

# 유진투자증권 OPEN API 개발 가이드

## CHAMPION OPENAPI USER GUIDE

## - 목 차 -

1. 소개 .....	3
1.1 Champion OpenAPI란 .....	3
1.2 특징 .....	3
1.3 지원개발환경 .....	3
1.4 제공 서비스 범위 .....	3
2. Champion OpenAPI 구성 .....	4
2.1 폴더 구성 .....	4
2.2 사용 전 필수 등록사항 .....	4
2.2.1 COM(ocx 모듈) 등록 .....	4
2.2.2 실행경로 레지스트리 등록 .....	5
3. Champion OpenAPI 실행하기 .....	6
3.1 Champion OpenAPI 실행 프로세스 .....	6
3.2 버전처리 및 로그인 .....	7
3.2.1 버전처리 .....	7
3.2.2 버전처리 시 유의사항 .....	8
3.2.3 로그인 .....	9
4. Champion OpenAPI interface .....	10
4.1 Method .....	10
4.2 Event .....	11
5. Champion OpenAPI interface 명세서 .....	12
5.1 Method .....	12
5.1.1 접속(통신 및 로그인) .....	12
5.1.2 조회공통 .....	13
5.1.3 Tran 조회 .....	14
5.1.4 FID 조회 .....	16
5.1.5 실시간처리 .....	17
5.1.6 부가기능 .....	17
5.1.7 종목코드 .....	18
5.2 Event .....	21
6. ChampionStudio 소개 .....	22
6.1 ChampionStudio 란 .....	22
6.2 ChampionStudio UI 설명 .....	22
6.3 ChampionStudio 사용 설명 .....	23
6.3.1 리소스 관련 파일 위치 .....	23

6.3.2 로그인 및 접속처리 .....	23
6.3.3 TR 데이터 로딩 화면 .....	24
6.3.4 TR 데이터 입력 방법 .....	25
6.3.5 TR 데이터 응답 처리 .....	26
6.3.6 FID 데이터 로딩 화면 .....	27
6.3.7 FID 데이터 입력 방법 .....	28
6.3.8 FID 데이터 응답 처리 .....	29
6.3.9 Real 데이터 로딩 화면 .....	30
6.3.10 Real 데이터 입력 방법 .....	31
6.3.11 Real 데이터 응답 처리 .....	31

## 1. 소개

### 1.1 Champion OpenAPI 란

챔피언플러스(HTS) 의 시세와 주문 등을 고객이 작성한 프로그램에서 사용할 수 있도록 제공하는 제품입니다.(ocx 모듈형태로 제공)

- 파일명 : ChampionCommAgent.ocx

### 1.1 특징

COM 모듈(ocx 형태) 로 제공하는 Champion OpenAPI는 C, Excel VBA, C#, python 등 다양한 언어에서 로드하여 프로그램을 구현할 수 있습니다.

조회, 주문 등 다양한 서비스 호출 및 응답을 받을 수 있는 API 함수를 제공하여, 누구나 함수를 활용한 개발이 쉽게 가능하도록 합니다.

	내용
컨트롤 모듈	ChampionCommAgent.ocx
라이브러리 형태	OCX – OLE Control Extension, DLL
호환성	Microsoft Windows 10 64비트 이상
개발환경	Microsoft Visual Studio 2012
버전	1.0.0.1

### 1.2 지원개발환경

Champion OpenAPI 는 Visual Studio 2012(32 bit) 버전으로 개발된 모듈입니다.

COM 객체이므로 사용자 편의에 따라 VB, C#, MFC 등 다양한 언어로 프로그램 제작이 가능합니다.

제공하는 서비스 목록은 함께 제공하는 서비스 목록 문서 또는 Champion Studio에서 서비스명을 검색하여 확인할 수 있습니다.

### 1.3 제공 서비스 범위

- 주식
- 지수 선물/옵션, 주식 선물/옵션, 변동성, 섹터 선물
- 해외주식(미국, 중국, 홍콩 \*일본제외)
- 조건검색

## 2. Champion OpenAPI 구성

### 2.1 폴더구성

“챔피언OpenAPI” 설치프로그램으로 프로그램을 설치하면 아래와 같은 폴더가 생성됩니다.

폴더명	기능
/Root	OpenAPI 실행에 필요한 각종 모듈
/Resource	제공 서비스 정보 리소스 파일
/system	통신 및 프로그램 실행에 필요한 환경파일
/mst	마스터정보파일
/user	사용자 환경파일
/EULog, /log	통신모듈 등 시스템 로그파일
/image, /netizen, /filter, /cert	기타 프로그램 실행 관련 파일(보안 등)

OpenAPI 실행에 필요한 필수적인 파일들이 Root 경로에 생성됩니다.

위의 폴더에 적절한 파일이 없을 경우, 프로그램이 정상적으로 실행되지 않습니다.

### 2.2 사용 전 필수 등록 사항

아래의 2가지 등록 사항은, “ChampionOpenAPI” 설치파일을 통해 PC에 설치 시 자동으로 등록됩니다.

아래 내용은, 수동 등록이 필요한 경우 참고해주시기 바랍니다.

#### 2.2.1 COM(ocx 모듈) 등록

ChampionCommAgent.ocx 파일은 windows에 등록되어 있어야 합니다.

모듈은 CMD 창을 이용해 파일이 설치된 경로를 포함한 파일명을 regsvr32 명령어로 등록합니다.

➤ regsvr32 [설치경로\\ChampionCommAgent.ocx]

```

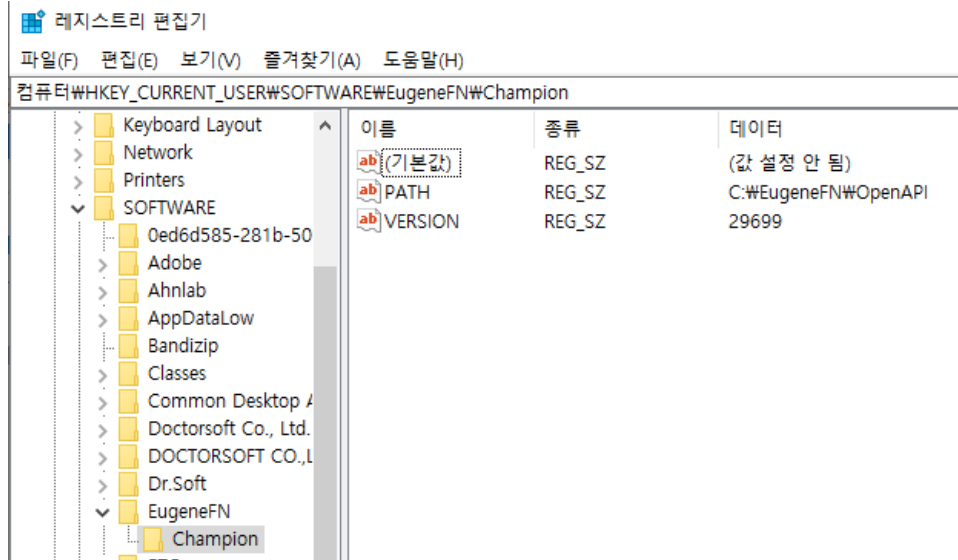
C:\> 관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Eugene>regsvr32 C:\\EugeneFN\\OpenAPI\\ChampionCommAgent.ocx
  
```

## 2.2.2 실행경로 레지스트리 등록

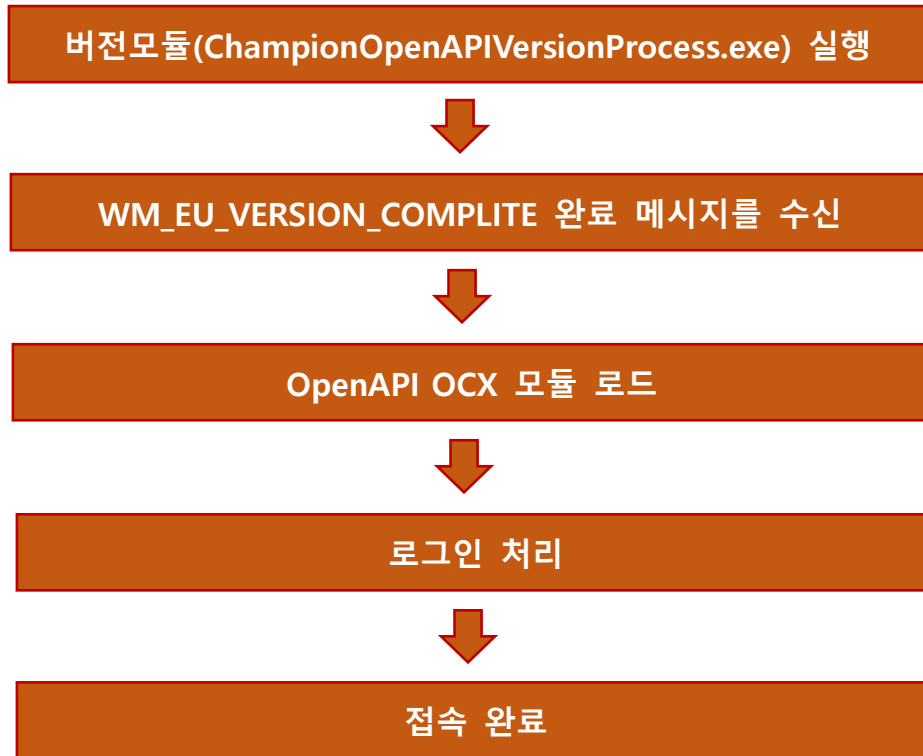
ChampionOpenAPI 가 설치된 경로를 레지스트리에 등록해야 합니다.

