

# OpenAPI(WMCA SDK) 사용자 설명서

1. 라이브러리 구성
2. 제공함수
3. 응답 메시지
4. 파일 로깅

# 1. 라이브러리 구성

제공되는 라이브러리는 아래 파일들로 구성되어있습니다.

파일 또는 폴더	용도	설명
wmca.dll	통신 Library	dll 함수 인터페이스를 제공하는 파일로서 서비스 <b>주요파일</b> 입니다.
wmca.ini		접속 서버 및 포트등을 지정할 수 있는 파일이며 지정할 내용이 없을 경우 이 파일은 사용하지 않아도 됩니다.
sk*.dll		공인인증을 위해 SignKorea(코스콤 공인인증센터)에서 제공하는 라이브러리이며 <b>필수 파일들</b> 입니다.
nsldap32v11.dll		
sha256w32.dll		암호화 라이브러리이며 <b>필수 파일</b> 입니다
SAMPLES	사용 예제	시세조회/실시간시세요청/잔고조회/주문/정정/취소등 유형별 예제를 제공합니다.

## 2. 제공함수

### ● 호출규약

제공되는 Open API 함수는 Windows 표준 dll 방식(regular dll)으로 제공되며 함수 호출 규약(calling convention)은 **stdcall** 입니다.  
호출 규약이 맞지 않을 경우 정상 동작하지 않으니 주의하시기 바랍니다. (cdecl, fastcall, safecall 방식 아님)

### ● 서비스 사용 순서

TR 호출: **wmcaConnect(1회) > wmcaQuery(요청시마다) > wmcaDisconnect(1회)**

실시간 시세: **wmcaConnect(1회) > wmcaAttach(등록) > 실시간 시세 수신... > wmcaDetach(취소) > wmcaDisconnect(1회)**

### ● 함수 목록

(주요 함수 굵은 글꼴 표기)

함수 프로토타입	기능
BOOL <b>wmcaConnect</b> (HWND hWnd, DWORD msg, char mt,char ut,const char* szID,const char* szPW, const char* szCertPW)	접속 후 로그인(인증)
BOOL <b>wmcaDisconnect()</b>	접속 해제
BOOL <b>wmcaIsConnected()</b>	접속 여부 확인
BOOL <b>wmcaQuery</b> (HWND hWnd, int nTRID, constchar* szTRCode, const char* szInput, int nInputLen, int nAccountIndex)	서비스(TR) 호출
BOOL <b>wmcaAttach</b> (HWND hWnd, const char* szBCType, const char* szInput, int nCodeLen, int nInputLen)	실시간 등록
BOOL <b>wmcaDetach</b> (HWND hWnd, const char* szBCType, const char* szInput, int nCodeLen, int nInputLen)	실시간 취소
BOOL <b>wmcaDetachWindow</b> (HWND hWnd)	실시간 일괄 취소
BOOL <b>wmcaDetachAll()</b>	실시간 일괄 취소
BOOL <b>wmcaSetAccountIndexPwd</b> (char* szHashOut44, int nAccIndex, const char* szPwd)	계좌비밀번호 입력(암호화)
BOOL <b>wmcaSetOrderPwd</b> (char* szHashOut44, const char* szPwd)	거래비밀번호 입력(암호화)

- **BOOL** **wmcaConnect(HWND hWnd, DWORD msg, char MediaType, char UserType, const char\* szID, const char\* szPW, const char\* szCertPW);**  
서버 접속 및 인증 기능을 제공하는 함수로 인증 실패시 접속은 자동 해제되며 성공시 system tray에 wmca 아이콘이 나타납니다.

#### - 입력값

<b>HWND hWnd</b>	: 응답 메시지를 수신할 윈도우 핸들값입니다. 인증 성공 및 실패 여부가 다음에 지정하는 메시지 코드( <b>msg</b> )로 반환됩니다.
<b>DWORD msg</b>	: 수신하고자 하는 메시지 코드입니다. 기본값은 0이며 0 이외의 값을 지정할 경우 ( <b>WM_USER+지정값</b> ) 형태로 메시지가 수신됩니다.
<b>char MediaType</b>	: 매체유형을 입력합니다. QV_계좌일 경우 ‘P’ Namuh_계좌일 경우 ‘T’를 입력합니다. (기타 매체인 경우 개별 할당된 코드를 지정합니다)
<b>char UserType</b>	: 사용자유형을 입력합니다. QV_계좌일 경우 ‘1’ Namuh_계좌일 경우 ‘W’를 입력합니다. (기타 매체인 경우 개별 할당된 코드를 지정합니다)
<b>char* szID</b>	: 사용자 ID를 입력합니다. QV 또는 Namuh에 대한 온라인 약정 및 OpenAPI 서비스 이용 등록이 사전에 되어있어야 사용이 가능합니다
<b>char* szPW</b>	: 사용자 ID에 대한 비밀번호를 입력합니다.
<b>char* szCertPW</b>	: 공인인증서 비밀번호를 입력합니다.

