

대화 상자

HCI Programming 2 (321190)
2008년 가을학기
11/18/2008
박경신

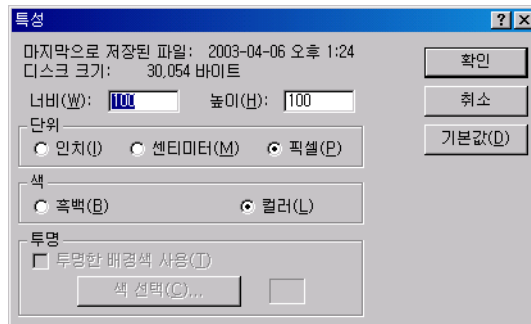
Overview

- 대화상자, 대화상자 템플릿, 대화상자 편집기
- 모드형 대화상자와 비모드형 대화상자의 차이
- 대화상자 기반 응용 프로그램
- 공통 대화상자

대화상자

□ 대화상자 (Dialog Box)

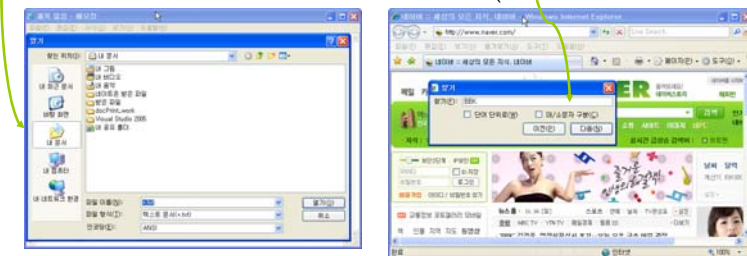
- 다양한 컨트롤을 포함하고 있는 일종의 윈도우
- 사용자로부터 입력을 받거나 정보를 출력
- 정적, 버튼, 편집 등 다양한 컨트롤들을 배치하고 관리하는 윈도우



대화상자 종류

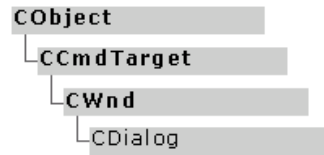
□ 대화상자 종류

- 윈도우 스타일
 - Overlapped, Pop-up, Child
- 동작방식
 - 모드형 대화상자 (Modal Dialog Box) - 대화상자를 닫지 않으면 응용 프로그램이 더 이상 진행할 수 없다.
 - 비 모드형 대화상자 (Modeless Dialog Box) - 대화상자를 닫지 않더라도 응용 프로그램이 계속 진행할 수 있다.
- 메시지 박스, 윈도우 공통 대화상자 (Window Common Dialog Box)



CDialog 클래스

- CDialog 클래스의 MFC 클래스 계층도

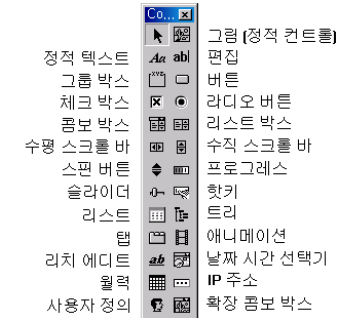
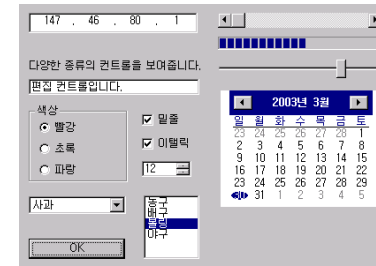


- 대화상자는 CDialog 클래스로 정의
- 모든 윈도우 메시지 처리 가능

5

Controls

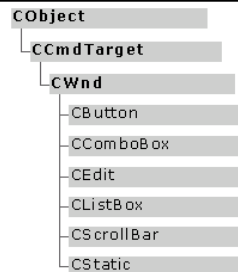
- 표준화된 형태와 특성을 가진 윈도우
- 사용자에게 입력을 받거나 정보를 보여주는 기능
- 다양한 종류의 컨트롤



6

Controls

- 표준 컨트롤 클래스



- 통지 메시지(Notification Message)

- 컨트롤에 발생한 이벤트를 부모윈도우로 전달하는 메시지
- 컨트롤의 상태가 변화되었음을 알려주는 메시지
- 종류
 - 버튼 컨트롤: BN_CLICKED, BN_DBLCLK
 - 편집 컨트롤: EN_CHANGE, EN_SETFOCUS, EN_KILLFOCUS
 - 콤보 박스 컨트롤: CBN_SELCHANGE
 - 리스트 박스 컨트롤: LBN_SELCHANGE

7

대화상자 템플릿

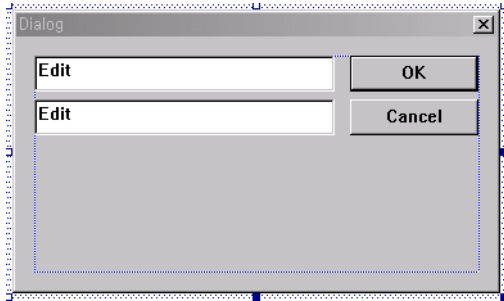
- 대화상자 템플릿 (Dialog Box Template)
 - 대화상자 자체와 포함된 컨트롤에 대한 모든 정보를 가지고 있는 이진 (Binary) 데이터
 - 리소스로 실행 파일에 포함
- 대화상자 템플릿 작성
 - 비주얼 C++ 리소스 편집기나 텍스트 편집기를 이용하여 리소스 스크립트(*.RC) 작성
 - 리소스 컴파일러로 이진 파일(*.RES) 파일 생성
 - 링크 과정에서 실행 파일에 포함

8

대화상자 템플릿

리소스 스크립트 작성

- Resource View에서 Insert Dialog를 사용하여 새로운 대화상자 생성
- 대화상자 속성 설정
- 필요한 컨트롤을 배치하고 속성 및 Tab순서 설정



