Coded by Nico
As of May 7, 2013

```
'------------------------------------------------------------------------------------------------------
'------------------------------------------------------------------------------------------------------
Option Explicit     'Requires that all variables to be declared explicitly.
Option Base 1       'The "Option Base" statement allows to specify 0 or 1 as the
                    'default first index of arrays.
'------------------------------------------------------------------------------------------------------
'------------------------------------------------------------------------------------------------------
Private PUB_WEB_DATA_ELEMENTS_HASH As clsTypeHash
'This hash table is used to store the company stock symbol and the index of financial data as one element.
'It is used in RETRIEVE_WEB_DATA_ELEMENT_FUNC to check if the requested financial data of the associated
'company already existed. If so, return the existing element; otherwise, proceed on collecting data based
'on the ticker symbol.

Public PUB_WEB_DATA_PAGES_HASH As clsTypeHash
'The advantage of using this hash table over the array is speed. You don't need to loop through all the URLs
'to find the web page that you want.

Private PUB_WEB_DATA_RECORDS_HASH As clsTypeHash
'The advantage to using this hash is the increase in speed. If this hash table was not used, for each ticker
'it would need to go through each element. Being able to have this in a hash table drastically increases its
'speed. The .exists method is used to find whether or not that key can be found in the hash table so that a
'new element can be created using that key.

'------------------------------------------------------------------------------------------------------
Public PUB_WEB_DATA_TABLES_FLAG As Boolean
'The Boolean variable PUB_WEB_DATA_TABLES_FLAG is used in order to determine if the hash tables have already
'been instantiated or not. If the variable is set to false, they have not been created and START_WEB_DATA_SYSTEM_FUNC
'is called in order to do so. The PUB_WEB_DATA_TABLES_FLAG is set to True when this function is called in order
'to indicate that they have been created.
'------------------------------------------------------------------------------------------------------
Private Const PUB_WEB_DATA_FILES_VAL As Long = 9 ' Number of external files with element definitions
'The constant PUB_WEB_DATA_FILES_VAL is set equal to 9 because there are 9 different text files being used. The
'files are saved in such a way that the file path is easy to create by simply looping to change the number at
'the end of the string from 1 to 9. The files and their descriptions are located below:
```

Coded by Nico
As of May 7, 2013

```
'smf-elements-0.txt = Calculated data elements
'smf-elements-1.txt = MSN data elements
'smf-elements-2.txt = Yahoo data elements
'smf-elements-3.txt = Google data elements
'smf-elements-4.txt = Morningstar data elements
'smf-elements-5.txt = Reuters data elements
'smf-elements-6.txt = Zacks data elements
'smf-elements-7.txt = AdvFN data elements
'smf-elements-8.txt = Earnings.com data elements
'smf-elements-9.txt = Other misc data elements

Private Const PUB_WEB_DATA_RECORDS_VAL As Long = 20000 ' Extraction parameters for each element
'The limit for this is 20,000 since anything over that wouldn't exist. There are only around 12,000 unique elements
'with numbers ranging from 1 to 17,006. The reason for such a large number is there are so many combinations of
'source/URL and data to be required. Say there were 43 sources. That would mean for each source/URL, there would be
'an average of 20,000/43 = 463 elements per source.

Private Const PUB_WEB_DATA_ELEMENTS_VAL As Long = 100000 ' Number of data elements
'There is a maximum of 20,000 entries in PUB_WEB_DATA_RECORDS_VAL; for every company the user trying to analyze,
'there will be a distinct 20,000 entries for web-data-element. With a maximum of 100,000, it is assumed to be never
'reached, since it is very unlikely that user will make more than 100000 requests of different pieces of data
'during one session.

Private Const PUB_WEB_DATA_PAGES_VAL As Long = 30000 ' Number of data pages to save
'The PUB_WEB_DATA_PAGES_VAL is used to defined the row size of the PUB_WEB_DATA_PAGES_MATRIX. PUB_WEB_DATA_PAGES_MATRIX
'is used in the function SAVE_WEB_DATA_PAGE_FUNC to store URLs where the user/other functions try to retrieve data
'from. The SAVE_WEB_DATA_PAGE_FUNC function is called by functions to retrieve web-data-elements, web-data-cells,
'web-data-tables, and web-data-pages. We know there are maximum of 20,000 web-data-elements and 10 web-data-pages.
'Web-data-cells and web-data-tables are called less frequently as they are only used for analysis purposes.

'Also, some of the sources where the user/other function retrieve web-data-cells and web-data-tables will be the same as
'the web-data-elements. Therefore, a good estimation of the number of different URLs for retrieving web-data-cells and
'web-data-tables is 10,000 which result the PUB_WEB_DATA_PAGES_VAL with a maximum of 30,000.

Private PUB_WEB_DATA_PAGES_OBJ As Collection
'Private PUB_WEB_DATA_PAGES_INDEX_VAL As Long
```

