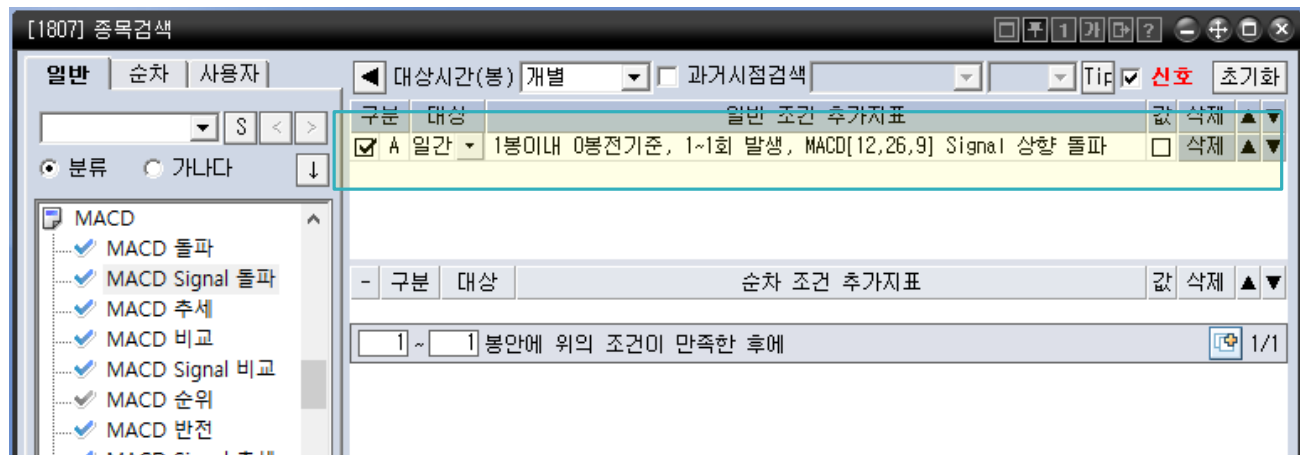


xingAPI 부가기능

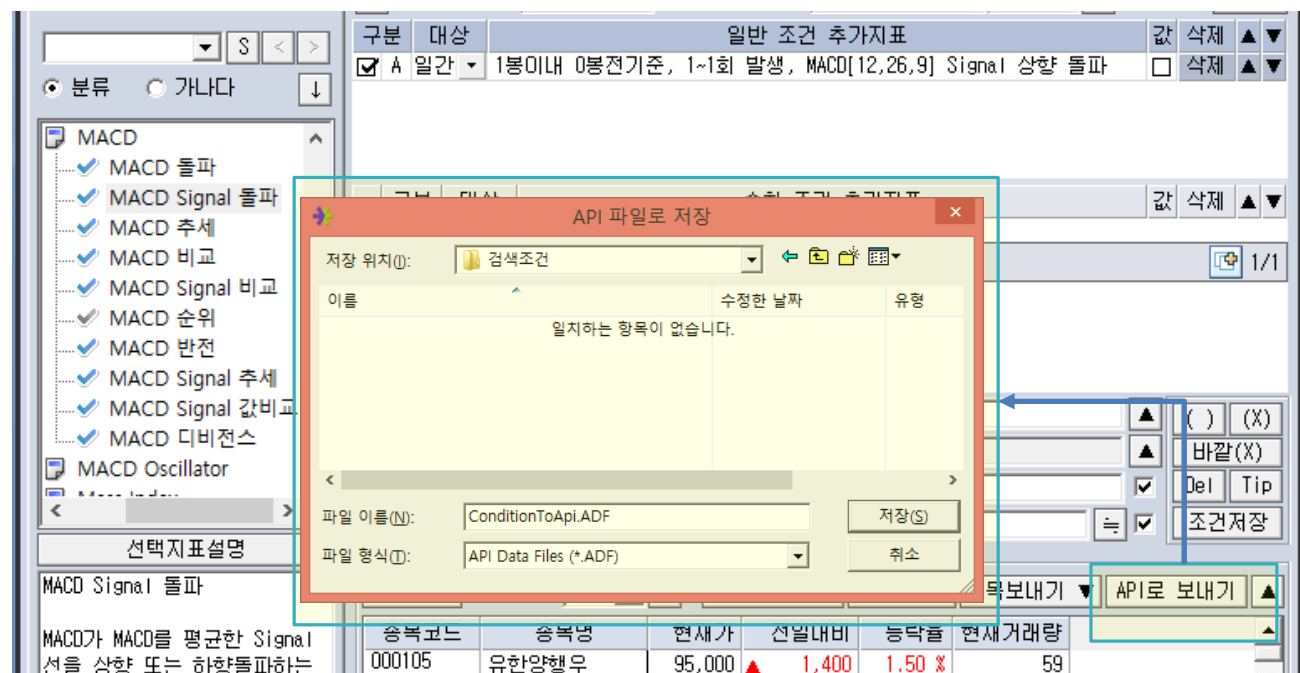
▶ HTS의 종목검색 기능을 지원합니다.

Step 1 HTS '[1807] 종목검색' 화면에서 검색 조건을 설정합니다.

조건 설정 방법은 HTS 도움말을 참조하시기 바랍니다.



Step 2 'API로 보내기' 버튼을 이용해 파일로 저장합니다.



Step 3 개발 프로그램에서 XAQuery 객체에 t1833 TR의 res를 할당합니다.

Step 4 개발 프로그램에서 [RequestService\(\)](#) 함수를 통해 데이터를 요청합니다.

※ 수신 처리는 *Request()* 함수와 동일합니다.

(xingAPI 설치 폴더\엑셀샘플' 폴더 아래 "[종목검색과HTS연동.xlsm](#)" 엑셀샘플 첫번째 시트 참고)

```
Set XAQuery_t1833 = CreateObject("XA_DataSet.XAQuery")
Call XAQuery_t1833.LoaddFromResFile("WResWt1833.res")
int nReqID = XAQuery_t1807.RequestService("t1833", "D:Wtest.adf")
If nReqID < 0 Then
    MsgBox "전송오류"
End If
```

차트 지표데이터는

HTS '[4201] xing차트1'의 [수식관리자 - 기술적지표] 데이터와 동일한 데이터를 API에서 제공해 드리는 기능으로, **ChartIndex(차트지표데이터조회)** TR과 **RequestService()** 메소드를 이용하여 조회하며, **실서버에서만** 조회 가능합니다.

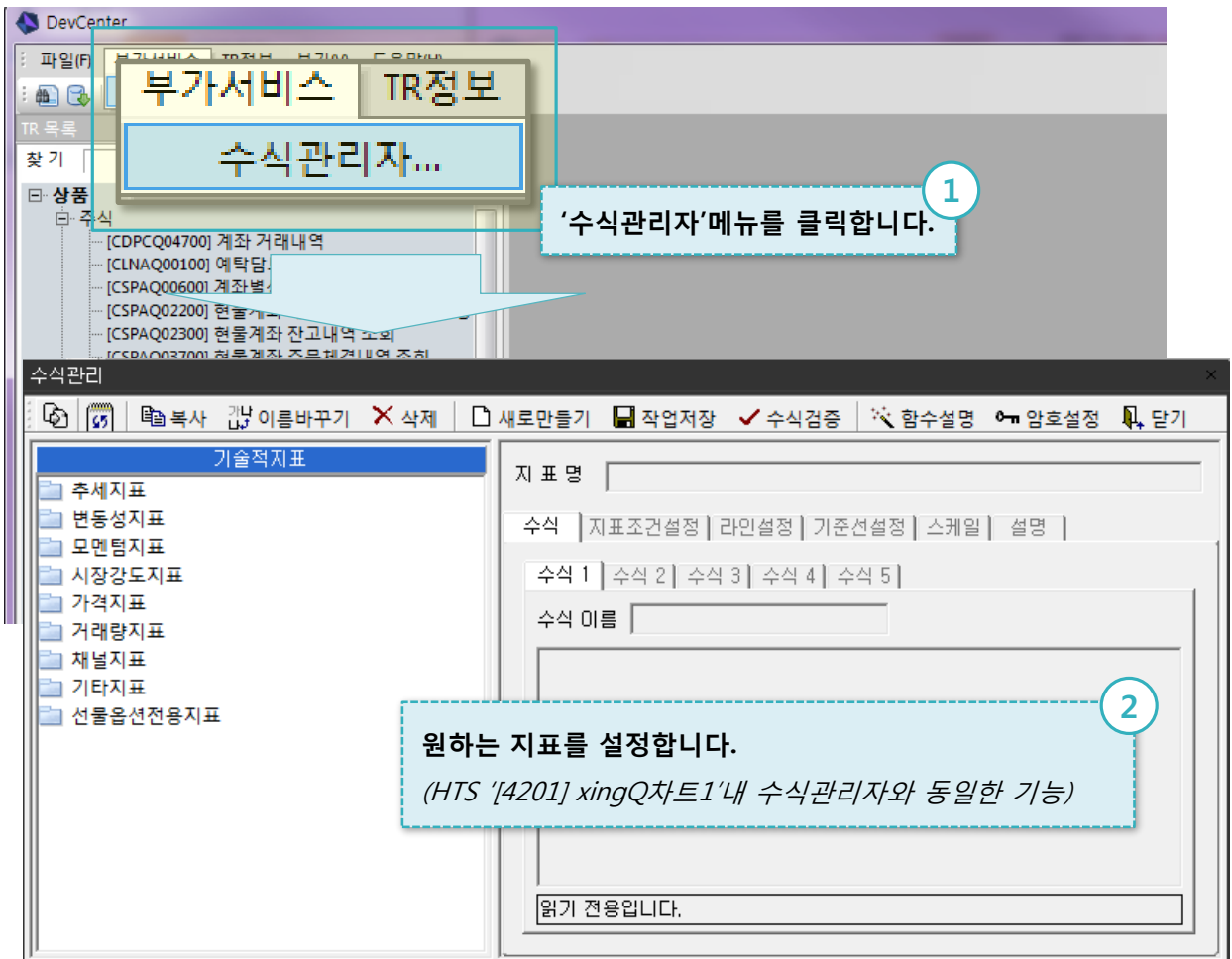
※ 차트 지표데이터는 API내부에서 차트 기초데이터를 가공하여 제공하는 것으로, 조회 및 실시간에 다소 시간이 걸릴 수 있습니다.

