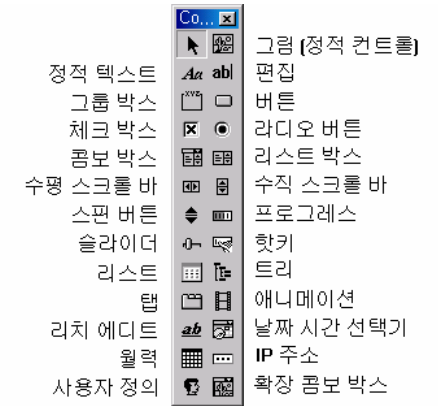


# 공통 컨트롤

HCI Programming 2 (321190)  
 2007년 가을학기  
 12/3/2007  
 박경신

## 공통 컨트롤 (Window Common Control)

- Animate Control
- Image List Control
- List Control
- Progress Bar Control
- Tab Control
- Tree Control
- Spin Control
- Slider Control



## 공통 컨트롤 종류

컨트롤 이름	MFC 클래스	역할
애니메이션	CAnimateCtrl	AVI 비디오 클립 재생
날짜 시간 선택기	CDateTimeCtrl	날짜 또는 시간 선택
확장 콤보 박스	CComboBoxEx	일종의 콤보 박스로 텍스트와 이미지 표시
헤더	CHeaderCtrl	여러 열(column)로 표시되는 텍스트의 폭 조절
핫키	CHotKeyCtrl	핫키 입력과 생성
이미지 리스트	CImageList	여러 개의 이미지 관리
리스트	CListCtrl	텍스트와 아이콘을 다양한 방법으로 표시
달력	CMonthCalCtrl	날짜 정보 표시와 선택
프로그레스	CProgressCtrl	작업 진행 상황 표시 <sup>3</sup>

## 공통 컨트롤 종류

컨트롤 이름	MFC 클래스	역할
리바 (Rebar)	CRebarCtrl	다른 컨트롤을 자식 윈도우로 포함하는 툴바
리치에디트	CRichEditCtrl	다양한 기능을 제공하는 텍스트 입력 컨트롤
슬라이더	CSliderEx	눈금 표시를 가질 수 있는 일종의 스크롤 바
스피너 버튼	CSpinButtonCtrl	두 개의 화살 표시로 값 증감
상태바	CStatusBarCtrl	프로그램의 상태 표시
탭	CTabCtrl	동일한 화면 영역에 서로 다른 페이지를 표시 및 전환
툴바	CToolBarCtrl	명령 메시지를 발생시키는 비트맵 버튼의 집합
툴팁	CTooltipCtrl	툴바 버튼 등 다른 컨트롤에 조그만 크기의 팝업 윈도우를 표시
트리	CTreeCtrl	항목을 계층적으로 표시 <sup>4</sup>

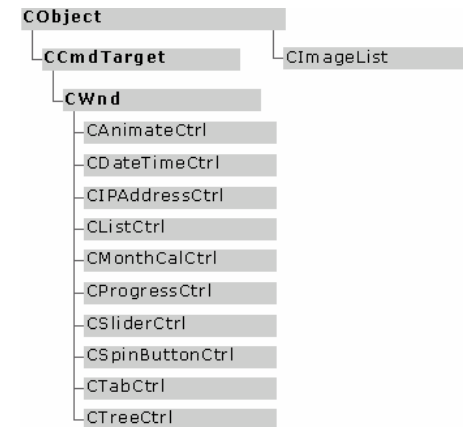
## 공통 컨트롤 특징

- 표준 컨트롤은 USER.EXE에 구현되어 있지만 **공통 컨트롤은 COMCTL32.DLL에 구현되어 있다.**
  - COMCTL32.DLL의 버전에 따라 사용할 수 있는 공통 컨트롤의 종류와 기능에 제약이 있다.
- 사용하기 전에 반드시 `::InitCommonControls()` 또는 `::InitCommonControlsEx()`를 호출해야 한다.
  - MFC를 사용하는 경우 이 과정은 자동으로 이루어진다.
- 일반적으로 통지 메시지를 위해 표준 컨트롤은 `WM_COMMAND` 메시지를 사용하지만 **공통 컨트롤은 `WM_NOTIFY` 메시지를 사용한다.**

