

# 대화상자

(1주차)

# 학습개요

- 학습 목표

- 대화상자 디자인 방법을 익힌다.
- 모드형 대화상자와 비모드형 대화상자의 차이를 이해하고 작성 방법을 익힌다.
- 대화상자 기반 응용프로그램의 동작 원리를 이해하고 작성 방법을 익힌다.
- 공통대화상자를 다루는 방법을 배운다.

- 학습 내용

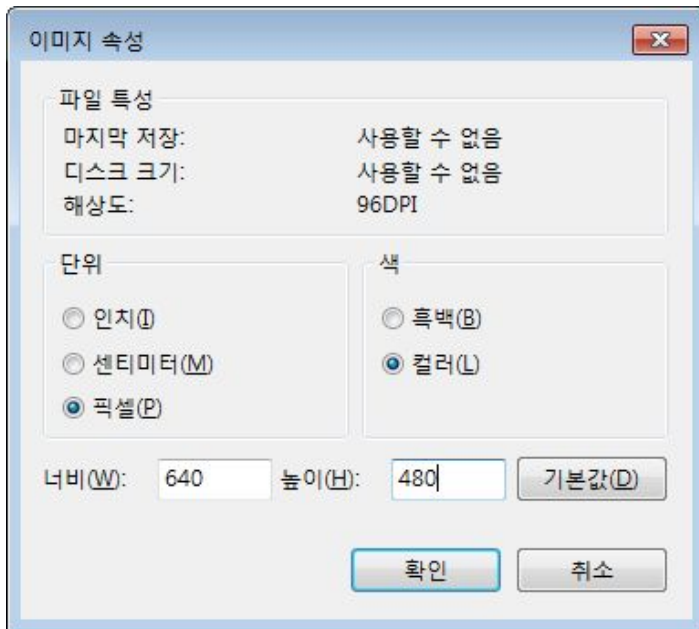
- 대화상자 기초
- 모드형 대화상자
- 비모드형 대화상자
- 대화상자 기반 응용프로그램
- 공용대화상자
- 실습