➤ 컴퓨터\HKEY\_CURRENT\_USER\SOFTWARE\EugeneFN\Champion



### 3. Champion OpenAPI 실행하기

#### 3.1 Champion OpenAPI 실행 프로세스



### 3.2 버전처리 및 로그인

Champion OpenAPI는 별도 로그인창을 제공하지 않습니다.

모듈에서 제공하는 접속 및 통신 interface를 활용하여 직접 로그인 창을 개발합니다.

프로그램에서 ChampionCommAgent.ocx 로드 전에 버전처리를 먼저 받아, 프로그램을 최신으로 유지한 후 접속을 시도합니다.

#### 3.2.1 버전처리

ChampionCommAgent.ocx 파일을 로드하기 전에,

ChampionOpenAPIVersionProcess.exe를 먼저 실행합니다.

버전처리를 먼저 받지 않는 경우(ocx 모듈 먼저 로드), 버전 처리 시 자신의 프로그램을 종료한 후에 재실행해야 하는 경우가 발생할 수 있습니다.

```
// 버전 Exe 실행 ////////////////////////////////////////
// 레지스트리에 등록된 경로를 얻어온다.
TCHAR      szVersion[MAX_PATH];
TCHAR      szPath[MAX_PATH];
HKEY       hKey;
DWORD      dwSize;

// find out from the registry where the favorites are located.
if(RegOpenKey(HKEY_CURRENT_USER, DEF_VERSION_PROCESS_KEY, &hKey) != ERROR_SUCCESS)
{
    AfxMessageBox("OpenAPI 경로를 확인 할 수 없습니다.");
    return;
}
dwSize = sizeof(szVersion);
RegQueryValueEx(hKey, _T("PATH"), NULL, NULL, (LPBYTE)szVersion, &dwSize);
ExpandEnvironmentStrings(szVersion, szPath, MAX_PATH);
RegCloseKey(hKey);
m_strModulePath = szPath;

//ChampionOpenAPIVersionProcess를 실행해서 버전처리를 실행한다.
DWORD dwRet = 0;

STARTUPINFO stinfo;
PROCESS_INFORMATION prinfo;

char szCommandLine[MAX_PATH] = { 0 };
CFileStatus status;

// ChampionOpenAPIVersionUp.exe에서 파일 복사 및 _system 폴더 삭제 후 버전처리 완료 메시지(WM_EU_VERSION_COMPLITE) 전송
sprintf_s( szCommandLine, MAX_PATH, _T("%s\\ChampionOpenAPIVersionProcess.exe"), m_strModulePath);

if (!CFile::GetStatus( szCommandLine, status ))
{
    CString strLog;
    strLog.Format(_T("버전처리 모듈의 경로를 찾을 수 없습니다. [%s]"), m_strModulePath);
    AfxMessageBox(strLog, MB_TOPMOST);
    return;
}
```

버전처리 프로그램 실행 시 옵션으로 "/" + "메시지 받을 핸들"을 지정해야 합니다.

➤ ChampionOpenAPIVersionProcess.exe [/핸들값]

```
sprintf_s( szCommandLine, MAX_PATH, "%s\\ChampionOpenAPIVersionProcess.exe /%ld", m_strModulePath, GetSafeHwnd());
```



버전처리 완료 메시지 : WM\_EU\_VERSION\_COMPLITE(WPARAM, LPARAM)

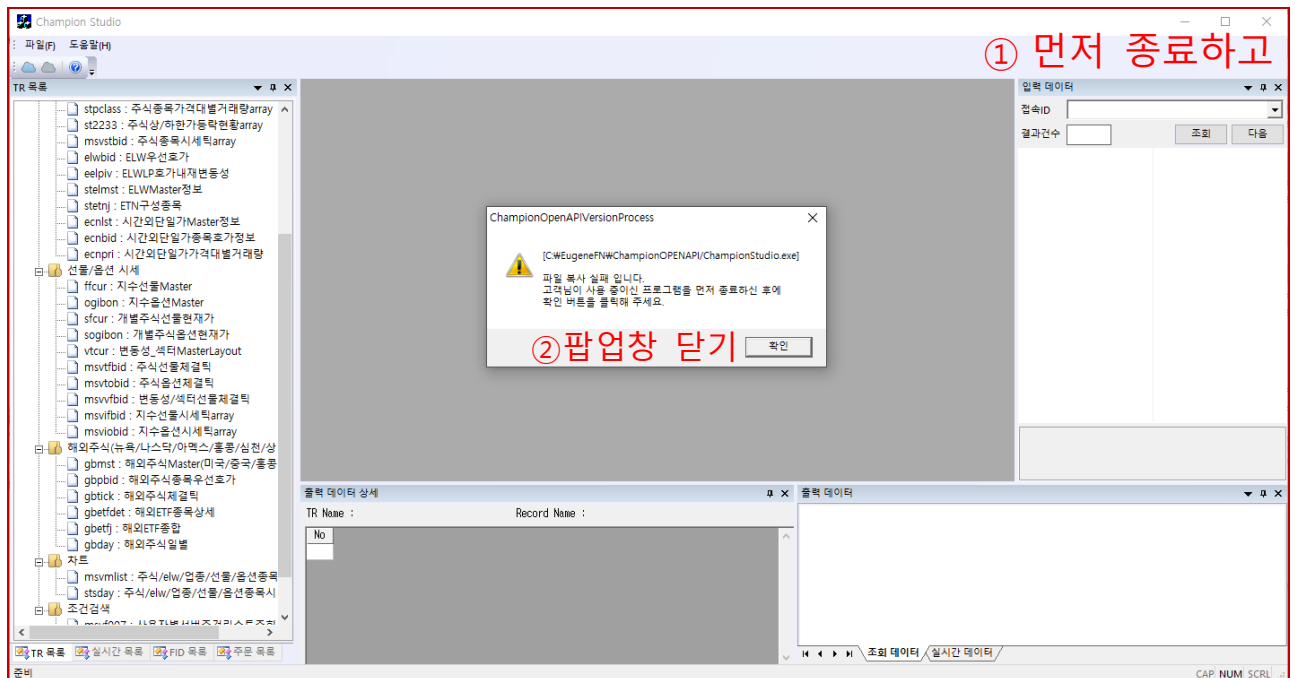
버전처리 완료 시 WM\_EU\_VERSION\_COMPLITE 로 완료메시지를 수신합니다.

WM_EU_VERSION_COMPLITE	<p>WPARAM</p> <p>버전처리 완료 후 OpenAPI 키값</p> <p>LPARAM</p> <p>버전처리 성공 여부 리턴 값</p> <p>1 : 성공</p> <p>0 : 버전처리 실패</p> <p>-1 : 서버로부터 파일다운로드 실패</p> <p>-2 : 파일복사 실패</p> <p>-3 : 알수 없는 에러</p> <p>-4 : ChampionOpenAPIVersionUp.exe 모듈 없음</p> <p>-5 : ChampionOpenAPIVersionUp.exe 실행 실패</p> <p>-6 : 버전서버 IP 취득 실패</p> <p>-7 : 사용자가 취소</p>
------------------------	--

### 3.2.2 버전처리 시 유의사항

버전처리 시, ChampionStudio.exe 나 샘플프로그램 파일이 업데이트 될 경우 “파일복사 실패” 안내 팝업이 제공됩니다.

실행프로그램이 열려있는 상태에서 업데이트를 정상적으로 받기 위해서는, 먼저 해당 프로그램을 종료하신 후 안내 팝업창을 닫아주셔야 합니다.