아래의 코드는

ChartIndex(차트지표데이터조회) TR을 이용하여 지표를 조회하는 예제입니다.

이것은 xingAPI 설치 폴더 내 "엑셀샘플W 차트지표데이터조회및실시간(ChartIndex).xlsm" 파일에 서도 확인 가능합니다.

Step 1 DevCenter 내 '[부가서비스 - 수식관리자]' 메뉴에서 '수식관리' 화면을 열어 필요한 지표를 설정합니다.



Step 2 XAQuery 객체를 선언합니다. (일반 데이터 조회 과정과 동일)

Visual Basic

1 [개발도구] 메뉴를 클릭합니다.

일반 데이터 조회시의 객체 선언과 동일하며 TR코드만 다른 것입니다.

```

Option Explicit
Dim g_currentpos As Integer
Dim g_indexId As String
Dim WithEvents XAQuery_ChartIndex As XAQuery
  
```

· 표시데이터의 현재 위치
· 지표의 indexId
· 지표데이터조회(API용)

Step 3 XAQuery 객체 생성 및 Res 파일 등록을 합니다. (일반 데이터 조회 과정과 동일)

차트지표데이터조회및실시간(ChartIndex).xslm - Sheet1 (코드)

[일반] [선언]

```

'xingAPI의 지표는 DevCenter내 '수식관리자' 화면에서 설정하며,
'HTS '[4201] xing차트1' 화면의 수식관리자에서 적용한 지표와 동일한 결과를 보여줍니다.

Option Explicit

Dim g_currentpos As Integer
Dim g_indexId As String
Dim WithEvents XAQuery_ChartIndex As XAQuery

' 차트의 지표데이터를 조회합니다
Private Sub btnQuery_Click()

  ' 객체 생성 및 Res 할당
  If XAQuery_ChartIndex Is Nothing Then
    Set XAQuery_ChartIndex = CreateObject("XA_DataSet.XAQuery")
    XAQuery_ChartIndex.ResFileName = "₩res₩ChartIndex.res"
  End If

  ' 객체 생성 및 Res 할당
  If XAQuery_ChartIndex Is Nothing Then
    Set XAQuery_ChartIndex = CreateObject("XA_DataSet.XAQuery")
    XAQuery_ChartIndex.ResFileName = "₩res₩ChartIndex.res"
  End If
  
```

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

Step 4 Inblock의 입력값을 모두 설정합니다. (일반 데이터 조회 과정과 동일)

이름	타입	크기	설명
ChartIndex		268	차트지표데이터조회
ChartIndexInBlock	input	127	In(*EMPTY*)
indexid	long	10	지표ID
indexname	string	40	지표명
indexparam	string	40	지표조건설정
market	string	1	시장구분
period	string	1	주기구분
shcode	string	8	단축코드
qrycnt	long	4	요청건수(최대 500개)
ncnt	long	4	단위(n틱/n분)
sdate	string	8	시작일자
edate	string	8	종료일자
Isamend	string	1	수정주가 반영 여부
Isgab	string	1	갭보정 여부
IsReal	string	1	실시간 데이터수신 자동등록 여부
value5	float	10	지표값5
pos	long	8	위치

```

btnQuery
차트의 지표데이터를 조회합니다
Private Sub btnQuery_Click()
' 이전에 조회한 데이터를 삭제
Dim sRange As String
If g_currentpos <> 0 Then
sRange = "B22" & ":" & CStr(g_currentpos)
Range(sRange).ClearContents
g_currentpos = 0
g_indexid = ""
End If

' 객체 생성 및 Res 할당
If XAQuery_ChartIndex Is Nothing Then
Set XAQuery_ChartIndex = CreateObject("XAQuery_ChartIndex")
XAQuery_ChartIndex.ResFileName = "ResData"
End If

' 이전의 indexid 값이 있다면, 차트지표데이터를 해제
Dim nSuccess As Integer
If Len(g_indexid) > 0 Then
Call XAQuery_ChartIndex.RemoveService("ChartIndex", g_indexid)
End If

' ChartIndex 요청
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "indexname", 0, Range("D9").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "market", 0, Range("D10").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "period", 0, Range("D11").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "shcode", 0, Range("D12").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "qrycnt", 0, Range("D13").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "ncnt", 0, Range("D14").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "sdate", 0, Range("D15").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "edate", 0, Range("D16").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "Isamend", 0, Range("D17").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "Isgab", 0, Range("D18").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "IsReal", 0, Range("D19").Value)

nSuccess = XAQuery_ChartIndex.RequestService("ChartIndex", "")
    
```

입력값을 설정합니다.

```

ChartIndex 요청
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "indexname", 0, Range("D9").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "market", 0, Range("D10").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "period", 0, Range("D11").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "shcode", 0, Range("D12").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "qrycnt", 0, Range("D13").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "ncnt", 0, Range("D14").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "sdate", 0, Range("D15").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "edate", 0, Range("D16").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "Isamend", 0, Range("D17").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "Isgab", 0, Range("D18").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "IsReal", 0, Range("D19").Value)
    
```

Step 5 차트지표 데이터를 요청합니다.

부가서비스용 메소드인 **RequestService() 메소드**를 사용합니다.

Long RequestService(BSTR szCode, BSTR szData)

※ 부가서비스 처리용 메소드로 Request()메소드와 파라미터만 다를 뿐 이용 방법은 유사

szCode : ChartIndex

szData : 사용안함

반환값 : 에러 코드

0 이상이면 API-> 서버로의 요청이 성공,

0 미만이면 API-> 서버로의 요청 실패

ex) "MACD" 지표 데이터 조회

' 일반 조회 TR과 동일하게 "ChartIndex" TR의 Inblock 값을 모두 입력

call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "indexname", 0, "MACD")

.... 생략...

call XAQuery_ChartIndex.RemoveService("ChartIndex", outblock.indexed)

int nReqID = XAQuery_ChartIndex.RequestService("ChartIndex", 0)

```

Private Sub btnQuery_Click()
    ' 차트의 지표데이터를 조회합니다
    ' 이전에 조회한 데이터를 삭제
    Dim sRange As String
    If g_currentpos <> 0 Then
        sRange = "B22" & ":" & CStr(g_currentpos + 22)
        Range(sRange).ClearContents

        g_currentpos = 0
        g_indexid = ""
    End If

    ' 객체 생성 및 Res 할당
    If XAQuery_ChartIndex Is Nothing Then
        Set XAQuery_ChartIndex = CreateObject("XA_DataSet.XAQuery")
        XAQuery_ChartIndex.ResFileName = "res\ChartIndex.res"
    End If

    ' 이전의 indexid 값이 있다면, 차트지표데이터를 해제
    Dim nSuccess As Integer
    If Len(g_indexid) > 0 Then
        Call XAQuery_ChartIndex.RemoveService("ChartIndex", g_indexid)
    End If

    ' ChartIndex 요청
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "indexname", 0, Range("D9").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "market", 0, Range("D10").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "period", 0, Range("D11").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "shcode", 0, Range("D12").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "qrcnt", 0, Range("D13").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "ncnt", 0, Range("D14").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "sdate", 0, Range("D15").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "edate", 0, Range("D16").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "lsamend", 0, Range("D17").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "lsgab", 0, Range("D18").Value)
    Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "lsReal", 0, Range("D19").Value)

    nSuccess = XAQuery_ChartIndex.RequestService("ChartIndex", "")

    If nSuccess < 0 Then
        MsgBox "전송에러 : " & nSuccess
    End If

```

지표명
시장구분 (주식:1, 업종:2, 선물옵션:5)
주기구분 (틱:0, 분:1, 일:2, 주:3, 월:4)
단위 (n틱/n분)
요청건수 (최대 500개)
시작일자 (일/주/월 해당)
시종일자 (일/주/월 해당)
수정주가 반영 여부 (0:반영안함, 1:반영)
값보정 여부 (0:보정안함, 1:보정)
실시간데이터 자동등록 여부 (0:조회만)

Step 6 차트지표 데이터 결과를 수신받습니다. (일반 데이터 조회 과정과 동일)

- ① ChartIndex TR은 API내부에서 차트 '기초데이터 -> 지표데이터' 로 가공된 후 이벤트가 발생합니다.
- ② 이벤트는 XAQuery 객체의 ReceiveData 이벤트를 사용합니다.

ChartIndex TR에 대한 API내부의 지표데이터 가공 처리가 완료되면

ReceiveMessage 이벤트는 지표데이터 요청에 대한 결과를 무조건 수신받습니다.

-> 지표 요청이 성공했는지, 실패했는지 판단

ReceiveData 이벤트는 지표데이터 요청이 성공했을 때, Outblock 의 데이터를 수신받습니다.

-> 지표데이터 요청이 실패했을 때는 수신받지 않습니다.

따라서, 지표 요청이 성공했을 때에만 데이터를 확인시기 바랍니다.

