



集算器

Excel集成集算器

润乾软件出品



Excel通过插件集成集算器，可在Excel中方便使用集算器的众多功能。本文详实介绍了集成的软件安装环境设置、操作说明、集算器dfx脚本的使用以及vba脚本调用esproc接口等，让用户快捷地了解熟悉并掌握在Excel中使用集算器。



- 1、序言
- 2、结构关系
- 3、环境设置
 - A. Java安装
 - B. 集算器设置
 - C. 加载插件
- 4、操作说明
- 5、esproc使用：
 - A. 单元格值操作
 - B. 表格填充操作
 - C. 复制操作
 - D. 向esproc传递参数
- 6、vba调用esproc接口样例
- 7、最后总结

1、序言

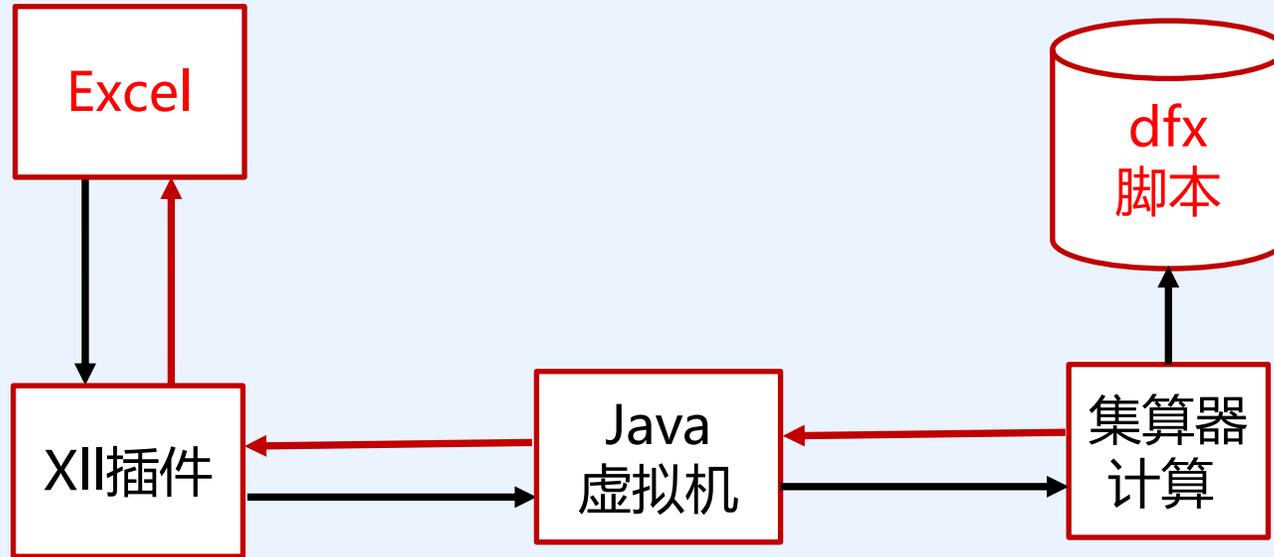


Excel是行业领先的电子表格程序,在各行各业得到广泛的应用,是最流行的个人计算机数据处理软件。

集算器面向结构化处理的数据计算引擎,具备敏捷语法、网格风格脚本和完备的调试功能,对结构化/半结构化数据进行了针对性设计,适合进行步骤多、业务规则复杂的商业计算和数据源多样的混合计算,能为业务逻辑复杂的海量数据提供运算服务。

使用Excel作为面向用户的前端,利用集算器SPL语言提供的接口、计算、业务逻辑和数据访问功能,实现两者的最佳结合。

2、结构关系



Excel通过扩展其功能的Xll插件，将数据传递给Java虚拟机，虚拟机通知集算器计算引擎，执行dfx脚本后将计算结果再按原路返回，在Excel中呈现出来。

对于用户来说，在了解Excel与集算器结构关系的基础上，重点是关注前端Excel如何调用dfx脚本，解决用户需求的。

3、环境设置



A. Java安装

I、要求同架构：Excel与java jdk同为x86或x64架构，不能用异构组合，否则出现异常。运行java -version可知道是否为x64架构。

II、SPL自带JDK：若系统注册表regedit中java jdk的jvm.dll没有安装或无效，则Excel将加载集算器中的jdk(位于raqsoft\common\jre\bin\server下)，即在注册表中的jdk优先于集算器自带的jdk。