### 3.2.3 로그인

로그인 및 통신을 진행하기 위해서는 ChampionCommAgent.ocx 를 로드합니다.  
모듈이 설치된 경로를 작업디렉토리로 설정해주어야 합니다.

```
TCHAR szOrgPath[MAX_PATH] = { 0 };
int nLen = GetCurrentDirectory( _countof(szOrgPath), szOrgPath );

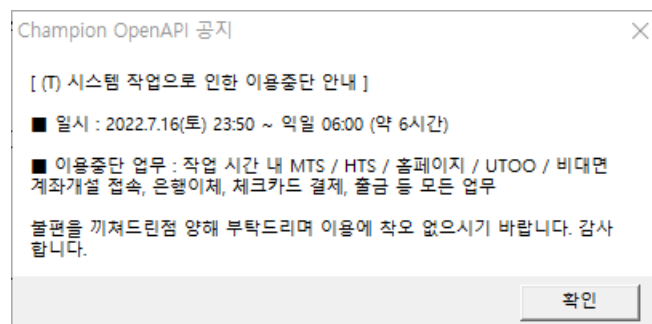
// 레지스트리에 등록된 설치 경로를 세팅한다.
SetCurrentDirectory(m_strModulePath);
```

로그인 함수인 CommLogin 을 통하여 OpenAPI 키 값, 로그인 ID, 로그인 비밀번호, 공인인증서 비밀번호, 제휴사코드(\*필요시)를 입력하여 로그인을 진행합니다.  
이 때 키 값은 버전처리를 진행하고 완료 메시지로 수신한 WPARAM 값을 사용합니다.

로그인 함수 호출 시 공인인증서 비밀번호를 입력하지 않을 경우 시세전용으로 접속되며, 이 경우 인증이 필요한 서비스 등을 사용할 수 없습니다.

CommLogin, CommLoginParter 호출 후 리턴 값이 0보다 작을 경우 로그인 실패이며, GetLastErrorMsg 함수를 사용하여 실패 정보를 확인할 수 있습니다.

로그인 성공 후, 당사에서 반드시 전달해야 하는 공지사항이 있는 경우 하루에 한번 공지사항이 팝업됩니다.



※ 로그인 ID는 반드시 대문자이어야 합니다.

※ 제휴사용 OpenAPI 신청계좌의 경우, 로그인 시 제휴사코드를 함께 입력해야만 합니다.(CommLoginParter Method 이용)

## 4. Champion OpenAPI interface

### 4.1 Method

분류1	분류2	반환	이름	설명
접속	통신	void	CommTerminate	통신연결 해제
		LONG	CommGetConnectState	통신연결상태 확인
	로그인	LONG	CommLogin	개인고객용 로그인 처리
		LONG	CommLoginPartner	제휴사 전용 로그인 처리
		LONG	CommLogout	로그아웃 처리
		LONG	GetLoginState	로그인상태 확인
		Long	GetAccCnt	ID별 사용가능한 계좌번호 개수 조회
		BSTR	GetAccInfo	ID별 사용가능한 계좌번호 조회
		Long	GetLoginMode	접속 모드 확인
조회 공통	RQ ID	LONG	CreateRequestID	조회고유ID 생성(통신 조회 시 필수)
	옵션값	BSTR	GetCommRecvOptionValue	조회 결과 부가정보/옵션값 반환
		void	ReleaseRqId	조회시 사용한 RqId 제거
Tran 조회	입력	LONG	SetTranInputData	Tran조회 항목별 입력값 입력
		LONG	RequestTran	Tran조회 요청
	출력	LONG	GetTranOutputRowCount	Tran조회 응답 데이터 건수 반환
		BSTR	GetTranOutputData	Tran조회 응답 데이터 항목별 값 반환
		LONG	SetTranInputArrayCnt	Tran조회 입력 데이터 건수 설정
		LONG	SetTranInputArrayData	Tran조회 복수건 입력
FID 조회	입력	LONG	RequestPortfolioFid	관심종목형 FID 시세조회 요청
	출력	LONG	GetFidOutputRowCount	FID조회 응답 데이터 건수 반환
		BSTR	GetFidOutputData	FID조회 응답 데이터 항목별 값 반환
실시간 처리	등록	LONG	RegisterReal	실시간 등록
	해제	LONG	UnRegisterReal	실시간등록 해제
	해제	LONG	AllUnRegisterReal	모든 실시간등록 해제
	출력	BSTR	GetRealOutputData	실시간 수신데이터 항목별 반환
부가 기능	메시지	BSTR	GetLastErrMsg	마지막으로 발생한 오류메시지 반환
	경로	BSTR	GetApiAgentModulePath	OpenAPI 에이전트 모듈 파일경로 반환
	옵션	BSTR	SetOptionalFunction	OpenAPI 에서 필요한 값 전달
종목 코드	변환	BSTR	GetExpCode	단축코드로 표준코드 변환
	변환	BSTR	GetShcode	표준코드로 단축코드 변환
	변환	BSTR	GetShCodeByName	종목명으로 단축코드 구하는 함수
	변환	BSTR	GetNameByCode	코드로 종목명 구하는 함수
	변환	BSTR	GetFutShCode	선물코드 구하는 함수
	변환	BSTR	GetOptShCode	옵션코드 구하는 함수
	변환	BSTR	GetOptionATMPrice	옵션 ATM 가격

	변환	BSTR	GetMarketKubun	종목코드의 장구분 값 제공
	변환	BSTR	GetUpjongByCode	코드로 해당 업종코드 제공
	변환	BSTR	GetOverseaStockInfo	해외주식 종목정보 제공

## 4.2 Event

분류1	반환	이름	설명
응답	void	OnGetTranData	Tran 조회에 대한 응답
	void	OnGetFidData	Fid 조회에 대한 응답
	void	OnGetRealData	실시간 데이터 수신
	void	OnAgentEventHandler	에이전트이벤트 핸들러 (통신/ 공지 이벤트 등)

## 5. Champion OpenAPI interface 명세서

### 5.1 Method

#### 5.1.1 접속(통신 및 로그인)

함수명	<b>void CommTerminate(Long bSocketClose)</b>	
설 명	통신모듈 종료	
인자값	LONG bSocketClose	TRUE:소켓 단절, FALSE: 내부핸들
비고	OpenAPI에서는 인자값에 TRUE만 사용	

함수명	<b>LONG CommGetConnectState()</b>	
설 명	통신 연결상태 반환	
	-1:소켓연결 실패 0:소켓연결 단절 1:정상	
인자값		
비고		

함수명	<b>LONG CommLogin(LONG nVersionPassKey, LPCTSTR sUserID, LPCTSTR sPwd, LPCTSTR sCertPwd)</b>	
설 명	통신모듈에 접속 및 로그인 처리를 하는 함수(개인 고객용)	
	<pre> #define LOGIN_SUCCESS 0 #define ERR_COMM_NOTCONNECT -1 //서버 접속 실패 #define LOGIN_ERR_PASSOVER -2 //비밀번호 5회이상 오류(지점방문) #define LOGIN_ERR_SERVERSYSTEM -3 //System 구분 오류 #define LOGIN_ERR_NOT_USER -4 //사용자 정보가 없음. #define LOGIN_ERR_UNUSE -5 //사용 권한이 없음. #define LOGIN_ERR_MENU -6 //메뉴파일이 존재하지 않음. #define LOGIN_ERR_PASSERR -7 //비밀번호 오류 #define LOGIN_ERR_USERFLAG -8 //사용자 구분값 오류 #define LOGIN_ERR_MULTICOM -9 //다중접속이 허용되지 않아 접속이 불가 #define LOGIN_ERR_CERTPASSOVER -20 //공인인증 비밀번호 오류 초과 #define LOGIN_ERR_CERTERR -21 //공인인증서 만료/폐기/미발급 등 오류 #define LOGIN_ERR_CERTUNREG -22 //공인인증서 미등록 #define LOGIN_ERR_CERT_PASSERR -23 //공인인증서 비밀번호 오류 #define LOGIN_ERR_CERT_CANCEL -24 //공인인증서 취소상태 #define LOGIN_ERR_CERT_LOCK -25 //공인인증서 잠긴 상태 #define LOGIN_ERR_CERT_ISSUE -26 //공인인증서 재발급 #define LOGIN_ERR_UNINITIALIZE -98 //INITIALIZE 되어 있지 않음 #define LOGIN_ERR_ETC -99 //기타 오류           </pre>	

인자값	LONG nVersionPassKey	버전처리 실행 시 넘어오는 버전처리 체크용 키값
	LPCTSTR sUserID	사용자 ID
	LPCTSTR sPwd	사용자 로그인 비밀번호
	LPCTSTR sCertPwd	사용자 공동인증서 비밀번호
비고	공동인증서 비밀번호를 입력하지 않을 경우 시세전용으로 접속	

