

키움 OpenAPI+ 개발가이드

1. 설치 및 개발환경

- 1.1 설치하기
- 1.2 개발환경

2. 접속하기

- 2.1 로그인 및 버전처리
- 2.2 모의투자 접속

3. KOA Studio 활용법

- 3.1 KOA Studio 개념 및 특징
- 3.2 KOA Studio 설치
- 3.3 KOA Studio 구성 및 사용법

4. 프로그램 구성파일

- 4.1 라이브러리 구성파일
- 4.2 프로그램 데이터 파일
- 4.3 시스템 데이터 파일

5. OpenAPI 명세서

- 5.1 OpenAPI 컨트롤
- 5.2 OpenAPI식별자
- 5.3 OpenAPI 메소드
- 5.4 OpenAPI 컨트롤 이벤트
- 5.5 Event 발생흐름

6. 제어 함수 상세명세서

- 6.1 메소드(Method)
- 6.2 이벤트(Event)

7. 에러코드표

8. 실시간 FID

- 8.1 주식시세
- 8.2 주식체결
- 8.3 주식우선호가
- 8.4 주식호가잔량

- 8.5 선물시세
- 8.6 선물호가잔량
- 8.7 선물이론가
- 8.8 옵션시세
- 8.9 옵션호가잔량
- 8.10 옵션이론가
- 8.11 주식옵션시세
- 8.12 주식옵션호가잔량
- 8.13 주식옵션이론가
- 8.14 선물옵션우선호가
- 8.15 업종지수
- 8.16 업종등락
- 8.17 장시작시간
- 8.18 투자자ticker
- 8.19 주문체결
- 8.20 잔고
- 8.21 신용잔고
- 8.22 주식시간외호가
- 8.23 주식당일거래원

9. 키움 Open API+ 추가 기능 가이드

- 9.1 조건검색 실시간 가이드
- 9.2 실시간 등록/해지 기능 가이드
- 9.3 대량 DATA(차트조회) 수신

10. 기타

- 10.1 신용주문 가이드

1. 설치 및 개발환경

1.1 설치하기

■ 모듈설치

키움증권 홈페이지>트레이딩채널>Open API>키움 Open API+>서비스 소개>'키움 Open API+ 모듈 다운로드' 버튼을 통해 다운로드 후 설치

키움 Open API+

☆ 즐겨찾기

서비스 소개 | 서비스 사용 등록/해지 | FAQ | 자료실 | 고객센터의 게시판

■ 서비스 소개

- 키움 Open API+ 란?**
당사가 제공하는 Open API 서비스 명으로, 고객님이 직접 프로그래밍한 투자전략을 당사가 제공하는 모듈에 연결하여, 시세조회/잔고조회/주문 등을 할 수 있도록 제공하는 서비스입니다.
기존 키움 Open API에 실시간 조건검색 제공, 실시간 등록/해지, DATA 수신 속도 개선 등 강력한 추가 기능을 제공하여 새롭게 오픈하였습니다.
- 가능 상품**
주식/Kospi200선물/Kospi200옵션/주식선물
- 적용 수수료율**
영웅문4(HTS)와 동일
- 키움 Open API+ 사용 요건**
당사 계좌를 보유하고 HTS ID를 연결하신 고객님께서 모두 이용 가능하며, '서비스 사용등록' 탭에서 사용 등록 후 바로 이용 가능합니다.
- 키움 Open API+ 사용절차**

Step 1

Open API 서비스를 사용 신청합니다

사용 신청하러 가기 >

Step 2

Open API 설치

키움 Open API+ 모듈 다운로드 >

Step 3

OCX 탑재 프로그램 제작

개발가이드 바로가기 >
KOA Studio 다운로드 >
교육 VOD 바로가기 >
자료실 바로가기 >

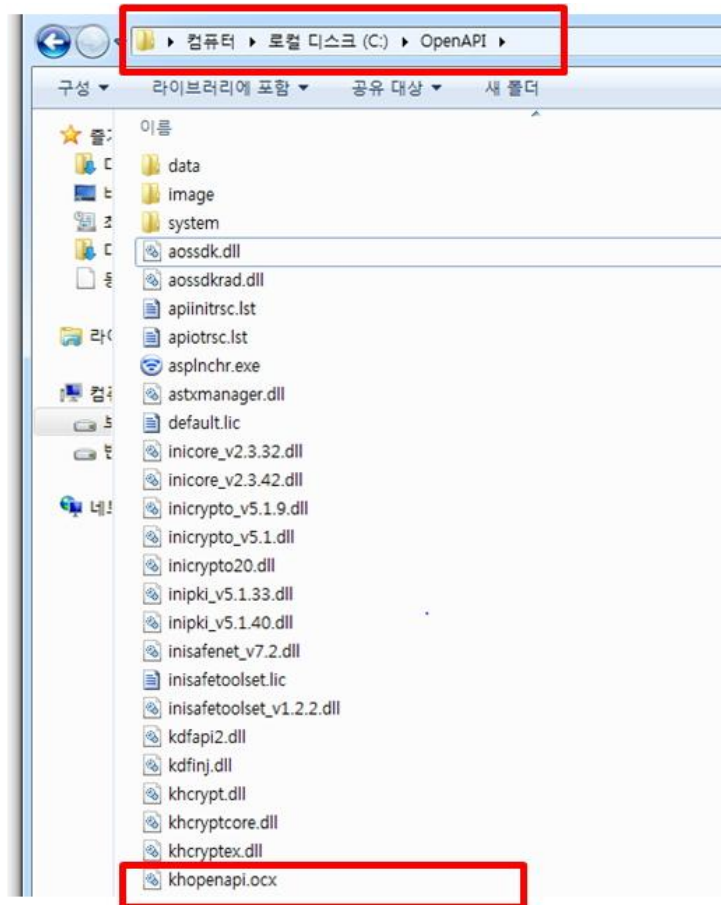
Step 4

테스트 및 디버깅

상시 모의투자 신청하러 가기 >

■ 디렉토리

- 모듈을 설치하면 그림과 같은 디렉토리가 생성됩니다. OpenAPI 의 인터페이스는 OCX컨트롤 방식으로 지원하며, 프로그램이 설치될 때 OCX가 레지스트리에 등록됩니다. (개발을 하실 때 다시 등록하지 않으셔도 됩니다.)
- OpenAPI 사용자들께서 자신의 프로그램을 개발시에 필요한 OCX컨트롤을 포함하여 실행시에 필요한 기타 모듈파일들을 프로그램 로그인시에 다운로드를 통해서 자동으로 업데이트 됩니다.



1.2 개발환경

- 키움 Open API 는 프로그램 개발 확장성을 고려하여 OCX 컨트롤로 제작 지원합니다.
- 사용자 편의에 따라 VB, 엑셀, 웹기반, MFC 등으로 프로그램 제작이 가능합니다.
- 데이터 요청 및 수신은 TR 서비스명을 검색한 후 OCX 를 통해 API 함수를 호출하는 방식으로 진행합니다.

2. 접속하기

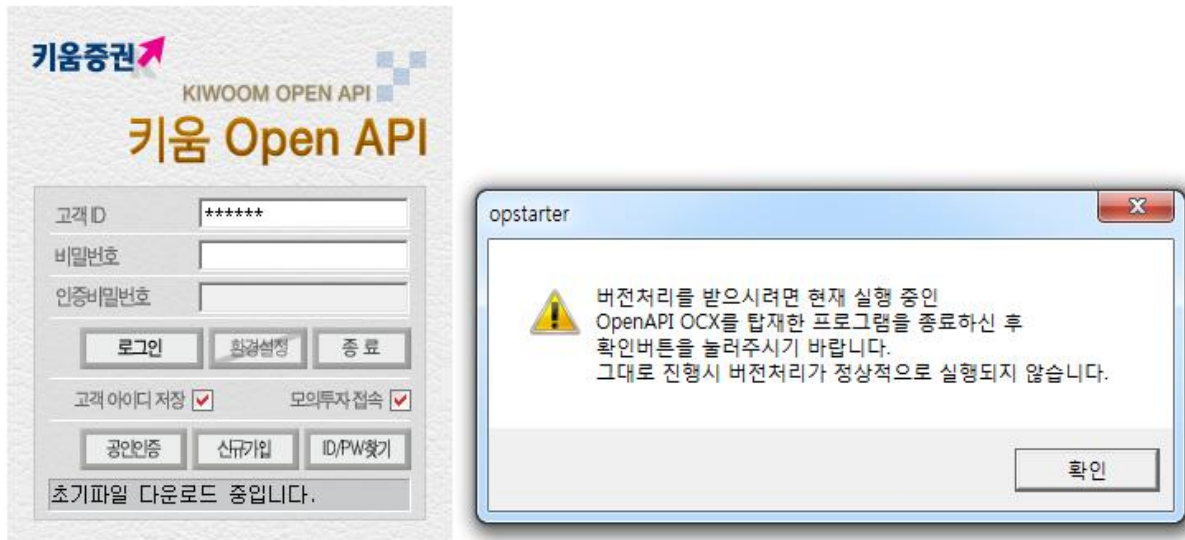
2.1 로그인 및 버전처리

키움 Open API+는 로그인창을 제공합니다.(공통)

제작한 프로그램을 통해 로그인 시 아래 절차를 참고하시면 됩니다.

- 로그인 시 로그인 창을 실행한 프로그램(Open API OCX를 탑재한 프로그램)이 열려있는 경우 버전처리 및 로그인이 진행 되지 않습니다.

- 아래 절차에 따라 로그인 및 버전처리를 진행하여 주시기 바랍니다.
- 프로그램 및 KOA스튜디오 접속>ID/PW/인증PW 입력 후 로그인 및 버전처리 진행>버전처리 안내 팝업 생성>팝업은 그대로 둔 상태에서 프로그램 및 KOA스튜디오 등 API와 관련 된 모든 프로그램 종료>버전 처리 종료 후 다시 프로그램 실행
- 실서버 접속의 경우 '모의투자 접속' 칸을 해제 한 후 접속하시면 됩니다.



2.2 모의투자 접속

- 모의투자 접속을 위해서는 로그인창 우측하단에 모의투자접속에 체크한 후 모의투자 ID 및 PW를 이용하여 접속하시면 됩니다.



- 모의투자신청은 키움증권 홈페이지>모의/실전투자>상시모의투자 메뉴에서 신청가능합니다.

3. KOA Studio 활용하기

3.1 KOA Studio 개념 및 특징

KOA Studio는 개발 편의를 위해 제공하는 TR의 목록과 정보를 간편하게 확인 할 수 있으며, 입력값 및 출력값을 확인 할 수 있는 화면을 제공하는 프로그램 입니다.

