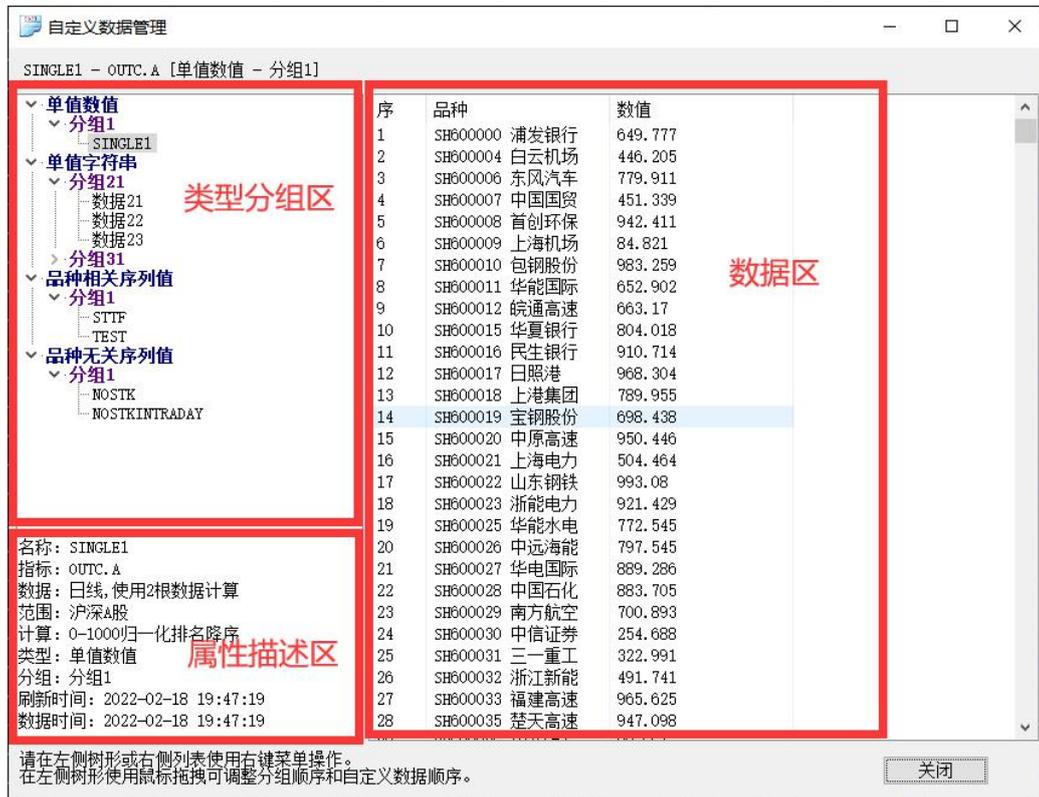


自定义数据

一、简介

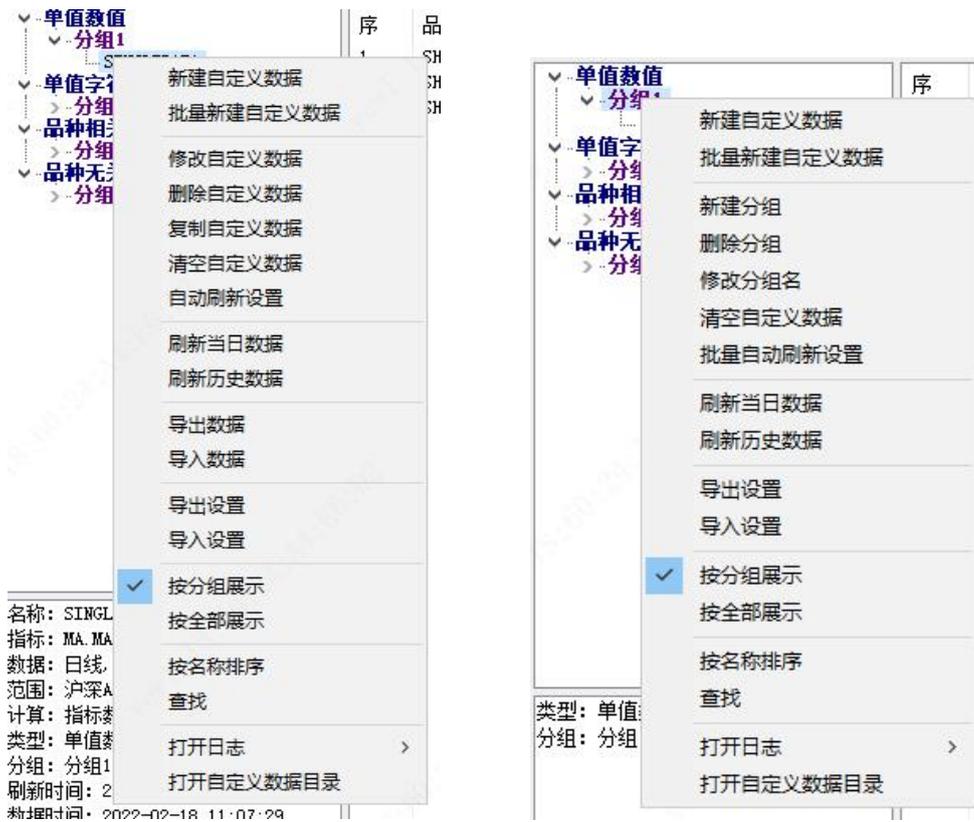
自定义数据是易得程序化非常重要的一个功能，它不仅仅支持对横向统计数据计算和引用，更重要的是作为易得程序化与外部进行数据交换的窗口功能。您可以自定义频率实时进行自定义数据的自动刷新计算并实时导出计算结果；也可以自定义频率实时从外部文件自动导入数据，将外部的任何数据引入易得为量化策略所用。