Coded by Nico
As of May 7, 2013

```
'Private PUB_WEB_DATA_PAGES_URL_ARR(1 To PUB_WEB_DATA_PAGES_VAL) As String
'Private PUB_WEB_DATA_PAGES_ARR(1 To PUB_WEB_DATA_PAGES_VAL) As String
Private PUB_WEB_DATA_PAGES_MATRIX(1 To PUB_WEB_DATA_PAGES_VAL, 1 To 2) As String ' Saved web page data (2) and its ticker-source (1)
'From a general perspective, the loop in Case 0 @ SAVE_WEB_DATA_PAGE_FUNC should be much slower than Case Else. This stems from the fact
'that the loop in Case 0 loops through an array and at the first empty position tries to download the web page. If the web page has been
'previously loaded, the array would contain HTTP_TYPE & ":" & SRC_URL_STR in the first column. Case Else uses the same key string as the
'key for a collection. If the collection doesn't contain the key, it downloads the web page and adds it to the collection.

'Through testing, the array took 745 milliseconds, while the collection took 637 milliseconds. These numbers will be substantially
'different the more web pages that are loaded.

'----------------------------------------------------------------------------------------------------------
Private Const PUB_ADVFN_SERVER_STR As String = "ca"
Private Const PUB_ADVFN_URL_STR As String = ".advfn.com/p.php?pid=financials"
'----------------------------------------------------------------------------------------------------------
Private Const PUB_WEB_DATA_VERSION_STR As String = "2.25.2013" 'Version number of add-in
Private Const PUB_WEB_DATA_FILES_PATH_STR As String = "https://raw.github.com/rnfermincota/BGCVI/master/WDS/smf-elements/smf-elements-
"
'"C:\Documents and Settings\HOME\Application Data\Microsoft\AddIns\smf-elements-"

'Using this web address allows for a centralized source of data elements. When using multiple instances of this library, all functions can
'load the same elements. The centralized nature of the text files allows for standardization and easy program maintenance.

Private Const PUB_WEB_DATA_ELEMENT_LOOK_STR As String = "~~~~~"
'PUB_WEB_DATA_ELEMENT_LOOK_STR acts as a placeholder for a ticker symbol. On line SRC_URL_STR = Replace(PARAM_RNG(2),
'PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER2_STR) @ RETRIEVE_WEB_DATA_ELEMENT_FUNC the placeholder is being replaced with the actual
'ticker symbol.

Private Const PUB_WEB_DATA_ELEMENT_DELIM_STR As String = ";"
'The delimiter character is what separates all of the fields for the elements. For the elements, they are organized as:
'#;source;element;url;cells;find1;find2;find3;find4;rows;end;look;type

'Specifically, on line PARAM_RNG = Split(PARAM_RNG & CASES_STR, PUB_WEB_DATA_ELEMENT_DELIM_STR) @
RETRIEVE_WEB_DATA_ELEMENT_FUNC , each
'string is split by the delimiter character to separate each field of the element. The fields are then stored in PARA_RNG which is
'used further in data segmentation of each record.
```

```
'----------------------------------------------------------------------------------------------------------
Public Const PUB_WEB_DATA_SYSTEM_ERROR_STR As String = "Error" ' Value to return if error
'The global error label allows for a standardized error message. The reason for the standardized error message is in the
'SAVE_WEB_DATA_PAGE_FUNC. If the function encounters an error saving the web page, it will return an error. That way, you can
'check anywhere else in the library if the SAVE_WEB_DATA_PAGE_FUNC had an error (since it is standardized), and handle it
'accordingly.

'----------------------------------------------------------------------------------------------------------

'Returns a specific data element from a specified data source (i.e. web page).

'This function returns a specific element from the data source. It uses the Ticker of the company (TICKER0_STR) and
'the number specifying the element of data to be retrieved (ELEMENT_VAL) as 2 main inputs. The third input is the error string.

'After declaring all supplementary variables the function checks if the web library has been initialized. If it hasn't it
'calls START_WEB_DATA_SYSTEM_FUNC to initialize the library. The function checks if ELEMENT_VAL a valid record number; if not,
'exit the function with an error.

'Then the function checks whether the data element to be retrieved exists in the hash table and returns it if it does. Otherwise
'it returns the N/A value. The function then gets the value from the hash table given the element value key. It then concatenates
'that with a placeholder string and splits that into an array.

'The function then goes to EVALUATE_LINE. The EVALUATE_LINE block checks for the element of data which is being retrieved using
'Select Case. If it is none of the defined cases, the RESULT_VAL is left empty.

'Then it checks whether the webpage has already been retrieved. If it hasn't it saves the data and adds the directory to the hash table.

'Given nothing is stored in RESULT_VAL after EVALUATE_LINE, go to label 1983. The function then proceeds on checking if an error
'occurred; if not, then a new element is found and is then stored in the PUB_WEB_DATA_ELEMENTS_HASH hash table.

'The function checks if the webpage has already been retrieved. If so, it will replace the third element of PARAM_RANGE with the
'existing web page.

'If the first element of PARAM_RANGE is not "Calculated" and the hash table doesn't contain the URL, the function downloads the
'HTML. If there was an error retrieving the HTML, the function will output an error. Next, the function makes a specific exception
```

```
'for Yahoo Finance and takes out some potentially malicious strings. Finally, the URL along with the source HTML are added to the hash.

'The PARSE_LINE block checks whether the PARAM_RNG contains "?" and assigns the parsed value from the hash table to the RESULT_VAL. If
'the PARAM_RNG contains "Obsolete" substring, then the PARAM_RNG(2) is assigned to the RESULT_VAL.

'In any other case the RESULT_VAL is set to be the parsed value of the directory from the hash table.

'It then adds the KEY_STR and RESULT_VAL to the hash table.

Function RETRIEVE_WEB_DATA_ELEMENT_FUNC(ByVal TICKER0_STR As String, _
Optional ByVal ELEMENT_VAL As Long = 1, _
Optional ByVal ERROR_STR As String = "Error") ', _
Optional ByVal FILE_NAME_STR As String = "")
'2012.05.13

'This is very similar to the RCHGetElementNumber:

'TICKER0_STR: A ticker symbol indicating which company data is to
'be returned for. In addition, there are a several literals that
'can be specified for this parameter to request other information.
'See the "Examples" section for more details.

'ELEMENT_VAL: A number specifying which data element is to be retrieved
'for a ticker symbol. A list of element numbers and the data sources and
'data elements

'ERROR_STR: A string or numeric value to be returned if there is an
'error in finding the data element. A default value of "error" is used
'if nothing is passed. This can prevent needing to put IF() statements
'in a cell to make a display or calculation of items easier to read.

Dim DATA_STR As String
Dim PARAM_RNG As Variant
Dim RESULT_VAL As Variant
Dim KEY_STR As String
Const CASES_STR As String = ";N/A;N/A;N/A;N/A;N/A;N/A;N/A;N/A;N/A;N/A" 'Additional special cases to return immediately
```

```
Dim TICKER1_STR As String
Dim TICKER2_STR As String
Dim SRC_URL_STR As String

On Error GoTo ERROR_LABEL

If PUB_WEB_DATA_TABLES_FLAG = False Then: Call START_WEB_DATA_SYSTEM_FUNC
'Additional special cases to return immediately

'PUB_WEB_DATA_ELEMENT_ERROR_STR = ERROR_STR
' Value to return if error

'-------------------------------------------------------------------------------------------------------------------
TICKER1_STR = UCase(TICKER0_STR)
Select Case TICKER1_STR
Case "": GoTo ERROR_LABEL
Case "NONE": GoTo ERROR_LABEL
Case "ERROR": GoTo ERROR_LABEL
Case "VERSION"
   RETRIEVE_WEB_DATA_ELEMENT_FUNC = "Market Data Functions add-in, Version " & PUB_WEB_DATA_VERSION_STR & " (" & ThisWorkbook.Path & "; " & Excel.Application.International(xlCountrySetting) & ")"
   Exit Function
Case "COUNTRY"
   RETRIEVE_WEB_DATA_ELEMENT_FUNC = Excel.Application.International(xlCountrySetting)
   Exit Function
End Select
If ELEMENT_VAL > PUB_WEB_DATA_RECORDS_VAL Then
  RETRIEVE_WEB_DATA_ELEMENT_FUNC = "EOL"
  Exit Function
End If
'-------------------------------------------------------------------------------------------------------------------
KEY_STR = TICKER0_STR & "|" & ELEMENT_VAL
If PUB_WEB_DATA_ELEMENTS_HASH.Exists(KEY_STR) = True Then
   RESULT_VAL = CONVERT_STRING_NUMBER_FUNC(PUB_WEB_DATA_ELEMENTS_HASH(KEY_STR))
   RETRIEVE_WEB_DATA_ELEMENT_FUNC = RESULT_VAL
   Exit Function
```

```
End If
PARAM_RNG = PUB_WEB_DATA_RECORDS_HASH(CStr(ELEMENT_VAL))
If PARAM_RNG = "" Then
   RETRIEVE_WEB_DATA_ELEMENT_FUNC = "N/A" 'Undefined
   Exit Function
End If
PARAM_RNG = Split(PARAM_RNG & CASES_STR, PUB_WEB_DATA_ELEMENT_DELIM_STR)
GoSub EVALUATE_LINE
If RESULT_VAL <> "" Then: GoTo 1983
'If FILE_NAME_STR <> "" Then: GoSub FILE_LINE

TICKER2_STR = CONVERT_YAHOO_TICKER_FUNC(TICKER1_STR, PARAM_RNG(0)) 'See if web page has already been retrieved
SRC_URL_STR = Replace(PARAM_RNG(2), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER2_STR)
'------------------------------------------------------------------------------------------------------------------------
If PARAM_RNG(0) <> "Calculated" And PUB_WEB_DATA_PAGES_HASH.Exists(SRC_URL_STR) = False Then
   DATA_STR = SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, PARAM_RNG(11), True, 0, False)
   If DATA_STR = PUB_WEB_DATA_SYSTEM_ERROR_STR Then: GoTo ERROR_LABEL
   Select Case PARAM_RNG(2)
   Case "http://finance.yahoo.com/advances"
      DATA_STR = Replace(DATA_STR, "<sup>1</sup>", "")
   End Select
   Call PUB_WEB_DATA_PAGES_HASH.Add(SRC_URL_STR, DATA_STR)
End If
GoSub PARSE_LINE
1983:
If RESULT_VAL = ERROR_STR Then: GoTo ERROR_LABEL
Call PUB_WEB_DATA_ELEMENTS_HASH.Add(KEY_STR, RESULT_VAL)
'------------------------------------------------------------------------------------------------------------------------
RESULT_VAL = CONVERT_STRING_NUMBER_FUNC(RESULT_VAL)
RETRIEVE_WEB_DATA_ELEMENT_FUNC = RESULT_VAL
'------------------------------------------------------------------------------------------------------------------------
Exit Function
'------------------------------------------------------------------------------------------------------------------------
EVALUATE_LINE:
'------------------------------------------------------------------------------------------------------------------------
   Select Case True
```

```
    Case TICKER1_STR = "SOURCE"
      RESULT_VAL = PARAM_RNG(0)
    Case TICKER1_STR = "ELEMENT"
      RESULT_VAL = PARAM_RNG(1)
    Case TICKER1_STR = "WEB PAGE"
      RESULT_VAL = PARAM_RNG(2)
    Case TICKER1_STR = "P-URL"
      RESULT_VAL = PARAM_RNG(2)
    Case TICKER1_STR = "P-CELLS"
      RESULT_VAL = PARAM_RNG(3)
    Case TICKER1_STR = "P-FIND1"
      RESULT_VAL = PARAM_RNG(4)
    Case TICKER1_STR = "P-FIND2"
      RESULT_VAL = PARAM_RNG(5)
    Case TICKER1_STR = "P-FIND3"
      RESULT_VAL = PARAM_RNG(6)
    Case TICKER1_STR = "P-FIND4"
      RESULT_VAL = PARAM_RNG(7)
    Case TICKER1_STR = "P-ROWS"
      RESULT_VAL = PARAM_RNG(8)
    Case TICKER1_STR = "P-END"
      RESULT_VAL = PARAM_RNG(9)
    Case TICKER1_STR = "P-LOOK"
      RESULT_VAL = PARAM_RNG(10)
    Case TICKER1_STR = "P-TYPE"
      RESULT_VAL = PARAM_RNG(11)
    Case UCase(PARAM_RNG(0)) = "ADVFN-A" 'PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC("MMM","A",1,">Year End Date")
      RESULT_VAL = PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, "A", PARAM_RNG(3), PARAM_RNG(4), PARAM_RNG(5), ERROR_STR)
    Case UCase(PARAM_RNG(0)) = "ADVFN-Q" 'PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC("MMM","A",1,">Year End Date")
      RESULT_VAL = PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, "Q", PARAM_RNG(3), PARAM_RNG(4), PARAM_RNG(5), ERROR_STR)
    Case UCase(PARAM_RNG(0)) = "EVALUATE"
      RESULT_VAL = Evaluate(Replace(PARAM_RNG(2), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER1_STR))
    Case Left(PARAM_RNG(2), 1) = "="
      RESULT_VAL = Evaluate(Replace(Mid(PARAM_RNG(2), 2), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER1_STR))
'       Debug.Print RESULT_VAL
    Case Else
```

```
      RESULT_VAL = ""
   End Select
'----------------------------------------------------------------------------
Return
'----------------------------------------------------------------------------
PARSE_LINE:
'----------------------------------------------------------------------------
   Select Case True
   Case Left(PARAM_RNG(2), 1) = "?" Or PARAM_RNG(2) = "?"
      RESULT_VAL = PARSE_WEB_DATA_FRAME_FUNC(PUB_WEB_DATA_PAGES_HASH(SRC_URL_STR), "" & PARAM_RNG(1), TICKER1_STR,
ERROR_STR)
   Case Left(PARAM_RNG(3), 1) = "?" Or PARAM_RNG(3) = "?"
      RESULT_VAL = PARSE_WEB_DATA_FRAME_FUNC(PUB_WEB_DATA_PAGES_HASH(SRC_URL_STR), "" & PARAM_RNG(1), TICKER1_STR,
ERROR_STR)
   Case Left(PARAM_RNG(2), 8) = "Obsolete"
      RESULT_VAL = PARAM_RNG(2)
   Case Else
      RESULT_VAL = PARSE_WEB_DATA_CELL_FUNC( _
         PUB_WEB_DATA_PAGES_HASH(SRC_URL_STR), _
         Replace(PARAM_RNG(4), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER2_STR), _
         Replace(PARAM_RNG(5), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER2_STR), _
         Replace(PARAM_RNG(6), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER2_STR), _
         Replace(PARAM_RNG(7), PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER2_STR), _
         PARAM_RNG(8), PARAM_RNG(9), PARAM_RNG(3), PARAM_RNG(10), _
         ERROR_STR)
   End Select
'----------------------------------------------------------------------------
Return
'----------------------------------------------------------------------------
ERROR_LABEL:
RETRIEVE_WEB_DATA_ELEMENT_FUNC = ERROR_STR
End Function

'Extracts a specified table cell from a web page.

Function RETRIEVE_WEB_DATA_CELL_FUNC(ByVal SRC_URL_STR As String, _
ByVal CELL_VAL As Long, _
Optional ByVal FIND1_STR As String = "<BODY", _
Optional ByVal FIND2_STR As String = " ", _
Optional ByVal FIND3_STR As String = " ", _
Optional ByVal FIND4_STR As String = " ", _
Optional ByVal NROWS As Long = 0, _
Optional ByVal END_SYNTAX As String = "</BODY", _
Optional ByVal LOOK_VAL As Long = 0, _
Optional ByVal ERROR_STR As String = "Error", _
Optional ByVal HTTP_TYPE As Integer = 0)
'2011.04.27

'This is similar to the RCHGetTableCell:
'RETRIEVE_WEB_DATA_CELL_FUNC("http://finance.yahoo.com/q/ks?s=MSFT",1,"Market Cap (intraday)")

'Usage Notes

'This is the general process the function uses to extract the data:
'1. The source of the web page specified by "URL" is retrieved from
'the Internet.

'2. A position pointer is set to 1.

'3. The position pointer is advanced to the next location of the string
'specified by "FIND1_STR" found in the web page source.

'4. If "FIND2_STR" is nonblank, the position pointer is advanced to the
'next location of the string specified by "FIND2_STR" found in the web
'page source.

'5. If "FIND3_STR" is nonblank, the position pointer is advanced to the
'next location of the string specified by "FIND3_STR" found in the web
'page source.

'6. If "FIND4_STR" is nonblank, the position pointer is advanced to the
'next location of the string specified by "FIND4_STR" found in the web
```

```
'page source.

'7. If "NROWS#" is not zero, the ending position of the table is set by
'finding the string specified by "END_SYNTAX".

'8. If "NROWS#" is not zero, the position pointer is advanced the number
'of table rows requested, to the start of the table row. If the next row
'found is beyond the position set by "END_SYNTAX", an extraction error is
'signaled.

'9. The position pointer is advanced the number of table cells specified
'by "CELL_VAL#". If the end of the current table row is hit before the
'cell is found, an extraction error is signaled.

'10. If "LOOK_VAL#" is zero, the current cell is returned. Otherwise, it
'looks for and returns the first non-empty cell up to the number specified
'by "LOOK_VAL#".

'If you are retrieving multiple elements from the same page, only
'one web page retrieval needs to be done. The source of the web page
'will be saved and used for extracton of later data elements.

Dim KEY_STR As String
Dim DATA_STR As String
Dim RESULT_VAL As Variant

On Error GoTo ERROR_LABEL

If PUB_WEB_DATA_TABLES_FLAG = False Then: Call START_WEB_DATA_SYSTEM_FUNC

KEY_STR = SRC_URL_STR & "|" & CELL_VAL & "|" & FIND1_STR & "|" & FIND2_STR & "|" & FIND3_STR & "|" & FIND4_STR & "|" & NROWS & "|" &
END_SYNTAX & "|" & LOOK_VAL
If PUB_WEB_DATA_ELEMENTS_HASH.Exists(KEY_STR) = True Then
    RESULT_VAL = CONVERT_STRING_NUMBER_FUNC(PUB_WEB_DATA_ELEMENTS_HASH(KEY_STR))
    RETRIEVE_WEB_DATA_CELL_FUNC = RESULT_VAL
    Exit Function
```

```
End If

DATA_STR = SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, 0, True, 0, False)
If DATA_STR = PUB_WEB_DATA_SYSTEM_ERROR_STR Then: GoTo ERROR_LABEL

RESULT_VAL = PARSE_WEB_DATA_CELL_FUNC(DATA_STR, FIND1_STR, FIND2_STR, FIND3_STR, FIND4_STR, NROWS, END_SYNTAX, CELL_VAL,
LOOK_VAL, ERROR_STR)
Call PUB_WEB_DATA_ELEMENTS_HASH.Add(KEY_STR, RESULT_VAL)
RETRIEVE_WEB_DATA_CELL_FUNC = CONVERT_STRING_NUMBER_FUNC(RESULT_VAL)

'This function uses the same caching technique as the
'RETRIEVE_WEB_DATA_ELEMENT_FUNC() function

'If someone does used this to define a number of elements on a page
'I haven't implemented, it should be a fairly trivial task for me to
'take their cell extraction data and convert them into a set of
'legitimate data elements.

Exit Function
ERROR_LABEL:
RETRIEVE_WEB_DATA_CELL_FUNC = ERROR_STR
End Function


'Similar to the Get HTML Table
'This function is used to extract an HTML table from a web page.

'SRC_URL_STR : URL of the web page to retrieve.

'FIND_BEGIN_STR: String to search for on web page to find start of table.

'BEGIN_DIRECTION_VAL: Number of <TABLE tags to searh for after finding
'the above string to find the start of the table. A negative number
'indicates to search backwards, positive number forwards.

'FIND_END_STR: String to search for on web page to find end of table. If
```

```
'blank, the "Find Begin" parameter will be reused.

'END_DIRECTION_VAL : Number of </TABLE tags to searh for after finding
'the above string to find the end of the table. A negative number
'indicates to search backwards, positive number forwards.

'Usage Notes:
'This function returns an array of data (the HTML table), so it needs
'to be array-entered. To array-enter a formula in EXCEL, first highlight
'the range of cells where you would like the returned data to appear --
'the number of rows and columns for the range will depend on the size
'of the table you are retrieving and how much of that table you want
'to see. Next, enter your formula and then press Ctrl-Shift-Enter.

'What it does -- given this invocation:

'RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ks?s=MMM", _
"Market Cap (intraday)",-1,"",1)

'The function will:
'1) Retrieve HTML source of the Yahoo! Key Statistics web page.

'2) Search for "Market Cap (intraday)" within the source of the web page.

'3) Set the start of the HTML table to be the first "<TABLE" tag prior
'to that string (i.e. -1).

'4) Search for "Market Cap (intraday)" within the source of the web page.

'5) Set the end of the HTML table to be the first "</TABLE" tag after that
'string (i.e. 1).

'6) Return the full table specified by and including the found "<TABLE"
'and "</TABLE" tags.

'Examples:
```

```
'RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ks?s=MMM","PEG Ratio",-1,"",1)
'RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ao?s=IBM", "Mean Recommendation", -3, "Mean Recommendation", 1)
'RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ao?s=IBM", "Mean Target", -3, "Mean Target", 1)
'RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ao?s=IBM", "Three Months Ago", -4, "Three Months Ago", 1)
'RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ud?s=IBM", "Research Firm", -1, "Research Firm", 1)

'Sample invocation to grab "Price Target Summary" from Yahoo for ticker IBM:
'=RETRIEVE_WEB_DATA_TABLE_FUNC("http://finance.yahoo.com/q/ao?s=IBM", "Mean Target", -3, "Mean Target", 1)
'=RETRIEVE_WEB_DATA_TABLE_FUNC("http://www.toteboard.net/Models/SecurityMasterFile.html","Security Master File",1,"EOF",-
1,FALSE,11000,9,0)

Function RETRIEVE_WEB_DATA_TABLE_FUNC(ByVal SRC_URL_STR As String, _
ByVal FIND_BEGIN_STR As String, _
ByVal BEGIN_DIRECTION_VAL As Long, _
ByVal FIND_END_STR As String, _
ByVal END_DIRECTION_VAL As Long, _
Optional ByVal ROW_ONLY_FLAG As Boolean = False, _
Optional ByVal AROWS As Long = 10403, _
Optional ByVal ACOLUMNS As Long = 10, _
Optional ByVal HTTP_TYPE As Integer = 0)

Dim NROWS As Long
Dim NCOLUMNS As Long
Dim KEY_STR As String
Dim DATA_STR As String

Dim TEMP_MATRIX As Variant

On Error GoTo ERROR_LABEL

If PUB_WEB_DATA_TABLES_FLAG = False Then: Call START_WEB_DATA_SYSTEM_FUNC
NROWS = AROWS  ' Rows
NCOLUMNS = ACOLUMNS  ' Columns
If AROWS = 0 Or ACOLUMNS = 0 Then
   If AROWS = 0 Then NROWS = 10   ' Old default
   If ACOLUMNS = 0 Then NCOLUMNS = 10   ' Old default
```

```
    On Error Resume Next
    NROWS = Excel.Application.Caller.Rows.COUNT
    NCOLUMNS = Excel.Application.Caller.Columns.COUNT
    On Error GoTo ERROR_LABEL
End If
KEY_STR = SRC_URL_STR & "|" & FIND_BEGIN_STR & "|" & BEGIN_DIRECTION_VAL & "|" & FIND_END_STR & "|" & END_DIRECTION_VAL & "|" &
ROW_ONLY_FLAG & "|" & NROWS & "|" & NCOLUMNS
'------------------------------------------------------------------------------
If PUB_WEB_DATA_ELEMENTS_HASH.Exists(KEY_STR) = True Then
'------------------------------------------------------------------------------
    TEMP_MATRIX = PUB_WEB_DATA_ELEMENTS_HASH(KEY_STR)
    If IsArray(TEMP_MATRIX) = False Then: GoTo ERROR_LABEL
    RETRIEVE_WEB_DATA_TABLE_FUNC = CONVERT_STRING_NUMBER_FUNC(TEMP_MATRIX)
    Exit Function
'------------------------------------------------------------------------------
End If
'------------------------------------------------------------------------------

DATA_STR = SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, HTTP_TYPE, False, 0, False)
If DATA_STR = PUB_WEB_DATA_SYSTEM_ERROR_STR Then: GoTo ERROR_LABEL
DATA_STR = PARSE_WEB_DATA_PAGE_SYNTAX_FUNC(DATA_STR, 1) '0)

TEMP_MATRIX = PARSE_WEB_DATA_TABLE_FUNC(DATA_STR, FIND_BEGIN_STR, BEGIN_DIRECTION_VAL, FIND_END_STR, END_DIRECTION_VAL,
ROW_ONLY_FLAG, NROWS, NCOLUMNS)
If IsArray(TEMP_MATRIX) = False Then: GoTo ERROR_LABEL

Call PUB_WEB_DATA_ELEMENTS_HASH.Add(KEY_STR, TEMP_MATRIX)
RETRIEVE_WEB_DATA_TABLE_FUNC = CONVERT_STRING_NUMBER_FUNC(TEMP_MATRIX)

'------------------------------------------------------------------------------
Exit Function
'------------------------------------------------------------------------------
ERROR_LABEL:
RETRIEVE_WEB_DATA_TABLE_FUNC = Err.number
'------------------------------------------------------------------------------
End Function
```