9

대화상자 템플릿

리소스 스크립트

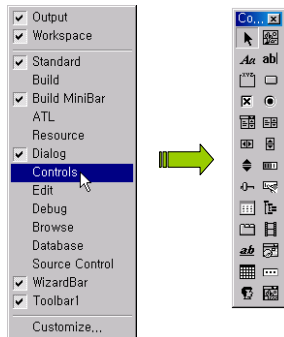
```
IDD_DIALOG1 DIALOG DISCARDABLE 0, 0, 187, 98
STYLE
    DS_MODALFRAME|WS_POPUP|WS_CAPTION|WS_SYSMENU
CAPTION "Dialog"
FONT 10, "System"
BEGIN
    DEFPUSHBUTTON "OK",IDOK,130,7,50,14
    PUSHBUTTON "Cancel",IDCANCEL,130,24,50,14
    EDITTEXT IDC_EDIT1,7,7,117,14,ES_AUTOHSCROLL
    EDITTEXT IDC_EDIT2,7,24,117,14,ES_AUTOHSCROLL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

10

대화상자 편집기

컨트롤 추가와 삭제

- 컨트롤(Control) 툴바 이용

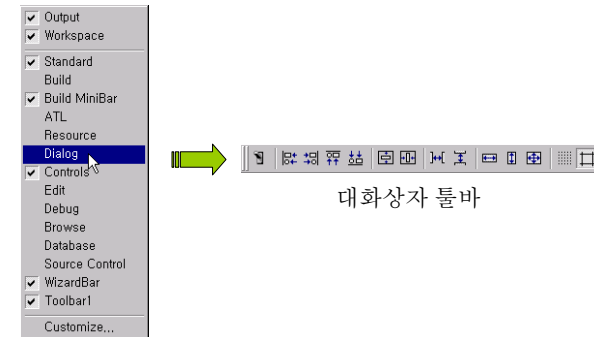


11

대화상자 편집기

컨트롤 배치

- 대화상자(Dialog) 툴바 이용
- 컨트롤 크기, 정렬, 간격 등을 편리하게 설정

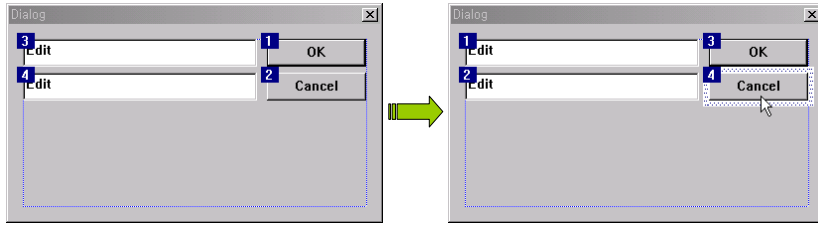


12

대화상자 편집기

□ 탭 순서

- Tab 키를 눌렀을 때 키보드 포커스가 이동하는 순서
- Format->Tab Order (Ctrl+D) 메뉴를 이용하여 변경
- 원하는 탭 순서에 따라 컨트롤들을 클릭



13

대화상자 편집기

□ 탭 순서 변경 후 - 리소스 스크립트

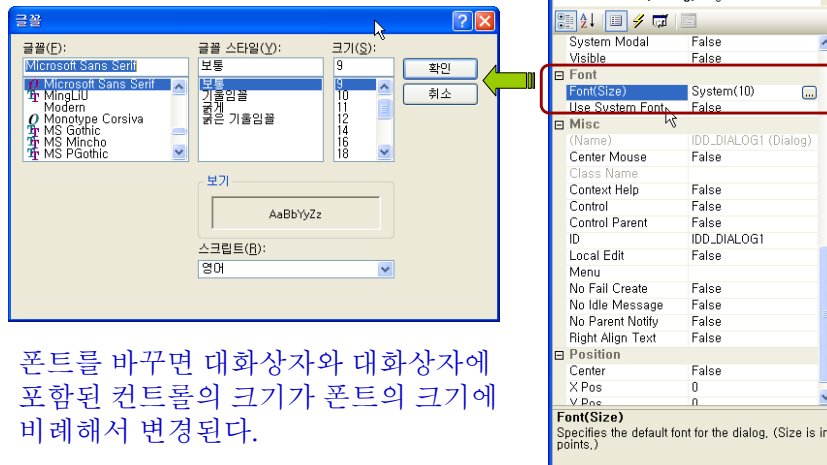
```

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

14

대화상자 편집기

□ 대화상자 속성



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

모드형 대화상자

□ 모드형 (Modal) 대화상자 작성 순서

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

16

모드형 대화상자 생성

□ 모달(Modal) 대화상자

- 대화상자가 출력되어 있는 동안 제어권을 독점하고 있는 형태
- 대화상자 생성 : `CDialog::DoModal()`
- 대화상자 종료 : `CDialog::EndDialog(int)`
 - 확인을 버튼을 누르면 `EndDialog(IDOK)`이 호출되어 대화상자가 종료되면서 `IDOK` 반환
 - 취소버튼을 누르면 `EndDialog(IDCANCEL)`이 호출되어 대화상자가 종료되면서 `IDCANCEL` 반환
- 대화상자 객체는 대화상자가 필요한 함수에 지역변수로 선언하여 모달 대화상자 생성함수 호출

```

CExDialog dlg; //대화상자 객체변수 선언
int nResponse = dlg.DoModal(); //모달 대화상자 생성
if (nResponse == IDOK) { ... } //확인버튼 처리
else if (nResponse == IDCANCEL) { ... } //취소버튼 처리
    
```