除了实时刷新导出和实时导入，易得自定义数据还支持分组、批量创建、批量设置自定义刷新等特性，功能非常强大易用。



通过上图的自定义数据主界面，我们看到左上角是类型分组区，其根据自定义数据的类型和分组，采用树状结构展示；左下角是属性描述区，用来描述树状结构当前选中节点的属性信息；右侧的是数据展示区，展示了自定义数据的数值，您可以在这里进行添加、删除、修改等编辑操作。

无论类型分组区还是数据区，您都可以通过右键菜单进行快捷操作。见下图：



序	品种	数值
1	SH600000 浦发银行	2.69
2	SH600001	添加数据
3	SH600001	修改数据
4	SH600001	复制数据
5	SH600001	删除选中数据
6	SH600001	清空所有数据
7	SH600001	刷新当日数据
8	SH600001	刷新历史数据
9	SH600001	打开日志
10	SH600001	打开自定义数据目录
11	SH600001	
12	SH600001	
13	SH600001	
14	SH600001	
15	SH600020 中原高速	3.26
16	SH600001 上海中	12.416

二、自定义数据类型

自定义数据有四种类型，分别为：

- 单值数值** - 每只品种仅有一个固定的数值
- 单值字符串** - 每只品种仅有一个固定的字符串

品种相关序列值 - 每只品种有一组时间序列数值

品种无关序列值 - 与品种无关的一组时间序列数值

四种类型的数据都可以是跟指标计算相关联的计算结果，也可以是由用户自己编辑或者从外部导入的数据。

1、单值数值

单值数值类型，每只品种仅有一个固定的数值。如果跟指标计算关联，会取指标计算最新的一个数值，可以是指标的计算结果，也可以是横向排名的数据。在存储上所有品种的数据都存储在一个文件中。

修改自定义数据

名称: SINGLE1

描述:

类型: 单值数值 每只品种仅有一个固定的数值

分组: 分组1 数值小数位数: 3

数据自动维护(刷新,导入)...

与指标最新值相关联 设为默认 确定 取消

指标: OUTC.A: 日线, 使用2根数据计算 选择指标

范围: 沪深A股 设置范围

计算时使用复权后数据 价格反转 剔除停牌品种 剔除ST品种

生成横向排名数据 数据排序方法: 0-1000归一化排名降序

无有效数据时: 取前一周期数据

计算时间段

单值类型只计算最新时间数据

导出设置 导入设置

无有效数据指的是指标计算结果为无效的数据，以及横向统计时间对齐时的停牌缺数据两种。如果非横向统计的指标数值计算，停牌日期只会缺数据，而不会当做无效数据处理。

2、单值字符串

单值字符串类型，每只品种仅有一个固定的字符串。单值字符串也可以指标计

算相关联, 比如您可以在公式取到 BLKNAME 所属板块的字符串, 在公式中导出到该自定义数据中; 如果是数值型的指标结果, 系统也会将最新数值转成字符串进行存储。

3、品种相关序列值

品种相关序列值, 每只品种有一组时间序列数值。如果跟指标计算关联, 取指标计算结果的时间序列数值, 可以是指标的计算结果, 也可以是横向排名的数据。在存储上每个品种的数据各存储一个文件。

修改自定义数据

名称:

描述:

类型: 品种相关序列值 每只品种有一组时间序列数值

分组: 分组1 数值小数位数: 3

数据自动维护(刷新, 导入)...

