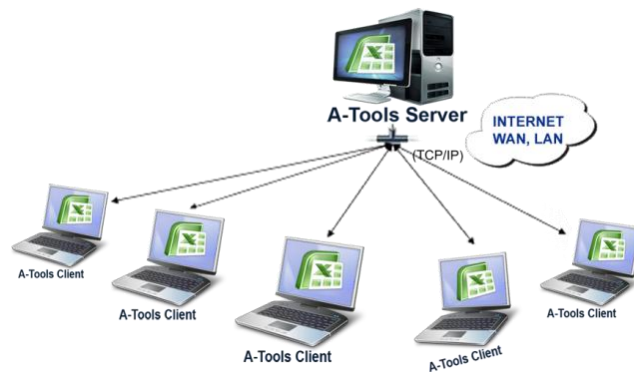


Công ty cổ phần BLUESOFTS



# Lập trình Add-in A-Tools

Chia sẻ tập tin Excel qua mạng LAN, Internet  
Trích lọc dữ liệu và làm báo cáo động trong Microsoft Excel

BLUESOFTS

Tác giả: ThS. Nguyễn Duy Tuân

## MỤC LỤC

<b>I. Lập trình hàm mảng trong Excel và A-Tools .....</b>	<b>4</b>
1. Nhúng file thư viện "addinatools.dll" vào file Excel của bạn.....	4
2. Tạo hàm mảng cùng đối tượng BSFormulaArray .....	5
a. Thế nào là một hàm mảng.....	5
b. Các thành phần của đối tượng BSFormulaArray .....	5
c. Cấu trúc hàm mảng .....	7
d. Các ví dụ tạo hàm mảng.....	7
e. Cấu trúc hàm BSFormulaArray.Add() và đối tượng BSFormulaInfo.....	9
f. Lập trình sự kiện với hàm mảng .....	14
<b>II. Lập trình truy xuất dữ liệu Excel qua mạng.....</b>	<b>17</b>
1. Thành phần của đối tượng BSNetwork .....	18
2. Lập trình truy xuất dữ liệu Excel qua mạng.....	18
3. Kết hợp giữa BSNetwork và BSFormulaArray .....	28
4. Lập trình sự kiện truy xuất dữ liệu giữa các máy trong mạng .....	29
<b>III. Lập trình truy xuất cơ sở dữ liệu bên ngoài vào Excel trong mạng LAN, Internet ..</b>	<b>29</b>
1. Tạo DBKEY tại máy chủ - Thiết lập kết nối CSDL.....	29
2. Kết nối CSDL trong máy chủ từ máy khách.....	30
3. Kết nối và truy vấn CSDL, thực hiện lệnh SELECT .....	31
4. Thực hiện lệnh INSERT – Chèn dữ liệu vào CSDL trong máy chủ từ máy khách .....	32
5. Thực hiện lệnh UPDATE, DELETE – Sửa, xóa dữ liệu .....	33
6. Lấy thông tin về CSDL đang mở .....	34
a. Lấy danh sách các bảng/table dữ liệu trong CSDL đã mở .....	34
b. Lấy danh sách các cột/field trong bảng dữ liệu .....	35
c. Lấy danh sách các thủ tục trong CSDL đã mở.....	36

## Giới thiệu

A-Tools là một công cụ (add-in) chạy trong môi trường Microsoft Excel. A-Tools có các chức năng chính sau:

**Chia sẻ tập tin Excel qua mạng LAN, Internet cho nhiều người dùng chung:** Để chia sẻ tập dữ liệu Excel cho nhiều người dùng trong mạng, nếu không có công cụ thì chỉ có một phương pháp là share full thư mục chứa file Excel đó (cách của Excel). Theo cách này, các máy trong mạng đều có thể truy cập vào copy và thậm chí xóa hẳn tập dữ liệu này. Người chủ của tập Excel này chỉ có thể gửi trọn cả tập mà rất khó che dấu những sheets hay vùng dữ liệu riêng của mình. Còn các vấn đề khác như cho người dùng truy cập vào vùng dữ liệu nhưng có các quyền: chỉ đọc, soạn thảo, giới hạn bao nhiêu người được kết nối,...không thể làm được.

Với Add-in A-Tools cho phép soạn thảo và quản trị dữ liệu Excel qua mạng. Công cụ này cho phép người dùng quản trị dữ liệu Excel qua mạng một cách mạnh mẽ, có thể kết nối các bảng tính, vùng dữ liệu qua TCP/IP (không phải share full thư mục), quản trị các kết nối tới vùng dữ liệu, phân quyền cho từng nhóm người dùng, khi dữ liệu tại một máy thay đổi thì các máy khác trong mạng được cập nhật ngay lập tức (sau khoảng ~ 1 mili giây).

Tìm hiểu chi tiết chức năng này đọc tài liệu: <C:\A-Tools\Documents\A-Tools-Help.pdf>

**Trích lọc dữ liệu và làm báo cáo động trong Microsoft Excel:** Tạo bảng dữ liệu động bởi việc sử dụng các hàm với ngôn ngữ truy vấn cơ sở dữ liệu (CSDL) T-SQL. Cho phép kết nối với các CSDL bên trong và bên ngoài như Excel, Access, Foxpro, MS SQL, MySQL, Firebird,... vào bảng tính Excel. Với công cụ SQL Builder sẽ giúp người dùng tạo công thức truy vấn dữ liệu một cách dễ dàng và hiệu quả. Người dùng có thể tạo ra những báo cáo trong bảng tính Excel có độ phức tạp cao, được lọc theo nhiều điều kiện, liên kết với các nguồn dữ liệu khác nhau, đảm bảo độ chính xác, ổn định và tốc độ xử lý nhanh.

Hướng dẫn chi tiết: [C:\A-Tools\Documents\A-Tools\\_Query\\_Report\\_Help.pdf](C:\A-Tools\Documents\A-Tools_Query_Report_Help.pdf)

**Lập trình cùng Add-in A-Tools:** Ngoài giao diện người dùng, Add-in A-Tools còn hỗ trợ các hàm API và các đối tượng COM như **BSNetwork**, **BSFormulaArray**, **BSConnection** để những người phát triển ứng dụng lập trình kết hợp cùng Add-in A-Tools nhằm khai thác tối đa năng lực của Add-in A-Tools, tăng sức mạnh cho ứng dụng của mình.

### **Các chức năng phụ:**

- Tạo công thức thống kê kết hợp với nhiều điều kiện, công thức mảng.
- Chuyển đổi mã văn bản từ TCVN3, VNI sang UNICODE và ngược lại.
- Đọc số thành chữ tiếng Việt, tiếng Anh
- ...

## Cài đặt Add-in A-Tools

Bạn hãy vào địa chỉ dưới đây để download bộ cài về máy và tiến hành cài đặt như các phần mềm khác.

<http://www.bluesofts.net/Products/AddIns/A-Tools/Index.htm>

Sau khi cài đặt xong, bạn mở Microsoft Excel sẽ thấy menu của A-Tools như hình dưới đây:



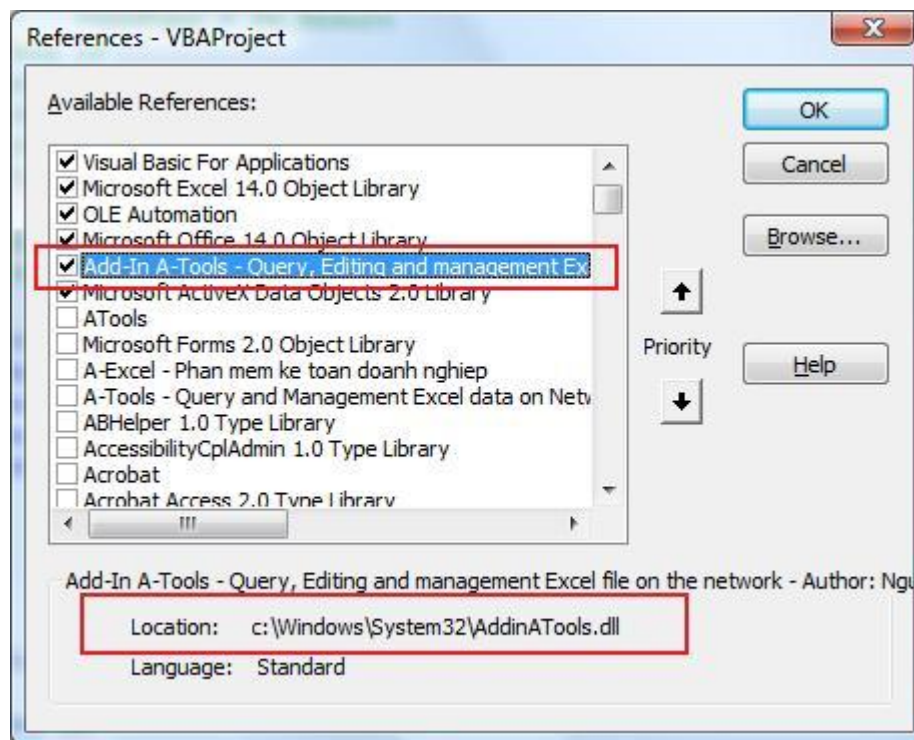
## I. Lập trình hàm mảng trong Excel và A-Tools

Trong A-Tools cung cấp hàm BS\_SQL(), truy vấn CSDL các loại và trả về một mảng giá trị. Hàm này được dùng để tạo ra các loại sổ sách rất linh hoạt. Để tìm hiểu cách dùng hàm BS\_SQL bạn mở các file trong thư mục cài đặt "C:\A-Tools\HELP & DEMOS", đặc biệt file "SoketoanSQL.xls" là một ví dụ điển hình về việc dùng hàm BS\_SQL để tạo ra một sổ kế toán động cho một doanh nghiệp với tốc độ nhanh và linh hoạt.

A-Tools là công cụ mở cho phép nhà phát triển lập trình để tạo ra các hàm với đặc tính giống như hàm BS\_SQL(). Để tạo được hàm mảng, chúng ta thực hiện những chuẩn bị sau:

### 1. Nhúng file thư viện "addinatools.dll" vào file Excel của bạn

Mở file Excel. Nhấn ALT+F11 để vào môi trường lập trình VBE (Visual Basic Editor). Vào menu A-Tools->References..., khi hiện ra hộp thoại bạn hãy chọn mục Add-in A-Tools, nếu không thấy hãy chọn nút "Browse..." và chọn file ở đường dẫn "C:\Windows\System32\addinatools.dll". Kết quả bạn thực hiện được như hình dưới đây:



## 2. Tạo hàm mảng cùng đối tượng BSFormulaArray

### a. Thế nào là một hàm mảng

Hàm mảng là hàm trả về một mảng giá trị, mảng có thể là 1 hoặc 2 chiều. Trong Excel có một hàm mảng tên là Table(), trong A-Tools là BS\_SQL().

Ví dụ về một hàm mảng:

```
Function GetArr() As Variant
    Dim Arr(2, 1)
    Arr(0, 0) = "Q1": Arr(0, 1) = 1
    Arr(1, 0) = "Q2": Arr(1, 1) = 2
    Arr(2, 0) = "Q3": Arr(2, 1) = 3
    GetArr = Arr
End Function
```

(trong lập trình VB, nếu không muốn xuống dòng ta gõ dấu ":" ngay ở đầu đoạn code.)

