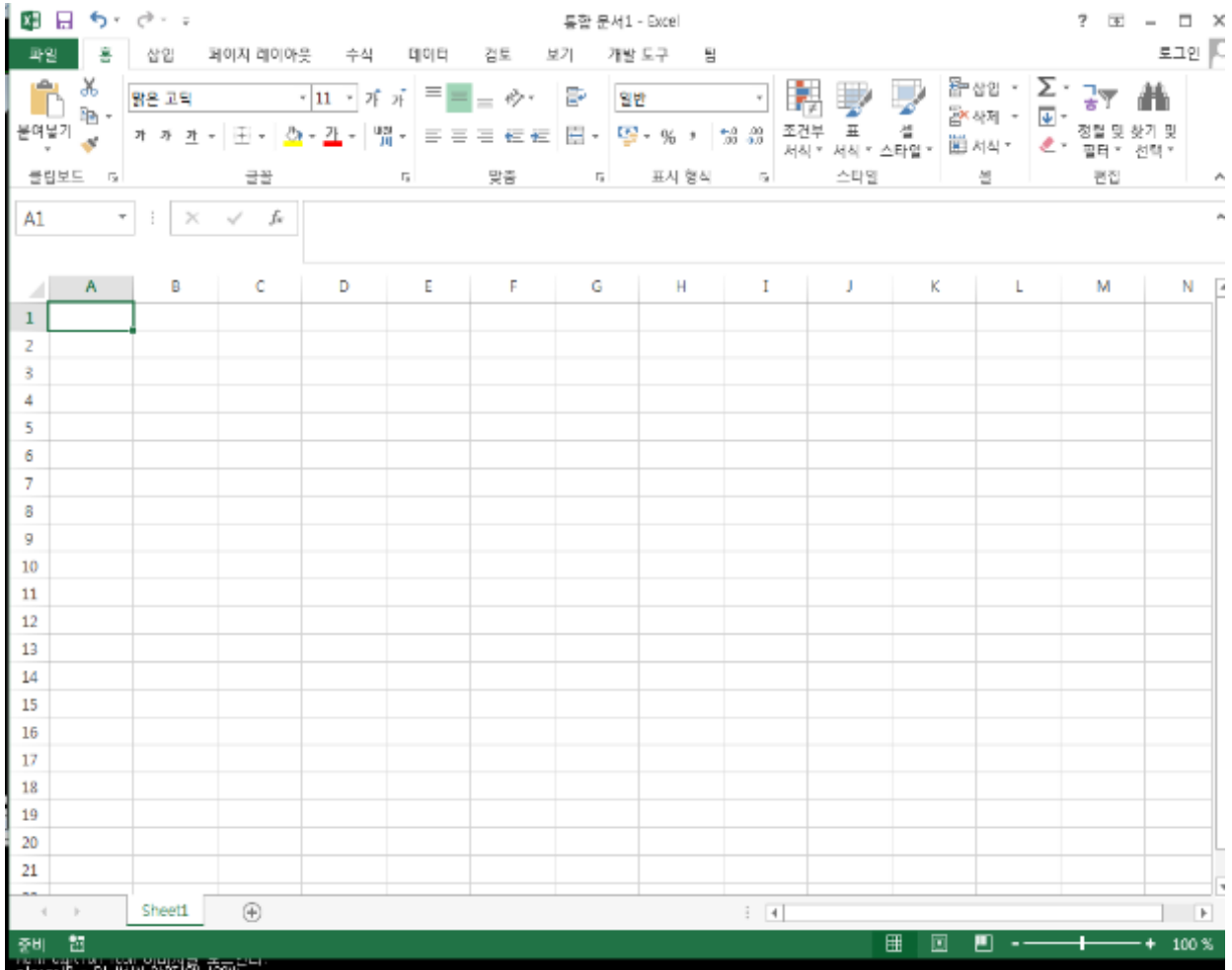


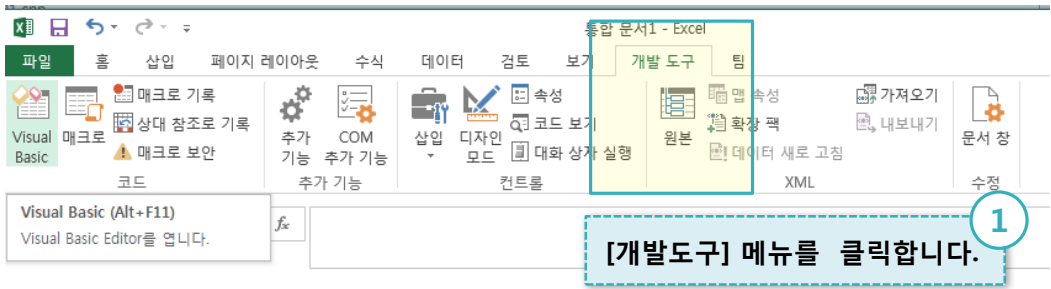
# 프로그램 개발 가이드

**Step1 ~ Step9**까지는 xingAPI COM을 이용한 엑셀 VBA 프로그램 개발을 하기 전에 필요한 준비 과정에 대한 내용입니다.

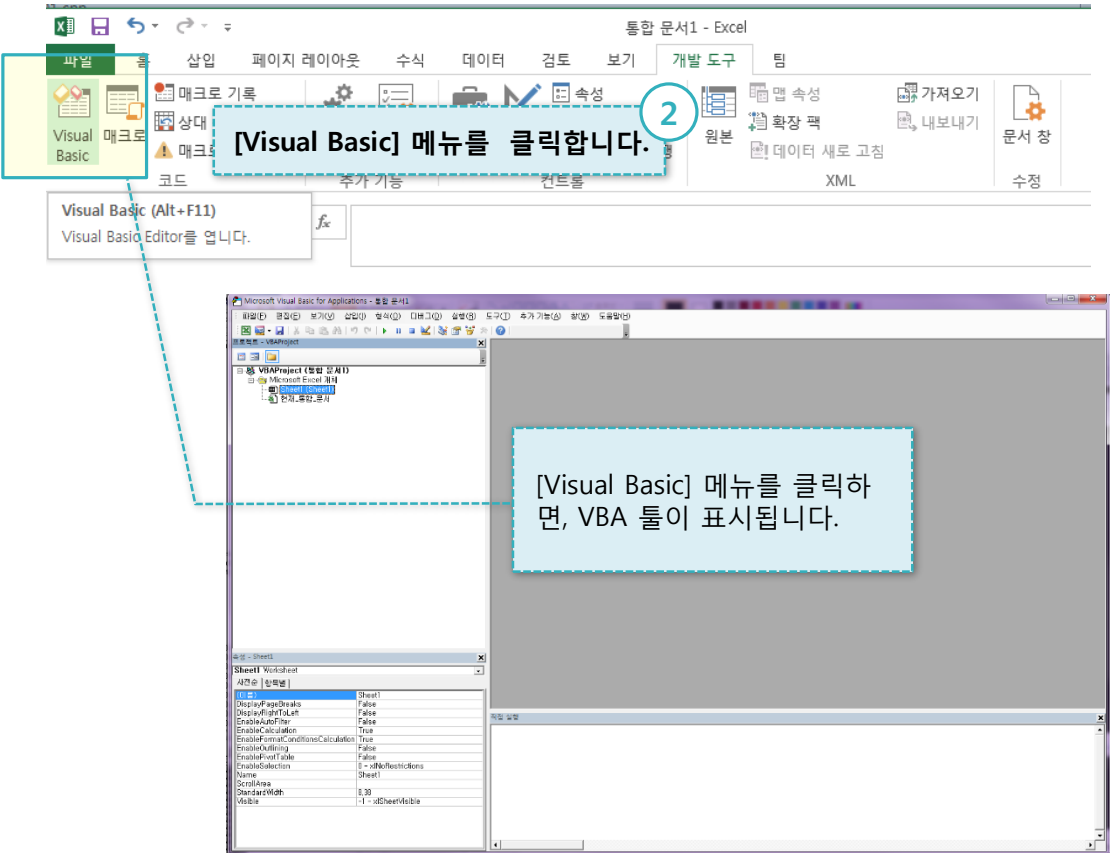
**Step 1** 엑셀을 열어 빈 엑셀 시트를 하나 생성합니다.



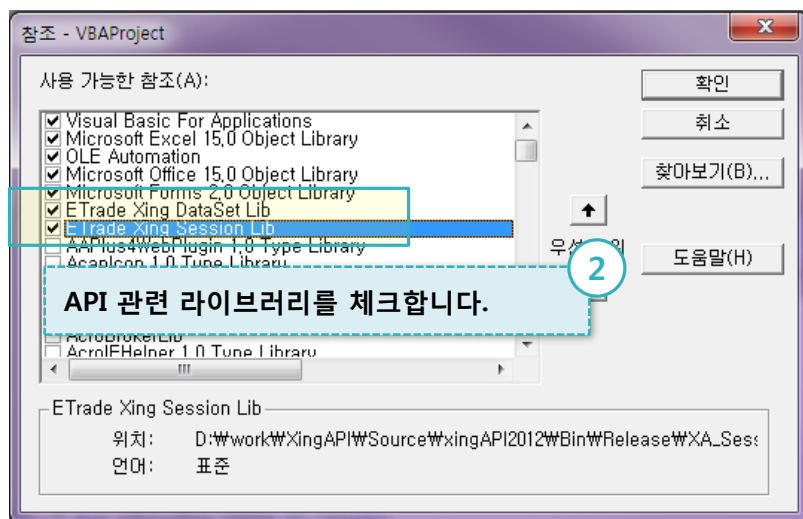
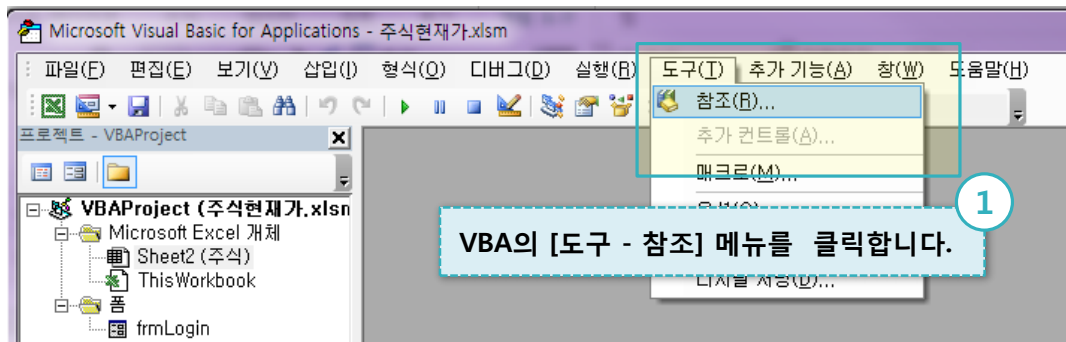
**Step 2** 엑셀 시트 메뉴의 [개발도구-Visual Basic] 을 눌러 VBA 툴을 표시합니다.



개발도구 메뉴가 보이지 않을 경우, [옵션 - 리본 사용자 지정] 창에서 표시체크를 하시면 됩니다.  
엑셀의 버전에 따라 개발도구 메뉴의 위치가 다를 수 있으니, 네이버 등을 이용하여 검색하시기 바랍니다.



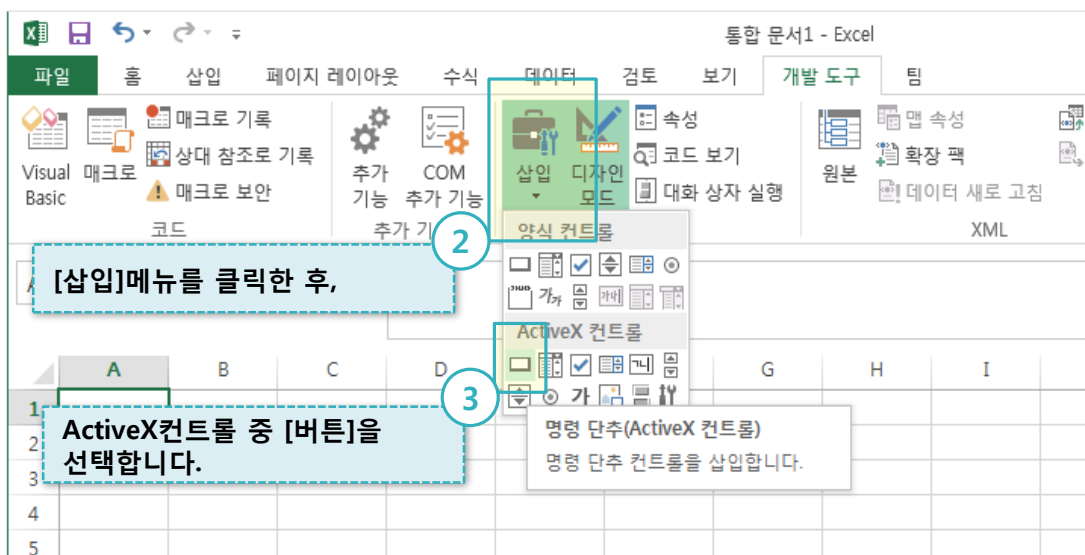
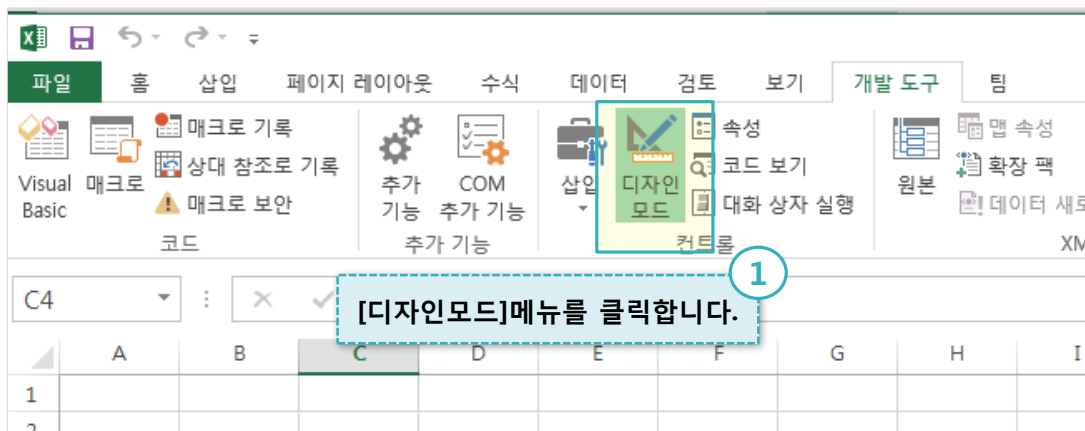
**Step 3** VBA의 [도구 - 참조] 메뉴를 눌러 표시된 '참조' 창에서 API 관련 라이브러리를 체크 표시합니다.



※ xingAPI COM 버전을 사용하려면 개발 플랫폼에 반드시 다음 2개의 xingAPI 라이브러리를 체크 하셔야 합니다.