与指标相关联 设为默认 确定 取消

指标: 选择指标

范围: 设置范围

计算时使用复权后数据 价格反转 剔除停牌品种 剔除ST品种

生成横向排名数据 数据排序方法: 1-N自然排名降序

无有效数据时: 取前一周期数据

计算时间段

指定起始日期 2018/ 1/ 1 - 最近

计算最近 1000 条数据

本地所有数据

历史时间段 2018/ 1/ 1 - 2022/ 2/17

横向统计序列时间基准: 000001 上证指数

选择

导出设置 导入设置

4、品种无关序列值

品种无关序列值, 与品种无关的一组时间序列数值。如果跟指标计算关联, 一般的使用场景是生成一组时间序列的横向统计结果数据。在存储上该组数据存储在一个文件上。

易得提供的横向统计方法有 12 种:

算术累加总和

算术平均

历史流通股加权平均
最新流通股加权平均
总股本加权平均
最大值
最小值
中位数
总体样本方差
估算样本方差
总体标准差
估算标准差

修改自定义数据

名称:

描述:

类型: 与品种无关的一组时间序列数值

分组: 数值小数位数:

与指标横向统计相关联

指标:

范围:

计算时使用复权后数据 价格反转 剔除停牌品种 剔除ST品种

横向统计方法:

无有效数据时:

计算时间段

指定起始日期 - 最近

计算最近 条数据

本地所有数据

历史时间段 -

横向统计序列时间基准:

三、常用操作

1、分组

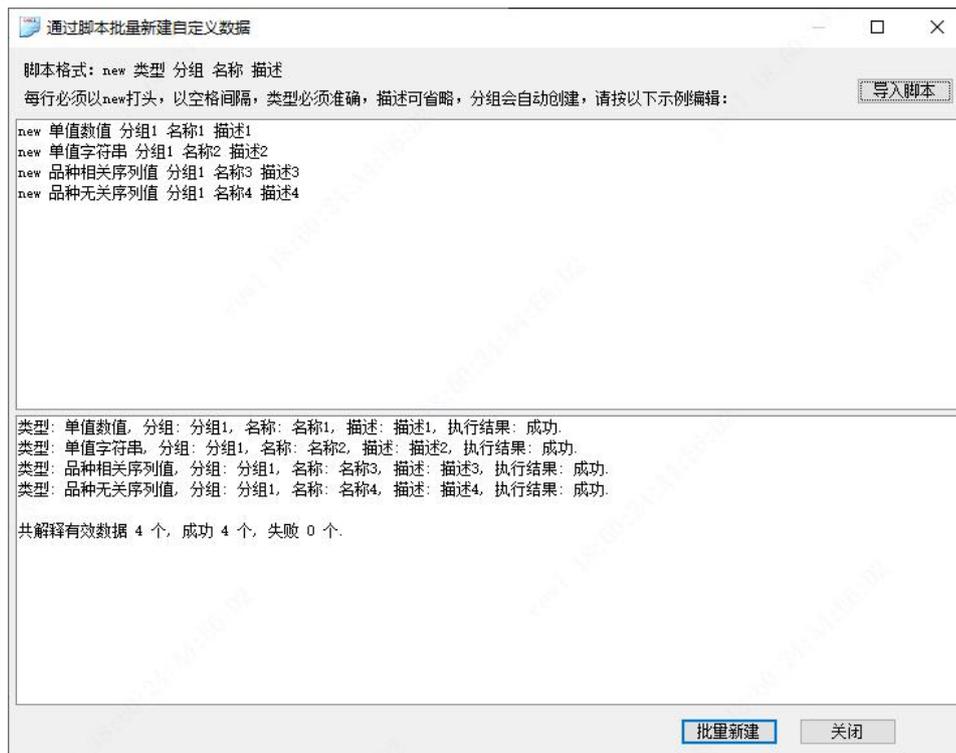
您可以在类型分组区树状结构中，使用右键菜单进行新建分组、删除分组、修改分组名等常规操作。可以通过鼠标拖拽移动自定义数据的所属分组。