Trong bảng tính bạn đặt công thức: =GetArr() rồi ENTER theo cách làm thông thường thì kết quả chỉ trả về phần tử đầu tiên của mảng giá trị là "Q1". Muốn nó trả về một mảng giá trị thì phải làm như sau:

Vì mảng trên là gồm 3 dòng và 2 cột (mảng 2 chiều: dòng & cột, phần tử chạy từ 0), ta bôi đen/chọn vùng ô gồm 3 dòng và 2 cột, sau đó gõ hàm

=GetArr() rồi nhấn CTRL + SHIFT + ENTER kết quả sẽ trả về toàn bộ mảng như hình dưới đây:

	A1				
	A	B	C	D	E
1	Q1	1			
2	Q2	2			
3	Q3	3			
4					

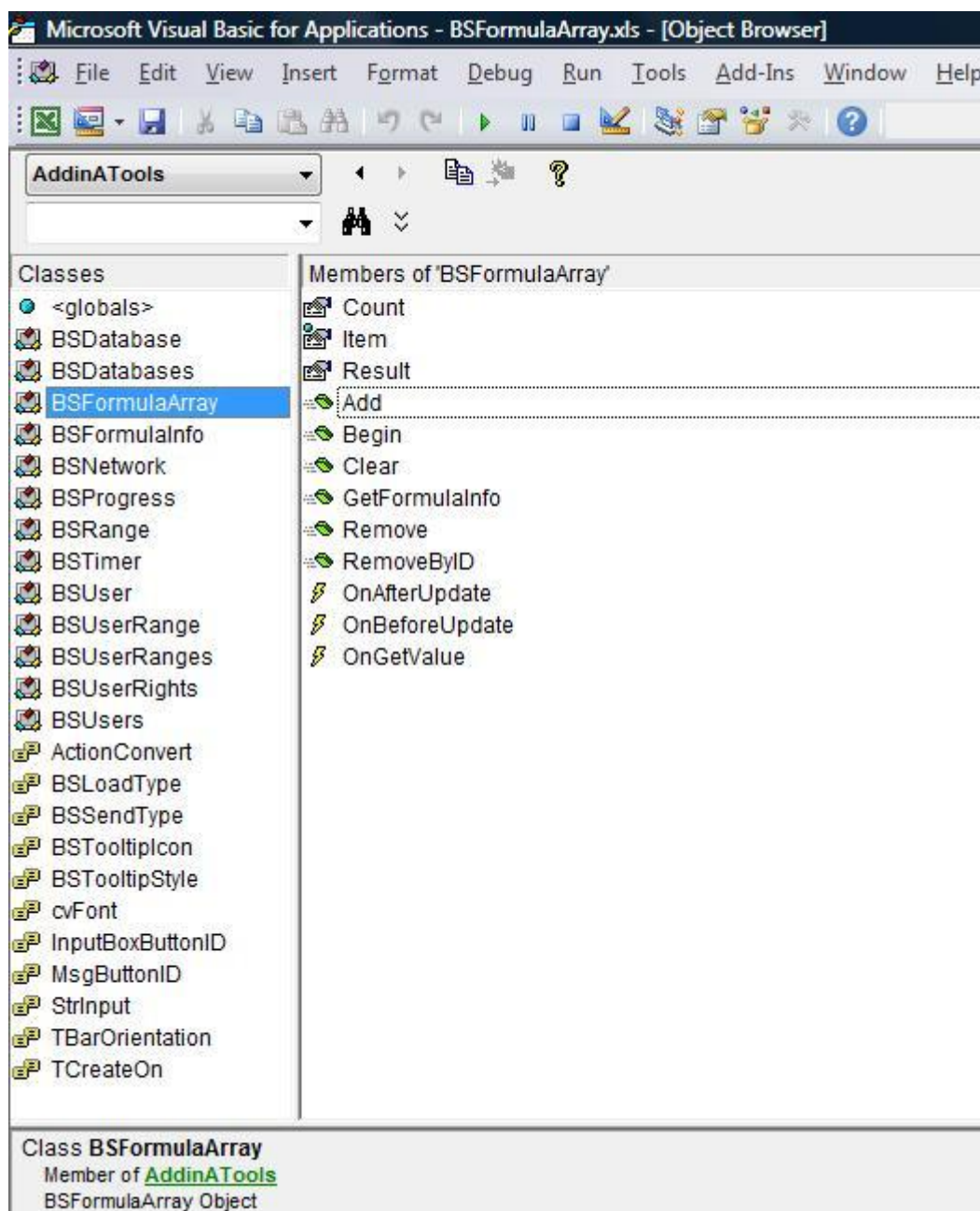
Nếu vùng chọn mà số dòng hoặc số cột lớn hơn số phần tử của mảng kết quả thì các ô thừa đó sẽ có giá trị "#N/A", nếu vùng chọn thiếu so với số phần tử của mảng thì kết quả thiếu giá trị. Vậy theo cách làm thông thường của Excel chúng ta phải chọn vùng đặt công thức với số dòng và cột phải đúng bằng số phần tử dòng và cột của mảng trả về. Đây là một sự cứng nhắc và người dùng khó mà tuân thủ đúng, vì mảng kết quả trả về của hàm với số phần tử sẽ thay đổi tùy thuộc vào hoàn cảnh nào đó, ví dụ hàm trả về danh sách nhân viên của một phòng, nếu phòng A thì danh sách là 10, phòng B thì danh sách là 5...

Add-in A-Tools cung cấp đối tượng BSFormulaArray để làm việc này một cách linh hoạt và dễ dàng. Chỉ cần đặt công thức, nhấn ENTER là kết quả trả về cả mảng đủ với số phần tử của mảng. Với đối tượng BSFormulaArray nhà phát triển có thể tạo ra bất kể hàm mảng nào một cách thuận tiện, có thể truy vấn dữ liệu từ nhiều loại CSDL như Excel, Access, Foxpro, MySQL, MS SQL Server,....

### b. Các thành phần của đối tượng BSFormulaArray

Để tìm hiểu các thành phần của A-Tools, trong VBE nhấn F2 để mở cửa sổ quản lý đối tượng "Object Browser". Khi cửa sổ hiện lên, góc đỉnh bên trái bạn chọn AddinATools trong combo box.

Các thành phần của đối tượng BSFormulaArray như hình dưới đây:



Tên thành phần	Mô tả	Ghi chú
Begin()	<u>Loại:</u> Hàm, giá trị trả về là kiểu Boolean. <u>Ứng dụng:</u> dùng để kiểm tra hàm của người dùng có phải trạng thái bắt đầu chạy hay không. Nếu kết quả là TRUE tức là bắt đầu và bạn hãy viết các đoạn code cho hàm và nạp thông tin về hàm cùng mảng kết quả vào hàm Add().	Bắt buộc chạy lúc ban đầu
Add()	<u>Loại:</u> Hàm, giá trị trả về là kiểu Variant chứa mảng kết quả. <u>Ứng dụng:</u> dùng để nạp thông tin về hàm của bạn và kết quả cần trả về của hàm.	Chạy sau hàm Begin()
Result	<u>Loại:</u> Thuộc tính, giá trị trả về là kiểu Variant chứa mảng kết quả mà hàm cần trả về bảng tính. <u>Ứng dụng:</u> dùng để trả về hàm của gọi. Nó được dùng	Chạy sau hàm Begin()



	khi hàm Begin() = FALSE	nếu Begin() = FALSE
--	-------------------------	---------------------------

## c. Cấu trúc hàm mảng

Là một hàm mảng, bạn bắt buộc làm theo cấu trúc dưới đây:

```
Function YourFunc() As Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        Dim Result 'It is an array, SQL statement or Recordset
        'Fill value to Result
        'Các đoạn code để tạo ra mảng kết quả của bạn
        YourFunc = fa.Add(fi, Result)
    Else
        YourFunc = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function
```

## d. Các ví dụ tạo hàm mảng

Các ví dụ dưới đây đều nằm trong file "**BSFormulaArray.xls**" tại đường dẫn "[C:\A-Tools\HELP & DEMOS\A-Tools VBA Programming](#)". Để xem và chạy thử bạn hãy mở file này ra.

+ Hàm **GetArray()** - Ví dụ đơn giản hàm trả về mảng gồm 3 dòng và 2 cột.

```
Function GetArray() As Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        Dim Arr(2, 1)
        Arr(0, 0) = "Q1": Arr(0, 1) = 1
        Arr(1, 0) = "Q2": Arr(1, 1) = 2
        Arr(2, 0) = "Q3": Arr(2, 1) = 3
        GetArray = fa.Add(fi, Arr)
    Else
        GetArray = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function
```

Bây giờ bạn ra bảng tính, gõ

=GetArray() rồi ENTER bạn sẽ thấy tự Excel điền cho bạn một mảng gồm 3 dòng và 2 cột.

+ Ví dụ hàm **GetRangeValue()** – Trả về một mảng giá trị, mảng này lấy từ một vùng dữ liệu nơi khác

```
Function GetRangeValue() As Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        GetRangeValue = fa.Add(fi, Range("data").Value)
    Else
        GetRangeValue = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function
```

+ Ví dụ hàm **DVLOOKUP()** – Tìm kiếm và trả về mảng giá trị tìm được. Hàm **VLOOKUP** trong Excel chỉ trả về duy nhất 1 giá trị tìm được, việc tìm là tìm trên cột đầu tiên. Hàm **DVLOOKUP()** tôi viết dưới đây có thể tìm giá trị "*LookupValue*" tại cột có vị trí "*LookupColIndex*" (ngầm định là 1) trong bảng "*LookupTable*" tại vị trí cột, nếu tìm được trả về các giá trị thỏa mãn tại vị trí cột "*ResultColIndex*",

```
Function DVLOOKUP(ByVal LookupValue As Variant, ByVal LookupTable As Range,
    ByVal ResultColIndex As Long, Optional ByVal LookupColIndex As Long = 1) As
Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    Dim I As Long, LastRow As Long, ArrCount As Long
    On Error GoTo lbEndFunc
    ReDim Arr(0)
    ArrCount = -1
    If fa.Begin Then
        'LastRow = LookupTable.Rows.Count
        LastRow = LookupTable.Cells(LookupTable.Rows.Count + 1,
            LookupColIndex).End(xlUp).Row - LookupTable.Row + 1
        For I = 1 To LastRow
            If LookupTable.Cells(I, LookupColIndex) = LookupValue Then
                ArrCount = ArrCount + 1
                ReDim Preserve Arr(ArrCount)
                Arr(ArrCount) = LookupTable.Cells(I, ResultColIndex)
            End If
        Next I
        DVLOOKUP = fa.Add(fi, Arr)
    Else
        DVLOOKUP = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function
```

Tại worksheet đặt công thức:  
=DVLOOKUP(G9,A4:D100,1,2) rồi nhấn ENTER



Kết quả sẽ trả về mảng giá trị thỏa mãn tại ô G9  
Xem ví dụ trong file **BSFormulaArray.xls**

G12		fx {=DVLOOKUP(G9,A4:D100,1,2)}					
	A	B	C	D	E	F	G
1							
2							
3	BILL#	CODE	NAME	QTY		CODE	NAME
4	01	A	Tivi	5		A	Tivi
5	02	B	Computer	10		B	Computer
6	03	A	Tivi	3		C	Mouse
7	04	A	Tivi	8			
8	05	B	Computer	4			
9	06	C	Mouse	6		CODE:	A
10							
11							BILL#
12							01
13							03
14							04

e. Cấu trúc hàm BSFormulaArray.Add() và đối tượng BSFormulaInfo

Function Add(AFormulaInfo As BSFormulaInfo, Aresult As Variant) As Variant

Hàm Add() được dùng để nạp thông tin hàm cho A-Tools, hàm trả về giá trị kiểu Variant cho hàm người dùng. Hàm gồm hai đối số.

**AformulaInfo** nhận kiểu **BSFormulaInfo** để lưu các thông tin cũng như tùy chọn cho hàm.

**Aresult** nhận kiểu Variant có thể là:

- + Mảng kết quả mà hàm cần trả về bảng tính
- + Chuỗi lệnh truy vấn CSDL SQL, ví dụ "SELECT \* FROM DATA"
- + Recordset – Đối tượng trong ADODB chứa bảng truy vấn bởi lệnh SQL.