모드형 대화상자 예제 작성

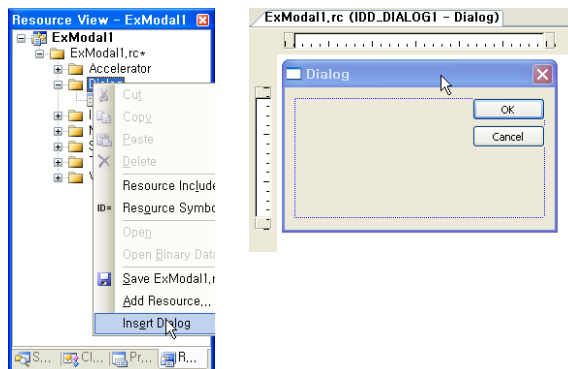
- 프로젝트 생성 - 기본 대화상자 작성법 예제
- 1~6단계 옵션 설정

단계	변경 사항
1	'Single document'를 선택한다.
2	Compound Document Support 변경 사항 없음
3	DB 변경 사항 없음
4	UI 변경 사항 없음
5	Advanced Features 'ActiveX Controls' 선택을 해제함
6	Generated Classes 변경 사항 없음

18

모드형 대화상자 예제 작성

- Resource View에서 Insert Dialog를 해서 `IDD_DIALOG1` 대화상자를 생성한다



19

모드형 대화상자 예제 작성

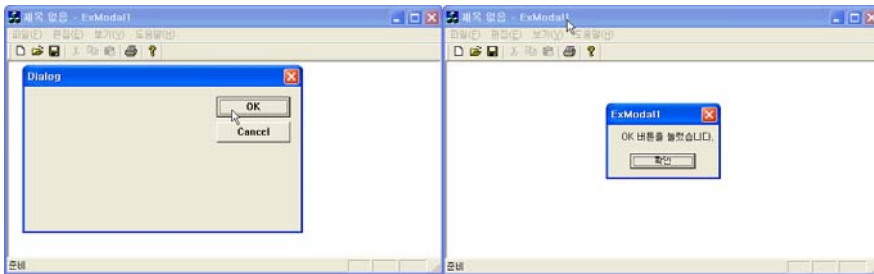
```

void CExModal1View::OnLButtonDbClick(UINT nFlags, Cpoint point) {
    CView::OnLButtonDbClick(nFlags, point);
    CDialog dlg(IDD_DIALOG1); // 대화상자 리소스 ID를 준다
    if (dlg.DoModal() == IDOK)
        MessageBox("OK 버튼을 눌렀습니다.");
    else
        MessageBox("CANCEL 버튼을 눌렀습니다.");
}
    
```

20

모드형 대화상자 예제 작성

□ 실행결과



21

모드형 대화상자 주요함수

□ CDialog 클래스 가상 함수

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

- 호출 시점: 대화상자가 화면에 보이기 직전에 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
- EndDialog(IDOK)를 호출하여 대화상자를 종료

22

모드형 대화상자 주요함수

□ CDialog 클래스 가상 함수

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

- 호출 시점: IDCANCEL 버튼을 누를 때나 윈도우의 종료 버튼 또는 ESC 키를 누르면
- 용도: 대화상자 종료
- EndDialog(IDCANCEL)를 호출하여 대화상자를 종료

23

모드형 대화상자 주요함수

□ OnOK(), OnCancel() 함수 내부 구현

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

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

24

모드형 대화상자 예제 작성

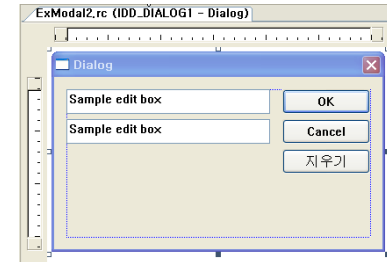
- 프로젝트 생성 - 대화상자에 편집컨트롤로 값 입력 받는 예제
- 1~6단계 옵션 설정

단계	변경 사항
1	'Single document'를 선택한다.
2	Compound Document Support 변경 사항 없음
3	DB 변경 사항 없음
4	UI 변경 사항 없음
5	Advanced Features 'ActiveX Controls' 선택을 해제함
6	Generated Classes 변경 사항 없음

25

대화상자 편집

- Resource View에서 Insert Dialog를 해서 IDD_DIALOG1 대화상자를 생성하고 컨트롤을 추가한다

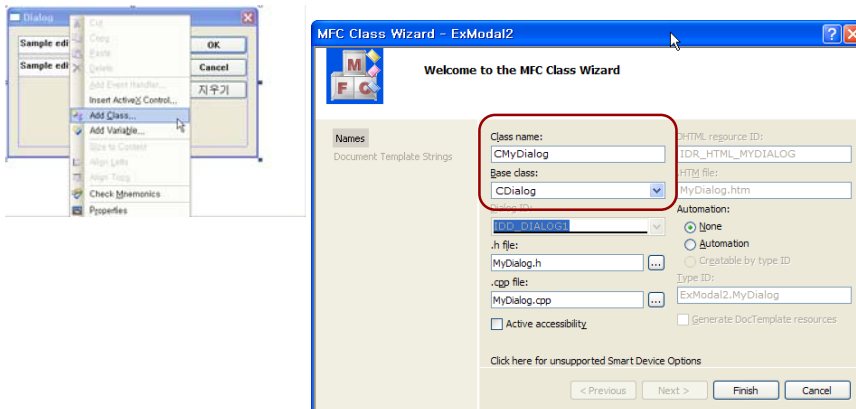


- 각 컨트롤의 속성

ID	Control	변경 사항
IDC_STR	Edit Control	없음
IDC_COLOR	Edit Control	없음
IDC_CLRSCR	Button Control	Caption을 “지우기”로 변경