2、批量新建自定义数据

批量新建可以根据规定的脚本格式，一键批量新建任意个自定义数据，以提高工作效率。

脚本格式：new 类型 分组 名称 描述

每行必须以 new 打头，以空格间隔，类型必须准确，描述可省略，分组会自动创建。



3、刷新历史、刷新当日

在树状结构点击自定义数据节点的右键菜单，我们可以对与指标关联的自定义数据进行刷新计算。刷新历史是对指定计算时间段的数据全部刷新计算；刷新当日只刷新当日的的数据。

如果在分组节点使用右键菜单进行刷新，将会批量刷新该分组下的所有自定义数据。

如果在数据类型点使用右键菜单进行刷新，将会批量刷新该数据类型下的所有自定义数据。

4、导出数据

在树状结构点击自定义数据节点的右键菜单，我们可以导出自定义数据。导出

的文件格式有两种：文本格式和系统格式。

文本格式优点是可使用文本编辑器进行编辑和阅读，缺点是存储较大读写较慢。

系统格式采用的是二进制格式存储。二进制格式的优点是存储较小读写较快，缺点是不可使用文本编辑器进行编辑和阅读。



自定义数据导出

自定义数据: SERIALSTK [品种相关序列值], 每个品种各导出一个文件

文件格式: 文本格式

导出位置: D:\YdSelfDataOut\SERIALSTK\ 选择路径

全部数据

时间段: 2022- 1- 1 - 2022- 2-19

确定 取消

5、导入数据

同样, 在树状结构点击自定义数据节点的右键菜单, 我们可以导入自定义数据。导入的文件格式也是有两种: 文本格式和系统格式, 跟导出的格式相同。

覆盖方式是以导入数据为准, 覆盖之前的数据。

补充方式是如果系统已存在该数据, 系统原有数据保持不变, 只补充系统不存在的新数据。



自定义数据导入

自定义数据: SERIALSTK [品种相关序列值], 每个品种各一个导入文件

文件格式: 文本格式

导入位置: D:\YdSelfDataOut\SERIALSTK\ 选择路径