III、用户安装JDK：用户若想用指定版本的java，对于java jdk是安装版本或绿色版本，都要确保jdk在注册表中的信息是有效的。

A. Java安装



下面以win10 jdk1.7.0.71 x86为例说明。

jdk在注册表中的位置: HKEY_LOCAL_MACHINE\SOFTWARE下

X86: SOFTWARE\JavaSoft\Java Runtime Environment

X64: SOFTWARE\WOW6432Node\JavaSoft\Java Runtime Environment

在版本1.7.0.71下RuntimeLib值是jvm.dll所在的文件位置

计算机\HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment

名称	类型	数据
(默认)	REG_SZ	(数值未设置)
BrowserJavaVersion	REG_SZ	10.79.2
CurrentVersion	REG_SZ	1.7
Java7FamilyVersion	REG_SZ	1.7.0_71
RuntimeLib	REG_SZ	D:\java\jdk1.7.0_71\jre\bin\server\jvm.dll

对于电脑上安装多个版本的java时, 需要注意jdk版本之间冲突问题, 可能造成Excel加载插件失败。

B. 集算器设置



I、Excel相关文件：

集算器安装后，与Excel相关的两个文件分别为：

esProc\lib\EsprocXll.jar,

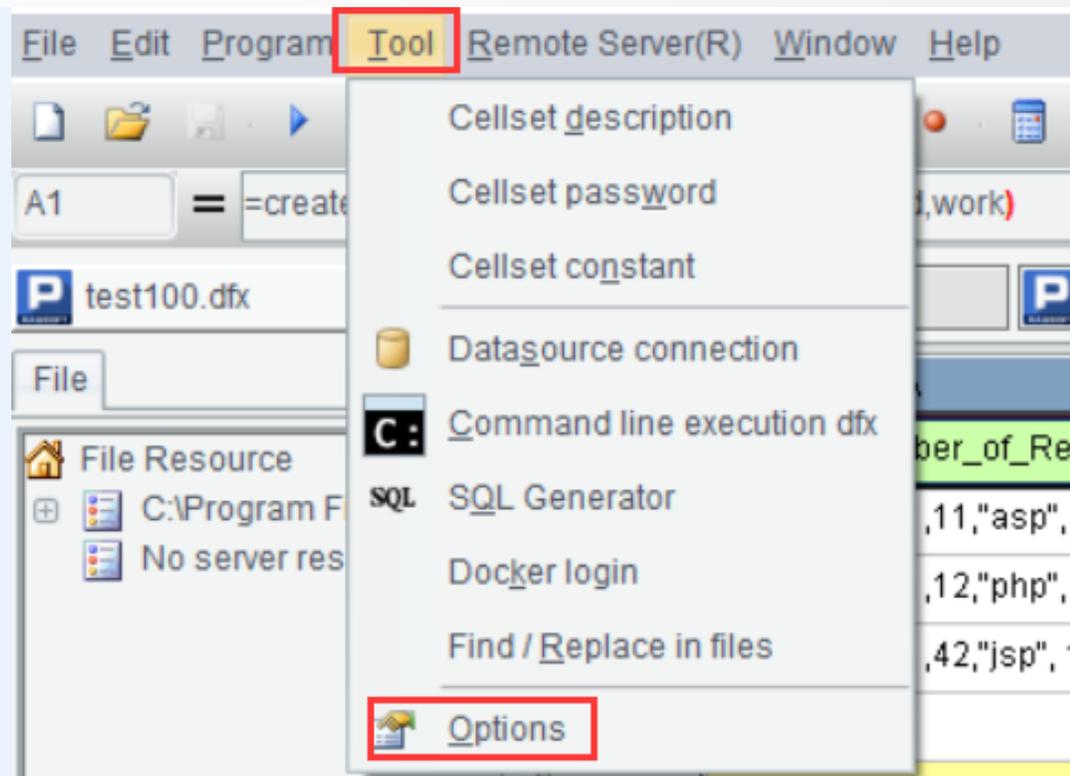
esProc\bin\ExcelRaq.xll。

其中ExcelRaq.xll是集算器的Excel插件，不能更换它们的位置，否则加载插件失败。

II、dfx文件位置设置：

设置dfx脚本文件存放位置，方便SPL调用。集算器界面中设置dfx文件路径操作如下：

Menu—>Tool—>Options—>Environment—>
Main path