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

Function RETRIEVE_WEB_DATA_PARAMETERS_FUNC( _
ByRef TICKERS_RNG As Variant, _
ByVal ELEMENT_VAL As Long, _
Optional ByVal ERROR_STR As String = "--")

Dim i As Long
Dim j As Long
Dim k As Long
Dim NROWS As Long
Dim HEADINGS_STR As String
Dim TEMP_MATRIX As Variant
Dim TICKERS_VECTOR As Variant

On Error GoTo ERROR_LABEL

If IsArray(TICKERS_RNG) = True Then
   TICKERS_VECTOR = TICKERS_RNG
   If UBound(TICKERS_VECTOR, 1) = 1 Then: _
   TICKERS_VECTOR = MATRIX_TRANSPOSE_FUNC(TICKERS_VECTOR)
Else
   ReDim TICKERS_VECTOR(1 To 1, 1 To 1)
   TICKERS_VECTOR(1, 1) = TICKERS_RNG
End If
NROWS = UBound(TICKERS_VECTOR, 1)

HEADINGS_STR = "TICKER,VALUE,VERSION,SOURCE,ELEMENT,P-URL,P-CELLS,P-FIND1,P-FIND2,P-FIND3,P-FIND4,P-ROWS,P-END,P-LOOK,P-TYPE,"
ReDim TEMP_MATRIX(0 To NROWS, 1 To 15)
i = 1
For k = 1 To 15
   j = InStr(i, HEADINGS_STR, ",")
   TEMP_MATRIX(0, k) = Mid(HEADINGS_STR, i, j - i)
   i = j + 1
Next k
For i = 1 To NROWS
```

```
   TEMP_MATRIX(i, 1) = TICKERS_VECTOR(i, 1)
   TEMP_MATRIX(i, 2) = RETRIEVE_WEB_DATA_ELEMENT_FUNC(CStr(TEMP_MATRIX(i, 1)), ELEMENT_VAL, ERROR_STR)
   For j = 3 To 15: TEMP_MATRIX(i, j) = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TEMP_MATRIX(0, j), ELEMENT_VAL, ERROR_STR): Next j
Next i

RETRIEVE_WEB_DATA_PARAMETERS_FUNC = TEMP_MATRIX

Exit Function
ERROR_LABEL:
RETRIEVE_WEB_DATA_PARAMETERS_FUNC = ERROR_STR
End Function

'---------------------------------------------------------------------------------------------------------*
'User defined function to extract an HTML table from a web page
'---------------------------------------------------------------------------------------------------------*
'Sample invocation to grab "Price Target Summary" from Yahoo for ticker IBM:
'=GET_PARSE_HTML_TABLE_FUNC("http://finance.yahoo.com/q/ao?s=IBM", "Mean Target", -3, "Mean Target", 1)
'=GET_PARSE_HTML_TABLE_FUNC("http://www.toteboard.net/Models/SecurityMasterFile.html","Security Master File",1,"EOF",-1)
'---------------------------------------------------------------------------------------------------------*

Function RETRIEVE_WEB_DATA_PAGE_FUNC(ByVal SRC_URL_STR As String, _
ByVal FIND1_STR As String, _
ByVal DIR1_INT As Integer, _
ByVal FIND2_STR As String, _
ByVal DIR2_INT As Integer, _
Optional ByVal ROW_FLAG As Boolean = False, _
Optional ByVal AROWS As Integer = 10, _
Optional ByVal ACOLUMNS As Integer = 10, _
Optional ByVal HTTP_TYPE As Integer = 0) 'As Variant

Dim NROWS As Long
Dim NCOLUMNS As Long
Dim DATA_STR As String

On Error GoTo ERROR_LABEL
```

```
NROWS = AROWS  ' Rows
NCOLUMNS = ACOLUMNS  ' Columns
If AROWS = 0 Or ACOLUMNS = 0 Then
   If AROWS = 0 Then NROWS = 10   ' Old default
   If ACOLUMNS = 0 Then NCOLUMNS = 10   ' Old default
   On Error Resume Next
   NROWS = Excel.Application.Caller.Rows.COUNT
   NCOLUMNS = Excel.Application.Caller.Columns.COUNT
   On Error GoTo ERROR_LABEL
End If
DATA_STR = SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, HTTP_TYPE, True, 0, False)
RETRIEVE_WEB_DATA_PAGE_FUNC = PARSE_WEB_DATA_TABLE_FUNC(DATA_STR, FIND1_STR, DIR1_INT, FIND2_STR, DIR2_INT, ROW_FLAG, NROWS, NCOLUMNS)

Exit Function
ERROR_LABEL:
RETRIEVE_WEB_DATA_PAGE_FUNC = Err.number
End Function


Function SAVE_WEB_DATA_PAGE_FUNC(ByVal SRC_URL_STR As String, _
Optional ByVal HTTP_TYPE As Integer = 0, _
Optional ByVal CLEAN_FLAG As Boolean = False, _
Optional ByVal HASH_TYPE As Integer = 0, _
Optional ByVal TRIM_FLAG As Boolean = False, _
Optional ByVal POS_VAL As Variant = 1, _
Optional ByVal LEN_VAL As Integer = 32767, _
Optional ByVal OFFSET_VAL As Integer = 0)
'2011.02.16
'RCHGetWebData: SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, 0, True, 0, True, 1, 32767, 0)
'Dim START_TIMER As Single: Dim END_TIMER As Single
'START_TIMER = Timer
'For i = 1 To 20000: Call RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER_STR, i): Next i
'END_TIMER = Timer
'Debug.Print Format(END_TIMER - START_TIMER, "#.###") & " seconds"
```

Coded by Nico
As of May 7, 2013

```
Dim i As Long
Dim j As Long
Dim k As Long
Dim KEY_STR As String
Dim DATA_STR As String

On Error GoTo ERROR_LABEL

'-----------------------------------------------------------------------------------
Select Case HASH_TYPE
'-----------------------------------------------------------------------------------
Case 0 'The DATA_STR variable is created as the returning variable of the function.
'The function first checks whether the storage method used for webpage data is array
'or collection. If an array is used, the function will proceed on looping through
'existing records by checking the URL. The associated source code will be stored in
'DATA_STR if the record already exists; otherwise, the source code will be retrieved
'through the CALL_WEB_DATA_PAGE_FUNC function with the option of cleaning the source
'code. A new record will then be created with the URL as the first element and the
'source code as the second. The source code will also be stored in DATA_STR. If the
'URL cannot be found within existing records and the array is full, return an error.
'Right after an existing record is found or an empty slot is filled, exit the for loop.

'If the collection method is used, a key string is then created with the HTTP_TYPE and
'the URL of the required webpage. Similarly, the source code is stored in DATA_STR if
'there is an existing record associated with the URL; otherwise, the source code will
'be retrieved through the CALL_WEB_DATA_PAGE_FUNC function with the option of cleaning
'the source code. A new record will be created with the URL and the type of the webpage
'being the key and the source code being the item. The source code is then stored in the
'DATA_STR.
'-----------------------------------------------------------------------------------
    For i = 1 To PUB_WEB_DATA_PAGES_VAL
        Select Case True
           Case PUB_WEB_DATA_PAGES_MATRIX(i, 1) = ""
                DATA_STR = CALL_WEB_DATA_PAGE_FUNC(SRC_URL_STR, HTTP_TYPE)
                If CLEAN_FLAG = True Then: GoSub CLEAN_LINE
                PUB_WEB_DATA_PAGES_MATRIX(i, 1) = HTTP_TYPE & ":" & SRC_URL_STR
                PUB_WEB_DATA_PAGES_MATRIX(i, 2) = DATA_STR
                Exit For
            Case PUB_WEB_DATA_PAGES_MATRIX(i, 1) = HTTP_TYPE & ":" & SRC_URL_STR: Exit For
            Case i = PUB_WEB_DATA_PAGES_VAL: GoTo ERROR_LABEL
        End Select
    Next i
    DATA_STR = PUB_WEB_DATA_PAGES_MATRIX(i, 2)

'    KEY_STR = HTTP_TYPE & ":" & SRC_URL_STR
'    If (UBound(Filter(PUB_WEB_DATA_PAGES_URL_ARR, KEY_STR)) > -1) Then
'        For i = 1 To PUB_WEB_DATA_PAGES_VAL
'            Select Case True
'               Case PUB_WEB_DATA_PAGES_URL_ARR(i) = KEY_STR: Exit For
'               Case i = PUB_WEB_DATA_PAGES_VAL: GoTo ERROR_LABEL
'            End Select
'        Next i
'        DATA_STR = PUB_WEB_DATA_PAGES_ARR(i)
        'DATA_STR = PUB_WEB_DATA_PAGES_ARR(CLng(PUB_WEB_DATA_PAGES_OBJ.Item(KEY_STR)))
'    Else
'        PUB_WEB_DATA_PAGES_INDEX_VAL = PUB_WEB_DATA_PAGES_INDEX_VAL + 1
        'Call PUB_WEB_DATA_PAGES_OBJ.Add(CStr(PUB_WEB_DATA_PAGES_INDEX_VAL), KEY_STR)
'        DATA_STR = CALL_WEB_DATA_PAGE_FUNC(SRC_URL_STR, HTTP_TYPE)
'        If CLEAN_FLAG = True Then: GoSub CLEAN_LINE
'        PUB_WEB_DATA_PAGES_URL_ARR(PUB_WEB_DATA_PAGES_INDEX_VAL) = KEY_STR
'        PUB_WEB_DATA_PAGES_ARR(PUB_WEB_DATA_PAGES_INDEX_VAL) = DATA_STR
'    End If
'----------------------------------------------------------------------------------
Case Else 'Collection
'----------------------------------------------------------------------------------
    On Error Resume Next
    KEY_STR = HTTP_TYPE & ":" & SRC_URL_STR
    DATA_STR = PUB_WEB_DATA_PAGES_OBJ.Item(KEY_STR)
    If Err.number <> 0 Then
        Err.Clear
        DATA_STR = CALL_WEB_DATA_PAGE_FUNC(SRC_URL_STR, HTTP_TYPE)
        If CLEAN_FLAG = True Then: GoSub CLEAN_LINE
```

```
      Call PUB_WEB_DATA_PAGES_OBJ.Add(DATA_STR, KEY_STR)
   End If
'--------------------------------------------------------------------------------
End Select
'--------------------------------------------------------------------------------
If TRIM_FLAG = True Then: GoSub TRIM_LINE 'If the TRIM_FLAG is true, a portion of the
'DATA_STR will be extracted and stored back in DATA_STR. The extracted portion start
'at the j th character and ends on the k th character of the string. If POS_VAL is
'numeric, then it is the starting position of the extraction; otherwise, the location
'is calculated by locating the position of POS_VAL in DATA_STR and add the optional
'OFFSET_VAL. If the starting position plus LEN_VAL is less than the length of DATA_STR,
'then the length of extraction is LEN_VAL; otherwise, the ending position of the
'extraction is the end of DATA_STR.

SAVE_WEB_DATA_PAGE_FUNC = DATA_STR

'--------------------------------------------------------------------------------
Exit Function
'--------------------------------------------------------------------------------
CLEAN_LINE: 'This line calls the PARSE_WEB_DATA_PAGE_SYNTAX_FUNC function to replace
'HTML coding and ASCII coding in the webpage source code with string and number values
'useful for further analysis.
'--------------------------------------------------------------------------------
   DATA_STR = PARSE_WEB_DATA_PAGE_SYNTAX_FUNC(DATA_STR, 1)
'--------------------------------------------------------------------------------
Return
'--------------------------------------------------------------------------------
TRIM_LINE: 'Preprocess web page data
'--------------------------------------------------------------------------------
'Preprocess Web Data

'Extracts source data from a web page. The primary purpose of this
'function is for testing, to examine the web page data being returned
'for processing. However, it can also be used for ad hoc extractions
'of data from a web page that isn't table oriented.
```

```
'SRC_URL_STR: Web page to retrieve source data to extract from.
'POS_VAL: An optional parameter that has a default value of 1,
'indicating either:
   '1. A number indicating an absolute position on the page to begin
   'the extraction of data

   '2. A string to search for on the page to indicate a relative
   'position on the page to begin the extraction of data

'LEN_VAL: An optional parameter that has a default value of 32767
'(the maximum possible value), indicating the length of the data to
'extract from the web page.

'OFF_VAL: An optional parameter that has a default value of 0,
'indicating the relative position to offset from parameter "Position"
'for extraction of data from the web page.

'Usage Notes

'A value of "Error" is returned if the "Position" or "Length" values are
'invalid for the web page. However, if the "Position" is within the web
'page and the "Length" would cause the extraction to go outside of the
'web page, the "Length" is reset so the extraction only goes to the length
'of the web page.

   j = IIf(IsNumeric(POS_VAL), POS_VAL, InStr(DATA_STR, POS_VAL) + OFFSET_VAL)
   k = IIf(j + LEN_VAL <= Len(DATA_STR), LEN_VAL, Len(DATA_STR) - j + 1)
   DATA_STR = Mid(DATA_STR, j, k)
'-----------------------------------------------------------------------------------
Return
'-----------------------------------------------------------------------------------
ERROR_LABEL:
SAVE_WEB_DATA_PAGE_FUNC = PUB_WEB_DATA_SYSTEM_ERROR_STR
End Function

'The XML object is called in the CALL_WEB_DATA_PAGE_FUNC function through the SAVE_WEB_DATA_PAGE_FUNC
```

```
'function. It is used to retrieve webpage source code with given webpage URL. The XML object is much
'preferred over IE since it is much faster and uses significantly less memory.