# 대화상자 기초

# 개요

- 대화상자

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



# 개요

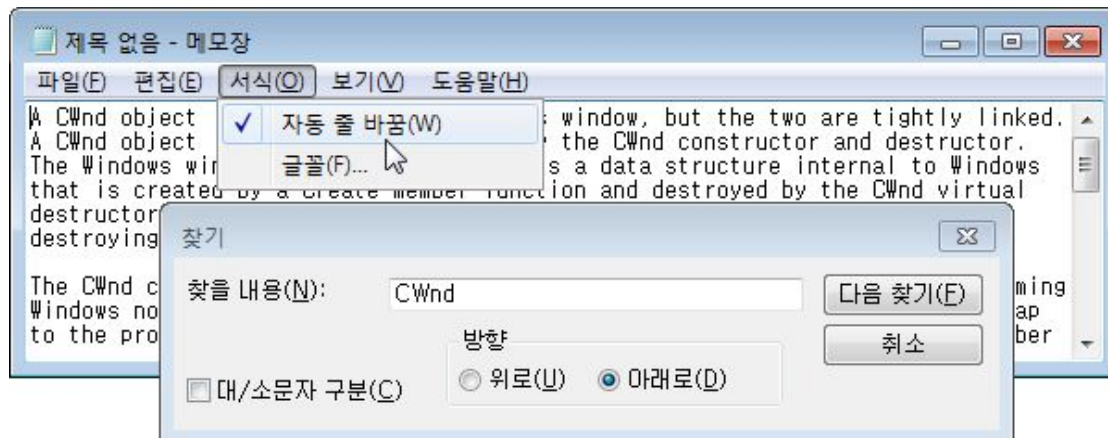
- 대화상자 종류

- 모드형 대화상자

- 대화상자를 닫아야 응용 프로그램이 다른 작업 가능

- 비모드형 대화상자

- 대화상자를 닫지 않아도 응용 프로그램이 다른 작업 가능

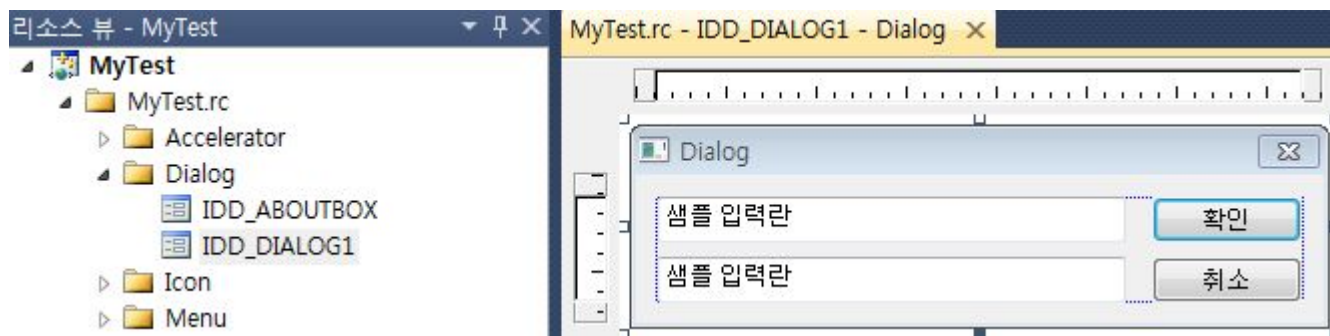


# 대화상자 템플릿

- 대화상자 템플릿

- 대화상자 자체와 대화상자에 포함된 컨트롤에 대한 모든 정보를 가진 이진(Binary) 데이터
- 실행 파일에 포함된 대화상자 리소스를 운영체제가 로드하여 대화상자 템플릿 생성

- 대화 상자 디자인



# 대화상자 템플릿

- 대화상자 템플릿 작성

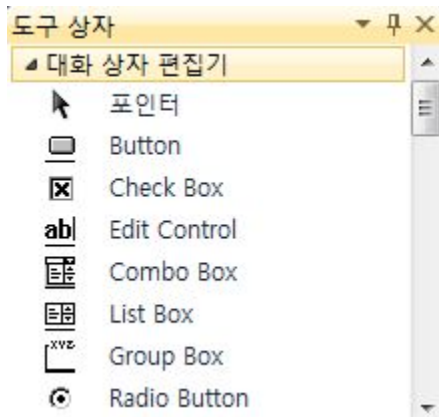
- 비주얼 C++ 리소스 편집기를 이용하여 리소스 스크립트(\*.RC) 작성
- 리소스 컴파일러로 이진 파일(\*.RES) 생성
- 링크 단계에서 실행 파일에 포함

- 리소스 스크립트 작성

```
IDD_DIALOG1 DIALOGEX 0, 0, 229, 46
STYLE DS_SETFONT | DS_MODALFRAME | DS_FIXEDSYS | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Dialog"
FONT 8, "MS Shell Dlg", 400, 0, 0x1
BEGIN
    DEFPUSHBUTTON "확인", IDOK, 172, 7, 50, 14
    PUSHBUTTON "취소", IDCANCEL, 172, 25, 50, 14
    EDITTEXT IDC_EDIT1, 7, 7, 156, 14, ES_AUTOHSCROLL
    EDITTEXT IDC_EDIT2, 7, 25, 156, 14, ES_AUTOHSCROLL
END
```

# 대화상자 편집기

- 컨트롤 추가와 삭제
  - 도구 상자
    - 마우스 클릭으로 추가, [Delete] 키로 삭제
    - 같은 종류를 계속 추가하고 싶으면 [Ctrl] 키를 누른 채 컨트롤 클릭