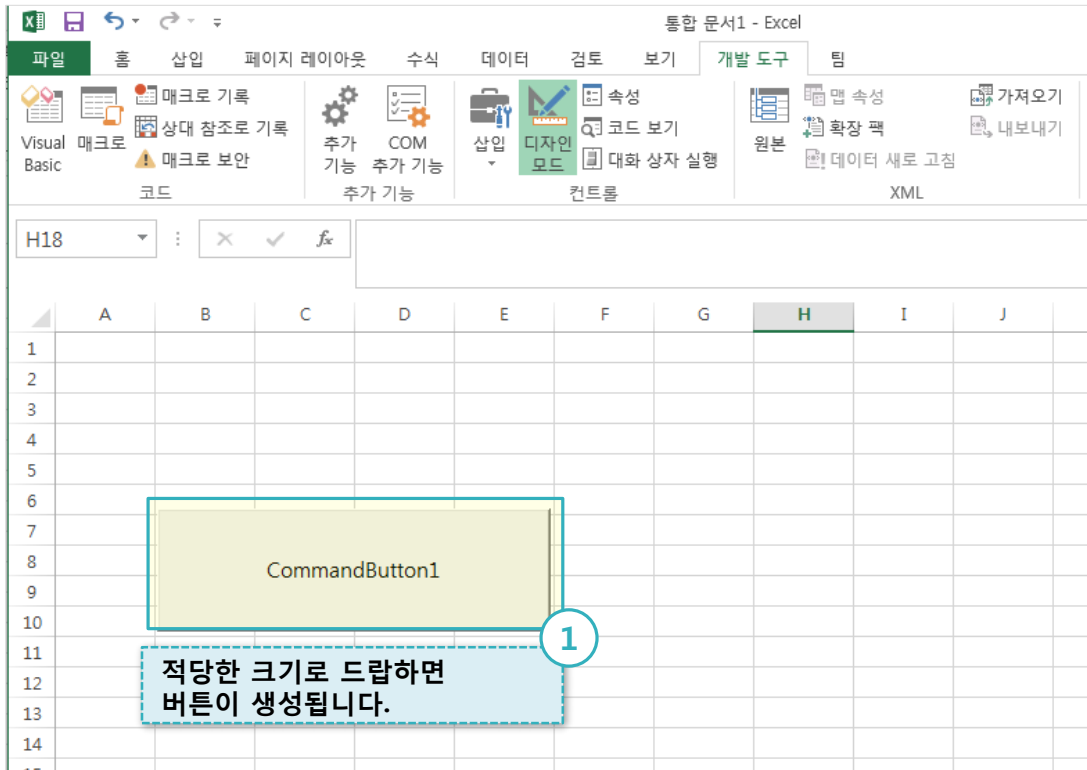
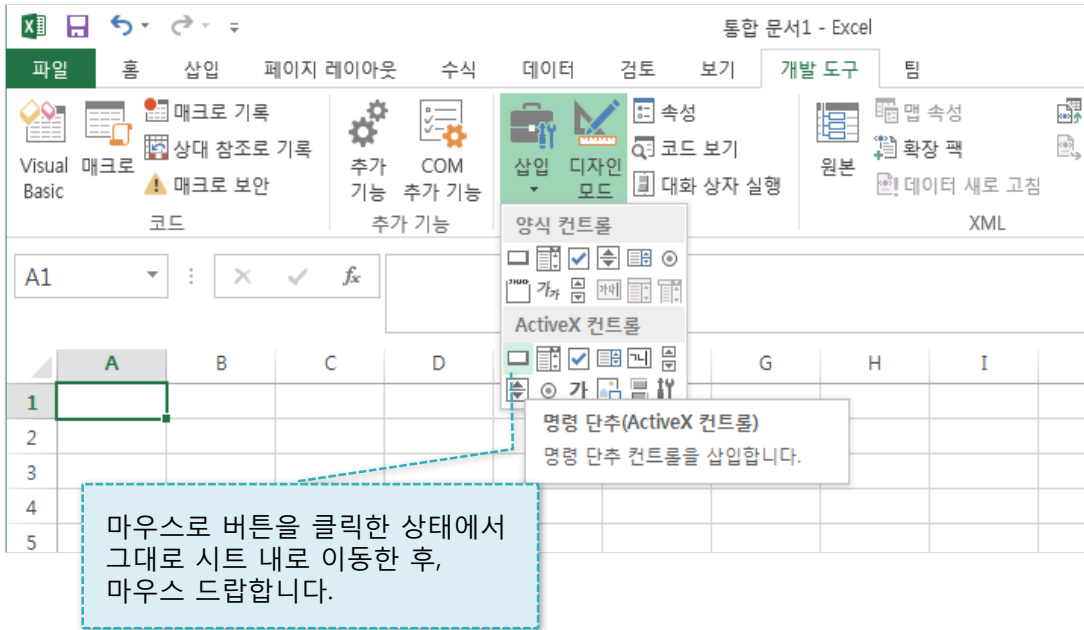
- Etrade Xing DataSet Lib
- Etrade Xing Session Lib

**Step 4** VBA의 엑셀 시트 메뉴의 [개발도구 - 디자인모드]를 누른 상태에서, [삽입] 메뉴내의 ActiveX 컨트롤 중 '버튼'을 선택합니다.



디자인 모드를 눌러야만 컨트롤 추가가 가능합니다.

**Step 5** VBA의 'Step4 - 3번'에서 버튼을 선택한 상태로 마우스를 시트 내로 이동한 후, 마우스 드래그 앤 드랍하여 버튼을 생성합니다.



**Step 6** VBA의 버튼을 선택하고, 마우스우측 버튼을 클릭하여 표시되는 팝업메뉴의 '속성'을 선택합니다.

Comman... : X ✓ fx =EMBED("Forms.CommandButton.1"; "")

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

CommandButton1

1

버튼을 선택하고  
마우스 우측버튼을 클릭하면  
표시되는 팝업메뉴에서  
'속성' 을 클릭합니다.

잘라내기(I)

복사(C)

붙여넣기(P)

속성(I)

코드 보기(V)

명령 단추 개체(O) >

그룹화(G)

순서(R)

컨트롤 서...

속성

CommandButton1 CommandButton

사전순 | 항목별

(이름) CommandButton1

Accelerator False

AutoLoad False

AutoSize False

BackColor &H8000000F&

BackStyle 1 - fmBackStyleOpaque

Caption CommandButton1

Enabled True

Font 맑은 고딕

ForeColor &H80000012&

Height 66,75

Left 55,5

Locked True

MouseIcon (없음)

Default

nAboveCente

TakeFocusOnClick True

Top 96

Visible True

Width 213

WordWrap False

속성을 클릭하면,  
버튼의 속성창이 표시됩니다.

**Step 7** 'Step6'의 속성 창에서 (이름)과 Caption을 변경합니다.

속성 창에서 CommandButton1의 속성을 변경합니다.

1. (이름)을 btnLogin으로 변경합니다. (버튼의 고유이름을 부여하는데, 로그인 처리를 할 것이므로 'btnLogin'으로 입력합니다.)

2. Caption을 로그인으로 변경합니다. (버튼의 표시이름을 부여하는데, 로그인 처리를 할 것이므로 '로그인'으로 입력합니다.)

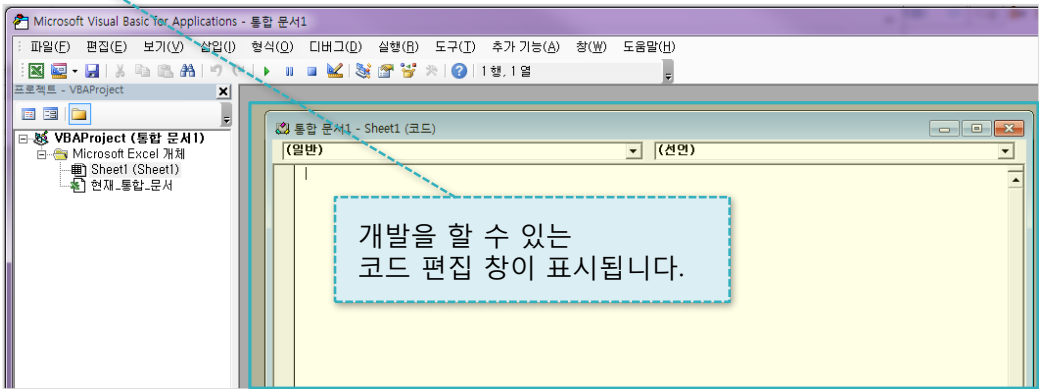
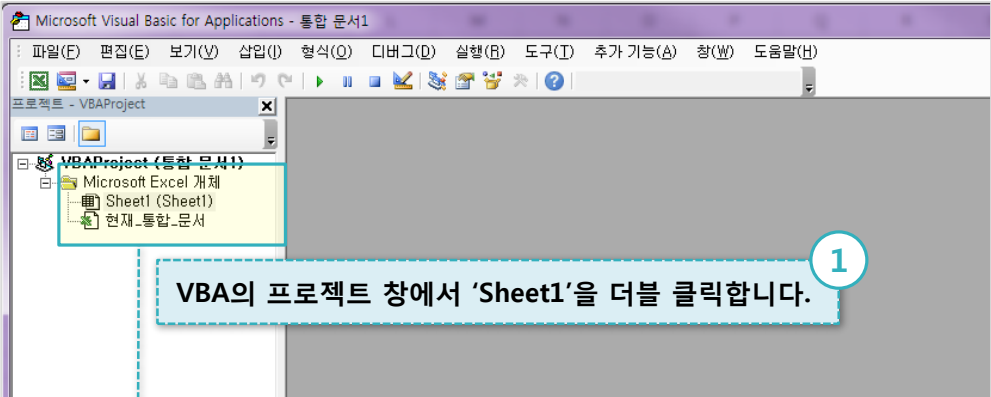
**Step 8** 'Step 7'에서 변경한 대로 '로그인' 버튼이 표시되는지 확인합니다.

Excel VBA 편집기에서 btnLogin 버튼을 확인합니다.

버튼명과 표시명이 정상으로 표시되는지 확인합니다.



Step 9 VBA창의 Sheet1을 엽니다. (더블 클릭)

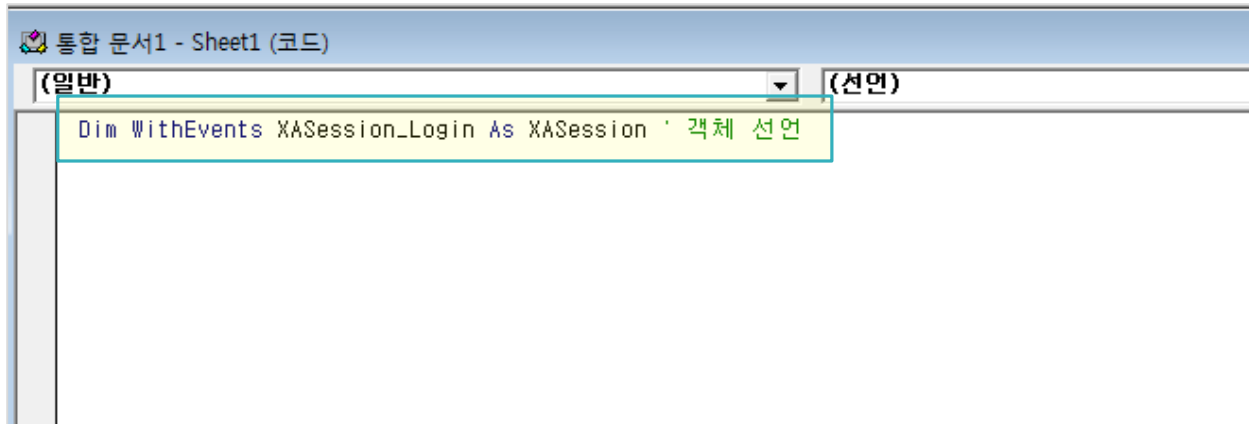


**Step10** 부터는 접속 및 로그인에 관한 개발 가이드입니다.

(메소드에 대한 상세한 설명은 COM 가이드를 참고하세요)

**Step 10** 로그인에 필요한 XASession 객체를 선언합니다.

※ VBA 문법입니다. 그냥 따라해 보세요.



**Step 11** 로그인 버튼을 클릭 할 때, 발생하는 클릭이벤트를 작성합니다.

통합 문서1 - Sheet1 (코드)

(일반) (선언)

btnLogin

Worksheet

1

객체 콤보 내의 'btnLogin' 객체를 선택  
(Step7~Step8 에서 본 sheet1의 로그인 버튼의 이름)

2

'btnLoin' 객체의 이벤트 콤보 내 리스트 중  
'Click' 이벤트를 선택하면

통합 문서1 - Sheet1 (코드)

btnLogin

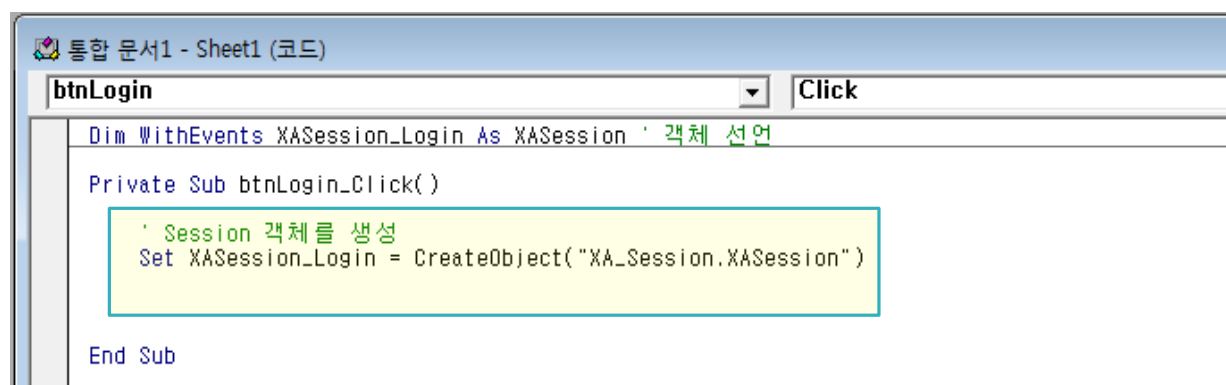
Dim WithEvents XASession\_Login As XASession ' 객체 선언

Private Sub btnLogin\_Click()  
End Sub

3

자동으로 이벤트 맵이 추가됩니다.

Click  
BeforeDropOrPaste  
Click  
DoubleClick  
Error  
GotFocus  
KeyDown  
KeyPress  
KeyUp  
LostFocus  
MouseDown  
MouseMove  
MouseUp

**Step 12** XASession 객체를 생성합니다.

## Step 13 서버에 접속을 합니다.

ConnectServer 함수의 결과로 접속 성공 여부를 판단하며, 접속 실패인 경우에만 메시지 박스를 표시하는 처리를 작성합니다.

### BOOL ConnectServer( BSTR szServerIP, LONG nServerPort )

szServerIP : 접속주소

실서버 : hts.ebestsec.co.kr

모의투자 : demo.ebestsec.co.kr

nServerPort : 포트 - 20001 고정

반환값 : 접속에 대한 성공 여부이며, 성공 시 TRUE

```

통합 문서1 - Sheet1 (코드)
btnLogin Click
Dim WithEvents XASession_Login As XASession ' 객체 선언

Private Sub btnLogin_Click()
    ' Session 객체를 생성
    Set XASession_Login = CreateObject("XA_Session.XASession")

    ' 모의투자 서버 접속
    If XASession_Login.ConnectServer( "demo.ebestsec.co.kr", 0) = False Then
        MsgBox "서버연결 실패"
    End If
End Sub
    
```

## Step 14 로그인을 요청합니다.

모의투자 접속 시 공인인증 비밀번호는 필요없습니다.