- 특징
 - 모의투자 서버로만 접속가능
 - 개발중인 자신의 프로그램상의 수신데이터와 KOA Studio 상의 수신데이터 비교 가능
 - 서비스 TR의 input 값 참고 가능
 - 실시간 타입과 FID값 참고 가능

3.2 KOA Studio 설치 및 로그인

- 설치

키움증권 홈페이지>트레이딩채널>Open API>키움 Open API+>서비스 소개>'키움 Open API+ 모듈 다운로드' 버튼을 통해 다운로드 후 설치

키움 Open API+

★ 즐겨찾기

서비스 소개 | 서비스 사용 등록/해지 | FAQ | 자료실 | 고객센터의 게시판

■ 서비스 소개

- 키움 Open API+ 란?
당사가 제공하는 Open API 서비스 명으로, 고객님이 직접 프로그래밍한 투자전략을 당사가 제공하는 모듈에 연결하여, 시세조회/잔고조회/주문 등을 할 수 있도록 제공하는 서비스 입니다.
기존 키움 Open API에 실시간 조건검색 제공, 실시간 등록/해지, DATA 수신 속도 개선 등 강력한 추가 기능을 제공하여 새롭게 오픈하였습니다.
- 가능 상품
주식/Kospi200선물/Kospi200옵션/주식선물
- 적용 수수료율
영웅문4(HTS)와 동일
- 키움 Open API+ 사용 요건
당사 계좌를 보유하고 HTS ID를 연결하신 고객님께서는 모두 이용 가능하며, '서비스 사용등록' 탭에서 사용 등록 후 바로 이용 가능합니다.
- 키움 Open API+ 사용절차

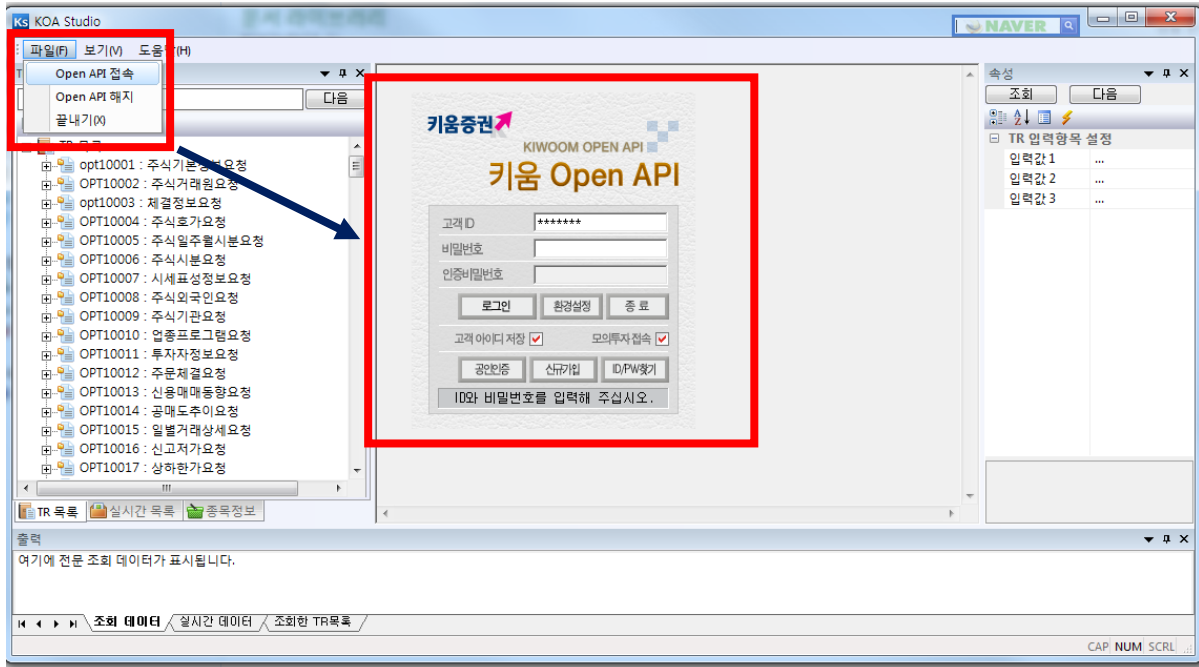
Step 1 Open API 서비스를 사용 신청합니다 > Step 2 Open API 설치 > Step 3 OCX 탑재 프로그램 제작 > Step 4 테스트 및 디버깅

사용 신청하러 가기 > 키움 Open API+ 모듈 다운로드 > 개발가이드 바로가기 > 상시 모의투자 신청하러 가기 >

KOA Studio 다운로드 > 교육 VOD 바로가기 > 자료실 바로가기 >

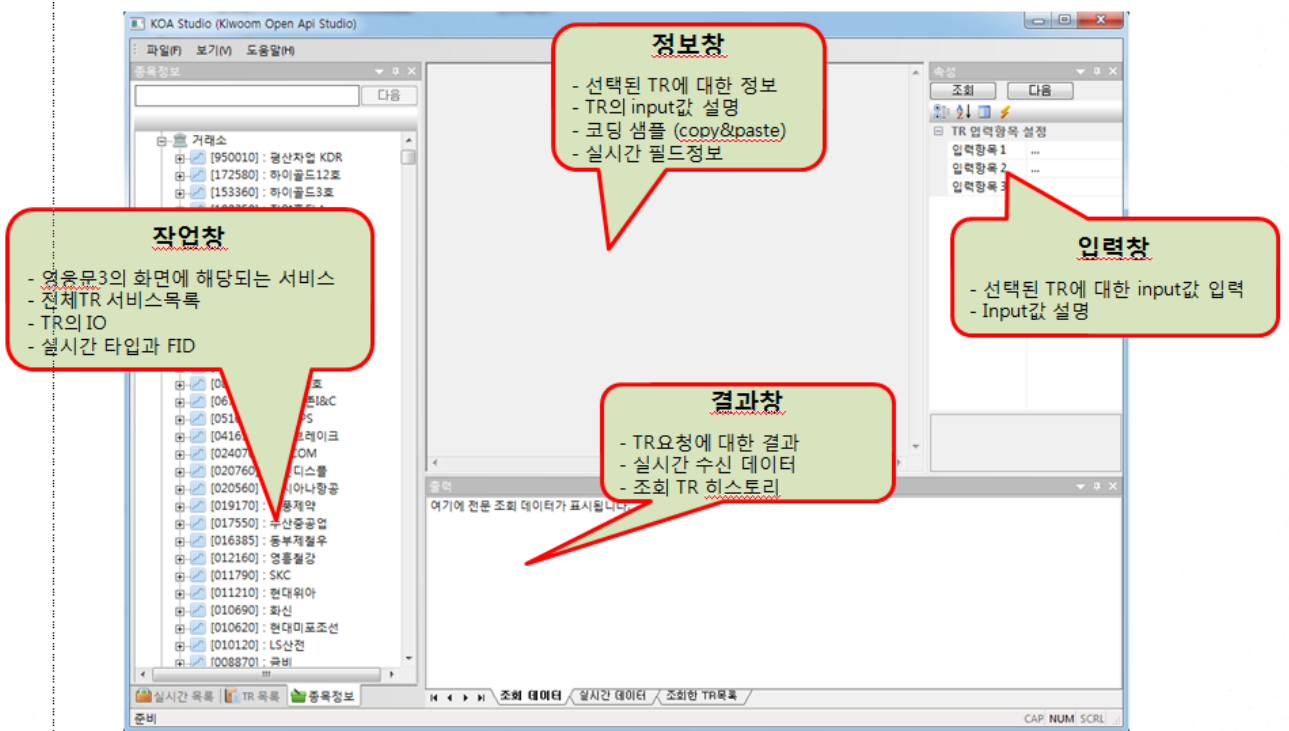
■ 로그인

- KOA Studio >파일(F)>Open API 접속을 클릭하시면 로그인창이 생성 됩니다.
- KOA Studio는 모의투자접속만 가능합니다.(디폴트 처리)



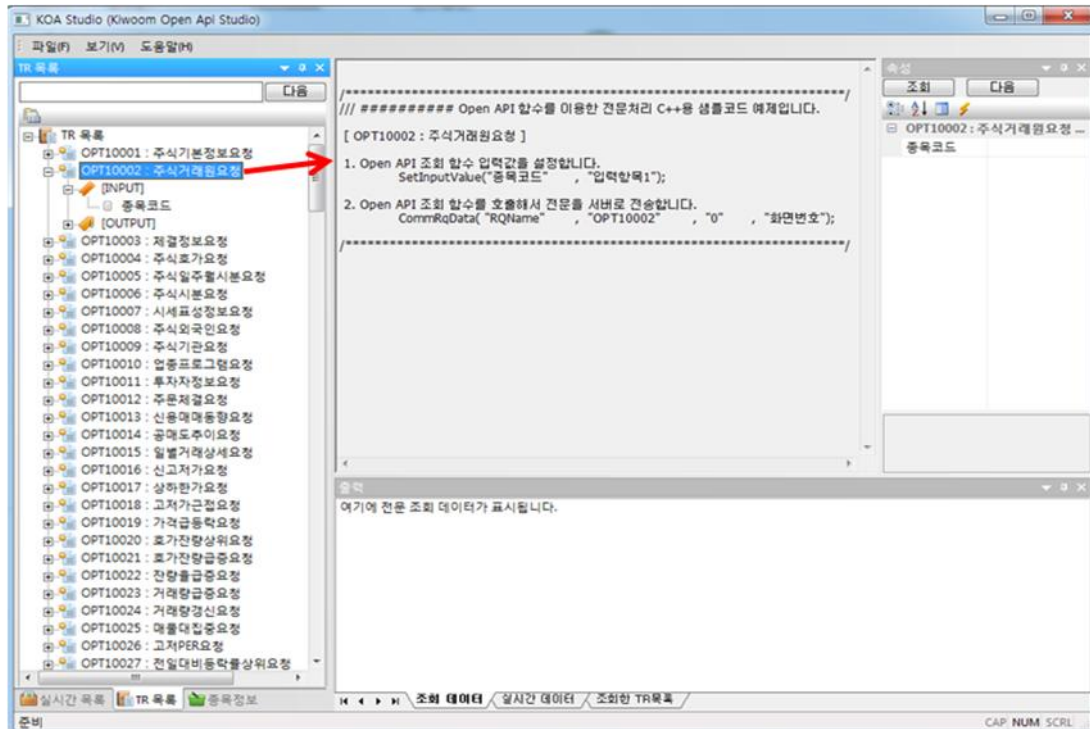
3.3 KOA Studio 구성 및 사용법

- 작업창/정보창/입력창/결과창으로 구성 됩니다.

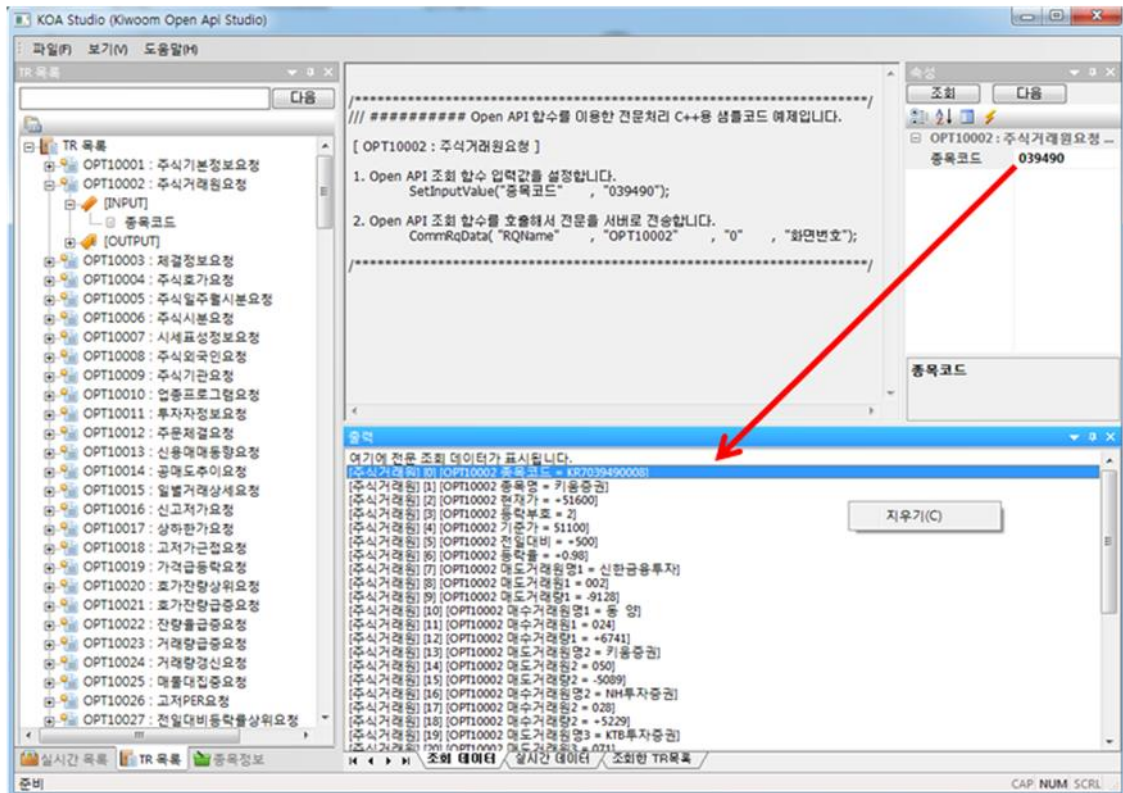


■ TR조회 방법

- TR목록을 클릭하시면 제공 TR리스트와 샘플코드 예제(C++ 기준)이 보여집니다.



- INPUT 값을 입력하면 출력 값이 창에 표시 됩니다.