B. 集算器设置



Option

General **Environment** Appearance

Log file name Browse

Searching path Browse

License file name Set

Main path Browse

Note: Relative path does not start with / or \

Temp path Edit

Initialization program Browse

External library directory Browse

Date format Time format

Date time format Default charset name

Local host Local port

File buffer(Byte) File loading lock time (Second)

Missing format Group table block size(bytes)

Cursor fetch count

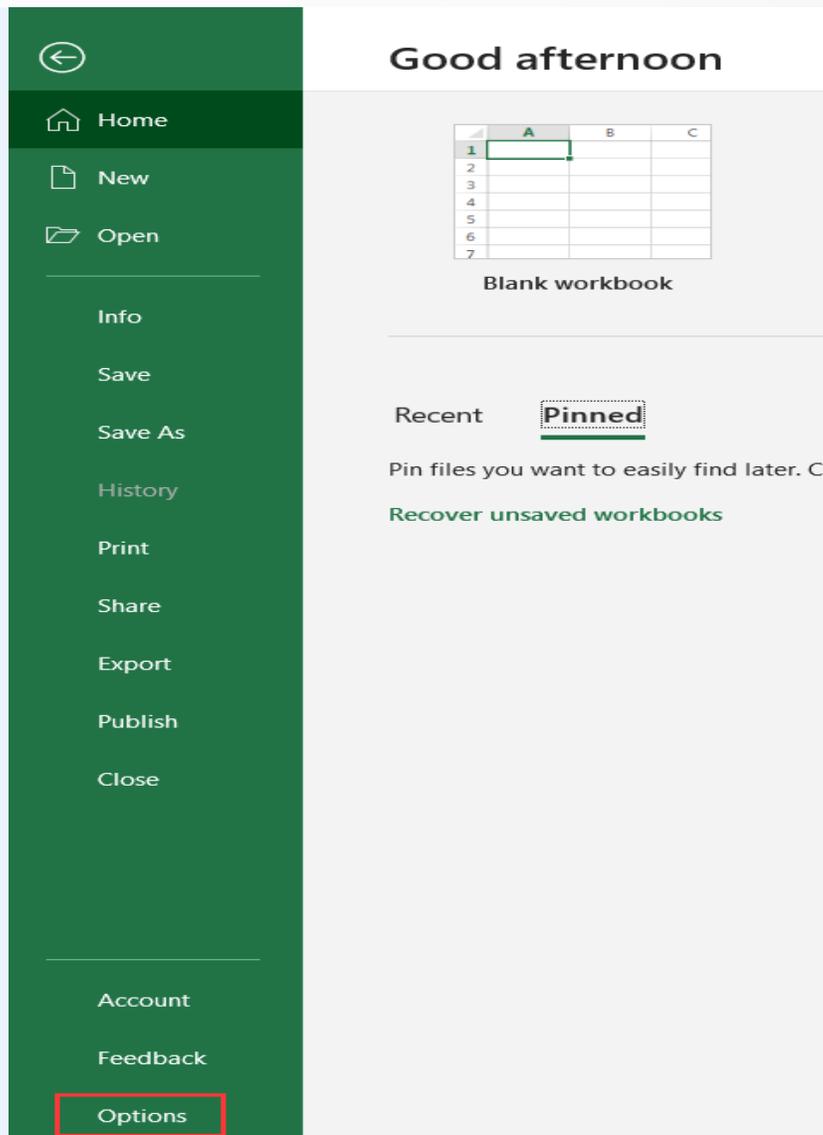
OK Cancel

缺省情况下dfx文件位置
设置在集算器安装目录
raqsoft\esProc\demo
下。

C. 加载插件



打开excel软件后，可以通过菜单中的文件—>选项-->加载项-->转到-->浏览-->选择ExcelRaq.xll文件，加载项中勾选EsprocXII，此时就启用了ExcelRaq文件。



C. 加载插件



Excel Options

- General
- Formulas
- Proofing
- Save
- Language
- Ease of Access
- Advanced
- Customize Ribbon
- Quick Access Toolbar
- Add-ins**
- Trust Center

View and manage Microsoft Office Add-ins.