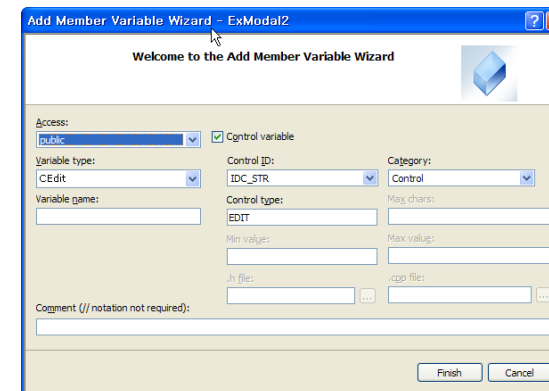
CDialog 클래스 생성

- 대화상자를 나타내는 클래스 생성
 - ClassWizard를 사용하여 Dialog class 생성
 - 원하는 Class name (예: CMyDialog)을 입력



CDialog 클래스 Member 정의

- 대화상자에 포함된 컨트롤을 연결 할 멤버변수 정의
 - 변수명, 연결형태(value, control), 자료형
 - ClassWizard의 Member Variables 탭 이용



28

클래스 위저스로 생성된 CMyDialog

```
//CDialog로부터 상속받은 클래스 생성(MyDialog.h)
class CMyDialog : public CDialog {
// Construction
public:
    CMyDialog(CWnd* pParent = NULL); // standard constructor
// Dialog Data
    enum { IDD = IDD_DIALOG1 };
    CString m_str; //생성된 member, DDX 연결변수(값 형식)
    int m_color; //생성된 member, DDX 연결변수(값 형식)

protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV
    ...
}
```

29

```
//대화상자 클래스 구현(MyDialog.cpp)
CMyDialog::CMyDialog(CWnd* pParent /*=NULL*/)
    : CDialog(CMyDialog::IDD, pParent) {
    m_str = ""; //추가된 member 초기화
}
//대화상자 클래스의 멤버와 대화상자의 컨트롤을 연결
void CMyDialog::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_STR, m_str); //컨트롤 값과 m_str 문자열변수 연결
    DDX_Text(pDX, IDC_COLOR, m_color); //컨트롤 값과 m_color 값 연결
}
void CMyDialog::OnClsrc() // 지우기버튼이 클릭됐을때
{
    EndDialog(100);
}
...
```

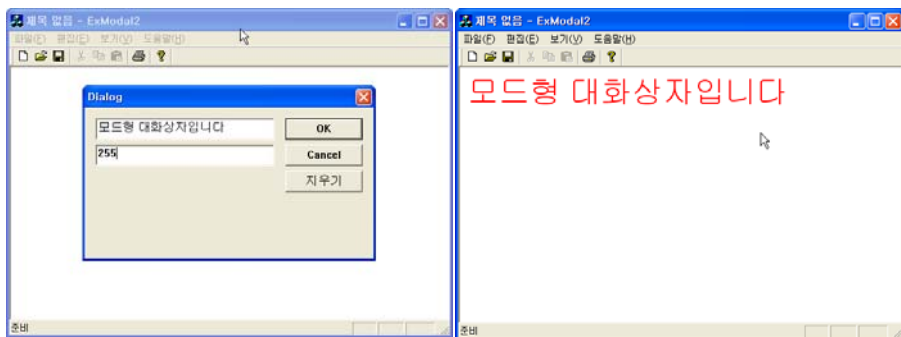
```
//뷰클래스
class CExModal2View : public CView {
// Construction
protected:
    CExModal2View();
    DECLARE_DYNCREATE(CExModal2View)
// Attributes
public:
    CExModal2Doc* GetDocument();
    CString m_str;
    int m_color;
    ...
}

//뷰클래스
CExModal2View : CExModal2View() {
m_str = "";
m_color = 0;
}
```

```
void CExModal2View::OnLButtonDbkClk(UINT nFlags, Cpoint point) {
CMyDialog dlg; // 대화상자 생성
dlg.m_str = m_str;
dlg.m_color = m_color;
int result = dlg.DoModal();
if (result == IDOK) { // OK 버튼을 누른 경우
    m_str = dlg.m_str;
    m_color = dlg.m_color;
    Invalidate();
}
else if (result == 100) { // 지우기 버튼을 누른 경우
    m_str = "";
    m_color = 0;
    Invalidate();
}
CView::OnLButtonDbkClk(nFlags, point);
}
// m_str과 m_color를 받아서 출력
void CExModal2View::OnDraw(CDC * pDC)
{ ..... }
```


모드형 대화상자 예제 작성

실행 결과

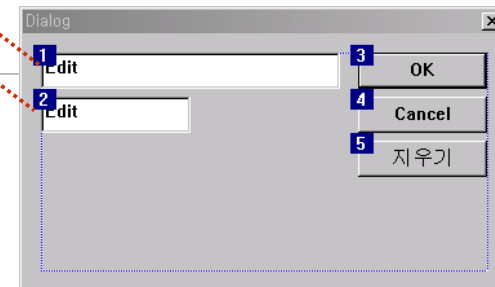


33

DDX/DDV

모드형 대화상자 구현

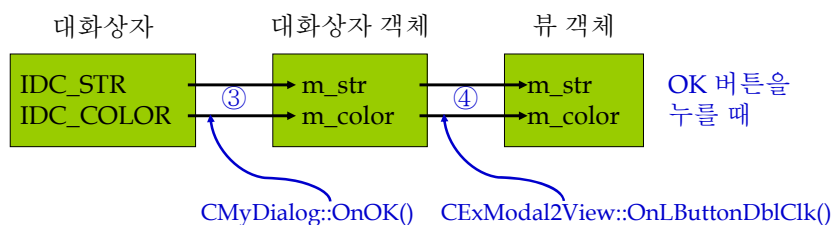
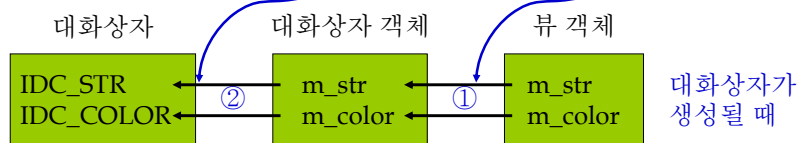
```
class CMyDialog : public CDialog
{
    ...
    CString m_str; // DDX 연결변수 (값 형식)
    int m_color; // DDX 연결변수 (값 형식)
    ...
}
```