Login() 메소드의 결과는 API에서 서버로의 로그인 요청에 대한 성공 여부이며,

'로그인의 성공/실패' 는 로그인 이벤트 맵에서 확인합니다. (Step15)

**BOOL Login( BSTR szID, BSTR szPwd, BSTR szCertPwd,  
LONG nServerType, BOOL bShowCertErrDlg )**

szID : 아이디  
szPwd : 로그인 비밀번호  
szCertPwd : 공인인증 비밀번호  
nServerType : 사용안함  
bShowCertErrDlg : 공인인증 과정에서 발생한 에러에 대해 미리 정의된 Dialog를 표시할지 여부

반환값 : API -> 서버로의 로그인 요청에 대한 성공 여부이며, 성공 시 TRUE

통합 문서1 - Sheet1 (코드)

btnLogin Click

```
Dim WithEvents XASession_Login As XASession ' 객체 선언

Private Sub btnLogin_Click()

    ' Session 객체를 생성
    Set XASession_Login = CreateObject("XA_Session.XASession")

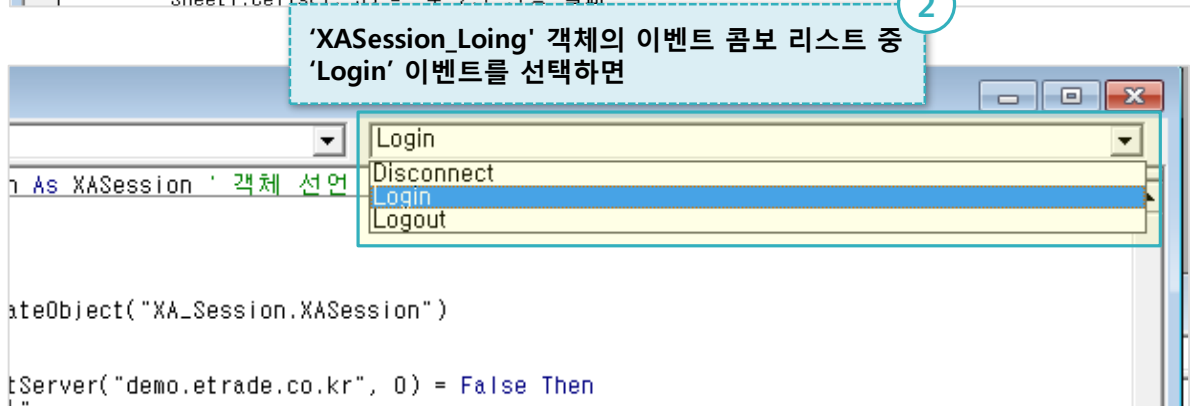
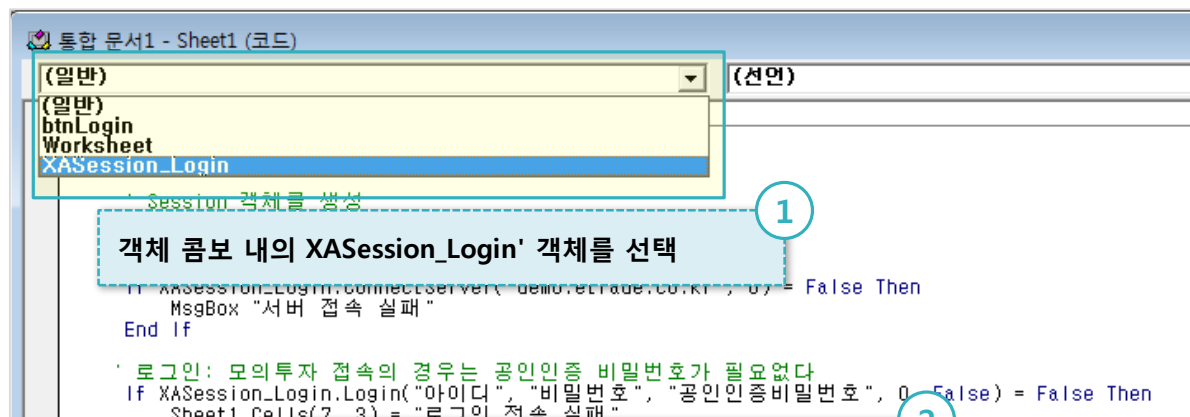
    ' 모의투자 서버 접속
    If XASession_Login.ConnectServer("demo.ebestsec.co.kr" 0) = False Then
        MsgBox "서버 접속 실패"
    End If

    ' 로그인: 모의투자 접속의 경우는 공인인증 비밀번호가 필요없다
    If XASession_Login.Login("아이디", "비밀번호", "공인인증비밀번호", 0, False) = False Then
        Sheet1.Cells(7, 3) = "로그인 전송 실패"
    End If

End Sub
```

**Step 15** 로그인 이벤트를 받습니다. (로그인 버튼 클릭 이벤트와 동일하게 처리)

'Step14'의 로그인 요청이 성공적으로 서버에 전송되면, 로그인의 성공여부에 대한 결과값은 Login 이벤트로 들어옵니다.



3

자동으로 이벤트 맵이 추가됩니다.

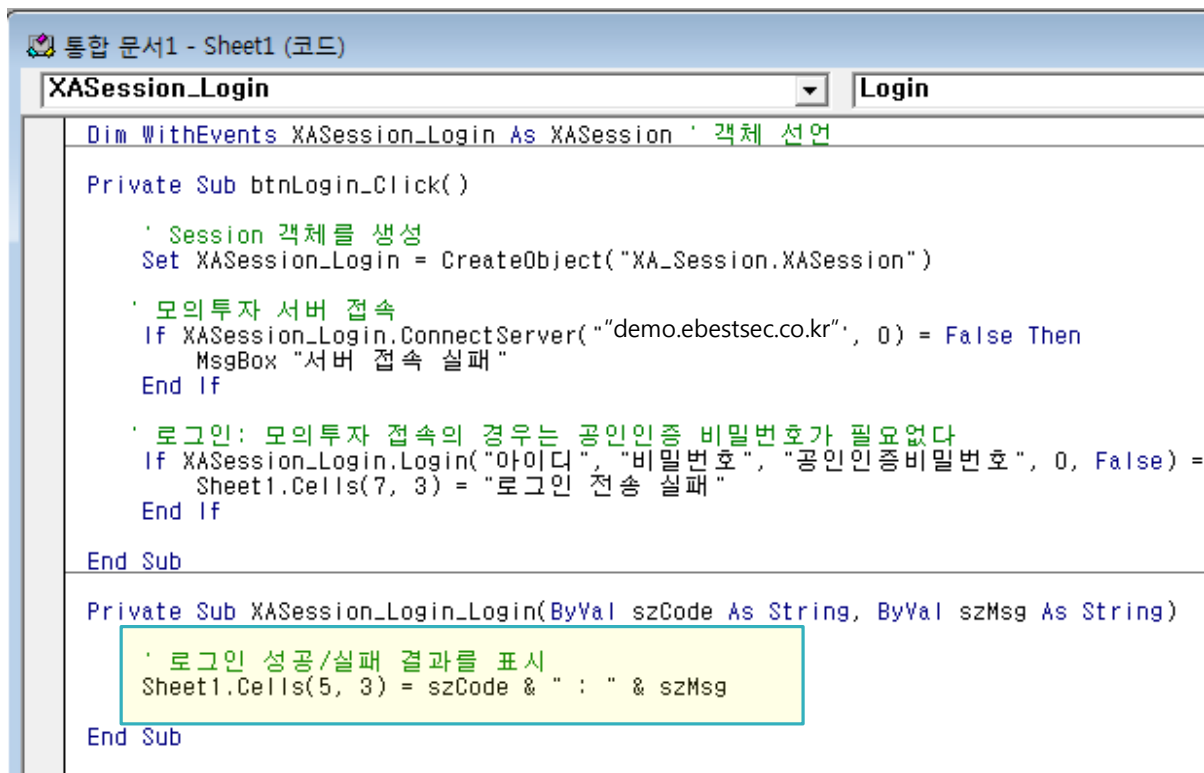
## Step 16 'Step15 - 3번'에서 로그인 성공/실패 결과를 확인합니다.

ex) 로그인 성공/실패 결과를 Sheet1의 Cell(5, 3)에 표시

```
Private Sub XASession_Login_Login( ByVal szCode As String, ByVal szMsg As String )

    Sheet1.Cells(5, 3) = szCode & " : " & szMsg

End Sub
```



The screenshot shows the VBA editor for '통합 문서1 - Sheet1 (코드)'. The 'XASession\_Login' module is selected, and the 'Login' subprocedure is highlighted. The code includes comments in Korean and VBA syntax for session management and login logic.

```
Dim WithEvents XASession_Login As XASession ' 객체 선언

Private Sub btnLogin_Click()

    ' Session 객체를 생성
    Set XASession_Login = CreateObject("XA_Session.XASession")

    ' 모의투자 서버 접속
    If XASession_Login.ConnectServer("demo.ebestsec.co.kr", 0) = False Then
        MsgBox "서버 접속 실패"
    End If

    ' 로그인: 모의투자 접속의 경우는 공인인증 비밀번호가 필요없다
    If XASession_Login.Login("아이디", "비밀번호", "공인인증비밀번호", 0, False) =
        Sheet1.Cells(7, 3) = "로그인 전송 실패"
    End If

End Sub

Private Sub XASession_Login_Login(ByVal szCode As String, ByVal szMsg As String)

    ' 로그인 성공/실패 결과를 표시
    Sheet1.Cells(5, 3) = szCode & " : " & szMsg

End Sub
```

※ szCode에 "0000"이 들어올 경우는 로그인 성공이고 그 이외의 값은 로그인 실패입니다.



## > 전체 코드

```
Dim WithEvents XASession_Login As XASession ' 객체 선언

'-----
' 로그인 버튼 클릭 처리
'-----

Private Sub btnLogin_Click()
    'XASession 객체를 생성
    Set XASession_Login = CreateObject("XA_Session.XASession")

    ' 모의투자 서버 접속
    If XASession_Login.ConnectServer("demo.ebestsec.co.kr", 0) = False Then
        MsgBox "서버 접속 실패"
    End If

    ' 로그인: 모의투자 접속의 경우는 공인인증 비밀번호가 필요없다
    ' 아이디, 비밀번호는 본인의 것을 입력
    If XASession_Login.Login("아이디", "비밀번호", "공인인증비밀번호", 0, False) = False Then
        Sheet1.Cells(7, 3) = "로그인 전송 실패"
    End If
End Sub

'-----
' 로그인 이벤트 수신 처리
'-----

Private Sub XASession_Login_Login(ByVal szCode As String, ByVal szMsg As String)
    ' 로그인 성공/실패 결과를 표시
    Sheet1.Cells(5, 3) = szCode & " : " & szMsg
End Sub
```

접속 및 로그인 코드가 완성되었습니다.

Step5 에서의 **[개발도구 - 디자인모드]**선택을 끄고, 로그인 버튼을 클릭해 보세요.

**데이터 조회**는 미리 정해진 형식의 입력값을 서버로 전송하고 서버로부터 데이터를 받는 것을 의미합니다. 한 번의 전송으로 한 번의 결과값을 받습니다.

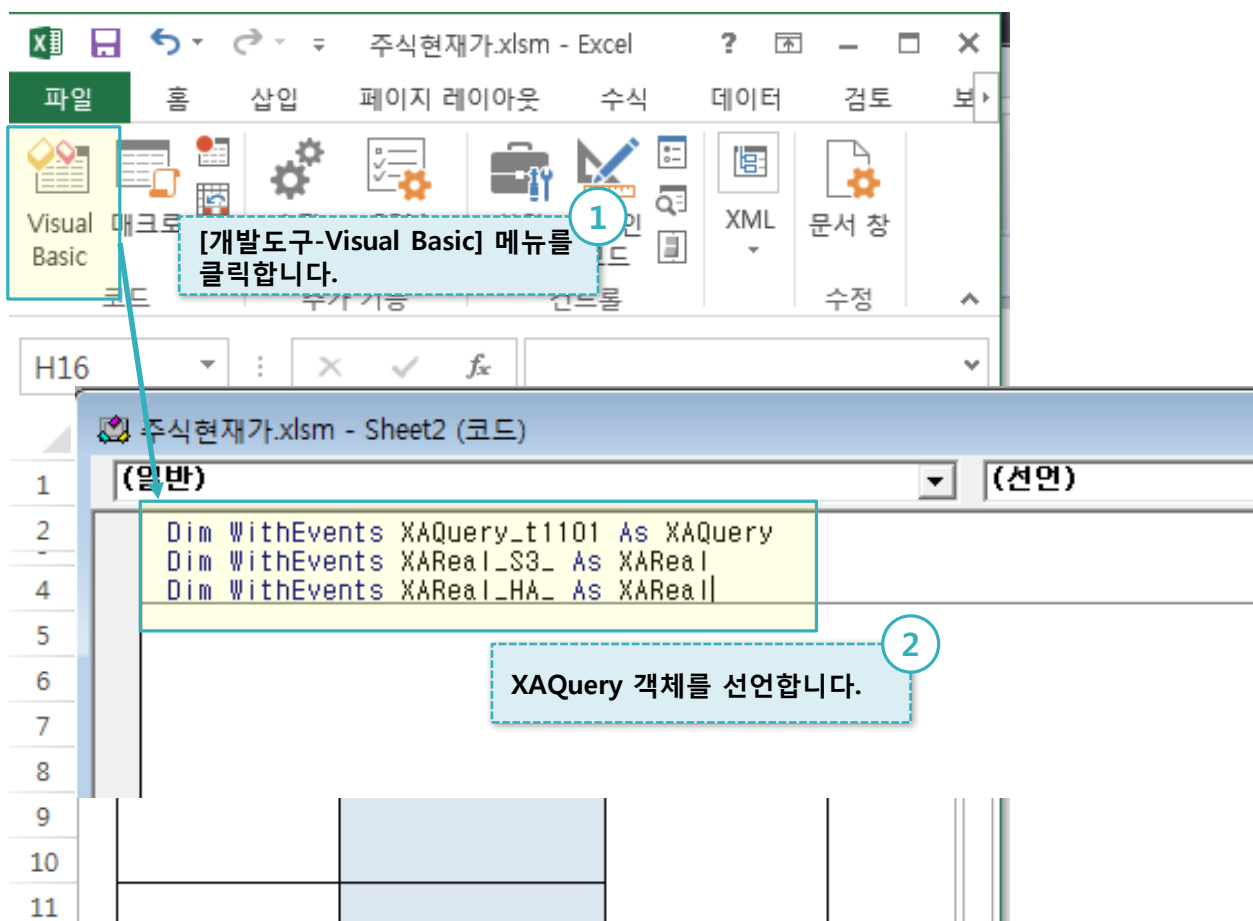
데이터 조회를 하기 위해서는 **XAQuery객체**를 사용하며, 객체 하나당 하나의 TR코드를 등록할 수 있습니다.