함수명	<b>LONG CommLoginPartner(LONG nVersionPassKey, LPCTSTR sUserID, LPCTSTR sPwd, LPCTSTR sCertPwd, LPCTSTR sPartnerCode)</b>	
설 명	통신모듈에 접속 및 로그인 처리를 하는 함수(제휴사 고객용)	
	<pre> #define LOGIN_SUCCESS 0 #define ERR_COMM_NOTCONNECT -1 //서버 접속 실패 #define LOGIN_ERR_PASSOVER -2 //비밀번호 5회이상 오류(지점방문) #define LOGIN_ERR_SERVERSYSTEM -3 //System 구분 오류 #define LOGIN_ERR_NOT_USER -4 //사용자 정보가 없음. #define LOGIN_ERR_UNUSE -5 //사용 권한이 없음. #define LOGIN_ERR_MENU -6 //메뉴파일이 존재하지 않음. #define LOGIN_ERR_PASSERR -7 //비밀번호 오류 #define LOGIN_ERR_USERFLAG -8 //사용자 구분값 오류 #define LOGIN_ERR_MULTICOM -9 //다중접속이 허용되지 않아 접속이 불가 #define LOGIN_ERR_CERTPASSOVER -20 //공인인증 비밀번호 오류 초과 #define LOGIN_ERR_CERTERR -21 //공인인증서 만료/폐기/미발급 등 오류 #define LOGIN_ERR_CERTUNREG -22 //공인인증서 미등록 #define LOGIN_ERR_CERT_PASSERR -23 //공인인증서 비밀번호 오류 #define LOGIN_ERR_CERT_CANCEL -24 //공인인증서 취소상태 #define LOGIN_ERR_CERT_LOCK -25 //공인인증서 잠긴 상태 #define LOGIN_ERR_CERT_ISSUE -26 //공인인증서 재발급 #define LOGIN_ERR_UNINITIALIZE -98 //INITIALIZE 되어 있지 않음 #define LOGIN_ERR_ETC -99 //기타 오류 </pre>	
인자값	LONG nVersionPassKey	버전처리 실행 시 넘어오는 버전처리 체크용 키값
	LPCTSTR sUserID	사용자 ID
	LPCTSTR sPwd	사용자 로그인 비밀번호
	LPCTSTR sCertPwd	사용자 공동인증서 비밀번호
	LPCTSTR sPartnerCode	제휴사코드번호 ※OpenAPI 서비스 등록 시 설정한 제휴사코드번호를 입력.
비고	공동인증서 비밀번호를 입력하지 않을 경우 시세전용으로 접속	

함수명	<b>LONG CommLogout(LPCTSTR sUserID)</b>	
설 명	통신모듈에 접속된 상태에서 로그아웃을 처리를 하는 함수	
	0 : 실패 1 : 성공	
인자값	LPCTSTR sUserID	사용자 ID
비고	서버에 전송만 하고 응답은 처리하지 않음.	

함수명	LONG GetLoginState()
설 명	로그인 상태 반환
	0 : 실패 1 : 성공
비고	서버에 전송만 하고 응답은 처리하지 않음.

함수명	LONG GetAccCnt()
설 명	OpenAPI 사용 계좌 개수 반환
	ID에 등록된 OpenAPI 보유계좌 개수 응답.
비고	

함수명	BSTR GetAccInfo()
설 명	OpenAPI 사용 계좌번호 반환
비고	ID에 등록된 OpenAPI 계좌번호를 반환하며, 세미콜론(;) 으로 분리함. Ex ) 계좌1;계좌2;

함수명	LONG GetLoginMode(LONG nOption)
설 명	OpenAPI 접속 모드 확인
인자 값	LONG nOption      0
	0 : 실제 서버 접속 1 : 테스트용 서버 접속
비고	

### 5.1.2 조회공통

함수명	LONG CreateRequestID()
설 명	통신조회고유ID 생성 및 반환
	특정 ID를 제외하고, 101~2000 사이에 유일한 RQ값 반환
인자값	Long nOptionType      수신데이터의 정보를 얻어오는 함수
비고	<b>- nOptionType 정의 ( ChampionCommAPIDLL.h )</b> namespace CommRecvOpt{ enum { TranCode = 0,      //Tr코드 PrevNextCode = 1,      //연속데이터 구분(사용안함) PrevNextKey = 2,      //연속조회키 MsgCode = 3,      //메시지코드 Msg = 4,      //메시지 SubMsgCode = 5,      //부가메시지코드 SubMsg = 6,      //부가메시지 Error = 7,      //에러여부 }

	}; } - 예시 CString strTrCode(pComAgent->GetCommRecvOptionValue(CommRecvOpt::TranCode)); //Tr코드 CString strNext(pComAgent->GetCommRecvOptionValue(CommRecvOpt::PrevNextKey)); //연속조회 키 CString strMessage(pComAgent->GetCommRecvOptionValue(CommRecvOpt::Msg)); //메시지 CString strSubMsg(pComAgent->GetCommRecvOptionValue(CommRecvOpt::SubMsgCode)); //부가 메시지 CString strError(pComAgent->GetCommRecvOptionValue(CommRecvOpt::Error)); //에러여부
--	--

함수명	<b>BSTR GetCommRecvOptionValue(LONG nOptionType)</b>	
설 명	조회결과 부가정보/옵션값 반환	
	특정 ID를 제외하고, 101~2000 사이에 유일한 RQ값 반환	
인자값	Long nOptionType	수신데이터의 정보를 얻어오는 함수
비고	<b>- nOptionType 정의 ( ChampionCommAPIDLL.h )</b> <pre> namespace CommRecvOpt{     enum     {         TranCode = 0,    //Tr코드         PrevNextCode    = 1,    //연속데이터 구분(사용안함)         PrevNextKey     = 2,    //연속조회키         MsgCode = 3,      //메시지코드         Msg             = 4,    //메시지         SubMsgCode      = 5,    //부가메시지코드         SubMsg          = 6,    //부가메시지         Error           = 7,    //에러여부     }; } </pre> <b>- 예시</b> CString strTrCode(pComAgent->GetCommRecvOptionValue(CommRecvOpt::TranCode)); //Tr코드 CString strNext(pComAgent->GetCommRecvOptionValue(CommRecvOpt::PrevNextKey)); //연속조회 키 CString strMessage(pComAgent->GetCommRecvOptionValue(CommRecvOpt::Msg)); //메시지 CString strSubMsg(pComAgent->GetCommRecvOptionValue(CommRecvOpt::SubMsgCode)); //부가 메시지 CString strError(pComAgent->GetCommRecvOptionValue(CommRecvOpt::Error)); //에러여부	

함수명	<b>void ReleaseRqId(LONG nRqId)</b>	
설 명	통신조회 시 사용한 RqID 제거	
인자값	Long nRqId	통신조회시 사용한 RqId
비고		

### 5.1.3 Tran조회

함수명	<b>LONG SetTranInputData(long nRqId, LPCTSTR strTrCode, LPCTSTR strRecName, LPCTSTR strItem, LPCTSTR strValue)</b>	
설 명	Tran 조회 항목별 입력값 입력	
	0 : 실패 1 : 성공	



	-22 : 계좌 비밀번호 암호화에 필요한 계좌 정보가 없는 경우 -25 : 조회 불가능한 계좌번호 입니다	
인자값	LONG nRqld	통신조회시 필요한 RQID
	LPCTSTR strTrCode	서비스명
	LPCTSTR strRecName	레코드명
	LPCTSTR strItem	아이템명
	LPCTSTR strValue	값
비고	통신요청에 필요한 Input에 필요한 값을 아이템 별로 입력해 두는 함수	

함수명	LONG RequestTran(long nRqld, LPCTSTR sTrCode, LPCTSTR sNextKey, LONG nRequestCount)	
설 명	Tran 조회	
	-1 : 소켓송신 에러 -18 : 레코드 구성 오류 -19 : 입력데이터 구성에 오류가 있습니다 -21 : Input값이 세팅되지 않은 RQID 이거나 통신정보가 없는 서비스 -22 : RQ데이터 부족 -25 : 조회 불가능한 계좌번호 -26 : 초당 전송 횟수 제한 -27 : 동기식 TR로 전송 제한 -28 : 존재하지 않는 연속키값이거나 연속조회가 불가능한 TR -99 : 로그아웃 또는 소켓 단절 상태	
인자값	LONG nRqld	통신조회시 필요한 RQID
	LPCTSTR sTrCode	서비스명
	LPCTSTR sNextKey	다음키값
	LONG nRequestCount	조회 요청 데이터 개수
비고	반환값의 에러코드는 ChampionCommAPI.dll.h 파일의 ChampionCommAPI_SendRq return Value 참조 서비스에 따라 데이터 개수는 인자값인 nRequestCount 가 아닌 Input 값으로 설정해 주어야 하는 서비스도 있음.  <b>※연속조회 방법</b> 1 최초 서비스 조회 : SetTranInputData 로 필요한 인풋값을 작성 후, RequestTran 으로 조회 1.1 최초 서비스 응답 : OnGetTranData 로 응답 데이터 수신 1.1.1 수신받은 TranData에서 GetCommRecvOptionValue(nOptionType) 을 통해 연속조회 키 값을 확인 (nOptionType = 2 : 연속조회 키 값) 1.1.2 연속조회 키 값이 있다면, 연속조회 할 값이 있음. 2 연속조회 키 값을 가지고, 기존 인풋값과 동일하게 RequestTran 으로 연속조회 2.1 연속조회를 위해 RequestTran 사용시, 3번째 인자 값으로 직전에 응답받은 연속조회 키 값으로 셋팅하여 서비스 송신. 2.2 연속조회 한 응답을 다시 OnGetTranData 로 응답 데이터 수신	

	2.2.1 최초 조회와 동일하게 연속조회 키 값이 있는 지 여부를 서비스 응답시 마다 반복하여 확인.
--	--

함수명	LONG GetTranOutputRowCnt(LPCTSTR strTrCode, LPCTSTR strRecName)		
설 명	TRAN 수신 데이터 건수		
인자값	LPCTSTR strTrCode	서비스명	
	LPCTSTR strRecName	출력 레코드명	
비고	데이터를 받았을 때 해당 서비스 출력 레코드의 데이터 개수를 반환 합니다.		