Add-ins

Name ^	Location
Active Application Add-ins	
EsprocXII	C:\...am Files\raqsoft\esProc\bin\ExcelRaq.xll
Example Standalone DLL	D:\...LES\EXAMPLE\x64\Debug\EXAMPLE.xll
Team Foundation Add-in	C:\...tion Server\14.0\x64\TFOfficeAdd-in.dll
中文转换加载项	C:\...\root\Office16\ADDINS\TCSCCONV.DLL
Inactive Application Add-ins	
Analysis ToolPak	C:\...Office16\Library\Analysis\ANALYS32.XLL
Analysis ToolPak - VBA	C:\...ice16\Library\Analysis\ATPVBAEN.XLAM
Date (XML)	C:\...s\Microsoft Shared\Smart Tag\MOFL.DLL
Euro Currency Tools	C:\...root\Office16\Library\EUROTOOL.XLAM
Excelraq	D:\...ll\winrun4j\prj2\x64\Debug\ExcelRaq.xll
Inquire	C:\...soft Office\Office16\DCF\NativeShim.dll
Interp	D:\tmp\logs\Interp.xll
Load Test Report Addin	C:\...ools.LoadTestExcelAddIn.vsto vstolocal
Load Test Report Addin	C:\...ools.LoadTestExcelAddIn.vsto vstolocal
Microsoft Actions Pane 3	

Add-in: EsprocXII
Publisher:
Compatibility: No compatibility information available
Location: C:\Program Files\raqsoft\esProc\bin\ExcelRaq.xll

Description:

Manage: **Excel Add-ins** Go...

Add-ins

Add-ins available:

- Analysis ToolPak
- Analysis ToolPak - VBA
- Esprocraq**
- Euro Currency Tools
- Example Standalone DLL
- Excelraq
- Interp
- PyXLL
- Solver Add-in

OK

Cancel

Browse...

Automation...

Example Standalone DLL

4、操作说明



esproc接口调用说明:

esproc (fmt, arg1,...) 函数接口。

参数fmt: 若为无后缀的dfx文件名, 表示调用dfx脚本文件; 若为以“=”开头的表达式字符串, 表示动态解析并计算的表达式。当为表达式时, 包括的?个数与arg个数相同并且一一对应。

参数arg1,...: 输入参数可以为空、一个或多个, 但最多50个参数, 参数之间用逗号隔开。参数类型可以为string, int, float, double, 一维数组, 二维数组等。

返回类型: 数组结构。

单元格值操作: 选择某个单元格后, 在输入栏输入类似=esproc("tsum", A1:C3), 然后点击enter键, 则只在此单元格显示返回值。

表格填充操作: 选择需要填充的网格, 在输入栏输入类似= esproc("demo",A1:C3), 然后按ctrl+shift+enter组合键, 则实现自动填充。

复制操作: 当dfx脚本使用clipboard(s)函数, Excel调用esproc执行之后, 可选择另外一个单元格, 将剪贴板中的计算结果复制过去。

5、esproc使用



下面通过不同操作方式的esproc来说明，Excel如何调用dfx脚本的。

A. 单元格值操作

我们先了解排名次的两种方法：

I、RANK式排名：

比如有五个数字100,100,80,60,50 排名次后的结果是

第1名：100,100

第2名：空

第3名：80

第4名：60

第5名：50

从上面排名可以看出，如果有并列排名的情况，有几个相同的，下面的名次就会空出几个位置。

II、中国式排名

同样相同的五个数字排名次后的结果是

第1名：100,100

第2名：80

第3名：60

第4名：50

从上面结果可以看出，中国式排名不会因为相同的数字排名就会让排名断开，而是连续的。

A. 单元格值操作



对于RANK式排名, Excel提供对应的接口rank()来实现, 对于中国式排名用vba脚本实现, 则比较繁琐, 代码如下:

```
Function cnrank(ByVal nm, ByVal rng As Range)
    Set d = CreateObject("scripting.dictionary")
    For Each rn In rng
        If VBA.IsNumeric(rn.Value) And Len(rn) > 0 Then
            d(rn.Value) = ""
        End If
    Next rn
    arr = d.keys
    d.RemoveAll
    For j = 0 To UBound(arr)
        d(WorksheetFunction.Large(arr, j + 1)) = j + 1
    Next j
    If VBA.IsNumeric(nm) Then
        pm = d(nm * 1)
    Else
        pm = ""
    End If
End Function
```