#### 아래의 코드는

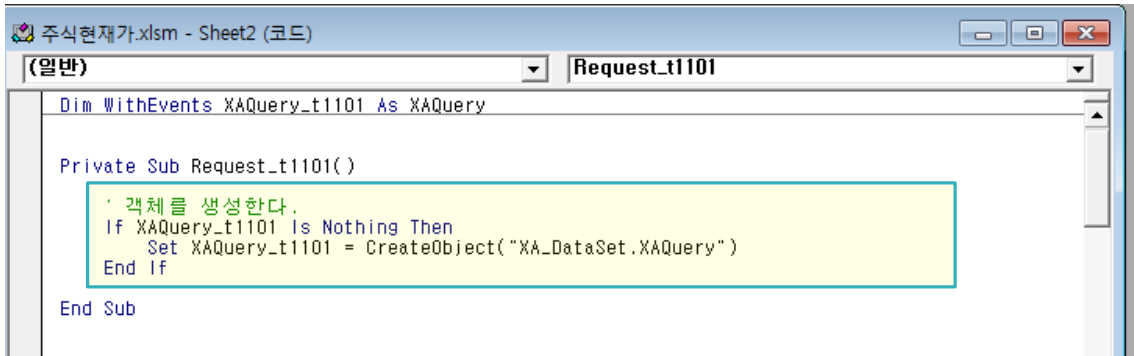
t1101 (주식 현재가 호가 조회) TR을 이용해 현재가를 조회하는 예제입니다.

이것은 xingAPI 설치 폴더 내 "엑셀샘플W주식현재가.xlsm" 파일에서도 확인 가능합니다.

#### Step 1 XAQuery 객체를 선언합니다. (접속 및 로그인의 XASession 선언과 동일)



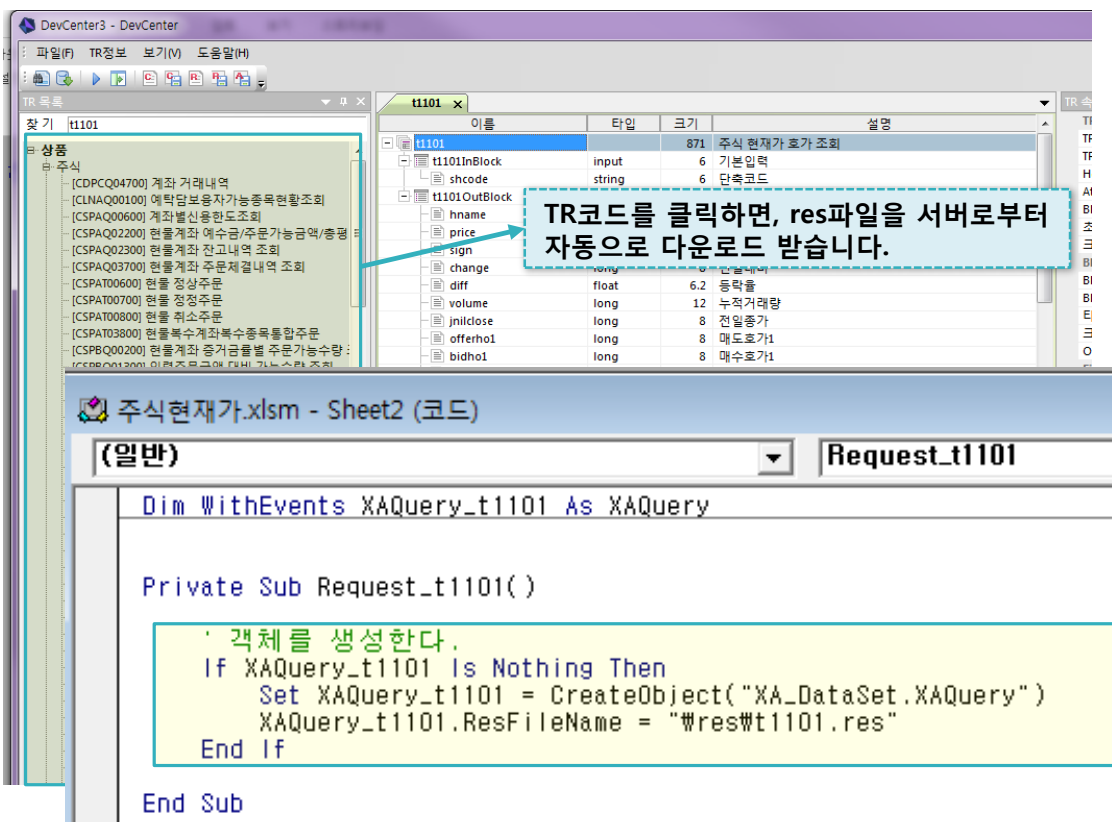
## Step 2 XAQuery 객체를 생성합니다. (접속 및 로그인의 XASession 생성과 동일)



**Step 3** Res 파일을 등록합니다.

- ① 데이터 조회를 원하는 TR코드의 RES 파일을 [DevCenter – TR목록 창]에서 다운로드 받습니다.  
- TR코드(TR명)를 클릭하면, 클릭하는 시점에 서버에서 자동으로 다운로드
- ② ResFileName 메소드를 이용하여 XAQuery 객체에 등록합니다.

res파일은 xingAPI COM버전에서 사용하는 것으로 TR의 입출력 정보를 정의하며,  
res파일이 없으면 COM버전을 이용할 수 없으므로 반드시 미리 다운받아야 합니다.



**Step 4** 조회를 요청하기 전에 조회 TR에 입력 데이터 값을 설정합니다.

서버에 조회 TR을 요청하기 전에, Inblock의 각 필드 입력값을 모두 입력해 주어야 합니다.

**void SetFieldData (BSTR szBlockName, BSTR szFieldName, LONG nOccursIndex, BSTR szData)**

szBlockName : 블록명

szFieldName : 필드명

nOccursIndex : '0' 고정

szData : 입력 데이터 (값)

이름	타입	크기	설명
t1101		871	주식 현재가 호가 조회
t1101InBlock	input	6	기본입력
shcode	string	6	단축코드
t1101OutBlock	output	865	출력
hname	string	20	한글명
price	long	8	현재가
sign	string	1	전일대비구분
change	long	8	전일대비

```

Dim WithEvents XAQuery_t1101 As XAQuery

Private Sub Request_t1101()
    ' 객체를 생성한다.
    If XAQuery_t1101 Is Nothing Then
        Set XAQuery_t1101 = CreateObject("XAQuery.XAQuery")
        XAQuery_t1101.ResFileName = "#res#wt1101.res"
    End If

    ' 데이터 세팅
    Call XAQuery_t1101.SetFieldData("t1101InBlock", "shcode", 0, "078020")
End Sub
    
```

## Step 6 서버로부터 요청한 조회 데이터를 수신 받습니다.

- ① 데이터 입력 후 조회 요청한 TR에 대해서 서버로부터 데이터를 수신했을 때 이벤트가 발생합니다.
- ② 이벤트는 XAQuery 객체의 ReceiveData 이벤트를 사용합니다.

조회 요청에 대한 서버로부터의

조회 결과 성공여부는 *ReceiveMessage* 이벤트로

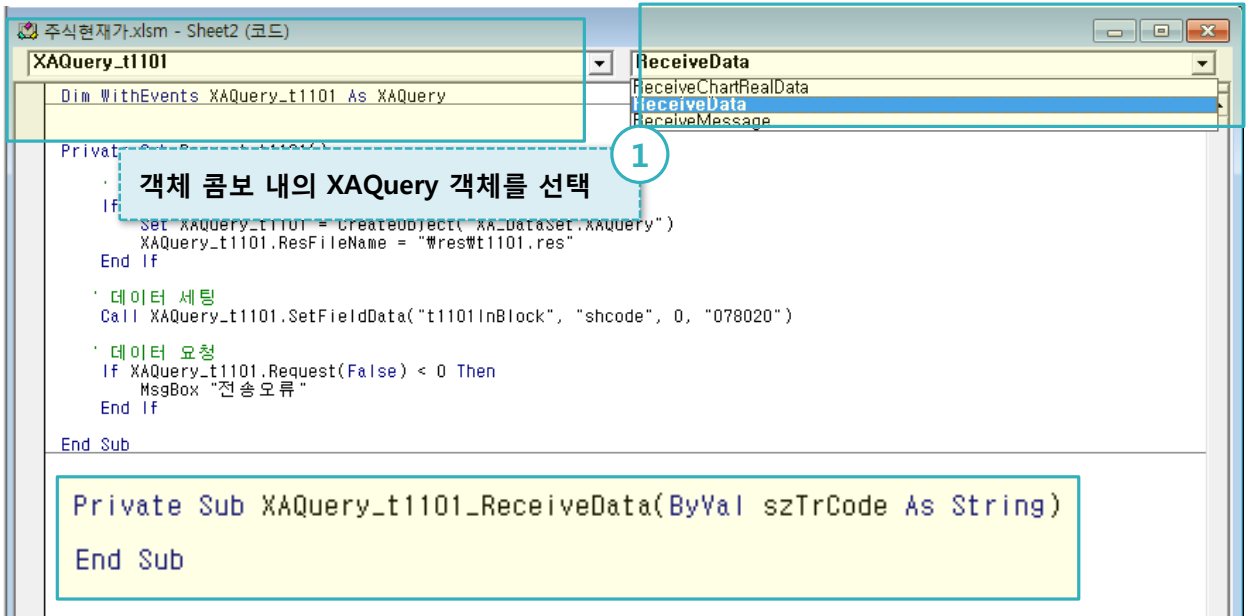
조회 결과 성공시, 결과 데이터는 *ReceiveData* 이벤트로

확인합니다.

**void ReceiveData ( BSTR szTrCode )**

szTrCode : TR코드

XAQuery 객체의 이벤트 콤보리스트 중  
'ReceiveData' 이벤트를 선택하면



자동으로 이벤트 맵이 추가됩니다.

**Step 7** 이벤트를 받았다면 GetFieldData 메소드를 사용하여 수신 데이터를 가져옵니다 .

**void BSTR GetFieldData (BSTR szBlockName, BSTR szFieldName, LONG nOccursIndex)**

szBlockName : 블록명

szFieldName : 필드명

nOccursIndex : 블록이 반복데이터일 경우, index정보

반환값 : 블록의 필드 값

```

주식현재가.xlsm - Sheet2 (코드)
Worksheet
Change

Dim WithEvents XAQuery_t1101 As XAQuery

Private Sub Request_t1101()
    ' 객체를 생성한다.
    If XAQuery_t1101 Is Nothing Then
        Set XAQuery_t1101 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_t1101.ResFileName = "WresWt1101.res"
    End If

    ' 데이터 세팅
    Call XAQuery_t1101.SetFieldData("t1101InBlock", "shcode", 0, "078020")

    ' 데이터 요청
    If XAQuery_t1101.Request(False) < 0 Then
        MsgBox "전송 오류"
    End If

    Sheet2.Cells(2, 5) = XAQuery_t1101.GetFieldData("t1101OutBlock", "hname", 0)
    Sheet2.Cells(4, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "price", 0)
    Sheet2.Cells(4, 3) = GetSign(XAQuery_t1101.GetFieldData("t1101OutBlock", "sign", 0))
    Sheet2.Cells(4, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "change", 0)
    Sheet2.Cells(4, 5) = XAQuery_t1101.GetFieldData("t1101OutBlock", "diff", 0)
    Sheet2.Cells(4, 6) = XAQuery_t1101.GetFieldData("t1101OutBlock", "volume", 0)

    Sheet2.Cells(4, 7) = XAQuery_t1101.GetFieldData("t1101OutBlock", "diff", 0)
    Sheet2.Cells(4, 6) = XAQuery_t1101.GetFieldData("t1101OutBlock", "volume", 0)

End Sub

Private Function GetSign(ByVal sSign As String)
    If sSign = "1" Then

```

조회 데이터를 가져옵니다.

주식현재가.xlsm - Excel

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보

Visual Basic 매크로 추가 기능 COM 추가 기능 삽입 디자인 모드 XML 문서 창

코드 추가 기능 컨트롤 수정

G17

	A	B	C	D	E	F	G
1							
2		종 목	078020				이트레이드증권
3							
4		9,810 ▲	20	0.20			118
5		매도잔량					매수잔량
6							

▶ 전체 코드

```
Dim WithEvents XAQuery_t1101 As XAQuery

'-----
' 주식 현재가 조회
'-----

Private Sub Request_t1101()

    ' 객체를 생성한다.
    If XAQuery_t1101 Is Nothing Then
        Set XAQuery_t1101 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_t1101.ResFileName = "WresWt1101.res"
    End If

    ' 데이터 세팅
    Call XAQuery_t1101.SetFieldData("t1101InBlock", "shcode", 0, "078020")

    ' 데이터 요청
    If XAQuery_t1101.Request(False) < 0 Then
        MsgBox "전송오류"
    End If

End Sub

'-----
' 주식 현재가 조회 요청 성공 시, 서버로부터 데이터 결과 수신 이벤트
'-----

Private Sub XAQuery_t1101_ReceiveData(ByVal szTrCode As String)
    Sheet2.Cells(2, 5) = XAQuery_t1101.GetFieldData("t1101OutBlock", "hname", 0) ' 종목명
    Sheet2.Cells(4, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "price", 0) ' 현재가
    Sheet2.Cells(4, 3) = GetSign(XAQuery_t1101.GetFieldData("t1101OutBlock", "sign", 0)) ' 전일대비구분
    Sheet2.Cells(4, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "change", 0) ' 전일대비
    Sheet2.Cells(4, 5) = XAQuery_t1101.GetFieldData("t1101OutBlock", "diff", 0) ' 등락율
    Sheet2.Cells(4, 6) = XAQuery_t1101.GetFieldData("t1101OutBlock", "volume", 0) ' 누적거래량
End Sub
```

```
'-----
' 대비구분 값을 등락 부호로 변경
'-----
```

```
Private Function GetSign(ByVal sSign As String)
```

```
    If sSign = "1" Then
```

```
        GetSign = "↑"
```

```
    ElseIf sSign = "2" Then
```

```
        GetSign = "▲"
```

```
    ElseIf sSign = "4" Then
```

```
        GetSign = "↓"
```

```
    ElseIf sSign = "5" Then
```

```
        GetSign = "▼"
```

```
    End If
```

```
End Function
```

```
'-----
' 엑셀 시트에 변경 발생시 수신 이벤트
'-----
```