#### - 반환 메시지

<b>CA_CONNECTED</b>	: 정상적으로 접속후 인증(로그인)이 성공할 경우 수신되며 계좌번호등이 전달됩니다. <u>수신된 정보 중 계좌번호의 순서는 계좌번호를 요구하는 호출 서비스(TR)에서 사용되므로 매우 중요합니다. (예제참조)</u>
<b>CA_RECEIVEMESSAGE</b>	: 인증 실패시 문자열 메시지가 전달됩니다.
<b>CA_SOCKETERROR</b>	: 서버 이상이나 네트워크 이상 등의 이유로 접속이 단절될 경우 반환되는 메시지입니다.

- **BOOL** **wmcaDisconnect();**

로그아웃 및 접속해제 함수이며 더 이상 서버와 통신을 원하지 않을 경우 호출합니다. 일반적인 경우 사용자 작성한 프로그램을 종료하는 시점에 호출합니다.

#### - 반환 메시지

<b>CA_DISCONNECTED</b>	: 로그아웃 및 접속해제 완료시 수신
------------------------	----------------------

## ● 접속 서버

사용할 수 있는 서버는 아래와 같습니다.

**wmca.nhqv.com (8200)**      <기본값>

### ● BOOL **wmcaIsConnected();**

접속 여부를 확인하는 함수입니다.

정상 로그인 후 통신장애 및 서버장애 등의 이유로 **wmca** 사용 중 접속이 끊길 수 있으며 이 때 메시지(CA\_SOCKETERROR, CA\_DISCONNECTED)로 끊겼음이 통보됩니다.  
이와는 별도로 프로그램 내부에서 현재 정상 서비스가 가능한 상태인지 확인이 필요할 경우 이 함수를 통해 접속상태를 확인할 수 있습니다.

#### -반환값

**TRUE** :정상 서비스 가능

**FALSE** :접속 해제 상태이며, 서비스 안됨

#### -반환 메시지

반환 메시지는 없습니다.

● **BOOL `wmcaQuery(HWND hWnd, int nTRID, const char* szTRCode, const char* szInput, int nInputLen, int nAccountIndex=0);`**

참고) 과거 제공되던 `wmcaTransact()` 함수는 이 함수로 대체되었으니 새로 프로그램 작성하실 경우 가급적 `wmcaQuery` 함수를 이용하시기 바랍니다.

시세조회, 주문, 잔고조회 등의 서비스(TR)를 요청할 경우 사용하는 함수로서 `wmca` 서비스 함수 중 가장 빈번하게 사용되는 함수입니다.

또한, 프로그램 작성시 실수 역시 잊은 부분입니다. (실시간 데이터 수신 서비스용 함수는 따로 존재하며 다음 페이지에서 설명함)

#### -입력값

`HWND hWnd`

:응답 메시지를 수신할 윈도우 핸들값입니다.

`int nTRID`

:TR 구분값입니다. 각 호출을 구분하기 위한 숫자값으로 사용자가 정한 임의의 정수값이 입력 가능합니다.(0 ~ 4294967296)

일종의 사용자 지정 태그(tag)값으로서 사용자가 입력한 값이 그대로 반환됩니다.

예) 현재가 조회 TR을 10회 호출할 경우 TRID에 10개의 서로 다른 값을 입력하면 각각의 호출을 모두 구분할 수 있음

`char* szTrCode`

:서비스 코드(TR)입니다. 원하는 서비스(시세조회, 주문등)에 대한 식별코드 5자리를 입력합니다. (서비스 코드는 별도 자료에 안내됨)

예) c1101, s8201,...

`char* szInput`

:호출하고자 하는 서비스(TR)의 입력값입니다. 서비스별로 입력값은 다르므로 별도로 제공되는 TR I/O 구조를 확인하시기 바랍니다.