함수명	BSTR GetTranOutputData(LPCTSTR strTrCode, LPCTSTR strRecName, LPCTSTR strItemName, LONG nRow)		
설 명	Tran 수신 데이터		
인자값	BSTR strTrCode	서비스명	
	BSTR strRecName	출력 레코드명	
	BSTR strItemName	아이템명	
	LONG nRow	데이터의 인덱스	
비고	nRow의 값은 싱글 데이터일 경우 0으로 입력		

함수명	LONG SetTranInputArrayCnt(long nRqld, LPCTSTR strTrCode, LPCTSTR strRecName, long nRecCnt)		
설 명	배열성 Input 데이터를 입력할 경우 입력 데이터의 개수를 설정하는 함수		
	0 : 실패 1 : 성공		
인자값	LONG nRqld	통신조회시 필요한 RQID	
	LPCTSTR strTrCode	서비스명	
	LPCTSTR strRecName	레코드명	
	LONG nRecCnt	데이터개수	
비고			

#### 5.1.4 FID조회

<b>함수명</b>	<b>LONG RequestPortfolioFid(LONG nRqld, LPCTSTR sFidName, LPCTSTR strOutputFidList, LPCTSTR strCodeList, BYTE chDelimiter, LONG nRequestCount)</b>		
<b>설 명</b>	관심종목형 FID 시세조회 요청		
	0 : 실패 1 : 성공		
<b>인자값</b>	LONG nRqld	통신조회시 필요한 RQID	
	LPCTSTR sFidName	FID 서비스명	
	LPCTSTR strOutputFidList	FID 번호	
	LPCTSTR strCodeList	아이템명	

	BYTE chDelimiter	입력값
	LONG nRequestCount	데이터의 개수
비고		

함수명	<b>LONG GetFidOutputRowCnt(LONG nRequestId)</b>	
설 명	FID 응답 데이터의 출력 레코드 데이터 개수	
	0 : 실패 1 : 성공	
인자값	LONG nRequestid	통신조회시 필요한 rqid
비고		

함수명	<b>BSTR GetFidOutputData(LONG nRequestId, LPCTSTR strFid, LONG nRow)</b>	
설 명	FID 응답 데이터 반환	
인자값	LONG nRequestid	통신조회시 필요한 rqid
	LPCTSTR strFid	FID 번호
	LONG nRow	출력 데이터의 인덱스
비고		

### 5.1.5 실시간처리

함수명	<b>LONG RegisterReal(short nRealType, LPCTSTR strRealKey)</b>	
설 명	실시간 등록	
	0 : 실패 1 : 성공 -1 : 로그아웃 상태 -2 : 제공되지 않는 실시간 종류 -31 : 실시간 패킷 구조체 없음	
인자값	SHORT nRealType	실시간 데이터 종류
	LPCTSTR strRealKey	실시간 등록 키값(종목코드, userID, 계좌번호 등) 키를 여러 개 등록하고 싶은 경우, 인풋 레코드 규격에 맞춰 ";" 세미 콜론으로 처리한다. (최대 등록 개수 : 100건) Ex ) RegisterReal(21, "KR7000660001;KR7005930003;")
비고	제공되는 실시간 데이터 종류는 「RealTran」 문서 참조	

함수명	<b>LONG UnRegisterReal(short nRealType, LPCTSTR strRealKey)</b>	
설 명	실시간 해제	
	0 : 실패 1 : 성공 -1 : 로그아웃 상태	

	-2 : 제공되지 않는 실시간 종류 -3 : 키값 없음	
인자값	SHORT nRealType	실시간 데이터 종류
	LPCTSTR strRealKey	실시간 등록 키값(종목코드, userID, 계좌번호 등)
비고	제공되는 실시간 데이터 종류는 「RealTran」 문서 참조	

함수명	LONG AllUnRegisterReal()	
설 명	등록된 모든 실시간 해제	
	0 : 실패 1 : 성공	
인자값		
비고		

함수명	BSTR GetRealOutputData(short nRealType, LPCTSTR strItemName)	
설 명	실시간 수신 시 데이터를 얻어오는 함수	
인자값	SHORT nRealType	실시간 데이터 종류
	LPCTSTR strItemName	아이템명
비고	제공되는 실시간 데이터 종류는 「RealTran」 문서 참조	

### 5.1.6 부가기능

함수명	BSTR GetLastErrMsg()	
설 명	마지막 오류메시지 반환	
인자값		
비고	함수의 반환값이 에러일 경우 GetLastErrorMsg로 메시지 가져와서 표시	

함수명	BSTR GetApiAgentModulePath()	
설 명	OpenAPI 모듈파일이 위치한 경로 반환	
인자값		
비고		

함수명	BSTR SetOptionalFunction(long nOption, long nValue1, long nValue2, LPCTSTR strValue1, LPCTSTR strValue2)	
설 명	OpenAPI에서 필요한 값을 전달하기 위한 함수	
인자값	LONG nOption	옵션 종류

	LONG nValue1	정수형 값
	LONG nValue2	정수형 값
	BSTR strValue1	문자형열 값
	BSTR strValue2	문자형열 값
비고	미사용	

### 5.1.7 종목코드

함수명	<b>BSTR</b> GetExpCode(LPCTSTR sShCode)	
설 명	단축코드로 표준코드를 구하는 함수	
인자값	LPCTSTR sShCode	단축코드
비고	000660 → KR7000660001	

함수명	<b>BSTR</b> GetShCode(LPCTSTR sExpCode)	
설 명	표준코드로 단축코드를 구하는 함수	
인자값	LPCTSTR sExpCode	표준코드
비고	KR7000660001 → 000660	

함수명	<b>BSTR</b> GetShCodeByName(LPCTSTR szName)	
설 명	종목명으로 단축코드를 구하는 함수	
인자값	LPCTSTR szName	종목명
비고		

함수명	<b>BSTR</b> GetNameByCode(LPCTSTR sCode)	
설 명	코드로 종목명을 구하는 함수	
인자값	LPCTSTR sCode	단축코드 또는 표준코드
비고		

함수명	<b>BSTR</b> GetUpjongByCode(LPCTSTR sCode)	
설 명	코드로 해당 업종코드를 구하는 함수	
인자값	LPCTSTR sCode	단축코드 또는 표준코드
비고		

<b>함수명</b>	<b>BSTR GetFutShCode(LONG nType, LONG nIndex)</b>	
<b>설 명</b>	선물 코드를 구하는 함수	
<b>인자값</b>	LONG nType	선물 종류
	LONG nIndex	최근월물, 차기월물, 차차기월물, 차차차기월물 (0 Base)
<b>비고</b>	- 선물종류 ( ChampionCommAPIDLL.h 파일에 정의 ) <pre> #define FUTURE_TYPE_KOSPI200 10 #define FUTURE_TYPE_KOSPI200_SPREAD 20 #define FUTURE_TYPE_STAR 30 #define FUTURE_TYPE_STAR_SPREAD 40 #define FUTURE_TYPE_MINIKOSPI 50 #define FUTURE_TYPE_MINIKOSPI_SPREAD 60 #define FUTURE_TYPE_KOSDAQ150 70 #define FUTURE_TYPE_KOSDAQ150_SPREAD 80 </pre>	

<b>함수명</b>	<b>BSTR GetOptionATMPrice()</b>	
<b>설 명</b>	옵션 ATM 가격을 구하는 함수	
<b>인자값</b>		
<b>비고</b>		

<b>함수명</b>	<b>BSTR GetOptShCode(LONG nMonthIndex, LONG nCallorPut, LONG nIndex)</b>	
<b>설 명</b>	옵션 코드를 구하는 함수	
<b>인자값</b>	LONG nMonthIndex	최근월물, 차기월물, 차차기월물, 차차차기월물 (0 Base)
	LONG nCallorPut	콜옵션, 풋옵션 1:콜옵션 2:풋옵션
	LONG nIndex	ATM 기준 +- index (0 Base)
<b>비고</b>		

<b>함수명</b>	<b>short GetMarketKubun(LPCTSTR sShCode, LPCTSTR sMarkets)</b>	
<b>설 명</b>	종목코드의 시장 구분값을 가져오는 함수	
	주식, 선물/옵션, ELE, ETF 등 종목의 시장을 구분 해야할때 사용.	
	<pre> #define CODE_JONGMOK_D 1 // 상장 종목 #define CODE_KJONGMOK_D 2 // 코스닥 종목 #define CODE_OTCBB_D 3 // 제3시장(OTCBB) #define CODE_KONEX_D 8 // 코넥스 #define CODE_FUTURE_D 24 // 선물 #define CODE_OPTION_D 25 // 옵션 #define CODE_UPJONG_D 26 // 장내업종 </pre>	