5

## MFC 클래스 계층도

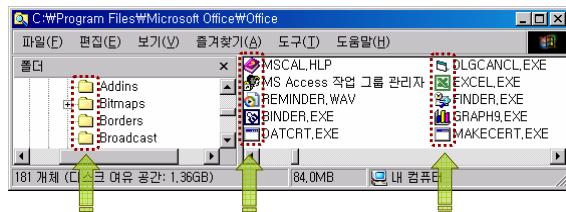
- 공통 컨트롤 클래스 계층도



6

## 이미지 리스트

- 이미지 리스트 (Image List)
  - 동일한 크기의 이미지 집합
  - 각각의 이미지를 배열 항목처럼 취급하여 0부터 시작하는 인덱스로 참조
  - 다른 컨트롤의 아이콘이나 비트맵을 관리하는 이미지 집합
- 사용 예
  - 트리 컨트롤 (CTreeCtrl)과 리스트 컨트롤 (CListCtrl)에 사용되는 비트맵



7

## 이미지 리스트

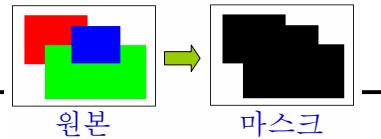
- 이미지 리스트 생성 방법

```

CImageList m_il;
m_il.Create(...);
    
```

8

## 이미지 리스트



### □ 이미지 리스트 생성 함수

```
BOOL CImageList::Create (int cx, int cy, UINT nFlags, int nInitial, int nGrow);
BOOL CImageList::Create (UINT nBitmapID, int cx, int nGrow, COLORREF crMask);
```

- cx, cy: 이미지의 폭과 높이(픽셀 단위)
- nFlags: 이미지 리스트의 타입 지정 (ILC\_COLOR4: 4비트16색상)
- nInitial: 초기에 포함할 이미지 개수
- nGrow: 새로운 이미지를 추가하기 위해 이미지 리스트의 메모리를 재할당할 때 얼마만큼 여분의 메모리를 할당할 것인가를 나타냄
- nBitmapID: 비트맵 리소스 ID
- crMask: 투명색으로 사용할 색상

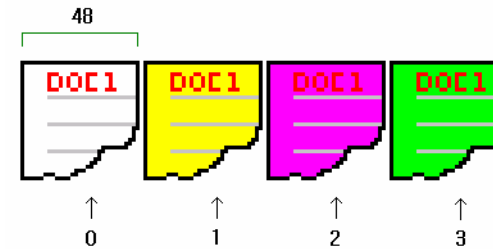
9

## 이미지 리스트

### □ 이미지 리스트 생성 예

```
CImageList m_il1, m_il2;
m_il1.Create(32, 32, ILC_COLOR4, 2, 1);
m_il2.Create(IDB_BITMAP1, 48, 1, RGB(255, 255, 255));
```

#### ■ IDB\_BITMAP1



10

## 이미지 리스트

### □ 이미지 추가 함수

- 이미지 리스트에 새로운 비트맵이나 아이콘을 추가

```
int CImageList::Add (HICON hIcon);
int CImageList::Add (CBitmap* pbmImage, COLORREF crMask);
```

- hIcon: 아이콘 리소스를 가리키는 핸들값
- pbmImage: 비트맵 리소스를 담고 있는 CBitmap 객체의 주소
- crMask: 투명색으로 사용할 색상

11

## 이미지 리스트

### □ 이미지 추가 예

```
// 두 개의 아이콘을 이미지 리스트에 추가한다.
m_il1.Add(AfxGetApp()->LoadIcon(IDI_ICON1));
m_il1.Add(AfxGetApp()->LoadIcon(IDI_ICON2));
// 비트맵을 이미지 리스트에 추가한다.
CBitmap bitmap;
bitmap.LoadBitmap(IDB_BITMAP1);
m_il2.Add(&bitmap, RGB(255, 255, 255));
```

12

## 이미지 리스트

### □ 화면 출력 함수

```
BOOL CImageList::Draw (CDC* pDC, int nIndex, POINT pt, UINT nStyle);
```

- pDC: 출력할 대상을 나타내는 디바이스 컨텍스트
- nIndex: 출력할 이미지의 인덱스
- pt: 출력 좌표
- nStyle: 그리기 스타일(다음 페이지 표 참조)