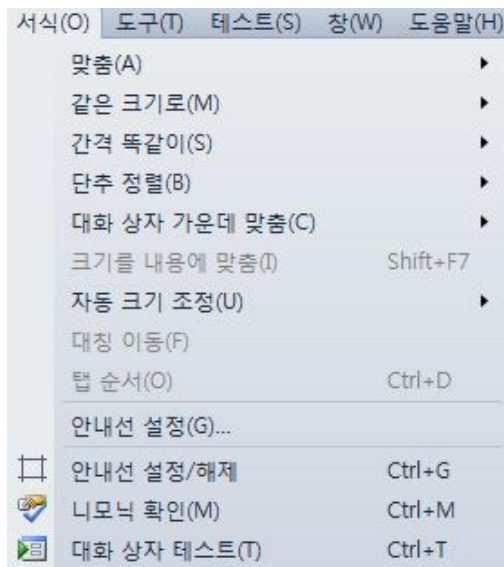




# 대화상자 편집기

- 컨트롤 배치

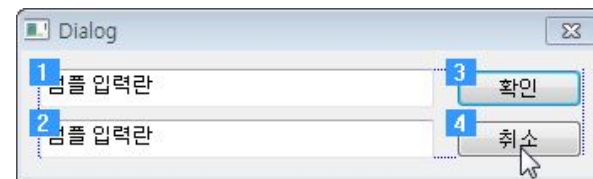
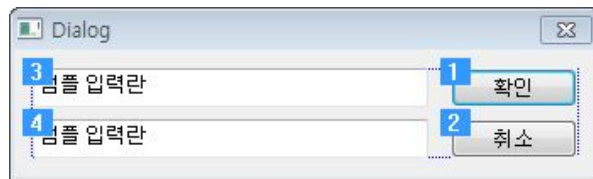
- 서식 메뉴 또는 대화 상자 편집기 툴바 이용
- 기준 컨트롤은 [Ctrl] 키를 누른 상태에서 원하는 컨트롤을 클릭하면 기준이 변경됨



# 대화상자 편집기

- 탭 순서

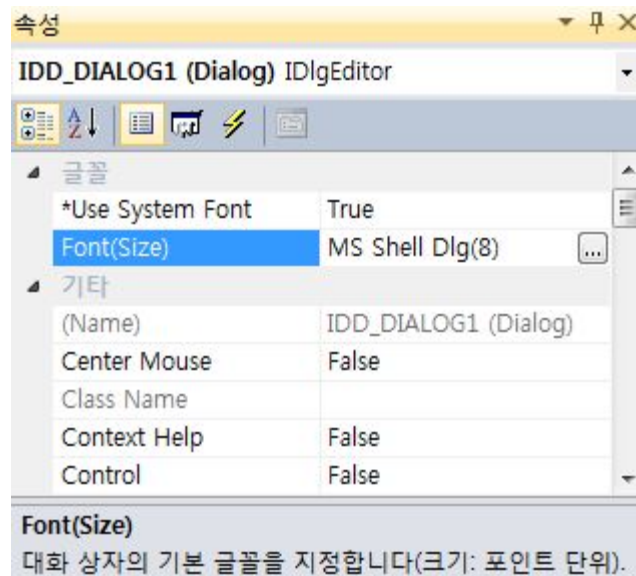
- Tab 키를 눌렀을 때 키보드 포커스가 이동하는 순서
- [서식] - [탭 순서] 메뉴를 이용하여 변경
- 마우스로 클릭하는 순서대로 재설정
- 탭 순서 1번에 해당하는 컨트롤은 대화 상자 생성시 자동으로 키보드 포커스를 갖게 됨



# 대화상자 편집기

- 대화상자 속성

- 폰트를 변경하면 대화상자 자체의 크기는 물론이고 대화상자에 포함된 컨트롤의 크기도 그에 비례해서 변경됨



모드형 대화상자

# 모드형 대화상자

- MFC 클래스 계층도



- 모드형 대화상자 생성 순서

- ① 대화상자 리소스 작성
- ② CDialog 클래스 또는 CDialog 파생 클래스 객체 생성
  - 이때 대화상자 리소스 ID를 생성자의 인자로 넘겨준다.
- ③ CDialog::DoModal() 함수 호출

# 모드형 대화상자

- CDialog 클래스 가상 함수

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

- 호출 시점: WM\_INITDIALOG 메시지 발생 시
- 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 닫기

# 모드형 대화상자

- CDialog 클래스 가상 함수

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

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

# 모드형 대화상자

- OnOK(), OnCancel() 함수의 MFC 내부 구현

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



# 모드형 대화상자

- 모드형 대화상자 구현 (1/2)

```
class CMyDialog : public CDialog
{
    ...
public:
    CComboBox m_combo;
    CString m_str; // 사용자가 입력한 텍스트 저장
    int m_font; // 사용자가 선택한 폰트 번호 저장
    ...
};
```