Function GET_WEB_DATA_PAGE_FUNC(ByVal pURL As String, _
Optional ByVal pPos As Variant = 1, _
Optional ByVal pLen As Integer = 32767, _
Optional ByVal pOffset As Integer = 0, _
Optional ByVal pUseIE As Integer = 0)
'(ByVal i As Integer, ByVal j As Integer)

'Inputs:
'pURL: URL of the required webpage

'pPos: starting position of extraction of the source code, or the key character that is searched in the target source code, default is 1

'pLen: length of the extraction of the source code, default is 32767 (maximum integer)

'pOffset: if pPos is a string (character), then this variable is used to determine the starting position of extraction relative to the location
'of the key character, default is 0

'pUseIE: type of the object used to retrieve the source code. The default is 0, corresponding to use the XMLHTTP object


'Debug.Print GET_WEB_DATA_PAGE_FUNC("http://www.barchart.com/data/performance.phpx?sym=MSFT", "sig=""5""", 50)

On Error GoTo ERROR_LABEL
'GET_WEB_DATA_PAGE_FUNC = Left(PUB_WEB_DATA_PAGES_MATRIX(i, j), 32767)
GET_WEB_DATA_PAGE_FUNC = SAVE_WEB_DATA_PAGE_FUNC(pURL, pUseIE, True, 0, True, pPos, pLen, pOffset)

Exit Function
ERROR_LABEL:
GET_WEB_DATA_PAGE_FUNC = Err.number
End Function


Private Function CALL_WEB_DATA_PAGE_FUNC(ByVal SRC_URL_STR As String, _
Optional ByVal HTTP_TYPE As Integer = 0)

Dim DATA_STR As String

On Error GoTo ERROR_LABEL

'--------------------------------------------------------------------------------
Select Case HTTP_TYPE
'--------------------------------------------------------------------------------
Case 0, 1 'XMLHTTP Get/Post Object
'--------------------------------------------------------------------------------
  Dim HTTP_OBJ As New MSXML2.XMLHTTP60
  If HTTP_TYPE = 0 Then 'Get
    HTTP_OBJ.Open "GET", SRC_URL_STR, False
'DoEvents
    HTTP_OBJ.send
'DoEvents
    Select Case HTTP_OBJ.Status
      Case 0: DATA_STR = HTTP_OBJ.ResponseText
      Case 200: DATA_STR = HTTP_OBJ.ResponseText
      Case Else: GoTo ERROR_LABEL
    End Select
  Else 'Post
    HTTP_OBJ.Open "POST", SRC_URL_STR, False
    HTTP_OBJ.send
    Select Case HTTP_OBJ.Status
      Case 0: DATA_STR = HTTP_OBJ.ResponseText
      Case 200: DATA_STR = HTTP_OBJ.ResponseText
      Case Else: GoTo ERROR_LABEL
    End Select
  End If
'--------------------------------------------------------------------------------
Case 2 'IE Object
'--------------------------------------------------------------------------------
  Dim IE_OBJ As InternetExplorer 'As Object
  'CreateObject("InternetExplorer.Application")
```

```
    On Error GoTo ERROR_LABEL
    Set IE_OBJ = New InternetExplorer 'CreateObject("InternetExplorer.Application")
    IE_OBJ.Visible = False
    With IE_OBJ
        .navigate SRC_URL_STR
        Do Until Not .Busy
            DoEvents
            Loop
        DATA_STR = .document.DocumentElement.outerHTML
        .Quit
        End With
    Set IE_OBJ = Nothing
'------------------------------------------------------------------------------
Case Else 'HTMLDocument Object
'------------------------------------------------------------------------------
    Dim START_VAL As Variant
    Dim DOC_OBJ As Object
    Dim HTML_OBJ As New HTMLDocument
    Set DOC_OBJ = HTML_OBJ.createDocumentFromUrl(SRC_URL_STR, vbNullString)
    Do: DoEvents: Loop Until DOC_OBJ.readyState = "complete"
    START_VAL = Timer
    Do While Timer < START_VAL + 2: DoEvents: Loop
    ' Wait for JavaScript to run on page?
    DATA_STR = DOC_OBJ.DocumentElement.outerHTML
'------------------------------------------------------------------------------
End Select
'------------------------------------------------------------------------------

CALL_WEB_DATA_PAGE_FUNC = DATA_STR

Exit Function
ERROR_LABEL:
CALL_WEB_DATA_PAGE_FUNC = PUB_WEB_DATA_SYSTEM_ERROR_STR
End Function