13

## 이미지 리스트

### □ 그리기 스타일 (nStyle)

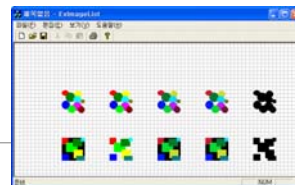
값	의미
ILD_BLEND25, ILD_FOCUS	시스템 하이라이트 색상을 25% 섞어서 출력
ILD_BLEND50, ILD_SELECTED, ILD_BLEND	시스템 하이라이트 색상을 50% 섞어서 출력
ILD_MASK	마스크를 출력
ILD_NORMAL	이미지의 배경색을 이용하여 출력하며 배경색이 CLR_NONE으로 설정된 경우 마스크를 이용하여 이미지를 투명하게 출력
ILD_TRANSPARENT	마스크를 이용하여 이미지를 투명하게 출력

14

```
void CEImageListView::OnDraw(CDC* pDC) {
    CBrush brush(HS_CROSS, RGB(200, 200, 200));
    pDC->SelectObject(&brush);
    CRect rect;
    GetClientRect(&rect);
    pDC->Rectangle(&rect);

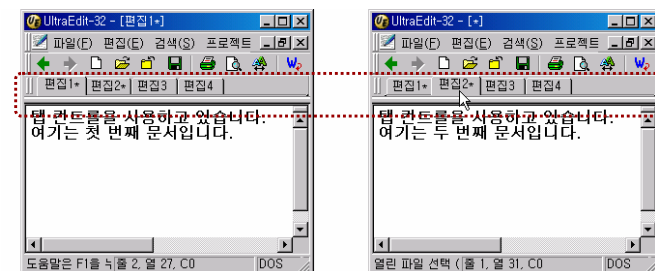
    m_il.SetBkColor(CLR_NONE); //이미지의 배경색을 투명하게 설정
    m_il.Draw(pDC, 0, CPoint(100, 100), ILD_NORMAL);
    m_il.Draw(pDC, 0, CPoint(200, 100), ILD_TRANSPARENT);
    m_il.Draw(pDC, 0, CPoint(300, 100), ILD_BLEND25);
    m_il.Draw(pDC, 0, CPoint(400, 100), ILD_BLEND50);
    m_il.Draw(pDC, 0, CPoint(500, 100), ILD_MASK);

    m_il.SetBkColor(RGB(0, 0, 0)); //이미지의 배경색을 검은색으로 설정
    m_il.Draw(pDC, 1, CPoint(100, 200), ILD_NORMAL);
    m_il.Draw(pDC, 1, CPoint(200, 200), ILD_TRANSPARENT);
    m_il.Draw(pDC, 1, CPoint(300, 200), ILD_BLEND25);
    m_il.Draw(pDC, 1, CPoint(400, 200), ILD_BLEND50);
    m_il.Draw(pDC, 1, CPoint(500, 200), ILD_MASK);
}
```



## 탭 컨트롤 (Tab Control)

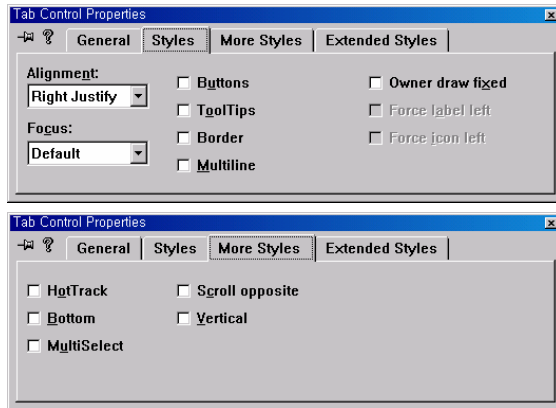
- 동일한 화면 영역에 서로 다른 내용을 표시
- 정보를 기능별로 분류하고자 할 때 사용
- 주어진 탭을 선택함에 따라 해당하는 정보를 분류하여 보여줌
- 작은 대화상자 안에 많은 정보를 표현하고자 할 때 사용



