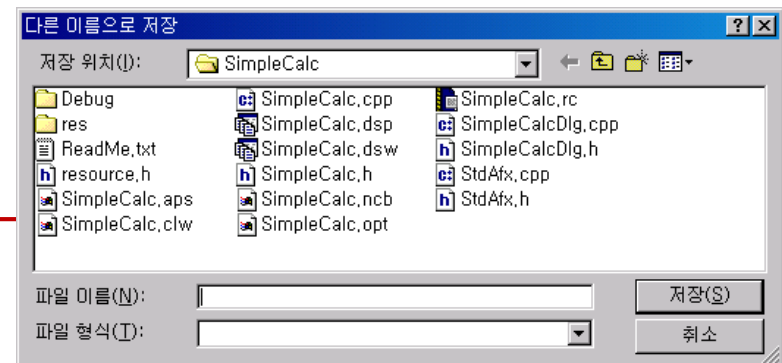
공통 대화상자 (4/10)

- CFileDialog

```
CFileDialog dlg(TRUE);  
if(dlg.DoModal() == IDOK)  
    MessageBox(dlg.GetPathName());
```

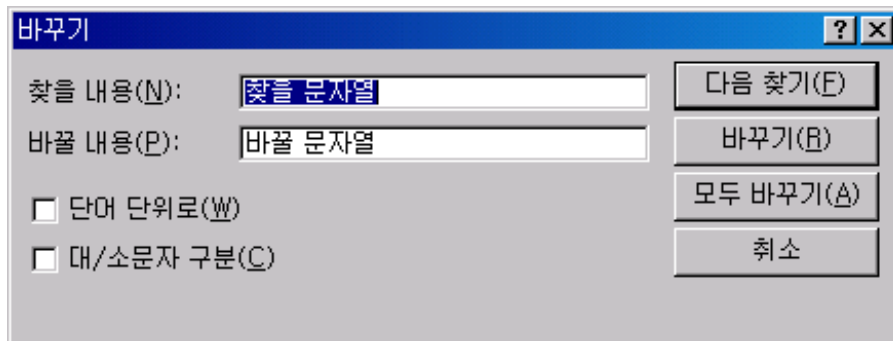
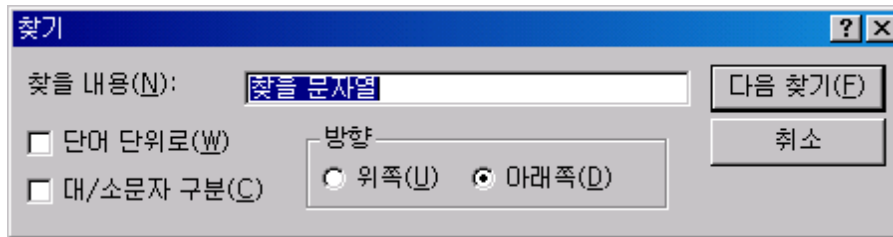


```
CFileDialog dlg(FALSE);  
if(dlg.DoModal() == IDOK)  
    MessageBox(dlg.GetPathName());
```



공통 대화상자 (5/10)

- CFindReplaceDialog
 - 사용 방법은 교재 참조



공통 대화상자 (6/10)

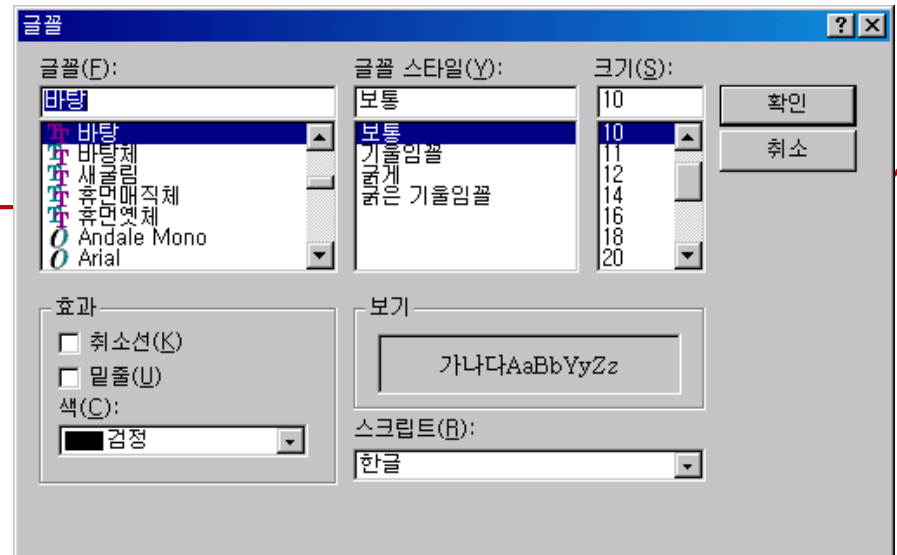
- CFontDialog

```
CFontDialog dlg;  
  
If (dlg.DoModal() == IDOK) {  
    CClientDC dc(this);  
  
    // 화면을 지운다.  
    CRect rect;  
    GetClientRect(&rect);  
    dc.SelectStockObject(WHITE_PEN);  
    dc.SelectStockObject(WHITE_BRUSH);  
    dc.Rectangle(&rect);  
  
    // 선택된 색상을 알아낸다.  
    COLORREF color = dlg.GetColor();  
    dc.SetTextColor(color);  
}
```