**BSFormulaInfo** chứa các thuộc tính như bảng dưới đây:

Classes	Members of 'BSFormulaInfo'
<globals>	AutoNames
BSDatabase	DBKEY
BSDatabases	Formula
BSFormulaArray	HeaderRow
<b>BSFormulaInfo</b>	ID
BSNetwork	Insert
BSProgress	Name
BSRange	OnAfterUpdate
BSTimer	OnBeforeUpdate
BSUser	OnGetValue
BSUserRange	OptionStr
BSUserRanges	Result
BSUserRights	SERVERSOURCE
BSUsers	Tag

Tên thành phần	Mô tả	Ghi chú
DBKEY	<u>Loại:</u> Thuộc tính, kiểu chuỗi <u>Ứng dụng:</u> Nhận mã DBKEY để kết nối với CSDL bên ngoài. DBKEY được tạo trong menu A-Tools->DBKEY-Kết nối CSDL bên ngoài. Chỉ phải dùng thuộc tính này nếu tham số Aresult trong hàm Add là câu lệnh SQL và truy vấn tới CSDL bên ngoài. Nếu không khai báo DBKEY A-Tools sẽ lấy dữ liệu tại tệp Excel đang mở.	Dùng với với SQL
HeaderRow	<u>Loại:</u> Thuộc tính, kiểu Boolean <u>Ứng dụng:</u> dùng nếu Aresult trong hàm Add là câu lệnh SQL hoặc Recordset. Nếu là TRUE (ngầm định) thì bảng kết quả có dòng tiêu đề, nếu FALSE thì bảng kết quả không có.	Dùng với SQL, Recordset
NAME	<u>Loại:</u> Thuộc tính, kiểu chuỗi <u>Ứng dụng:</u> Cho phép đặt tên vùng kết quả theo tên được khai báo. Bình thường trong Excel ta phải làm: chọn (bôi đen) vùng, nhấn CTRL+F3 và tạo tên.	
AUTONAMES	<u>Loại:</u> Thuộc tính, kiểu chuỗi nhận hàm AutoNames() để đặt tên tự động. <u>Ứng dụng:</u> Được dùng nếu Aresult trong hàm Add là câu lệnh SQL hoặc Recordset. Nếu là "AutoNames()" thì tất cả các cột dữ liệu trong bảng kết quả sẽ được đặt tên theo tên của trường dữ liệu. Nếu là "AutoNames(TEN_BANG)" thì bảng kết quả được đặt tên (define name) là TEN_BANG, bạn thay "TEN_BANG" bằng một cái tên hợp lệ bất kỳ. Nếu là "AutoNames(TEN_BANG, PREFIX_)" thì bảng kết quả được đặt tên là "TEN_BANG", các cột trong bảng đều được đặt tên với tiếp đầu ngữ là "PREFIX_". Việc đặt tên rất hữu ích khi vùng dữ liệu được đặt tên lại được dùng trong một công thức khác.	Dùng với SQL, Recordset
Insert	<u>Loại:</u> Thuộc tính, kiểu Boolean <u>Ứng dụng:</u> Nếu là TRUE, mảng kết quả của hàm trả về bảng tính với phương thức chèn dữ liệu, các dữ liệu đứng sau sẽ được đẩy xuống dưới. Nếu là FALSE (ngầm định), mảng kết quả sẽ được ghi đè.	
SERVERSOURCE	<u>Loại:</u> Thuộc tính, kiểu Boolean <u>Ứng dụng:</u> dùng nếu Aresult trong hàm Add là câu lệnh SQL. Nếu là TRUE thì dữ liệu trong bảng kết quả được truy vấn dữ liệu từ máy chủ. Nếu FALSE (ngầm định) thì dữ liệu trong bảng kết quả được truy vấn dữ liệu từ máy đang làm việc.	Dùng với SQL
OptionStr	<u>Loại:</u> Thuộc tính, kiểu chuỗi <u>Ứng dụng:</u> Cho phép gán giá trị cho các thành phần của BSFormulaInfo trong chuỗi, các cặp gán cách nhau bởi dấu ";". Ví dụ "DBKEY=MDB; INSERT=YES" Tương đương với việc gán  fi.DBKEY = "MDB" fi.Insert = True	Tham khảo cách dùng hàm BS_SQL() của A-Tools

ID, Tag	<p><u>Loại:</u> Thuộc tính, kiểu Long</p> <p><u>Ứng dụng:</u> Cho phép gán một số nguyên bất kỳ để làm dấu hiệu nhận biết hàm. Chỉ cần thiết nếu dùng kết hợp sự kiện.</p>	
OnBeforeUpdate	<p><u>Loại:</u> Thuộc tính, kiểu chuỗi</p> <p><u>Ứng dụng:</u> Gán tới tên một thủ tục khi đó thủ tục này được chạy trước khi bảng kết quả được cập nhật lên bảng tính.</p> <p>Cấu trúc thủ tục quy định như sau:</p> <pre>Sub DoBeforeUpdate(ByVal OldDataTable As Range, ByVal NewDataTable As Range, ByVal DataArray)</pre> <p>End Sub</p>	Thủ tục có cấu trúc quy định.
OnAfterUpdate	<p><u>Loại:</u> Thuộc tính, kiểu chuỗi</p> <p><u>Ứng dụng:</u> Gán tới tên một thủ tục khi đó thủ tục này được chạy sau khi bảng kết quả được cập nhật trên bảng tính.</p> <p>Cấu trúc thủ tục quy định như sau:</p> <pre>Sub DoAfterUpdate(ByVal DataTable As Range)</pre> <p>End Sub</p>	Thủ tục có cấu trúc quy định.
OnGetValue	<p><u>Loại:</u> Thuộc tính, kiểu chuỗi</p> <p><u>Ứng dụng:</u> Gán tới tên một hàm khi đó hàm này được chạy trong khi giá trị của từng bản ghi trong Recordset đưa về mảng kết quả. Hàm này có thể giữ nguyên hoặc thay đổi giá trị trước khi nó được điền vào bảng tính.</p> <p>Cấu trúc thủ tục quy định như sau:</p> <pre>Function GetValue(ByVal DataArray, ByVal Row As Integer, ByVal Column As Integer, ByVal Value As Variant)</pre> <p>End Sub</p>	Thủ tục có cấu trúc quy định. Dùng với SQL, Recordset
Formula	<p><u>Loại:</u> Thuộc tính, kiểu chuỗi, chỉ đọc</p> <p><u>Ứng dụng:</u> lưu cấu trúc của hàm đang chạy.</p>	Dùng kết hợp với các sự kiện trong Class Module

+ Ví dụ hàm **GetXISQL()** – Truy vấn dữ liệu trong tệp Excel đang mở bằng câu lệnh SQL. Đối số Aresult của hàm Add là chuỗi khai báo SQL.

```
Function GetXISQL(ByVal SQL As String) As Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        GetXISQL = fa.Add(fi, SQL)
    Else
        GetXISQL = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function
```

Bây giờ bạn có thể lấy toàn bộ dữ liệu trong vùng data tại trong tệp Excel đang mở bằng công thức:

=GetXISQL("select \* from data")

+ Ví dụ hàm **GetBySQL()** – Kết quả trả về bởi câu lệnh truy vấn vào CSDL Access "Examble.mdb" được kết nối tới DBKEY có tên MDB. Đối số Aresult của hàm Add là chuỗi khai báo SQL.

```
Function GetBySQL(ByVal SQL As String) As Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        fi.DBKEY = "MDB"
        fi.HeaderRow = True
        'the same ==> fi.OptionStr = "DBKEY=MDB;HR=YES"
        GetBySQL = fa.Add(fi, SQL)
    Else
        GetBySQL = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function
```

Bây giờ bạn lập công thức:

=GetBySQL("select \* from nkc") rồi ENTER

Kết quả là toàn bộ dữ liệu trong bảng NKC trong CSDL Access "Examble.mdb".

+ Ví dụ hàm **GetByRecordset()** – Kết quả trả lấy từ đối tượng ADODB.Recordset với câu lệnh SQL. Đối số Aresult của hàm Add là đối tượng Recordset.

```
Function GetByRecordset(ByVal SQL As String) As Variant
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        Set cnn = GetConnDB(ActiveWorkbook.FullName)
        Set Rs = GetRS(SQL, cnn)
        fi.HeaderRow = True ' <=> fi.OptionStr = "HR=YES"
        GetByRecordset = fa.Add(fi, Rs)
    Else
        GetByRecordset = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        MsgBox Err.Description, vbCritical
    End If
End Function
```

Hàm GetConnDB() và GetRS() được khai báo trong tệp BSFormulaArray.xls

Ứng dụng trên bảng tính. Đặt công thức  
=GetByRecordset("select \* from data")

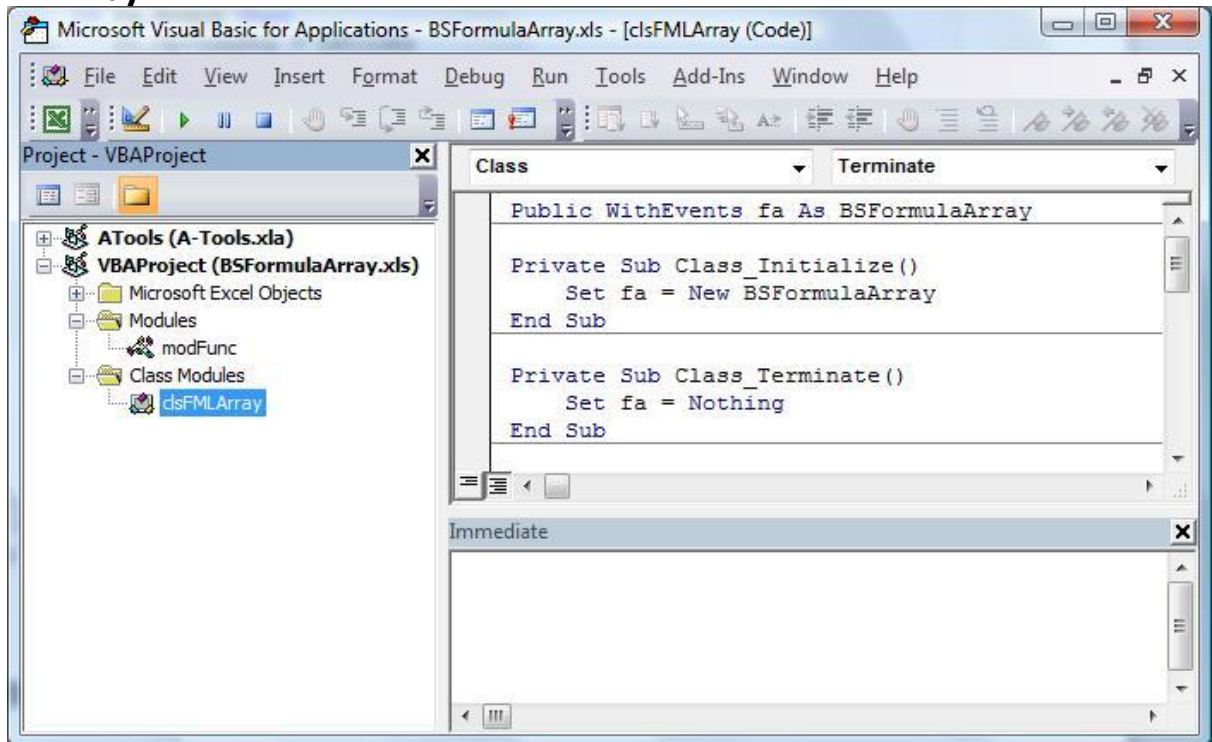
+ Ví dụ hàm **GetSQL()** – Kết quả trả về lấy từ câu lệnh SQL và chuỗi tham số tùy chọn OPTIONS. Đối số Aresult của hàm Add() là chuỗi khai báo SQL. Hàm **GetSQL()** tương tự hàm **BS\_SQL()** của Add-in A-Tools.

```
Function GetSQL(ByVal SQL As String, Optional ByVal OPTIONS As String = vbNullString) As Variant
    'Function GetSQL() is the same BS_SQL() in A-Tools
    Dim fa As New BSFormulaArray
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If fa.Begin Then
        fi.OptionStr = OPTIONS
        GetSQL = fa.Add(fi, SQL)
    Else
        GetSQL = fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    Set fa = Nothing
    If Err <> 0 Then
        MsgBox Err.Description, vbCritical
    End If
End Function
```

f. Lập trình sự kiện với hàm mảng

Phương pháp dùng Class Module

Bước 1: tạo Class Module. Trong VBE, vào menu Insert chọn Class Module. Đặt tên là **clsFMLArray**