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
    ' 종목코드 입력
```

```
    If Target.Row = 2 And Target.Column = 3 Then
```

```
        Call Request_t1101
```

```
    End If
```

```
End Sub
```



주문은 조회 TR과 TR코드만 다를 뿐 처리하는 방식은 동일합니다.

(데이터 조회를 위한 XAQuery 객체 사용법과 동일)

즉, 주문 TR에 입력값을 모두 입력한 후 서버로 전송하면, 서버로부터 주문 성공여부에 대한 결과를 받습니다. 주문이 성공했다는 것은 HTS 주문화면에서의 주문 성공과 동일한 의미이며, 실시간 주문 체결 TR을 이용하여 실시간 주문 결과를 수신 받을수도 있습니다.

### 아래의 코드는

CSPAT00600(현물정상주문) TR을 이용해 주식 종목을 주문하는 예제입니다.

이것은 xingAPI 설치 폴더 내 “엑셀샘플W xingAPI활용교육샘플v1.2.xlsm” 파일에서도 확인 가능합니다.

## Step 1 XAQuery 객체를 선언합니다. (일반 데이터 조회 과정과 동일)

The screenshot shows the Excel application with the '개발도구-Visual Basic' menu highlighted. A callout box points to the 'Visual Basic' button in the ribbon, stating: "[개발도구-Visual Basic] 메뉴를 클릭합니다." (Click the [Developer-Visual Basic] menu).

The Excel spreadsheet displays a table with columns for '종목' (Stock), '종목명' (Stock Name), '잔고' (Balance), '평균단가' (Average Price), '현재가' (Current Price), '매입금액' (Purchase Amount), '평가금액' (Evaluation Amount), '평가손익' (Evaluation Profit/Loss), and '손익률' (Profit/Loss Rate). The table is divided into sections for '잔고조회' (Balance Inquiry) and '잔고다음조회' (Next Balance Inquiry).

The VBA editor shows the 'Microsoft Visual Basic for Applications - xingAPI활용교육샘플v1.2.xlsm' window. The 'VBAProject (xingAPI활용교육)' tree on the left shows the 'Sheet2 (코드)' (Code) sheet. The code in the 'Sheet2 (코드)' sheet is as follows:

```

Dim WithEvents XASession_Account As XASession
Dim WithEvents XAQuery_CSPAT00600 As XAQuery
Dim WithEvents XAQuery_CSPAT00700 As XAQuery
Dim WithEvents XAQuery_CSPAT00800 As XAQuery
Dim WithEvents XAReal_K3 As XAReal
Dim nOrderListNextRow As Integer
Dim nBalanceListNextRow As Integer

Private Sub cbAccount_Change()
    szTemp = cbAccount.Text
    Range("L32").Value = XASession_Account.GetAccountName(szTemp)

```

A callout box points to the code, stating: "일반 데이터 조회시의 객체 선언과 동일하며 TR코드만 다른 것입니다." (Same as object declaration for general data retrieval, only the TR code is different).

**Step 2** XAQuery 객체 생성 및 Res파일을 등록합니다. (일반 데이터 조회 과정과 동일)

Res파일이 없다고 오류가 발생할 때에는  
[DevCenter - TR목록 창]에서 TR코드를 클릭하면,  
클릭하는 시점에 서버에서 자동으로 다운로드 받습니다.

```

Private Sub btnOrder1_Click()
    Call Order(2)
End Sub

Private Sub Order(nOrderType)
    If XAQuery_CSPAT00600 Is Nothing Then
        Call Order2_Click()
    End If
    If XAQuery_CSPAT00600 Is Nothing Then
        Set XAQuery_CSPAT00600 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_CSPAT00600.ResFileName = "wres\CSPAT00600.res"
    End If
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "IsuNo", 0, Range("L5").Value)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdQty", 0, Range("L9").Value)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdPrc", 0, Range("L10").Value)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "BnsTpCode", 0, nOrderType)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdprcPtnCode", 0, Left(Range("L7").Value, 2))
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "MgntrnCode", 0, "000")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "LoanDt", 0, "")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdCndiTpCode", 0, Left(Range("L8").Value, 1))

```

**Step 3** 주문 요청 전에 입력 데이터 값을 설정합니다. ((일반 데이터 조회 과정과 동일))

서버에 주문 TR을 요청하기 전에, InBlock 데이터를 입력해 주어야 합니다.

AcntNo	string	20	계좌번호
InptPwd	string	8	입력비밀번호
IsuNo	string	12	종목번호
OrdQty	long	16	주문수량
OrdPrc	double	13.2	주문가
BnsTpCode	string	1	매매구분
OrdprcPtnCode	string	2	호가유형코드
MgntrnCode	string	3	신용거래코드
LoanDt	string	8	대출일
OrdCndiTpCode	string	1	주문조건구분

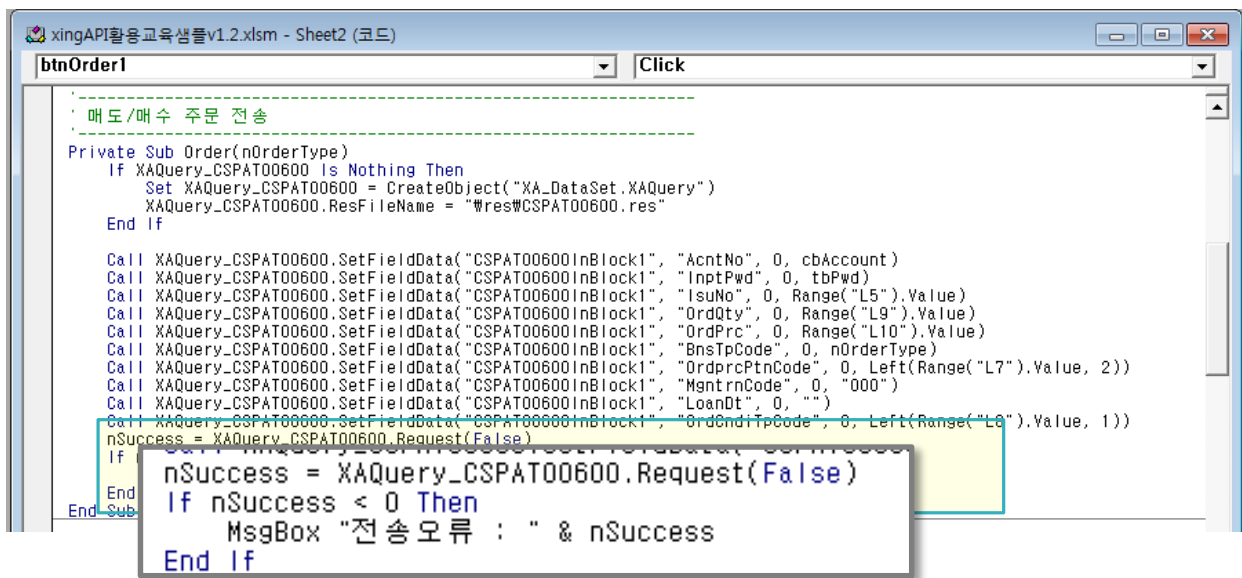
Inblock의 필드값을 모두 입력해야만  
주문이 가능합니다.

```

Private Sub Order(nOrderType)
    If XAQuery_CSPAT00600 Is Nothing Then
        Call Order2_Click()
    End If
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "AcntNo", 0, cbAccount)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "InptPwd", 0, tbPwd)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "IsuNo", 0, Range("L5").Value)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdQty", 0, Range("L9").Value)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdPrc", 0, Range("L10").Value)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "BnsTpCode", 0, nOrderType)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdprcPtnCode", 0, Left(Range("L7").Value, 2))
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "MgntrnCode", 0, "000")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "LoanDt", 0, "")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdCndiTpCode", 0, Left(Range("L8").Value, 1))

```

**Step 4** 서버에 주문TR을 요청합니다. (일반 데이터 조회 과정과 동일)



**Step 5** 서버로부터 주문 요청한 결과 데이터를 수신받습니다. (일반 데이터 조회 과정과 동일)

- ① 서버로부터 주문 결과 데이터를 수신했을 때 이벤트가 발생합니다.
- ② 이벤트는 XAQuery 객체의 ReceiveData 이벤트를 사용합니다.

**주문 TR의 요청이 서버에 정상적으로 전송되면**

**ReceiveMessage 이벤트**는 주문에 대한 결과를 무조건 수신받습니다.

-> 주문이 성공했는지, 실패했는지 판단

**ReceiveData 이벤트**는 주문이 성공했을 때, Outblock 의 데이터를 수신받습니다.

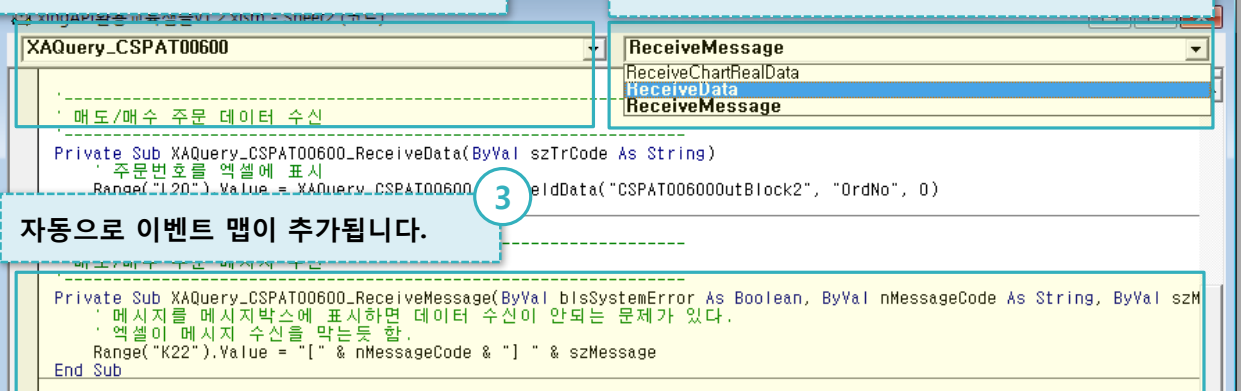
-> 주문이 실패했을 때는 수신받지 않을 수도 있고, 수신을 받았더라도 유효하지 않은 데이터입니다. 따라서, 주문이 성공했을 때에만 데이터를 확인시기 바랍니다.

**객체 콤보 내의 XAQuery 객체를 선택**

1

**XAQuery 객체의 이벤트 콤보리스트 중 'ReceiveData' 이벤트를 선택하면**

2



**자동으로 이벤트 맵이 추가됩니다.**

3

**Step 6** 이벤트를 받았다면 GetFieldData 메소드를 사용하여 수신된 주문 결과 데이터를 가져옵니다. (일반 데이터 조회 과정과 동일)

xingAPI활용교육샘플v1.2.xlsm - Sheet2 (코드)

XAQuery\_CSPAT00600

ReceiveMessage

ReceiveChartRealData

ReceiveData

ReceiveMessage

```

' 매도/매수 주문 데이터 수신
Private Sub XAQuery_CSPAT00600.ReceiveData(ByVal szTrCode As String)
    ' 주문번호를 엑셀에 표시
    Range("L20").Value = XAQuery_CSPAT00600.GetFieldData("CSPAT00600OutBlock2", "OrdNo", 0)
End Sub

' 매도/매수 주문 메시지 수신
Private Sub XAQuery_CSPAT00600.ReceiveMessage(ByVal blsSystemError As Boolean, ByVal nMessageCode As String, ByVal szMessage As String)
    ' 메시지를 메시지박스에 표시하면 데이터 수신에 안되는 문제가 있다.
    ' 엑셀이 메시지 수신을 막는듯 함.
    Range("K22").Value = "[" & nMessageCode & "]" & szMessage
End Sub

```

코드 | 추가 기능 | 컨트롤 | XML | 수정

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1						매입금액	평가금액	평가손익	손익률					
2						6,917,863	0	0	0.00%					
3						매입금액	평가금액	평가손익	손익률					
4	A005930	삼성전자	잔고	평균단가	현재가	6,308,000					계좌번호		주문	
5	A009540	현대중공업	3	148,833	103,000	446,500					비밀번호	****		
6	A078020	이트레이드	13	9,863	9,670	128,213					종목코드	A005930		
7	A105560	KB금융	1	35,150	41,050	35,150					종목명	삼성전자		
8											호가유형	00:지정가		
9											주문조건	0:없음		
10											주문수량	1		
11											주문단가	1,085,000		
12											원주문번호			
13											현재가	1,095,000		
14														
15											매도	매수		
16											정정	취소		
17														
18														
19														
20														
21														
22														
23														
24														

주문번호

메시지

종료 상태입니다

➤ 전체코드

```
Dim WithEvents XAQuery_CSPAT00600 As XAQuery

'-----
' 매도/매수 주문 전송
'-----

Private Sub Order(nOrderType)
    ' 객체 생성 및 Res 파일 등록
    If XAQuery_CSPAT00600 Is Nothing Then
        Set XAQuery_CSPAT00600 = CreateObject( " XA_DataSet.XAQuery " )
        XAQuery_CSPAT00600.ResFileName = " WresWCSPAT00600.res "
    End If
    ' Inblok 채우기
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "AcntNo", 0, "XXXXXXXXXX")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "InptPwd", 0, "XXXX")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "IsuNo", 0, "A078020")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdQty", 0, "10")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdPrc", 0, "10000")
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "BnsTpCode", 0, nOrderType)
    Call XAQuery_CSPAT00600.SetFieldData("CSPAT00600InBlock1", "OrdprcPtnCode", 0, "00")
    Call XAQuery_CSPAT00600.SetFieldData( " CSPAT00600InBlock1 " , "MgntrnCode", 0, "000")
    Call XAQuery_CSPAT00600.SetFieldData( " CSPAT00600InBlock1 " , " LoanDt " , 0, "")
    Call XAQuery_CSPAT00600.SetFieldData( " CSPAT00600InBlock1 " , " OrdCndiTpCode " , 0, "")
    nSuccess = XAQuery_CSPAT00600.Request(False)
    If nSuccess < 0 Then
        MsgBox " 전송오류 : " & nSuccess
    End If
End Sub

'-----
' 매도/매수 주문 요청 성공 시, 서버로부터 데이터 결과 수신 이벤트
'-----

Private Sub XAQuery_CSPAT00600_ReceiveData(ByVal szTrCode As String)
    ' 주문번호
    OrgNo = XAQuery_CSPAT00600.GetFieldData("CSPAT00600OutBlock2", "OrdNo", 0)
End Sub
```

```
' -----  
' 매도/매수 주문 요청에 대해 서버로부터 성공/실패 결과 수신 이벤트  
' -----  
  
Private Sub XAQuery_CSPAT00600_ReceiveMessage(ByVal bIsSystemError As Boolean, ByVal  
nMessageCode As String, ByVal szMessage As String)  
    Message = "[" & nMessageCode & "]" & szMessage  
End Sub  
  