공통 대화상자 (7/10)

- CFontDialog (cont'd)

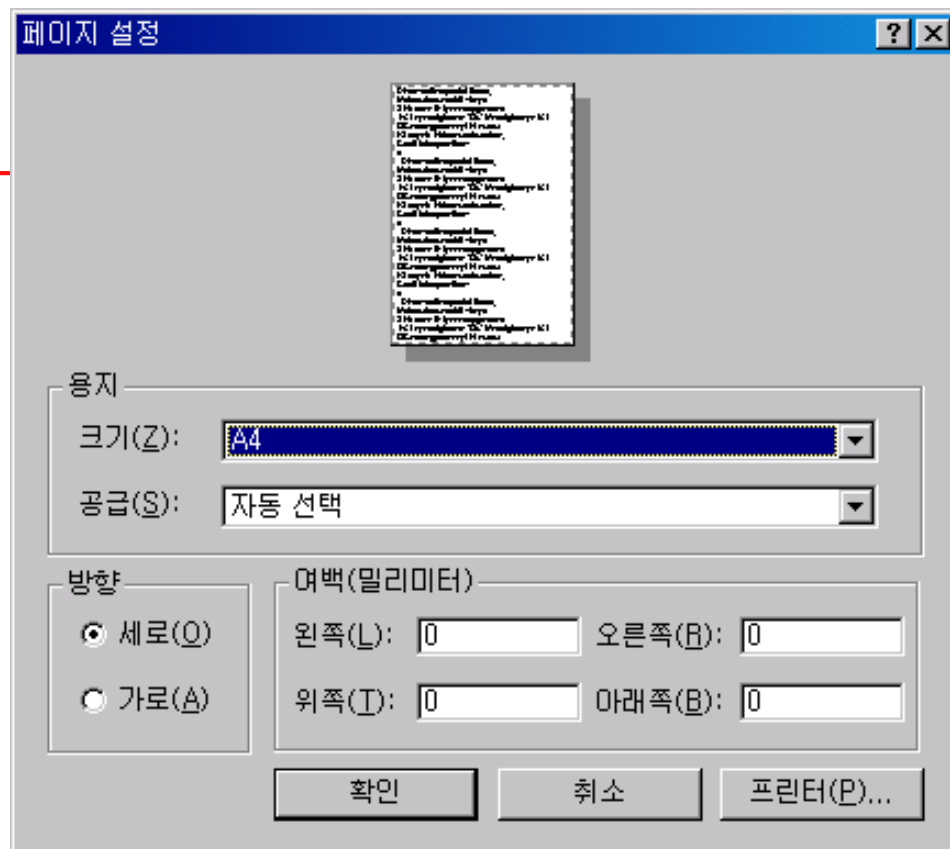
```
// 선택된 폰트를 알아낸다.  
LOGFONT lf;  
dlg.GetCurrentFont(&lf);  
CFont font;  
font.CreateFontIndirect(&lf);  
dc.SelectObject(&font);  
  
// 텍스트를 출력한다.  
dc.TextOut(10, 10,  
    CString("한글 & English"));  
}
```



공통 대화상자 (8/10)