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

Coded by Nico
As of May 7, 2013

```
'The idea for this function came from Randy Harmelink's stock market functions add-in.
'finance.groups.yahoo.com/group/smf_addin//

'This function is very useful since it encapsulates all of the variability associate with
'grabbing data from different web sources within one function. The function's input includes
'DATA1_STR, which is the source HTML code, along with 3 other optional inputs. One of the
'optional inputs, the OUTPUT is very important since it provides context regarding the HTML
'code. This function uses this context and grabs the actual data value from the HTML code.
'The benefits to having this is one function is maintainability. In the future, this function
'can be edited to grab specific data points if the web site changes.

Private Function PARSE_WEB_DATA_FRAME_FUNC(ByVal DATA1_STR As String, _
Optional ByVal OUTPUT As String, _
Optional ByVal TICKER1_STR As String = "", _
Optional ByVal ERROR_STR As String = "Error")

Dim h As Long
Dim i As Long
Dim j As Long
Dim k As Long
Dim l As Long

Dim N1_VAL As Variant
Dim N2_VAL As Variant
Dim N3_VAL As Variant
Dim N4_VAL As Variant
Dim N5_VAL As Variant
Dim N6_VAL As Variant
Dim N7_VAL As Variant
Dim N8_VAL As Variant
Dim N9_VAL As Variant
Dim N10_VAL As Variant
Dim N11_VAL As Variant
Dim N12_VAL As Variant
Dim N13_VAL As Variant
Dim N14_VAL As Variant
Dim N15_VAL As Variant

Dim TEMP1_STR As String
Dim TEMP2_STR As String

Dim DATA2_STR As String

On Error GoTo ERROR_LABEL

DATA2_STR = UCase(DATA1_STR)

'------------------------------------------------------------------------------
Select Case OUTPUT
'------------------------------------------------------------------------------
Case "Financial Statements Currency Magnitude" ' Google
'------------------------------------------------------------------------------
   j = InStr(DATA2_STR, "(EXCEPT FOR PER SHARE ITEMS)")
   j = InStrRev(DATA2_STR, " OF ", j)
   i = InStrRev(DATA2_STR, ">IN ", j)
   PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 4, j - i - 4)
'------------------------------------------------------------------------------
Case "Financial Statements Currency Type" ' Google
'------------------------------------------------------------------------------
   j = InStr(DATA2_STR, "(EXCEPT FOR PER SHARE ITEMS)")
   i = InStrRev(DATA2_STR, " OF ", j)
   j = InStr(i, DATA2_STR, "<")
   PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 4, j - i - 4)
'------------------------------------------------------------------------------
Case "FYI Alerts" ' MSN
'------------------------------------------------------------------------------
   PARSE_WEB_DATA_FRAME_FUNC = "No longer available"
   'iPos1 = InStr(sData(3), ">ADVISOR FYI<")
'------------------------------------------------------------------------------
Case "Company Description" ' MSN
'------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "<BODY")
    i = InStr(i, DATA2_STR, "COMPANY REPORT")
    i = InStr(i, DATA2_STR, "<P>") + 2
    j = InStr(i, DATA2_STR, "</P>")
    PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 1, j - i - 1)
'--------------------------------------------------------------------------------
Case "Company Name" ' MSN
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "<TITLE>")
    j = InStr(i, DATA2_STR, " REPORT - ")
    PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 7, j - i - 7)
'--------------------------------------------------------------------------------
Case "Risk Grade"  ' MSN
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "RISK:") + 6
    If i = 6 Then
      PARSE_WEB_DATA_FRAME_FUNC = ERROR_STR
    Else
      h = CInt(Mid(DATA1_STR, i, 1))
      PARSE_WEB_DATA_FRAME_FUNC = Mid("ABCDF", h, 1)
    End If
'--------------------------------------------------------------------------------
Case "Return Grade"  ' MSN
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "RETURN:") + 8
    If i = 8 Then
      PARSE_WEB_DATA_FRAME_FUNC = ERROR_STR
    Else
      h = CInt(Mid(DATA1_STR, i, 1))
      PARSE_WEB_DATA_FRAME_FUNC = Mid("FDCBA", h, 1)
    End If
'--------------------------------------------------------------------------------
Case "Quick Summary"  ' MSN
'--------------------------------------------------------------------------------
    PARSE_WEB_DATA_FRAME_FUNC = ""
    i = InStr(DATA2_STR, "RETURN:") + 8
    j = InStr(DATA2_STR, "QUICK SUMMARY")
    For h = 1 To 20
        j = InStr(j, DATA2_STR, "<DD>") + 4
        If j > i Or j = 4 Then Exit For
        k = InStr(j, DATA2_STR, "<B>")
        l = InStr(j, DATA2_STR, "</B>")
        TEMP1_STR = Mid(DATA1_STR, k + 3, l - k - 3)
        TEMP2_STR = Mid(DATA1_STR, j, k - j)
        PARSE_WEB_DATA_FRAME_FUNC = PARSE_WEB_DATA_FRAME_FUNC & TEMP1_STR & " -- " & TEMP2_STR & vbLf
    Next h
'------------------------------------------------------------------------------
Case "StockScouter Rating -- Summary"  ' MSN
'------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "ALT=""STOCKSCOUTER RATING: ")
    i = InStr(i, DATA2_STR, "<P>") + 3
    j = InStr(i, DATA2_STR, "</P>")
    PARSE_WEB_DATA_FRAME_FUNC = Replace(Replace(Mid(DATA1_STR, i, j - i), "<b>", ""), "</b>", "")
'------------------------------------------------------------------------------
Case "Short Term Outlook"  ' MSN
'------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "SHORT-TERM OUTLOOK")
    i = InStr(i, DATA2_STR, "<P>") + 3
    j = InStr(i, DATA2_STR, "</P>")
    PARSE_WEB_DATA_FRAME_FUNC = Replace(Replace(Mid(DATA1_STR, i, j - i), "<b>", ""), "</b>", "")
'------------------------------------------------------------------------------
Case "StockScouter Rating -- Current"  ' MSN
'------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "ALT=""STOCKSCOUTER RATING: ")
    j = InStr(i, DATA2_STR, ":") + 2
    k = InStr(j, DATA2_STR, """")
    PARSE_WEB_DATA_FRAME_FUNC = CInt(Mid(DATA1_STR, j, k - j))
'------------------------------------------------------------------------------
Case "Risk Alert Level" ' Reuters
'------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "IMAGES/SELLALERT")
    i = InStr(i, DATA2_STR, "ALT=""") + 5
    j = InStr(i, DATA2_STR, """")
```

```
   PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i, j - i)
'-----------------------------------------------------------------------------
Case "P&F -- Pattern" ' Stockcharts
'-----------------------------------------------------------------------------
   i = InStr(DATA2_STR, "P&F PATTERN:")
   If i = 0 Then
     PARSE_WEB_DATA_FRAME_FUNC = "No P&F Pattern Found"
     Exit Function
   End If
   j = InStr(i, DATA2_STR, "</DIV")
   i = InStrRev(DATA2_STR, ">", j) + 1
   k = InStrRev(DATA2_STR, "#00AA00", i)
   If i - k < 40 Then
     PARSE_WEB_DATA_FRAME_FUNC = "Bullish -- " & Trim(Mid(DATA1_STR, i, j - i))
     Exit Function
   End If
   k = InStrRev(DATA2_STR, "#FF0000", i)
   If i - k < 40 Then
     PARSE_WEB_DATA_FRAME_FUNC = "Bearish -- " & Trim(Mid(DATA1_STR, i, j - i))
     Exit Function
   End If
   PARSE_WEB_DATA_FRAME_FUNC = "Unknown -- " & Trim(Mid(DATA1_STR, i, j - i))
'-----------------------------------------------------------------------------
Case "P&F -- Price Objective"  ' Stockcharts
'-----------------------------------------------------------------------------
   i = InStr(DATA2_STR, " PRICE OBJ. ")
   If i = 0 Then GoTo ERROR_LABEL
   i = InStr(i, DATA2_STR, ":") + 2
   j = InStr(i, DATA2_STR, "<")
   PARSE_WEB_DATA_FRAME_FUNC = Trim(Mid(DATA1_STR, i, j - i))
'-----------------------------------------------------------------------------
Case "P&F -- Trend"  ' Stockcharts
'-----------------------------------------------------------------------------
   i = InStr(DATA2_STR, " PRICE OBJ. ")
   If i > 0 Then
     PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i - 7, 7)
```

```
    Else
      PARSE_WEB_DATA_FRAME_FUNC = "Unknown"
  End If
'--------------------------------------------------------------------------------
Case "Next Earnings Date" ' Yahoo
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "NEXT EARNINGS DATE: ") + 20
    If i = 20 Then
      PARSE_WEB_DATA_FRAME_FUNC = "N/A"
      Exit Function
    End If
    j = InStr(i, DATA2_STR, " - ")
    If j = 0 Then GoTo ERROR_LABEL
    PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i, j - i)
'--------------------------------------------------------------------------------
Case "Sector Number" ' Yahoo
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, "HTTP://BIZ.YAHOO.COM/P/")
    If i = 0 Then GoTo ERROR_LABEL
    PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 23, 1)
'--------------------------------------------------------------------------------
Case "Industry Number" ' Yahoo
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, ">INDUSTRY:<")
    i = InStr(i, DATA2_STR, "HTTP://BIZ.YAHOO.COM/IC/")
    If i = 0 Then GoTo ERROR_LABEL
    PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 24, 3)
'--------------------------------------------------------------------------------
Case "Industry Symbol" ' Yahoo
'--------------------------------------------------------------------------------
    i = InStr(DATA2_STR, ">^")
    If i = 0 Then GoTo ERROR_LABEL
    PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 1, 8)
'--------------------------------------------------------------------------------
Case "Company Name" ' Yahoo
'--------------------------------------------------------------------------------
```

```
   j = InStr(DATA2_STR, " (" & TICKER1_STR & ")</B>")
   i = InStrRev(DATA2_STR, "<B>", j)
   PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 3, j - i - 3)
'------------------------------------------------------------------------------
Case "Fund Profile -- Morningstar Rating" ' Yahoo
'------------------------------------------------------------------------------
  Select Case True
  Case InStr(DATA2_STR, "/STAR1.GIF") > 0
    PARSE_WEB_DATA_FRAME_FUNC = 1
  Case InStr(DATA2_STR, "/STAR2.GIF") > 0
    PARSE_WEB_DATA_FRAME_FUNC = 2
  Case InStr(DATA2_STR, "/STAR3.GIF") > 0
    PARSE_WEB_DATA_FRAME_FUNC = 3
  Case InStr(DATA2_STR, "/STAR4.GIF") > 0
    PARSE_WEB_DATA_FRAME_FUNC = 4
  Case InStr(DATA2_STR, "/STAR5.GIF") > 0
    PARSE_WEB_DATA_FRAME_FUNC = 5
  Case Else
    PARSE_WEB_DATA_FRAME_FUNC = ERROR_STR
  End Select
'------------------------------------------------------------------------------
Case "Fund Profile -- Last Dividend -- Date" ' Yahoo
'------------------------------------------------------------------------------
   i = InStr(DATA2_STR, ">FUND OPERATIONS")
   If i = 0 Then GoTo ERROR_LABEL
   i = InStr(i, DATA2_STR, "LAST DIVIDEND")
   If i = 0 Then GoTo ERROR_LABEL
   i = InStr(i, DATA2_STR, "(")
   If i = 0 Then GoTo ERROR_LABEL
   j = InStr(i, DATA2_STR, ")")
   If j < i Then GoTo ERROR_LABEL
   PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 1, j - i - 1)
'------------------------------------------------------------------------------
Case "Fund Profile -- Last Cap Gain -- Date" ' Yahoo
'------------------------------------------------------------------------------
   i = InStr(DATA2_STR, ">FUND OPERATIONS")
   If i = 0 Then GoTo ERROR_LABEL
   i = InStr(i, DATA2_STR, "LAST CAP GAIN")
   If i = 0 Then GoTo ERROR_LABEL
   i = InStr(i, DATA2_STR, "(")
   If i = 0 Then GoTo ERROR_LABEL
   j = InStr(i, DATA2_STR, ")")
   If j < i Then GoTo ERROR_LABEL
   PARSE_WEB_DATA_FRAME_FUNC = Mid(DATA1_STR, i + 1, j - i - 1)
'------------------------------------------------------------------------------
Case "Piotroski 1 (Positive Net Income)"
'------------------------------------------------------------------------------
   N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8806, ERROR_STR) ' FQ1, Net Income (Continuing Operations)
   N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8807, ERROR_STR) ' FQ2, Net Income (Continuing Operations)
   N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8808, ERROR_STR) ' FQ3, Net Income (Continuing Operations)
   N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8809, ERROR_STR) ' FQ4, Net Income (Continuing Operations)
   PARSE_WEB_DATA_FRAME_FUNC = IIf((N1_VAL + N2_VAL + N3_VAL + N4_VAL) > 0, 1, 0)
'------------------------------------------------------------------------------
Case "Piotroski 2 (Positive Operating Cash Flow)"
'------------------------------------------------------------------------------
   N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 11326, ERROR_STR) ' FQ1, YTD Net Cash Flow (Continuing Operations)
   N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 11330, ERROR_STR) ' FQ5, YTD Net Cash Flow (Continuing Operations)
   N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6856, ERROR_STR) ' FY1, Net Cash Flow (Continuing Operations)
   PARSE_WEB_DATA_FRAME_FUNC = IIf(N1_VAL - N2_VAL + N3_VAL > 0, 1, 0)
'------------------------------------------------------------------------------
Case "Piotroski 3 (Increasing Net Income)"
'------------------------------------------------------------------------------
   N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8066, ERROR_STR)    ' FQ1, Ending Quarter
   If N1_VAL = 4 Then
    N6_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5596, ERROR_STR) ' FY1, Net Income (Continuing Operations)
    N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5597, ERROR_STR) ' FY2, Net Income (Continuing Operations)
   Else
    N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8806, ERROR_STR) ' FQ1, Net Income (Continuing Operations)
    N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8807, ERROR_STR) ' FQ2, Net Income (Continuing Operations)
    N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8808, ERROR_STR) ' FQ3, Net Income (Continuing Operations)
    N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8809, ERROR_STR) ' FQ4, Net Income (Continuing Operations)
    N6_VAL = N2_VAL + N3_VAL + N4_VAL + N5_VAL
```

```
      N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5596, ERROR_STR) ' FY1, Net Income (Continuing Operations)
    End If
    PARSE_WEB_DATA_FRAME_FUNC = IIf(N6_VAL > N7_VAL, 1, 0)
'-------------------------------------------------------------------------------
Case "Piotroski 4 (Operating Cash flow exceeds Net Income)"
'-------------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 11326, ERROR_STR) ' FQ1, YTD Net Cash Flow (Continuing Operations)
    N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 11330, ERROR_STR) ' FQ5, YTD Net Cash Flow (Continuing Operations)
    N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6856, ERROR_STR) ' FY1, Net Cash Flow (Continuing Operations)
    N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8806, ERROR_STR) ' FQ1, Net Income (Continuing Operations)
    N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8807, ERROR_STR) ' FQ2, Net Income (Continuing Operations)
    N6_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8808, ERROR_STR) ' FQ3, Net Income (Continuing Operations)
    N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8809, ERROR_STR) ' FQ4, Net Income (Continuing Operations)
    PARSE_WEB_DATA_FRAME_FUNC = IIf(N1_VAL - N2_VAL + N3_VAL > N4_VAL + N5_VAL + N6_VAL + N7_VAL, 1, 0)
'-------------------------------------------------------------------------------
Case "Piotroski 5 (Decreasing ratio of long-term debt to assets )"
'-------------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8066, ERROR_STR)    ' FQ1, Ending Quarter
    If N1_VAL = 4 Then
      N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6376, ERROR_STR) ' FY1, Long Term Debt
      N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6266, ERROR_STR) ' FY1, Total Assets
      N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6377, ERROR_STR) ' FY2, Long Term Debt
      N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6267, ERROR_STR) ' FY2, Total Assets
    Else
      N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10366, ERROR_STR) ' FQ1, Long Term Debt
      N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10146, ERROR_STR) ' FQ1, Total Assets
      N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6376, ERROR_STR) ' FY1, Long Term Debt
      N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6266, ERROR_STR) ' FY1, Total Assets
    End If
    PARSE_WEB_DATA_FRAME_FUNC = IIf((N2_VAL / N3_VAL) < (N4_VAL / N5_VAL), 1, 0)
'-------------------------------------------------------------------------------
Case "Piotroski 6 (Increasing Current Ratio)"
'-------------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8066, ERROR_STR)    ' FQ1, Ending Quarter
    If N1_VAL = 4 Then
      N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6116, ERROR_STR) ' FY1, Current Assets
```

```
     N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6366, ERROR_STR) ' FY1, Current Liabilities
     N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6117, ERROR_STR) ' FY2, Current Assets
     N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6367, ERROR_STR) ' FY2, Current Liabilities
   Else
     N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 9846, ERROR_STR) ' FQ1, Current Assets
     N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10346, ERROR_STR) ' FQ1, Current Liabilities
     N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6116, ERROR_STR) ' FY1, Current Assets
     N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6366, ERROR_STR) ' FY1, Current Liabilities
   End If
   PARSE_WEB_DATA_FRAME_FUNC = IIf((N2_VAL / N3_VAL) > (N4_VAL / N5_VAL), 1, 0)
'------------------------------------------------------------------------------
Case "Piotroski 7 (No increase in outstanding shares)"
'------------------------------------------------------------------------------
   N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8066, ERROR_STR)    ' FQ1, Ending Quarter
   If N1_VAL = 4 Then
     N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6646, ERROR_STR) ' FY1, Total Common Shares Out
     N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6647, ERROR_STR) ' FY2, Total Common Shares Out
   Else
     N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10906, ERROR_STR) ' FQ1, Total Common Shares Out
     N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6646, ERROR_STR) ' FY1, Total Common Shares Out
   End If
   PARSE_WEB_DATA_FRAME_FUNC = IIf(N2_VAL > N3_VAL, 0, 1)
'------------------------------------------------------------------------------
Case "Piotroski 8 (Increasing Gross Margins)"
'------------------------------------------------------------------------------
   N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8066, ERROR_STR)    ' FQ1, Ending Quarter
   If N1_VAL = 4 Then
     N6_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5346, ERROR_STR) ' FY1, Gross Operating Profit
     N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5347, ERROR_STR) ' FY2, Gross Operating Profit
     N8_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5286, ERROR_STR) ' FY1, Operating Revenue
     N9_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5287, ERROR_STR) ' FY2, Operating Revenue
   Else
     N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8306, ERROR_STR) ' FQ1, Gross Operating Profit
     N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8307, ERROR_STR) ' FQ2, Gross Operating Profit
     N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8308, ERROR_STR) ' FQ3, Gross Operating Profit
     N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8309, ERROR_STR) ' FQ4, Gross Operating Profit
```

```
      N6_VAL = N2_VAL + N3_VAL + N4_VAL + N5_VAL
      N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5346, ERROR_STR) ' FY1, Gross Operating Profit
      N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8186, ERROR_STR) ' FQ1, Operating Revenue
      N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8187, ERROR_STR) ' FQ2, Operating Revenue
      N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8188, ERROR_STR) ' FQ3, Operating Revenue
      N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8189, ERROR_STR) ' FQ4, Operating Revenue
      N8_VAL = N2_VAL + N3_VAL + N4_VAL + N5_VAL
      N9_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5286, ERROR_STR) ' FY1, Operating Revenue
    End If
    PARSE_WEB_DATA_FRAME_FUNC = IIf((N6_VAL / N8_VAL) > (N7_VAL / N9_VAL), 1, 0)
'-----------------------------------------------------------------------------
Case "Piotroski 9 (Increasing Asset Turnover)"
'-----------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8066, ERROR_STR)    ' FQ1, Ending Quarter
    If N1_VAL = 4 Then
      N6_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5286, ERROR_STR) ' FY1, Operating Revenue
      N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6266, ERROR_STR) ' FY1, Total Assets
      N8_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5287, ERROR_STR) ' FY2, Operating Revenue
      N9_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6267, ERROR_STR) ' FY2, Total Assets
    Else
      N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8186, ERROR_STR) ' FQ1, Operating Revenue
      N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8187, ERROR_STR) ' FQ2, Operating Revenue
      N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8188, ERROR_STR) ' FQ3, Operating Revenue
      N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8189, ERROR_STR) ' FQ4, Operating Revenue
      N6_VAL = N2_VAL + N3_VAL + N4_VAL + N5_VAL
      N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10146, ERROR_STR) ' FQ1, Total Assets
      N8_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 5286, ERROR_STR) ' FY1, Operating Revenue
      N9_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 6266, ERROR_STR) ' FY1, Total Assets
    End If
    PARSE_WEB_DATA_FRAME_FUNC = IIf((N6_VAL / N7_VAL) > (N8_VAL / N9_VAL), 1, 0)
'-----------------------------------------------------------------------------
Case "Piotroski F-Score" 'http://moneyterms.co.uk/f-score/
'-----------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15001, ERROR_STR) 'Piotroski 1 (Positive Net Income)
    N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15002, ERROR_STR) 'Piotroski 2 (Positive Operating Cash Flow)
    N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15003, ERROR_STR) 'Piotroski 3 (Increasing Net Income)
    N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15004, ERROR_STR) 'Piotroski 4 (Operating Cash flow exceeds Net Income)
    N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15005, ERROR_STR) 'Piotroski 5 (Decreasing ratio of long-term debt to assets )
    N6_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15006, ERROR_STR) 'Piotroski 6 (Increasing Current Ratio)
    N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15007, ERROR_STR) 'Piotroski 7 (No increase in outstanding shares)
    N8_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15008, ERROR_STR) 'Piotroski 8 (Increasing Gross Margins)
    N9_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 15009, ERROR_STR) 'Piotroski 9 (Increasing Asset Turnover)
    PARSE_WEB_DATA_FRAME_FUNC = N1_VAL + N2_VAL + N3_VAL + N4_VAL + N5_VAL + N6_VAL + N7_VAL + N8_VAL + N9_VAL
'-------------------------------------------------------------------------------
Case "Altman Z-Score"
'-------------------------------------------------------------------------------
'Altman Z-Score for stock in the site http://www.grahamin vestor.com/
'n1 = FQ1, Working Capital
'n2 = FQ1, Total Assets
'n3 = FQ1, Retained Earnings
'n4 = FQ1, EBIT
'n5 = FQ2, EBIT
'n6 = FQ3, EBIT
'n7 = FQ4, EBIT
'n8 = n4 + n5 + n6 + n7
'n9 = Market Capitalization
'n10 = Total Liabilities
'n11 = FQ1, Operating Revenue
'n12 = FQ2, Operating Revenue
'n13 = FQ3, Operating Revenue
'n14 = FQ4, Operating Revenue
'n15 = n11 + n12 + n13 + n14
'SpecialExtractio n = 1.2 * (n1 / n2) + 1.4 * (n3 / n2)
'+ 3.3 * (n8 / n2) + 0.6 * (n9 / n10 / 1000) + (n15 / n2)

    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10786, ERROR_STR) ' FQ1, Working Capital
    N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10146, ERROR_STR) ' FQ1, Total Assets
    N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10646, ERROR_STR) ' FQ1, Retained Earnings
    N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8666, ERROR_STR)  ' FQ1, EBIT
    N5_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8667, ERROR_STR)  ' FQ2, EBIT
    N6_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8668, ERROR_STR)  ' FQ3, EBIT
    N7_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8669, ERROR_STR)  ' FQ4, EBIT
    N8_VAL = N4_VAL + N5_VAL + N6_VAL + N7_VAL
    N9_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 941, ERROR_STR)  ' Market Capitalization
    N10_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10526, ERROR_STR) ' Total Liabilities
    N11_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8186, ERROR_STR) ' FQ1, Operating Revenue
    N12_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8187, ERROR_STR) ' FQ2, Operating Revenue
    N13_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8188, ERROR_STR) ' FQ3, Operating Revenue
    N14_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 8189, ERROR_STR) ' FQ4, Operating Revenue
    N15_VAL = N11_VAL + N12_VAL + N13_VAL + N14_VAL
    PARSE_WEB_DATA_FRAME_FUNC = 1.2 * (N1_VAL / N2_VAL) + 1.4 * (N3_VAL / N2_VAL) + 3.3 * (N8_VAL / N2_VAL) + 0.6 * (N9_VAL / N10_VAL / 1000) + (N15_VAL / N2_VAL)
'-------------------------------------------------------------------------------
Case "Rule #1 MOS Price"
'-------------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 99, ERROR_STR)    ' 5-Year High P/E
    N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 102, ERROR_STR)   ' 5-Year Low P/E
    N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 44, ERROR_STR)    ' Current EPS
    N4_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 442, ERROR_STR)   ' 5-Year Projected Growth Rate
    If N1_VAL > 50 Then N1_VAL = 50
    N5_VAL = N3_VAL * (1 + N4_VAL) ^ 10
    'FV(N4_VAL, 10, 0, -N3_VAL)

    N6_VAL = ((N5_VAL * (N1_VAL + N2_VAL) / 2) / (1 + 0.15) ^ 10) / 2
    'PV(0.15, 10, 0, -N5_VAL * (N1_VAL + N2_VAL) / 2) / 2

    PARSE_WEB_DATA_FRAME_FUNC = N6_VAL

'-------------------------------------------------------------------------------
Case "Magic Formula Investing -- Earnings Yield"
'-------------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 949, ERROR_STR)   ' Enterprise value to EBITDA
    PARSE_WEB_DATA_FRAME_FUNC = 1 / N1_VAL
'-------------------------------------------------------------------------------
Case "Magic Formula Investing -- Return on Capital"
'-------------------------------------------------------------------------------
    N1_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 960, ERROR_STR)   ' EBITDA
    N2_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 964, ERROR_STR)   ' Cash
    N3_VAL = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER1_STR, 10026, ERROR_STR) ' FQ1, Net Fixed Assets (Plant & Equipment)
    PARSE_WEB_DATA_FRAME_FUNC = N1_VAL / (N2_VAL + 1000 * N3_VAL)
'-------------------------------------------------------------------------------
Case Else
   PARSE_WEB_DATA_FRAME_FUNC = ERROR_STR
'-------------------------------------------------------------------------------
End Select
'-------------------------------------------------------------------------------

Exit Function
ERROR_LABEL:
PARSE_WEB_DATA_FRAME_FUNC = ERROR_STR
End Function

'User defined function to return content from between a paired HTML tags

Function PARSE_WEB_DATA_TAG_FUNC(ByVal SRC_URL_STR As String, _
ByVal TAG_STR As String, _
Optional ByVal TAGS_VAL As Integer = 1, _
Optional ByVal FIND1_STR As String = "<", _
Optional ByVal FIND2_STR As String = " ", _
Optional ByVal FIND3_STR As String = " ", _
Optional ByVal FIND4_STR As String = " ", _
Optional ByVal CONV_FLAG As Boolean = False, _
Optional ByVal ERROR_STR As Variant = "Error", _
Optional ByVal HTTP_TYPE As Integer = 0, _
Optional ByVal MAX_LEN As Integer = 32767) 'As Variant

'2012.01.27
'smfGetTagContent
'Example of an invocation:
'Debug.Print PARSE_WEB_DATA_TAG_FUNC("http://www.google.com/finance?client=ob&q=MUTF:GLRBX", "TD", 2, "Sharpe ratio")

Dim h As Long
Dim i As Long
Dim j As Long
```