' -----  
' 매도 버튼클릭시 이벤트  
' -----  
  
Private Sub btnOrder1_Click()  
    Call Order(1)  
End Sub  
  
' -----  
' 매수 버튼클릭시 이벤트  
' -----  
  
Private Sub btnOrder2_Click()  
    Call Order(2)  
End Sub
```

**연속 조회**는 Occurs(반복) 데이터의 총 갯수가 너무 많아서 한 번에 다 가져오지 못하는 것을 여러 번 나누어서 가져오는 것을 의미합니다.

HTS의 경우 스크롤바를 아래로 내리거나 다음 버튼 클릭 시, 연속 데이터를 가져오는 경우가 연속 조회에 해당됩니다.

### 아래의 코드는

xingAPI홈페이지 [자료실 - 히어로샘플자료실] "[주체별 수급분석 가져오기1 \(TR1702이용\)](#)" 게시물 내 '2014.03.17 - 히어로 등록파일1.xlsm' 엑셀 파일에서 확인 가능합니다.

조회 요청은 일반 조회와 동일한 Request() 메소드를 사용하지만, Request() 메소드의 입력 파라미터를 구분하여 일반 조회인지 연속 조회인지를 서버에 알려줍니다.

### 연속조회를 하기 위해서는

- ① 최초 조회의 결과 데이터에서 연속조회 키를 가져와 다음 조회의 입력값으로 입력 후
  - ② 서버에 연속 조회 요청
- 처리를 해야 합니다.

The image shows a screenshot of an Excel spreadsheet and the Visual Basic Editor. The Excel spreadsheet is titled '주체별 수급분석 가져오기1 (TR1702이용)\_2014.' and contains a table with columns for '외인/기관 중목별 동향 조회 (t1702)', '일자', '증가', and '전일대'. The Visual Basic Editor shows the code for the '기간별주가를 요청' (Request for Periodic Stock Prices) subroutine, which includes logic for handling continuous data retrieval using the 'Request(bnext)' method.

**1** [개발도구-Visual Basic] 메뉴를 클릭합니다.

**2** 코드를 확인할 수 있습니다.

외인/기관 중목별 동향 조회 (t1702)	일자	증가	전일대
122630	20140103	11.520	
	20140102	11.780	
	20131230	12.425	
	20131227	12.365	
	20131226	12.160	
	20131224	12.215	
	20131223	12.145	
	20131220	11.950	
	20131219	11.865	
	20131218	11.930	
	20131217	11.770	
	20131216	11.655	
	20131213	11.635	

```

Private Sub DisplayPrice(ByVal bNext As Boolean)
    '각해 생성 및 Res 할당
    If XAQuery.t1702 Is Nothing Then
        Set XAQuery.t1702 = CreateObject("XA.DataSet.XAQuery")
        XAQuery.t1702.ResFileName = "t1702.res"
    End If

    't1702 요청 : 공통 필드
    Call XAQuery.t1702.SetFieldData("t1702InBlock", "shcode", 0, Range("C2").Value)
    Call XAQuery.t1702.SetFieldData("t1702InBlock", "todt", 0, Range("J2").Value)
    Call XAQuery.t1702.SetFieldData("t1702InBlock", "voicld", 0, "1")
    Call XAQuery.t1702.SetFieldData("t1702InBlock", "mndp", 0, "0")
    Call XAQuery.t1702.SetFieldData("t1702InBlock", "cumulgb", 0, "0")

    '최초 조회시
    If bNext = False Then
        Call XAQuery.t1702.SetFieldData("t1702InBlock", "cts_date", 0, "")
        Call XAQuery.t1702.SetFieldData("t1702InBlock", "cts_id", 0, "")
    End If

    '연속 조회시
    Else
        Call XAQuery.t1702.SetFieldData("t1702InBlock", "cts_id", 0, Range("E2").Value)
        Call XAQuery.t1702.SetFieldData("t1702InBlock", "cts_date", 0, Range("E3").Value)
    End If

    Dim nReqID As Integer
    nReqID = XAQuery.t1702.Request(bNext)
    If nReqID < 0 Then
        MsgBox "전송에러 : " & nReqID
    End If
End Sub
    
```

## Step 1 연속 조회 값 입력하기

- ① 연속 조회 TR인 경우에는 TR의 Layout에 연속 조회를 위한 필드가 존재합니다.  
(일반적으로 CTS로 시작하는 필드명을 가지고 있음)
- ② 입력값을 위해 InBlock에 연속 조회 필드 (Outblock과 동일한 필드)가 있고 다음값을 알려주기 위해 OutBlock에 연속 조회 필드(Inblock과 동일한 필드)가 있습니다.
- ③ OutBlock 의 연속 조회용 필드의 값을 다음 조회 시에 InBlock의 연속 조회용 필드에 넣어주어야 합니다.

계좌 관련(잔고, 체결 등) 조회 등의 B타입 헤더 TR은 연속 조회용 필드가 없을 수 있으며, 이 경우에는 API내부에서 자동으로 연속조회 값을 입력합니다.

연속 조회를 하기 위해 Request() 요청 전에,

Outblock의 연속 조회 필드 값을 Inblock에 연속 조회 필드 값에 입력한다

```
key_idx = XAQuery_t1702.GetFieldData("t1702OutBlock", "cts_idx", 0)
```

```
key_date = XAQuery_t1702.GetFieldData("t1702OutBlock", "cts_date", 0)
```

```
call XAQuery_t1702..SetFieldData("t1702InBlock", "cts_idx", 0, key_idx)
```

```
call XAQuery_t1702..SetFieldData("t1702InBlock", "cts_date", 0, key_date)
```

이름	타입	크기	설명
t1702		264	외인/기관 종목별동향
t1702InBlock	input	29	기본입력
shcode	string	6	종목코드
todt	string	8	종료일자
volvalgb	string	1	금액수량구분(0:금액1:수량2:단가)
msmdgb	string	1	매수매도구분(0:순매수1:매수2:매도)
cumulgb	string	1	누적구분(0:일간1:누적)
cts_date	string	8	CTSDATE
cts_idx	long	4	CTSIDX
t1702OutBlock	output	12	기본출력
cts_idx	long	4	CTSIDX
cts_date	string	8	CTSDATE
t1702OutBlock1 - OCCURS	output	223	출력
date	string	8	일자
close	long	8	종가
sian	string	1	전일대비구분

연속조회를 위한  
InBlock의 필드

연속조회를 위한  
OutBlock의 필드

```

' 최초 조회시
If bNext = False Then
  Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_date", 0, "")
  Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_idx", 0, "")
' 최초시 공백
' 최초시 공백

' 연속 조회시
Else
  Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_idx", 0, Range("E2").Value)
  Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_date", 0, Range("E3").Value)
' 연속 조회시, 이전 조회 값
' 연속 조회시, 이전 조회 값
End If
    
```



## Step 2 연속조회 데이터 요청하기

- ① 연속조회 입력값을 입력하였다면 Request 메소드를 이용하여 서버에 요청합니다.  
이 때, Request 의 파라미터를 TRUE로 하여 연속조회임을 서버에 알려줍니다.

### Request() 메소드의 입력 파라미터로 구분

Request( FALSE ) : 입력파라미터가 False면 최초 조회  
Request( **TRUE** ) : 입력파라미터가 True면 연속 조회

```

' -----
' t1702 요청 : 공통 필드
Call XAQuery_t1702.SetFieldData("t1702InBlock", "shcode", 0, Range("C2").Value)
Call XAQuery_t1702.SetFieldData("t1702InBlock", "todt", 0, Range("J2").Value)
Call XAQuery_t1702.SetFieldData("t1702InBlock", "volvalgb", 0, "1")
Call XAQuery_t1702.SetFieldData("t1702InBlock", "msmdgb", 0, "0")
Call XAQuery_t1702.SetFieldData("t1702InBlock", "cumulgb", 0, "0")

' 최초 조회시
If bNext = False Then
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_date", 0, "")
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_idx", 0, "")

' 연속 조회시
Else
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_idx", 0, Range("E2").Value)
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_date", 0, Range("E3").Value)
End If

' -----

Dim nReqID As Integer
nReqID = XAQuery_t1702.Request(bNext)
If nReqID < 0 Then
    MsgBox "전 송 에 러 : " & nReqID
End If
    
```

bNext 값이 **True**면 연속 조회

1

▶ 전체코드

xingAPI홈페이지 [자료실 - 히어로샘플자료실] "[주체별 수급분석 가져오기1 \(TR1702이용\)](#)" 게시글 내 '2014.03.17 - 히어로 등록파일1.xlsm' 엑셀 파일의 코드를 '**최초 조회 후 연속 조회를 한 번만 요청하도록**' 수정한 코드입니다.

```
Dim WithEvents XAQuery_t1702 As XAQuery      ' [조회TR] 기간별투자자매매추이
Public g_bInit As Boolean

'-----
' 셀의 값이 변경되면 호출
'-----

Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Address = "$C$2" Then
        g_bInit = False
        DisplayPrice1 (g_bInit)              ' displayprice1에 g_blnit 값을 주면서 실행
    End If
End Sub

'-----
' 기간별주가를 요청
'-----

Private Sub DisplayPrice1(ByVal bNext As Boolean)
    ' 객체 생성 및 Res 할당
    If XAQuery_t1702 Is Nothing Then
        Set XAQuery_t1702 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_t1702.ResFileName = "WresWt1702.res"
    End If

    '-----
    ' t1702 요청 : 공통 필드
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "shcode", 0, Range("C2").Value) ' 종목코드
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "todt", 0, Range("J2").Value)   ' 종료일자
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "volvalgb", 0, "1") ' 0:금액, 1:수량, 2:단가
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "msmdgb", 0, "0") ' 0:순매수, 1:매수, 2:매도
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cumulgb", 0, "0") ' 0:일간, 1:누적
```

```

If bNext = False Then ' 최초 조회시
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_date", 0, "") ' 최초시 공백
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_idx", 0, "") ' 최초시 공백
Else ' 연속 조회시
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_idx", 0, Range("E2").Value) ' 연속키값
    Call XAQuery_t1702.SetFieldData("t1702InBlock", "cts_date", 0, Range("E3").Value) '연속키값
End If

'-----

Dim nReqID As Integer
nReqID = XAQuery_t1702.Request(bNext)
If nReqID < 0 Then
    MsgBox "전송에러 : " & nReqID
End If

End Sub

'-----
' t1702 데이터 수신
'-----

Private Sub XAQuery_t1702_ReceiveData(ByVal szTrCode As String)

    Range("E2").Value = XAQuery_t1702.GetFieldData("t1702OutBlock", "cts_idx", 0)
    Range("E3").Value = XAQuery_t1702.GetFieldData("t1702OutBlock", "cts_date", 0)

    g_bInit = Range("G2").Value

    '-----
    ' Array에 데이터를 담은 다음에 Array를 통째로 엑셀에 표시한다.
    ' 이렇게 하지 않으면 표시속도가 너무 느리다.
    Dim arrPrice() ' 동적 배열 설정

    '-----
    ' t1702OutBlock1 은 Occurs 이므로 Occurs 갯수를 먼저 알아낸다
    Dim nCount As Integer
    Dim i As Integer

```

```

nCount = XAQuery_t1702.GetBlockCount("t1702OutBlock1") ' 받아오는 데이터 양 확인

ReDim Preserve arrPrice(0 To nCount, 18) ' 받아오는 데이터 양에 따른 배열 재설정

' 임시 배열에 값 입력
For i = 0 To nCount - 1
    arrPrice(i, 0) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "date", i) ' 일자
    arrPrice(i, 1) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "close", i) ' 종가
    arrPrice(i, 2) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "change", i) ' 전일대비
    arrPrice(i, 3) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "diff", i) ' 등락율
    arrPrice(i, 4) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "volume", i) ' 거래량
    arrPrice(i, 5) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "amt0008", i) ' 개인
    arrPrice(i, 6) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "amt0088", i) ' 외국계
    arrPrice(i, 7) = XAQuery_t1702.GetFieldData("t1702OutBlock1", "amt0018", i) ' 기관계
Next i

' 최초 조회일 때와 연속 조회일 때를 나누어서 셀 입력
If g_bInit = False Then ' 최초 조회

    For j = 0 To nCount - 1
        Cells(j + 5, 3).Value = arrPrice(j, 0)
        Cells(j + 5, 4).Value = arrPrice(j, 1)
        Cells(j + 5, 5).Value = arrPrice(j, 2)
        Cells(j + 5, 6).Value = arrPrice(j, 3)
        Cells(j + 5, 7).Value = arrPrice(j, 4)
        Cells(j + 5, 8).Value = arrPrice(j, 5)
        Cells(j + 5, 9).Value = arrPrice(j, 6)
        Cells(j + 5, 10).Value = arrPrice(j, 7)

        If j = 19 Then
            Range("G2").Value = "TRUE"
        End If
    Next j

Else ' 연속 조회
    lastrow = Cells(Rows.Count, "C").End(3).Row ' 마지막행 찾기
    Range("F2").Value = lastrow ' 마지막행 셀에 입력 : 마지막 값이 사라지기 때문에

```

```

For j = 0 To nCount - 1
    Cells(j + lastrow + 1, 3).Value = arrPrice(j, 0)
    Cells(j + lastrow + 1, 4).Value = arrPrice(j, 1)
    Cells(j + lastrow + 1, 5).Value = arrPrice(j, 2)
    Cells(j + lastrow + 1, 6).Value = arrPrice(j, 3)
    Cells(j + lastrow + 1, 7).Value = arrPrice(j, 4)
    Cells(j + lastrow + 1, 8).Value = arrPrice(j, 5)
    Cells(j + lastrow + 1, 9).Value = arrPrice(j, 6)
    Cells(j + lastrow + 1, 10).Value = arrPrice(j, 7)
Next j

```

End If

' 최초 조회 했으면 연속조회 한 번만 더 해보기