-
- KOA Studio (Kiwon Open Api Studio)
- 파일(F) 보기(V) 도움말(H)
- TR 목록
- 다음
- TR 목록
- OPT10001 : 주식기본정보요청
 - OPT10002 : 주식거래현황요청
 - [INPUT]
 - 종목코드
 - [OUTPUT]
 - OPT10003 : 체결정보요청
 - OPT10004 : 주식호가요청
 - [INPUT]
 - [OUTPUT]
 - OPT10005 : 주식일주환시분요청
 - OPT10006 : 주식시분요청
 - OPT10007 : 시세교정정보요청
 - OPT10008 : 주식외국인요청
 - OPT10009 : 주식기관요청
 - OPT10010 : 업종프로그램요청
 - OPT10011 : 투자자정보요청
 - OPT10012 : 주문제결요청
 - OPT10013 : 신용매매종량요청
 - OPT10014 : 공매도주요청
 - OPT10015 : 일별거래상세요청
 - OPT10016 : 신고지요청
 - OPT10017 : 상하한가요청
 - OPT10018 : 고저가근접요청
 - OPT10019 : 가격급동확인요청
 - OPT10020 : 호가잔량상위요청
 - OPT10021 : 호가잔량중요요청
 - OPT10022 : 잔량분급중요요청
 - OPT10023 : 거래량급중요요청
 - OPT10024 : 거래량경신요청
 - OPT10025 : 매물대집중요청
- /// ##### Open API 함수를 이용한 건문처리 C++용 샘플코드 예제입니다.
[OPT10004 : 주식호가요청]
1. Open API 조회 함수 입력값을 설정합니다.
SetInputValue("종목코드" , "005930");
 2. Open API 조회 함수를 호출해서 건문을 서버로 전송합니다.
CommRqData("RQName" , "OPT10004" , "0" , "화면번호");
- ///
- 속성
- 조회 다음
- OPT10004 : 주식호가요청 설정
- 종목코드 005930
- 종목코드
- 출력
- 여기에 조회한 전무목록(TR목록)이 최근순으로 표시됩니다.
- OPT10004 : 주식호가요청
OPT10002 : 주식거래현황요청
- 실행시간 목록 TR 목록 등록정보
- 준비
- 조회 데이터 실행시간 데이터 조회된 TR목록
- CAP. NUM. SCRL

- The screenshot displays the KOA Studio environment. The central editor contains a C++ header file for 'Real Name : 주식시세'. The code defines a 'Real Name' structure with fields for stock codes, prices, and volumes, and includes a list of stock codes. The left sidebar shows a project tree with folders for '실시간 목록' (Real-time List) and '종목정보' (Stock Information). The bottom status bar indicates the current view is '실시간 데이터' (Real-time Data).

4. 프로그램 구성 파일

4.1 라이브러리 구성 파일

- 파일경로 : [SYSTEM DRIVE:]/Program Root/

구 분	파일명	설 명	비고
DLL	KHOpenAPI.ocx	OpenApi 실행 모듈	
	OPCommApi.dll	통신 모듈	
	OPComms.dll	서버와의 TCP/IP 연결 모듈	
	OPSecurity.dll	통신 암호화 모듈	
	libsadlx.dll		
	nsldap32v11.dll		
	ZIPDLL.DLL	압축 모듈	

4.2 프로그램 데이터 파일

- 파일경로 : [SYSTEM DRIVE:]/Program Root/Data

구 분	파 일 명	설 명	비고
데이터파일	OPTxxxxx.enc OPWxxxxx.enc	통신용 데이터 파일 (약 100여개) 상세 내역은 첨부 자료 확인	

4.3 시스템 데이터 파일

- 파일경로 : [SYSTEM DRIVE:]/Program Root/System

구 분	파 일 명	설 명	비고
시스템	jongacc.dat	계좌 관련 데이터 파일	
	OPComms.ini	통신 속성 관련 데이터 파일	

5. OpenAPI 명세서

5.1 OpenAPI 컨트롤

- OpenApi는 ActiveX Control (OCX) 형태의 OLE 컨트롤을 기반으로 설계 및 구현되어있다.
일반적인 OCX와 동일하게 윈도우 운영체제에 컨트롤을 등록 해야 사용 할 수 있다.

	내용	비고
컨트롤 모듈	KHOpenApi.OCX	
라이브러리 형태	OCX - OLE Control Extension, DLL	
호 환 성	Microsoft Windows XP SP3 이상	
개발 환경	Mocrosoft Visual C++ 2010	
버 전		

5.2 OpenAPI 식별자

- OpenAPI 컨트롤의 인터페이스 식별자(Interface ID : GUID)는 유일성(Unique)을 지니며 이중으로 등록 되어서는 안된다.
- OpenAPI 컨트롤의 생성 및 데이터 상호 교환을 위한 컨트롤 인터페이스 식별자는 다음과 같다.

구 분	식별자	내용
Control	A1574A0D-6BFA-4BD7-9020-DED88711818D	컨트롤 클래스
Dispatch	CF20FBB6-EDD4-4BE5-A473-FEF91977DEB6	프로퍼티/메소드 제어
Event	7335F12D-8973-4BD5-B7F0-12DF03D175B7	이벤트 제어
Module	6D8C2B4D-EF41-4750-8AD4-C299033833FB	컨트롤 모듈

5.3 OpenAPI 메소드

- OpenAPI 제어 및 편집을 위한 인터페이스 메소드(Interface Method) 목록은 다음과 같다.

ID	타입	이름	설명
1	LONG	CommConnect	로그인 윈도우를 실행한다.
2	void	CommTerminate	더 이상 지원하지 않는 함수(기능없음)
3	LONG	CommRqData	통신 데이터를 송신한다.
4	BSTR	GetLoginInfo	로그인 정보를 반환한다.
5	LONG	SendOrder	주식주문 Tran을 송신한다.
6	LONG	SendOrderCredit	주식 신용주문 Tran을 송신한다.
7	void	SetInputValue	
8	LONG	SetOutputFID	

9	BSTR	CommGetData	이 함수대신 용도에 맞는 전용 함수를 사용할 것 조회 정보 요청 : GetCommData() 실시간정보 요청 : GetCommRealData() 체결정보 요청 : GetChejanData()
10	void	DisconnectRealData	리얼 시세를 끊는다.
11	LONG	GetRepeatCnt	수신 받은 데이터의 반복 개수를 반환한다.
12	LONG	CommKwRqData	관심종목을 조회 한다.
13	BSTR	GetAPIModulePath	OpenApi 모듈이 설치되어 있는 디렉토리를 반환한다.
14	BSTR	GetCodeListByMarket	장구분별 종목코드 리스트를 반환한다.
15	LONG	GetConnectState	통신 접속 상태를 반환한다.
16	BSTR	GetMasterCodeName	종목코드의 종목명을 반환한다.
17	LONG	GetMasterListedStockCnt	종목코드의 상장주식수를 반환한다.
18	BSTR	GetMasterConstruction	종목코드의 감리구분을 반환한다.
19	BSTR	GetMasterListedStockDate	종목코드의 상장일을 반환한다.
20	BSTR	GetMasterLastPrice	종목코드의 전일가를 반환한다.
21	BSTR	GetMasterStockState	종목코드의 종목상태를 반환한다.
22	LONG	GetDataCount	레코드의 반복개수를 반환한다.
23	BSTR	GetOutputValue	수신 데이터를 반환한다.
24	BSTR	GetCommData	수신 데이터를 반환한다.
25	BSTR	GetCommRealData	실시간 데이터를 반환한다.
26	BSTR	GetChejanData	체결잔고 데이터를 반환한다.
27	BSTR	GetThemeGroupList	테마 코드와 그룹의 리스트를 반환한다.
28	BSTR	GetThemeGroupCode	테마코드에 해당하는 종목코드를 반환한다.
29	BSTR	GetFutureList	지수선물 코드 리스트를 반환한다.
30	BSTR	GetFutureCodeByIndex	지수선물 코드를 반환한다.
31	BSTR	GetActPriceList	지수옵션 행사가 리스트를 반환한다.
32	BSTR	GetMonthList	지수옵션 월물 리스트를 반환한다.
33	BSTR	GetOptionCode	지수옵션 종목코드를 반환한다.
34	BSTR	GetOptionCodeByMonth	같은 행사가 다른 월물의 종목코드를 반환한다.
35	BSTR	GetOptionCodeByActPrice	같은 월물 다른 행사가의 종목코드를 반환한다.
36	BSTR	GetSFutureList	주식선물 코드 리스트를 반환한다.
37	BSTR	GetSFutureCodeByIndex	주식선물 코드를 반환한다.
38	BSTR	GetSActPriceList	주식옵션 행사가를 반환한다.
39	BSTR	GetSMonthList	주식옵션 월물을 반환한다.
40	BSTR	GetSOptionCode	주식옵션 코드를 반환한다.
41	BSTR	GetSOptionCodeByMonth	주식옵션 월물만 변경된 코드를 반환한다.
42	BSTR	GetSOptionCodeByActPrice	주식옵션 행사가만 변경된 코드를 반환한다.
43	BSTR	GetSFOBasisAssetList	주식선물 기초자산코드와 이름을 반환한다.
44	BSTR	GetOptionATM	지수옵션 ATM을 반환한다.
45	BSTR	GetSOptionATM	주식옵션 ATM을 반환한다.

46	BSTR	GetBranchCodeName	회원사 코드와 이름을 반환한다.
47	BSTR	CommInvestRqData	투자자정보를 조회한다.
48	LONG	SetInfoData	아이디별 자동로그인이 접속이 필요할 때 사용한다.
49	LONG	SetRealReg	실시간 등록을 한다.
50	Void	SetRealRemove	종목별 실시간 해제 (SetRealReg로 등록한 종목만 해제 가능)
51	long	GetConditionLoad	서버에 저장된 사용자 조건식을 가져온다.
52	BSTR	GetConditionNameList	조건검색에 필요한 조건명 리스트를 받아온다.
53	void	SendCondition	조건검색 조회TR을 송신한다.
54	void	SendConditionStop	실시간 조건검색을 중지한다.
55	Variant	GetCommDataEx	차트 조회한 데이터 전부를 배열로 받아온다.

5.4 OpenAPI컨트롤 이벤트

- 차트 컨트롤에서 발생하는 컨트롤 이벤트 목록은 다음과 같다.

ID	타입	이름	설명
1	Void	OnReceiveTrData	Tran 수신시 이벤트
2	Void	OnReceiveRealData	실시간 시세 이벤트
3	Void	OnReceiveMsg	수신 메시지 이벤트
4	Void	OnReceiveChejanData	주문 접수/확인 수신시 이벤트
5	Void	OnEventConnect	통신 연결 상태 변경시 이벤트
6	void	OnReceiveRealCondition	조건검색 실시간 편입,이탈종목 이벤트
7	void	OnReceiveTrCondition	조건검색 조회응답 이벤트
8	Void	OnReceiveConditionVer	로컬에 사용자조건식 저장 성공여부 응답 이벤트

6. 제어 함수 상세 명세서

- OpenAPI Active X 컨트롤은 프로퍼티(property) 및 메소드(method)를 통하여 제어 및 조작을 할 수 있다. 이러한 차트를 제어 하는 함수에 대한 상세 명세는 다음과 같습니다.
- 개별 항목 검색의 편의를 위하여 ID 순으로 정렬하였습니다.

6.1 메소드(Method)

1) CommConnect

원형	LONG CommConnect()
설명	로그인 윈도우를 실행한다.
입력값	없음
반환값	0 - 성공, 음수값은 실패
비고	로그인이 성공하거나 실패하는 경우 OnEventConnect 이벤트가 발생하고 이벤트의 인자 값으로 로그인 성공 여부를 알 수 있다.

2) CommTerminate

원형	void CommTerminate()
설명	더 이상 지원하지 않는 함수(기능없음)
입력값	없음
반환값	없음
비고	통신 연결 상태는 GetConnectState 메소드로 알 수 있다.