Coded by Nico
As of May 7, 2013

```
Dim k As Long
Dim l As Long

Dim DATA1_STR As String
Dim DATA2_STR As String

On Error GoTo ERROR_LABEL

DATA1_STR = SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, HTTP_TYPE, True, 0, False) 'Stripped data
If DATA1_STR = PUB_WEB_DATA_SYSTEM_ERROR_STR Then: GoTo ERROR_LABEL

DATA2_STR = UCase(DATA1_STR) 'Upper case of stripped data
'Find initial position on web page
i = 0
i = InStr(i + 1, DATA2_STR, UCase(FIND1_STR))
If i = 0 Then GoTo ERROR_LABEL
If FIND2_STR > " " Then
   i = InStr(i + 1, DATA2_STR, UCase(FIND2_STR))
   If i = 0 Then GoTo ERROR_LABEL
End If
If FIND3_STR > " " Then
   i = InStr(i + 1, DATA2_STR, UCase(FIND3_STR))
   If i = 0 Then GoTo ERROR_LABEL
End If
If FIND4_STR > " " Then
   i = InStr(i + 1, DATA2_STR, UCase(FIND4_STR))
   If i = 0 Then GoTo ERROR_LABEL
End If

'--------------------------------> Skip forward or backward number of HTML tags
l = Abs(TAGS_VAL)
For h = 1 To l
   If TAGS_VAL > 0 Then
      i = InStr(i + 1, DATA2_STR, "<" & UCase(TAG_STR))
   Else
      i = InStrRev(DATA2_STR, "<" & UCase(TAG_STR), i)
   End If
   If i = 0 Then GoTo ERROR_LABEL
Next h

'-------------------------------> Extract data between HTML tags
j = InStr(i, DATA2_STR, ">")
k = InStr(j, DATA2_STR, "</" & UCase(TAG_STR))
If UCase(TAG_STR) = "TD" Then
   l = InStr(j, DATA2_STR, "<TD")
   If l > 0 And (k = 0 Or k > l) Then k = l
   l = InStr(j, DATA2_STR, "</TR")
   If l > 0 And (k = 0 Or k > l) Then k = l
   l = InStr(j, DATA2_STR, "<TR")
   If l > 0 And (k = 0 Or k > l) Then k = l
   l = InStr(j, DATA2_STR, "</TABLE")
   If l > 0 And (k = 0 Or k > l) Then k = l
End If

DATA2_STR = Trim(Mid(DATA1_STR, j + 1, k - j - 1))
If Len(DATA2_STR) > MAX_LEN Then DATA2_STR = Left(DATA2_STR, MAX_LEN) 'MAX_LEN prevents excessive length of returned data
If CONV_FLAG = True Then
   PARSE_WEB_DATA_TAG_FUNC = CONVERT_STRING_NUMBER_FUNC(DATA2_STR)
Else
   PARSE_WEB_DATA_TAG_FUNC = DATA2_STR
End If

'-------------------------------------------------------------------------------
Exit Function
'-------------------------------------------------------------------------------
ERROR_LABEL:
'-------------------------------------------------------------------------------
PARSE_WEB_DATA_TAG_FUNC = ERROR_STR
End Function

Function PARSE_WEB_DATA_CELL_FUNC( _
ByVal DATA1_STR As String, _
```

Coded by Nico
As of May 7, 2013

```
ByVal FIND1_STR As String, _
ByVal FIND2_STR As String, _
ByVal FIND3_STR As String, _
ByVal FIND4_STR As String, _
ByVal NO_ROWS As Integer, _
ByVal END_STR As String, _
ByVal CELLS_INT As Integer, _
ByVal LOOK_INT As Integer, _
Optional ByVal ERROR_STR As String = "--") 'As Variant

'After making all the text uppercase so that the search is not case sensitive, the
'function finds the first position where FIND1_STR occurs in DATA_STR.  If there are
'values in FIND2_STR, FIND3_STR, and FIND4_STR, the string is searched starting from
'the last search position. Once the FIND4_STR is found, it is split by "|" and put into
'an array. Each of these are searched for in the DATA_STR until one is found. The function
'then skips the number of rows indicated by the NO_ROWS input as long as it is before the
'position of the END_STR, and then the number of cells indicated by the CELLS_INT input.
'The contents of this cell between the html tags is what is returned from the function.

'RCHExtractData

'PARAMETERS
'URL : Web page to retrieve the table cell from.

'FIND1_STR : An optional string value to search for to position the function
'on the page before skipping ahead rows and cells to find the data to
'return. Defaults to "<BODY".

'FIND2_STR : An optional string value to search for to further position the
'function on the page (after finding the "FIND1_STR" string) before skipping
'ahead rows and cells to find the data to return. Defaults to " ".

'FIND3_STR : An optional string value to search for to further position the
'function on the page (after finding the "FIND1_STR" thru "FIND2_STR" strings)
'before skipping ahead rows and cells to find the data to return.
'Defaults to " ".
```

```
'FIND4_STR : An optional string value to search for to further position
'the function on the page (after finding the "FIND1_STR" thru "FIND3_STR"
'strings) before skipping ahead rows and cells to find the data to
'return. Defaults to " ".

'NO_ROWS : An option number of rows to skip ahead (after function is positioned
'on the page by "FIND1_STR" thru "FIND4_STR") before skipping ahead the
'specified number of table cells to find the data to return. Defaults to 0.

'END_STR : An optional string value that marks the end of the skip aheads
'based on "CELL_INT#" and "NO_ROWS#". If the next found table cell ia after this
'point, the error message is returned. Defaults to "</BODY", but is usually
'set to "</TABLE" when using "NO_ROWS#" to ensure that the search doesn't go
'outside the current table when skipping ahead by table rows.

'CELL_INT# : The number of cells to skip forward (after function is
'positioned on the page by "FIND1_STR" thru "FIND4_STR" and "NO_ROWS#") before
'returning data.

'LOOK_INT# : An optional number of consecutive cells to search for data
'in (ignoring empty table cells). Rarely used. Defaults to 0.

'ERROR_STR : An optional value to return if the table cell cannot be
'found based on specified parameters. Defaults to "Error".

Dim h As Long
Dim i As Long 'row beg
Dim j As Long 'row end
Dim k As Long 'iLoop
Dim l As Long

Dim hh As Long
Dim ii As Long
Dim jj As Long
Dim kk As Long
Dim ll As Long

Dim DATA2_STR As String
Dim TEMP_VAL As Variant
Dim TEMP_ARR As Variant

On Error GoTo ERROR_LABEL
'Find initial position on web page

DATA2_STR = UCase(DATA1_STR)
ii = 0
ii = InStr(ii + 1, DATA2_STR, UCase(FIND1_STR))
If ii = 0 Then GoTo ERROR_LABEL
If FIND2_STR > " " Then
   ii = InStr(ii + 1, DATA2_STR, UCase(FIND2_STR))
   If ii = 0 Then GoTo ERROR_LABEL
End If
If FIND3_STR > " " Then
   ii = InStr(ii + 1, DATA2_STR, UCase(FIND3_STR))
   If ii = 0 Then GoTo ERROR_LABEL
End If
If FIND4_STR > " " Then
   TEMP_ARR = Split(UCase(FIND4_STR), "|")
   h = UBound(TEMP_ARR, 1)
   For l = 0 To h
      jj = InStr(ii + 1, DATA2_STR, TEMP_ARR(l))
      If jj > 0 Then Exit For
      If l = UBound(TEMP_ARR, 1) Then GoTo ERROR_LABEL
   Next l
   ii = jj
End If

'Skip backward/forward the number of specified table rows
Select Case True
Case NO_ROWS > 0
   jj = InStr(ii, DATA2_STR, UCase(END_STR))
   For l = 1 To NO_ROWS
      ii = InStr(ii + 1, DATA2_STR, "<TR")
      kk = InStr(ii, DATA2_STR, "</TR")
      If kk > jj Then: GoTo ERROR_LABEL
   Next l
Case NO_ROWS < 0
   jj = InStrRev(DATA2_STR, UCase(IIf(END_STR = "</BODY", "<BODY", END_STR)), ii)
   h = Abs(NO_ROWS)
   For l = 1 To h
      ii = InStrRev(DATA2_STR, "<TR", ii - 1)
      If ii < jj Then: GoTo ERROR_LABEL
   Next l
End Select

'Skip forward or backward the number of specified table cells
jj = ii
i = InStrRev(DATA2_STR, "<TR", jj)

If CELLS_INT = 0 Then
   j = InStr(jj, DATA2_STR, "</TR")
   k = 1
ElseIf CELLS_INT < 0 Then
   j = InStr(jj, DATA2_STR, "</TR")
   jj = j
   k = -CELLS_INT
Else
   k = CELLS_INT
   If END_STR <> "" Then
      j = InStr(jj, DATA2_STR, "</TR")
   Else
      j = Len(DATA2_STR)
   End If
End If

For l = 1 To k + LOOK_INT
   If CELLS_INT > 0 Then
```

```
      jj = InStr(jj, DATA2_STR, "<TD")
   Else
      jj = InStrRev(DATA2_STR, "<TD", jj)
   End If
   If jj = 0 Or jj < i Or jj > j Then GoTo ERROR_LABEL
   jj = InStr(jj, DATA2_STR, ">")
   If l >= k Then
      kk = InStr(jj, DATA2_STR, "</TD")
      'Extract cell contents and strip out HTML tags
      TEMP_VAL = Trim(Mid(DATA1_STR, jj + 1, kk - jj - 1))
      TEMP_VAL = Replace(Trim(TEMP_VAL), "<br>", Chr(10))
      Do
         ll = InStr(TEMP_VAL, "<")
         If ll = 0 Then Exit Do
         hh = InStr(ll, TEMP_VAL, ">")
         If hh = 0 Then Exit Do
         TEMP_VAL = IIf(ll = 1, "", Left(TEMP_VAL, ll - 1)) & Trim(Mid(TEMP_VAL & " ", hh + 1, 99999))
      Loop
      If TEMP_VAL <> "" Then Exit For
   End If
   If CELLS_INT < 0 Then
      jj = InStrRev(DATA2_STR, "<TD", jj) - 1
   End If
Next l

'If InStr(TEMP_VAL, "/") > 0 Then
'    PARSE_WEB_DATA_CELL_FUNC = TEMP_VAL
'Else
   'PARSE_WEB_DATA_CELL_FUNC = CONVERT_STRING_NUMBER_FUNC(TEMP_VAL)
'End If

PARSE_WEB_DATA_CELL_FUNC = TEMP_VAL

Exit Function
ERROR_LABEL:
PARSE_WEB_DATA_CELL_FUNC = ERROR_STR
```

```
End Function

'Function to return a financial statements data element from AdvFN
'=PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC("MMM","A",1,">Year End Date")

Function PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC(ByVal TICKER_STR As String, _
ByVal PERIOD_STR As String, _
ByVal CELLS_INT As Integer, _
Optional ByVal FIND1_STR As String = "", _
Optional ByVal FIND2_STR As String = "", _
Optional ByVal ERROR_STR As Variant = "Error", _
Optional ByVal TYPE_INT As Integer = 0) 'As Variant
'2012.02.04

Dim i As Long
Dim j As Long
Dim k As Long
Dim l As Long 'no periods

Dim ii As Long
Dim jj As Long

Dim URL_STR As String
Dim DATA_STR As String

Dim PAGE_STR As String
Dim LABEL1_STR As String
Dim LABEL2_STR As String
Dim SERVER_STR As String

On Error GoTo ERROR_LABEL

'Create labels for annual vs quarterly processing
TICKER_STR = CONVERT_YAHOO_TICKER_FUNC(TICKER_STR, "ADVFN")
Select Case UCase(PERIOD_STR)
Case "A"
```

```
   LABEL1_STR = "&mode=annual_reports"
   LABEL2_STR = "start_date"
Case "Q"
   LABEL1_STR = "&mode=quarterly_reports"
   LABEL2_STR = "istart_date"
Case Else
   ERROR_STR = "Improper period -- should be A or Q"
   GoTo ERROR_LABEL
End Select

SERVER_STR = PUB_ADVFN_SERVER_STR
URL_STR = "http://" & SERVER_STR & PUB_ADVFN_URL_STR '& "&btn="
URL_STR = URL_STR & "&symbol=" & TICKER_STR & LABEL1_STR

'Determine # of available periods and paging points (5 periods per page)
DATA_STR = SAVE_WEB_DATA_PAGE_FUNC(URL_STR, 0, False, 0, True, "name='" & LABEL2_STR & "'")
If DATA_STR = PUB_WEB_DATA_SYSTEM_ERROR_STR Then: GoTo ERROR_LABEL

i = InStr(UCase(DATA_STR), "</SELECT") - 25
j = InStr(i, UCase(DATA_STR), "='")
k = InStr(j, UCase(DATA_STR), "'>")
l = CInt(Mid(DATA_STR, j + 2, k - j - 2))
If CELLS_INT = 999 Then
   PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC = l + 1
   Exit Function
End If

If CELLS_INT > l + 1 Then GoTo ERROR_LABEL
ii = l - 5 * (Int((CELLS_INT - 1) / 5) + 1)
jj = (200 - CELLS_INT) Mod 5 + 1

Select Case ii
Case Is < 0
   If l < 5 Then
      PAGE_STR = ""
   Else
```

```
        PAGE_STR = "&" & LABEL2_STR & "=0"
    End If
    jj = jj + ii + 1
Case Is = l - 5
    PAGE_STR = ""
Case Else
    PAGE_STR = "&" & LABEL2_STR & "=" & (ii + 1)
End Select

'Return data element
PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC = RETRIEVE_WEB_DATA_CELL_FUNC(URL_STR & PAGE_STR, jj, FIND1_STR, FIND2_STR, , , , "</TABLE", ,
ERROR_STR)

Exit Function
ERROR_LABEL:
PARSE_ADVFN_WEB_DATA_ELEMENT_FUNC = ERROR_STR
End Function

'-------------------------------------------------------------------------------------------------------------*
'User defined function to extract an HTML table from a web page
'-------------------------------------------------------------------------------------------------------------*

Function PARSE_WEB_DATA_TABLE_FUNC(ByVal DATA1_STR As String, _
ByVal FIND1_STR As String, _
ByVal DIR1_INT As Integer, _
ByVal FIND2_STR As String, _
ByVal DIR2_INT As Integer, _
ByVal ROW_FLAG As Boolean, _
ByVal NROWS As Integer, _
ByVal NCOLUMNS As Integer) 'As Variant
'RCHGetHTMLTable
'2011.04.28

'User defined function to parse a web page
'FIND_BEGIN_STR: String to search for on web page to find start of table.
```

