HWND(Handle to a window)

- WinMain 내의 CreateWindow()를 사용하여 윈도우를 만들면 해당 윈도우를 사용할 수 있는 HWND형식의 핸들값 반환

- 윈도우를 생성한 프로그램 포함에서 다른 프로그램에서도 해당 윈도우 핸들을 알면 사용 가능

Desktop Window : 바탕화면 윈도우를 뜻한다

Top-Level Window

- 부모가 없거나, 데스크탑 윈도우가 부모인 윈도우를 뜻한다

- WS\_CHILD 스타일이 설정되지 않은 윈도우는 전부 탑레벨 윈도우이다.

- 자식 윈도우와는 다르게 탑레벨 윈도우는 화면 어디에도 디스플레이 될 수 있다

- 일반적으로 WS\_OVERLAPPED 또는 WS\_POPUP 스타일을 설정한다

Child Window

- WS\_CHILD 스타일이 설정된 창이며, 부모 윈도우가 존재한다

- 항상 부모의 클라이언트 영역에 유지된다

- 부모가 이동하면 자식 윈도우도 같은 방식으로 이동

- 부모가 파괴되면 자식 윈도우 또한 같이 파괴된다

- 메뉴, 버튼, 목록 상자 등의 컨트롤들은 자식 윈도우이다

WM\_COMMAND

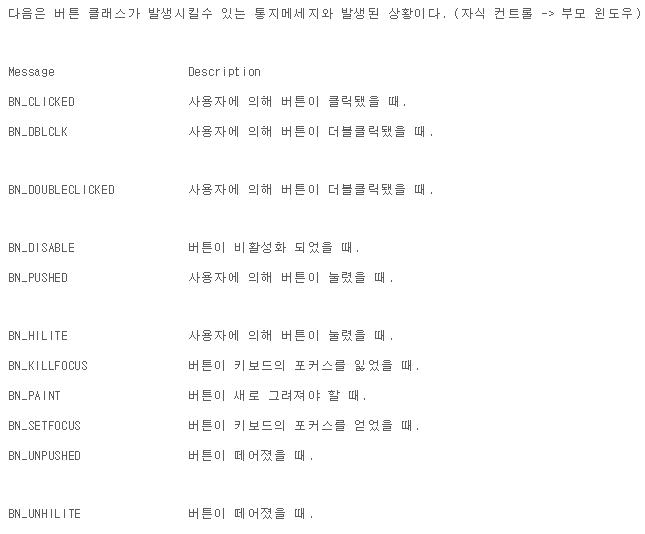
- CreateWindow로 생성된 버튼 혹은 단축키 등의 컨트롤을 누를 때 발생하는 메시지

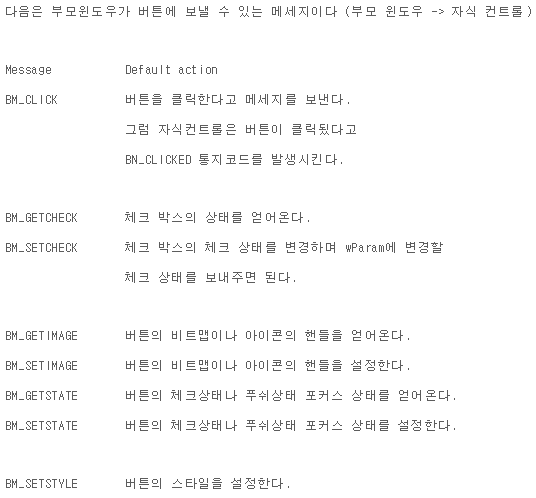
- 다양한 상황에서 발생하기 때문에 wParam과 lParam에 저장된 정보로 구분

- lParam에 저장되는 값은 컨트롤 핸들로, 메시지가 발생하면 발생한 컨트롤의 HWND 값이 저장

- wParam의 하위 16비트에는 Command ID가 포함되어 있다 -> LOWORD(wParam)

- wParam의 상위 16비트에는 통지 코드가 저장되어 있다 -> HIWORD(wParam)





HWND FindWindow(LPCTSTR lpClassName, LPCTSTR lpWindowName)

- Window Class 또는 Window Caption으로 윈도우 핸들을 얻는 함수, 탑레벨 윈도우만 탐색 가능(탑레벨 윈도우란 DESKTOP 윈도우를 부모로 갖는 윈도우를 말한다)

- 윈도우를 찾으면 그 핸들을 반환하며, 못 찾을 경우 NULL을 반환한다.(프로그램이 실행 중인지 아닌지 알아낼 수 있음)

- 지정할 수 있는 조건이 단순하기 때문에 원하는 윈도우를 정확하게 검색하기 힘들다(Window Caption은 고정되어 있지 않고 수시로 변경됨, 캡션이 같은 경우도 있기 때문에 신뢰하기 힘들다)

🡺 FindWindow()는 자신이 직접 만든 윈도우에만 제한적으로 사용 가능하며, 캡션 변경 불가능

HWND FindWindowEx(HWND hwndParent, HWND hwndChildAfter, LPCTSTR lpszClass, LPCTSTR lpszWindow)

- 특정 윈도우에 포함된 자식 윈도우를 탐색할 때 사용하는 함수

- NULL일 경우 데스크탑 윈도우의 자식 윈도우를 검색한다.