A. 单元格值操作



相对于vba代码的实现，SPL代码则精简得多，实现如下：

	A	B	C	D	E
1	Name	score	order		
2	Tom1	99	1		
3	Tom2	99	1		
4	Tom3	25	19	=dfx("=[?].rank@z(?)", B2:B21, 28)	
5	Tom4	26	18		16
6	Tom5	12.3	20		
7	Tom6	28	16		
8	Tom7	29	15		
9	Tom8	30	14	=dfx("=[?].rank@zi(?)", B2:B21, 28)	
10	Tom9	31	13		14
11	Tom10	32	12		
12	Tom11	33	11		
13	Tom12	28	16		
14	Tom13	35	10		
15	Tom14	36	9		
16	Tom15	37	8		
17	Tom16	38	7		
18	Tom17	39	6		
19	Tom18	40	5		
20	Tom19	41	4		
21	Tom20	42	3		

Rank式排名

中国式排名

B. 表格填充操作



下面对服装表进行分类汇总计算，在Excel中将计算结果显示出来。加载clothing.xls数据，按type, category分组求每个分店的销售量。

clothing.xls数据:

type	category	store1	store2	store3	store4	store5
man	A	100	200	300	400	350
man	B	300	400	500	600	522
man	A	800	900	1000	1100	450
man	B	200	300	400	500	300
man	C	400	500	600	700	891
woman	B	600	700	800	900	257
woman	C	500	600	700	800	880
woman	A	700	800	900	1000	750
woman	C	900	1000	1100	1200	440
woman	E	280	0	0	360	600

B. 表格填充操作



vba代码实现如下:

```
Sub clothing()  
    Set d = CreateObject("scripting.dictionary")  
    Application.ScreenUpdating = False  
  
    arr = Sheets(1).[a1].CurrentRegion  
    c = UBound(arr, 2)  
    For j = 3 To UBound(arr)  
        For i = 3 To UBound(arr, 2)  
            d(arr(j, 1) & "##" & arr(j, 2) & "##" & arr(2,  
i)) = d(arr(j, 1) & "##" & arr(j, 2) & "##" & arr(2,  
i)) + arr(j, i)  
        Next i  
    Next j  
    Sheets(2).UsedRange.ClearContents  
    Sheets(1).Rows(2).Copy Sheets(2).[a1]  
    r = 2  
    With Sheets(2)
```

```
        For j = 0 To d.Count - 1  
            arr = Split(d.keys()(j), "##")  
            If d.exists(arr(0) & arr(1)) Then  
                a = d(arr(0) & arr(1))  
            Else  
                a = r  
                .Cells(a, 1) = arr(0)  
                .Cells(a, 2) = arr(1)  
                d(arr(0) & arr(1)) = a  
                r = r + 1  
            End If  
            For i = 3 To c  
                .Cells(a, i) = d(arr(0) & "##" & arr(1) & "##"  
& .Cells(1, i))  
            Next i  
        Next j  
    End With  
    Application.ScreenUpdating = True  
End Sub
```