Bước 2: Soạn thảo các mã lệnh khởi hoạt, giải phóng bộ nhớ và các thủ tục sự kiện trong class **clsFMLArray**

```
Public WithEvents fa As BSFormulaArray

Private Sub Class_Initialize()
    Set fa = New BSFormulaArray
End Sub

Private Sub Class_Terminate()
    Set fa = Nothing
End Sub

Private Sub fa_OnBeforeUpdate(ByVal OldRange As Object, _
    ByVal NewRange As Object, _
    Result As Variant, _
    ByVal AFormulaInfo As AddinATools.IBSFormulaInfo)
    If AFormulaInfo.ID <> 100 Then Exit Sub
    'Run for function GetBySqlWithEvents() only!
    MsgBox "OldRange: " & OldRange.Address & Chr(13) & "NewRange: " &
    NewRange.Address & Chr(13) & "Replace CODE=""A"" with ""AAAAA"" in event
    ""OnGetValue""", vbInformation, "Event: OnBeforeUpdate"
    Result(0, 0) = "www.atoolspro.com"
End Sub

Private Sub fa_OnAfterUpdate(ByVal NewRange As Object, _
    ByVal AFormulaInfo As AddinATools.IBSFormulaInfo)
    If AFormulaInfo.ID <> 100 Then Exit Sub
    'Run for function GetBySqlWithEvents() only!
    MsgBox "NewRange: " & NewRange.Address, vbInformation, "Event: OnAfterUpdate"
```



```

End Sub

Private Sub fa_OnGetValue(ByVal Row As Long, _
                        ByVal Column As Long, _
                        Value As Variant, _
                        ByVal AFormulaInfo As AddinATools.IBSFormulaInfo)
    If AFormulaInfo.ID <> 100 Then Exit Sub
    'Run for function GetBySqlWithEvents() only!
    If Column = 1 And Row > 0 Then
        If Value = "A" Then
            Value = "AAAAA"
        End If
    End If
End Sub
End Sub

```

Bước 3: Khởi tạo biến kiểu clsFMLArray và tạo hàm để chạy các sự kiện trên.  
+ Ví dụ hàm **GetBySqlWithEvents** . Trong Module khai báo như sau:

```

Dim faEvent As New clsFMLArray

Function GetBySqlWithEvents(ByVal SQL As String) As Variant
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If faEvent.fa.Begin Then
        If Not faEvent Is Nothing Then Set faEvent = Nothing
        Set faEvent = New clsFMLArray
        Set cnn = GetConnDB(ActiveWorkbook.FullName)
        Set Rs = GetRS(SQL, cnn)
        fi.ID = 100
        GetBySqlWithEvents = faEvent.fa.Add(fi, Rs)
    Else
        GetBySqlWithEvents = faEvent.fa.Result
    End If
lbEndFunc:
    Set fi = Nothing
    If Err <> 0 Then
        'MsgBox Err.Description, vbCritical
    End If
End Function

```

Khác với các hàm ở ví dụ trước ta dùng biến fa As BSFormulaArray, hàm ta dùng biến faEvent As clsFMLArray

Nếu không muốn dùng Class Module thì ta có thể khai báo tên các thủ tục sự kiện vào trong các thuộc tính của BSFormulaInfo như sau:

+ Ví dụ hàm **GetBySqlWithEvents2**

```

Function GetBySqlWithEvents2(ByVal SQL As String) As Variant
    Dim fi As New BSFormulaInfo
    On Error GoTo lbEndFunc
    If faEvent.fa.Begin Then
        If Not faEvent Is Nothing Then Set faEvent = Nothing
        Set faEvent = New clsFMLArray
        Set cnn = GetConnDB(ActiveWorkbook.FullName)
        Set Rs = GetRS(SQL, cnn)
        fi.ID = 100
        fi.OnGetValue = "GetValueForSpeedup"
        'GetValueForSpeedup is faster then event "OnGetValue" in class
    End If
End Function

```

```
fi.OnBeforeUpdate = "DoBeforeUpdate"  
fi.OnAfterUpdate = "DoAfterUpdate"  
'If assign fi.OnGetValue = "GetValueForSpeedup" then event "OnGetValue" in  
class do not run.  
GetBySqlWithEvents2 = faEvent.fa.Add(fi, Rs)  
Else  
    GetBySqlWithEvents2 = faEvent.fa.Result  
End If  
lbEndFunc:  
Set fi = Nothing  
If Err <> 0 Then  
    'MsgBox Err.Description, vbCritical  
End If  
End Function  
  
Function GetValueForSpeedup(ByVal DataArray, ByVal Row As Long, ByVal Column As  
Long, ByVal Value As Variant)  
    'DataArray: two-dimensional array, lowest element at index 0  
    'Gia tri tai dong 2 cot 3 = DataArray(1,2) , tren bang tinh thi la gia tri  
    cua dong 2 cot 3  
    If Column = 1 And Row > 0 Then  
        If Value = "A" Then  
            GetValueForSpeedup = "AAAAA"  
        End If  
    End If  
End Function  
  
Sub DoBeforeUpdate(ByVal OldDataTable As Range, ByVal NewDataTable As Range,  
ByVal DataArray)  
    MsgBox "DoBeforeUpdate"  
End Sub  
  
Sub DoAfterUpdate(ByVal DataTable As Range)  
    MsgBox "DoAfterUpdate"  
End Sub
```

Câu lệnh SQL trong A-Tools được thừa kế từ ngôn ngữ chuẩn T-SQL và được phát triển thêm các hàm khai báo cùng chuỗi SQL để làm được những báo cáo phức tạp. Để khai thác được cách dùng SQL trong A-Tools, người dùng nên đọc LẦN LƯỢT các file trong thư mục cài đặt A-Tools.

[Bai 1 - Huong dan su dung ham BS SQL](#)

[Bai 2 - Cac ham ho tro nhom ham Database Functions.xls](#)

[Bai 3 Hàm CSDL \(Database Functions\).xls](#)

[Bai 4 - Huong dan tao bao cao.xls](#)

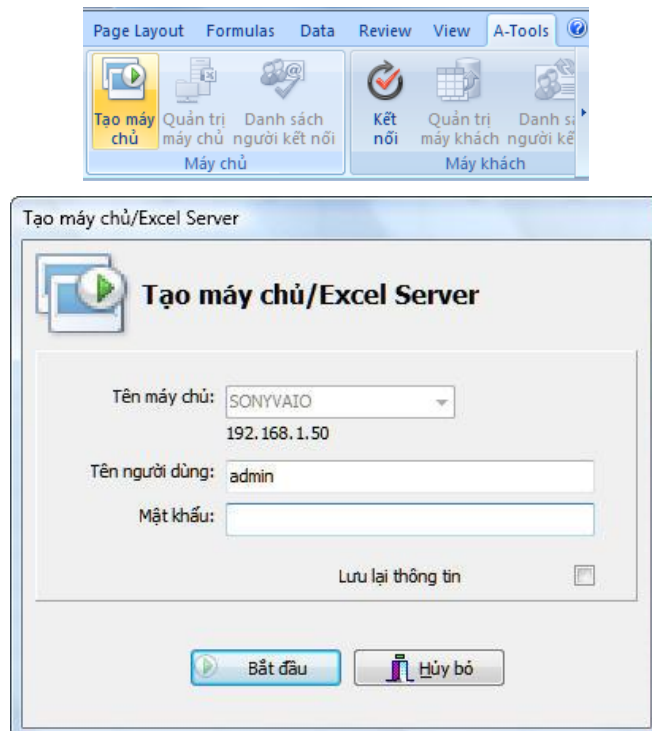
Để tìm hiểu sâu phương pháp lập trình sự kiện chạy cùng hàm mảng, các bạn nên tìm hiểu hàm BS\_SQL() trong tệp "[Report 3 So ke toan A-Tools VBA.xls](#)" (đường dẫn C:\A-Tools\HELP & DEMOS\A-Tools VBA Programming).

## II. Lập trình truy xuất dữ liệu Excel qua mạng

Để chạy các mã lập trình kết nối mạng dưới đây, máy chủ phải được khởi tạo trước.

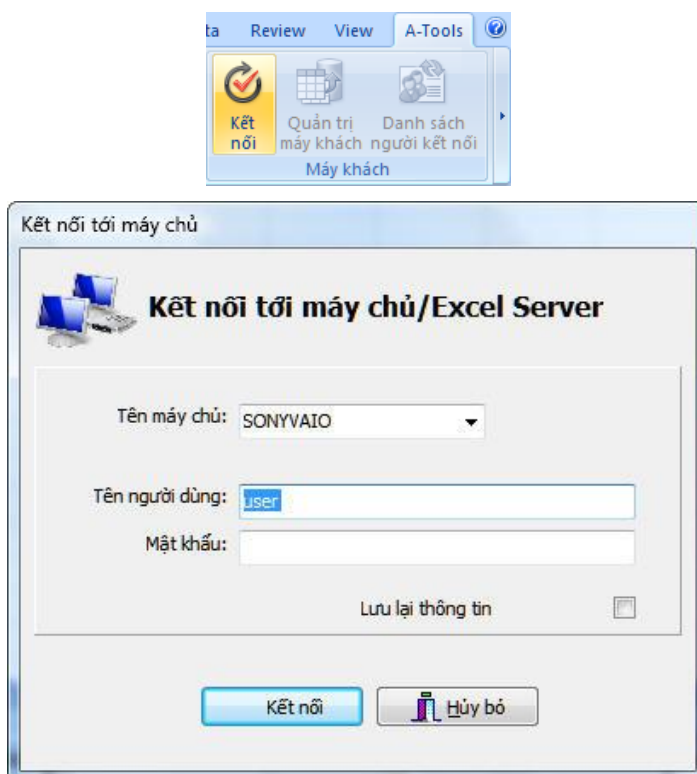
### Khởi tạo máy chủ

Từ menu A-Tools, bấm chọn chức năng “Tạo máy chủ”



### Máy khách kết nối tới máy chủ

Từ menu A-Tools, bấm chọn chức năng “Kết nối”



Để lập trình truy xuất dữ liệu qua mạng, A-Tools cung cấp nhà lập trình đối tượng **BSNetwork**.

## 1. Thành phần của đối tượng BSNetwork



## 2. Lập trình truy xuất dữ liệu Excel qua mạng

Để tiện cho thực hành, các bạn mở tệp "[BSNetworkInfo.xls](#)" để xem mã nguồn và chạy thử.

### + StartServer()

Loại: Hàm, giá trị trả về là kiểu Boolean.

Ứng dụng: dùng để tạo máy chủ để chia sẻ và quản trị dữ liệu cho các máy khách từ xa. Nếu tạo máy chủ thành công hàm trả về TRUE, ngược lại là FALSE.

Cấu trúc:

```
Function StartServer([UserName As String], [Password As String]) As Boolean
```

UserName, Password: Tên và mật khẩu truy cập để tạo/chạy máy chủ. Khi cài đặt A-Tools cung cấp sẵn hai user là "admin" và "user", mật khẩu để trống. Thiết lập UserName & Password, vào màn hình "Quản trị máy chủ".