객체 콤보 내의 XAQuery 객체를 선택

1

XAQuery 객체의 이벤트 콤보리스트 중 'ReceiveData' 이벤트를 선택하면

2

차트지표데이터조회및실시간(CharIndex).xlsm - Sheet1 (코드)

XAQuery_ChartIndex

ReceiveMessage
ReceiveChartRealData
ReceiveData
ReceiveMessage

자동으로 이벤트 맵이 추가됩니다.

```
Private Sub XAQuery_ChartIndex_ReceiveData(ByVal szTrCode As String)
    g_indexId = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock", "indexId", 0)

    ' 검색 항목수
    g_currentpos = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock", "rec_cnt", 0)
    If g_currentpos = 0 Then
        Exit Sub
    End If

    ' 데이터 저장
    Dim arrData(10000, 13)
    Dim i, cnt As Integer

    ' 맨 첫번째 데이터에서 지표값 1-5는 각 컬럼의 제목이다
    arrData(0, 0) = "순서"
    arrData(0, 1) = "일자"
    arrData(0, 2) = "시간"
    arrData(0, 3) = "시가"
    arrData(0, 4) = "고가"
    arrData(0, 5) = "저가"
    arrData(0, 6) = "종가"
    arrData(0, 7) = "거래량"
```


Step 6 이벤트를 받았다면 GetFieldData 메소드를 사용하여 수신 데이터를 가져옵니다.
(일반 데이터 조회 과정과 동일)

차트지표데이터조회및실시간(ChartIndex).xslm - Sheet1 (코드)

XAQuery_ChartIndex ReceiveMessage

```

' 차트 지표데이터의 조회 결과 성공시, 결과를 수신받습니다
Private Sub XAQuery_ChartIndex.ReceiveData(ByVal szTrCode As String)
' 지표의 indexid (지표별 구분키로 사용할 수 있다)
g_indexid = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock", "indexid", 0)
' 검색 종목수
g_currentpos = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock", "rec_cnt", 0)
If g_currentpos = 0 Then
Exit Sub
End If
' 데이터 저장
Dim arrData(10000, 13)
Dim i, cnt As Integer
' 맨 첫번째 데이터에서 지표값 1-5는 각 컬럼의 제목이다
arrData(0, 0) = "순서"
arrData(0, 1) = "일자"
arrData(0, 2) = "시간"
arrData(0, 3) = "시가"
arrData(0, 4) = "고가"
arrData(0, 5) = "저가"
arrData(0, 6) = "종가"
arrData(0, 7) = "거래량"
arrData(0, 8) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value1", 0)
arrData(0, 9) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value2", 0)
arrData(0, 10) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value3", 0)
arrData(0, 11) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value4", 0)
arrData(0, 12) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value5", 0)
' 두번째 데이터부터 지표이다
cnt = g_currentpos
i = 1 To cnt - 1
arrData(i, 0) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "pos", cnt - i)
arrData(i, 1) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "date", cnt - i)

```

차트지표데이터조회및실시간(ChartIndex).xslm - Excel

지표 결과를 표시합니다.

ChartIndex - 지표(AP종) TR																																																																																																																																																																																																																																																																					
xingAPI의 지표는 DevCenter의 '수식관리자' 화면에서 설정하며, HTS '4201 xing차트1' 화면의 수식관리자에서 적용한 지표와 동일한 결과를 보여줍니다.																																																																																																																																																																																																																																																																					
지표데이터 조회 조건을 입력한 후, 지표데이터 조회를 누르세요.																																																																																																																																																																																																																																																																					
Inblock	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>지표명</th> <th>가격 이동평균</th> <th>수식관리화면의 한글지표명과 동일해야함</th> </tr> <tr> <td>시장구분</td> <td>1</td> <td>주식, 알종2, 선물옵션5</td> </tr> <tr> <td>주기구분</td> <td>0</td> <td>틱0, 분1, 일2, 주3, 월4</td> </tr> <tr> <td>종목코드</td> <td>000660</td> <td>주식/업종/선종 가능</td> </tr> <tr> <td>요청건수</td> <td>100</td> <td>최대 500개</td> </tr> <tr> <td>단위</td> <td>3</td> <td>틱분 해당(n틱/n분)</td> </tr> <tr> <td>시작일자</td> <td></td> <td>일주일 해당</td> </tr> <tr> <td>종료일자</td> <td></td> <td>일주일 해당</td> </tr> <tr> <td>수정주파</td> <td>1</td> <td>0:반영안함, 1:반영</td> </tr> <tr> <td>검보정</td> <td>1</td> <td>0:보정안함, 1:보정</td> </tr> <tr> <td>실시간자동등록</td> <td>0</td> <td>0:조회만, 1:실시간 자동등록</td> </tr> <tr> <td>조회결과</td> <td>00000</td> <td>조회완료</td> </tr> </table>	지표명	가격 이동평균	수식관리화면의 한글지표명과 동일해야함	시장구분	1	주식, 알종2, 선물옵션5	주기구분	0	틱0, 분1, 일2, 주3, 월4	종목코드	000660	주식/업종/선종 가능	요청건수	100	최대 500개	단위	3	틱분 해당(n틱/n분)	시작일자		일주일 해당	종료일자		일주일 해당	수정주파	1	0:반영안함, 1:반영	검보정	1	0:보정안함, 1:보정	실시간자동등록	0	0:조회만, 1:실시간 자동등록	조회결과	00000	조회완료																																																																																																																																																																																																																																
지표명	가격 이동평균	수식관리화면의 한글지표명과 동일해야함																																																																																																																																																																																																																																																																			
시장구분	1	주식, 알종2, 선물옵션5																																																																																																																																																																																																																																																																			
주기구분	0	틱0, 분1, 일2, 주3, 월4																																																																																																																																																																																																																																																																			
종목코드	000660	주식/업종/선종 가능																																																																																																																																																																																																																																																																			
요청건수	100	최대 500개																																																																																																																																																																																																																																																																			
단위	3	틱분 해당(n틱/n분)																																																																																																																																																																																																																																																																			
시작일자		일주일 해당																																																																																																																																																																																																																																																																			
종료일자		일주일 해당																																																																																																																																																																																																																																																																			
수정주파	1	0:반영안함, 1:반영																																																																																																																																																																																																																																																																			
검보정	1	0:보정안함, 1:보정																																																																																																																																																																																																																																																																			
실시간자동등록	0	0:조회만, 1:실시간 자동등록																																																																																																																																																																																																																																																																			
조회결과	00000	조회완료																																																																																																																																																																																																																																																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>순서</th> <th>일자</th> <th>시간</th> <th>시가</th> <th>고가</th> <th>저가</th> <th>종가</th> <th>거래량</th> <th>0</th> <th>10</th> <th>20</th> <th>60</th> <th>240</th> </tr> </thead> <tbody> <tr><td>100</td><td>20141030</td><td>103006</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>67</td><td>47120</td><td>47120</td><td>47125</td><td>47145</td><td>0</td></tr> <tr><td>99</td><td>20141030</td><td>103001</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>23</td><td>47110</td><td>47120</td><td>47122.5</td><td>47145.83</td><td>0</td></tr> <tr><td>98</td><td>20141030</td><td>103001</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>26</td><td>47110</td><td>47120</td><td>47122.5</td><td>47147.5</td><td>0</td></tr> <tr><td>97</td><td>20141030</td><td>102957</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>131</td><td>47120</td><td>47120</td><td>47122.5</td><td>47149.17</td><td>0</td></tr> <tr><td>96</td><td>20141030</td><td>102955</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>30</td><td>47120</td><td>47125</td><td>47125</td><td>47150.83</td><td>0</td></tr> <tr><td>95</td><td>20141030</td><td>102949</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>542</td><td>47120</td><td>47125</td><td>47125</td><td>47150.83</td><td>0</td></tr> <tr><td>94</td><td>20141030</td><td>102947</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>391</td><td>47130</td><td>47130</td><td>47130</td><td>47152.5</td><td>0</td></tr> <tr><td>93</td><td>20141030</td><td>102941</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>121</td><td>47130</td><td>47135</td><td>47132.5</td><td>47154.17</td><td>0</td></tr> <tr><td>92</td><td>20141030</td><td>102941</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>58</td><td>47120</td><td>47135</td><td>47132.5</td><td>47155.83</td><td>0</td></tr> <tr><td>91</td><td>20141030</td><td>102937</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>26</td><td>47130</td><td>47135</td><td>47135</td><td>47157.5</td><td>0</td></tr> <tr><td>90</td><td>20141030</td><td>102930</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>26</td><td>47130</td><td>47130</td><td>47137.5</td><td>47158.33</td><td>0</td></tr> <tr><td>89</td><td>20141030</td><td>102927</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>51</td><td>47130</td><td>47125</td><td>47137.5</td><td>47160</td><td>0</td></tr> <tr><td>88</td><td>20141030</td><td>102925</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>84</td><td>47140</td><td>47125</td><td>47142.5</td><td>47161.67</td><td>0</td></tr> <tr><td>87</td><td>20141030</td><td>102921</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>112</td><td>47150</td><td>47125</td><td>47145</td><td>47164.17</td><td>0</td></tr> <tr><td>86</td><td>20141030</td><td>102917</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>61</td><td>47140</td><td>47125</td><td>47147.5</td><td>47165.83</td><td>0</td></tr> <tr><td>85</td><td>20141030</td><td>102915</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>3,572</td><td>47130</td><td>47125</td><td>47150</td><td>47167.5</td><td>0</td></tr> <tr><td>84</td><td>20141030</td><td>102915</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>216</td><td>47120</td><td>47130</td><td>47152.5</td><td>47169.17</td><td>0</td></tr> <tr><td>83</td><td>20141030</td><td>102910</td><td>47150</td><td>47150</td><td>47150</td><td>47150</td><td>378</td><td>47110</td><td>47130</td><td>47155</td><td>47170.83</td><td>0</td></tr> <tr><td>82</td><td>20141030</td><td>102909</td><td>47100</td><td>47100</td><td>47100</td><td>47100</td><td>116</td><td>47100</td><td>47130</td><td>47155</td><td>47172.5</td><td>0</td></tr> </tbody> </table>		순서	일자	시간	시가	고가	저가	종가	거래량	0	10	20	60	240	100	20141030	103006	47150	47150	47150	47150	67	47120	47120	47125	47145	0	99	20141030	103001	47100	47100	47100	47100	23	47110	47120	47122.5	47145.83	0	98	20141030	103001	47100	47100	47100	47100	26	47110	47120	47122.5	47147.5	0	97	20141030	102957	47100	47100	47100	47100	131	47120	47120	47122.5	47149.17	0	96	20141030	102955	47150	47150	47150	47150	30	47120	47125	47125	47150.83	0	95	20141030	102949	47100	47100	47100	47100	542	47120	47125	47125	47150.83	0	94	20141030	102947	47100	47100	47100	47100	391	47130	47130	47130	47152.5	0	93	20141030	102941	47150	47150	47150	47150	121	47130	47135	47132.5	47154.17	0	92	20141030	102941	47100	47100	47100	47100	58	47120	47135	47132.5	47155.83	0	91	20141030	102937	47150	47150	47150	47150	26	47130	47135	47135	47157.5	0	90	20141030	102930	47150	47150	47150	47150	26	47130	47130	47137.5	47158.33	0	89	20141030	102927	47100	47100	47100	47100	51	47130	47125	47137.5	47160	0	88	20141030	102925	47100	47100	47100	47100	84	47140	47125	47142.5	47161.67	0	87	20141030	102921	47150	47150	47150	47150	112	47150	47125	47145	47164.17	0	86	20141030	102917	47150	47150	47150	47150	61	47140	47125	47147.5	47165.83	0	85	20141030	102915	47150	47150	47150	47150	3,572	47130	47125	47150	47167.5	0	84	20141030	102915	47150	47150	47150	47150	216	47120	47130	47152.5	47169.17	0	83	20141030	102910	47150	47150	47150	47150	378	47110	47130	47155	47170.83	0	82	20141030	102909	47100	47100	47100	47100	116	47100	47130	47155	47172.5	0
순서	일자	시간	시가	고가	저가	종가	거래량	0	10	20	60	240																																																																																																																																																																																																																																																									
100	20141030	103006	47150	47150	47150	47150	67	47120	47120	47125	47145	0																																																																																																																																																																																																																																																									
99	20141030	103001	47100	47100	47100	47100	23	47110	47120	47122.5	47145.83	0																																																																																																																																																																																																																																																									
98	20141030	103001	47100	47100	47100	47100	26	47110	47120	47122.5	47147.5	0																																																																																																																																																																																																																																																									
97	20141030	102957	47100	47100	47100	47100	131	47120	47120	47122.5	47149.17	0																																																																																																																																																																																																																																																									
96	20141030	102955	47150	47150	47150	47150	30	47120	47125	47125	47150.83	0																																																																																																																																																																																																																																																									
95	20141030	102949	47100	47100	47100	47100	542	47120	47125	47125	47150.83	0																																																																																																																																																																																																																																																									
94	20141030	102947	47100	47100	47100	47100	391	47130	47130	47130	47152.5	0																																																																																																																																																																																																																																																									
93	20141030	102941	47150	47150	47150	47150	121	47130	47135	47132.5	47154.17	0																																																																																																																																																																																																																																																									
92	20141030	102941	47100	47100	47100	47100	58	47120	47135	47132.5	47155.83	0																																																																																																																																																																																																																																																									
91	20141030	102937	47150	47150	47150	47150	26	47130	47135	47135	47157.5	0																																																																																																																																																																																																																																																									
90	20141030	102930	47150	47150	47150	47150	26	47130	47130	47137.5	47158.33	0																																																																																																																																																																																																																																																									
89	20141030	102927	47100	47100	47100	47100	51	47130	47125	47137.5	47160	0																																																																																																																																																																																																																																																									
88	20141030	102925	47100	47100	47100	47100	84	47140	47125	47142.5	47161.67	0																																																																																																																																																																																																																																																									
87	20141030	102921	47150	47150	47150	47150	112	47150	47125	47145	47164.17	0																																																																																																																																																																																																																																																									
86	20141030	102917	47150	47150	47150	47150	61	47140	47125	47147.5	47165.83	0																																																																																																																																																																																																																																																									
85	20141030	102915	47150	47150	47150	47150	3,572	47130	47125	47150	47167.5	0																																																																																																																																																																																																																																																									
84	20141030	102915	47150	47150	47150	47150	216	47120	47130	47152.5	47169.17	0																																																																																																																																																																																																																																																									
83	20141030	102910	47150	47150	47150	47150	378	47110	47130	47155	47170.83	0																																																																																																																																																																																																																																																									
82	20141030	102909	47100	47100	47100	47100	116	47100	47130	47155	47172.5	0																																																																																																																																																																																																																																																									