16

## 탭 컨트롤 스타일

### □ 탭 컨트롤 스타일



17

## 탭 컨트롤 통지 메시지

- WM\_NOTIFY 통지 메시지 발생
- 통지 메시지와 더불어 다음과 같은 통지 코드 전달

통지 코드	의미
TCN_SELCHANGING	탭 선택이 변경되기 직전이다.
TCN_SELCHANGE	탭 선택이 변경되었다.
TCN_KEYDOWN	키를 눌렀다.
NM_CLICK	마우스 왼쪽 버튼을 클릭하였다.
NM_RCLICK	마우스 오른쪽 버튼을 클릭하였다.

18

## 탭 컨트롤 클래스

### □ 탭에 사용할 이미지 리스트 설정하기

```
CImageList * CTabCtrl::SetImageList (CImageList * pImageList);
```

### □ 탭 추가하기

```
LONG CTabCtrl::InsertItem (int nIndex, TCITEM* pItem);
LONG CTabCtrl::InsertItem (int nIndex, LPCTSTR lpszItem);
LONG CTabCtrl::InsertItem (int nIndex, LPCTSTR lpszItem,
                           int nImage);
```

- nIndex: 탭의 인덱스(0, 1, ...)
- TCITEM: 하나의 탭에 대한 정보를 가진 구조체
- lpszItem: 새로운 탭에 표시할 텍스트(레이블)
- nImage: 탭에 표시할 이미지(이미지 리스트 인덱스값)

19

## 탭 컨트롤 클래스

### □ TCITEM 구조체

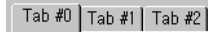
```
typedef struct tagTCITEM {
    UINT mask;
    DWORD dwState;
    DWORD dwStateMask;
    LPTSTR pszText;
    int cchTextMax;
    int iImage;
    LPARAM lParam;
} TCITEM, FAR *LPTCITEM;
```

20

## 탭 컨트롤 클래스

### □ 탭 추가 예

```
TCITEM tcltem;           // 구조체 이용 탭추가
tcltem.mask = TCIF_TEXT;
tcltem.pszText = "Tab #0";
m_tab.InsertItem(0, &tcltem);
tcltem.pszText = "Tab #1";
m_tab.InsertItem(1, &tcltem);
tcltem.pszText = "Tab #2";
m_tab.InsertItem(2, &tcltem);
```



```
m_tab.InsertItem(0, "Tab #0"); // 매개변수 이용 탭추가
m_tab.InsertItem(1, "Tab #1");
m_tab.InsertItem(2, "Tab #2");
```

21

## 탭 컨트롤 클래스

### □ 탭 선택 변경과 선택된 탭 알아내기

```
int CTabCtrl::SetCurSel (int nItem);
int CTabCtrl::GetCurSel ();
```

22

## 스핀 버튼 컨트롤 (Spin Button Control)

- 스핀 버튼 컨트롤 = 업다운 컨트롤
- 화살표가 두 개 붙어 있어서 마우스 클릭이나 키보드로 값을 증감하는 용도로 사용
- Edit Box안에 붙여서 함께 사용
  - Auto buddy 스타일을 설정하고, 탭순서를 테스트 박스 - 스핀 버튼 순으로 설정



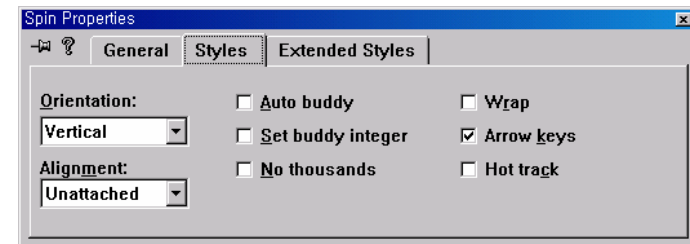
스핀 버튼 컨트롤

버디 컨트롤

23

## 스핀 버튼 컨트롤

### □ 스핀 버튼 컨트롤 스타일



24

## 스핀 버튼 컨트롤 클래스

### □ 범위 설정과 현재 범위 얻기

```
void CSpinButtonCtrl::SetRange (short nLower, short nUpper);  
void CSpinButtonCtrl::GetRange (int &lower, int &upper);
```