```

If g_bInit = False Then
    g_bInit = True
    DisplayPrice1 (g_bInit)
End If

```

End Sub

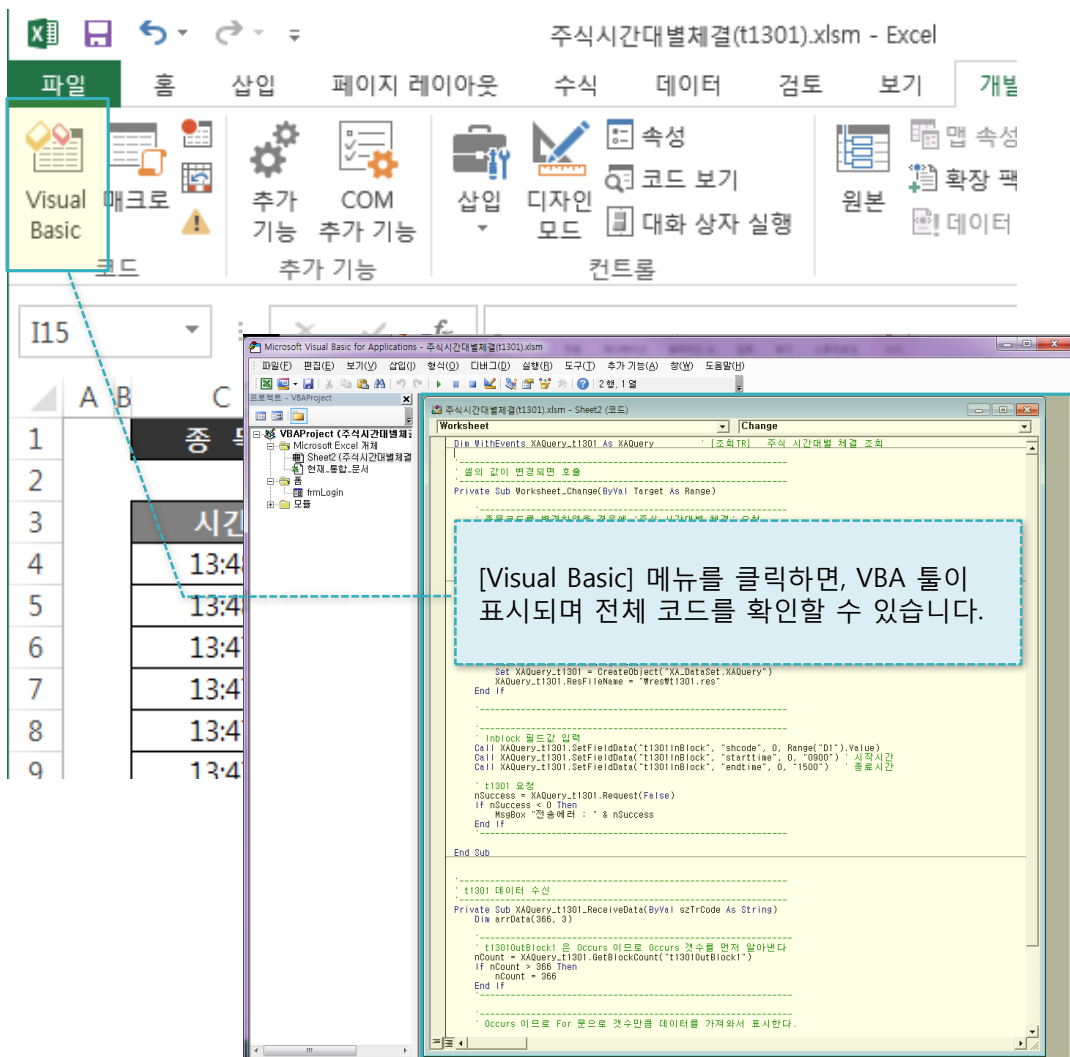
Occurs는 프로그래밍 언어인 COBOL에서 나온 단어로 동일한 구조(Layout)의 데이터가 반복해서 나오는 구조를 의미합니다. HTS의 '시간대별체결' 화면의 그리드처럼 동일한 구조의 데이터가 반복해서 나오며, 프로그래밍 언어에서의 배열(Array)와 같은 의미입니다.

### 아래의 코드는

t1301 (주식 시간대별 체결 조회) TR을 이용해 조회하는 예제입니다.

이것은 xingAPI 설치 폴더 내 "엑셀샘플W주식시간대별체결(t1301).xslm" 파일에서도 확인 가능합니다.

- ▶ 조회 요청은 일반 조회와 동일하며, 데이터 수신 시 반복되는 데이터 부분을 가져오기 위한 index 처리가 필요합니다.



## Step 1 Occurs(반복데이터) 여부를 확인하는 방법

- ① HTS 의 "[1301] 시간대별체결" 화면을 보시면 아래와 같이 시간대별 체결이 그리드에 됩니다.
- ② "[1301] 시간대별체결" 화면은 t1301 TR을 사용해서 만들어진 부분으로 그리드 영역은 t1301OutBlock1 에 해당합니다.
- ③ DevCenter의 TR Layout에서 t1301OutBlock1 을 선택하시면, TR속성창의 Occurs여부 속성에 Occurs 임을 표시합니다.

HTS의 [1301] 시간대별체결

078020	관	30	미트레이드증권	특이거래량		항목	차트	조회
07:30	~	15:15	매도체결량	981	(5) 매수체결량	507	(4)	
시간	현재가	대비	등락율	체결수량	체결강도	거래량		
11:18:37	10,250 ▼	50	-0.49	180	51.68	1,568		
10:53:56	10,250 ▼	50	-0.49	500	63.30	1,388		
10:51:32	10,250 ▼	50	-0.49	1	0.87	888		
10:50:24	10,250 ▼	50	-0.49	590	0.75	887		
10:43:18	10,250 ▼	50	-0.49	100	2.84	297		
10:15:50	10,250 ▼	50	-0.49	100	5.41	197		
09:37:25	10,300	0	0.00	5	54.55	97		
09:24:54	10,300	0	0.00	1	9.09	92		
09:07:40	10,250 ▼	50	-0.49	11	0.00	91		
09:04:46	10,250 ▼	50	-0.49	80	0.00	80		

t1301 x

이름	타입	크기	설명
t1301		171	주식 시간대별 체결 조회
t1301InBlock	input	36	기본입력
shcode	string	6	단축코드
cvolume	long	12	특이거래량
starttime	string	4	시작시간
endtime	string	4	종료시간
cts_time	string	10	시간CTS
t1301OutBlock	output	10	출력
cts_time	string	10	시간CTS
t1301OutBlock1	output	125	출력1
chetime	string	10	시간
price	long	8	현재가
sign	string	1	전일대비구분
change	long	8	전일대비
diff	float	6.2	등락율
cvolume	long	12	체결수량
chdegree	float	8.2	체결강도
volume	long	12	거래량
mdvolume	long	12	매도체결수량
mdchecnt	long	8	매도체결건수
msvolume	long	12	매수체결수량
mschecnt	long	8	매수체결건수
revolume	long	12	순체결량
rechecnt	long	8	순체결건수

TR 속성

TR 정보	
TR 코드	t1301
TR 명	주식 시간대별 체결 ...
Header	A
Attribute	×
Block Mode	×
초당 전송 수	2
크기	171
Block 정보	
Block 코드	t1301OutBlock1
Block 명	출력1
타입	output
크기	125
Occurs 여부	<input checked="" type="radio"/>
Field 정보	
Field 코드	
Field 명	
타입	
크기	
Field 속성	

DevCenter의 t1301 Layout

## Step 1 Occurs(반복데이터) 총 Row갯수(행수) 알아오기

- ① HTS의 "[1301] 시간대별체결" 화면의 그리드 총 행수는 20개입니다.
- ② API에서는 총 Row(행)수를 알기 위해 XAQuery 객체의 GetBlockCount 메소드를 사용합니다.

총 Row 갯수

```
nCount = XAQuery_t1301.GetBlockCount( "t1301OutBlock1" )
```

t1301 데이터 수신

```
Private Sub XAQuery_t1301_ReceiveData(ByVal szTrCode As String)  
    Dim arrData(366, 3)
```

```
    ' t1301OutBlock1 은 Occurs 이므로 Occurs 갯수를 먼저 알아낸다  
    nCount = XAQuery_t1301.GetBlockCount("t1301OutBlock1")  
    If nCount > 366 Then  
        nCount = 366  
    End If
```

```
End Sub
```



## Step 2 Occurs(반복데이터)의 특정한 Row(행) 데이터 가져오기

- ① Occurs의 데이터를 가져오는 방식은 일반 데이터 조회와 동일하게 GetFieldData 메소드를 사용합니다.
- ② "[1301] 시간대별체결" 화면의 Row별 데이터를 가져오기 위해서는 3번째 파라미터에 가져오려는 데이터 행의 Index를 넣어주어야 합니다.

Row값  
price = XAQuery\_t1301.GetFieldData( "t1301OutBlock1", "price", **Index** )

※ Index는 1부터 시작하지 않고 0부터 시작함

078020	관	미트레이드증권	특이거래량	항목	차트	조회
07:30	~	15:15	매도체결량 981	(5) 매수체결량 507	(4)	
시간	현재가	대비	등락율	체결수량	체결강도	거래량
0 11:18:37	10,250	▼	50	-0.49	180	51.68 1,568
1 10:53:56	10,250	▼	50	-0.49	500	63.30 1,388
2 10:51:32	10,250	▼	50	-0.49	1	
3 10:50:24	10,250	▼	50	-0.49	590	
4 10:43:18	10,250	▼	50	-0.49	100	
5 10:15:50	10,250	▼	50	-0.49	100	
6 09:37:25	10,300		0	0.00	5	5
7 09:24:54	10,300		0	0.00	1	
8 09:07:40	10,250	▼	50	-0.49	11	
9 09:04:46	10,250	▼	50	-0.49	80	

t1301OutBlock1	output	125	출력1
chetime	string	10	시간
price	long	8	현재가
sign	string	1	전일대비구분
change	long	8	전일대비
diff	float	6.2	등락율
cvolume	long	12	체결수량
chdegree	float	8.2	체결강도
volume	long	12	거래량
mdvolume	long	12	매도체결수량
mdchecnt	long	8	매도체결건수
msvolume	long	12	매수체결수량
mschecnt	long	8	매수체결건수
revolume	long	12	순체결량
rechecnt	long	8	순체결건수

```

' Occurs 이므로 For 문으로 갯수만큼 데이터를 가져와서 표시한다.
For i = 0 To nCount - 1
    sTime = XAQuery_t1301.GetFieldData("t1301OutBlock1", "chetime", i) ' 수신시간
    arrData(i, 0) = Left(sTime, 2) & ":" & Mid(sTime, 3, 2)
    arrData(i, 1) = XAQuery_t1301.GetFieldData("t1301OutBlock1", "price", i) ' 현재가
    arrData(i, 2) = XAQuery_t1301.GetFieldData("t1301OutBlock1", "cvolume", i) ' 체결수량
Next i
Range("C4:E369") = arrData

```

▶ 전체코드

```
Dim WithEvents XAQuery_t1301 As XAQuery      ' [조회TR]  주식 시간대별 체결 조회

' 셀의 값이 변경되면 호출
Private Sub Worksheet_Change(ByVal Target As Range)
    ' 종목코드를 변경하였을 경우에 '주식 시간대별 체결' 요청
    ' 종목코드를 입력한 셀인지 체크
    If Target.Address = "$D$1" Then
        Call t1301Data
    End If
End Sub

' 주식 시간대별 체결 조회
Private Sub t1301Data()
    ' 객체 생성 및 Res 할당
    If XAQuery_t1301 Is Nothing Then
        Set XAQuery_t1301 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_t1301.ResFileName = "WresWt1301.res"
    End If

    ' Inblock 필드값 입력
    Call XAQuery_t1301.SetFieldData("t1301InBlock", "shcode", 0, Range("D1").Value)
    Call XAQuery_t1301.SetFieldData("t1301InBlock", "starttime", 0, "0900") ' 시작시간
    Call XAQuery_t1301.SetFieldData("t1301InBlock", "endtime", 0, "1500") ' 종료시간

    ' t1301 요청
    nSuccess = XAQuery_t1301.Request(False)
    If nSuccess < 0 Then
        MsgBox "전송에러 : " & nSuccess
    End If
End Sub
```

' t1301 데이터 수신

```
Private Sub XAQuery_t1301_ReceiveData(ByVal szTrCode As String)
```

```
    Dim arrData(366, 3)
```

' t1301OutBlock1 은 Occurs 이므로 Occurs 갯수를 먼저 알아낸다

```
nCount = XAQuery_t1301.GetBlockCount("t1301OutBlock1")
```

```
If nCount > 366 Then
```

```
    nCount = 366
```

```
End If
```

' Occurs 이므로 For 문으로 갯수만큼 데이터를 가져와서 표시한다.

```
For i = 0 To nCount - 1
```

```
    sTime = XAQuery_t1301.GetFieldData("t1301OutBlock1", "chetime", i) ' 수신시간
```

```
    arrData(i, 0) = Left(sTime, 2) & ":" & Mid(sTime, 3, 2)
```

```
    arrData(i, 1) = XAQuery_t1301.GetFieldData("t1301OutBlock1", "price", i) ' 현재가
```

```
    arrData(i, 2) = XAQuery_t1301.GetFieldData("t1301OutBlock1", "cvolume", i) ' 체결수량
```

```
Next i
```

```
Range("C4:E369") = arrData
```

```
End Sub
```

주식 현재가와 같은 데이터는 장중에도 계속 바뀌게 됩니다.

**조회 TR** 은 요청(Request)시점의 데이터를 전송해 주므로, 그 이후 변경된 데이터를 받지 못합니다.

#### 실시간 TR은 (조회TR과는 다르게)

요청시점 이후에 데이터가 변경되면 그 때 데이터를 전송해주며,  
요청을 해제하기 전까지 데이터가 변경 될 때마다 데이터를 전송해 줍니다.  
다만, 요청시점 이후에 데이터가 변경이 되면 전송해 주기 때문에  
요청시점에서의 데이터는 전송해주지 않습니다.

그러므로, 주식 현재가를 구성하기 위해서는

조회TR로 요청시점의 데이터를 가져온 후에 실시간 데이터를 요청하여  
그 이후 데이터를 실시간으로 받으셔야 합니다.

### Step 1 XAReal 객체를 선언합니다. (일반 데이터 조회하기의 XAQuery 객체 선언과 동일)

**1** [개발도구-Visual Basic] 메뉴를 클릭합니다.

**2** XAReal 객체를 선언합니다.

```
Dim WithEvents XAQuery_t1101 As XAQuery
Dim WithEvents XAReal_S3_ As XAReal
Dim WithEvents XAReal_HA_ As XAReal
```

**Step 2** XAReal 객체를 생성합니다. (일반 데이터 조회하기의 XAQuery 객체 생성과 동일)

```

' 객체를 생성한다.
If XAReal_S3_ Is Nothing Then
    Set XAReal_S3_ = CreateObject("XA_DataSet.XAReal")
End If

```

**Step 3** Res 파일을 등록합니다. (일반 데이터 조회하기의 XAQuery 객체 생성과 동일)

- ① 실시간 데이터 조회를 원하는 TR코드의 RES 파일을 [DevCenter – Real목록 창]에서 다운로드 받습니다.
  - TR코드(TR명)를 클릭하면, 클릭하는 시점에 서버에서 자동으로 다운로드
- ② ResFileName 메소드를 이용하여 XAReal 객체에 등록합니다.

res파일은 xingAPI COM버전에서 사용하는 것으로, TR의 입출력 정보를 정의하며,  
res파일이 없으면 COM버전을 이용할 수 없으므로 반드시 미리 다운받아야 합니다.