`int nInputLen`

:입력문자열에 대한 길이를 입력합니다.(byte 단위)

`int nAccountIndex`

:호출하려는 서비스가 계좌번호를 요구할 경우 해당 계좌번호의 인덱스(순서)를 지정합니다.

**0** 계좌번호를 요구하지 않는 TR - 일반 시세조회에 해당합니다.

**기타값** 로그인시 수신한 계좌목록 인덱스(순서) 사용 TR - 계좌번호를 요구하는 잔고조회/주문등에 해당합니다

#### -반환 메시지

<구체적인 방식은 예제참조>

`CA_RECEIVEDATA`

:서비스 결과 수신

`CA_RECEIVEMESSAGE`

:서비스 진행 상태, 서비스가 정상 처리되지 않을 경우 이유를 출력함

`CA_RECEIVECOMPLETE`

:서비스 정상 완료

`CA_RECEIVEERROR`

:서비스 실패

● **BOOL `wmcaAttach(HWND hWnd, const char* szBCType, const char* szInput, int nCodeLen, int nInputLen);`**

실시간 패킷을 등록 요청하는 함수입니다. 수신 받고자 하는 실시간 서비스(BC) 코드 및 해당 서비스에서 요구하는 값(예:종목코드)을 입력하여 서비스를 신청하면 명시적으로 등록 해제를 하거나 프로그램이 종료될 때까지 실시간으로 데이터가 지속적으로 수신됩니다.

예) 특정 주식종목에 대해서 실시간으로 체결가 및 호가등을 요청해 놓으면 해당 종목에 대해 시세 변동이 발생할 때마다 실시간으로 지정 윈도우(hWnd)로 전달됩니다.

-입력값

**HWND hWnd** :실시간 데이터를 수신할 윈도우 핸들값입니다. 여러 윈도우를 사용할 경우 각 윈도우 핸들값으로 호출하면 개별 윈도우로 모두 전달됩니다.

**char\* szBCType** :실시간 서비스 코드(BC)입니다. 원하는 서비스(실시간 시세)에 대한 식별 코드 2자리를 입력합니다. (서비스 코드는 별도 안내됨)  
예) ‘j8’(주식체결가), ‘h1’(주식호가), ...

**char\* szInput** :해당 서비스가 요구하는 입력값입니다. 입력값이 여러 개일 경우 **구분자없이 연속으로** 입력합니다.  
예) 3개 종목(SK하이닉스, NH투자증권, 삼성증권)에 대한 종목코드를 입력하는 경우, ‘000660005940005930’

**int nCodeLen** :입력값 개별 길이입니다. (byte 단위)      예) 주식종목코드를 요구할 경우 종목코드 개별길이는 6자리이므로 ‘6’을 입력합니다.

**int nInputLen** :입력값 전체 길이입니다. (byte 단위)      예) 주식종목 10개를 입력할 경우 6자리x10종목=60 자리이므로 ‘60’을 입력합니다.

-반환 메시지 <구체적인 방식은 예제참조>

**CA\_RECEIVESISE** :실시간 데이터 수신

● **BOOL `wmcaDetach(HWND hWnd, const char* szTran, const char* szInput, int nCodeLen, int nInputLen);`**

실시간 패킷 수신이 더 이상 필요하지 않을 경우(등록취소) 사용하는 함수입니다. `wmcaAttach()` 함수의 반대 기능을 합니다.

필요하지 않은 실시간 시세에 대해서 등록 취소를 하지 않을 경우 불필요한 데이터 송수신으로 인해 프로그램 성능 저하가 올 수 있으므로 더 이상 필요없는 실시간 데이터는 이 함수를 통해 등록취소를 합니다.

-입력값

**HWND hWnd**

:응답 메시지를 수신할 윈도우 핸들값

**char\* szTran**

:서비스 코드(BC). 등록취소를 원하는 서비스(실시간 시세)에 대한 실별코드 2자리를 입력함. 서비스 코드는 별도 안내됨.

예) ‘j8’(주식체결가), ‘h1’(주식호가), ...

**char\* szInput**

:등록취소를 원하는 서비스가 요구하는 입력값. 입력값이 여러 개일 경우 구분자없이 연속으로 입력함.