B. 表格填充操作



由于vba缺少分组接口，代码细节需要自己来实现，但SPL对此类问题考虑就周全得多，实现也比较容易，使用`clothing.dfx`脚本代码如下：

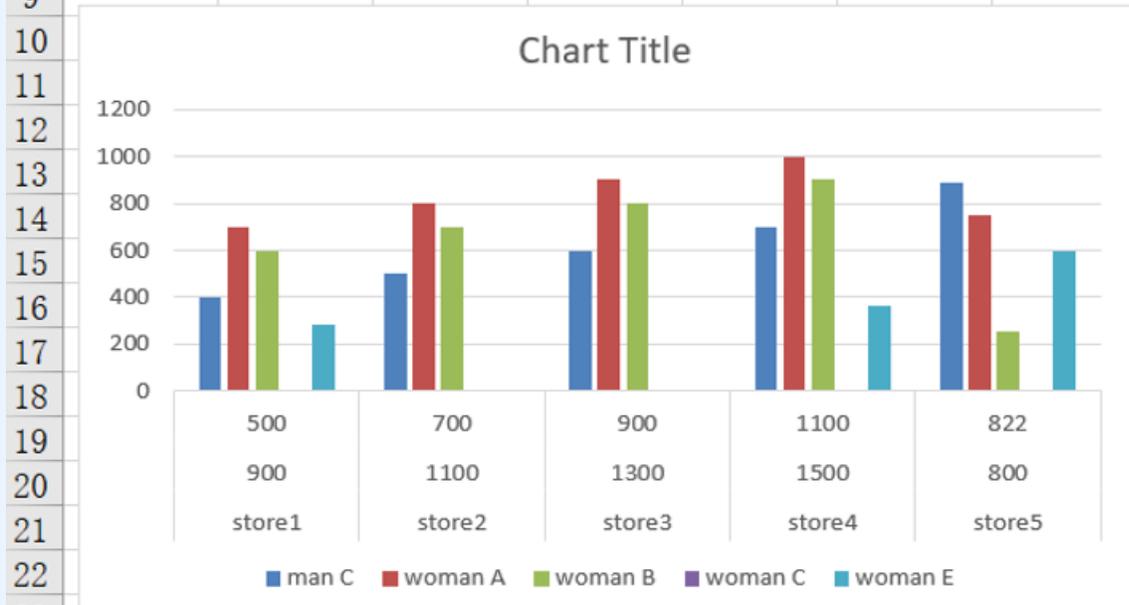
	A
1	<code>=file("D:/dev/clothing.xls").xlsopen()</code>
2	<code>=A1.xlsimport@t(;A1(1).stname, 1)</code>
3	<code>=A2.group(type, category)</code>
4	<code>=A3.new(type, category, ~.sum(#3):store1, ~.sum(#4):store2,~.sum(#5):store3,~.sum(#6):store4,~.sum(#7):store5)</code>
5	<code>>A1.xlsclose()</code>
6	<code>return A4</code>

B. 表格填充操作



	A	B	C	D	E	F	G
1	type	category	store1	store2	store3	store4	store5
2	man	A	900	1100	1300	1500	800
3	man	B	500	700	900	1100	822
4	man	C	400	500	600	700	891
5	woman	A	700	800	900	1000	750
6	woman	B	600	700	800	900	257
7	woman	C	1400	1600	1800	2000	1320
8	woman	E	280	0	0	360	600

操作说明: Excel中, 先选择A1:G8区域, 然后在输入框上输入: `=esproc("clothing")`后, 同时按ctrl+shift+enter键, 则实现自动填充, 结果如左图所示:



也可以插入**柱状图**, 使计算结果显示更直观。

C. 复制操作



还是用上面的clothing测试数据，只是dfx脚本加上了复制功能。
使用`clipboard.dfx`脚本代码如下：

	A
1	<code>=file("D:/dev/clothing.xls").xlsopen()</code>
2	<code>=A1.xlsimport@t(;A1(1).stname, 2)</code>
3	<code>=A2.group(type, category)</code>
4	<code>=A3.new(type, category, ~.sum(#3):store1, ~.sum(#4):store2,~.sum(#5):store3, ~.sum(#6):store4,~.sum(#7):store5)</code>
5	<code>>A1.xlsclose()</code>
6	<code>=clipboard(export@t(A4))</code>

将计算结果复制
到剪贴板上

操作说明： Excel中，先选择A1单元格，然后在输入框上输入：`=esproc("clipboard")`，点击enter键后，使用`ctrl+v`组合键从A2开始粘贴，结果如下：

A1	A	B	C	D	E	F	G
	TRUE						
2	type	category	store1	store2	store3	store4	store5
3	man	A	900	1100	1300	1500	800
4	man	B	500	700	900	1100	822
5	man	C	400	500	600	700	891
6	woman	A	700	800	900	1000	750
7	woman	B	600	700	800	900	257
8	woman	C	1400	1600	1800	2000	1320
9	woman	E	280	0	0	360	600

