



# Window API Programming- 입문 4

[afewhee@gmail.com](mailto:afewhee@gmail.com)





- 1. Common Dialog – 파일, 색상 폰트 열기
- 2. Common Control
- 3. Memory Allocation
- 4. File 입/출력
- 5. Image 출력





# 1. Common Dialog – File Open

- OPENFILENAME 구조체

- GetOpenFileName()

- 기본 예제)

```
OPENFILENAME OFN;
```

```
char lpstrFile[MAX_PATH]="";
```

```
memset(&OFN, 0, sizeof(OPENFILENAME));
```

```
OFN.lStructSize = sizeof(OPENFILENAME);
```

```
OFN.hwndOwner=hWnd;
```

```
OFN.lpstrFilter="모든 파일 (*.*)\0*.*\0";
```

```
OFN.lpstrFile=lpstrFile;
```

```
OFN.nMaxFile=MAX_PATH;
```

```
if (GetOpenFileName(&OFN)!=0)
```

```
...
```

- 여러 파일을 선택할 경우 구분자를 NULL 문자를 사용하고 있어서 OFN.lpstrFile의 대상은 전부 NULL문자로 초기화 해야 함.





# 1. Common Dialog – File Open

- 확장자에 대한 필터를 사용하는 경우 NULL 문자로 구분  
Ex) `OFN.lpstrFilter="Text Files(*.txt)\0*.txt\0All Files(*.*)\0*.*\0";`
- 디폴트 확장자: `OFN.lpstrDefExt="abc";`
- 시작 폴더 지정: `OFN.lpstrInitialDir=InitDir;`
- 여러 파일을 선택하는 경우:
  - ◆ `OFN.nMaxFile=10000;`
  - ◆ `OFN.Flags=OFN_EXPLORER | OFN_ALLOWMULTISELECT;`





- CHOOSECOLOR 구조체 사용
- ChooseColor() 함수 이용

- Ex)

```
static COLORREF Color=RGB(0,0,255);  
static COLORREF crTemp[16];  
CHOOSECOLOR COL;
```

```
memset(&COL, 0, sizeof(CHOOSECOLOR));  
COL.lStructSize = sizeof(CHOOSECOLOR);  
COL.hwndOwner=m_hWnd;  
COL.lpCustColors=crTemp;
```

// Default color

```
if( TRUE != ChooseColor(&COL))  
{  
    // Process Color  
}
```



- CHOOSEFONT 구조체 사용
- ChooseFont() 함수 이용

- Ex)

```
LOGFONT lf;
```

```
CHOOSEFONT CFT;
```

```
memset(&CFT, 0, sizeof(CHOOSEFONT));
```

```
CFT.lStructSize = sizeof(CHOOSEFONT);
```

```
CFT.hwndOwner=m_hWnd;
```

```
CFT.lpLogFont=&lf;
```

```
CFT.Flags=CF_EFFECTS | CF_SCREENFONTS;
```

```
if (ChooseFont(&CFT))
```

```
...
```





- 표준 컨트롤과 유사. 시스템에서 지원하지 않고 DLL에 의해 제공
- 헤더파일:
  - ◆ `#include <commctrl.h>`
- 링크
  - ◆ `#pragma comment(lib, "comctl32.lib")`
- DLL 로드:
  - ◆ `InitCommonControls()/ InitCommonControlsEx()` 함수 사용





- 특정한 크기의 이미지 필요
- 버튼 아이디 연결: `ToolBtn[i].idCommand= ID;`
- 객체 생성: `CreateToolbarEx()`
- 메시지 처리: `WM_COMMAND`
- 툴 팁: `WM_NOTIFY`

- Ex)

```
TBBUTTON ToolBtn[3];
```

```
...
```

```
ToolBtn[i].iBitmap=i;
```

```
ToolBtn[i].idCommand=ID_BTN_CIRCLE+i;
```

```
ToolBtn[i].fsState=TBSTATE_ENABLED;
```

```
ToolBtn[i].fsStyle=TBSTYLE_GROUP;
```

```
ToolBtn[i].dwData=0;
```

```
ToolBtn[i].iString=0;
```

```
...
```

```
m_hToolBar=CreateToolbarEx(...)
```

```
...
```

```
case WM_COMMAND:
```

```
...
```

```
case WM_NOTIFY:
```

```
...
```