3) CommRqData

원형	LONG CommRqData(BSTR sRQName, BSTR sTrCode, long nPrevNext, BSTR sScreenNo)
설명	Tran을 서버로 송신한다.
입력값	BSTR sRQName BSTR sTrCode long nPrevNext BSTR sScreenNo
반환값	OP_ERR_SISE_OVERFLOW - 과도한 시세조희로 인한 통신불가 OP_ERR_RQ_STRUCT_FAIL - 입력 구조체 생성 실패 OP_ERR_RQ_STRING_FAIL - 요청전문 작성 실패 OP_ERR_NONE - 정상처리
비고	sRQName - 사용자구분 명 sTrCode - Tran명 입력 nPrevNext - 0:조회, 2:연속 sScreenNo - 4자리의 화면번호 Ex) openApi.CommRqData("RQ_1", "OPT00001", 0, "0101");

4) GetLoginInfo

원형	BSTR GetLoginInfo(BSTR sTag)
설명	로그인한 사용자 정보를 반환한다.
입력값	BSTR sTag : 사용자 정보 구분 TAG값 (비고)

반환값	TAG값에 따른 데이터 반환
비고	<p>BSTR sTag에 들어 갈 수 있는 값은 아래와 같음</p> <p>"ACCOUNT_CNT" - 전체 계좌 개수를 반환한다.</p> <p>"ACCNO" - 전체 계좌를 반환한다. 계좌별 구분은 ':'이다.</p> <p>"USER_ID" - 사용자 ID를 반환한다.</p> <p>"USER_NAME" - 사용자명을 반환한다.</p> <p>"KEY_BSECGB" - 키보드보안 해지여부. 0:정상, 1:해지</p> <p>"FIREW_SECGB" - 방화벽 설정 여부. 0:미설정, 1:설정, 2:해지</p> <p>Ex) openApi.GetLoginInfo("ACCOUNT_CNT");</p>

5) SendOrder

원형	<p>LONG SendOrder(BSTR sRQName, BSTR sScreenNo, BSTR sAccNo, LONG nOrderType, BSTR sCode, LONG nQty, LONG nPrice, BSTR sHogaGb, BSTR sOrgOrderNo)</p>
설명	주식 주문을 서버로 전송한다.
입력값	<p>sRQName - 사용자 구분 요청 명</p> <p>sScreenNo - 화면번호[4]</p> <p>sAccNo - 계좌번호[10]</p> <p>nOrderType - 주문유형 (1:신규매수, 2:신규매도, 3:매수취소, 4:매도취소, 5:매수정정, 6:매도정정)</p> <p>sCode, - 주식종목코드</p> <p>nQty - 주문수량</p> <p>nPrice - 주문단가</p> <p>sHogaGb - 거래구분</p> <p>sOrgOrderNo - 원주문번호</p>
반환값	에러코드 <4.에러코드표 참고>
비고	<p>sHogaGb - 00:지정가, 03:시장가, 05:조건부지정가, 06:최유리지정가, 07:최우선지정가, 10:지정가IOC, 13:시장가IOC, 16:최유리IOC, 20:지정가FOK, 23:시장가FOK, 26:최유리FOK, 61:장전시간외종가, 62:시간외단일가, 81:장후시간외종가</p> <p>※ 시장가, 최유리지정가, 최우선지정가, 시장가IOC, 최유리IOC, 시장가FOK, 최유리FOK, 장전시간외, 장후시간외 주문시 주문가격을 입력하지 않습니다.</p> <p>ex)</p> <p>지정가 매수 - openApi.SendOrder("RQ_1", "0101", "5015123410", 1, "000660", 10, 48500, "00", "");</p> <p>시장가 매수 - openApi.SendOrder("RQ_1", "0101", "5015123410", 1, "000660", 10, 0, "03", "");</p> <p>매수 정정 - openApi.SendOrder("RQ_1", "0101", "5015123410", 5, "000660", 10, 49500, "00", "1");</p> <p>매수 취소 - openApi.SendOrder("RQ_1", "0101", "5015123410", 3, "000660", 10, 0, "00", "2");</p>

6) SendOrderCredit

원형	LONG SendOrderCredit(LPCTSTR sRQName , LPCTSTR sScreenNo , LPCTSTR sAccNo , LONG nOrderType , LPCTSTR sCode , LONG nQty , LONG nPrice , LPCTSTR sHogaGb , LPCTSTR sCreditGb , LPCTSTR sLoanDate , LPCTSTR sOrgOrderNo)
설명	신용주식 주문을 서버로 전송한다.
입력값	sRQName - 사용자 구분 요청 명 sScreenNo - 화면번호[4] sAccNo - 계좌번호[10] nOrderType - 주문유형 (1:신규매수, 2:신규매도, 3:매수취소, 4:매도취소, 5:매수정정, 6:매도정정) sCode, - 주식종목코드 nQty - 주문수량 nPrice - 주문단가 sHogaGb - 거래구분 sCreditGb - 신용구분 sLoanDate - 대출일 sOrgOrderNo - 원주문번호
반환값	에러코드 <4.에러코드표 참고>
비고	sCreditGb - 신용구분 (신용매수:03, 신용매도 융자상환:33, 신용매도 융자합:99) 신용매수 주문 - 신용구분값 "03", 대출일은 "공백" 신용매도 융자상환 주문 - 신용구분값 "33", 대출일은 종목별 대출일 입력 - OPW00005/OPW00004 TR조회로 대출일 조회 신용매도 융자합 주문시 - 신용구분값 "99", 대출일은 "99991231" - 단 신용잔고 5개까지만 융자합 주문가능 나머지 입력값은 SendOrder()함수 설명참고

7) SetInputValue

원형	void SetInputValue(BSTR sID, BSTR sValue)
설명	Tran 입력 값을 서버통신 전에 입력한다.
입력값	sID - 아이템명 sValue - 입력 값
반환값	없음
비고	Ex) openApi.SetInputValue("종목코드", "000660"); openApi.SetInputValue("계좌번호", "5015123401");

8) SetOutputFID

원형	LONG SetOutputFID(BSTR sID)
설명	1.0.0.1 버전 이후 사용하지 않음.
입력값	
반환값	
비고	

9) CommGetData

원형	BSTR CommGetData(...)
설명	이 함수는 지원하지 않을 것이므로 용도에 맞는 전용 함수를 사용할 것(비고참고)
입력값	
반환값	
비고	조회 정보 요청 - openApi.GetCommData("OPT00001", RQName, 0, "현재가"); 실시간정보 요청 - openApi.GetCommRealData("000660", 10); 체결정보 요청 - openApi.GetChejanData(9203);

10) DisconnectRealData

원형	void DisconnectRealData(LPCTSTR sScnNo)
설명	화면 내 모든 리얼데이터 요청을 제거한다.
입력값	sScnNo - 화면번호[4]
반환값	없음
비고	화면을 종료할 때 반드시 위 함수를 호출해야 한다. Ex) openApi.DisconnectRealData("0101");

11) GetRepeatCnt

원형	LONG GetRepeatCnt(LPCTSTR sTrCode, LPCTSTR sRecordName)
설명	레코드 반복횟수를 반환한다.
입력값	sTrCode - Tran 명 sRecordName - 레코드 명
반환값	레코드의 반복횟수
비고	Ex) openApi.GetRepeatCnt("OPT00001", "주식기본정보");

12) CommKwRqData

원형	LONG CommKwRqData(LPCTSTR sArrCode, BOOL bNext, int nCodeCount, int nTypeFlag, LPCTSTR sRQName, LPCTSTR sScreenNo)
설명	복수종목조회 Tran을 서버로 송신한다.
입력값	sArrCode - 종목리스트 bNext - 연속조회요청 nCodeCount - 종목개수 nTypeFlag - 조회구분 sRQName - 사용자구분 명 sScreenNo - 화면번호[4]
반환값	OP_ERR_RQ_STRING - 요청 전문 작성 실패 OP_ERR_NONE - 정상처리
비고	sArrCode - 종목간 구분은 ';'이다.

	nTypeFlag - 0:주식관심종목정보, 3:선물옵션관심종목정보 ex) openApi.CommKwRqData("000660:005930", 0, 2, 0, "RQ_1", "0101");
--	---

13) GetAPIModulePath

원형	BSTR GetAPIModulePath()
설명	OpenAPI모듈의 경로를 반환한다.
입력값	없음
반환값	경로
비고	

14) GetCodeListByMarket

원형	BSTR GetCodeListByMarket(LPCTSTR sMarket)
설명	시장구분에 따른 종목코드를 반환한다.
입력값	sMarket - 시장구분
반환값	종목코드 리스트, 종목간 구분은 ';'이다.
비고	sMarket - 0:장내, 3:ELW, 4:뮤추얼펀드, 5:신주인수권, 6:리츠, 8:ETF, 9:하이일드펀드, 10:코스닥, 30:제3시장

15) GetConnectState

원형	LONG GetConnectState()
설명	현재접속상태를 반환한다.
입력값	없음
반환값	접속상태
비고	0:미연결, 1:연결완료

16) GetMasterCodeName

원형	BSTR GetMarsterCodeName(LPCTSTR strCode)
설명	종목코드의 한글명을 반환한다.
입력값	strCode - 종목코드
반환값	종목한글명
비고	장내외, 지수선옵, 주식선옵 검색 가능.

17) GetMasterListedStockCnt

원형	LONG GetMasterListedStockCnt(LPCTSTR strCode)
설명	종목코드의 상장주식수를 반환한다.
입력값	strCode - 종목코드
반환값	상장주식수
비고	

18) GetMasterConstruction

원형	BSTR GetMasterConstruction(LPCTSTR strCode)
----	---

설명	종목코드의 감리구분을 반환한다.
입력값	strCode - 종목코드
반환값	감리구분
비고	감리구분 - 정상, 투자주의, 투자경고, 투자위험, 투자주의환기종목

19) GetMasterListedStockDate

원형	BSTR GetMasterListedStockDate(LPCTSTR strCode)
설명	종목코드의 상장일을 반환한다.
입력값	strCode - 종목코드
반환값	상장일
비고	상장일 포맷 - xxxxxxxx[8]

20) GetMasterLastPrice

원형	BSTR GetMasterLastPrice(LPCTSTR strCode)
설명	종목코드의 전일가를 반환한다.
입력값	strCode - 종목코드
반환값	전일가
비고	

21) GetMasterStockState

원형	BSTR GetMasterStockState(LPCTSTR strCode)
설명	종목코드의 종목상태를 반환한다.
입력값	strCode - 종목코드
반환값	종목상태
비고	종목상태 - 정상, 증거금 100%, 거래정지, 관리종목, 감리종목, 투자유의종목, 담보대출, 액면분할, 신용가능

22) GetDataCount

원형	LONG GetDataCount(LPCTSTR strRecordName)
설명	레코드의 반복개수를 반환한다.
입력값	strRecordName - 레코드명
반환값	레코드 반복개수
비고	Ex) openApi.GetDataCount("주식기본정보");

23) GetOutputValue

원형	BSTR GetOutputValue(LPCTSTR strRecordName, long nRepeatIdx, long nIndex)
설명	레코드의 반복순서와 아이템의 출력순서에 따라 수신데이터를 반환한다.
입력값	nRepeatIdx - 반복순서 nIndex - 아이템 순서
반환값	수신 데이터
비고	Ex) 현재가출력 - openApi.GetOutputValue("주식기본정보", 0, 36);

24) GetCommData

원형	BSTR GetCommData(LPCTSTR strTrCode, LPCTSTR strRecordName, long nIndex, LPCTSTR strItemName)
설명	수신 데이터를 반환한다.
입력값	strTrCode - Tran 코드 strRecordName - 레코드명 nIndex - 복수데이터 인덱스 strItemName - 아이템명
반환값	수신 데이터
비고	Ex) 현재가출력 - openApi.GetCommData("OPT00001", "주식기본정보", 0, "현재가");

25) GetCommRealData

원형	BSTR GetCommRealData(LPCTSTR strCode, long nFid)
설명	실시간 시세 데이터를 반환한다.
입력값	strCode - 종목코드 nFid - 실시간 아이템
반환값	수신 데이터
비고	Ex) 현재가출력 - openApi.GetCommRealData("039490", 10); 참고) strCode는 OnReceiveRealData 첫번째 매개변수를 사용

26) GetChejanData

원형	BSTR GetChjanData(long nFid)
설명	체결잔고 데이터를 반환한다.
입력값	nFid - 체결잔고 아이템
반환값	수신 데이터
비고	Ex) 현재가출력 - openApi.GetChejanData(10);

27) GetThemeGroupList

원형	BSTR GetThemeGroupList(long nType)
설명	테마코드와 테마명을 반환한다.
입력값	nType - 정렬순서 (0:코드순, 1:테마순)
반환값	코드와 코드명 리스트
비고	반환값의 코드와 코드명 구분은 ' ' 코드의 구분은 ';' Ex) 100 태양광_폴리실리콘;152 합성섬유

28) GetThemeGroupCode

원형	BSTR GetThemeGroupCode(LPCTSTR strThemeCode)
설명	테마코드에 소속된 종목코드를 반환한다.
입력값	strThemeCode - 테마코드
반환값	종목코드 리스트
비고	반환값의 종목코드간 구분은 ';' Ex) A000660:A005930

29) GetFutureList

원형	BSTR GetFuttrueList()
설명	지수선물 리스트를 반환한다.
입력값	
반환값	종목코드 리스트
비고	반환값의 종목코드간 구분은 ';' Ex) 101J9000;101JC000