	#define CODE_KUPJONG_D 33 // 코스닥 업종	
인자값	LPCTSTR sCode	단축코드 또는 표준코드
	LPCTSTR sMarkets	시장구분값
비고		

함수명	BSTRT GetOverseaStockInfo(LPCTSTR sCode, LONG nIndex)	
설 명	해외주식 종목정보를 구하는 함수	
인자값	LPCTSTR sCode	해외 종목 표준 코드 ( 거래소코드+종목코드 )
	LONG nIndex	옵션 구분 인덱스
비고	- 해외종목코드는 「Tran서비스IO」 문서 파일 참조 - nIndex 옵션 설명 0 : 자체단축코드(shortcode) 1 : 자체표준코드(expcode) 5 : 한글명(hname) 6 : 영문명(ename) 7 : 국제표준코드(isincode) 8 : 업종코드(upcodes) 9 : 종목분류(div_code, 1:주식, 7:ETF) 10 : 주문수량단위(tradeunit) 11 : 가격 소숫점 자릿수(floatpoint) 12 : 틱사이즈(ticktype, 0:미정의, 나머지는 표 참조) 13 : 외환코드(currency) by Code 14 : 영업일자(bymd) 15 : 계정계용 거래소 코드 16 : 계정계용 거래소+심볼코드를 풀코드로 변환	

## 5.2 Event

함수명	void OnGetTranData(long nRequestId, LPCTSTR pBlock, long nBlockLength)	
설 명	Tran 형식의 데이터 수신시 호출되는 이벤트	
인자값	long nRequestId	조회 요청시 입력한 RQID
	LPCTSTR pBlock	전체 데이터 문자열
	long nBlockLength	데이터의 길이
비고		

함수명	void OnGetFidData(long nRequestId, LPCTSTR pBlock, long nBlockLength)	
설 명	Fid 형식의 데이터 수신시 호출되는 이벤트	
인자값	long nRequestId	조회 요청시 입력한 RQID
	LPCTSTR pBlock	전체 데이터 문자열
	long nBlockLength	데이터의 길이
비고		

함수명	void OnGetRealData(short nPBID, LPCTSTR strRealKey, , LPCTSTR pBlock, long nBlockLength)		
설 명	실시간 데이터 수신시 호출되는 이벤트		
인자값	short nPBID	실시간 데이터 종류	
	LPCTSTR strRealKey	실시간 등록 키값	
	LPCTSTR pBlock	전체 데이터 문자열	
	long nBlockLength	데이터의 길이	
비고			

함수명	void OnAgentEventHandler(long nEventType, long nParam, LPCTSTR strParam)		
설 명	소켓 단절 또는 다중 접속시 호출되는 이벤트		
인자값	long nEventType	이벤트 종류	
	long nParam	정수형 값 ( 미정 )	
	LPCTSTR strParam	문자열 값 ( 미정 )	
비고	<b>- nEventType 정의</b> NOTI_MULTICONNECT_DENIED 50 //다중접속에 의한 접속 해제 NOTI_SOCKET_CLOSED 51 //통신 단절		

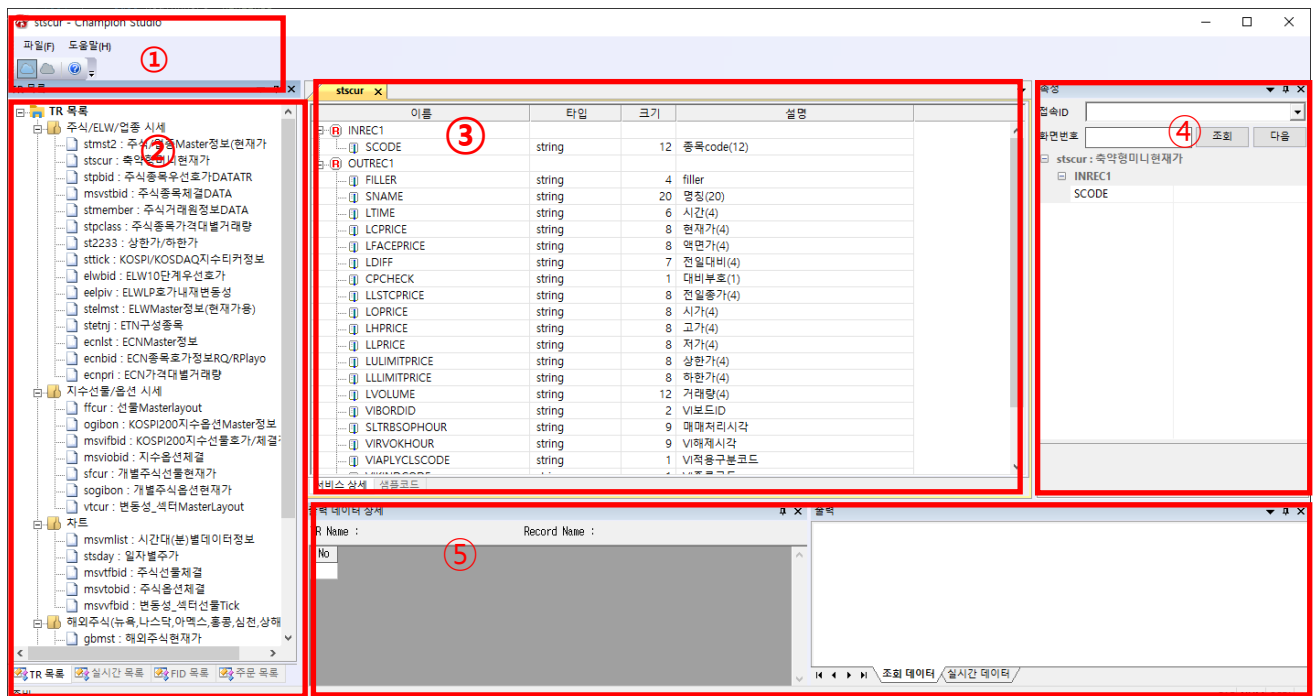


## 6. ChampionStudio 소개

### 6.1 ChampionStudio 란

Champion OpenAPI를 이용한 프로그램 개발 시 편의성을 높이기 위해, 당사에서 제공하는 서비스에 대한 I/O 등 통신규약을 체크하고 샘플 소스 코드를 확인할 수 있는 제품입니다.

### 6.2 ChampionStudio UI 설명



#### ① 메뉴 및 툴바 영역

- 메뉴 : 접속/해제 및 정보/종료 기능 처리
- 툴바 : 접속/해제 및 정보 관련 기능 처리

#### ② TR 목록 뷰

- 4 가지 구분으로 테스트할 TR 리스트 목록을 표시한다.
- 하단 탭을 이용 각 리스트 항목을 이동할 수 있다.

#### ③ 서비스 정보 뷰

- 여러 개 리소스 창이 열려있는 경우 상단 탭을 이용 이동할 수 있다.
- 선택한 특정 리소스 I/O 정보를 표시한다.
- 하단 탭으로 I/O, 샘플코드 정보 내용을 이동할 수 있다.

#### ④ 통신 요청 값 입력 뷰

- 선택된 리소스에 해당 하는 Input 데이터를 입력할 수 있다.
- 선택된 리소스에 해당 하는 동작버튼이 상단에 표시된다.

#### ⑤ 통신 응답 출력 뷰

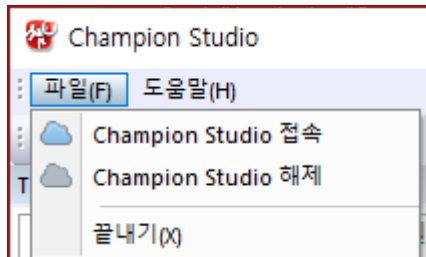
- 조회 응답에 대한 결과를 표시한다.
- 우측영역은 응답상태등 간략한 정보를 표시하며, 하단 탭을 이용하여 조회/리얼 형태에 응답 상태를 확인 할 수 있다.
- 좌측영역은 응답 데이터에 대한 상세한 값을 표시한다.

## 6.3 ChampionStudio 사용 설명

### 6.3.1 리소스 관련 파일 위치

- OpenApi 모듈 실행 폴더 아래에 Resource 폴더 존재해야 합니다.
- Resource 폴더에 제공할 리소스 리스트 관련 파일(\*.dat), 리소스 파일(\*.res)이 있어야 스튜디오 목록이 정상적으로 표시됩니다.