Nếu bỏ qua tất cả các tham số thì khi chạy hàm sẽ hiện một hộp thoại để nhập UserName và Password.

### + StopServer()

Loại: Thủ tục.

Ứng dụng: dùng để tắt máy chủ. Khi máy chủ tắt thì các máy khách từ xa không thể kết nối tới nó.

Cấu trúc: Thủ tục này không có tham số.

**+ IsRunning**

Loại: Thuộc tính. Kiểu giá trị Boolean, chỉ đọc.

Ứng dụng: dùng để kiểm tra trạng thái máy chủ hay máy khách có đang chạy không. Nếu đang chạy giá trị trả về TRUE.

Cấu trúc: Thuộc tính này không có tham số.

**+ IsServer**

Loại: Thuộc tính. Kiểu giá trị Boolean, chỉ đọc.

Ứng dụng: dùng để kiểm tra trạng A-Tools đang chạy có phải là máy chủ không. Nếu là máy chủ giá trị trả về TRUE.

Cấu trúc: Thuộc tính này không có tham số.

**+ LocalHost**

Loại: Thuộc tính. Kiểu giá trị String, chỉ đọc.

Ứng dụng: cho biết tên của máy tính đang chạy.

Cấu trúc: Thuộc tính này không có tham số.

**+ LocalIP**

Loại: Thuộc tính. Kiểu giá trị String, chỉ đọc.

Ứng dụng: cho biết địa chỉ IP của máy tính đang chạy trong mạng LAN.

Cấu trúc: Thuộc tính này không có tham số.

**+ RemoteHost**

Loại: Thuộc tính. Kiểu giá trị String, chỉ đọc. Chỉ dùng cho máy khách.

Ứng dụng: cho biết tên của máy chủ đang được kết nối.

Cấu trúc: Thuộc tính này không có tham số.

**+ RemoteIP**

Loại: Thuộc tính. Kiểu giá trị String, chỉ đọc. Chỉ dùng cho máy khách.

Ứng dụng: cho biết địa chỉ IP của máy chủ đang được kết nối trong mạng.

Cấu trúc: Thuộc tính này không có tham số.

**+ Port**

Loại: Thuộc tính. Kiểu giá trị Long, chỉ đọc.

Ứng dụng: cho biết cổng được dùng để kết nối giữa máy chủ và máy khách. Ngầm định khi cài đặt A-Tools, cổng là 1711. Máy chủ và các máy khách muốn kết nối với nhau phải chung một cổng. Người dùng có thể thay đổi cổng bằng cách mở màn hình "Tùy chọn" để thay đổi.

Cấu trúc: Thuộc tính này không có tham số.

Ví dụ: khởi tạo máy chủ

Tạo một Module trong VBE. Dán các đoạn code sau và chạy.

```
#If VBA7 Then
Declare PtrSafe Function MsgBoxW2 Lib "AddinATools.dll" (ByVal MSG As Variant, Optional ByVal uType As VbMsgBoxStyle = vbOKOnly, Optional ByVal Caption As Variant = vbNullString, Optional ByVal hwnd As Long = 0) As VbMsgBoxResult
Declare PtrSafe Function ValueToXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal Value As Variant) As Variant
Declare PtrSafe Function ValueFromXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal XmlData As Variant) As Variant
#Else
Declare Function MsgBoxW2 Lib "AddinATools.dll" (ByVal MSG As Variant, Optional ByVal uType As VbMsgBoxStyle = vbOKOnly, Optional ByVal Caption As Variant = vbNullString, Optional ByVal hwnd As Long = 0) As VbMsgBoxResult
```

```

Declare Function ValueToXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal Value As Variant) As Variant
Declare Function ValueFromXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal XmlData As Variant) As Variant
#End If

'+ Hàm MsgBoxW2 để hiển thị hộp thoại thông báo giống hàm MsgBox của VB6, VBA
nhưng nó cho phép hiển thị chuỗi unicode và tên các nút bấm theo ngôn ngữ lựa
chọn trong Add-in A-Tools.
'+ Hàm ValueToXml trả về chuỗi có cấu trúc XML chứa giá trị cần gửi đi. Dữ liệu
truyền qua mạng của A-Tools là theo XML. Hàm này được dùng để tạo chuỗi có cấu
trúc XML và gửi đi bởi hàm Send().
'+ Hàm ValueFromXml nhận giá trị từ chuỗi có cấu trúc XML. Chuỗi này được trả về
bởi hàm Receive, Wait, thủ tục sự kiện OnReceive().

Function RunServer() As Boolean
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    If Not XNet.IsRunning Then
        RunServer = XNet.StartServer("admin", "")
        If Not RunServer Then Exit Function
        MsgBox "Server is started.", vbInformation
    End If
lbEndProc:
    Set XNet = Nothing
    If Err <> 0 Then
        MsgBoxW2 Err.Description, vbCritical, Application.Caption
    End If
End Function

Sub StopServer()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    If XNet.IsRunning And XNet.IsServer Then
        XNet.StopServer
        MsgBox "Server is stopped.", vbInformation
    End If
lbEndProc:
    Set XNet = Nothing
    If Err <> 0 Then
        MsgBoxW2 Err.Description, vbCritical, Application.Caption
    End If
End Sub

```

Hàm **MsgBoxW2()** của A-Tools cho phép hiển thị chuỗi Unicode và tùy chỉnh ngôn ngữ theo thiết lập của A-Tools. Cách dùng tương tự như hàm MsgBox() của VBA.

#### + Connect()

Loại: Hàm. Kiểu giá trị Boolean.

Ứng dụng: dùng để kết nối tới máy chủ. Nếu kết nối thành công giá trị trả về TRUE.

Cấu trúc:

```

Function Connect([Host As String], [UserName As String], [Password As String])
As Boolean

```

Host: là tên hoặc địa chỉ IP của máy chủ, cũng có thể là tên miền/domain (ví dụ: [www.bluesofts.net](http://www.bluesofts.net))

UserName, Password: Tên và mật khẩu để tạo kết nối máy chủ. Tên và mật khẩu được tạo trong máy chủ, người dùng phải biết mới kết nối được.



**+ Disconnect()**

Loại: Thủ tục.

Ứng dụng: dùng để ngắt kết nối tới máy chủ.

Cấu trúc: Thủ tục này không có tham số.

**+ Connected**

Loại: Thuộc tính. Kiểu giá trị Boolean, chỉ đọc.

Ứng dụng: dùng để kiểm tra trạng máy khách đã được kết nối thành công chưa. Nếu kết nối thành công giá trị trả về TRUE.

Cấu trúc: Thuộc tính này không có tham số.

**+ RemoteBusy**

Loại: Thuộc tính. Kiểu giá trị Boolean. Chỉ đọc.

Ứng dụng: Nếu trả về TRUE tức là máy chủ đang bận, ngược lại có thể nhận và gửi dữ liệu.

**+ StopRead**

Loại: Thuộc tính. Kiểu giá trị Boolean.

Ứng dụng: Nếu là TRUE sẽ dừng chế độ tự động nhận dữ liệu của máy tính từ xa truyền tới, FALSE là tự động truyền và sự kiện **OnReceive()** sẽ nhận dữ liệu. Máy tính từ xa có thể là máy chủ hoặc máy khách. Thường sử dụng thuộc tính này khi máy khách cần tự nhận dữ liệu từ máy chủ, bởi hàm **Wait()** hoặc **Receive()**.

**+ Pause**

Loại: Thủ tục.

Ứng dụng: Tạm dừng chế độ tự động nhận dữ liệu của máy tính từ xa truyền tới. Máy tính từ xa có thể là máy chủ hoặc máy khách. Thường sử dụng thuộc tính này khi máy khách cần tự nhận dữ liệu từ máy chủ, bởi hàm **Wait()** hoặc **Receive()**.

Cấu trúc: Thủ tục này không có tham số.

**+ Resume**

Loại: Thủ tục.

Ứng dụng: Khôi phục chế độ tự động nhận dữ liệu của máy tính từ xa truyền tới. Sự kiện **OnReceive()** sẽ tự động nhận dữ liệu mỗi khi có máy gửi đến. Máy tính từ xa có thể là máy chủ hoặc máy khách. Thường sử dụng thủ tục này nếu trước đó đã dùng **Pause**.

Cấu trúc: Thủ tục này không có tham số.

**+ Resume**

Loại: Hàm.

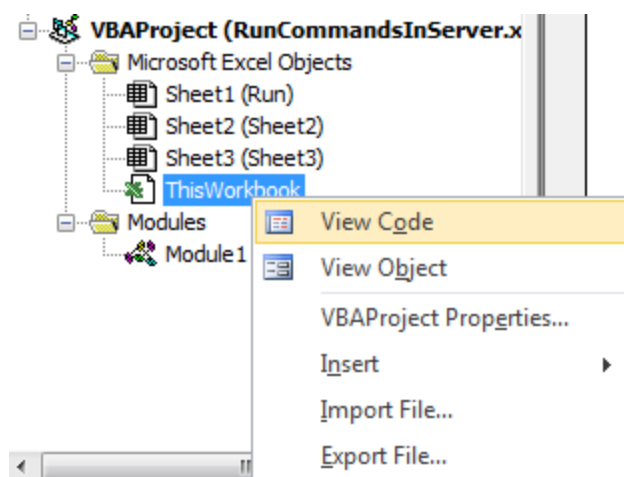
Cấu trúc:

**Function Wait**(MessageID As Long, XmlData As String, [MessageCheckID As Long = 4], [TimeOut As Long = -1]) As Boolean

Ứng dụng: Hàm sẽ nhận dữ liệu gửi đến từ máy chủ với thông điệp chỉ định ở MessageCheckID, khoảng thời gian đợi là "TimeOut" (-1 là không giới hạn thời gian). Nếu nhận được giá trị hàm Wait sẽ trả về TRUE, ngược lại là FALSE. MessageID và XmlData là dữ liệu máy chủ gửi về cho máy khách. Hàm này hữu dụng khi máy khách cần "hỏi đáp" với máy chủ để giải quyết một công việc cụ thể.

Ví dụ: Máy khách yêu cầu máy chủ chạy một ứng dụng trong máy chủ. Ví dụ phần mềm Calculator(Calc.exe).

Bước 1. Tạo tập tin Excel. Vào môi trường VBE, nhấp chuột phải trên đối tượng "ThisWorkbook", chọn "View Code". Để xem và soạn thảo code cho tập tin này.



Option Explicit

Public WithEvents XNet As BSNetwork 'Khai báo biến đối tượng BSNetwork để làm việc với mạng. Bạn cần references tới thư viện AddinATools.dll mới có đối tượng này.

Dim sError As String

Private Sub Workbook\_BeforeClose(Cancel As Boolean)

'Sự kiện này chạy khi mở workbook

Auto\_Close 'Auto run before this workbook closing

End Sub

Private Sub Workbook\_Open()

'Sự kiện này chạy khi mở workbook

Auto\_Open 'Auto run when this workbook opening

End Sub

Private Sub XNet\_OnReceive(ByVal MessageID As Long, ByVal XmlData As String, ByVal User As AddinATools.BSUser)

'Sự kiện này chạy mỗi khi có dữ liệu từ máy tính khác gửi đến

'This event is fired when remote machine (clients or server) send data

If Not XNet.IsServer Then Exit Sub 'Chỉ chạy nếu là máy chủ

If MessageID = XNM\_SHELL Then 'Run in server

'Thông điệp XNM\_SHELL được gửi đến từ máy khách cho máy chủ với App khách yêu cầu chạy.