▶ 전체코드

Option Explicit

```
Dim g_currentpos As Integer           ' 표시데이터의 현재 위치
Dim g_indexId As String              ' 지표의 indexid
Dim WithEvents XAQuery_ChartIndex As XAQuery ' 지표데이터조회(API용)
```

```
'-----
' 차트의 지표데이터를 조회합니다
'-----
```

Private Sub btnQuery_Click()

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

' 이전에 조회한 데이터를 삭제

```
Dim sRange As String
```

```
If g_currentpos <> 0 Then
```

```
    sRange = "B22" & ":" & CStr(g_currentpos + 22)
```

```
    Range(sRange).ClearContents
```

```
    g_currentpos = 0
```

```
    g_indexId = ""
```

```
End If
```

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

' 객체 생성 및 Res 할당

```
If XAQuery_ChartIndex Is Nothing Then
```

```
    Set XAQuery_ChartIndex = CreateObject("XA_DataSet.XAQuery")
```

```
    XAQuery_ChartIndex.ResFileName = "WresWChartIndex.res"
```

```
End If
```

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

' 이전의 indexid 값이 있다면, 차트지표데이터를 해제

```
Dim nSuccess As Integer
```

```
If Len(g_indexId) > 0 Then
```

```
    Call XAQuery_ChartIndex.RemoveService("ChartIndex", g_indexId)
```

```
End If
```

```

'-----
' ChartIndex 요청
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "indexname", 0, Range("D9").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "market", 0, Range("D10").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "period", 0, Range("D11").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "shcode", 0, Range("D12").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "qrycnt", 0, Range("D13").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "ncnt", 0, Range("D14").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "sdate", 0, Range("D15").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "edate", 0, Range("D16").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "Isamend", 0, Range("D17").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "Isgab", 0, Range("D18").Value)
Call XAQuery_ChartIndex.SetFieldData("ChartIndexInBlock", "IsReal", 0, Range("D19").Value)

```

```

nSuccess = XAQuery_ChartIndex.RequestService("ChartIndex", "")
If nSuccess < 0 Then
    MsgBox "전송에러 : " & nSuccess
End If

```

End Sub

```

'-----
' 차트 지표데이터의 조회 결과의 성공여부를 수신받습니다
'-----

```

```

Private Sub XAQuery_ChartIndex_ReceiveMessage(ByVal bIsSystemError As Boolean, ByVal
nMessageCode As String, ByVal szMessage As String)

```

```

    Range("D20").Value = CStr(nMessageCode) & " : " & szMessage

```

End Sub

```

'-----
' 차트 지표데이터의 조회 결과 성공시, 결과를 수신받습니다
'-----

```

```

Private Sub XAQuery_ChartIndex_ReceiveData(ByVal szTrCode As String)

```

```

'-----
' 지표의indexid (지표별 구분키로 사용할 수 있다)
g_indexId = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock", "indexid", 0)

' 검색 종목수
g_currentpos = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock", "rec_cnt", 0)
If g_currentpos = 0 Then
    Exit Sub
End If

'-----
' 데이터 저장
Dim arrData(10000, 13)
Dim i, cnt As Integer

' 맨 첫번째 데이터에서 지표값 1~5는 각 컬럼의 제목이다
arrData(0, 0) = "순서"
arrData(0, 1) = "일자"
arrData(0, 2) = "시간"
arrData(0, 3) = "시가"
arrData(0, 4) = "고가"
arrData(0, 5) = "저가"
arrData(0, 6) = "종가"
arrData(0, 7) = "거래량"
arrData(0, 8) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value1", 0)
arrData(0, 9) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value2", 0)
arrData(0, 10) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value3", 0)
arrData(0, 11) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value4", 0)
arrData(0, 12) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value5", 0)

' 두번째 데이터부터 지표이다
cnt = g_currentpos
For i = 1 To cnt - 1
    arrData(i, 0) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "pos", cnt - i)
    arrData(i, 1) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "date", cnt - i)

```

```

arrData(i, 2) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "time", cnt - i)
arrData(i, 3) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "open", cnt - i)
arrData(i, 4) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "high", cnt - i)
arrData(i, 5) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "low", cnt - i)
arrData(i, 6) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "close", cnt - i)
arrData(i, 7) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "volume", cnt - i)
arrData(i, 8) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value1", cnt - i)
arrData(i, 9) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value2", cnt - i)
arrData(i, 10) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value3", cnt - i)
arrData(i, 11) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value4", cnt - i)
arrData(i, 12) = XAQuery_ChartIndex.GetFieldData("ChartIndexOutBlock1", "value5", cnt - i)

```

Next

'-----

' 위치를 찾아서 데이터를 넣어준다

Dim sRange As String

sRange = "B22" & ":" & CStr(g_currentpos + 22)

Range(sRange) = arrData

End Sub

'-----

' 차트 지표데이터의 실시간 지표를 수신받습니다

'-----

Private Sub XAQuery_ChartIndex_ReceiveChartRealData(ByVal szTrCode As String)

On Error Resume Next

' 실시간 지표데이터의 위치를 확인

Dim pos As Integer

pos = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "pos")

If pos = g_currentpos Then ' 이전 위치에 표시

g_currentpos = pos

' 새로운 지표이면, 맨 위에 추가하고 나머지 데이터를 아래로 한줄 내린다

ElseIf pos > g_currentpos Then

Range("B23:N1022").Copy Destination:=Range("B24:N1023")

```
        g_currentpos = pos
    Else
        Exit Sub
    End If
    '-----
    ' 실시간 지표데이터 가져오기
    Dim arrData(1, 13)

    arrData(0, 0) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "pos")
    arrData(0, 1) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "date")
    arrData(0, 2) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "time")
    arrData(0, 3) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "open")
    arrData(0, 4) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "high")
    arrData(0, 5) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "low")
    arrData(0, 6) = XAQuery_ChartIndex.GetFieldChartRealData("ChartIndexOutBlock1", "close")

End Sub
```

아래의 코드는

xingAPI 설치 시, 설치 폴더 내의 "ChartAPISample.exe" 프로그램의 소스 코드로

xingAPI홈페이지 [자료실 - 일반자료실] "**VC++ 샘플 - ChartAPISample(2014.10) : 차트지표데이터**"
게시글 내 샘플 소스와 동일합니다.