예) 2개 종목(하이닉스,NH투자증권,삼성증권)에 대한 종목코드를 입력하는 경우, ‘000660005940005930’

**int nCodeLen**

:입력값 개별 길이입니다. (byte 단위)

예) 주식종목코드를 요구할 경우 종목코드 개별길이는 6자리이므로 ‘6’을 입력합니다.

**int nInputLen**

:입력값 전체 길이입니다. (byte 단위)

예) 주식종목 10개를 입력할 경우 6자리x10종목=60 자리이므로 ‘60’을 입력합니다.

-반환 메시지

반환 메시지는 없습니다.

●      BOOL      **wmcaDetachWindow(HWND hWnd);**

지정한 윈도우 핸들로 등록된 실시간 서비스(BC)를 일괄 취소합니다.

종목을 개별적으로 취소하지 않고 해당 원도우 핸들로 등록한 실시간 서비스를 일괄적으로 취소하고자 할 경우 사용합니다.

### - 입력값

**HWND hWnd** :취소하고자 하는 윈도우 핸들값을 입력.

●      BOOL      **wmcaDetachAll();**

등록된 모든 실시간 서비스(BC)를 일괄 취소합니다.

종목별 또는 원도우 단위가 아닌 현재까지 등록한 모든 실시간 서비스를 취소합니다.

●      **BOOL      `wmcaSetAccountIndexPwd(char* szHashOut44, int nAccIndex, const char* szPwd);`**

사용자가 지정한 '계좌 비밀번호'를 (복호화할 수 없는) 안전한 형태의 hash 비밀번호로 변경합니다.  
(서버로는 평문이 아닌 hash 비밀번호가 전달됨)

- 입력값

`char* szHashOut44`      :암호화된(hash) 비밀번호 출력 변수를 지정합니다  
`int nAccIndex`      :대상 계좌번호 인덱스를 지정합니다  
`const char* szPwd`      :암호화하려는 평문 비밀번호를 지정합니다

●      **BOOL      `wmcaSetOrderPwd(char* szHashOut44, const char* szPwd);`**

사용자가 지정한 '거래 비밀번호'를 (복호화할 수 없는) 안전한 형태의 hash 비밀번호로 변경합니다.  
(서버로는 평문이 아닌 hash 비밀번호가 전달됨)

- 입력값

`char* szHashOut44`      :암호화된(hash) 비밀번호 출력 변수를 지정합니다  
`const char* szPwd`      :암호화하려는 평문 비밀번호를 지정합니다

## 3. 응답 메시지

CA_CONNECTED	=WM_USER+110;	//접속 및 로그인 성공후 수신되며 서비스 이용이 가능함을 의미합니다
CA_DISCONNECTED	=WM_USER+120;	//통신 연결이 끊긴 경우 발생합니다
CA_SOCKETERROR	=WM_USER+130;	//네트워크 장애등의 이유로 통신 오류 발생할 경우 수신되는 메시지이며 접속환경 점검이 필요합니다
CA_RECEIVEDATA	=WM_USER+210;	//wmcaQuery() 함수 호출에 따른 결과 수신시 발생합니다
CA_RECEIVESISE	=WM_USER+220;	//wmcaAttach() 함수 호출에 따른 실시간 데이터 수신시 발생합니다
CA_RECEIVEMESSAGE	=WM_USER+230;	//요청한 서비스에 대한 처리상태가 문자열 형태로 수신되며 정상처리 및 처리실패등의 각 상태를 보여줍니다.
CA_RECEIVECOMPLETE	=WM_USER+240;	//요청한 서비스에 대한 처리가 정상 완료될 경우 수신됩니다
CA_RECEIVEERROR	=WM_USER+250;	//요청한 서비스에 대한 처리가 실패할 경우 수신되며 사용자가 잘못된 값을 입력하는 등의 이유로 발생합니다.

## 4. 파일 로깅

wmca를 이용하여 프로그램을 작성한 후 서버와 주고 받는 값을 확인할 때 사용합니다.

wmca.ini 파일에 아래와 같이 입력 후 재실행하면 실행 프로그램 폴더내에 ‘wmca.log’ 파일이 생성되며 송수신 자료가 기록됩니다.

binary 형태의 데이터도 기록되므로 모든 자료가 눈으로 식별되지는 않을 수 있습니다.

로그사용=Y

주의) 파일 기록으로 인해 프로그램 성능이 현저하게 저하되며 개인정보가 노출될 수 있으므로 테스트 용도로만 제한하여 사용하시기 바랍니다.