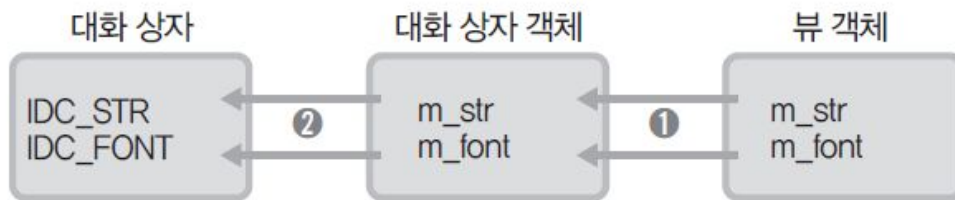
# 모드형 대화상자

- 모드형 대화상자 구현 (2/2)

```
BOOL CMyDialog::OnInitDialog()  
{  
    CDialog::OnInitDialog();  
    SetDlgItemText(IDC_STR, m_str);  
    m_combo.SetCurSel(m_font);  
    return TRUE;  
}  
  
void CMyDialog::OnOK()  
{  
    GetDlgItemText(IDC_STR, m_str);  
    m_font = m_combo.GetCurSel();  
    CDialog::OnOK();  
}
```

# DDX/DDV

- 데이터 흐름도와 관련 함수



(a) 대화 상자가 생성될 때

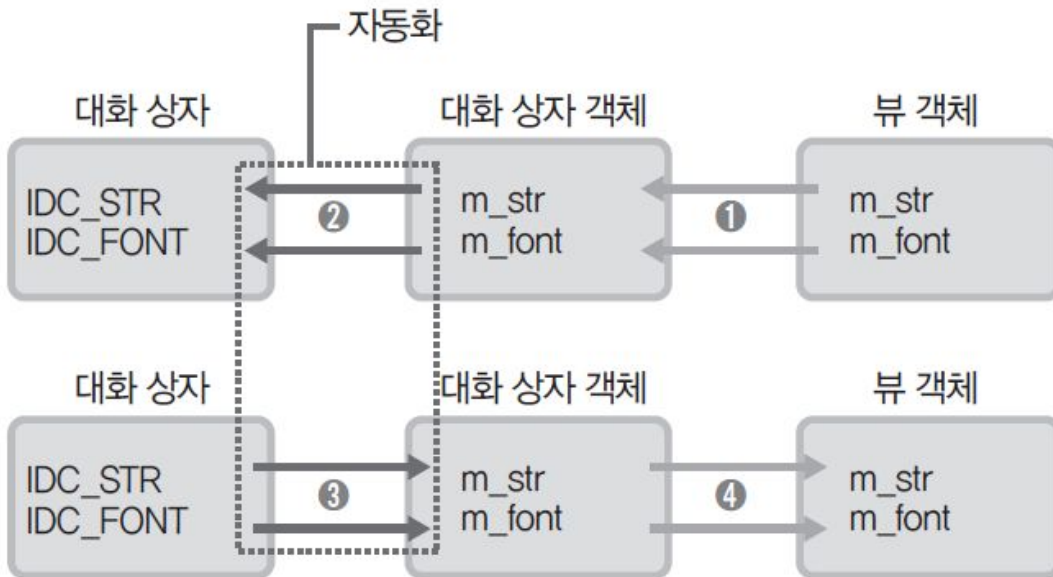


(b) <확인> 버튼을 누를 때

- 1 CModalDialog2View::OnLButtonDown()
- 2 CMyDialog::OnInitDialog()
- 3 CMyDialog::OnOK()
- 4 CModalDialog2View::OnLButtonDown()

# DDX/DDV

- DDX(Dialog Data eXchange)



# DDX/DDV

- OnInitDialog(), OnOK() 함수의 MFC 내부 구현

```
BOOL CDialog::OnInitDialog()  
{  
    ...  
    UpdateData(FALSE);  
    ...  
}  
  
void CDialog::OnOK()  
{  
    ...  
    UpdateData(TRUE);  
    ...  
}
```