Step 1 차트지표 데이터를 요청합니다.

- ① ChartIndex TR을 ETK_RequestService() 함수를 이용해 요청하며,
- ② 결과를 처리하는 방식은 일반 데이터 조회와 동일합니다.

int ETK_RequestService(HWND hWnd, LPCTSTR pszCode, LPCTSTR pszData)

(부가서비스 처리용 함수 Request()함수와 파라미터만 다를 뿐 이용 방법은 유사)

hwnd : 결과 수신 메시지를 받을 윈도우 핸들

szCode : t1833

szData : HTS '[1807] 종목검색' 화면에서 설정한 검색 조건을 저장한 파일의 전체 경로

반환값 : 에러 코드

0 이상이면 API-> 서버로의 요청이 성공,

0 미만이면 API-> 서버로의 요청 실패

ex) HTS '[1807] 종목검색' 에서 'API 로 내보내기' 저장한 파일이 "D:\wtest.adf" 이라면,

szCode = "t1833", szData = "D:\wtest.adf"

int nReqID = g_XingAPI.RequestService(GetSafeHwnd(), "t1833", "D:\wtest.adf")

※ 오류 발생시, 확인 함수

nErrCode = GetLastError() : API에서의 마지막 에러값을 반환

GetErrorMessage(nErrCode) : 에러 코드에 대한 에러 메시지를 반환

```

//-----
// 차트 지표데이터 조회를 요청합니다
//-----
void CIndexView::OnBtnQuery()
{
    ... 생략...

    ChartIndexInBlock inblock;
    memset(&inblock, 0x20, sizeof(ChartIndexInBlock));
    strcpy(inblock.indexname, m_strIndexName);           // [string, 40] 지표명
    inblock.market[0]= nType+'0';                        // [string, 1] 시장구분
    inblock.period[0]= nPeriod+'0' ;                    // [string, 1] 주기구분
    strcpy(inblock.shcode, strCode);                    // [string, 8] 단축코드
    sprintf(inblock.qrycnt, "%d", nCount);               // [long , 4] 요청건수(최대 500개)
    sprintf(inblock.ncnt, "%d", 1);                     // [long , 4] 단위(n틱/n분)
    strcpy(inblock.edate, strDate);                     // [string, 8] 종료일자(일/주/월 해당)
    inblock.Isamend[0]= '1';                             // [string, 1] 수정주가 반영여부(
    inblock.Isgab[0]= '1';                              // [string, 1] 갭보정 여부
    inblock.IsReal[0]= nReal+'0';                       // [string, 1] 실시간 데이터 자동 등록
    nRQID = g_iXingAPI.RequestService(GetSafeHwnd(), NAME_CHARTINDEX, (LPCTSTR)&inblock);
}

```


Step 2 차트지표 데이터 결과를 수신받습니다.

ETK_RequestService()가 성공하면, 서버로부터 XM_RECEIVE_DATA 메시지를 수신 받아 지표데이터 결과를 확인합니다.

```

BEGIN_MESSAGE_MAP(CIndexView, CFormView)
    ON_MESSAGE(WM_USER + XM_RECEIVE_DATA,          OnXMReceiveData)
    ON_MESSAGE(WM_USER + XM_TIMEOUT_DATA,          OnXMTimeoutData)
    ON_MESSAGE(WM_USER + XM_RECEIVE_CHARTREALDATA,  OnXMReceiveRealData)
END_MESSAGE_MAP()

LRESULT CIndexView::OnXMReceiveData( WPARAM wParam, LPARAM lParam )
{
    //-----
    // Data를 받음 : 지표데이터 요청 성공시에 서버로부터 수신된다.
    //           지표데이터 요청 실패시에 발생하는 경우도 있지만 더미데이터이므로 무시한다
    if( wParam == REQUEST_DATA )
    {
        // ... 필요한 데이터 이용
    }
    //-----
    // Release Data를 받음
    else if( wParam == RELEASE_DATA )
    {
        g_iXingAPI.ReleaseRequestData( (int)lParam );
    }

    return 0L;
}

```

e종목검색은

HTS '[1892] e종목검색'의 API 보내기, 전략관리를 이용한 실시간 종목검색 기능입니다.

기존 파일은 ADF, 신규파일은 ACF 입니다. **기존파일은 실시간 종목검색에서 사용이 불가합니다.**

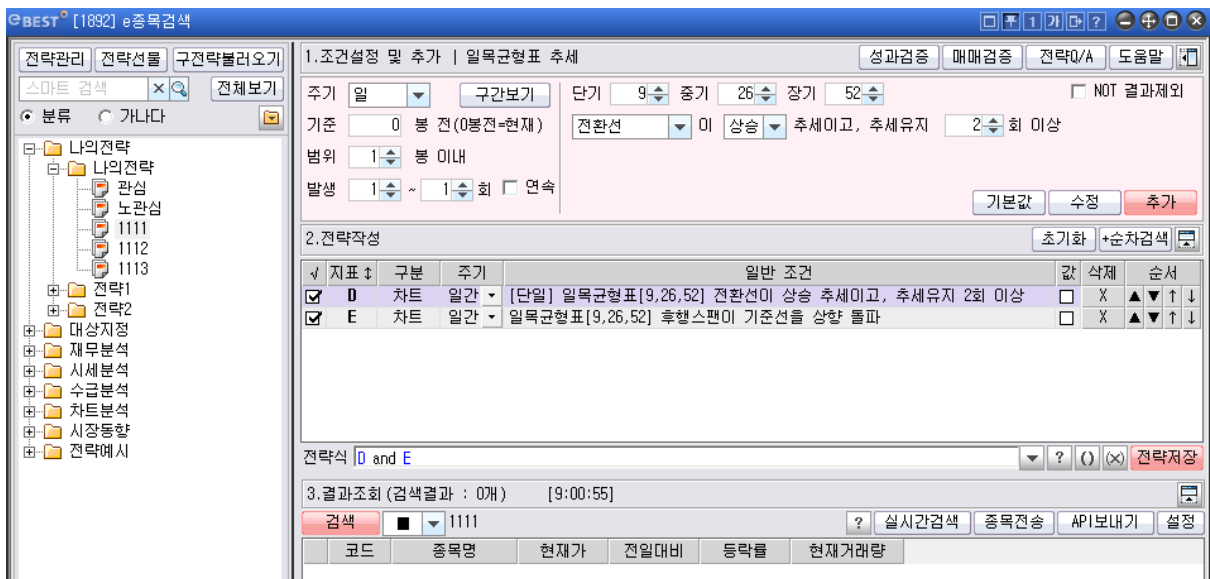
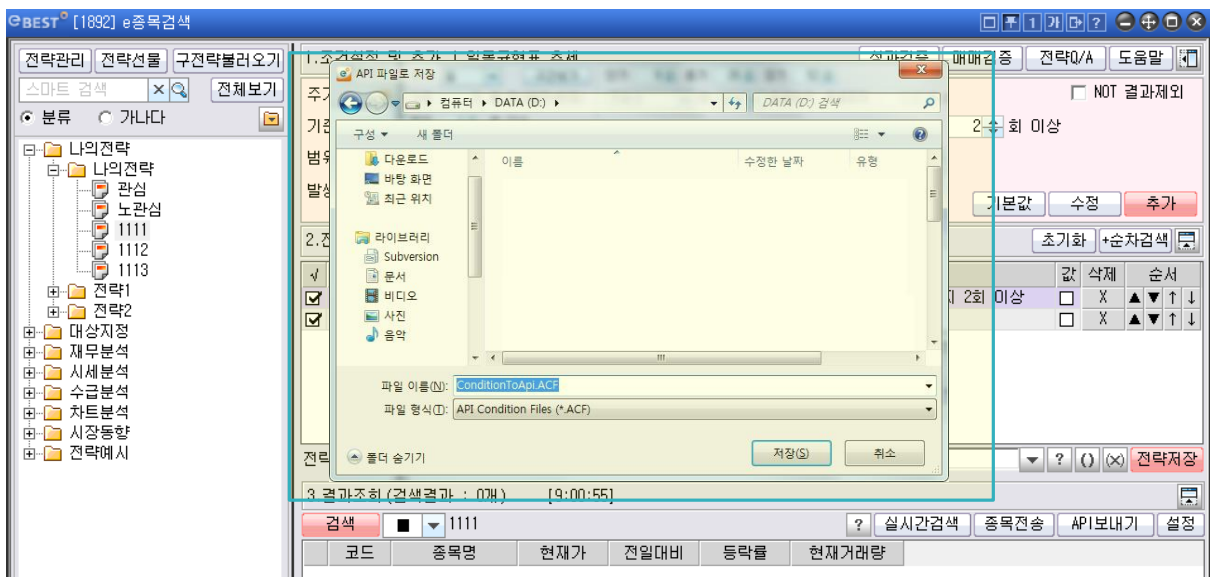
아래의 코드는