DDX/DDV

동작 원리

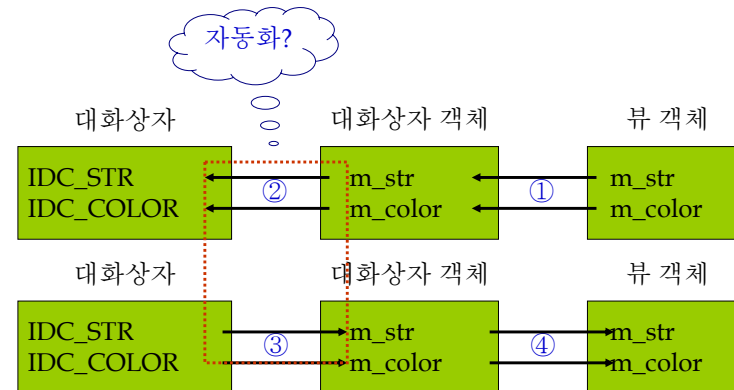
`CMyDialog::OnInitDialog()` `CExModal2View::OnLButtonDbClick()`



35

DDX/DDV

DDX(Dialog Data eXchange)



36

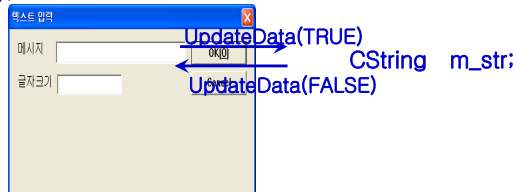
DDX/DDV

□ OnInitDialog(), OnOK() 함수 내부 구현

```

BOOL CDialog::OnInitDialog()
{
    ...
    UpdateData(FALSE); //컨트롤에 설정된 값을 변수로 가져온다
    ...
}

void CDialog::OnOK()
{
    ...
    UpdateData(TRUE); //변수 값을 컨트롤에 설정
    ...
}
    
```



37

DDX/DDV

□ CWnd::UpdateData() 함수 내부 구현

```

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

38

DDX/DDV

□ CWnd::UpdateData() 함수

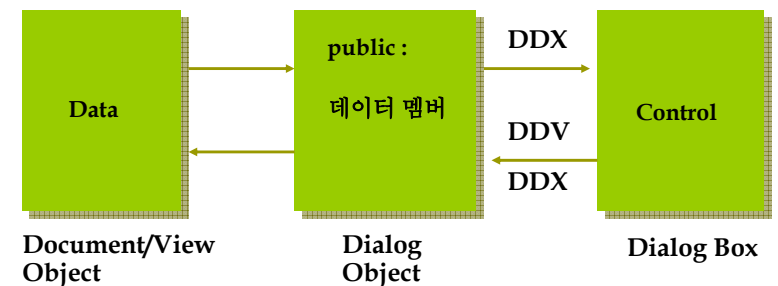
- DDX_Value 형태로 연결되었을 경우, 자원을 update하기 위한 함수
- UpdateData(TRUE)
 - 컨트롤에 있는 값을 연결된 멤버변수에 적용
 - 대화상자 종료(OnOK)시 호출
 - 내부적으로 DoDataExchange() 함수가 호출되어 정의된 DDV_*함수호출로 유효성 검사 후 DDX_*함수를 호출하여 데이터 전송
- UpdateData(FALSE)
 - 멤버변수에 설정된 값을 컨트롤에 적용
 - 대화상자 초기화 함수(OnInitDialog)에서 호출
 - 내부적으로 DoDataExchange() 함수가 호출되어 정의된 DDX_*함수를 호출하여 데이터 전송

39

DDX/DDV

□ MFC에서 제공하는 대화상자컨트롤과 대화상자 객체의 멤버변수 사이의 데이터 교환 기술

- DDX(Dialog Data Exchange) : 대화상자 데이터 교환
- DDV(Dialog Data Validation) : 대화상자 데이터 유효성 확인



40

DDX/DDV

□ DDX 구현

- 컨트롤의 값을 저장할 변수 선언
- 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
}
```

41

DDX/DDV

□ DDV(Dialog Data Validation) 구현

- 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
}
```

42

대화상자 클래스에서 컨트롤 제어

1. 컨트롤 객체로 접근
2. 컨트롤에 설정된 값을 접근하여 제어
3. DDX를 이용하여 컨트롤과 연결하는 방법

43

대화상자 클래스에서 컨트롤 제어

1. 컨트롤 객체로 접근
 - GetDlgItem(컨트롤ID) 함수로 얻어온 컨트롤 객체의 포인터를 이용하여 컨트롤 제어

```
CEdit *pText;
pText = (CEdit *) GetDlgItem(IDC_STR);
pText->GetWindowText(m_str); //Edit 컨트롤의 문자열 얻기
...
pText->SetWindowText(m_str); //Edit 컨트롤에 문자열 설정
...
```

44

대화상자 클래스에서 컨트롤 제어

2. 컨트롤에 설정된 값을 접근하여 제어