If OpenAppOrURL(ValueFromXml(XNM\_KEY\_APP, XmlData)) <> 0 Then

'Tell client that is fine

User.Send XNM\_RESULT, ValueToXml(XNM\_KEY\_VALUE, True)

Else

'Tell client that is error

User.Send XNM\_RESULT, ValueToXml(XNM\_KEY\_VALUE, False) & ValueToXml(XNM\_KEY\_PROMPT,

sError)

End If

End If

End Sub

Function OpenAppOrURL(App As String) As Double 'Hàm chạy App

On Error GoTo lbEnd

OpenAppOrURL = Shell(App)

Exit Function

lbEnd:

sError = Err.Description

End Function

Bước 2. Tạo module1 để khai báo các hằng số, thủ tục, các hàm

```
Option Explicit
'Author: Nguyen Duy Tuan - duy tuan@bluesofts.net
'Website: www.atoolspro.com - www.bluesofts.net
'Used Message for Send(), OnReceive procedure.
'Public Const XNM_USER = 1000
'server/client receive user's data with message ID >=1000 ONLY!
Public Const XNM_SHELL = XNM_USER + 1
Public Const XNM_RESULT = XNM_USER + 2
Public Const XNM_KEY_APP = "APP"
Public Const XNM_KEY_VALUE = "VALUE"
Public Const XNM_KEY_PROMPT = "PROMPT"

#If VBA7 Then
Declare PtrSafe Function MsgBoxW2 Lib "AddinATools.dll" (ByVal MSG As Variant, Optional ByVal uType As VbMsgBoxStyle = vbOKOnly, Optional ByVal Caption As Variant = vbNullString, Optional ByVal hwnd As Long = 0) As VbMsgBoxResult
Declare PtrSafe Function ValueToXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal Value As Variant) As Variant
Declare PtrSafe Function ValueFromXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal XmlData As Variant) As Variant
#Else
Declare Function MsgBoxW2 Lib "AddinATools.dll" (ByVal MSG As Variant, Optional ByVal uType As VbMsgBoxStyle = vbOKOnly, Optional ByVal Caption As Variant = vbNullString, Optional ByVal hwnd As Long = 0) As VbMsgBoxResult
Declare Function ValueToXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal Value As Variant) As Variant
Declare Function ValueFromXml Lib "AddinATools.dll" (ByVal KEY As Variant, ByVal XmlData As Variant) As Variant
#End If

Function GetMessageID(ByVal XmlData As String) As Long
    If XmlData <> "" Then
        GetMessageID = CLng(ValueFromXml("MSG", XmlData))
    End If
End Function

Sub Auto_Open()
    Auto_Close 'Giải phóng đối tượng trước khi khởi tạo mới
    'Tạo biến XNet
    Set ThisWorkbook.XNet = New BSNetwork
End Sub

Sub Auto_Close()
    'Giải phóng đối tượng
    Set ThisWorkbook.XNet = Nothing
End Sub

Sub RunCalc() 'Chạy Calc từ nút lệnh. Nếu là máy khách thì yêu cầu máy chủ chạy
    If ThisWorkbook.XNet.IsServer Then
        Shell "calc.exe"
    Else
        SendCommand "calc.exe"
    End If
End Sub

'Gửi yêu cầu/lệnh tới máy chủ để chạy "Calc"
Function SendCommand(ByVal App As String) As Boolean ' Run in client
```

```

'Client send command to server to run App in server
On Error GoTo lbEnd 'Important for resume reading data from server
Dim MsgID As Long
Dim XmlData As String, Ok As Boolean
'Tạm ngừng chế độ tự nhận dữ liệu
ThisWorkbook.XNet.Pause 'Let client stop auto read data from server.
'Gửi thông điệp XNM_SHELL với tên APP cần chạy cho máy chủ
ThisWorkbook.XNet.Send XNM_SHELL, ValueToXml(XNM_KEY_APP, App)
'In server, event OnReceive() get this message (XNM_SHELL) and send feedback
'message XNM_RESULT from server. Client should catch message from server to tell user
'Dợi máy chủ thông báo kết quả nhận lệnh và chạy
If ThisWorkbook.XNet.Wait(MsgID, XmlData, XNM_RESULT, 5000) Then 'Check for sending
    Ok = CBool(ValueFromXml(XNM_KEY_VALUE, XmlData))
    If Ok Then
        SendCommand = True
        MsgBox "Success!", vbInformation
    Else
        MsgBox ValueFromXml(XNM_KEY_PROMPT, XmlData), vbCritical
    End If
Else
    'Lỗi không nhận được giá trị từ máy chủ trong khoảng thời gian 5000 ms
    MsgBox "Time out.", vbCritical
End If
lbEnd:
ThisWorkbook.XNet.Resume 'Let client continue auto read data from server
End Function

```

Ví dụ trên nằm trong tập tin:

<C:\A-Tools\HELP & DEMOS\A-Tools VBA Programming\Network\RunCommandsInServer.xls>

### + Databases

Loại: Thuộc tính. Kiểu class BSDatabases, chỉ đọc.

Ứng dụng: chứa danh sách CSDL/Database mà máy chủ đang chia sẻ. Mỗi phần tử của danh sách (Item) có kiểu BSDatabase.

Cấu trúc: Đối tượng này có các thành phần như sau

Count: cho biết số lượng CSDL/Database

Update: cập nhật danh sách CSDL. Có tham số LoadType

```
Function Update (LoadType As BSLoadType) As Boolean
```

Nếu LoadType là *ItActive* (ngầm định), cập nhật danh sách đang hoạt động tại máy chủ

Nếu LoadType là *ItInactive*, cập nhật danh sách KHÔNG hoạt động tại máy chủ

Nếu LoadType là *ItAll*, cập nhật tất cả danh sách CSDL máy chủ đang quản lý.

### + Database

Loại: Thuộc tính. Kiểu class BSDatabase, chỉ đọc.

Ứng dụng: chứa thông tin về CSDL/Database.

Cấu trúc: Đối tượng này có các thành phần như sau:

UserRanges: kiểu class **BSUserRanges**, chứa danh sách các vùng chia sẻ của CSDL

CA, FullName, Type....các bạn tự tìm hiểu.

### + UserRanges

Loại: Thuộc tính. Kiểu class BSUserRanges, chỉ đọc.

Ứng dụng: chứa danh sách các vùng chia sẻ của CSDL. Mỗi phần tử của danh sách là một BSUserRange.

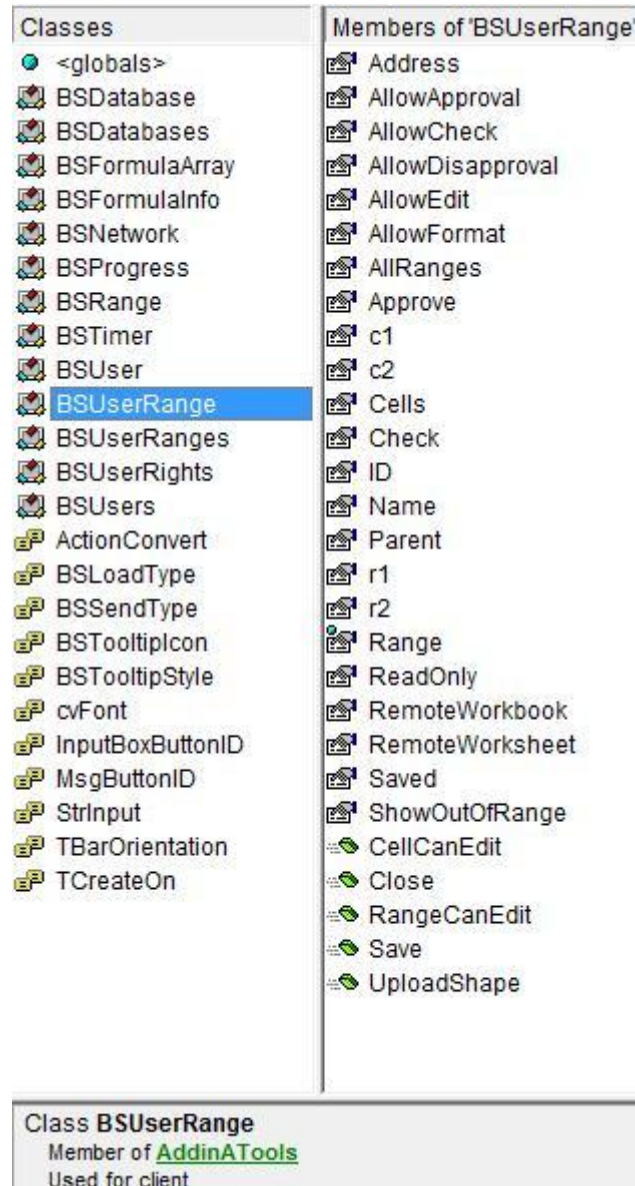
Cấu trúc: Đối tượng này có các thành phần như Count, Updated, Open,...các bạn tự tìm hiểu.

#### + UserRange

Loại: Thuộc tính. Kiểu class BSUserRange, chỉ đọc.

Ứng dụng: chứa và thiết lập các thông tin của vùng chia sẻ.

Cấu trúc:



Các thành phần của class BSUserRange can thiệp vào vùng dữ liệu như ở giao diện người dùng.

Đối tượng Range, Cells có kiểu **BSRange** gần giống với class Range của Excel, cho phép nhận và ghi dữ liệu lên vùng dữ liệu trong máy chủ.

+ Ví dụ: máy khách kết nối và đọc các thông tin CSDL trong máy chủ.

```
Function ConnectToServer() As Boolean
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
```

```

ConnectToServer = XNet.IsRunning And XNet.Connected
If Not ConnectToServer Then
    ConnectToServer = XNet.Connect("localhost", "user", "")
    'ConnectToServer = XNet.Connect("127.0.0.1", "user", "")
    'ConnectToServer = XNet.Connect("www.bluesofts.net", "user", "")
    If Not ConnectToServer Then Exit Function
    MsgBox "Client is connecting to server.", vbInformation
End If
lbEndProc:
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Caption
End If
End Function

Sub Disconnect()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    If XNet.IsRunning And Not XNet.IsServer Then
        XNet.Disconnect
        MsgBox "Client is disconnected.", vbInformation
    End If
lbEndProc:
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Caption
End If
End Sub

Sub GetDatabasesInfo()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim db As BSDatabase
    Dim rng As BSUserRange
    Dim I&
    If Not ConnectToServer Then Exit Sub
    Cells.ClearContents
    Range("A1").Value = "DATABASE INFORMATIONS"
    I = 2
    For Each db In XNet.Databases
        I = I + 1
        Cells(I, 1).Value = db.Name
        For Each rng In db.UserRanges
            I = I + 1
            Cells(I, 2).Value = rng.ID
            Cells(I, 3).Value = rng.Name
        Next
    Next
    Set XNet = Nothing
lbEndProc:
Set UR = Nothing
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Caption
End If
End Sub

```

**+ BSRRange**

Loại: Thuộc tính. Kiểu class BSRRange, chỉ đọc.



Ứng dụng: đọc và ghi dữ liệu và công thức lên vùng dữ liệu. Cách đọc và ghi dữ liệu tương tự như class Range của Excel.