D. 向esproc传递参数



向dfx脚本传递不同类型的参数。使用tsum.dfx脚本代码如下：

	A	B	C	D	E
1	0	0	0	0	
2	if (ifa(arg1))	>A1=arg1.sum()	else	if(ifnumber(arg1))	>A1=arg1
3				else	>A1=len(arg1)
4	if (ifa(arg2))	>B1=arg2.sum()	else	if(ifnumber(arg2))	>B1=arg2
5				else	>B1=len(arg2)
6	if (ifa(arg3))	>C1=arg3.sum()	else	if(ifnumber(arg3))	>C1=arg3
7				else	>C1=len(arg3)
8	if (ifa(arg4))	for arg4.count()			
9			if (ifa(arg4(B8)))	>D1=D1+arg4(B8).sum()	
10	return A1+B1+C1+D1				

D. 向esproc传递参数



传递的参数分别为整数，单行多值，单列多值及多行多列值。

	A	B	C	D	E	F	G	H
1								
2	1	15	29	10	45			
3	2	16	30	20	=dfx("tsum",88,A5:D5,B2:B6,A1:C3)			
4	3	17	31	30				
5	4	18	32	40				
6	5	19	33	50				
7	6	20	34	60				
8	7	21	35	70				

对于日期类型，需要转换成字符串，代码如下：

```
=esproc("testDate",23.33,13,"2011-02-03")
```

```
=esproc("testDate",A1,C1,TEXT(B1,"yyyy-mm-dd")), 其中B1为日期类型数据。
```

6、vba调用esproc接口样例



在vba脚本中调用SPL接口，实现它们的交互使用。tdemo.dfx脚本代码如下：

	A
1	=create(age, name,pid,work)
2	>A1.insert(0,11,"asp", 100, "techer")
3	>A1.insert(0,12,"php", 200, "manager")
4	>A1.insert(0,42,"jsp", 300, "java")
5	>A1.insert(0,43,"java", 400, "help")
6	=clipboard(export@t(A1))
7	return A1