### □ 위치 설정과 현재 위치 얻기

```
int CSpinButtonCtrl::SetPos (int nPos);  
int CSpinButtonCtrl::GetPos ();
```

### □ 버디 설정

```
CWnd* CSpinButtonCtrl::SetBuddy (CWnd* pWndBuddy);
```

25

## 스핀 버튼 컨트롤 클래스

### □ 숫자의 진수 변경

```
int CSpinButtonCtrl::SetBase (int nBase);
```

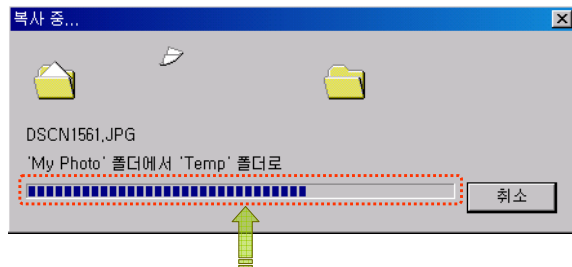
### □ 스핀 버튼 컨트롤 통지 메시지

- 마우스로 조작하면 WM\_VSCROLL 또는 WM\_HSCROLL 메시지 발생

26

## 프로그레스 컨트롤 (Progress Bar Control)

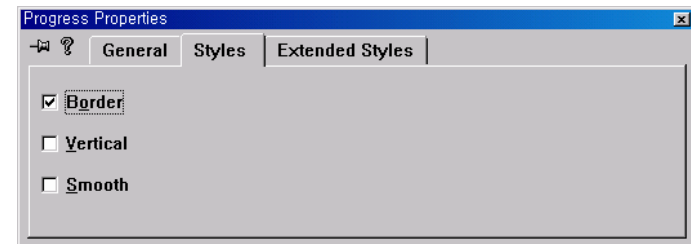
- 오랜 시간이 걸리는 작업의 진행 상황을 시각적으로 표시하는 용도로 사용



27

## 프로그레스 컨트롤

- 프로그레스 컨트롤 스타일



28

## 프로그레스 컨트롤 클래스

- 범위 설정과 현재 범위 얻기

```
void CProgressCtrl::SetRange (short nLower, short nUpper);
void CProgressCtrl::GetRange (int& nLower, int& nUpper);
```

- 위치 설정과 현재 위치 얻기

```
int CProgressCtrl::SetPos (int nPos);
int CProgressCtrl::GetPos ();
```

- 위치값 증감

```
int CProgressCtrl::OffsetPos (int nPos);
int CProgressCtrl::SetStep (int nStep);
int CProgressCtrl::StepIt ();
```

29

## 슬라이더 컨트롤 (Slider Control)

- 주어진 범위의 값이나 비 연속적인 값을 입력하고자 할 때 사용
- 주어진 범위 안에서 마우스를 드래그 함으로써 값을 증가, 감소할 수 있게 하는 컨트롤



30

## 슬라이더 컨트롤 클래스

- 범위 설정과 현재 범위 얻기

```
void CSliderCtrl::SetRange( int nMin, int nMax, BOOL bRedraw = FALSE );
void CSliderCtrl::GetRange( int& nMin, int& nMax )
```

- 위치 설정과 현재 위치 얻기

```
int CSliderCtrl::SetPos (int nPos);
int CSliderCtrl::GetPos ();
```

- 슬라이더 컨트롤 통지 메시지
  - NM\_CUSTOMDRAW 발생

31

## Animate Control

- AVI 동영상 파일을 재생하고 제어하는데 사용 (사운드 지원 안됨)
- CAnimateCtrl 주요 메소드
  - 리소스나 파일로부터 AVI clip 열기
    - BOOL Open( LPCTSTR lpszFileName );
    - BOOL Open( UINT nID );
  - AVI clip 실행
    - BOOL Play( UINT nFrom, UINT nTo, UINT nRep );
    - ACS\_AUTOPLAY로 스타일이 설정된 경우는 불필요
  - AVI clip의 지정된 프레임으로 이동하여 보여줌
    - BOOL Seek( UINT nTo );
  - AVI clip의 실행 정지
    - BOOL Stop();
  - AVI clip 닫기
    - BOOL Close();