Ví dụ:

BSRange("A1").Value = "Nhập dữ liệu lên máy chủ"

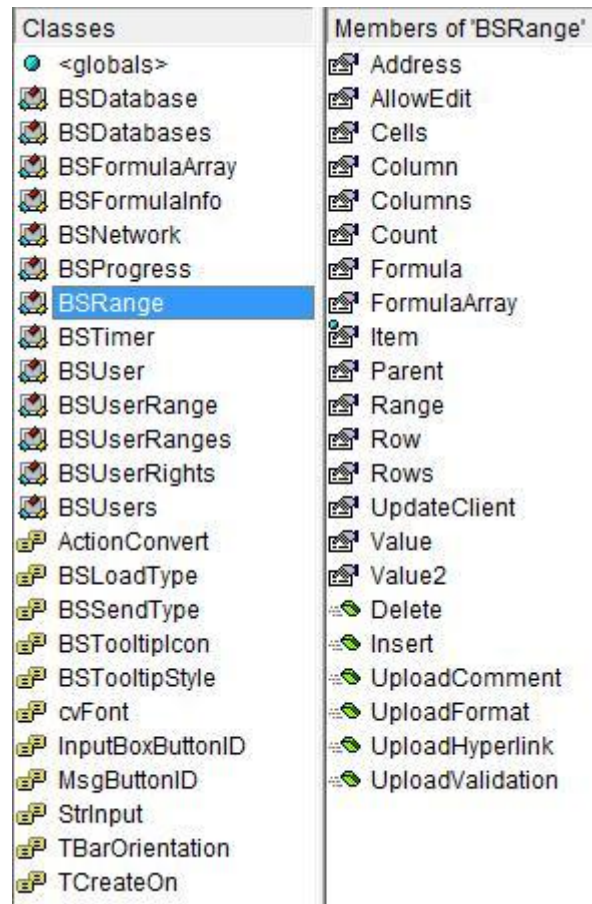
BSRange("A2").Formula = "=SUM(B1:B10)"

BSRange("A3").FormulaArray = "=SUM(IF(B1:B10 > 10, B1:B10, 0))"

Địa chỉ vùng có thể dùng theo chỉ mục dòng và cột.

BSRange(2, 2).Formula = "=SUM(B1:B10)"

Cấu trúc:



+ Ví dụ: Máy khách ghi dữ liệu lên CSDL "Shops.xls", sheet "Shop 1" trong máy chủ.

```
Sub WriteDataToServer()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim UR As BSUserRange
    Dim I&
    If Not ConnectToServer Then Exit Sub
    Set UR = XNet.Databases("Shops.xls").UserRanges("Shop 1")
    UR.Range("A7").Value = "Writing data by VBA!"
    For I = 9 To 30
        UR.Cells(I, 4).Value = I * 1000
        'UR.Range("D" & I).Value = I * 1000
        'Use Cells() is faster Range() !!!
    Next I
lbEndProc:
```

```

Set UR = Nothing
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Caption
End If
End Sub

```

+ Ví dụ: Máy khách lấy dữ liệu lưu trong CSDL "Shops.xls", sheet "Shop 1" ở máy chủ.

```

Sub ReadDataFromServer()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim UR As BSUserRange
    Dim I&
    If Not ConnectToServer Then Exit Sub
    Set UR = XNet.Databases("Shops.xls").UserRanges("Shop 1")
    Cells.ClearContents
    UR.Range("A7").Value = "Reading data in ""Shop 1"" by VBA!"
    For I = 9 To 30
        Cells(I, 1).Value = UR.Cells(I, 4).Value
        UR.Range("D" & I).Value = I * 1000
    Next I
lbEndProc:
    Set UR = Nothing
    Set XNet = Nothing
    If Err <> 0 Then
        MsgBoxW2 Err.Description, vbCritical, Application.Caption
    End If
End Sub

```

### 3. Kết hợp giữa BSNetwork và BSFormulaArray

Khi máy khách đang kết nối tới máy chủ, ta có thể dùng hàm BS\_SQL để truy vấn dữ liệu trong máy chủ.

+ Ví dụ: lấy toàn bộ dữ liệu trong sổ KHO của file Access "Examble.mdb" đã kết nối với DBKEY là "MDB" trên máy chủ.

=BS\_SQL("select \* from KHO", "SERVERSOURCE=YES; DBKEY=MDB")

Trong VBA ta dùng hàm GetRecordSet().

+ ExecSql()

Loại: Hàm. Kiểu giá trị đối tượng Recordset.

Ứng dụng: dùng để truy vấn câu lệnh SQL, lấy dữ liệu từ máy chủ. Nếu kết nối thành công, hàm trả về bảng dữ liệu trong đối tượng Recordset, ngược lại trả về Nothing.

Cấu trúc:

```
Function ExecSql(SQLText As String, Options As String) As Unknown
```

Tham số *SQLText*, *Options* dùng tương tự như hàm BS\_SQL().

+ Ví dụ: lấy toàn bộ dữ liệu trong sổ KHO của file Access "Examble.mdb" đã kết nối với DBKEY là "MDB" trên máy chủ.

```

Sub GetRecordSetFromServer()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork

```

```

Dim Rst As Object
If Not ConnectToServer Then Exit Sub
Cells.ClearContents
Set Rst = XNet.ExecSql("select * from KHO", "DBKEY=MDB")
Range("A1").CopyFromRecordset Rst
lbEndProc:
Set Rst = Nothing
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Caption
End If
End Sub

```

#### 4. Lập trình sự kiện truy xuất dữ liệu giữa các máy trong mạng

Bạn có thể xem ví dụ trong các tệp [Client Chat.xls](#), [Server Chat.xls](#) – Tạo chương trình Chat trong Excel.

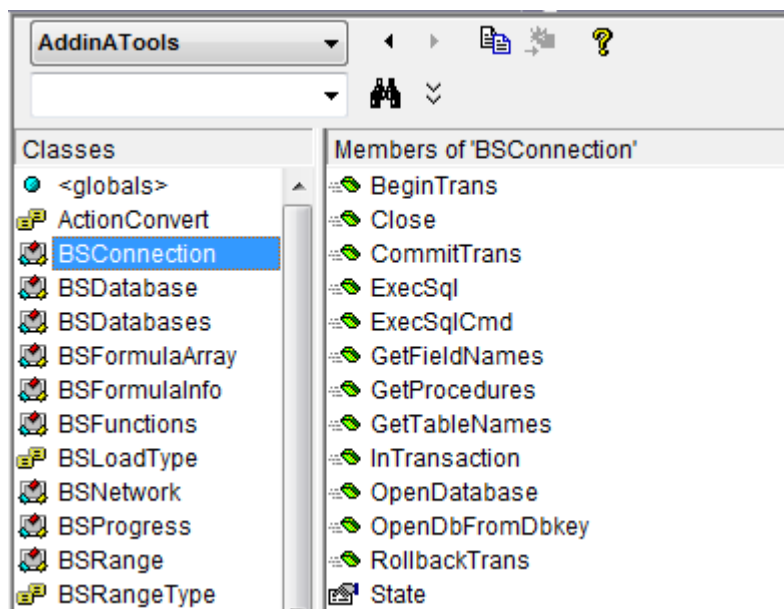
### III. Lập trình truy xuất cơ sở dữ liệu bên ngoài vào Excel trong mạng LAN, Internet

Các cơ sở dữ liệu bên ngoài như Access, Excel, Foxpro, SQL Server, MySQL,... Add-in A-Tools đều cho phép lấy và cập nhật qua mạng LAN, Internet. Các loại CSDL dạng file server như: Access, Excel, Foxpro bình thường không thể kết nối qua internet, chỉ có thể share full thư mục trong mạng LAN mà thôi. Nay với Add-in A-Tools chúng ta có thể giải quyết việc kết nối với chúng qua mạng với giao thức TCP/IP – Client – Server.

Các ví dụ trong phần này nằm trong file: "[C:\A-Tools\HELP & DEMOS\A-Tools VBA Programming\Network\ADO db Access.xls](#)"

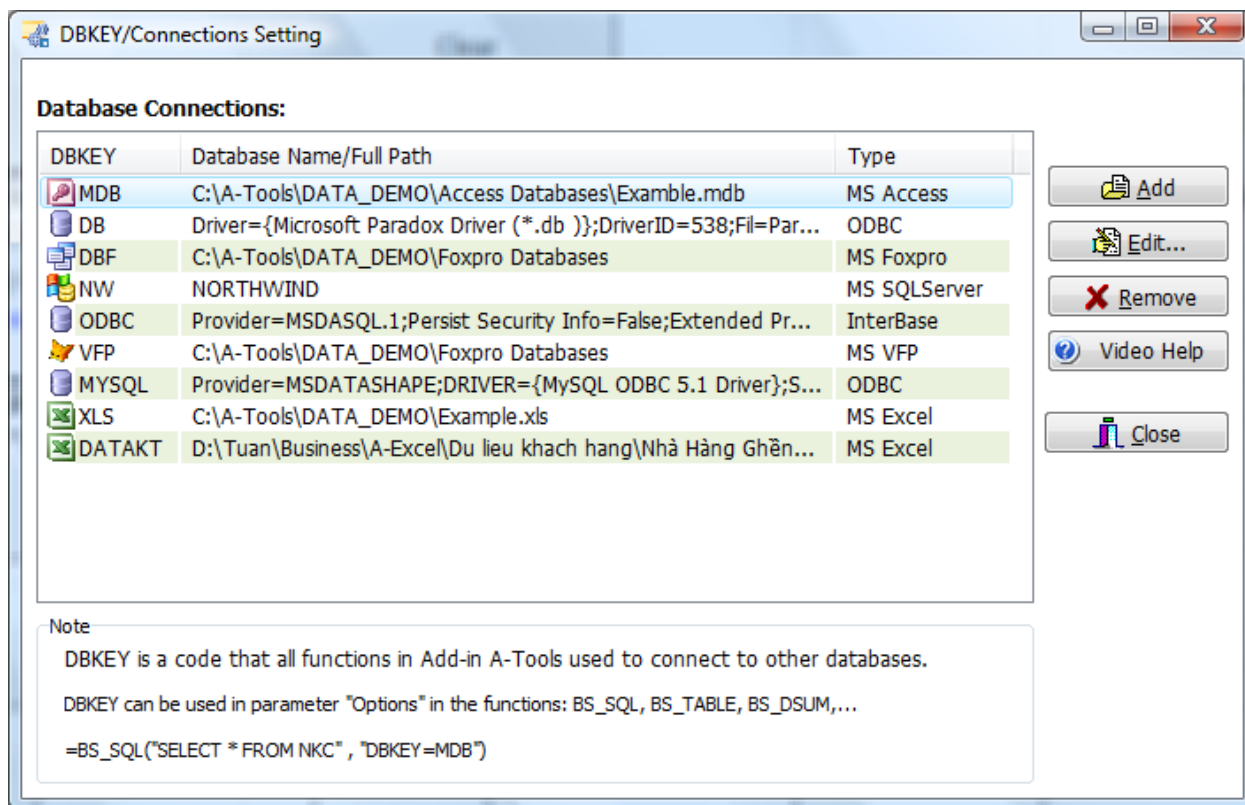
#### + BSCConnection

Để lập trình với CSDL chúng ta sử dụng class BSCConnection. Đối tượng này được trả về bởi hàm **BSNetwork.OpenConnection(...)**.

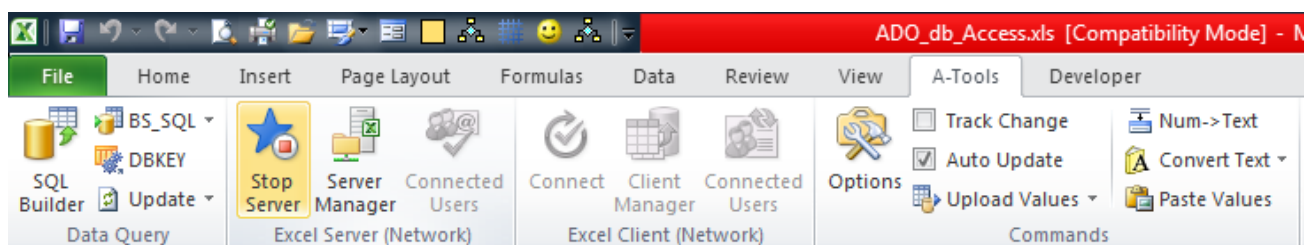


#### 1. Tạo DBKEY tại máy chủ - Thiết lập kết nối CSDL

Để chia sẻ các loại CSDL từ máy chủ cho máy khách kết nối và truy xuất, tại máy chủ A-Tools, tạo DBKEY liên kết tới các loại CSDL trong máy chủ. Trong tài liệu này tôi dùng DBKEY "MDB" kết nối tới "C:\A-Tools\DATA\_DEMO\Access Databases\Examble.mdb".