Coded by Nico
As of May 7, 2013

```
'BEGIN_DIRECTION_VAL: Number of <TABLE tags to searh for after finding
'the above string to find the start of the table. A negative number
'indicates to search backwards, positive number forwards.

'FIND_END_STR: String to search for on web page to find end of table.
'If blank, the "Find Begin" parameter will be reused.

'END_DIRECTION_VAL: Number of </TABLE tags to searh for after finding
'the above string to find the end of the table. A negative number
'indicates to search backwards, positive number forwards.

'This function returns an array of data (the HTML table), so it needs to be
'array-entered. The number of rows and columns for the range will depend on
'the size of the table you are retrieving and how much of that table you want
'to see.


Dim h As Long
Dim i As Long
Dim j As Long
Dim k As Long
Dim l As Long
Dim m As Long 'iColSpan

Dim hh As Long
Dim ii As Long
Dim jj As Long
Dim kk As Long
Dim ll As Long

'Dim NROWS As Long
'Dim NCOLUMNS As Long

Dim TEMP_VAL As Variant
Dim TEMP_STR As String
'Dim DATA1_STR As String
```

```
Dim DATA2_STR As String

On Error GoTo ERROR_LABEL

'------------------> Initialize returning array
ReDim TEMP_MATRIX(1 To NROWS, 1 To NCOLUMNS) As Variant
For i = 1 To NROWS: For j = 1 To NCOLUMNS: TEMP_MATRIX(i, j) = "": Next j: Next i
j = 20
ReDim MIN_ROW_ARR(1 To j)
ReDim MAX_ROW_ARR(1 To j)
ReDim MIN_COL_ARR(1 To j)
ReDim MAX_COL_ARR(1 To j)
For i = 1 To j
  MIN_ROW_ARR(i) = 0: MAX_ROW_ARR(i) = 0
  MIN_COL_ARR(i) = 0: MAX_COL_ARR(i) = 0
Next i

'Download web page
DATA2_STR = UCase(DATA1_STR)

'------------------> Look for the start and the end of the desired data table(s) on the page
ii = InStr(DATA2_STR, UCase(FIND1_STR))
For i = 1 To Abs(DIR1_INT)
    If DIR1_INT < 0 Then
        ii = InStrRev(DATA2_STR, IIf(ROW_FLAG, "<TR", "<TABLE"), ii - 1)
    Else
        ii = InStr(ii + 1, DATA2_STR, IIf(ROW_FLAG, "<TR", "<TABLE"))
    End If
Next i

'--------------------------------------------------------------------------------
'--------------------------------------------------------------------------------
'   Set the start of the HTML table to be the first "<TABLE" tag prior to that
'   string (i.e. -1).
'--------------------------------------------------------------------------------
'--------------------------------------------------------------------------------
```

```
'   Set the end of the HTML table to be the first "</TABLE" tag after that
'   string (i.e. 1).
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
'   Return the full table specified by and including the found "<TABLE" and
'   "</TABLE" tags.
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------

If FIND2_STR = "" Then FIND2_STR = FIND1_STR
hh = InStr(DATA2_STR, UCase(FIND2_STR))
For i = 1 To Abs(DIR2_INT)
   If DIR2_INT < 0 Then
      hh = InStrRev(DATA2_STR, IIf(ROW_FLAG, "</TR", "</TABLE"), hh - 1)
   Else
      hh = InStr(hh + 1, DATA2_STR, IIf(ROW_FLAG, "</TR", "</TABLE"))
   End If
Next i

'-----------------> Parse the table into rows and columns
h = 1
i = 0: j = 0
k = 1: l = 0

Do While True

   ii = InStr(ii, DATA2_STR, "<")
   If ii = 0 Or ii > hh Then Exit Do
   jj = InStr(ii, DATA2_STR, ">")
   If jj = 0 Or jj < ii Then Exit Do

   If Mid(DATA2_STR, ii, 6) = "<TABLE" Then
      l = 0                          ' Previous table cell start is not a data cell
      h = h + 1                      ' Start of new table
      MIN_ROW_ARR(h) = i             ' Save row that table began at
      If i > 0 And h > 2 Then i = i - 1   ' Need next row to start on current row
   ElseIf Mid(DATA2_STR, ii, 7) = "</TABLE" Then
      If h > 0 Then
         If h = 2 Then
            j = 0
         Else
            i = MIN_ROW_ARR(h)          ' Restore row that table begain at
            j = MAX_COL_ARR(h)          ' Set column to max column used by table
            MIN_ROW_ARR(h) = 0
            MAX_COL_ARR(h) = 0
         End If
         h = h - 1                   ' End of current table
      End If
   ElseIf Mid(DATA2_STR, ii, 3) = "<TR" Or Mid(DATA2_STR, ii, 6) = "<THEAD" Then
      k = k + 1                      ' Start of new row
      i = i + 1                    ' Point to next row of array
      MIN_COL_ARR(k) = j             ' Save column that row began at
   ElseIf Mid(DATA2_STR, ii, 4) = "</TR" Or Mid(DATA2_STR, ii, 7) = "</THEAD" Then
      MAX_COL_ARR(h) = MAXIMUM_FUNC(MAX_COL_ARR(h), j)
      j = MIN_COL_ARR(k)             ' Restore column that the row started at, for next row
      k = k - 1                      ' End of current row
      If k = 0 Then Exit Do
      MAX_ROW_ARR(k) = MAXIMUM_FUNC(MAX_ROW_ARR(k + 1), i)
      i = MAX_ROW_ARR(k)             ' Set row to max row used during this row
   ElseIf Mid(DATA2_STR, ii, 3) = "<TD" Or Mid(DATA2_STR, ii, 3) = "<TH" Then
      l = jj + 1                   ' Save possible start of cell data
      TEMP_STR = Mid(DATA2_STR, ii, jj - ii + 1)
      kk = InStr(TEMP_STR, "COLSPAN=")
      If kk > 0 Then
         ll = InStr(kk, TEMP_STR, " ")
         If ll = 0 Then ll = Len(TEMP_STR)
         m = CInt(Replace(Replace(Mid(TEMP_STR, kk + 8, ll - kk - 8), """", ""), "'", ""))
      Else
         m = 1
      End If
   ElseIf Mid(DATA2_STR, ii, 4) = "</TD" Or Mid(DATA2_STR, ii, 4) = "</TH" Then
      If l > 0 Then
```

```
         j = j + 1
         TEMP_STR = Mid(DATA1_STR, l, ii - l)
         Do While True
            kk = InStr(TEMP_STR, "<")
            If kk = 0 Then Exit Do
            ll = InStr(TEMP_STR, ">")
            If ll = 0 Then Exit Do
            TEMP_STR = Mid(TEMP_STR, 1, kk - 1) & Mid(TEMP_STR, ll + 1)
         Loop
         If i <= NROWS And j <= NCOLUMNS Then
            TEMP_VAL = Trim(Left(TEMP_STR, 255))
            'On Error Resume Next
            'TEMP_VAL = CDec(TEMP_VAL)
            'On Error GoTo 0
            TEMP_MATRIX(i, j) = TEMP_VAL
         End If
      j = j + m - 1
      l = 0
      End If
   End If
   ii = jj + 1
Loop

PARSE_WEB_DATA_TABLE_FUNC = TEMP_MATRIX

Exit Function
ERROR_LABEL:
PARSE_WEB_DATA_TABLE_FUNC = Err.number
End Function


Function EXTRACT_WEB_DATA_STRING_FUNC(ByVal DATA_STR As String, _
ByVal START_STR As String, _
ByVal END_STR As String, _
Optional ByVal LOOK_STR As String = "~") 'Same as smfStrExtr
```

```
Dim i As Long
Dim j As Long
Dim k As Long

On Error GoTo ERROR_LABEL

If START_STR = LOOK_STR Then
   i = 1
   k = 2
Else
   i = InStr(DATA_STR, START_STR) + Len(START_STR)
   k = i
   If i = Len(START_STR) Then: GoTo ERROR_LABEL
End If
If END_STR = LOOK_STR Then j = Len(DATA_STR) + 1 Else j = InStr(k, DATA_STR, END_STR)
If j = 0 Then: GoTo ERROR_LABEL

EXTRACT_WEB_DATA_STRING_FUNC = Mid(DATA_STR, i, j - i)

Exit Function
ERROR_LABEL:
EXTRACT_WEB_DATA_STRING_FUNC = ""
End Function


Function PARSE_WEB_DATA_PAGE_SYNTAX_FUNC(ByVal DATA_STR As String, _
Optional ByVal VERSION As Integer = 0)
'2011.04.27

On Error GoTo ERROR_LABEL

DATA_STR = Replace(DATA_STR, "&amp;", "&")
DATA_STR = Replace(DATA_STR, "&nbsp;<b>", "<b> ")
DATA_STR = Replace(DATA_STR, "&nbsp;", " ")
DATA_STR = Replace(DATA_STR, Chr(9), " ")
DATA_STR = Replace(DATA_STR, Chr(10), "")
DATA_STR = Replace(DATA_STR, Chr(13), "")
DATA_STR = Replace(DATA_STR, "&#48;", "0")
DATA_STR = Replace(DATA_STR, "&#49;", "1")
DATA_STR = Replace(DATA_STR, "&#50;", "2")
DATA_STR = Replace(DATA_STR, "&#51;", "3")
DATA_STR = Replace(DATA_STR, "&#52;", "4")
DATA_STR = Replace(DATA_STR, "&#53;", "5")
DATA_STR = Replace(DATA_STR, "&#54;", "6")
DATA_STR = Replace(DATA_STR, "&#55;", "7")
DATA_STR = Replace(DATA_STR, "&#56;", "8")
DATA_STR = Replace(DATA_STR, "&#57;", "9")
DATA_STR = Replace(DATA_STR, "&#150;", Chr(150))
DATA_STR = Replace(DATA_STR, "&#151;", "-")
DATA_STR = Replace(DATA_STR, "&mdash;", "-")
DATA_STR = Replace(DATA_STR, "&#160;", " ")
DATA_STR = Replace(DATA_STR, Chr(160), " ")

Select Case VERSION
Case 0
  DATA_STR = Replace(DATA_STR, "<td></td>", "<td> </td>")
  DATA_STR = Replace(DATA_STR, "<th></th>", "<th> </th>")
Case 1
  DATA_STR = Replace(DATA_STR, "<TH", "<td")
  DATA_STR = Replace(DATA_STR, "</TH", "</td")
  DATA_STR = Replace(DATA_STR, "<th", "<td")
  DATA_STR = Replace(DATA_STR, "</th", "</td")
End Select

PARSE_WEB_DATA_PAGE_SYNTAX_FUNC = DATA_STR


Exit Function
ERROR_LABEL:
PARSE_WEB_DATA_PAGE_SYNTAX_FUNC = Err.number
End Function
```