30) GetFutureCodeByIndex

원형	BSTR GetFutreCodeByIndex(int nIndex)
설명	지수선물 코드를 반환한다.
입력값	nIndex - 0~3 지수선물코드, 4~7 지수스프레드
반환값	종목코드
비고	Ex) 최근월선물 - openApi.GetFutureCodeByInex(0); 최근월스프레드 - openApi.GetFutureCodeByInex(4);

31) GetActPriceList

원형	BSTR GetActPriceList()
설명	지수옵션 행사가 리스트를 반환한다.
입력값	
반환값	행사가
비고	반환값의 행사가간 구분은 ';' Ex) 265.00;262.50;260.00

32) GetMonthList

원형	BSTR GetMonthList()
설명	지수옵션 월물 리스트를 반환한다.
입력값	
반환값	월물
비고	반환값의 월물간 구분은 ';' Ex) 201412;201409;201408;201407;201407;201408;201409;201412

33) GetOptionCode

원형	BSTR GetOptionCode(LPCTSTR strActPrice, int nCp, LPCTSTR strMonth)
설명	행사가와 월물 콤보로 종목코드를 구한다.
입력값	strActPrice - 행사가(소수점포함) nCp - 콤보구분 2:콜, 3:풋 strMonth - 월물(6자리)
반환값	종목코드
비고	Ex) openApi.GetOptionCode("260.00", 2, "201407");

34) GetOptionCodeByMonth

원형	BSTR GetOptionCodeByMonth(LPCTSTR strCode, int nCp, LPCTSTR strMonth)
설명	입력된 종목코드와 동일한 행사가의 코드중 입력한 월물의 코드를 구한다.
입력값	strCode - 종목코드 nCp - 콜풋구분 2:콜, 3:풋 strMonth - 월물(6자리)
반환값	종목코드
비고	Ex) openApi.GetOptionCodeByMonth("201J7260", 2, "201412"); 결과값 = 201JC260

35) GetOptionCodeByActPrice

원형	BSTR GetOptionCodeByActPrice(LPCTSTR strCode, int nCp, int Tick)
설명	입력된 종목코드와 동일한 월물의 코드중 입력한 틱만큼 벌어진 코드를 구한다.
입력값	strCode - 종목코드 nCp - 콜풋구분 2:콜, 3:풋 nTick - 행사가 틱
반환값	종목코드
비고	Ex) openApi.GetOptionCodeByActPrice("201J7260", 2, -1); 결과값 = 201J7262

36) GetFutureList

원형	BSTR GetFutureList(LPCTSTR strBaseAssetCode)
설명	주식선물 코드 리스트를 반환한다.
입력값	strBaseAssetCode - 기초자산코드
반환값	종목코드 리스트
비고	출력값의 코드간 구분은 ';'이다.

37) GetFutureCodeByIndex

원형	BSTR GetFutureCodeByIndex(LPCTSTR strBaseAssetCode, int nIndex)
설명	주식선물 코드를 반환한다.
입력값	strBaseAssetCode - 기초자산코드 nIndex - 0~3 지수선물코드, 4~7 지수스프레드, 8~11 스타 선물, 12~ 스타 스프레드
반환값	종목코드
비고	Ex) openApi.GetFutureCodeByIndex("11", 0);

38) GetActPriceList

원형	BSTR GetActPriceList(LPCTSTR strBaseAssetGb)
설명	주식옵션 행사가 리스트를 반환한다.
입력값	strBaseAssetGb - 기초자산코드구분
반환값	행사가 리스트, 행사가간 구분은 ';'
비고	Ex) openApi.GetActPriceList("11");

39) GetSMonthList

원형	BSTR GetSMonthList(LPCTSTR strBaseAssetGb)
설명	주식옵션 월물 리스트를 반환한다.
입력값	strBaseAssetGb - 기초자산코드구분
반환값	월물 리스트, 월물간 구분은 ‘;’
비고	Ex) openApi.GetSActPriceList("11");

40) GetOptionCode

원형	BSTR GetOptionCode(LPCTSTR strBaseAssetGb, LPCTSTR strActPrice, int nCp, LPCTSTR strMonth)
설명	주식옵션 코드를 반환한다.
입력값	strBaseAssetGb - 기초자산코드구분 strActPrice - 행사가 nCp - 콜풋구분 strMonth - 월물
반환값	주식옵션 코드
비고	Ex) openApi.GetOptionCode("11", "1300000", 2, "1412");

41) GetOptionCodeByMonth

원형	BSTR GetOptionCodeByMonth(LPCTSTR strBaseAssetGb, LPCTSTR strCode, int nCp, LPCTSTR strMonth)
설명	입력한 주식옵션 코드에서 월물만 변경하여 반환한다.
입력값	strBaseAssetGb - 기초자산코드구분 strCode - 종목코드 nCp - 콜풋구분 strMonth - 월물
반환값	주식옵션 코드
비고	Ex) openApi.GetOptionCodeByMonth("11", "211J8045", 2, "1412");

42) GetOptionCodeByActPrice

원형	BSTR GetOptionCodeByActPrice(LPCTSTR strBaseAssetGb, LPCTSTR strCode, int nCp, int nTick)
설명	입력한 주식옵션 코드에서 행사가만 변경하여 반환한다.
입력값	strBaseAssetGb - 기초자산코드구분 strCode - 종목코드 nCp - 콜풋구분 nTick - 월물
반환값	주식옵션 코드
비고	Ex) openApi.GetOptionCodeByActPrice("11", "211J8045", 2, 4);

43) GetSFOBasisAssetList

원형	BSTR GetSFOBasisAssetList()
설명	주식선옵 기초자산코드/종목명을 반환한다.

입력값	
반환값	기초자산코드/종목명, 코드와 종목명 구분은 ' ' 코드간 구분은';' Ex) 211J8045 삼성전자 C 201408;212J8009 SK텔레콤 C 201408
비고	Ex) openApi.GetSFOBasisAssetList();

44) GetOptionATM

원형	BSTR GetOptionATM()
설명	지수옵션 ATM을 반환한다.
입력값	
반환값	ATM
비고	Ex) openApi.GetOptionATM();

45) GetSOptionATM

원형	BSTR GetSOptionATM(LPCTSTR strBaseAssetGb)
설명	주식옵션 ATM을 반환한다.
입력값	
반환값	ATM
비고	Ex) openApi.GetSOptionATM("11");

46) GetBranchCodeName

원형	BSTR GetBranchCodeName()
설명	회원사 코드와 이름을 반환합니다.
입력값	
반환값	회원사코드 회원사명;회원사코드 회원사명;...
비고	Ex) openApi.GetBranchCodeName();

47) CommInvestRqData

원형	BSTR CommInvestRqData(LPCTSTR sMarketGb, LPCTSTR sRQName, LPCTSTR sScreenNo)
설명	투자자 정보조회를 요청한다.
입력값	sMarketGb - 시장구분 001:코스피, 002:코스닥, 003:선물, 004:콜옵션, 005:풋옵션, 006:스타선물 007:주식선물, 008:3년국채, 009:5년국채, 010:10년국채, 011:달러선물, 012:엔선물 013:유로선물, 014:미니금선물, 015:금선물, 016:돈육선물, 017:달러콜옵션, 018:달러풋옵션 sRQName - 사용자구분값 sScreenNo - 화면번호
반환값	통신결과
비고	Ex) openApi.CommInvestRqData("T00108:T00109", 0, 2, "RQ_1", "0101");

48) SetInfoData

원형	LONG SetInfoData(LPCTSTR sInfoData)
설명	다수의 아이디로 자동로그인이 필요할 때 사용한다.

입력값	sInfoData - 아이디
반환값	통신결과
비고	Ex) openApi.SetInfoData("UserID");

49) SetRealReg

원형	SetRealReg(LPCTSTR strScreenNo, LPCTSTR strCodeList, LPCTSTR strFidList, LPCTSTR strRealType)
설명	실시간 등록을 한다.
입력값	strScreenNo - 실시간 등록할 화면 번호 strCodeList - 실시간 등록할 종목코드(복수종목가능 - "종목1;종목2;종목3;...") strFidList - 실시간 등록할 FID("FID1:FID2:FID3;...") strRealType - "0", "1" 타입
반환값	통신결과
비고	strRealType이 "0" 으로 하면 같은화면에서 다른종목 코드로 실시간 등록을 하게 되면 마지막에 사용한 종목코드만 실시간 등록이 되고 기존에 있던 종목은 실시간이 자동 해지됨. "1"로 하면 같은화면에서 다른 종목들을 추가하게 되면 기존에 등록한 종목도 함께 실시간 시세를 받을 수 있음. 꼭 같은 화면이어야 하고 최초 실시간 등록은 "0"으로 하고 이후부터 "1"로 등록해야함.

50) SetRealRemove

원형	Void SetRealRemove(LPCTSTR strScrNo, LPCTSTR strDelCode)
설명	종목별 실시간 해제.
입력값	strScrNo - 실시간 해제할 화면 번호 strDelCode - 실시간 해제할 종목.
반환값	통신결과
비고	SetRealReg() 함수로 실시간 등록한 종목만 실시간 해제 할 수 있다.

51) GetConditionLoad

원형	long GetConditionLoad()
설명	서버에 저장된 사용자 조건식을 조회해서 임시로 파일에 저장.
입력값	
반환값	사용자 조건식을 파일로 임시 저장.
비고	System 폴더에 아이디_NewSaveIndex.dat파일로 저장된다. Ocx가 종료되면 삭제시킨다. 조건검색 사용시 이함수를 최소 한번은 호출해야 조건검색을 할 수 있다. 영웅문에서 사용자 조건을 수정 및 추가하였을 경우에도 최신의 사용자 조건을 받고 싶으면 다시 조회해야한다.

52) GetConditionNameList

원형	BSTR GetConditionNameList()
설명	조건검색 조건명 리스트를 받아온다.
입력값	
반환값	조건명 리스트(인덱스^조건명)
비고	조건명 리스트를 구분(";")하여 받아온다. Ex) 인덱스1^조건명1;인덱스2^조건명2;인덱스3^조건명3;...

53) SendCondition

원형	BOOL SendCondition(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex, int nSearch)
설명	조건검색 종목조회TR송신한다.
입력값	LPCTSTR strScrNo : 화면번호 LPCTSTR strConditionName : 조건명 int nIndex : 조건명인덱스 int nSearch : 조회구분(0:일반조회, 1:실시간조회, 2:연속조회)
반환값	성공 1, 실패 0
비고	단순 조건식에 맞는 종목을 조회하기 위해서는 조회구분을 0으로 하고, 실시간 조건검색을 하기 위해서는 조회구분을 1로 한다. OnReceiveTrCondition으로 결과값이 온다. 연속조회가 필요한 경우에는 응답받는 곳에서 연속조회 여부에 따라 연속조회를 송신하면된다.

54) SendConditionStop

원형	Void SendConditionStop(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex)
설명	조건검색 실시간 중지TR을 송신한다.
입력값	LPCTSTR strScrNo : 화면번호 LPCTSTR strConditionName : 조건명 int nIndex : 조건명인덱스
반환값	
비고	해당 조건명의 실시간 조건검색을 중지하거나, 다른 조건명으로 바꿀 때 이전 조건명으로 실시간 조건검색을 반드시 중지해야한다. 화면 종료시에도 실시간 조건검색을 한 조건명으로 전부 중지해줘야 한다.

55) GetCommDataEx

원형	VARIANT GetCommDataEx(LPCTSTR strTrCode, LPCTSTR strRecordName)
설명	차트 조회 데이터를 배열로 받아온다.
입력값	LPCTSTR strTrCode : 조회한TR코드 LPCTSTR strRecordName: 조회한 TR명
반환값	
비고	조회 데이터가 많은 차트 경우 GetCommData()로 항목당 하나씩 받아오는 것 보다 한번에 데이터 전부를 받아서 사용자가 처리할 수 있도록 배열로 받는다.

6.2 이 벤 트

1) OnReceiveTrData

원형	void OnReceiveTrData(LPCTSTR sScrNo, LPCTSTR sRQName, LPCTSTR sTrCode, LPCTSTR sRecordName, LPCTSTR sPreNext, LONG nDataLength, LPCTSTR sErrorCode, LPCTSTR sMessage, LPCTSTR sSplmMsg)
설명	서버통신 후 데이터를 받은 시점을 알려준다.
입력값	sScrNo - 화면번호 sRQName - 사용자구분 명 sTrCode - Tran 명 sRecordName - Record 명

	sPreNext - 연속조회 유무 nDataLength - 1.0.0.1 버전 이후 사용하지 않음. sErrorCode - 1.0.0.1 버전 이후 사용하지 않음. sMessage - 1.0.0.1 버전 이후 사용하지 않음. sSplmMsg - 1.0.0.1 버전 이후 사용하지 않음.
반환값	없음
비고	sRQName - CommRqData의 sRQName과 매핑되는 이름이다. sTrCode - CommRqData의 sTrCode와 매핑되는 이름이다.