# DDX/DDV

- CWnd::UpdateData() 함수의 MFC 내부 구현

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

# DDX/DDV

- DDX 구현

- 컨트롤의 값을 저장할 변수를 선언한다.
- DoDataExchange() 함수를 자신의 프로그램에 맞게 재정의한다.
  - DDX\_\* 형태의 MFC 전역 함수 사용

```
void CMyDialog::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_STR, m_str);
    DDX_CBIndex(pDX, IDC_FONT, m_font);
}
```

# DDX/DDV

- DDX(Dialog Data Validation)

- 대화상자의 컨트롤에 입력한 데이터의 타당성(텍스트 길이, 값의 범위, 날짜 범위 등)을 자동으로 검사
  - DDX\_\* 형태의 MFC 전역 함수 사용

```
void CMyDialog::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_STR, m_str);
    DDX_MaxChars(pDX, m_str, 10);
    DDX_CBIndex(pDX, IDC_FONT, m_font);
}
```



실습

비모드형 대화상자

# 비모드형 대화상자

- 작성 시 주의 사항

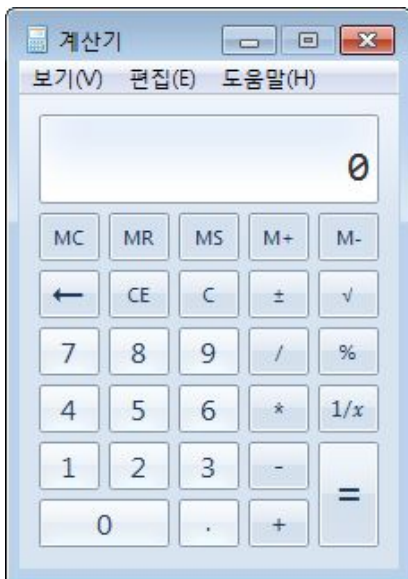
- CDialog::DoModal() 대신 CDialog::Create() 함수를 이용하여 생성한다.
- 대화상자를 닫을 때 CDialog::EndDialog() 대신 CWnd::DestroyWindow() 함수를 호출한다.
- 모드형 대화상자 객체는 대개 스택에 생성하지만 비모드형 대화상자 객체는 힙에 생성한다.

실습

대화상자 기반 응용프로그램

# 대화상자 기반 응용프로그램

- 대화상자 기반 응용프로그램
  - 대화상자가 메인윈도우 역할을 하는 응용프로그램



# 대화상자 기반 응용프로그램

- 대화상자 기반 응용프로그램
  - 응용프로그램 마법사로 프로젝트 생성 시 '대화상자 기반' 선택



# 대화상자 기반 응용프로그램

- InitInstance() 함수

```
BOOL CDialogBaseApp::InitInstance()
{
    ...
    CDialogBaseDlg dlg; // 대화상자 객체를 생성한다.
    m_pMainWnd = &dlg; // 대화상자 객체의 주소를 저장해 둔다.
    INT_PTR nResponse = dlg.DoModal(); // 모드형 대화상자를 생성한다.
    if(nResponse == IDOK)
    {
        // <확인> 버튼을 누른 경우를 처리한다.
    }
    else if(nResponse == IDCANCEL)
    {
        // <취소> 버튼을 누른 경우를 처리한다.
    }
    ...
    return FALSE; // 응용 프로그램을 종료한다.
}
```