主要使用Application的函数Run()及ExecuteExcel4Macro()。

A. Application.Run()调用

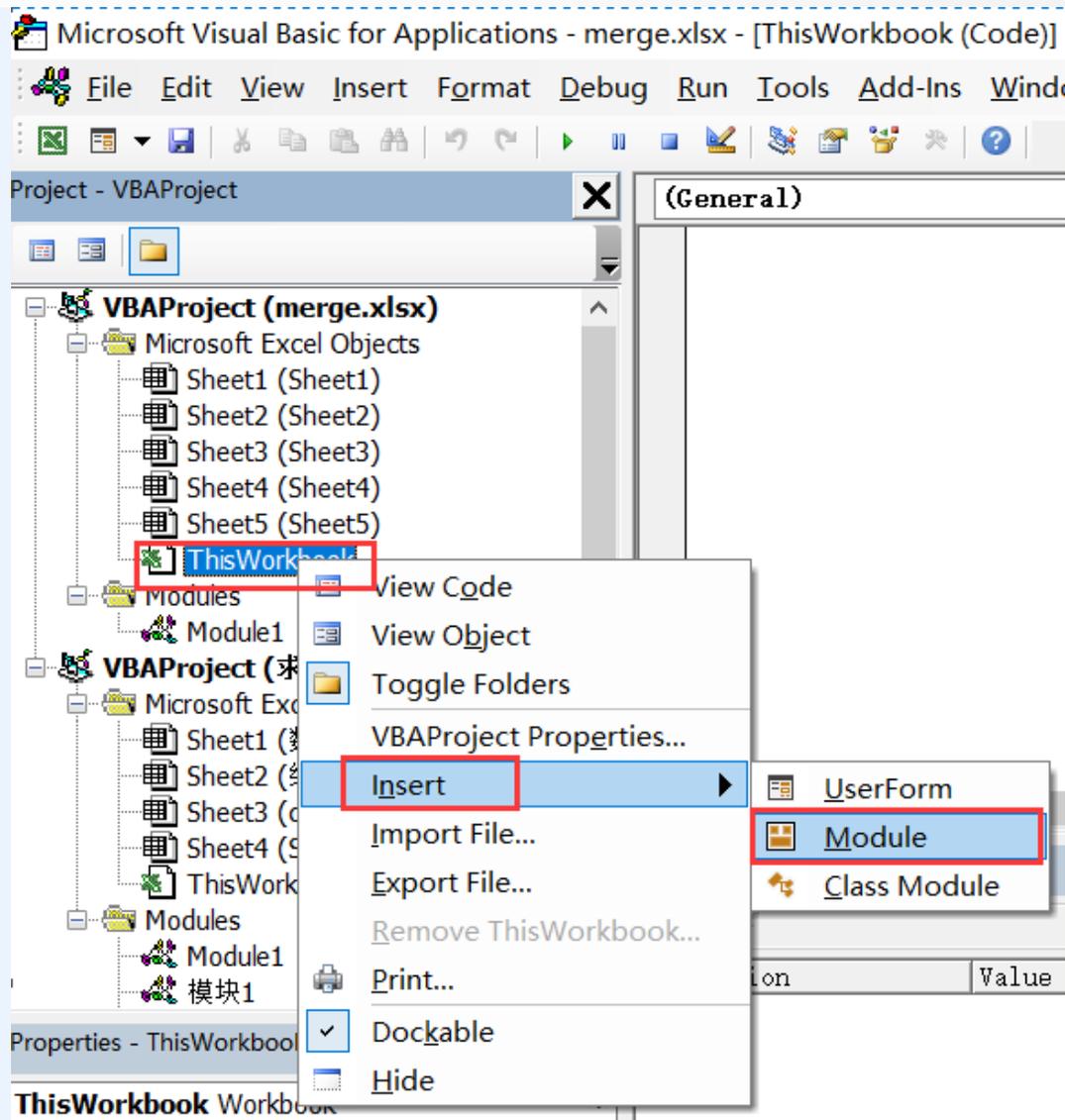


操作步骤:

A: Alt+F11打开vba编译器, 在ThisWorkbook上右击选择insert Module.

B. 输入下面Sub test()函数代码, 它实现将结果显示在Excel中.

C: 执行函数test(),vba脚本自动把结果填充到A1:D5.



A. Application.Run()调用



vba脚本通过Application.Run()调用 dfx脚本，将结果输出到网格中：

	A	B	C	D	E	F	G	H	I	J
1	age	name	pid	work						
2	11	asp	100	techer						
3	12	php	200	manager						
4	42	jsp	300	java						
5	43	java	400	help						

显示结果

Microsoft Visual Basic for Applications - merge.xlsx - [Module1 (Code)]

File Edit View Insert Format Debug Run Tools Add-Ins Window Help

Ln 4, Col 37

Project - VBAProject

(General)

```
Sub Test()  
Application.ScreenUpdating = False  
With Sheets(5)  
arr = Application.Run("dfx", "tdemo")  
. [a1].Resize(UBound(arr, 1), UBound(arr, 2)) = arr  
End With  
Application.ScreenUpdating = True  
End Sub
```

调用dfx脚本

B. Application.ExecuteExcel4Macro()调用



上面调用的dfx脚本返回值为列表，若dfx返回值是单值，则可使用ExecuteExcel4Macro 接口，vba脚本实现代码如下：

```
Sub Test()  
  With Sheets(5)  
    Cells(6, 1) = Application.ExecuteExcel4Macro("esproc(""tdemo"")")  
  End With  
End Sub
```

注意：dfx接口传递参数字符串中含有引号需要用双引号代替

7、最后总结



Excel集成集算器后，用户可根据自己的业务需求，通过SPL语言来实现不同功能的dfx脚本，Excel调用这些dfx脚本，SPL返回计算后的数据，在Excel中或再加工处理，或报表化处理后呈现出来，可交互式实现不同的应用需求。

面对多样复杂业务逻辑、不同的数据源处理、大数据计算等，集算器有自己的独特之处，在这些方面恰好也能弥补Excel的不足。相对于VBA开发，SPL语言适应性更强，业务处理更专业，也比较容易上手。同时，借助于Excel操作的易用性，报表化、可视化功能，两者结合实现优势互补，充分发挥各自的特长，让数据呈现更专业直观、丰富多彩。