2) OnReceiveRealData

원형	void OnReceiveRealData(LPCTSTR sJongmokCode, LPCTSTR sRealType, LPCTSTR sRealData)
설명	실시간데이터를 받은 시점을 알려준다.
입력값	sJongmokCode - 종목코드 sRealType - 리얼타입 sRealData - 실시간 데이터전문
반환값	없음
비고	

3) OnReceiveMsg

원형	void OnReceiveMsg(LPCTSTR sScrNo, LPCTSTR sRQName, LPCTSTR sTrCode, LPCTSTR sMsg)
설명	서버통신 후 메시지를 받은 시점을 알려준다.
입력값	sScrNo - 화면번호 sRQName - 사용자구분 명 sTrCode - Tran 명 sMsg - 서버메시지
반환값	없음
비고	sScrNo - CommRqData의 sScrNo와 매핑된다. sRQName - CommRqData의 sRQName 와 매핑된다. sTrCode - CommRqData의 sTrCode 와 매핑된다.

4) OnReceiveChejanData

원형	void OnReceiveChejanData(LPCTSTR sGubun, LONG nItemCnt, LPCTSTR sFidList);
설명	체결데이터를 받은 시점을 알려준다.
입력값	sGubun - 체결구분 nItemCnt - 아이템갯수 sFidList - 데이터리스트
반환값	없음
비고	sGubun - 0:주문체결통보, 1:잔고통보, 3:특이신호 sFidList - 데이터 구분은 ':' 이다.

5) OnEventConnect

원형	void OnEventConnect(LONG nErrCode);
설명	서버 접속 관련 이벤트
입력값	LONG nErrCode : 에러 코드
반환값	없음
비고	nErrCode가 0이면 로그인 성공, 음수면 실패 음수인 경우는 에러 코드 참조

6) OnReceiveCondition

원형	void OnReceiveRealCondition(LPCTSTR strCode, LPCTSTR strType, LPCTSTR strConditionName, LPCTSTR strConditionIndex)
설명	조건검색 실시간 편입,이탈 종목을 받을 시점을 알려준다.
입력값	LPCTSTR strCode : 종목코드 LPCTSTR strType : 편입("I"), 이탈("D") LPCTSTR strConditionName : 조건명 LPCTSTR strConditionIndex : 조건명 인덱스
반환값	없음
비고	strConditionName에 해당하는 종목이 실시간으로 들어옴. strType으로 편입된 종목인지 이탈된 종목인지 구분한다.

7) OnReceiveTrCondition

원형	void OnReceiveTrCondition(LPCTSTR sScrNo, LPCTSTR strCodeList, LPCTSTR strConditionName, int nIndex, int nNext)
설명	조건검색 조회응답으로 종목리스트를 구분자(";")로 붙여서 받는 시점.
입력값	LPCTSTR sScrNo : 종목코드 LPCTSTR strCodeList : 종목리스트(";")로 구분) LPCTSTR strConditionName : 조건명 int nIndex : 조건명 인덱스 int nNext : 연속조회(2:연속조회, 0:연속조회없음)
반환값	없음
비고	

8) OnReceiveConditionVer

원형	void OnReceiveConditionVer(long lRet, LPCTSTR sMsg)
설명	로컬에 사용자 조건식 저장 성공 여부를 확인하는 시점
입력값	long lRet : 사용자 조건식 저장 성공여부 (1: 성공, 나머지 실패)
반환값	없음
비고	

7. 에러코드표

리턴코드		설 명
0	OP_ERR_NONE	정상처리
-10	OP_ERR_FAIL	실패
-100	OP_ERR_LOGIN	사용자정보교환실패
-101	OP_ERR_CONNECT	서버접속실패
-102	OP_ERR_VERSION	버전처리실패
-103	OP_ERR_FIREWALL	개인방화벽실패
-104	OP_ERR_MEMORY	메모리보호실패
-105	OP_ERR_INPUT	함수입력값오류
-106	OP_ERR_SOCKET_CLOSED	통신연결종료
-200	OP_ERR_SISE_OVERFLOW	시세조회과부하
-201	OP_ERR_RQ_STRUCT_FAIL	전문작성초기화실패
-202	OP_ERR_RQ_STRING_FAIL	전문작성입력값오류
-203	OP_ERR_NO_DATA	데이터없음.
-204	OP_ERR_OVER_MAX_DATA	조회가능한종목수초과
-205	OP_ERR_DATA_RCV_FAIL	데이터수신실패
-206	OP_ERR_OVER_MAX_FID	조회가능한FID수초과
-207	OP_ERR_REAL_CANCEL	실시간해제오류
-300	OP_ERR_ORD_WRONG_INPUT	입력값오류
-301	OP_ERR_ORD_WRONG_ACCTNO	계좌비밀번호없음
-302	OP_ERR_OTHER_ACC_USE	타인계좌사용오류
-303	OP_ERR_MIS_2BILL_EXC	주문가격이20억원을초과
-304	OP_ERR_MIS_5BILL_EXC	주문가격이50억원을초과
-305	OP_ERR_MIS_1PER_EXC	주문수량이총발행주수의1%초과오류
-306	OP_ERR_MIS_3PER_EXC	주문수량은총발행주수의3%초과오류
-307	OP_ERR_SEND_FAIL	주문전송실패
-308	OP_ERR_ORD_OVERFLOW	주문전송과부하
-309	OP_ERR_MIS_300CNT_EXC	주문수량300계약초과
-310	OP_ERR_MIS_500CNT_EXC	주문수량500계약초과
-340	OP_ERR_ORD_WRONG_ACCTINFO	계좌정보없음
-500	OP_ERR_ORD_SYMCODE_EMPTY	종목코드없음

8. 실시간 FID

8.1 주식시세

FID	설명
10	현재가, 체결가, 실시간종가
11	전일 대비
12	등락율
27	(최우선)매도호가
28	(최우선)매수호가
13	누적거래량, 누적체결량
14	누적거래대금
16	시가
17	고가
18	저가
25	전일대비기호
26	전일거래량 대비(계약,주)
29	거래대금 증감
30	전일거래량 대비(비율)
31	거래회전율
32	거래비용
311	시가총액(억)

8.2 주식체결

FID	설명
20	체결시간 (HHMMSS)
10	현재가, 체결가, 실시간종가
11	전일 대비
12	등락율
27	(최우선)매도호가
28	(최우선)매수호가
15	거래량, 체결량
13	누적거래량, 누적체결량
14	누적거래대금
16	시가
17	고가
18	저가
25	전일대비 기호
26	전일거래량 대비(계약, 주)
29	거래대금 증감
30	전일거래량 대비(비율)
31	거래회전율
32	거래비용
228	체결강도
311	시가총액(억)
290	장구분
691	K,O 접근도 (ELW조기종료발생 기준가격, 지수)

8.3 주식우선평가

FID	설명
27	(최우선)매도호가
28	(최우선)매수호가

8.4 주식호가잔량

FID	설명
21	호가시간
41	매도호가1
61	매도호가 수량1
81	매도호가 직전대비1
51	매수호가1
71	매수호가 수량1
91	매수호가 직전대비1
42	매도호가1
62	매도호가 수량2
82	매도호가 직전대비2
52	매수호가2
72	매수호가 수량2
92	매수호가 직전대비2
43	매도호가3
63	매도호가 수량3
83	매도호가 직전대비3
53	매수호가3
73	매수호가 수량3
93	매수호가 직전대비3
44	매도호가4
64	매도호가 수량4
84	매도호가 직전대비4
54	매수호가4
74	매수호가 수량4
94	매수호가 직전대비4
45	매도호가5
65	매도호가 수량5
85	매도호가 직전대비5
55	매수호가5
75	매수호가 수량5
95	매수호가 직전대비5
46	매도호가6
66	매도호가 수량6
86	매도호가 직전대비6
56	매수호가6
76	매수호가 수량6
96	매수호가 직전대비6
47	매도호가7

67	매도호가 수량7
87	매도호가 직전대비7
57	매수호가7
77	매수호가 수량7
97	매수호가 직전대비7
48	매도호가8
68	매도호가 수량8
88	매도호가 직전대비8
58	매수호가8
78	매수호가 수량8
98	매수호가 직전대비8
49	매도호가9
69	매도호가 수량9
89	매도호가 직전대비9
59	매수호가9
79	매수호가 수량9
99	매수호가 직전대비9
50	매도호가10
70	매도호가 수량10
90	매도호가 직전대비10
60	매수호가10
80	매수호가 수량10
100	매수호가 직전대비10
121	매도호가 총잔량
122	매도호가 총잔량 직전대비
125	매수호가 총잔량
126	매수호가 총잔량 직전대비
23	예상체결가
24	예상체결 수량
128	순매수잔량(총매수잔량-총매도잔량)
129	매수비율
138	순매도잔량(총매도잔량-총매수잔량)
139	매도비율
200	예상체결가 전일종가 대비
201	예상체결가 전일종가 대비 등락율
238	예상체결가 전일종가 대비기호
291	예상체결가
292	예상체결량
293	예상체결가 전일대비기호
294	예상체결가 전일대비
295	예상체결가 전일대비등락율
621	LP매도호가 수량1
631	LP매수호가 수량1
622	LP매도호가 수량2
632	LP매수호가 수량2
623	LP매도호가 수량3
633	LP매수호가 수량3

624	LP매도호가 수량4
634	LP매수호가 수량4
625	LP매도호가 수량5
635	LP매수호가 수량5
626	LP매도호가 수량6
636	LP매수호가 수량6
627	LP매도호가 수량7
637	LP매수호가 수량7
628	LP매도호가 수량8
638	LP매수호가 수량8
629	LP매도호가 수량9
639	LP매수호가 수량9
630	LP매도호가 수량10
640	LP매수호가 수량10
13	누적거래량, 누적체결량
299	전일거래량대비예상체결률
215	장운영구분
216	투자자별 ticker

8.5 선물시세

FID	설명
20	체결시간(HHMMSS)
10	현재가, 체결가, 실시간종가
11	전일 대비
12	등락율
27	(최우선)매도호가
28	(최우선)매수호가
15	거래량, 체결량
13	누적거래량, 누적체결량
14	누적거래대금
16	시가
17	고가
18	저가
195	미결제약정
182	이론가
184	이론베이스스
183	시장베이스스
186	괴리율
181	미결제 약정 전일대비
185	괴리도
25	전일대비 기호
197	KOSPI200
26	전일거래량 대비(계약, 주)
246	시초 미결제 약정수량
247	최고 미결제 약정수량
248	최저 미결제 약정수량

30	전일거래량 대비(비율)
196	미결제 증감

8.6 선물호가잔량

FID	설명
21	호가시간(HHMMSS)
27	(최우선)매도호가
28	(최우선)매수호가
41	매도호가1
61	매도호가 수량1
81	매도호가 직전대비1
101	매도호가 건수1
51	매수호가1
71	매수호가 수량1
91	매수호가 직전대비1
111	매수호가 건수1
42	매도호가2
62	매도호가 수량1
82	매도호가 직전대비2
102	매도호가 건수2
52	매수호가2
72	매수호가 수량2
92	매수호가 직전대비2
112	매수호가 건수2
43	매도호가3
63	매도호가 수량3
83	매도호가 직전대비3
103	매도호가 건수3
53	매수호가3
73	매수호가 수량3
93	매수호가 직전대비3
113	매수호가 건수3
44	매도호가4
64	매도호가 수량4
84	매도호가 직전대비4
104	매도호가 건수4
54	매수호가4
74	매수호가 수량4
94	매수호가 직전대비4
114	매수호가 건수4
45	매도호가5
65	매도호가 수량5
85	매도호가 직전대비5
105	매도호가 건수5
55	매수호가5
75	매수호가 수량5