- 객체 생성: `CreateWindow(WC_LISTVIEW, ...)`
- 이미지 연결: `ListView_SetImageList(..., LVSIL_SMALL/LVSIL_NORMAL)`
- 헤더 추가:
  - ◆ `LVCOLUMN` 구조체 사용
  - ◆ `ListView_InsertColumn()`
- 아이템 추가:
  - ◆ `LVITEM` 구조체 사용
  - ◆ 아이템 추가: `ListView_InsertItem()`
  - ◆ 텍스트 설정: `ListView_SetItemText()`
- 메시지 처리: `WM_NOTIFY`  
Ex)  

```
case WM_NOTIFY:  
{  
    LPNMHDR hdr; LPNMLISTVIEW nlv;  
    hdr=(LPNMHDR)IParam; nlv=(LPNMLISTVIEW)IParam;  
  
    if (hdr->hwndFrom == m_hList)  
    {  
        switch (hdr->code)  
        {  
            case LVN_ITEMCHANGED:  
            {  
                if (nlv->uChanged == LVIF_STATE && nlv->uNewState == (LVIS_SELECTED | LVIS_FOCUSED))  
                {  
                    ...  
                }  
            }  
        }  
    }  
}
```





## 2. Common Control – Tree View

- 객체 생성: `CreateWindow(WC_TREEVIEW, ...)`
- 이미지 연결: `TreeView_SetImageList(..., LVSIL_SMALL/LVSIL_NORMAL)`
- 아이템:
  - ◆ 아이템 추가
    - `TVINSERTSTRUCT` 구조체 사용
    - `TreeView_InsertItem ()`
  - ◆ 아이템 삭제: `TreeView_DeleteItem()`
  - ◆ 아이템 선택/수정: `TreeView_GetItem()/TreeView_SetItem()`
- 노드 선택: `TreeView_GetSelection()`
- 노드 확장: `TreeView_Expand()`

- 메시지 처리: `WM_NOTIFY`

Ex)

```
case WM_NOTIFY:
```

```
{
```

```
    LPNMHDR hdr; LPNMLISTVIEW nlv;
```

```
    hdr=(LPNMHDR)lparam; nlv=(LPNMLISTVIEW)lparam;
```

```
    if (hdr->hwndFrom == m_hTree)
```

```
    {
```

```
        switch (hdr->code)
```

```
        {
```

```
            case TVN_SELCHANGED:
```

```
            {
```

```
...
```





## 2. Common Control – Tab Control

- 객체 생성: 리소스에서 다이얼로그를 만든 후 Tab Control을 붙임
- 자식이면서 Border가 없는 윈도우 생성 → Tab 선택에 따라 윈도우를 show/hide 함
- 이미지 연결: TabCtrl\_SetImageList()
- 아이템: TabCtrl\_InsertItem()

- 메시지 처리: WM\_NOTIFY

Ex)

```
case WM_NOTIFY:
```

```
{
```

```
    switch (((LPMHDR)lParam)->code)
```

```
    {
```

```
        case TCN_SELCHANGE:
```

```
        {
```

```
            nTab=TabCtrl_GetCurSel(GMAIN->m_hwWrkTb);
```

```
            if (nTab==0)
```

```
            {
```

```
                ShowWindow(ChildWindow, SW_SHOW);
```

```
...
```





## 2. Common Control – Tool Tip

- 객체 생성: `CreateWindow(TOOLTIPS_CLASS, ...)`
- 툴팁 연결:
  - ◆ `TTTOOLINFO` 구조체 이용
  - ◆ `ti.ulid= 연결 대상 핸들;`
  - ◆ `ti.lpszText = "연습 버튼 입니다";`
- 연결 메시지 전달:
  - ◆ `SendMessage(m_hTip, TTM_ADDTOOL, 0, (LPARAM) (LPTOOLINFO) &ti);`