```

' 객체를 생성한다.
If XAReal_S3_ Is Nothing Then
    Set XAReal_S3_ = CreateObject("XA_DataSet.XAReal")
    XAReal_S3_.ResFileName = "₩res₩S3_.res"
End If

```

**Step 4** 실시간 데이터를 등록하기 전에 입력 데이터 값을 설정합니다.  
(일반 데이터 조회하기의 XAQuery 객체 생성과 동일)

계좌에 관한 실시간 데이터는 입력값이 없습니다. (SC0 ~ SC4, O01, C01...)

**void SetFieldData (BSTR szBlockName, BSTR szFieldName, BSTR szData)**

szBlockName : 블록명

szFieldName : 필드명

szData : 입력 데이터 (값)

The screenshot displays the XAQuery development environment. At the top, a table lists the fields of the S3\_ block:

이름	타입	크기	설명
S3_		203	KOSPI체결
InBlock	input	6	입력
shcode	string	6	단축코드
OutBlock	output	197	출력
chetime	string	6	체결시간
sign	string	6	체결구분
change	long	8	현재가
drate	float	6.2	시가시간
price	long	8	시가
opentime	string	6	고가시간
open	long	8	고가
hightime	string	6	저가시간
high	long	8	저가
lowtime	string	6	체결구분
cgubun	string	1	체결량
cvolume	long	8	누적거래량
volume	long	12	

A callout box with a circled '1' points to the 'InBlock' section, stating: **Inblock 내 필드의 입력 데이터값을 설정합니다.**

Below the table, the code editor shows the following VBA code:

```

' 객체를 생성한다.
If XAReal_S3_ Is Nothing Then
    Set XAReal_S3_ = CreateObject("XA_DataSet.XAReal")
    XAReal_S3_.ResFileName = "res#S3_.res"
End If

Call XAReal_S3_.SetFieldData("InBlock", "shcode", "078020")

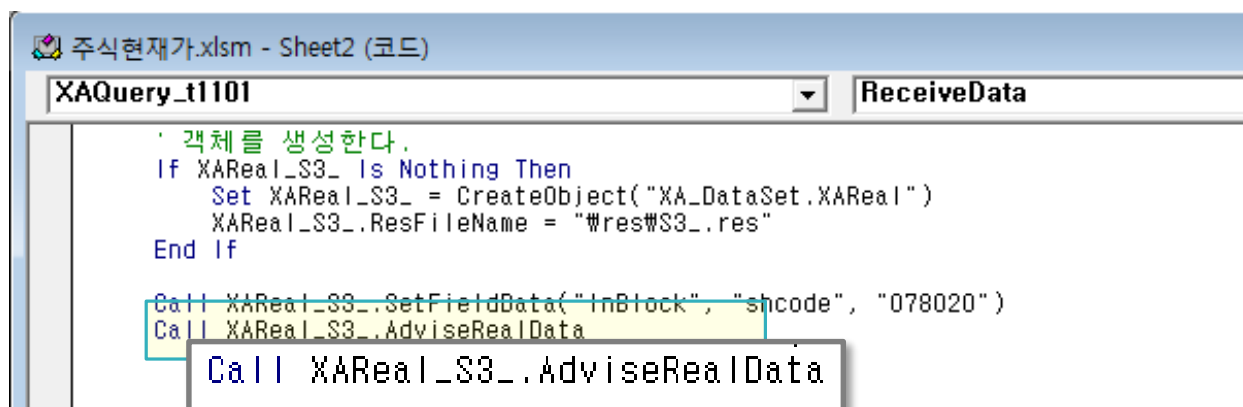
```

The final line of code, **Call XAReal\_S3\_.SetFieldData("InBlock", "shcode", "078020")**, is highlighted in a yellow box.

**Step 5** 서버로 실시간 데이터를 등록합니다.

XAReal 객체로 AdviseRealData()를 한 후에 다른 종목도 추가로 실시간 시세를 받기를 원하면 새로운 종목을 입력한 후에 AdviseRealData()로 요청하면 됩니다.

-> 즉, 10개의 종목을 등록할 경우, SetFieldData() 와 AdviseRealData()를 10번 반복합니다.

**void AdviseRealData()**

The screenshot shows a VBA code editor window titled '주식현재가.xlsm - Sheet2 (코드)'. The active sheet is 'XAQuery\_t1101' and the active procedure is 'ReceiveData'. The code is as follows:

```
' 객체를 생성한다.  
If XAReal_S3_ Is Nothing Then  
    Set XAReal_S3_ = CreateObject("XA_DataSet.XAReal")  
    XAReal_S3_.ResFileName = "₩res₩S3_.res"  
End If  
  
Call XAReal_S3_.SetFieldData("InBlock", "shcode", "078020")  
Call XAReal_S3_.AdviseRealData  
Call XAReal_S3_.AdviseRealData
```

The line 'Call XAReal\_S3\_.AdviseRealData' is highlighted in yellow, and a tooltip box shows the same line of code.

**Step 6** 서버로부터 실시간 데이터를 받습니다.

서버로부터 실시간 데이터를 수신했을 때 이벤트가 발생하며, 이벤트는 XAReal 객체의 ReceiveRealData 이벤트를 사용합니다.

```
void ReceiveRealData ( BSTR szTrCode )
```

szTrCode : TR코드

1 객체 콤보 내의 XAReal 객체를 선택

2 XAReal 객체의 이벤트 콤보리스트 중 'ReceiveRealData' 이벤트를 선택하면

3 자동으로 이벤트 맵이 추가됩니다.

```

주식현재가.xlsm - Sheet2 (코드)
XAReal_S3_
XAReal_S3_

' 객체를 생성한다.
If XAReal_S3_ Is Nothing Then
    Set XAReal_S3_ = CreateObject("XA_DataSet.XAReal")
    XAReal_S3_.ResFileName = "res\res\S3_.res"
End If

Call XAReal_S3_.SetFieldData("InBlock", "shcode", "078020")
Call XAReal_S3_.AdviseRealData

End Sub

Private Sub XAReal_S3_ReceiveRealData(ByVal szTrCode As String)
End Sub
  
```



**Step 7** 이벤트를 받았다면 GetFieldData 메소드를 사용하여 수신 데이터를 가져옵니다 .

실시간 데이터는 Occurs의 개념이 없이 단일 데이터입니다.

**void BSTR GetFieldData (BSTR szBlockName, BSTR szFieldName)**

szBlockName : 블록명  
szFieldName : 필드명

반환값 : 블록의 필드 값

주식현재가.xlsm - Sheet2 (코드)

XAReal\_S3\_ ReceiveRealData

```
Call XAReal_S3_.AdviseRealData
If XAReal_HA_ Is Nothing Then
    Set XAReal_HA_ = CreateObject("XA_DataSet.XAReal")
End If
Private Sub XAReal_S3_ReceiveRealData(ByVal szTrCode As String)
    Sheet2.Cells(4, 2) = XAReal_S3_.GetFieldData("OutBlock", "price")
    Sheet2.Cells(4, 3) = GetSign(XAReal_S3_.GetFieldData("OutBlock", "sign"))
    Sheet2.Cells(4, 4) = XAReal_S3_.GetFieldData("OutBlock", "change")
    Sheet2.Cells(4, 5) = XAReal_S3_.GetFieldData("OutBlock", "drate")
    Sheet2.Cells(4, 6) = XAReal_S3_.GetFieldData("OutBlock", "volume")
End Sub
```

1

실시간 데이터를 수신할 때마다 새로운 데이터를 표시합니다.

Visual Basic 매크로 추가 기능 COM 추가 기능 삽입 디자인 모드 XML 문서 창

G17

	A	B	C	D	E	F	G
1							
2		종 목	078020		이트레이드증권		
4		9,810 ▲	20		0.20		118
5		매도잔량					매수잔량
6							
7							
8							
9							

2

## ➤ 전체코드

```

Dim WithEvents XAQuery_t1101 As XAQuery
Dim WithEvents XAReal_S3_ As XAReal
Dim WithEvents XAReal_H1_ As XAReal

'-----
' 전일대비 구분 기호 변환
Private Function GetSign(ByVal sSign As String)
    If sSign = "1" Then
        GetSign = "↑"
    ElseIf sSign = "2" Then
        GetSign = "▲"
    ElseIf sSign = "4" Then
        GetSign = "↓"
    ElseIf sSign = "5" Then
        GetSign = "▼"
    End If
End Function

'-----
' 로그인 버튼 클릭 이벤트 수신
Private Sub cbLogin_Click()
    frmLogin.Show
End Sub

'-----
' 엑셀 시트 변환 이벤트 수신
Private Sub Worksheet_Change(ByVal Target As Range)
    ' 종목코드 입력
    If Target.Row = 2 And Target.Column = 3 Then
        Call Request_t1101
    End If
End Sub

```

```

'-----
' 주식 현재가 요청
Private Sub Request_t1101()

    ' 기존에 걸어놓은 실시간을 해제한다.
    If XAReal_S3_ Is Nothing Then
    Else
        Call XAReal_S3_.UnadviseRealData
    End If
    If XAReal_H1_ Is Nothing Then
    Else
        Call XAReal_H1_.UnadviseRealData
    End If

    ' 객체를 생성한다.
    If XAQuery_t1101 Is Nothing Then
        Set XAQuery_t1101 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_t1101.ResFileName = "WresWt1101.res"
    End If

    ' 데이터 세팅
    Call XAQuery_t1101.SetFieldData("t1101InBlock", "shcode", 0, Sheet2.Cells(2, 3))

    ' 데이터 요청
    If XAQuery_t1101.Request(False) = False Then
        MsgBox "전송오류"
    End If

End Sub

'-----
' 주식 현재가 요청 성공 시, 서버로부터 수신받는 이벤트
Private Sub XAQuery_t1101_ReceiveData(ByVal szTrCode As String)

    Dim arrPrice(5)
    Sheet2.Cells(2, 5) = XAQuery_t1101.GetFieldData("t1101OutBlock", "hname", 0) ' 종목명

```

```

arrPrice(0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "price", 0)
arrPrice(1) = GetSign(XAQuery_t1101.GetFieldData("t1101OutBlock", "sign", 0))
arrPrice(2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "change", 0)
arrPrice(3) = XAQuery_t1101.GetFieldData("t1101OutBlock", "diff", 0)
arrPrice(4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "volume", 0)
Range("B4:F4").Value = arrPrice

```

' 현재가

' 전일대비구분

' 전일대비

' 등락율

' 누적거래량

' 10호가

```

Dim arrHoga(21, 5)
sTime = XAQuery_t1101.GetFieldData("t1101OutBlock", "hotime", 0)
Sheet2.Cells(5, 4) = Left(sTime, 2) & ":" & Mid(sTime, 3, 2) & ":" & Mid(sTime, 5, 2)

```

```

arrHoga(0, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem10", 0)
arrHoga(0, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho10", 0)
arrHoga(19, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho10", 0)
arrHoga(19, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem10", 0)
arrHoga(1, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem9", 0)
arrHoga(1, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho9", 0)
arrHoga(18, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho9", 0)
arrHoga(18, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem9", 0)

```

```

arrHoga(2, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem8", 0)
arrHoga(2, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho8", 0)
arrHoga(17, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho8", 0)
arrHoga(17, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem8", 0)

```

```

arrHoga(3, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem7", 0)
arrHoga(3, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho7", 0)
arrHoga(16, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho7", 0)
arrHoga(16, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem7", 0)

```

```

arrHoga(4, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem6", 0)
arrHoga(4, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho6", 0)
arrHoga(15, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho6", 0)
arrHoga(15, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem6", 0)

```

```
arrHoga(5, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem5", 0)
arrHoga(5, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho5", 0)
arrHoga(14, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho5", 0)
arrHoga(14, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem5", 0)
```

```
arrHoga(6, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem4", 0)
arrHoga(6, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho4", 0)
arrHoga(13, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho4", 0)
arrHoga(13, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem4", 0)
```

```
arrHoga(7, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem3", 0)
arrHoga(7, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho3", 0)
arrHoga(12, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho3", 0)
arrHoga(12, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem3", 0)
```

```
arrHoga(8, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem2", 0)
arrHoga(8, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho2", 0)
arrHoga(11, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho2", 0)
arrHoga(11, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem2", 0)
```

```
arrHoga(9, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerrem1", 0)
arrHoga(9, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offerho1", 0)
arrHoga(10, 2) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidho1", 0)
arrHoga(10, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bidrem1", 0)
```

```
arrHoga(20, 0) = XAQuery_t1101.GetFieldData("t1101OutBlock", "offer", 0)
arrHoga(20, 4) = XAQuery_t1101.GetFieldData("t1101OutBlock", "bid", 0)
```

```
Range("B6:F26").Value = arrHoga
```

---

'실시간을 등록한다.

```
If XAReal_S3_ Is Nothing Then
```

```
    Set XAReal_S3_ = CreateObject("XA_DataSet.XAReal")
```

```
    XAReal_S3_.ResFileName = "WresWS3_.res"
```

```
End If
```

```
Call XAReal_S3_.SetFieldData("InBlock", "shcode", Sheet2.Cells(2, 3).Text)
Call XAReal_S3_.AdviseRealData
```

```
If XAReal_H1_ Is Nothing Then
```

```
    Set XAReal_H1_ = CreateObject("XA_DataSet.XAReal")
```

```
    XAReal_H1_.ResFileName = "WresWH1_.res"
```

```
End If
```

```
Call XAReal_H1_.SetFieldData("InBlock", "shcode", Sheet2.Cells(2, 3).Text)
```

```
Call XAReal_H1_.AdviseRealData
```

```
End Sub
```

```
'-----
```

```
' KOSPI호가잔량 실시간 데이터 발생 시, 서버로부터 수신받는 이벤트
```

```
Private Sub XAReal_H1_ReceiveRealData(ByVal szTrCode As String)
```

```
    Dim arrHoga(21, 5)
```

```
    sTime = XAReal_H1_.GetFieldData("OutBlock", "hotime")
```

```
    Sheet2.Cells(5, 4) = Left(sTime, 2) & ":" & Mid(sTime, 3, 2) & ":" & Mid(sTime, 5, 2)
```

```
    arrHoga(0, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem10")
```

```
    arrHoga(0, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho10")
```

```
    arrHoga(19, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho10")
```

```
    arrHoga(19, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem10")
```

```
    arrHoga(1, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem9")
```

```
    arrHoga(1, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho9")
```

```
    arrHoga(18, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho9")
```

```
    arrHoga(18, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem9")
```

```
    arrHoga(2, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem8")
```

```
    arrHoga(2, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho8")
```

```
    arrHoga(17, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho8")
```

```
    arrHoga(17, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem8")
```

```
    arrHoga(3, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem7")
```

```
arrHoga(3, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho7")
arrHoga(16, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho7")
arrHoga(16, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem7")

arrHoga(4, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem6")
arrHoga(4, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho6")
arrHoga(15, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho6")
arrHoga(15, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem6")
arrHoga(5, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem5")
arrHoga(5, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho5")
arrHoga(14, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho5")
arrHoga(14, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem5")

arrHoga(6, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem4")
arrHoga(6, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho4")
arrHoga(13, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho4")
arrHoga(13, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem4")

arrHoga(7, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem3")
arrHoga(7, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho3")
arrHoga(12, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho3")
arrHoga(12, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem3")

arrHoga(8, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem2")
arrHoga(8, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho2")
arrHoga(11, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho2")
arrHoga(11, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem2")

arrHoga(9, 0) = XAReal_H1_.GetFieldData("OutBlock", "offerrem1")
arrHoga(9, 2) = XAReal_H1_.GetFieldData("OutBlock", "offerho1")
arrHoga(10, 2) = XAReal_H1_.GetFieldData("OutBlock", "bidho1")
arrHoga(10, 4) = XAReal_H1_.GetFieldData("OutBlock", "bidrem1")

arrHoga(20, 0) = XAReal_H1_.GetFieldData("OutBlock", "totofferrem")
arrHoga(20, 4) = XAReal_H1_.GetFieldData("OutBlock", "totbidrem")
```

End Sub

```
'-----  
' KOSPI체결 실시간 데이터 발생 시, 서버로부터 수신받는 이벤트  
Private Sub XAReal_S3_ReceiveRealData(ByVal szTrCode As String)  
    Sheet2.Cells(4, 2) = XAReal_S3_GetFieldData("OutBlock", "price")           ' 현재가  
    Sheet2.Cells(4, 3) = GetSign(XAReal_S3_GetFieldData("OutBlock", "sign"))   ' 전일대비구분  
    Sheet2.Cells(4, 4) = XAReal_S3_GetFieldData("OutBlock", "change")         ' 전일대비  
    Sheet2.Cells(4, 5) = XAReal_S3_GetFieldData("OutBlock", "drate")          ' 등락율  
    Sheet2.Cells(4, 6) = XAReal_S3_GetFieldData("OutBlock", "volume")         ' 누적거래량  
End Sub
```



감사합니다