- CPageSetupDialog

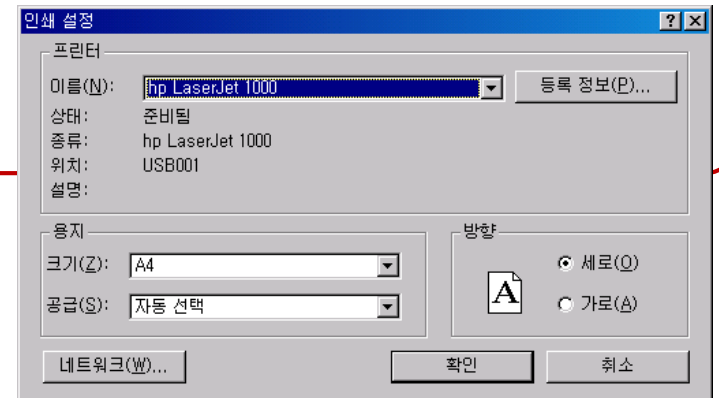
```
CPageSetupDialog dlg;  
dlg.DoModal();
```



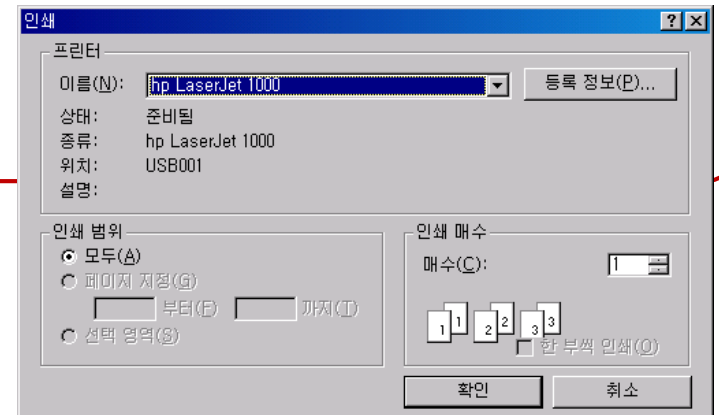
공통 대화상자 (9/10)

- CPrintDialog

```
CPrintDialog dlg(TRUE);  
dlg.DoModal();
```



```
CPrintDialog dlg(FALSE);  
dlg.DoModal();
```

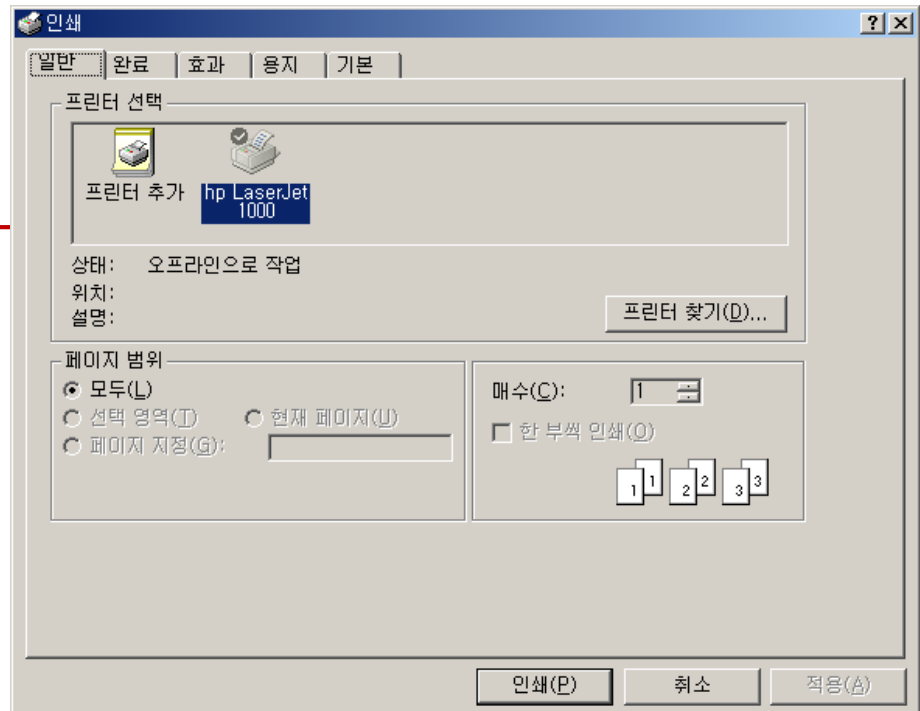


공통 대화상자 (10/10)

- CPrintDialogEx

```
// stdafx.h 파일에서 WINVER 상수값을 0x0500 이상으로 정의한다.  
// (예) #define WINVER 0x0500
```

```
CPrintDialogEx dlg;  
dlg.DoModal();
```



Q & A