```
BOOL CMyDialog::OnInitDialog() {
    CDialog::OnInitDialog();
    //문자열 형태로 IDC_STR 컨트롤에 m_str값을 설정
    SetDlgItemText(IDC_STR, m_str);
    //정수형 형태로 IDC_COLOR 컨트롤에 m_color값을 설정
    SetDlgItemInt(IDC_COLOR, m_color);
    return TRUE;
}

void CMyDialog::OnOK() {
    //문자열 형태로 IDC_STR컨트롤의 값을 m_str로 얻어오기
    GetDlgItemText(IDC_STR, m_str);
    //정수형 형태로 IDC_COLOR 컨트롤의 값을 얻어 m_color에 대입
    m_color = GetDlgItemInt(IDC_COLOR);
    CDialog::OnOK();
}
```

45

대화상자 클래스에서 컨트롤 제어

3. DDX를 이용하여 컨트롤과 연결하는 방법

- 클래스 위저드를 통하여 컨트롤과 연관된 멤버변수를 설정하면 DoDataExchange() 함수 내에 연결 관련 함수 호출
- Value는 컨트롤에 설정되어 있는 값만 연결
 - DDX_Text 계열 함수를 이용하여 자원에 연결
- Control은 해당 컨트롤을 제어할 수 있는 컨트롤 클래스와 연결
 - DDX_Control 계열 함수를 이용하여 자원에 연결

```
void CMyDialog::DoDataExchange(CDataExchange* pDX) {
    CDialog::DoDataExchange(pDX);
    // Edit 컨트롤과 컨트롤형의 변수와 연결
    DDX_Control(pDX, IDC_EDIT, m_strCtrl);
    // Edit 컨트롤의 값을 문자열형의 변수와 연결
    DDX_Text(pDX, IDC_STR, m_str);
}
```

46

비 모드형 대화상자

□ 비모드형 (Modeless) 대화상자 작성 순서

- CDialog::DoModal() 함수 대신 **CDialog::Create()** 함수를 이용하여 생성한다.
- 대화상자를 닫을 때 CDialog::EndDialog() 함수 대신 **CWnd::DestroyWindow()** 함수를 호출한다.
- 모드형 대화상자 객체는 대개 스택에 생성하지만 비모드형 대화상자 객체는 힙에 생성한다.
 - 스택에 생성할 경우 변수 선언 영역을 벗어나면 소멸자가 호출되면서 대화상자가 자동으로 파괴되므로, 비모드형 대화상자를 계속 유지할 수 없기 때문이다.

47

비 모드형 대화상자 생성

□ 비모드형 대화상자 (Modeless Dialog Box)

- 제어권을 독점하고 있지 않아 메인 윈도우와 정보를 교환할 수 있는 형태
- 대화상자 생성 : CDialog::Create()
- 대화상자 화면표시 : CWnd::ShowWindow()
- 대화상자 종료 : CWnd::DestroyWindow()
- 대화상자 객체는 대화상자와 정보를 교환할 메인 윈도우에 동적 객체를 유지시킬 멤버변수로 선언

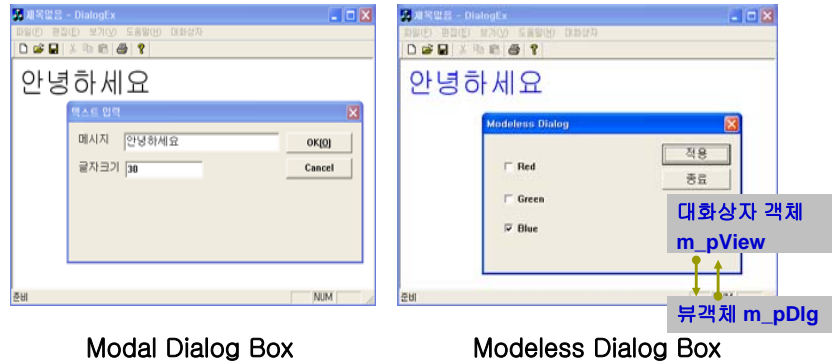
```
//ExModalessView.h
class CMyDialog; //CExDialog 클래스 사용을 선언
...
CMyDialog * m_Dlg; //대화상자 객체에 대한 멤버변수 선언
```

```
//ExModalessView.cpp
#include "MyDialog.h" // 대화상자의 헤더파일 포함
...
m_Dlg = new CMyDialog(); // 대화상자 객체 동적 생성
m_Dlg->Create(IDD_EXDIALOG,this); // 대화상자 생성
m_Dlg->ShowWindow(SW_SHOW); // 대화상자 화면 표시
```

48

SDI의 Dialog Box

- SDI/MDI 프로그램에서 pop-up 스타일로 대화상자 기능 이용



Modal Dialog Box

Modeless Dialog Box

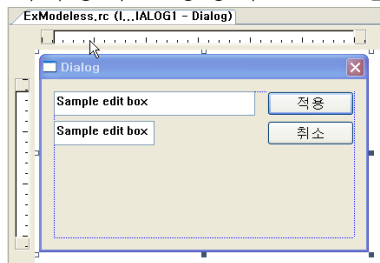
비 모드형 대화상자 예제 작성

- 프로젝트 생성
- 1~6단계 옵션 설정

단계	변경 사항
1	'Single document'를 선택한다.
2	Compound Document Support 변경 사항 없음
3	DB 변경 사항 없음
4	UI 변경 사항 없음
5	Advanced Features 'ActiveX Controls' 선택을 해제함
6	Generated Classes 변경 사항 없음

비 모드형 대화상자 예제 작성

- Resource View에서 Insert Dialog를 해서 IDD_DIALOG1 대화상자를 생성하고 컨트롤을 추가한다



ID	Control	변경 사항
IDC_STR	Edit Control	없음
IDC_COLOR	Edit Control	없음
IDAPPLY	Button Control	Caption을 "적용"으로 변경
IDCANCEL	Button Control	Caption을 "취소"로 변경