### 6.3.2 로그인 및 접속 처리

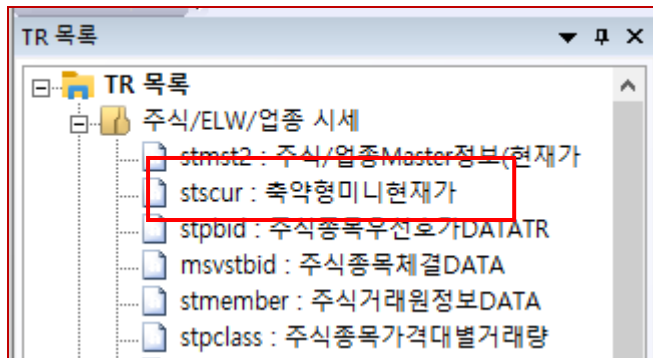


상단 메뉴를 이용하여 OpenApi 접속 및 해제 처리  
(접속 시 OpenApi OCX 모듈을 로드합니다.)

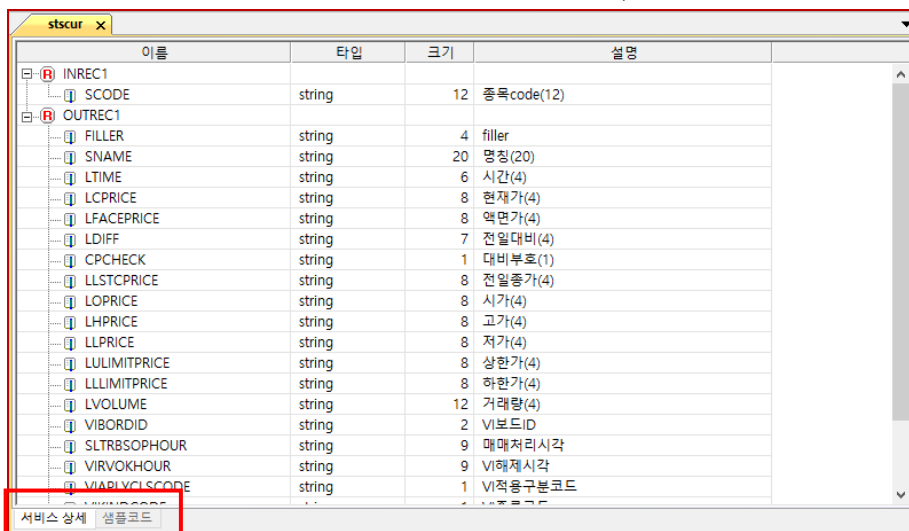
 로그인 창을 보여주는 이미지입니다. 창에는 '고객 ID', 'ID 비밀번호', '인증비밀번호', '제휴사코드' (선택사항) 입력 필드가 있습니다. 각 필드에는 입력 가이드 텍스트가 표시되어 있습니다. '개발서버 접속' 체크박스가 선택되어 있습니다. 하단에는 '로그인' (녹색)과 '취소' (회색) 버튼이 있습니다.

접속 시 로그인창 실행되며,  
ID, 비밀번호, 인증 비밀번호를 입력하고 접속처리 해야 정상실행됩니다.  
OpenAPI 사용 신청 시 제휴사 계좌로 신청한 경우,  
제휴사코드 체크 후 해당 코드를 입력하고 로그인처리 해야합니다.

### 6.3.3 TR 데이터 로딩 화면

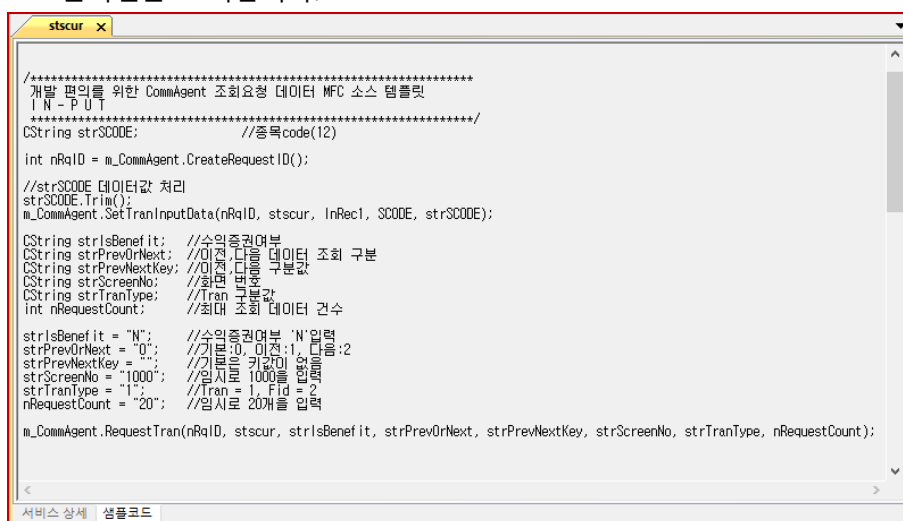


리스트 목록 중 처리할 항목을 더블클릭 하면, 데이터가 로드됩니다.



리소스 정보는 I/O에 해당하는 정보를 표시합니다.

하단탭으로 정보화면을 변경할 수 있으며, 「샘플코드」 선택 시 MFC 개발용 샘플화면을 표시합니다.



#### 6.3.4 TR 데이터 입력 방법

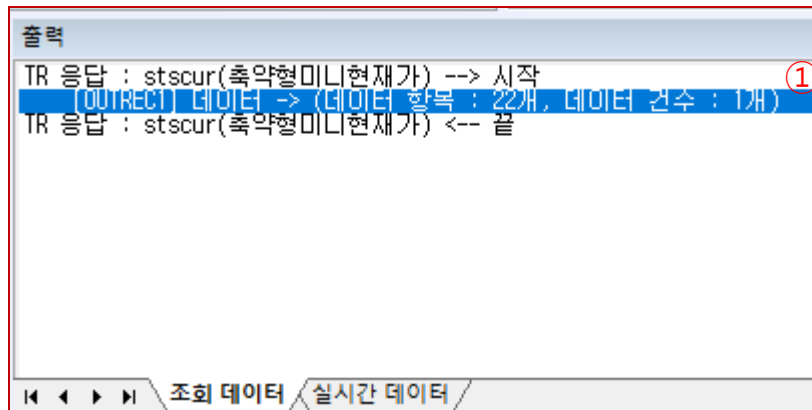
- ① 로그인 중인 아이디 정보를 표시합니다.
- ② 입력 아이템값 적용 후 조회버튼 클릭으로 초기 데이터를 요청합니다.
- ③ 입력 아이템값 적용 후 다음버튼 클릭으로 다음 데이터를 요청합니다.
- ④ 입력아이템 선택시 하단 설명화면(⑤)에 간략한 내용 표시합니다.

입력아이템 값 적용

- 해당 아이템 항목에 클릭 후 적용할 데이터 입력(④)합니다.
- 해당 아이템 항목이 배열성인 경우, 배열입력 버튼(⑥)이 생성됩니다.
- 배열입력 버튼(⑥) 클릭 시 배열전용 입력 창 화면(⑦) 실행합니다.

배열위치	입력값
0 번째 데이터	
1 번째 데이터	
2 번째 데이터	
3 번째 데이터	
4 번째 데이터	

## 6.3.5 TR 데이터 응답처리



출력 창에서는 응답 상태에 대한 정보(TR 이름, 데이터 요약)을 표시합니다

해당 데이터를 클릭(①) 하면 상세한 내용을 표시합니다.

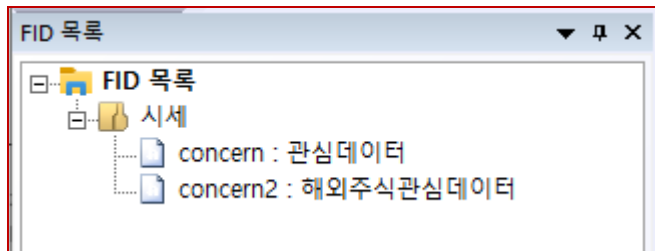
선택된 내용을 상세히 제공합니다.