# 3. Memory Allocation

- 동적 Memory(Heap 영역)
  - ◆ C 언어: malloc, calloc, realloc – free
  - ◆ C++: new – delete 연산자
  - ◆ 내부적으로 윈도우의 메모리 관리자가 이를 정함
- Win API Memory:
  - ◆ Heap 영역은 윈도우에서 가상 메모리
  - ◆ 가상 메모리: 윈도우는 {주 기억장치 + 보조기억장치 일부}를 가상 기억공간(페이징)으로 활용 → 페이지 테이블
  - ◆ VirtualAlloc() / VirtualFree() – 가상 메모리에 할당
  - ◆ GlobalAlloc() / GlobalFree() - Global Heap 영역에 메모리 할당
  - ◆ HeapCreate() / HeapDestroy(), HeapAlloc() / HeapFree(), LocalAlloc() / LocalFree() - Local Heap 영역에 메모리 할당
- VirtualAlloc와 GlobalAlloc, LocalAlloc의 차이점:
  - ◆ VirtualAlloc은 64비트 메모리 주소. GlobalAlloc과 LocalAlloc은 32비트 메모리 주소
  - ◆ 윈도우95 이상에서는 차이 없이 가상 메모리 형태로 사용 → Heap 영역
- Lock()/Unlock()
  - ◆ 운영체제는 자주 사용되는 공간은 RAM에, 빈도가 낮은 주소는 보조기억장치의 페이징 파일로 보냄. Lock()은 이러한 페이지 교체를 막아줌 → 성능 저하 유발





- Memory Mapping File: **파일 = 메모리**
  - ◆ 파일 생성 :
    - g\_hFile=CreateFile("파일 이름", ...)
  - ◆ 파일 맵핑 오브젝트 생성:
    - g\_hMap=CreateFileMapping(g\_hFile, ...)
  - ◆ 주소 공간 맵핑:
    - g\_pMsg=(char\*)MapViewOfFile(g\_hMap, ...)
  - ◆ 주소 공간 맵핑 해제:
    - UnmapViewOfFile(g\_pMsg)
  
- Memory Share Mapping : 프로세스간 메모리 공유
  - ◆ 메모리 맵 파일 이용
  - ◆ 파일 맵핑 오브젝트의 첫 번째 인수를 0xFFFFFFFF로 생성
    - g\_hMap=CreateFileMapping(INVALID\_HANDLE\_VALUE, ...)





- C 언어 파일 함수
  - ◆ fopen() / fclose()
  - ◆ fgets(), fprintf() / fread(), fwrite()
  - ◆ fseek(), ftell()
- Win API
  - ◆ CreateFile() / CloseHandle()
  - ◆ ReadFile(), WriteFile()
  - ◆ SetFilePointer()
- 비동기 파일 입/출력
  - ◆ 비동기 입출력 파일 생성:  
CreateFile("File Name", ...  
, 0, NULL, ...  
, FILE\_ATTRIBUTE\_NORMAL | FILE\_FLAG\_OVERLAPPED, NULL);
  - ◆ OVERLAPPED 구조체 변수에 완료에 대한 이벤트 연결
    - OVERLAPPED g\_OL; g\_OL.hEvent = CreateEvent(NULL, TRUE, FALSE, NULL);
  - ◆ 비동기 입/출력 실행
    - WriteFile(g\_hFile, pBuf, MAX\_BUF\_SIZE, &g\_dBuf, &g\_OL)
  - ◆ 비동기 입/출력 완료: GetOverlappedResult() 함수로 확인
    - BOOL hr = GetOverlappedResult(g\_hFile, &g\_OL, &g\_dBuf, FALSE);
  - ◆ 파일, 이벤트 해제:
    - CloseHandle(...); CloseHandle(...);





- DDB(Device Dependent Bitmap)

- ◆ 8bit(256) color
- ◆ 출력 장치에 의존

Ex)

```
hMemDC=CreateCompatibleDC(hdc);
```

```
hBitmap=CreateBitmap();
```

```
SetBitmapBits(hBitmap, 16, Bits);
```

```
hOldBitmap=(HBITMAP)SelectObject(hMemDC, hBitmap);
```

```
BitBlt(hdc, 2, 2, 8, 8, hMemDC, 0, 0, SRCCOPY);
```

```
SelectObject(hMemDC, hOldBitmap);
```

```
DeleteDC(hMemDC);
```

```
DeleteObject(hBitmap);
```