```

//MyDialog.h
Class CExModalessView; // 참조할 클래스 선언
class CMyDialog : public CDialog {
public:
    CMyDialog(CWnd* pParent = NULL); // standard constructor
    CExModalessView *m_pView;
    ... // 생략
}
//ExModalessView.h
Class CExModalessView: public Cview {
protected:
    CExModalessView();
    ...
public:
    CExModalessDoc * GetDocument();
    CMyDialog *m_pDlg;
    ... // 생략
}
    
```

비 모드형 대화상자 예제 작성

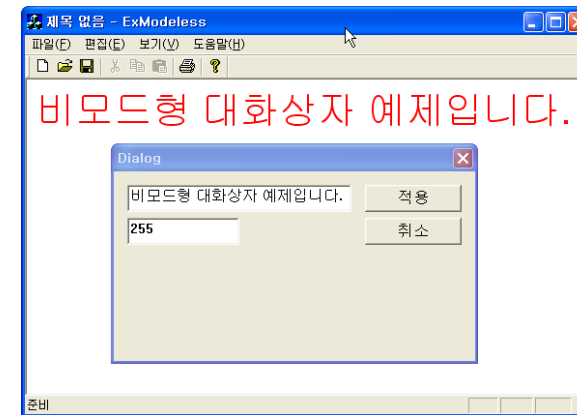
```

void CExModallessView::OnLButtonDbIClk(UINT nFlags, Cpoint point)
{
    if(m_pDlg != NULL)
        m_pDlg->SetFocus(); // 키보드 포커스를 대화상자로 옮김
    else{
        m_pDlg = new CMyDialog(); // 대화상자객체를 생성
        m_pDlg->m_pView = this; // 뷰 객체의 주소를 넘겨줌
        m_pDlg->m_str = m_str; // IDC_STR 컨트롤의 초기값
        m_pDlg->m_color = m_color; // IDC_COLOR 컨트롤의 초기값
        m_pDlg->Create(IDD_DIALOG1); // 대화상자 생성
        m_pDlg->ShowWindow(SW_SHOW); // 대화상자가 화면에 출력
    }
    CView::OnLButtonDbIClk(nFlags, point);
}
    
```

53

비 모드형 대화상자 예제 작성

실행결과

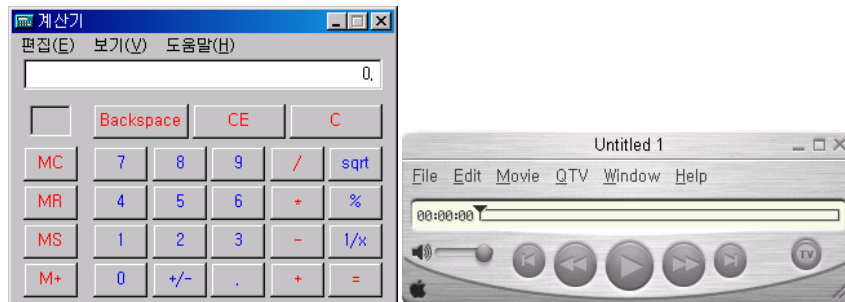


54

대화상자 기반 응용 프로그램

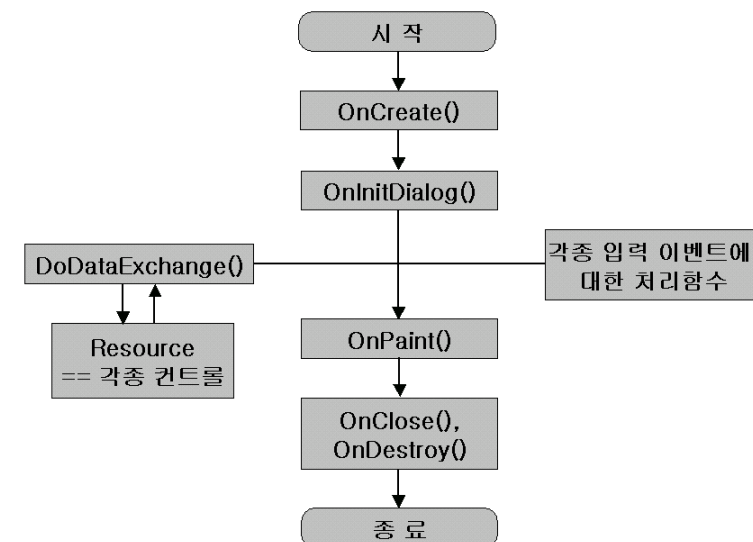
대화상자 기반 응용 프로그램

- 대화상자가 메인 윈도우 역할을 하는 응용 프로그램
- AppWizard의 응용프로그램 형식을 Dialog Based로 선택
- CWinApp, CDialog 클래스만 생성
- CWinApp::InitInstance()에서 DoModal()로 대화상자 생성



55

대화상자 기반 응용 프로그램 동작



대화상자 기반 응용 프로그램

□ InitInstance() 함수

```

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

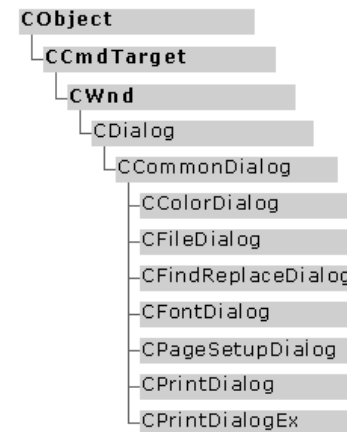
    }
    else if (nResponse == IDCANCEL) {

    }
    return FALSE;
}
    
```

57

공통 대화상자 (Common Dialog Box)

□ MFC 클래스 계층도



58

공통 대화상자

□ 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 (> 윈도 우 2000)

공통 대화상자

□ CColorDialog