Coded by Nico
As of May 7, 2013

```
'This function removes the cache for all URLs downloaded with the XML object
'and calls the START_WEB_DATA_SYSTEM_FUNC, which resets the hash tables and
'recalculates the cells in excel.

Sub RESET_WEB_DATA_SYSTEM_FUNC()

'An artificial limit of web page retrievals before its "cache" area
'needs to be reset.  One way to reset the "cache" area is to use the
'RESET_WEB_DATA_SYSTEM_FUNC.

On Error Resume Next
If XML_CHECK_HTTP_CONNECTION_FUNC() = True Then
   Call REMOVE_CACHE_HISTORY_FUNC
   Call START_WEB_DATA_SYSTEM_FUNC

   If Val(Excel.Application.VERSION) < 10 Then
      Excel.Application.CalculateFull
   Else
      Excel.Application.CalculateFullRebuild
   End If
End If
End Sub


'This subroutine basically initiates the data gathering activity by creating collection and hash tables.
'First, the PUB_WEB_DATA_TABLES_FLAG is set false to indicate that no hash table has been created yet.
'Then it goes on creating the collection for web pages, the hash table for web data records and web data
'elements.  If LOAD_WEB_DATA_RECORD_FUNC  returns false, means there was error on cleaning the webpage
'data; proceed to error label and destroy the afforementioned objects, and exit the subroutine; otherwise,
'set the PUB_WEB_DATA_TABLES_FLAGS to true and exit the subroutine.

Sub START_WEB_DATA_SYSTEM_FUNC()

Dim i As Long

On Error GoTo ERROR_LABEL
```

```
PUB_WEB_DATA_TABLES_FLAG = False

For i = 1 To PUB_WEB_DATA_PAGES_VAL
'    PUB_WEB_DATA_PAGES_URL_ARR(i) = "": PUB_WEB_DATA_PAGES_ARR(i) = ""
    PUB_WEB_DATA_PAGES_MATRIX(i, 1) = "": PUB_WEB_DATA_PAGES_MATRIX(i, 2) = ""
Next i

Set PUB_WEB_DATA_PAGES_OBJ = New Collection

Set PUB_WEB_DATA_PAGES_HASH = New clsTypeHash
PUB_WEB_DATA_PAGES_HASH.SetSize PUB_WEB_DATA_PAGES_VAL
PUB_WEB_DATA_PAGES_HASH.IgnoreCase = False

Set PUB_WEB_DATA_RECORDS_HASH = New clsTypeHash
PUB_WEB_DATA_RECORDS_HASH.SetSize PUB_WEB_DATA_RECORDS_VAL
PUB_WEB_DATA_RECORDS_HASH.IgnoreCase = False

Set PUB_WEB_DATA_ELEMENTS_HASH = New clsTypeHash
PUB_WEB_DATA_ELEMENTS_HASH.SetSize PUB_WEB_DATA_ELEMENTS_VAL
PUB_WEB_DATA_ELEMENTS_HASH.IgnoreCase = False

If LOAD_WEB_DATA_RECORDS_FUNC() = False Then: GoTo ERROR_LABEL

PUB_WEB_DATA_TABLES_FLAG = True

Exit Sub
ERROR_LABEL:
PUB_WEB_DATA_TABLES_FLAG = False
Set PUB_WEB_DATA_PAGES_OBJ = Nothing
Set PUB_WEB_DATA_RECORDS_HASH = Nothing
Set PUB_WEB_DATA_PAGES_HASH = Nothing
Set PUB_WEB_DATA_ELEMENTS_HASH = Nothing
End Sub
```

```
Function LOAD_WEB_DATA_RECORDS_FUNC()

'The function begins with declaring i, j, k, SROW and NROWS as variables which will be used for
'indexing iterations. Variables DATA_STR, TEMP_STR and DATA_ARR will be used for performing operations
'on the data and SRC_URL_STR that will which will contain the web directory to the data file

'The function loops for each existing data file, which in this case is 9. In each iteration, the
'SRC_URL_STR is constructed to contain the web address of the data file. The function then downloads the file.

'If an error occurred while retrieving the file, DATA_ARR will return an error.

'A for loop is then used to trim each element and replace add-in functions from other websites with existing
'functions programmed in this module. Then the For loop is used, with NROWS - SROW iterations.
'First the DATA_ARR of index j is assigned to DATA_STR. If DATA_STR doesn't equal to the Chr(13) delimiter and
'Trim(DATA_ARR) doesn't return empty value (DATA_ARR doesn't contain spaces only) and DATA_STR doesn't equal to 0,
'then the function assigns the trimmed DATA_ARR to the DATA_ARR, then deletes all delimiters Chr(13) from the
'string. Next, "smfGetTagContent" substring is replaced with the return of PARSE_WEB_DATA_TAG_FUNC,
'"RCHGetTableCell" substring is replaced with the return of RETRIEVE_WEB_DATA_CELL_FUNC, "smfStrExtr" is replaced
'with the return of EXTRACT_WEB_DATA_STRING_FUNC.

'The TEMP_STR is then assigned the first character from DATA_ARR. If it is not empty, then the position of the
'PUB_WEB_DATA_ELEMENT_DELIM_STR position in the DATA_STR is assigned to k. Then the previous element is assigned
'to the TEMP_STR. If the Value of TEMP_STR doesn't equal 0, then the nested loop If checks, whether the
'PUB_WEBDATA_RECODRS_HASH of TEMP_STR exists, then deletes it and adds the substring from the DATA_STR positioned
'at k+1 to the PUB_WEBDATA_RECORDS_HASH and begins the next iteration.

Dim i As Long
Dim j As Long
Dim k As Long
Dim SROW As Long
Dim NROWS As Long

Dim DATA_STR As String
Dim TEMP_STR As String
Dim DATA_ARR As Variant
Dim SRC_URL_STR As String

On Error GoTo ERROR_LABEL

LOAD_WEB_DATA_RECORDS_FUNC = False

'-----------------------------------------------------------------------------
For i = 0 To PUB_WEB_DATA_FILES_VAL
'-----------------------------------------------------------------------------
   SRC_URL_STR = PUB_WEB_DATA_FILES_PATH_STR & CStr(i) & ".txt"
'    Debug.Print SRC_URL_STR
   DATA_STR = SAVE_WEB_DATA_PAGE_FUNC(SRC_URL_STR, 0, False, 0, False)
   If DATA_STR = PUB_WEB_DATA_SYSTEM_ERROR_STR Then: GoTo 1984
   DATA_ARR = Split(DATA_STR, Chr(10), -1, vbTextCompare)
   If IsArray(DATA_ARR) = False Then: GoTo 1983
   DATA_STR = ""
'-----------------------------------------------------------------------------
   SROW = LBound(DATA_ARR)
   NROWS = UBound(DATA_ARR)
'-----------------------------------------------------------------------------
   For j = SROW To NROWS
      DATA_STR = DATA_ARR(j)
      If DATA_STR <> Chr(13) And Trim(DATA_STR) <> "" And DATA_STR <> "0" Then
         DATA_STR = Trim(DATA_ARR(j))
         DATA_STR = Replace(DATA_STR, Chr(13), "")
         DATA_STR = Replace(DATA_STR, "smfGetTagContent", "PARSE_WEB_DATA_TAG_FUNC")
         DATA_STR = Replace(DATA_STR, "RCHGetTableCell", "RETRIEVE_WEB_DATA_CELL_FUNC")
         DATA_STR = Replace(DATA_STR, "smfStrExtr", "EXTRACT_WEB_DATA_STRING_FUNC")
         TEMP_STR = Left(DATA_STR, 1)
         If TEMP_STR <> "'" Then
           k = InStr(1, DATA_STR, PUB_WEB_DATA_ELEMENT_DELIM_STR)
           TEMP_STR = Left(DATA_STR, k - 1)
            If Val(TEMP_STR) <> 0 Then
               If PUB_WEB_DATA_RECORDS_HASH.Exists(TEMP_STR) = True Then
                  'Debug.Print SRC_URL_STR
                  'Debug.Print PUB_WEB_DATA_RECORDS_HASH(TEMP_STR)
```

```
                        'Debug.Print Mid(DATA_STR, k + 1)
                        PUB_WEB_DATA_RECORDS_HASH.Remove (TEMP_STR)
                    End If
                    PUB_WEB_DATA_RECORDS_HASH.Add TEMP_STR, Mid(DATA_STR, k + 1)
                End If
            End If
        End If
1983:
    Next j
1984:
'--------------------------------------------------------------------------
Next i
'--------------------------------------------------------------------------

LOAD_WEB_DATA_RECORDS_FUNC = True

Exit Function
ERROR_LABEL:
LOAD_WEB_DATA_RECORDS_FUNC = False
End Function

'This function takes INDEX_RNG as an input and ensures that INDEX_RNG is a 2D array with either
'multiple rows, or is a single cell. If either of these are untrue, they are adjusted.

'The headings string that has been set, is separated into an array by "," in order to set the
'first row of the matrix to these headings. Other than the headings row, the first column is
'the INDEX_VECTOR. For each element in the INDEX_VECTOR, RETRIEVE_WEB_ELEMENT_FUNC is used to
'download that element's value from the web.

Function RETRIEVE_WEB_DATA_RECORDS_FUNC(ByVal INDEX_RNG As Variant, _
Optional ByVal ERROR_STR As String = "--")

Dim i As Long
Dim j As Long
Dim k As Long
Dim NROWS As Long
Dim NCOLUMNS As Long
Dim HEADINGS_STR As String
Dim TEMP_MATRIX As Variant
Dim INDEX_VECTOR As Variant

On Error GoTo ERROR_LABEL

If IsArray(INDEX_RNG) = True Then
   INDEX_VECTOR = INDEX_RNG
   If UBound(INDEX_VECTOR, 1) = 1 Then: _
   INDEX_VECTOR = MATRIX_TRANSPOSE_FUNC(INDEX_VECTOR)
Else
   ReDim INDEX_VECTOR(1 To 1, 1 To 1)
   INDEX_VECTOR(1, 1) = INDEX_RNG
End If
NROWS = UBound(INDEX_VECTOR, 1)

NCOLUMNS = 14
HEADINGS_STR = "ID,VERSION,SOURCE,ELEMENT,P-URL,P-CELLS,P-FIND1,P-FIND2,P-FIND3,P-FIND4,P-ROWS,P-END,P-LOOK,P-TYPE,"
ReDim TEMP_MATRIX(0 To NROWS, 1 To NCOLUMNS)
i = 1
For k = 1 To NCOLUMNS
   j = InStr(i, HEADINGS_STR, ",")
   TEMP_MATRIX(0, k) = Mid(HEADINGS_STR, i, j - i)
   i = j + 1
Next k

For i = 1 To NROWS
   k = INDEX_VECTOR(i, 1)
   TEMP_MATRIX(i, 1) = k
   For j = 2 To NCOLUMNS: TEMP_MATRIX(i, j) = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TEMP_MATRIX(0, j), k, ERROR_STR): Next j
Next i

RETRIEVE_WEB_DATA_RECORDS_FUNC = TEMP_MATRIX

Exit Function
```

```
ERROR_LABEL:
RETRIEVE_WEB_DATA_RECORDS_FUNC = ERROR_STR
End Function


'-----------------------------------------------------------------------------------------------------------
' Macro to download data to fill in a 2-dimensional table
' 1. The upper left hand corner cell of the table needs to be named name "Ticker"
' 2. The cells below the "Ticker" cell should be filled in with ticker symbols, one per cell
' 3. The cells to the right of the "Ticker" cell should be filled with column titles
' 4. The cells above the column titles need to be filled in with formulas or element numbers.  Use
'    five tildas as a substitute for a ticker symbol.  For example, any of the following text
'    strings could be used to get "Market Capitalization" from Yahoo:
'    RETRIEVE_WEB_DATA_ELEMENT_FUNC(PUB_WEB_DATA_ELEMENT_LOOK_STR, 941)
'-----------------------------------------------------------------------------------------------------------

Function RNG_FILL_ELEMENTS_TABLE_FUNC(ByRef SRC_RNG As Excel.Range)
'2012.07.14

Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim l As Integer
Dim NSIZE As Integer
Dim NROWS As Integer
Dim NCOLUMNS As Integer

Dim TEMP_STR As String
Dim TICKER_STR As String
'Dim STATUS_STR As String
Dim FORMULA_STR As String

On Error GoTo ERROR_LABEL

'STATUS_STR = Excel.Application.DisplayStatusBar
'Excel.Application.DisplayStatusBar = True
```

```
RNG_FILL_ELEMENTS_TABLE_FUNC = False
If PUB_WEB_DATA_TABLES_FLAG = False Then: Call START_WEB_DATA_SYSTEM_FUNC

l = 20
NROWS = 999 ' Maximum number of rows to gather data for
NCOLUMNS = 200 ' Maximum number of columns to gather data for

NSIZE = Excel.Application.WorksheetFunction.CountA(Range(SRC_RNG.Offset(1, 0), SRC_RNG.Offset(NROWS, 0)))

For i = 1 To NROWS
   TICKER_STR = SRC_RNG.Offset(i, 0)
   If TICKER_STR = "" Then Exit For
   'Excel.Application.StatusBar = Round(100 * ((i - 1) / NSIZE), 0) & "% Completed " & _
                     " -- now processing " & TICKER_STR & " -- #" & i & " of " & NSIZE
   For j = 1 To NCOLUMNS
      FORMULA_STR = SRC_RNG.Offset(-1, j)
      If FORMULA_STR = "" Then Exit For
      If UCase(FORMULA_STR) <> "X" Then
         If IsNumeric(FORMULA_STR) Then
            If PUB_WEB_DATA_RECORDS_HASH(CStr(1)) = "" Then TEMP_STR = RETRIEVE_WEB_DATA_ELEMENT_FUNC("Source", 1)
            TEMP_STR = Split(PUB_WEB_DATA_RECORDS_HASH(CStr(0 + FORMULA_STR)), ";")(3 - 1)
            If Left(TEMP_STR, 1) = "=" Then
               FORMULA_STR = TEMP_STR
            Else
               FORMULA_STR = "RETRIEVE_WEB_DATA_ELEMENT_FUNC(" & """" & PUB_WEB_DATA_ELEMENT_LOOK_STR & """" & ", " & FORMULA_STR &
")"
            End If
         End If
         FORMULA_STR = Replace(FORMULA_STR, PUB_WEB_DATA_ELEMENT_LOOK_STR, TICKER_STR)
         For k = 1 To l
            TEMP_STR = Mid(PUB_WEB_DATA_ELEMENT_LOOK_STR, 1, 3)
            If InStr(FORMULA_STR, TEMP_STR) = 0 Then Exit For
            If InStr(FORMULA_STR, TEMP_STR & k & TEMP_STR) > 0 Then
               FORMULA_STR = Replace(FORMULA_STR, TEMP_STR & k & TEMP_STR, SRC_RNG.Offset(i, j).Offset(0, -k).Value2)
            End If
            Next k
            SRC_RNG.Offset(i, j) = Evaluate(FORMULA_STR)
        End If
    Next j
Next i

RNG_FILL_ELEMENTS_TABLE_FUNC = True

Exit Function
ERROR_LABEL:
RNG_FILL_ELEMENTS_TABLE_FUNC = False
'Excel.Application.StatusBar = False
'Excel.Application.DisplayStatusBar = STATUS_STR
End Function


'Subroutine to update a number of stock databases, one sheet per data source
'NSIZE = 20000 Number of data elements

Function RNG_UPDATE_WEB_DATA_RECORDS_FUNC(Optional ByVal NSIZE As Integer = 20000, _
Optional ByRef SRC_WBOOK As Excel.Workbook)

Dim i As Integer 'iTicker
Dim j As Integer 'iElement
Dim k As Integer 'iColumn
Dim l As Integer 'iSheet

Dim TICKER_STR As String
Dim SOURCE_STR As String
Dim VERSION_STR As String

Dim DATE_VAL As Variant

Dim DCELL As Excel.Range
Dim DSHEET As Excel.Worksheet
```

Coded by Nico
As of May 7, 2013

```
On Error GoTo ERROR_LABEL

If SRC_WBOOK Is Nothing Then: Set SRC_WBOOK = ActiveWorkbook

RNG_UPDATE_WEB_DATA_RECORDS_FUNC = False

VERSION_STR = RETRIEVE_WEB_DATA_ELEMENT_FUNC("Version")    ' Initialize the list of available elements
For Each DSHEET In SRC_WBOOK.Worksheets
    For j = 1 To NSIZE
        Select Case True
           Case DSHEET.Name = RETRIEVE_WEB_DATA_ELEMENT_FUNC("Source", j): Exit For
           Case j = NSIZE: GoTo 1985
        End Select
    Next j
    i = 2    ' Set initial ticker pointer
    Do While True
       i = i + 1       ' Go to next ticker symbol in list
       TICKER_STR = DSHEET.Cells(i, 1)  ' Get ticker symbol of company
       If TICKER_STR = "" Then GoTo 1985    ' No more ticker symbols

       DATE_VAL = DSHEET.Cells(i, 2)    ' Get date of last update for company
       If DATE_VAL <> 0 Then GoTo 1984 ' Valid date, no need to update
       DSHEET.Cells(i, 2) = Date     ' Update the last update date

       j = 0 ' Set initial element pointer
       k = 2 ' Set initial column pointer
       l = 1  ' Set sheet pointer for 256+ element sources
       Set DCELL = DSHEET

       Do While True
          j = j + 1     ' Go to next available element
          SOURCE_STR = RETRIEVE_WEB_DATA_ELEMENT_FUNC("Source", j)   ' Get data source of element
          If SOURCE_STR = "EOL" Then GoTo 1984
          If SOURCE_STR <> DSHEET.Name Then GoTo 1983    ' Not an applicable element for worksheet

          k = k + 1       ' Go to next output column
          If DCELL.Cells(2, k) = "" Then
            DCELL.Cells(1, k) = j
            DCELL.Cells(2, k) = RETRIEVE_WEB_DATA_ELEMENT_FUNC("Element", j)
          End If

          'Excel.Application.StatusBar = "Now updating ticker " & TICKER_STR & " on worksheet " & _
          DCELL.Name

          DCELL.Cells(i, k) = RETRIEVE_WEB_DATA_ELEMENT_FUNC(TICKER_STR, j)

          If k = 256 Then
            l = l + 1
            Set DCELL = SRC_WBOOK.Worksheets(SOURCE_STR & "_" & l)
            DCELL.Cells(i, 1) = DSHEET.Cells(i, 1)
            DCELL.Cells(i, 2) = DSHEET.Cells(i, 2)
            k = 2
          End If
          'Call TickerReset
1983:
        Loop 'Next_Element

1984:
    Loop 'Next_Company

1985:
Next DSHEET 'Next_WorkSheet

'    Excel.Application.StatusBar = False

RNG_UPDATE_WEB_DATA_RECORDS_FUNC = True

Exit Function
ERROR_LABEL:
RNG_UPDATE_WEB_DATA_RECORDS_FUNC = False
End Function
```