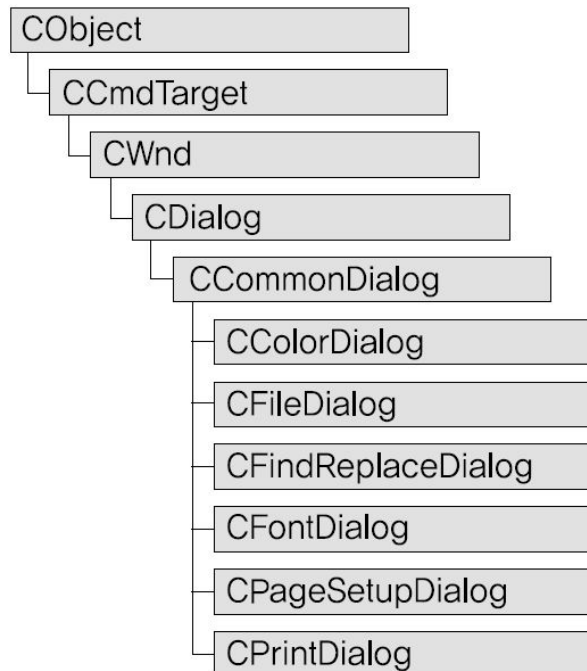


실습

공용대화상자

# 공용대화상자

- MFC 클래스 계층도



# 공용대화상자

- MFC 공용대화상자 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, 여백 등)	PageSetupDlg()
CPrintDialog	인쇄 설정 (프린터, 인쇄 범위 등)	PrintDlg()

# 공용대화상자

## • CColorDialog

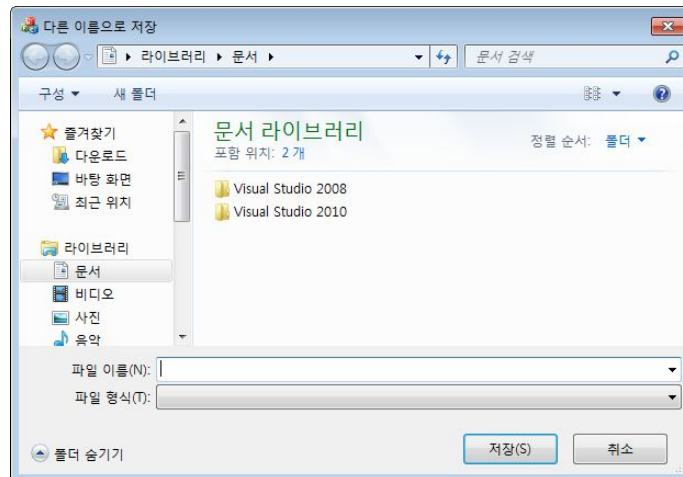
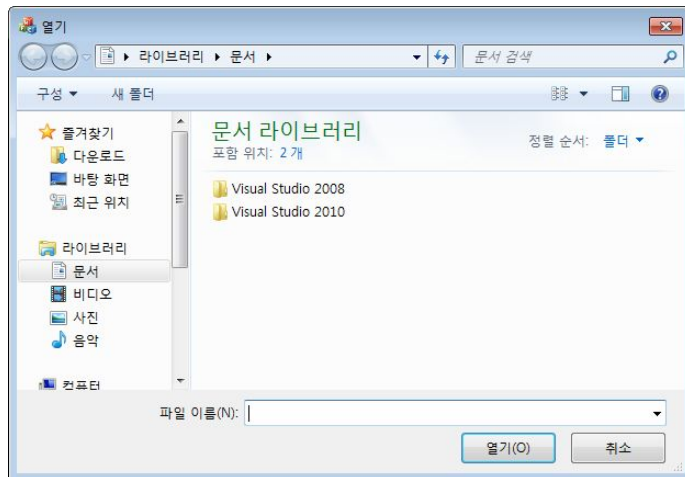
- ❶ CColorDialog dlg;  
dlg.DoModal();  
COLORREF color = dlg.GetColor(); // 사용자가 선택한 색상을 알아낸다.
- ❷ CColorDialog dlg(RGB(255, 0, 0), CC\_FULLOPEN);  
dlg.DoModal();  
COLORREF color = dlg.GetColor(); // 사용자가 선택한 색상을 알아낸다.



# 공용대화상자

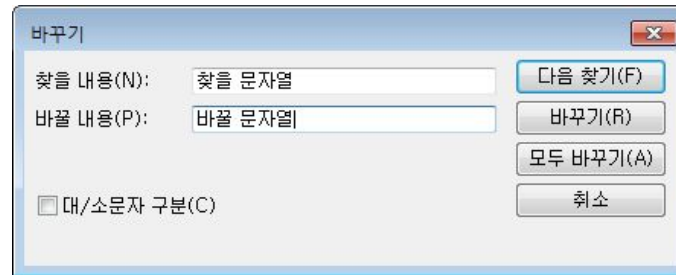
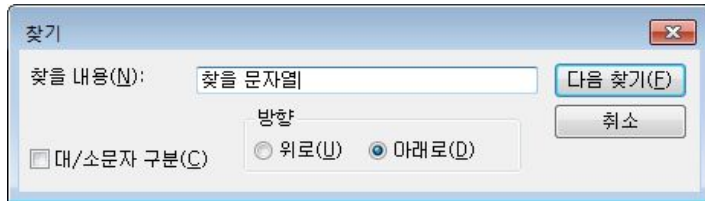
## • CFileDialog