95	매수호가 직전대비5
115	매수호가 건수5
121	매도호가 총잔량
122	매도호가 총잔량 직전대비
123	매도호가 총 건수
125	매수호가 총잔량
126	매수호가 총잔량 직전대비
127	매수호가 총 건수
137	호가 순잔량
128	순매수잔량(총매수잔량-총매도잔량)
13	누적거래량, 누적체결량
23	예상체결가
238	예상체결가 전일종가 대비기호
200	예상체결가 전일종가 대비
201	예상체결가 전일종가대비 등락율
291	예상체결가
293	예상체결가 전일대비기호
294	예상체결가 전일대비
295	예상체결가 전일대비등락율

8.7 선물이론가

FID	설명
195	미결제 약정
182	이론가
184	이론베이스스
183	시장베이스스
186	괴리율
181	미결제 약정 전일대비
185	괴리도
246	시초 미결제 약정수량
247	최고 미결제 약정수량
248	최저 미결제 약정수량

8.8 옵션시세

FID	설명
20	체결시간(HHMMSS)
10	현재가, 체결가, 실시간종가
11	전일 대비
12	등락율
27	(최우선)매도호가
28	(최우선)매수호가
15	거래량, 체결량
13	누적거래량, 누적체결량
14	누적거래대금
16	시가
17	고가

18	저가
195	미결제 약정
182	이론가
186	괴리율
190	델타
191	감마
193	세타
192	베가
194	로
181	미결제 약정 전일대비
25	전일대비 기호
26	전일거래량 대비(계약, 주)
137	호가 순잔량
187	내재가치
197	KOSPI200
246	시초 미결제 약정수량
247	최고 미결제 약정수량
248	최저 미결제 약정수량
219	선물 최근 월물지수
196	미결제 증감
188	시간가치
189	내재변동성(I.V.)
30	전일거래량 대비(비율)
391	기준가대비 시고등락율
392	기준가대비 고가등락율
393	기준가대비 저가등락율

8.9 옵션호가잔량

FID	설명
21	호가시간(HHMMSS)
27	(최우선)매도호가
28	(최우선)매수호가
41	매도호가1
61	매도호가 수량1
81	매도호가 직전대비1
101	매도호가 건수1
51	매수호가1
71	매수호가 수량1
91	매수호가 직전대비1
111	매수호가 건수1
42	매도호가2
62	매도호가 수량1
82	매도호가 직전대비2
102	매도호가 건수2
52	매수호가2
72	매수호가 수량2

92	매수호가 직전대비2
112	매수호가 건수2
43	매도호가3
63	매도호가 수량3
83	매도호가 직전대비3
103	매도호가 건수3
53	매수호가3
73	매수호가 수량3
93	매수호가 직전대비3
113	매수호가 건수3
44	매도호가4
64	매도호가 수량4
84	매도호가 직전대비4
104	매도호가 건수4
54	매수호가4
74	매수호가 수량4
94	매수호가 직전대비4
114	매수호가 건수4
45	매도호가5
65	매도호가 수량5
85	매도호가 직전대비5
105	매도호가 건수5
55	매수호가5
75	매수호가 수량5
95	매수호가 직전대비5
115	매수호가 건수5
121	매도호가 총잔량
122	매도호가 총잔량 직전대비
123	매도호가 총 건수
125	매수호가 총잔량
126	매수호가 총잔량 직전대비
127	매수호가 총 건수
137	호가 순잔량
128	순매수잔량(총매수잔량-총매도잔량)
13	누적거래량, 누적체결량
23	예상체결가
238	예상체결가 전일종가 대비기호
200	예상체결가 전일종가 대비
201	예상체결가 전일종가대비 등락율
291	예상체결가
293	예상체결가 전일대비기호
294	예상체결가 전일대비
295	예상체결가 전일대비등락율

8.10 옵션이론가

FID	설명
195	미결제 약정
182	이론가
186	괴리율
190	델타
191	감마
193	세타
192	베가
194	로
181	미결제 약정 전일대비
246	시초 미결제 약정수량
247	초고 미결제 약정수량
248	최저 미결제 약정수량
187	내재가치
188	시간가치
189	내재변동성(I.V.)

8.11 주식옵션시세

FID	설명
20	체결시간(HHMMSS)
10	현재가, 체결가, 실시간종가
11	전일 대비
12	등락율
27	(최우선)매도호가
28	(최우선)매수호가
15	거래량, 체결량
13	누적거래량, 누적체결량
14	누적거래대금
16	시가
17	고가
18	저가
195	미결제약정
182	이론가
186	괴리율
190	델타
191	감마
193	세타
192	베가
194	로
181	미결제 약정 전일대비
25	전일대비 기호
26	전일거래량 대비(계약, 주)

8.12 주식옵션호가잔량

FID	설명
21	호가시간(YYYYMMDD)
27	(최우선)매도호가
28	(최우선)매수호가
41	매도호가1
42	매도호가2
43	매도호가3
44	매도호가4
45	매도호가5
61	매도호가 수량1
62	매도호가 수량2
63	매도호가 수량3
64	매도호가 수량4
65	매도호가 수량5
101	매도호가 건수1
102	매도호가 건수2
103	매도호가 건수3
104	매도호가 건수4
105	매도호가 건수5
51	매수호가1
52	매수호가2
53	매수호가3
54	매수호가4
55	매수호가5
71	매수호가 수량1
72	매수호가 수량2
73	매수호가 수량3
74	매수호가 수량4
75	매수호가 수량5
111	매수호가 건수1
112	매수호가 건수2
113	매수호가 건수3
114	매수호가 건수4
115	매수호가 건수5
121	매도호가 총잔량
123	매도호가 총 건수
125	매수호가 총잔량
127	매수호가 총 건수
23	예상체결가
238	예상체결가 전일종가 대비기호
200	예상체결가 전일종가 대비
201	예상체결가 전일종가 대비 등락율
291	예상체결가
293	예상체결가 전일대비 기호
294	예상체결가 전일대비

295	예상체결가 전일대비 등락율
-----	----------------

8.13 주식옵션이론가

FID	설명
195	미결제 약정
182	이론가
186	과리율
190	델타
191	감마
193	세타
192	베가
194	로
181	미결제 약정 전일대비

8.14 주식옵션우선호가

FID	설명
10	현재가, 체결가, 실시간종가
27	(최우선)매도호가
28	(최우선)매수호가

8.15 업종지수

FID	설명
20	체결시간(HHMMSS)
10	현재가, 체결가, 실시간종가
11	전일 대비
12	등락율
15	거래량, 체결량
13	누적거래량, 누적체결량
14	누적거래대금
16	시가
17	고가
18	저가
25	전일대비 기호
26	전일거래량 대비(계약, 주)

8.16 업종등락

FID	설명
20	체결시간(HHMMSS)
252	상승종목수
251	상한종목수
253	보합종목수
255	하락종목수
254	하한종목수
13	누적거래량, 누적체결량
14	누적거래대금
10	현재가, 체결가, 실시간종가

11	전일 대비
12	등락율
256	거래형성 종목수
257	거래형성 비율
25	전일대비 기호

8.17 장시작시간

FID	설명
215	장운영구분(0:장시작전, 2:장종료전, 3:장시작, 4,8:장종료, 9:장마감)
20	체결시간(HHMMSS)
214	장시작 예상잔여시간

8.18 투자자 ticker

FID	설명
216	투자자별 ticker

8.19 주문체결

FID	설명
9201	계좌번호
9203	주문번호
9205	관리자사번
9001	종목코드, 업종코드
912	주문업무분류(J:주식주문, F:선물옵션, JG:주식잔고, FG:선물옵션잔고)
913	주문상태(10:원주문, 11:정정주문, 12:취소주문, 20:주문확인, 21:정정확인, 22:취소확인, 90-92:주문거부)
302	종목명
900	주문수량
901	주문가격
902	미체결수량
903	체결누계금액
904	원주문번호
905	주문구분(+현금내수, -현금매도...)
906	매매구분(보통,시장가...)
907	매도수구분 (1:매도,2:매수)
908	주문/체결시간(HHMMSSMS)
909	체결번호
910	체결가
911	체결량
10	현재가, 체결가, 실시간종가
27	(최우선)매도호가
28	(최우선)매수호가
914	단위체결가
915	단위체결량
938	당일매매 수수료
939	당일매매세금

8.20 잔고

FID	설명
9201	계좌번호
9001	종목코드, 업종코드
302	종목명
10	현재가, 체결가, 실시간종가
930	보유수량
931	매입단가
932	총매입가
933	주문가능수량
945	당일순매수량
946	매도/매수구분
950	당일 총 매도 손익
951	예수금
27	(최우선)매도호가
28	(최우선)매수호가
307	기준가
8019	손익율
397	주식옵션거래단위

8.21 신용잔고

FID	설명
9201	계좌번호
9001	종목코드, 업종코드
917	신용구분
916	대출일
302	종목명
10	현재가, 체결가, 실시간종가
930	보유수량
931	매입단가
932	총매입가
933	주문가능수량
945	당일순매수량
946	매도/매수 구분
950	당일 총 매도 손익
951	예수금
27	(최우선)매도호가
28	(최우선)매수호가
307	기준가
8019	손익율
957	신용금액
958	신용이자
918	만기일
990	당일실현손익(유가)

991	당일실현손익률(유가)
992	당일실현손익(신용)
993	당일실현손익률(신용)
959	담보대출수량

8.22 주식시간외호가

FID	설명
21	호가시간(HHMMSS)
131	시간외 매도호가 총잔량
132	시간외 매도호가 총잔량 직전대비
135	시간외 매수호가 총잔량
136	시간외 매수호가 총잔량 직전대비

8.23 주식당일거래원

FID	설명
141	매도 거래원1
161	매도 거래원 수량1
166	매도 거래원별 증감1
146	매도 거래원 코드1
271	매도 거래원 색깔1
151	매수 거래원1
171	매수 거래원 수량1
176	매수 거래원별 증감1
156	매수 거래원 코드1
281	매수 거래원 색깔1
142	매도 거래원2
162	매도 거래원 수량2
167	매도 거래원별 증감2
147	매도 거래원 코드2
272	매도 거래원 색깔2
152	매수 거래원2
172	매수 거래원 수량2
177	매수 거래원별 증감2
157	매수 거래원 코드2
282	매수 거래원 색깔2
143	매도 거래원3
163	매도 거래원 수량3
168	매도 거래원별 증감3
148	매도 거래원 코드3
273	매도 거래원 색깔3
153	매수 거래원
173	매수 거래원 수량3
178	매수 거래원별 증감3
158	매수 거래원 코드3
183	매수 거래원 색깔3
144	매도 거래원4

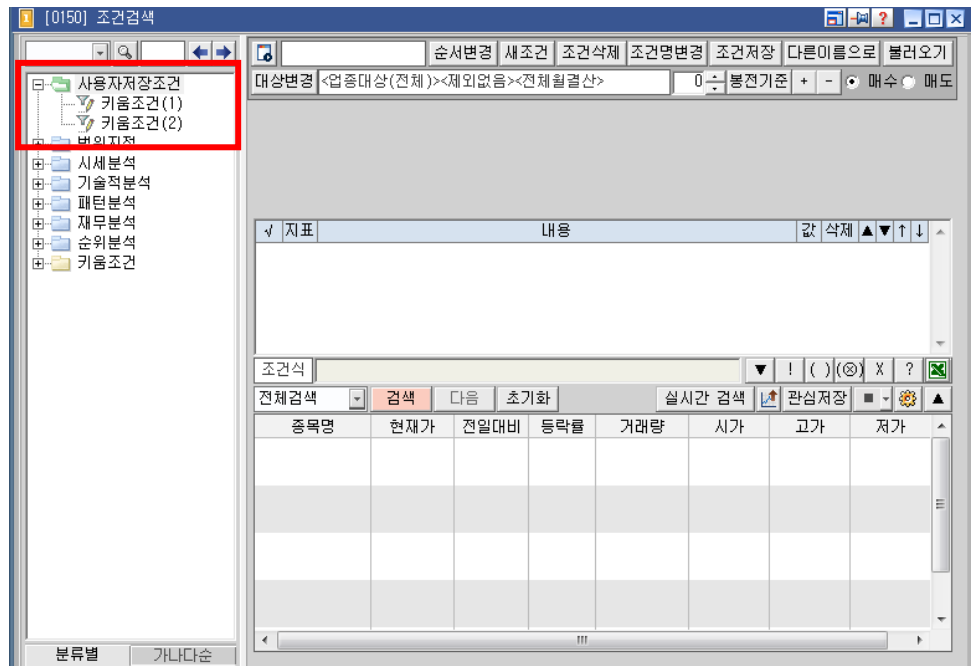
164	매도 거래원 수량4
169	매도 거래원별 증감4
149	매도 거래원 코드4
274	매도 거래원 색깔4
154	매수 거래원4
174	매수 거래원 수량4
179	매수 거래원별 증감4
159	매수 거래원 코드4
284	매수 거래원 색깔4
145	매도 거래원5
165	매도 거래원 수량5
170	매도 거래원별 증감5
150	매도 거래원 코드5
275	매도 거래원 색깔5
155	매수 거래원5
175	매수 거래원 수량5
180	매수 거래원별 증감5
160	매수 거래원 코드5
285	매수 거래원 색깔5
261	외국계 매도 추정합
262	외국계 매도 추정합 변동
263	외국계 매수 추정합
264	외국계 매수 추정합 변동
267	외국계 순매수 추정합
268	외국계 순매수 변동
337	거래소구분 (1, KOSPI, 2:KOSDAQ, 3:OTCCBB, 4:KOSPI200선물, 5:KOSPI200옵션, 6:개 별주식옵션, 7:채권)

9. 키움 Open API+ 추가 기능 가이드

9.1 조건검색(실시간) 가이드

1) 영웅문(HTS) 조건검색[0150] 통해 사용자저장조건을 설정합니다.