FindWindow & FindWindowEx의 한계

- 타이틀 바만 동일한 것의 윈도우 핸들을 반환하므로 저 함수들을 통해 가져온 것이 내가 작성한 프로그램이 아닐 수도 있다

- FindWindow는 타이틀 바 뿐만 아니라 클래스 이름으로도 윈도우 탐색 작업이 가능, 하지만 동일한 이름의 클래스와 윈도우명을 가진 윈도우가 다수 존재할 때, 그 중 첫번째 윈도우만 검색하고 탐색 작업을 중지함

BOOL EnumWindows(WNDENUMPROC lpEnumFunc, LPARAM lParam);

- 윈도우 운영체제 상의 모든 최상위 윈도우 핸들을 조사해서 콜백함수로 넘겨준다

- 마지막 최상위 창을 열거하거나 콜백 함수가 False를 반환할 때까지 수행 -> 사실상 끝까지

- 콜백함수에서는 넘겨받은 핸들을 이용하여 윈도우 핸들로 가능한 모든 작업을 할 수 있음

- WS\_CHILD를 포함한 자식 윈도우들은 열거하지 않고 제외

BOOL CALLBACK EnumWindowsProc(HWND hwnd, LPARAM lParam);

- EnumWindows함수로부터 최상위 윈도우 핸들을 받아서 처리해주는 콜백 함수

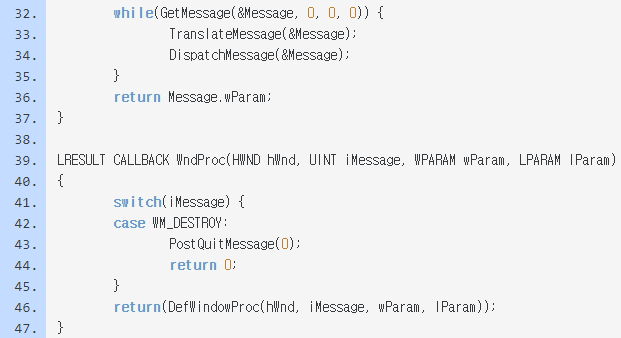
- 콜백 함수는 열거를 계속하려면 TRUE를 반환하고, 열거를 중지하려면 FALSE를 반환

콜백(Call Back) 함수란

- 함수의 호출 실행 시점을 프로그래머가 아닌 시스템에서 결정하는 함수

- Win API를 예로 들어보면, WndProc 함수를 직접 호출하는 것이 아니라 프로세스에 이벤트가 들어왔을 때 DispatchMessage 함수를 호출하여 이벤트를 처리하고 시스템에서 WndProc함수를 호출하게 된다.

- 따라서 호출의 주도권을 프로그래머가 아닌 시스템이 쥔다고 말할 수 있다.



BOOL IsIconic(HWND hwnd);

- 윈도우가 최소화되어 있으면 0이 아닌 값을 반환, 그렇지 않으면 0을 반환한다

BOOL IsZoomed(HWND hwnd);

- 윈도우가 최대화되어 있으면 0이 아닌 값을 반환, 그렇지 않으면 0을 반환한다

BOOL IsWindowVisible(HWND hwnd);

- 윈도우가 화면에서 보이는 상태인지 숨겨진 상태인지 조사한다

- 윈도우가 화면에 보이려면 부모 윈도우를 포함해서 WS\_VISIBLE 스타일을 갖고 있어야 한다.

- 보인다는 뜻은 WS\_VISIBLE 스타일이 설정되어 있다는 뜻이지, 실제로 화면에 나타나 있다는 뜻은 아님(TRUE를 반환하더라도 다른 윈도우에 완전히 가려져 있을 경우 숨겨져서 보이지 않음)

HWND GetActiveWindow();

- 함수를 호출하는 스레드와 연관된 액티브 윈도우에게 윈도우 핸들을 반환한다

- 현재 내 프로그램이 실행 중이고 윈도우가 많을 때, 어떤 윈도우가 활성화 상태인지 알고 싶음

HWND GetForegroundWindow();

- 사용자가 현재 작업중인 윈도우의 포그라운드 윈도우에게 윈도우 핸들을 반환한다

- 어떤 프로그램이든 관계없이 최상위 윈도우를 알고 싶음

BOOL GetClientRect(HWND hwnd, LPRECT lpRect);

- 대상 윈도우의 좌표를 구해 RECT 구조체에 채워서 반환(반환되는 값은 화면 좌표 기준)

BOOL GetWindowRect(HWND hwnd, LPRECT lpRect);

- 대상 윈도우의 좌표를 구해 RECT 구조체에 채워서 반환(반환되는 값은 작업영역(0, 0) 기준)