Máy chủ phải đang chạy



## 2. Kết nối CSDL trong máy chủ từ máy khách

### + OpenConnection

Loại: Hàm. Kiểu giá trị đối tượng BSConnection. Hàm OpenConnection trả về đối tượng BSConnection, từ đối tượng này người lập trình có thể thực hiện các lệnh tới CSDL như SELECT, INSERT, UPDATE, DELETE,...

Ứng dụng: Hàm này dùng để mở kết nối tới CSDL trong máy chủ.

Cấu trúc:

**Function** OpenConnection(DBKEY As String, [Password As String], [Type As TBSDatabaseType = dtUnknown]) As BSConnection

- DBKEY: là mã DBKEY đã tạo trong máy chủ. DBKEY cũng có thể là đường dẫn đến một file CSDL Access, Excel trong máy chủ.
- Password: là mật khẩu dùng để mở CSDL, chỉ phải dùng nếu CSDL mở là loại Access có mật khẩu. Tham số này có thể bỏ qua.
- Type: loại CSDL. Tham số này có thể bỏ qua.

**+ CloseConnection**

- Loại: Thủ tục.
- Ứng dụng: dùng để đóng kết nối CSDL trong máy chủ. Giải phóng bộ nhớ.

**+ BSConnection.ExecSql()**

- Loại: Hàm. Kiểu giá trị trả về đối tượng ADODB.Recordset bởi việc thực thi câu lệnh SELECT
- Ứng dụng: Hàm này dùng để truy vấn và lấy ra dữ liệu từ máy chủ.
- Cấu trúc:

Function **ExecSql**(SQLStatement As String) As Object

- SQLStatement: là chuỗi khai báo câu lệnh SELECT

**3. Kết nối và truy vấn CSDL, thực hiện lệnh SELECT**

**Ví dụ: lấy ra danh sách khách hàng trong bảng DMKH và trả về recordset**

```
Sub GetRecordSetFromServer()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim Rst As Object 'Recordset
    Dim cnn As BSConnection, I As Long
    If Not ConnectToServer Then Exit Sub 'Kiểm tra và kết nối máy chủ
    'Clear values
    DoClearContents
    'Connect to database in server and get instance to BSConnection
    Set cnn = XNet.OpenConnection("MDB") 'MDB is Dbkey code that added in "Dbkey Manager in
server". You can replace "MDB" with "C:\A-Tools\DATA_DEMO\Access Databases\Examble.mdb"
    Set Rst = cnn.ExecSql("select * from dmkh") 'Get recordset by executing SQL statement
    'Get column header
    For I = 0 To Rst.Fields.Count - 1
        Cells(5, 7 + I).Value = Rst.Fields(I).Name
    Next I
    'Paste values from recordset to sheet
    Range("G6").CopyFromRecordset Rst
    'cnn.Close 'Close connection
lbEndProc:
    If Not Rst Is Nothing Then
        If Rst.State = adStateOpen Then
            Rst.Close
        End If
    End If
    Set Rst = Nothing
    Set cnn = Nothing
    Set XNet = Nothing
```

```

If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Name
End If
End Sub

'Hàm kết nối tới máy chủ
Function ConnectToServer() As Boolean
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    ConnectToServer = XNet.IsRunning And XNet.Connected
    If Not ConnectToServer Then
        ConnectToServer = XNet.Connect("bluesofts.dyndns.org", "user")
        'ConnectToServer = XNet.Connect("localhost", "user", "")
        'ConnectToServer = XNet.Connect("127.0.0.1", "user", "")
        If Not ConnectToServer Then Exit Function
        MsgBoxW2 "Client is connecting to server.", vbInformation
    End If
lbEndProc:
    Set XNet = Nothing
    If Err <> 0 Then
        MsgBoxW2 Err.Description, vbCritical, Application.Name
    End If
End Function

'Thủ tục xóa vùng dữ liệu
Sub DoClearContents()
    Range("G5:U20000").ClearContents
End Sub

```

#### 4. Thực hiện lệnh INSERT – Chèn dữ liệu vào CSDL trong máy chủ từ máy khách

##### + BSConnection.ExecSqlCommand()

- Loại: Hàm. Kiểu giá trị trả về số dòng bị thay đổi
- Ứng dụng: Hàm này dùng để thực thi các câu lệnh INSERT, UPDATE, DELETE,... và trả về số dòng được tác động vào.
- Cấu trúc:

```
Function ExecSqlCommand (SQLStatement As String) As Object
```

- SQLStatement: là chuỗi khai báo câu lệnh INSERT, UPDATE, DELETE,...

##### Ví dụ: chèn 1 dòng dữ liệu (bản ghi) vào bảng DMKH.

```

Sub DoExecuteCommandINSERT()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim X As Long, I As Long

```



```

Dim cnn As BSCConnection
If Not ConnectToServer Then Exit Sub
'MA_KH, TEN_KH, MS_THUE, DIA_CHI, DIEN_THOAI, VIP, TINH_TP
Set cnn = XNet.OpenConnection("MDB")
cnn.BeginTrans ' Đặt trạng thái CSDL có thể khôi phục - Undo
X = cnn.ExecSqlCommand("INSERT INTO dmkh (MA_KH,TINH_TP) VALUES ('NEW 1', 'HN' ")
If MsgBoxW2("You have inserted " & X & " records." & Chr(13) & _
    "Do you want to Undo/RollbackTrans) INSERT?", vbQuestion + vbYesNo, _
    Application.Name) = vbYes Then
    cnn.RollbackTrans 'Undo – Khôi phục lệnh (hủy bỏ lệnh đã làm)
Else
    MsgBoxW2 "Records affected: " & X, , "INSERT NEW1"
End If
lbEndProc:
If Not cnn Is Nothing Then
    If cnn.InTransaction Then 'Nếu có trạng thái khôi phục
        cnn.CommitTrans 'Chấp nhận thay đổi vào CSDL
    End If
End If
Set cnn = Nothing
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Name
End If
End Sub

```

## 5. Thực hiện lệnh UPDATE, DELETE – Sửa, xóa dữ liệu

```

Sub DoExecuteCommandUPDATE()
    On Error GoTo lbEndProc
    Dim XNet As New BSCNetwork
    Dim X As Long
    Dim cnn As BSCConnection
    If Not ConnectToServer Then Exit Sub
    Set cnn = XNet.OpenConnection("MDB")
    X = cnn.ExecSqlCommand("UPDATE dmkh SET TEN_KH= 'NEW VALUE' WHERE TINH_TP='HN' ")
    MsgBoxW2 "Records affected: " & X, , Application.Name
lbEndProc:
    Set cnn = Nothing
    Set XNet = Nothing

```

```

If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Name
End If
End Sub

Sub DoExecuteCommandDELETE()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim X As Long
    Dim cnn As BSNConnection
    If Not ConnectToServer Then Exit Sub
    Set cnn = XNet.OpenConnection("MDB")
    X = cnn.ExecSqlCmd("DELETE FROM dmkh WHERE MA_KH LIKE 'NEW%'")
    MsgBoxW2 "Records affected: " & X, , Application.Name
lbEndProc:
    Set cnn = Nothing
    Set XNet = Nothing
    If Err <> 0 Then
        MsgBoxW2 Err.Description, vbCritical, Application.Name
    End If
End Sub

```

## 6. Lấy thông tin về CSDL đang mở

**Function GetTableNames**(ListTableNames, [IncludeSysTable As Boolean = False]) As Long

- Hàm trả về số bảng có trong CSDL
- ListTableNames: là mảng được trả về danh sách các tên bảng dữ liệu
- IncludeSysTable: có lấy các bảng hệ thống không

**Function GetFieldNames**(TableName As String, ListFieldNames) As Long

- Hàm trả về số cột có trong bảng xác định
- ListFieldNames: là mảng được trả về danh sách các tên cột trong bảng

**Function GetProcedures**(ListProcNames) As Long

- Hàm trả về số cột có trong bảng xác định
- ListProcNames: là mảng được trả về danh sách các tên thủ tục có trong CSDL

a. Lấy danh sách các bảng/table dữ liệu trong CSDL đã mở

```

Sub GetTableNames()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim X As Long, list, I&
    Dim cnn As BSNConnection
    If Not ConnectToServer Then Exit Sub
    Set cnn = XNet.OpenConnection("MDB")
    X = cnn.GetTableNames(list)

```

```

Range("G6:G1000").ClearContents
'fill list
Range("G6").Resize(X).Value = WorksheetFunction.Transpose(list)
'fill list
'For I = 0 To X - 1
' Debug.Print list(I)
'Next I
MsgBoxW2 "Count list: " & X, , "GetTableNames"
XNet.CloseConnection 'Close and free connection in server
lbEndProc:
Set cnn = Nothing
Set XNet = Nothing
If Err <> 0 Then
    MsgBoxW2 Err.Description, vbCritical, Application.Name
End If
End Sub

```

b. Lấy danh sách các cột/field trong bảng dữ liệu

```

Sub GetFieldNames()
    On Error GoTo lbEndProc
    Dim XNet As New BSNetwork
    Dim X As Long, list, I&
    Dim cnn As BSConnection
    If Not ConnectToServer Then Exit Sub
    Set cnn = XNet.OpenConnection("MDB")
    X = cnn.GetFieldNames("DMKH", list)
    Range("G6:G1000").ClearContents
    'fill list
    Range("G6").Resize(X).Value = WorksheetFunction.Transpose(list)
    'fill list
    'For I = 0 To X - 1
    ' Debug.Print list(I)
    'Next I
    MsgBoxW2 "Count list: " & X, , "GetFieldNames"
    XNet.CloseConnection 'Close and free connection in server
lbEndProc:
Set cnn = Nothing
Set XNet = Nothing
If Err <> 0 Then

```

```
MsgBoxW2 Err.Description, vbCritical, Application.Name
End If
End Sub

c. Lấy danh sách các thủ tục trong CSDL đã mở
Sub GetProcedures()
On Error GoTo lbEndProc
Dim XNet As New BSNetwork
Dim X As Long, list, I&
Dim cnn As BSConnection
If Not ConnectToServer Then Exit Sub
Set cnn = XNet.OpenConnection("MDB")
X = cnn.GetProcedures(list)
Range("G6:G1000").ClearContents
'fill list
If X > 0 Then
Range("G6").Resize(X).Value = WorksheetFunction.Transpose(list)
End If
'fill list
'For I = 0 To X - 1
' Debug.Print list(I)
'Next I
MsgBoxW2 "Count list: " & X, , "GetProcedures"
XNet.CloseConnection 'Close and free connection in server
lbEndProc:
Set cnn = Nothing
Set XNet = Nothing
If Err <> 0 Then
MsgBoxW2 Err.Description, vbCritical, Application.Name
End If
End Sub
```

---

Để tìm hiểu về lập trình mạng nâng cao trong A-Tools nhà lập trình nên đọc và chạy thử các tệp:

- "[BSNetworkInfo.xls](#)" , "[BSNetwork.xls](#)" (đường dẫn C:\A-Tools\HELP & DEMOS\A-Tools VBA Programming).
- "[Chat](#)" – Tạo chương trình Chat trong Excel.
- "[ExcelCaro.xls](#)" – Tạo game chạy mạng trong Excel.
- "[ADO db Access.xls](#)" – Lập trình với CSDL trong Excel.

Tài liệu hướng dẫn chi tiết về lập trình tác giả sẽ bổ sung trong các phiên bản mới của A-Tools hoặc cập nhật trên website: [www.bluesofts.net](http://www.bluesofts.net)

---

Nếu áp dụng triệt để các tính năng của Add-in A-Tools chắc chắn bạn sẽ có một ứng dụng mạnh mẽ chạy trong Excel VBA trong môi trường chia sẻ dữ liệu trong mạng LAN, Internet.

Tác giả: ThS. Nguyễn Duy Tuân