- 1 CFileDialog dlg(TRUE);  
if(dlg.DoModal() == IDOK)  
    MessageBox(dlg.GetPathName());
- 2 CFileDialog dlg(FALSE);  
if(dlg.DoModal() == IDOK)  
    MessageBox(dlg.GetPathName());



# 공용대화상자

- CFindReplaceDialog
  - 다른 대화상자와 달리 비모드형으로 동작



# 공용대화상자

- CFontDialog (1/2)
  - 사용자가 선택한 폰트를 이용해 텍스트 출력

```
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);
```



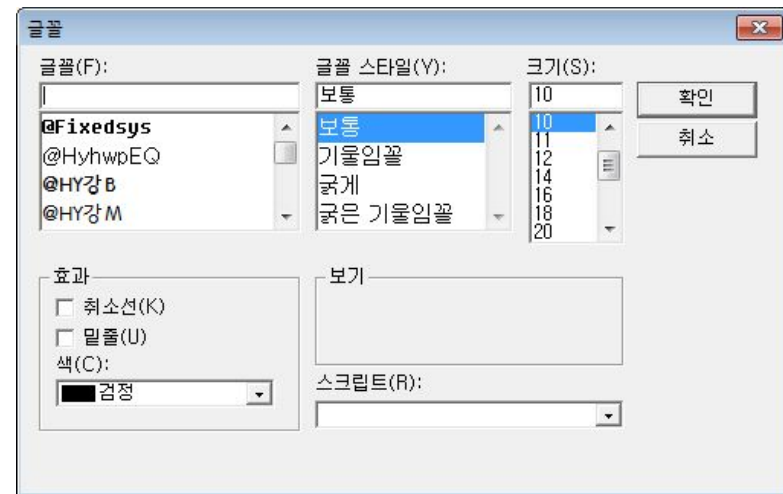
# 공용대화상자

## • CFontDialog (2/2)

```
// 선택된 색상을 알아낸다.  
COLORREF color = dlg.GetColor();  
dc.SetTextColor(color);
```

```
// 선택된 폰트를 알아낸다.  
LOGFONT lf;  
dlg.GetCurrentFont(&lf);  
CFont font;  
font.CreateFontIndirect(&lf);  
dc.SelectObject(&font);
```

```
// 텍스트를 출력한다.  
dc.TextOut(10, 10, CString(_T("한글 & English")));
```