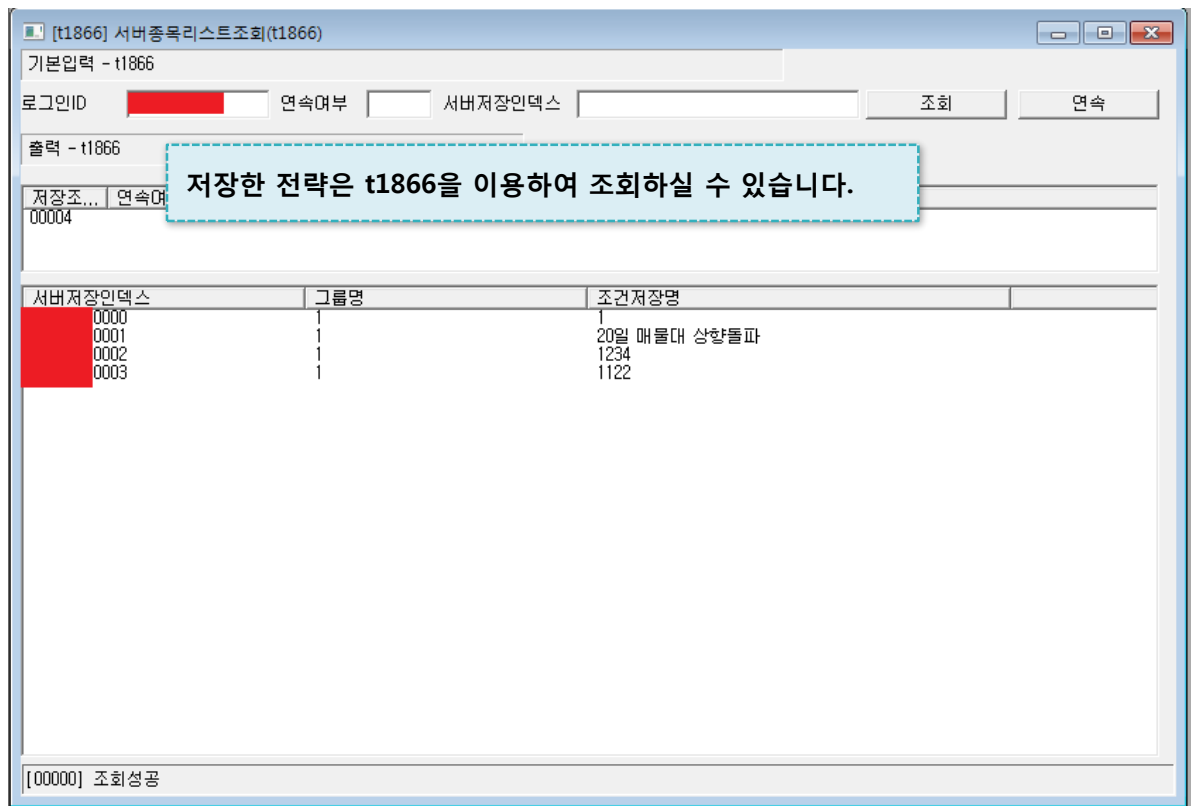
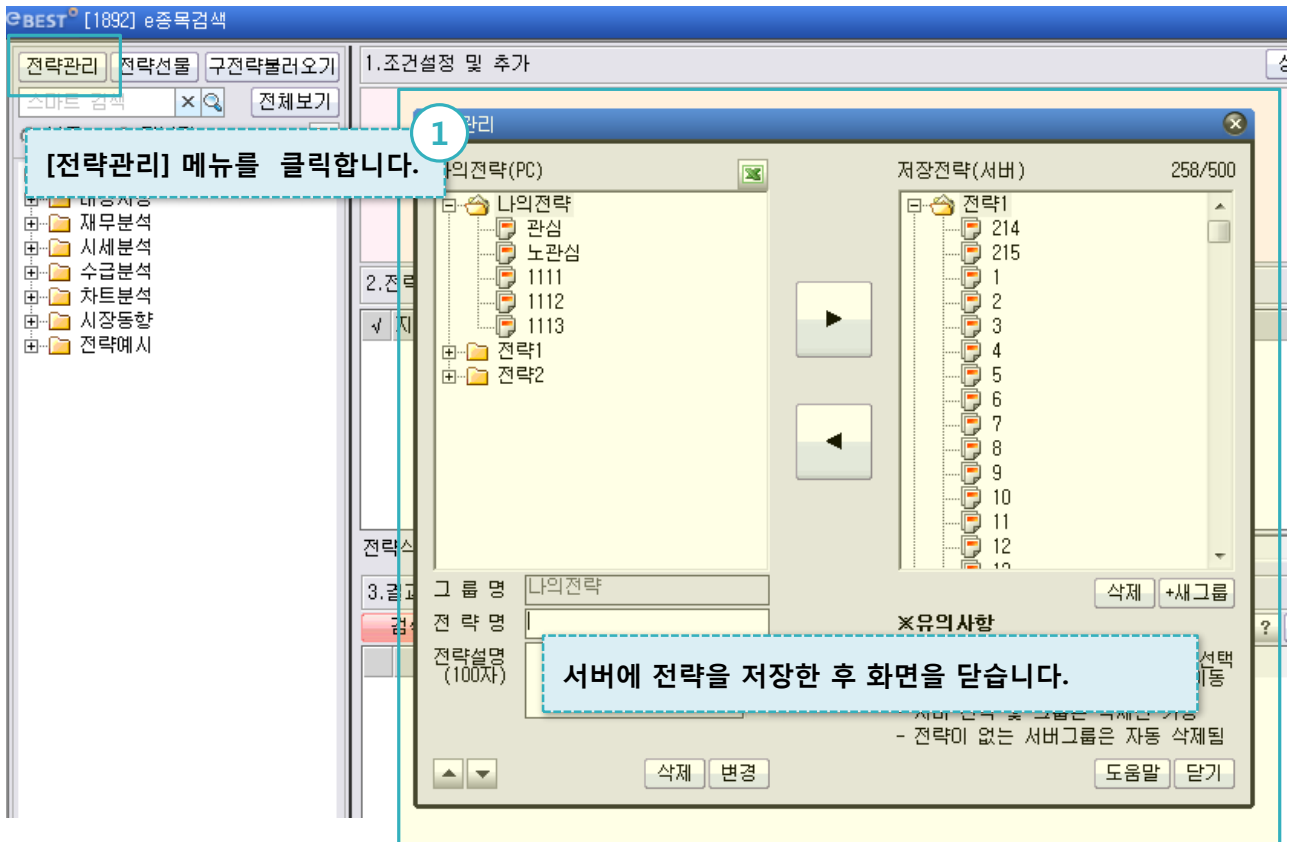
t1857(종목검색(신버전API용)) TR을 이용하여 지표를 조회하는 예제입니다.

이것은 xingAPI 설치 폴더 내 **"엑셀샘플We종목검색(t1857).xlsm"** 파일에서도 확인 가능합니다.

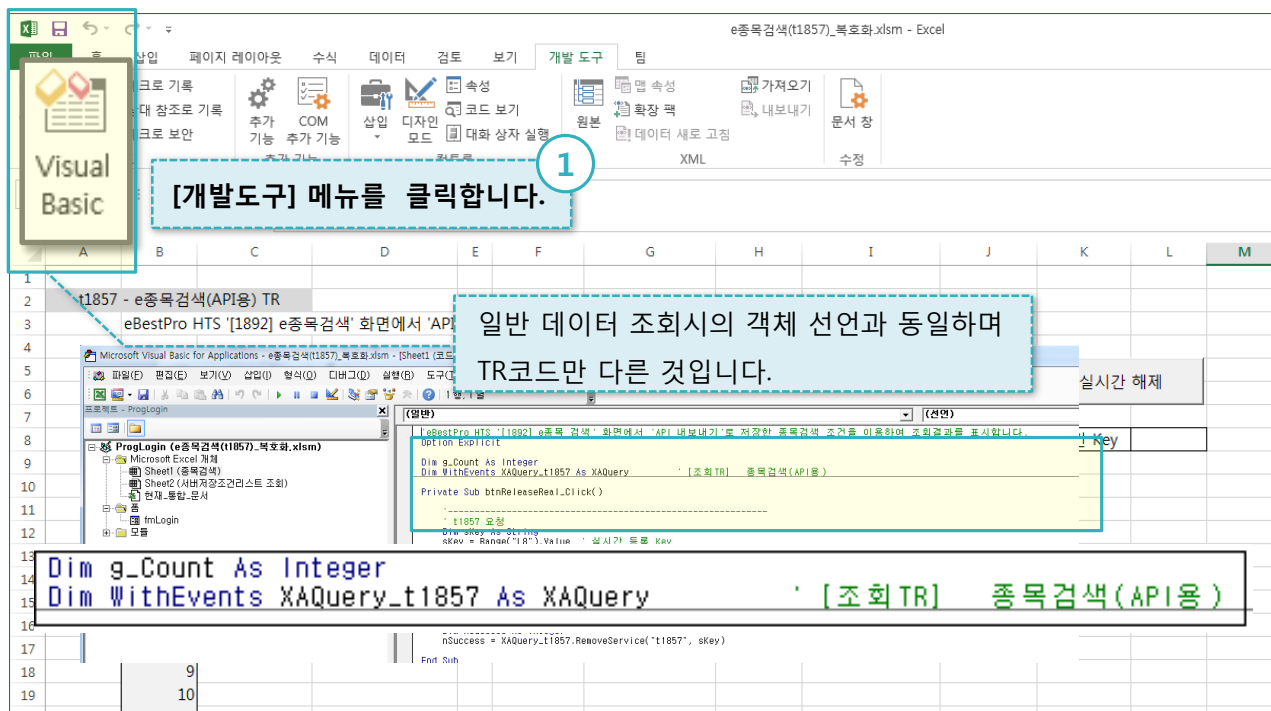
Step 1 HTS [1892]e종목검색 화면에서 검색 조건을 설정합니다.