- DIB(Device Independent Bitmap)
  - ◆ DDB에 비해 색상 테이블로가 해상도 정보 등을 가지고 있어 장치에 독립

Ex)

Read File :

```
BYTE* pData = ReadFile();
```

...

```
BITMAPFILEHEADER fh = {0};
```

```
BITMAPINFOHEADER ih = {0};
```

```
int bx = 0;
```

```
int by = 0;
```

```
BYTE* pPixel = NULL;
```

```
CopyMemory(&fh, pData , sizeof fh );
```

```
CopyMemory(&ih, pData + sizeof(BITMAPFILEHEADER) , sizeof ih );
```

```
bx = ih.biWidth;
```

```
by = ih.biHeight;
```

```
pPixel = m_pData + fh.bfOffBits;
```

```
SetDIBitsToDevice( hdc, ... , (BITMAPINFO *)&ih, DIB_RGB_COLORS);
```

...

```
StretchDIBits( hdc, ... , (BITMAPINFO *)&ih, DIB_RGB_COLORS);
```





- DIB to DDB
  - ◆ CreateDIBitmap() 함수 이용

Ex)

```
BITMAPFILEHEADER fh = {0};
```

```
BITMAPINFOHEADER ih = {0};
```

```
BYTE* pPixel= NULL;
```

```
CopyMemory(&fh, pData, sizeof fh );
```

```
CopyMemory(&ih, pData + sizeof(BITMAPFILEHEADER) , sizeof ih );
```

```
pPixel = pData + fh.bfOffBits;
```

```
HDC hdc = GetDC(hWnd);
```

```
hBit=CreateDIBitmap(hdc, &ih, CBM_INIT, pPixel, (BITMAPINFO *) &ih,  
DIB_RGB_COLORS);
```

```
ReleaseDC(hWnd, hdc);
```





- DIB(Device Independent Bitmap)

- ◆ DDB에 비해 색상 테이블로 해상도 정보 등을 가지고 있어 장치에 독립 Ex)

```
BYTE* pData = ReadFile();
```

```
...
```

```
BITMAPFILEHEADER fh = {0};
```

```
BITMAPINFOHEADER ih = {0};
```

```
int bx = 0;
```

```
int by = 0;
```

```
BYTE* pPixel = NULL;
```

```
CopyMemory(&fh, m_pData , sizeof fh );
```

```
CopyMemory(&ih, m_pData + sizeof(BITMAPFILEHEADER) , sizeof ih );
```

```
bx = ih.biWidth; by = ih.biHeight;
```

```
pPixel = m_pData + fh.bfOffBits;
```

```
SetDIBitsToDevice( hdc, ... , (BITMAPINFO *)&ih, DIB_RGB_COLORS);
```

```
...
```

```
StretchDIBits( hdc, ... , (BITMAPINFO *)&ih, DIB_RGB_COLORS);
```

- DIB Section

- ◆ 장치에 독립이면 출력할 때마다 장치에 의존하는 포맷변환이 필요.
- ◆ DIB Section는 응용프로그램이 즉시 쓸 수 있는 DIB 제공 → BitBlt, StretchBlt 가능





## ● Targa Format

- ◆ 포맷이 일정해서 초보자도 쉽게 파일을 해석 → 사용자가 파일 포맷을 구성해서 이용
- ◆ R, G, B, A 32bit 채널 지원
- ◆ Ex) LoadTGAFfile() 예제 참조

## ● 리소스에서 읽기

- ◆ HRSRC hrsc = FindResource(..., "TGA");
- ◆ HGLOBAL hglobal = LoadResource(..., hrsc);
- ◆ void\* pMemory = LockResource(hglobal);
- ◆ m\_pPixel = (BYTE\*)ReadTGAFFromMemory(...);
- ◆ UnlockResource(hglobal);
- ◆ FreeResource(hglobal);





- 이미지 파일들을 Packing하는 툴을 작성하시오.