```

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



```

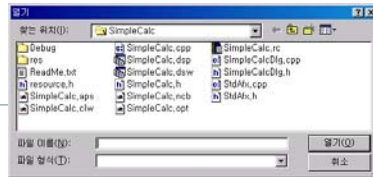
CColorDialog dlg(RGB(255, 0, 0), CC_FULLOPEN);
dlg.DoModal();
COLORREF color = dlg.GetColor();
    
```



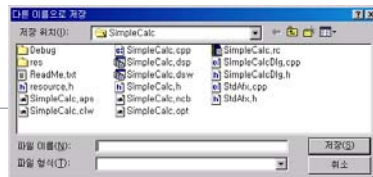
공통 대화상자

□ CFileDialog

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



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



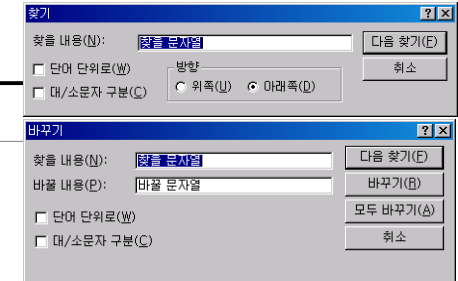
61

공통 대화상자

□ CFindReplaceDialog

```
CFindReplaceDialog *pDlg;
pDlg = new CFindReplaceDialog();
pDlg->Create(FALSE, "", "");
...
if(pDlg->IsTerminating()){ pDlg = NULL; return 0;}
else if(pDlg->FindNext()){
    str.Format("찾을 문자열: %s", pDlg->GetFindString());
}
else if(pDlg->ReplaceCurrent()){
    str.Format("찾을 문자열: %s\n\r바꿀 문자열: %s",
        pDlg->GetFindString(), pDlg->GetReplaceString());
}
else if(pDlg->ReplaceAll()){
    str.Format("찾을 문자열: %s\n\r바꿀 문자열: %s",
        pDlg->GetFindString(), pDlg->GetReplaceString());
}
...

```



62

공통 대화상자

□ CFontDialog

```
CFontDialog dlg; // 글꼴 대화상자
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);
    ...
}
```

63

공통 대화상자

□ CFontDialog

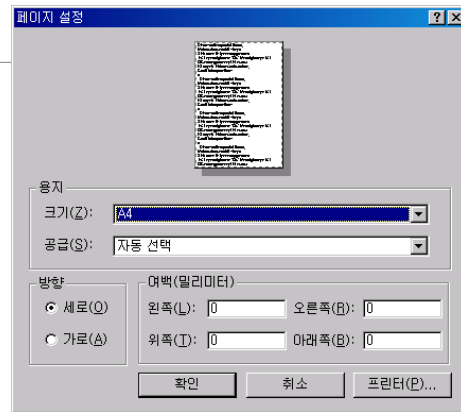
```
// 선택된 폰트를 알아낸다
LOGFONT lf;
dlg.GetCurrentFont(&lf);
CFont font;
font.CreateFontIndirect(&lf);
dc.SelectObject(&font);
// 텍스트를 출력한다
dc.TextOut(10, 10,
    CString("한글 & English"));
}
```



공통 대화상자

□ CPageSetupDialog

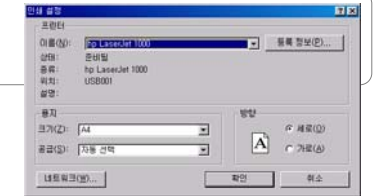
CPageSetupDialog dlg;
dlg.DoModal();



공통 대화상자

□ CPrintDialog

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



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

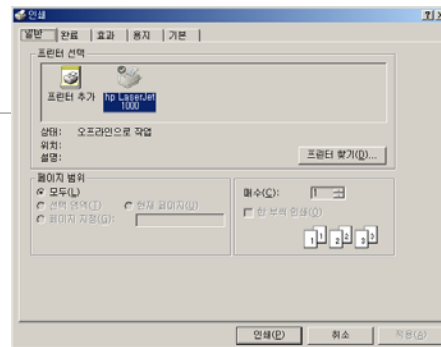


공통 대화상자

□ CPrintDialogEx

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

CPrintDialogEx dlg;
dlg.DoModal();



Practice

- SimpleMouseDrawing 프로그램을 이용하여 도형의 색과 선 두께를 선택할 수 있는 Dialog Box를 구현하라
 - Drawing 메뉴에서 색(Color)을 선택하면, 색을 자유롭게 선택할 수 있게 CColorDialog를 추가
 - Resource에서 메뉴의 Drawing->Color 에, add event handler 추가
ON_COMMAND(ID_DRAWING_COLOR, &CChildView::OnDrawingColor)
 - OnDrawingColor에서 DoModal()로 CColorDialog를 호출
 - Drawing 메뉴에서 선의 두께(Thickness)를 선택하면, 콤보박스로 1~5까지의 두께를 선택할 수 있는 Thickness Dialog를 생성
 - Add new MFC class wizard로 CThicknessDlg 생성 (Base class로 CDialog, ID로 IDD_THICKNESS_DIALOG)
 - Resource에서 IDD_THICKNESS_DIALOG 편집기에 콤보박스(IDC_COMBO1) 추가
 - 콤보박스에 Add member variable wizard로 멤버변수 정의
 - Resource에서 메뉴의 Drawing->Thickness 에, add event handler 추가
ON_COMMAND(ID_DRAWING_THICKNESS, &CChildView::OnDrawingThickness)
 - OnDrawingThickness에서 DoModal()로 CThicknessDlg를 호출
 - 툴바에서 두께아이콘을 추가하고, Thickness Dialog를 연계
 - Resource에서 두께를 나타내는 툴바 아이콘을 생성하고, ID_DRAWING_THICKNESS 지정