조건 설정 방법은 HTS 도움말을 참조하시기 바랍니다.

**Step 2-1 'API로 보내기' 버튼을 이용해 파일로 저장합니다.**

Step 2-2 전략관리를 이용해 서버에 전략을 저장합니다.(저장한 전략리스트는 t1866으로 확인가능)


Step 3 XAQuery 객체를 선언합니다. (일반 데이터 조회 과정과 동일)



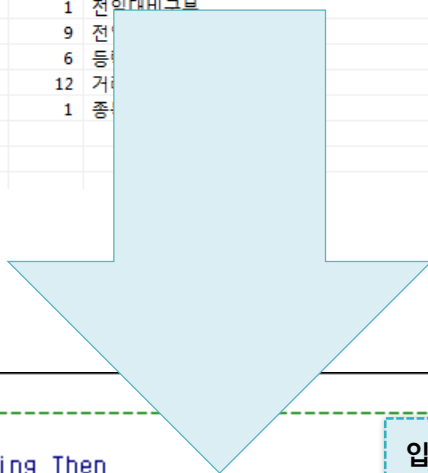
Step 4 XAQuery 객체 생성 및 Res 파일 등록을 합니다. (일반 데이터 조회 과정과 동일)



Step 5 Inblock의 입력값을 모두 설정합니다. (일반 데이터 조회 과정과 동일)

이름	타입	크기	설명
t1857		365	서버저장조건종목검색(API/MTS)
t1857InBlock	input	258	In("EMPTY")
sRealFlag	string	1	실시간구분
sSearchFlag	string	1	종목검색구분
query_index	string	256	종목검색입력값
t1857OutBlock	output	22	Out("EMPTY")

sRealFlag	string	1	실시간구분
sSearchFlag	string	1	종목검색구분
query_index	string	256	종목검색입력값



· 객체 생성 및 Res 할당

```
If XAQuery_t1857 Is Nothing Then
    Set XAQuery_t1857 = CreateObject("XA_DataSet.XAQuery")
    XAQuery_t1857.ResFileName = "₩res₩t1857.res"
End If
```

```
Dim sKey As String
sKey = Range("L8").Value ' 실시간 등록 Key
```

```
Dim nSuc As Integer
nSuc = XAQuery_t1857.RemoveService("t1857", sKey)
```

· t1857 요청

```
Call XAQuery_t1857.SetFieldData("t1857InBlock", "sRealFlag", 0, Range("G8").Value)
Call XAQuery_t1857.SetFieldData("t1857InBlock", "sSearchFlag", 0, Range("J8").Value)
Call XAQuery_t1857.SetFieldData("t1857InBlock", "query_index", 0, Range("C8").Value)
Dim nSuccess As Integer
nSuccess = XAQuery_t1857.RequestService("t1857", "")
If nSuccess < 0 Then
    MsgBox "전송에러 : " & nSuccess
End If
```

입력값을 설정합니다.

Step 6 e종목검색 데이터를 요청합니다.

부가서비스용 메소드인 **RequestService() 메소드**를 사용합니다.

Long RequestService(BSTR szCode, BSTR szData)

※ 부가서비스 처리용 메소드로 Request()메소드와 파라미터만 다를 뿐 이용 방법은 유사

szCode : ChartIndex

szData : 사용안함

반환값 : 에러 코드

0 이상이면 API-> 서버로의 요청이 성공,

0 미만이면 API-> 서버로의 요청 실패

ex) ' 일반 조회 TR과 동일하게 "t1857" TR의 Inblock 값을 모두 입력
Call XAQuery_t1857.SetFieldData("t1857InBlock", "sRealFlag", 0, Range("G8").Value)
.... 생략...
int nReqID = XAQuery_t1857.RequestService("t1857", "")



Step 7 e종목검색 데이터 결과를 수신받습니다. (일반 데이터 조회 과정과 동일)

① 이벤트는 XAQuery 객체의 ReceiveData 이벤트를 사용합니다.

1

객체 콤보 내의 XAQuery 객체를 선택

2

XAQuery 객체의 이벤트 콤보리스트 중 'ReceiveData' 이벤트를 선택하면

3

자동으로 이벤트 맵이 추가됩니다.

```

XAQuery_t1857
Private Sub btnSearch_Click()
    ' 이전에 조회한 데이터를 삭제
    Dim sRange As String
    If g_Count <= 0 Then
        sRange = "B10" & ":" & J & CStr(g_Count + 10)
        Range(sRange).ClearContents
    End If

    ' t1857 요청
    Dim sFile As String
    sFile = Range("C8").Value ' 파일명 (전체 경로 포함)

    ' 객체 생성 및 Res 할당
    If XAQuery_t1857 Is Nothing Then
        Set XAQuery_t1857 = CreateObject("XA_DataSet.XAQuery")
        XAQuery_t1857.ResFileName = "wresw1857.res"
    End If

    Dim sKey As String
    sKey = Range("L8").Value ' 실시간 등록 Key

    Dim nSuc As Integer
    nSuc = XAQuery_t1857.RemoveService("t1857", sKey)

    ' t1857 요청
    Call XAQuery_t1857.SetFieldData("t1857InBlock", "sRealFlag", 0, Range("G8").Value)
    Call XAQuery_t1857.SetFieldData("t1857InBlock", "sSearchFlag", 0, Range("J8").Value)
    Call XAQuery_t1857.SetFieldData("t1857InBlock", "query_index", 0, Range("C8").Value)
    Dim nSuccess As Integer
    nSuccess = XAQuery_t1857.RequestService("t1857", "")
    If nSuccess < 0 Then

Private Sub XAQuery_t1857_ReceiveData(ByVal szTrCode As String)
    End Sub
    Private Sub XAQuery_t1857_ReceiveData(ByVal szTrCode As String)
    
```

Step 8 이벤트를 받았다면 GetFieldData 메소드를 사용하여 수신 데이터를 가져옵니다.
(일반 데이터 조회 과정과 동일)

```

XAQuery_t1857
ReceiveData

Private Sub XAQuery_t1857_ReceiveData(ByVal szTrCode As String)

On Error Resume Next

' 검색 종목수
g_Count = XAQuery_t1857.GetFieldData("t1857OutBlock", "result_count", 0)

Range("L8") = XAQuery_t1857.GetFieldData("t1857OutBlock", "AlertNum", 0)
Dim arrData(10000, 9)

If g_Count = 0 Then
Exit Sub
End If

' 데이터 저장
Dim i As Integer
For i = 0 To g_Count - 1
arrData(i, 0) = i + 1
arrData(i, 1) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "shcode", i)
arrData(i, 2) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "hname", i)
arrData(i, 3) = GetSignPic(XAQuery_t1857.GetFieldData("t1857OutBlock1", "sign", i))
arrData(i, 4) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "change", i)
arrData(i, 5) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "price", i)
arrData(i, 6) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "diff", i)
arrData(i, 7) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "volume", i)
arrData(i, 8) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "JobFlag", i)
Next

' 위치를 찾아서 데이터를 넣어준다
Dim sRange As String
sRange = "B10" & ":" & CStr(g_Count + 10)
Range(sRange) = arrData

End Sub

```

종목코드
종목명
종목영역
전일대비구분
전일대비
현재가
등락률
거래량
연속등수

지표 결과를 표시합니다.

종목코드	종목명	구분	전일대비	현재가	등락률	거래량	전일여부
1	000660	SK하이닉스	▲ 120000	7,890,000	1.54	123,407	N
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							

▶ 전체코드

Option Explicit

Dim g_Count As Integer

Dim WithEvents XAQuery_t1857 As XAQuery -'[조회]TR] 종목검색(API용)

'-----

' 실시간 해제 요청

Private Sub btnReleaseReal_Click()

'-----

' t1857 요청

Dim sKey As String

sKey = Range("L8").Value ' 실시간 등록 Key

'-----

' 객체 생성 및 Res 할당

If XAQuery_t1857 Is Nothing Then

Set XAQuery_t1857 = CreateObject("XA_DataSet.XAQuery")

XAQuery_t1857.ResFileName = "WresWt1857.res"

End If

Dim nSuccess As Integer

nSuccess = XAQuery_t1857.RemoveService("t1857", sKey)

End Sub

'-----

' 종목검색 요청

Private Sub btnSearch_Click()

'-----

' 이전에 조회한 데이터를 삭제

Dim sRange As String

If g_Count <> 0 Then

sRange = "B10" & ":" & CStr(g_Count + 10)

Range(sRange).ClearContents

End If

```

'-----
' t1857 요청
Dim sFile As String
sFile = Range("C8").Value ' 파일명 (전체 경로 포함)

'-----
' 객체 생성 및 Res 할당
If XAQuery_t1857 Is Nothing Then
    Set XAQuery_t1857 = CreateObject("XA_DataSet.XAQuery")
    XAQuery_t1857.ResFileName = "WresWt1857.res"
End If

Dim sKey As String
sKey = Range("L8").Value ' 실시간 등록 Key

Dim nSuc As Integer
nSuc = XAQuery_t1857.RemoveService("t1857", sKey)

'-----
' t1857 요청

Call XAQuery_t1857.SetFieldData("t1857InBlock", "sRealFlag", 0, Range("G8").Value) '
Call XAQuery_t1857.SetFieldData("t1857InBlock", "sSearchFlag", 0, Range("J8").Value) '
Call XAQuery_t1857.SetFieldData("t1857InBlock", "query_index", 0, Range("C8").Value) '
Dim nSuccess As Integer
nSuccess = XAQuery_t1857.RequestService("t1857", "")
If nSuccess < 0 Then
    MsgBox "전송에러 : " & nSuccess
End If
'-----

End Sub

Private Sub XAQuery_t1857_ReceiveData(ByVal szTrCode As String)

```

On Error Resume Next

' -----

' 검색 종목수

g_Count = XAQuery_t1857.GetFieldData("t1857OutBlock", "result_count", 0)