출력 데이터 상세

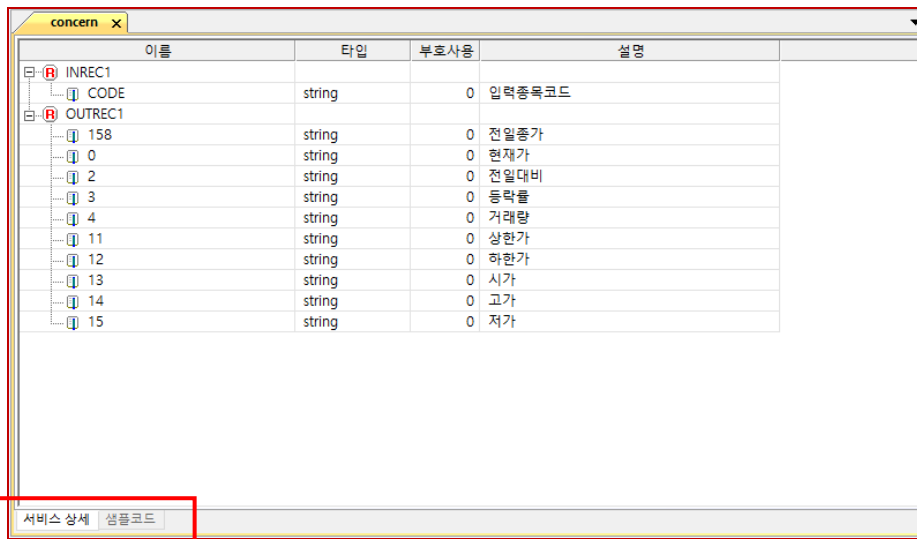
TR Name : stscur Record Name : OUTREC1

No	FILLER	SNAME	LTIME	LCPRICE	LFACEPRICE	LDIFF	CPCHECK	LLSTC
0	1	삼성전자	91224	59200	100	-200	5	59400

### 6.3.6 FID 데이터 로딩 화면

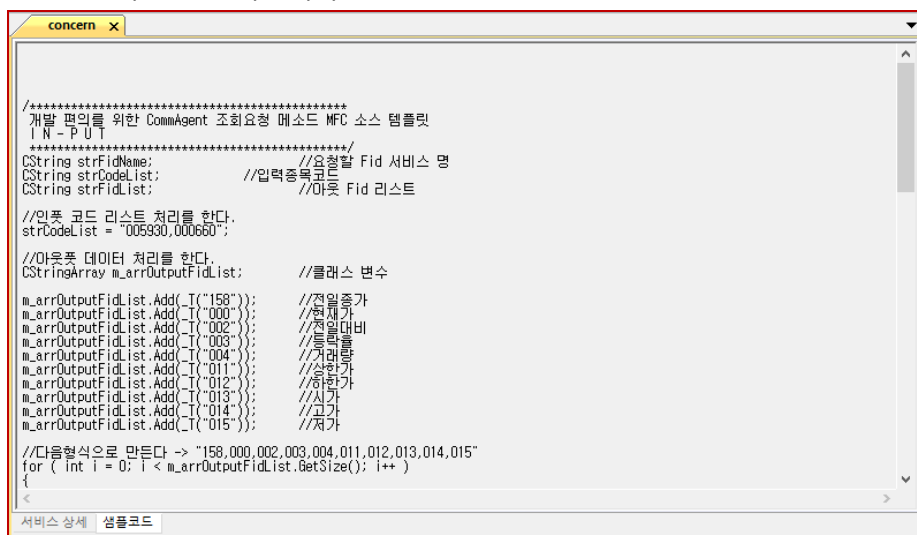


리스트 목록 중 처리할 항목을 더블클릭 하면, 데이터가 로드됩니다.



리소스 정보는 I/O에 해당하는 정보를 표시합니다.

하단 탭으로 정보화면을 변경할 수 있으며, 「샘플코드」 선택 시 MFC 개발용 샘플화면을 표시합니다.



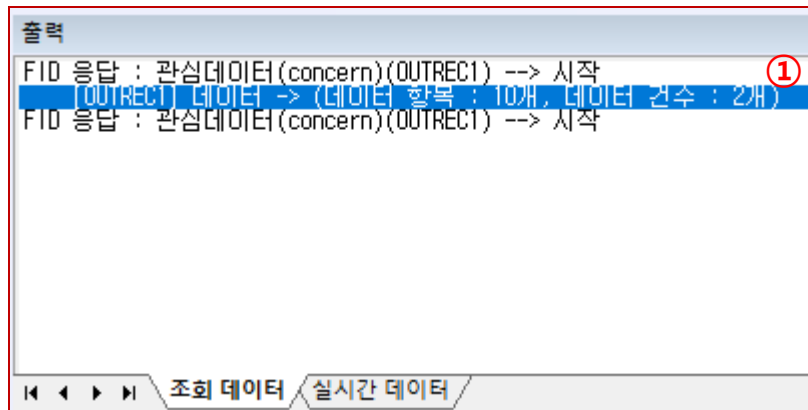
### 6.3.7 FID 데이터 입력 방법

- ① 로그인 중인 아이디 정보를 표시합니다.
- ② 입력 아이템값 적용 후 조회버튼 클릭으로 데이터를 요청합니다.
- ③ 입력아이템 선택시 하단 설명화면(④) 에 간략한 내용 표시합니다.

#### 입력아이템 값 적용

- 해당 아이템 항목에 클릭 후 적용할 데이터 입력(③)합니다.
- 해당 아이템 항목이 배열성인 경우, 배열입력 버튼(⑤)이 생성됩니다.
- 배열입력 버튼(⑤) 클릭 시 배열전용 입력 창 화면(⑥) 실행합니다.

## 6.3.8 FID 데이터 응답처리



출력 창에서는 응답 상태에 대한 정보(TR 이름, 데이터 요약)를 표시합니다.

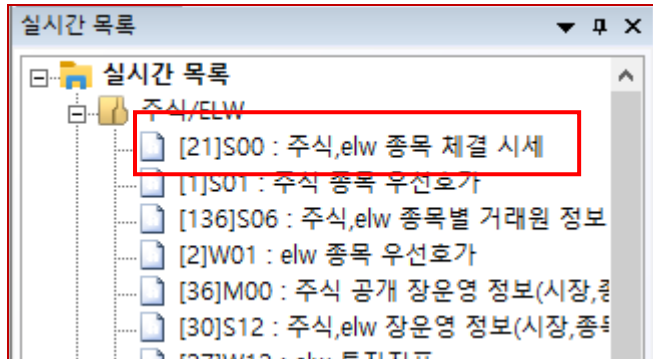
해당 데이터를 클릭(①) 하면 상세한 내용을 표시합니다.

선택된 내용을 상세히 제공합니다.

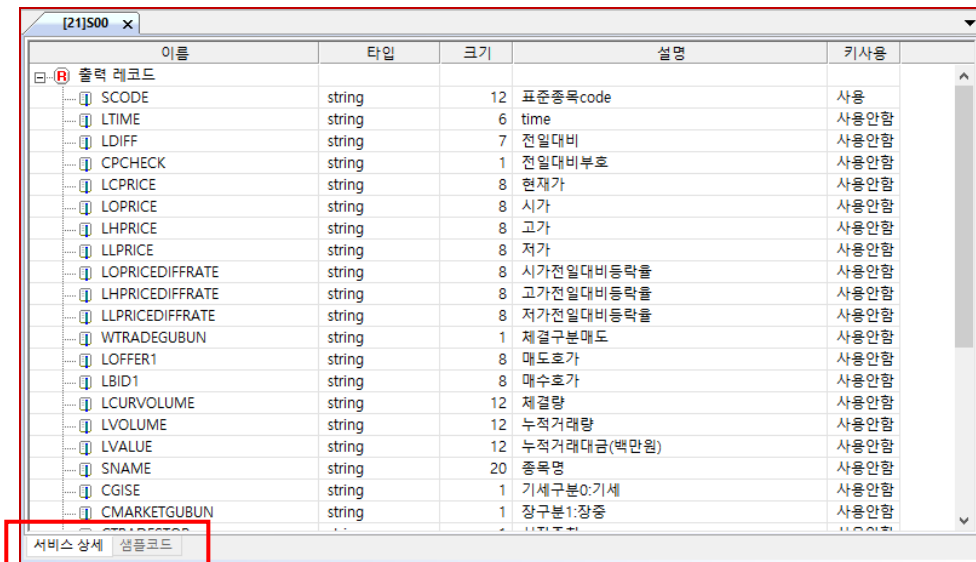
출력 데이터 상세							
FID Name : concern				Record Name : OUTREC1			
No	전일증가	현재가	전일대비	등락률	거래량	상한가	하한가
0	00093900	00092900	-0001000	-00106	000000759113	00122000	00065000
1	00059400	00059500	00000100	000017	000005961328	00077200	00041000



### 6.3.9 Real 데이터 로딩 화면



리스트 목록 중 처리할 항목을 더블클릭 하면, 데이터가 로드됩니다.



리소스 정보는 I/O에 해당하는 정보를 표시합니다.

하단탭으로 정보화면을 변경할 수 있으며, 「샘플코드」 선택 시 MFC 개발용 샘플화면을 표시합니다.



### 6.3.10 Real 데이터 입력 방법

- ① 로그인 중인 아이디 정보를 표시합니다.
- ② 자동갱신 키 코드값을 입력합니다.
- ③ 입력된 자동갱신 키 값으로 등록처리(③) 합니다.
  - 등록 후 데이터 발생 시 응답이 자동처리됩니다.
  - 해제시 까지 유지되며 최대 200 건의 데이터를 저장합니다.
- ④ 입력된 키 코드값으로 자동갱신 해제 처리합니다.

### 6.3.11 Real 데이터 응답 처리

출력 창에서는 응답 상태에 대한 정보(자동갱신 이름, 데이터 요약)를 표시합니다.

해당 데이터를 클릭(①)하면 상세한 내용을 표시합니다.

선택된 데이터 내용을 상세히 제공합니다.

- 개정 내역 -

[illegible]