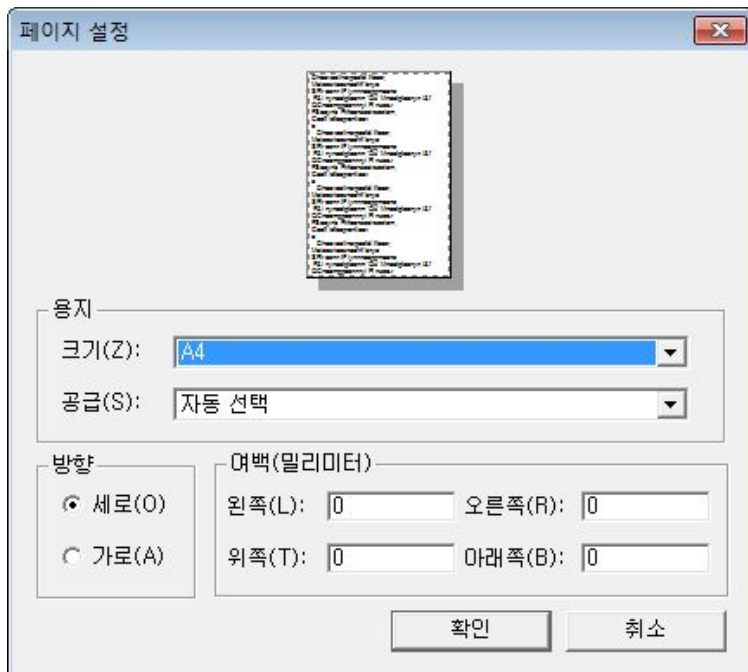


```
}
```

# 공용대화상자

- CPageSetupDialog

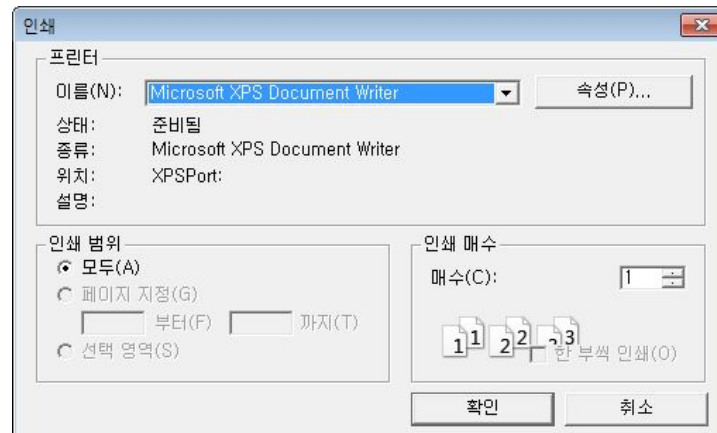
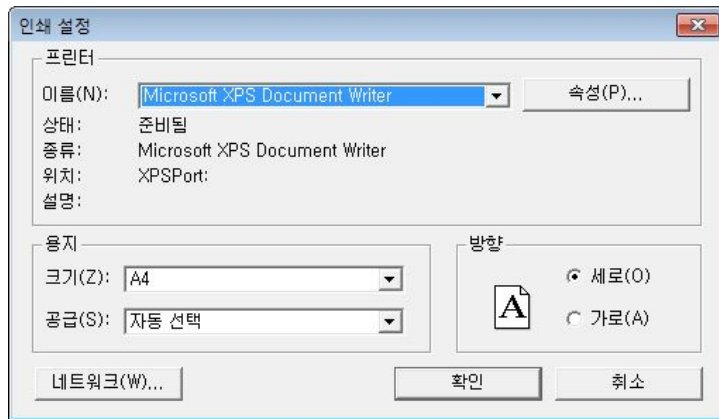
```
CPageSetupDialog dlg;  
dlg.DoModal();
```



# 공용대화상자

- CPrintDialog

- ❶ CPrintDialog dlg(TRUE);  
dlg.DoModal();
- ❷ CPrintDialog dlg(FALSE);  
dlg.DoModal();



# 학습정리

- 대화상자는 다양한 컨트롤을 포함하고 있는 일종의 윈도우로 사용자로부터 입력을 받거나 정보를 출력하는 용도로 사용한다.
- 대화상자는 대화상자를 닫아야 응용 프로그램이 다른 작업 가능한 모드형 대화상자와 대화상자를 닫지 않아도 응용 프로그램이 다른 작업 가능한 비모드형 대화상자로 구분된다.
- 대화상자 템플릿은 대화상자 자체와 대화상자에 포함된 컨트롤에 대한 모든 정보를 가진 이진 데이터이다.
- 모드형 대화상자는 대화상자 리소스 작성 -> CDialog 클래스 또는 CDialog 파생 클래스 객체 생성-> CDialog::DoModal() 함수 호출 절차를 통해 생성된다.
- DDX는 컨트롤과 멤버 변수 간의 데이터 교환을 자동화한다.
- DDV는 컨트롤에 입력한 데이터의 타당성을 검사한다.
- 비모드형 대화상자는 CDialog::Create() 함수를 이용하여 생성하며, 대화상자를 닫을 때 CWnd::DestroyWindow() 함수를 호출한다
- 대화상자가 메인윈도우 역할을 하는 응용프로그램을 대화상자 기반 응용프로그램이라고 부른다.
- 공용대화상자는 운영체제 수준에서 제공하는 대화상자로, CCommonDialog의 파생클래스로 제공된다.