- 모든 조건설정은 HTS를 통해 이루어지며, 실시간 조건검색 종목이 API를 통해 송신되는 구조입니다.



2) 서버로부터 사용자 조건식을 불러와서 파일로 로컬에 임시 저장합니다.

-함수 : long GetConditionLoad();

-반환값 : 0(실패), 1(성공)

OpenAPI가 종료되면 임시 저장된파일을 삭제합니다.

OnReceiveConditionVer 이벤트 함수로 로컬에 조건식 저장의 성공(TRUE), 실패(FALSE) 여부와 메시지를 확인하실 수 있습니다.

이벤트에서 성공여부를 확인하신 다음에 아래의 내용으로 진행하시면 됩니다.

OnReceiveConditionVer(BOOL bRet, LPCTSTR sMsg)

파라미터 설명

bRet : 조건식 저장의 성공여부(TRUE, FALSE)

sMsg : 메시지

3) 임시로 저장한 사용자 조건식 파일에서 조건명리스트를 전부 불러옵니다.

-함수 : CString GetConditionNameList();

-반환값 : ex: 조건인덱스1^조건명1;조건인덱스2^조건명2;…;

-“^”으로 조건인덱스와 조건명을 함께 넘겨줍니다.

-“;”(세미콜론)으로 구분하여 사용하시면 됩니다.

4) 조건명(식)에 해당하는 종목리스트를 불러옵니다.

-함수

:BOOL SendCondition(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex, int nSearch);

-반환값 : FALSE(실패), TRUE(성공)

-파라미터 설명

strScrNo : 화면번호

strConditionName :GetConditionNameList()로 불러온 조건명중 하나의 조건명.

nIndex : GetCondionNameList()로 불러온 조건인덱스.

nSearch : 일반조회(0), 실시간조회(1), 연속조회(2)

nSearch 를 0으로 조회하면 단순 해당 조건명(식)에 맞는 종목리스트를

받아올 수 있습니다. 1로 조회하면 해당 조건명(식)에 맞는 종목리스트를 받아

오면서 실시간으로 편입, 이탈하는 종목을 받을 수 있는 조건이 됩니다.

-1번으로 조회 할 수 있는 화면 개수는 최대 10개까지 입니다.

-2은 OnReceiveTrCondition 이벤트 함수에서 마지막 파라미터인 nNext가 “2”로 들어오면 종목이 더 있기 때문에 다음 조회를 원할 때 OnReceiveTrCondition 이벤트 함수에서 사용하시면 됩니다.

-결과값

OnReceiveTrCondition(LPCTSTR sScrNo, LPCTSTR strCodeList, LPCTSTR

strConditionName, int nIndex, int nNext) 이벤트 함수로 종목리스트가 들어옵니다.

-파라미터 설명

sScrNo : 화면번호

strCodeList : 조회된 종목리스트(ex:039490:005930:036570:…;)

strConditionName : 조회된 조건명

nIndex : 조회된 조건명 인덱스

nNext : 연속조회 여부(0:연속조회없음, 2:연속조회 있음)

-실시간 검색 조회 시

OnReceiveRealCondition(LPCTSTR strCode, LPCTSTR strType, LPCTSTR

strConditionName, LPCTSTR strConditionIndex) 이벤트 함수로 편입, 이탈 종목이

실시간으로 들어옵니다.

-파라미터 설명

strCode : 종목코드

strType : 편입("I"), 이탈("D")

strConditionName : 조건명

strConditionIndex : 조건명 인덱스

5) 실시간 조건검색을 중지합니다.

※ 화면당 실시간 조건검색은 최대 10개로 제한되어 있어서 더 이상 실시간 조건검색을 원하지 않는 조건은 중지해야만 카운트 되지 않습니다.

-함수

Void SendConditionStop(LPCTSTR strScrNo, LPCTSTR strConditionName, int
nIndex);

-파라미터 설명

strScrNo : 화면번호

strConditionName : 조건명

nIndex : 조건명 인덱스

9.2 실시간 등록/해제 기능 가이드

실시간 DATA 수신 시 화면별 등록, 화면별 실시간 해제, 전체 해제 등 기능을 제공합니다.

1) 실시간 등록 함수

```
LONG SetRealReg(LPCTSTR strScreenNo, LPCTSTR strCodeList, LPCTSTR  
strFidList, LPCTSTR strOptType)
```

– 파라미터 설명

strScreenNo : 화면번호

strCodeList : 종목코드리스트(ex: 039490:005930;...)

strFidList : FID번호(ex:9001;10;13;...)

9001 - 종목코드

10 - 현재가

13 - 누적거래량

※ FID번호는 KOA Studio에서 실시간 목록을 참고하시기 바랍니다.

strOptType : 타입("0", "1")

타입 "0"은 항상 마지막에 등록한 종목들만 실시간등록이 됩니다.

타입 "1"은 이전에 실시간 등록한 종목들과 함께 실시간을 받고
싶은 종목을 추가로 등록할 때 사용합니다.

※ 종목, FID는 각각 한번에 실시간 등록 할 수 있는 개수는 100개 입니다.

Ex)

strScreenNo = "0001" (화면번호)

strCodeList = "039490:005930;..." (종목코드 리스트)

strFidList = "9001;10;13;..." (FID 번호 리스트)

strOptType = "0" (타입)

SetRealReg(strScreenNo, strCodeList, strFidList, strOptType);

2) 실시간 해제 함수

화면별 실시간 해제와 화면의 종목당 실시간 해제를 하실수 있습니다.

Void SetRealRemove(LPCTSTR strScrNo, LPCTSTR strDelCode)

SetRealReg() 함수로 실시간 등록한 종목만 해제할 수 있습니다.

-파라미터 설명

strScrNo : 화면번호

strDelCode : 실시간 해제할 종목코드

ex)

-화면별 실시간해제

여러 화면번호로 걸린 실시간을 해제하려면 파라미터의 화면번호와 종목코드에
“ALL”로 입력하여 호출하시면 됩니다.

SetRealRemove(“ALL”, “ALL”);

개별화면별로 실시간 해제 하시려면 파라미터에서 화면번호는 실시간해제할
화면번호와 종목코드에는 “ALL”로 해주시면 됩니다.

SetRealRemove(“0001”, “ALL”);

-화면의 종목별 실시간해제

화면의 종목별로 실시간 해제하려면 파라미터에 해당화면번호와 해제할
종목코드를 입력하시면 됩니다.

SetRealRemove(“0001”, “039490”);

9.3 대량 DATA(차트 조회) 수신

기존 GetCommData()함수로 항목당 한 개의 데이터를 받아오는 처리

에서 배열로 조회된 전 항목의 데이터를 한꺼번에 받을 수 있게 함수입니다.

차트 조회처럼 반복데이터가 많은 데이터를 한꺼번에 받아서 처리하고 싶을 때 사용하시면
됩니다.

1) 추가함수

- 함수 : VARIANT GetCommDataEx(LPCTSTR strTrCode, LPCTSTR strRecordName);

- 파라미터 설명

strTrCode : 조회한 TR코드(ex: “OPT10080”)

strRecordName : 출력항목 중에서 얻어오려는 부분 이름

OPT10080 : 주식분봉차트조회요청의 출력부분을 보면 “주식분차트”, “주식분봉차트조회”로 2개 부분으로 나뉘어 있는데 GetCommDataEx()함수 strRecordName에 “주식분봉차트조회”로 설정하면 현재가부터 전일종가까지 차트 데이터를 한번에 얻을 수 있습니다.

2) 사용 예 - 조회한 데이터 전체를 가져와서 처리하고 싶을 때

```
// 주식일봉조회(OPT10081)결과 값을 가져오는 코드
```

```
// 최대 반복수를 가져옵니다.
```

```
int nMaxRow = theApp.m_khOpenApi.GetRepeatCnt(strTRCode, strRQName);
```

※항목의 위치는 KOA Studio의 TR목록 순서로 데이터를 가져옵니다.

예로 OPT10080을 살펴보면 OUTPUT의 멀티데이터의 항목처럼 현재가, 거래량, 체결시간, 시가, 고가, 저가, 수정주가구분 등 순으로 항목의 위치가 0부터 1씩 증가합니다.

```
// OPT10080 출력 항목수는 현재가, 거래량, 체결시간에서 전일종가까지 13개
```

```
Int nMaxCol = 13; // 사용하는 TR별로 출력 항목갯수를 지정, TR출력에 명시된  
항목갯수가 큰 값을 넣으면 안됨.
```

```
int nIndex[2];
```

```
VARIANT vReturn = theApp.m_khOpenApi.GetCommDataEx(strTrCode,  
“주식분봉차트조회”);
```

```
COleSafeArray saMatrix(vReturn);
```

```
VARIANT vTemp;
```

```
VariantInit(&vTemp);
```

```
CString strOneData;
```

```
for(int nRow = 0; nRow < nMaxRow; nRow ++)
```

```
{
```

```
for(int nCol = 0; nCol < nMaxCol; nCol ++)
```

```
{
```

```
nIndex[0] = nRow;
```

```
nIndex[1] = nCol;
```

```
saMatrix.GetElement(nIndex, &vTemp);
```

```

        strOneData = (CString)vTemp.bstrVal;

        ::SysFreeString(vTemp.bstrVal);

    }

}

saMatrix.Clear();

VariantClear(&vReturn);

```

10. 기타

10.1 신용주문 가이드

1. 대상

- 신용주문 대상은 “자기용자”만 제공, “대주”등 나머지는 지원안함.
- 영웅문(HTS) [4989] 키움주문 - 신용설정과 동일한 방식.

2. 신용합수

- 신용주문 전용 **SendOrderCredit()**함수추가
- KOA현금주문 함수를 이용하는 고객이 기존에 작성한 코드변경을 없애고 현금주문과 명확한 구별을 위해 새로 함수를 만들
- 신용 매수주문, 신용 매도주문, 취소/정정주문은 현금주문과 동일한 함수사용

3. 신용주문 설정

- 신용주문의 경우 용자를 받아서 주문내는 것이므로 “대출일” 설정이 매우 중요.
- 대출일은 상환일 기준으로 지정.
- 2월 12일 주문한 신용주문 종목 대출일은 2월 16일이 됨.
- **신용구분**(9번째 함수입력값)과 **대출일**(10번째 함수입력값)설정이 중요(아래 표 참고)

4. KOA 신용매수 주문

- 신용구분 값 “03”로 설정하여 주문, 대출일은 공백처리

5. KOA 신용매도 주문

- 용자상환 주문은 신용구분 “33”로 설정
- OPW00005(체결잔고요청)TR이나 OPW00004(계좌평가현황요청)TR을 이용해서 대출일 조회해야 함. OPW00005(체결잔고요청)TR은 신용잔고 조회용으로 사용.
- **용자합** 주문은 신용구분 “99”로 설정 대출일을 “99991231”로 입력.
- **용자합** 주문은 신용잔고가 5개 이하인 경우만 가능.
- 신용잔고가 7개라면 2종목을 용자상환으로 주문해야 나머지를 용자합으로 주문가능.

〈신용주문 주요 설정값 정리〉

	주문구분	구분코드	대출일	비 고
신용매수	신용매수	03	공백입력	
신용매도	용자상환	33	종목별대출일	
	용자합	99	99991231	5종목 이하잔고만 주문가능

* 유의사항 : 신용주문은 실거래만 가능하며 모의투자 불가