Range("L8") = XAQuery_t1857.GetFieldData("t1857OutBlock", "AlertNum", 0)

Dim arrData(10000, 9)

If g_Count = 0 Then

Exit Sub

End If

' -----

' 데이터 저장

Dim i As Integer

For i = 0 To g_Count - 1

arrData(i, 0) = i + 1

arrData(i, 1) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "shcode", i)

' 종목코드

arrData(i, 2) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "hname", i)

' 종목명

arrData(i, 3) = GetSignPic(XAQuery_t1857.GetFieldData("t1857OutBlock1", "sign", i))

' 전일대비구분

arrData(i, 4) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "change", i)

' 전일대비

arrData(i, 5) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "price", i)

' 현재가

arrData(i, 6) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "diff", i)

' 등락율

arrData(i, 7) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "volume", i)

' 거래량

arrData(i, 8) = XAQuery_t1857.GetFieldData("t1857OutBlock1", "JobFlag", i)

' 진입여부

Next

' -----

' 위치를 찾아서 데이터를 넣어준다

Dim sRange As String

sRange = "B10" & ":" & "J" & CStr(g_Count + 10)

Range(sRange) = arrData

End Sub

```

Private Sub XAQuery_t1857_ReceiveSearchRealData(ByVal szTrCode As String)
    On Error Resume Next
    Dim pos As Integer
    pos = g_Count

    If pos >= 0 Then
        Range("C10:J10009").Copy Destination:=Range("C11:J10010")
    Else
        Exit Sub
    End If

    If pos >= 10000 Then
        g_Count = 10000
    ElseIf pos >= 0 Then
        g_Count = pos + 1
    End If

    Dim arrData(1, 9)
    arrData(0, 0) = 1
    arrData(0, 1) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "shcode") ' 종목코드
    arrData(0, 2) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "hname") ' 종목명
    arrData(0, 3) = GetSignPic(XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "sign"))
    ' 전일대비구분
    arrData(0, 4) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "change") ' 전일대비
    arrData(0, 5) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "price") ' 현재가
    arrData(0, 6) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "diff") ' 등락율
    arrData(0, 7) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "volume") ' 거래량
    arrData(0, 8) = XAQuery_t1857.GetFieldSearchRealData("t1857OutBlock1", "JobFlag") ' 진입여부

    ' -----
    ' 위치를 찾아서 데이터를 넣어준다
    Dim sRange As String
    sRange = "B10" & ":J10"
    Range(sRange) = arrData

End Sub

```

아래의 코드는

xingAPI 설치 시, 설치 폴더 내의 "xingAPI_Sample_eSearch.exe" 프로그램의 소스 코드로 xingAPI홈페이지 [자료실 - 일반자료실] "**VC++ 샘플 - xingAPI_Sample_eSearch(2018.02) : 실시간 종목검색**" 게시물 내 샘플 소스와 동일합니다.

Step 1 종목검색을 요청합니다.

- ① t1857 TR을 ETK_RequestService() 함수를 이용해 요청하며,
- ② 결과를 처리하는 방식은 일반 데이터 조회와 동일합니다.

```
int ETK_RequestService( HWND hWnd, LPCTSTR pszCode, LPCTSTR pszData )
```

(부가서비스 처리용 함수 Request()함수와 파라미터만 다를 뿐 이용 방법은 유사)

hWnd : 결과 수신 메시지를 받을 윈도우 핸들

szCode : t1857

szData : t1857InBlock 데이터

반환값 : 에러 코드

0 이상이면 API-> 서버로의 요청이 성공,

0 미만이면 API-> 서버로의 요청 실패

ex) HTS ' [1892] e종목검색' 에서 'API 로 내보내기' 저장한 파일이 "D:\Wtest.acf" 이라면,

szCode = "t1857", szData = "D:\Wtest.acf"

szFlag = "F", szReal = "1"

```
SetPacketData( pckInBlock.sRealFlag , sizeof( pckInBlock.sRealFlag), szReal, DATA_TYPE_STRING );
```

```
SetPacketData(pckInBlock.sSearchFlag,sizeof( pckInBlock.sSearchFlag),szFlag,DATA_TYPE_STRING );
```

```
SetPacketData( pckInBlock.query_index,sizeof( pckInBlock.query_index),szData,DATA_TYPE_STRING );
```

```
int nRqID = g_iXingAPI.RequestService( GetSafeHwnd(), szCode, (LPCTSTR)&pckInBlock )
```

※ 오류 발생시, 확인 함수

nErrCode = GetLastError() : API에서의 마지막 에러값을 반환

GetErrorMessage(nErrCode) : 에러 코드에 대한 에러 메시지를 반환

```

//-----
// 데이터 조회
void CDlg_t1857::RequestData( BOOL bNext )
{
    t1857InBlock    pckInBlock;
    TCHAR           szTrNo[]      = "t1857";
    char            szNextKey[]    = "";
//-----
// 데이터 취득
CString str_Index; GetDlgItemText( IDC_INBLOCK_INDEX1857, str_Index );// 종목검색입력값
CString str_Flag; m_combo_Flag.GetLBText(m_combo_Flag.GetCurSel(),str_Flag);
str_Flag = str_Flag.Left(1);      // 조회구분값 S:서버 F:파일
CString str_Real; m_combo_Real.GetLBText(m_combo_Real.GetCurSel(),str_Real);
str_Real = str_Real.Left(1);      // 실시간 여부 1:등록 0:조회만
FillMemory( &pckInBlock, sizeof( pckInBlock ), ' ' );          // 입력값 초기화
//-----
// 데이터 입력
SetPacketData( pckInBlock.sRealFlag, sizeof( pckInBlock.sRealFlag), str_Real , DATA_TYPE_STRING );
SetPacketData(pckInBlock.sSearchFlag,sizeof(pckInBlock.sSearchFlag),str_Flag ,DATA_TYPE_STRING );
SetPacketData(pckInBlock.query_index,sizeof(pckInBlock.query_index),str_Index,DATA_TYPE_STRING);
// 종목검색입력값

//-----
// 데이터 전송
int nRqID = g_iXingAPI.RequestService( GetSafeHwnd(),szTrNo,(LPCTSTR)&pckInBlock);
if( bNext == FALSE )
{ m_ctrlOutBlock1.DeleteAllItems(); }          // 기존데이터 삭제
//-----
// Request ID가 0보다 작을 경우에는 에러이다.
if( nRqID < 0 )
{ MessageBox( "조회실패", "에러", MB_ICONSTOP ); }
}

```

Step 2 종목검색 결과를 수신받습니다.

ETK_RequestService()가 성공하면, 서버로부터 XM_RECEIVE_DATA 메시지를 수신 받아 종목검색 결과를 확인합니다.

```

BEGIN_MESSAGE_MAP(CDlg_t1857, CDialog)
    ON_MESSAGE(WM_USER + XM_RECEIVE_DATA,          OnXMReceiveData)
    ON_MESSAGE(WM_USER + XM_TIMEOUT_DATA,          OnXMTimeoutData)
    ON_MESSAGE(WM_USER + XM_RECEIVE_REAL_DATA_SEARCH, OnXMReceiveRealSearchData)
END_MESSAGE_MAP()

LRESULT CDlg_t1857::OnXMReceiveData( WPARAM wParam, LPARAM lParam )
{
    //-----
    // Data를 받음 : 지표데이터 요청 성공시에 서버로부터 수신된다.
    //           지표데이터 요청 실패시에 발생하는 경우도 있지만 더미데이터이므로 무시한다
    if( wParam == REQUEST_DATA )
    {
        // ... 필요한 데이터 이용
    }
    //-----
    // Release Data를 받음
    else if( wParam == RELEASE_DATA )
    {
        g_iXingAPI.ReleaseRequestData( (int)lParam );
    }

    return 0L;
}

```

Step 3 종목검색 실시간을 수신받습니다.

실시간 데이터가 발생하면, 서버로부터 XM_RECEIVE_REAL_DATA_SEARCH 메시지를 수신 받아 종목검색 신호를 확인합니다.

```

BEGIN_MESSAGE_MAP(CDlg_t1857, CDialog)
    ON_MESSAGE(WM_USER + XM_RECEIVE_DATA,          OnXMReceiveData)
    ON_MESSAGE(WM_USER + XM_TIMEOUT_DATA,          OnXMTimeoutData)
    ON_MESSAGE(WM_USER + XM_RECEIVE_REAL_DATA_SEARCH, OnXMReceiveRealSearchData)
END_MESSAGE_MAP()

LRESULT CDlg_t1857:: OnXMReceiveRealSearchData( WPARAM wParam, LPARAM lParam )
{
    LPRECV_REAL_PACKET pRealPacket = (LPRECV_REAL_PACKET)lParam;

    LPt1857OutBlock1 pOutBlock = (LPt1857OutBlock1)pRealPacket->pszData;
    int nRowCount = m_ctrlOutBlock1.GetItemCount();
    m_ctrlOutBlock1.InsertItem( 0, "" );
    m_ctrlOutBlock1.SetItemText( 0, 0, GetDispData( pOutBlock->shcode,
        sizeof( pOutBlock->shcode ), DATA_TYPE_STRING ) );    // 종목코드
    .....
    CString strJobFlag;
    strJobFlag = GetDispData( pOutBlock->JobFlag ,
        sizeof( pOutBlock->JobFlag ), DATA_TYPE_STRING );
        if(strcmp( strJobFlag, "N" ) == 0)          strJobFlag = "신규";
        else if(strcmp( strJobFlag, "R" ) == 0)      strJobFlag = "재진입";
        else if(strcmp( strJobFlag, "O" ) == 0)      strJobFlag = "탈퇴";
        m_ctrlOutBlock1.SetItemText( 0, 7, strJobFlag );          // 종목상태

    return 0L;
}

```


감사합니다