清除原有数据 覆盖方式 补充方式

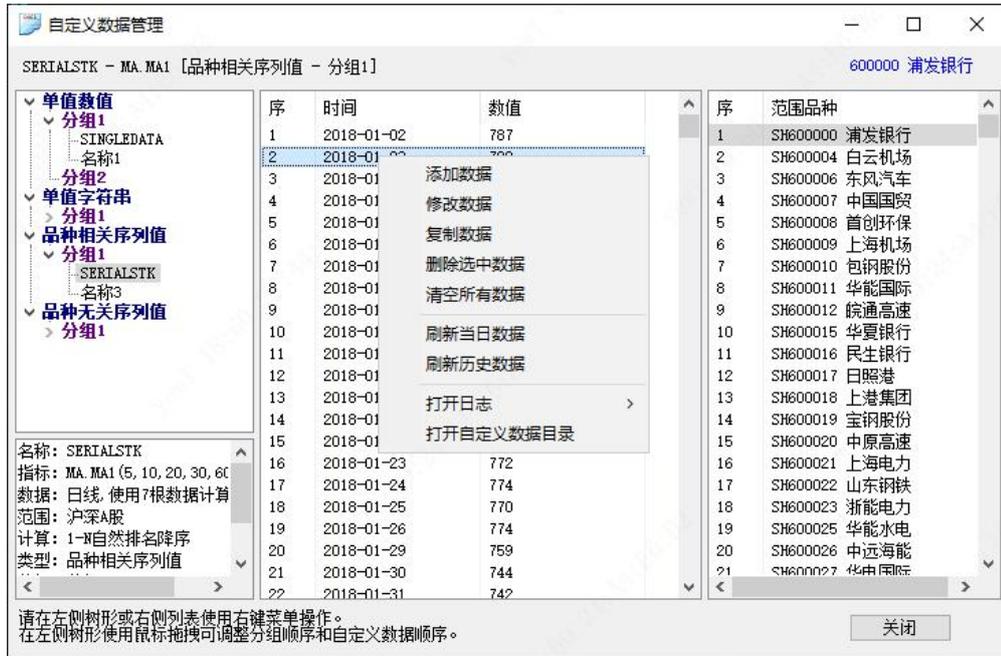
全部数据

时间段: 2022- 1- 1 - 2022- 2-19

确定 取消

6、数据维护

在右侧数据区，使用右键菜单我们可以对数据进行维护，可以增加、删除、修改、复制、清空等常规操作。



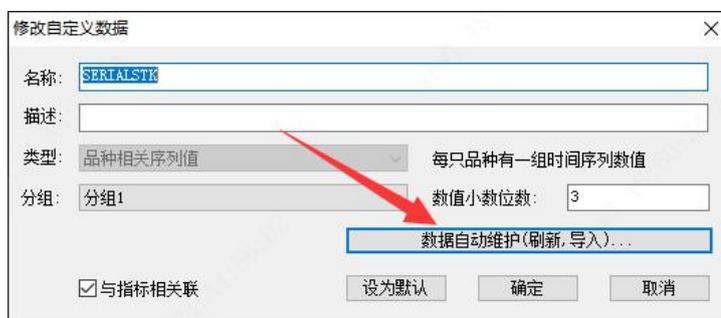
7、导出导入设置

导出导入设置可以将类型分组区选中节点下所有的自定义数据项的设置导出导入。方便用户对自定义数据迁移。



四、自动刷新

在自定义数据设置的对话框中，点击“数据自动维护”按钮。我们可以进行自动刷新和自动导入的设置。



见下图，自动刷新支持**后台多线程无感知刷新**、可以**打印自动刷新的日志**、可选择**盘中刷新最新一笔**还是**刷新当日**，但不能自动刷新历史全部数据。

自动刷新仅在指定范围的品种的**交易日才刷新**。

自动刷新频率有三种选择：

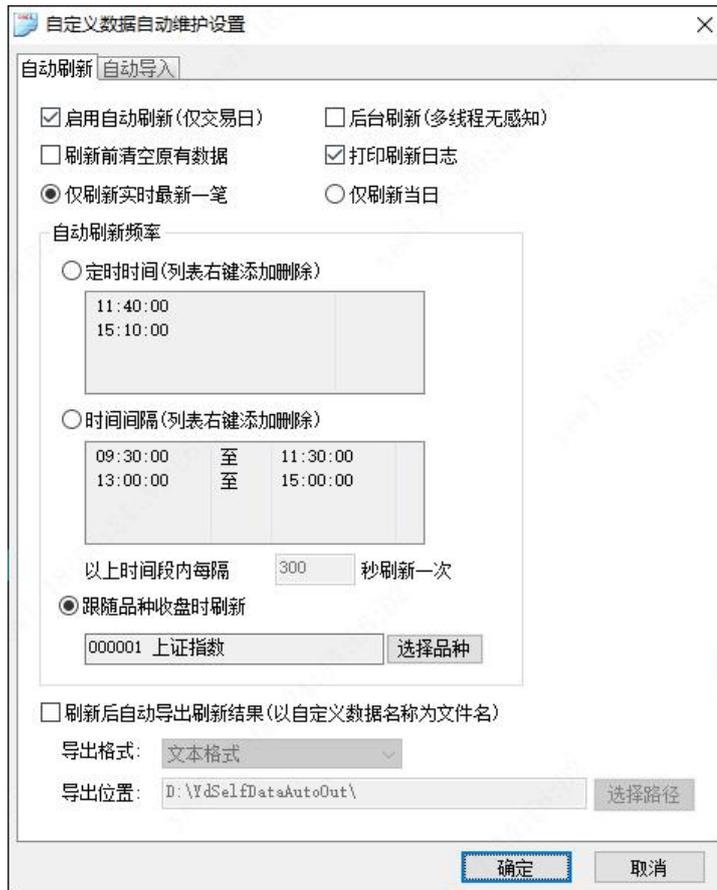
定时时间刷新：可以指定多个时间，各时间达到各执行一次。

时间间隔：可以指定多个时间段，在这些时间段内每隔多少秒刷新一次。

跟随品种收盘时刷新：指定一个品种，在此品种每日进行收盘后刷新一次。

每次刷新后还支持将**最新的刷新结果导出到文件**，**实现与外部数据交换**。支持文本

格式和系统格式两种，格式跟导出导入功能的文件格式相同。



五、批量设置自动刷新

您可以选择一个分组，使用右键菜单的“批量自动刷新设置”，对该分组的自定义数据进行批量的统一自动刷新设置。设置将对该分组下的所有自定义数据起效。



六、自动导入数据

在自定义数据设置的对话框中，点击“数据自动维护”按钮。我们可以进行自动导入的设置。

见下图，自动导入支持**后台多线程无感知**导入、可以**打印自动导入的日志**、可选择覆盖方式和补充方式导入。

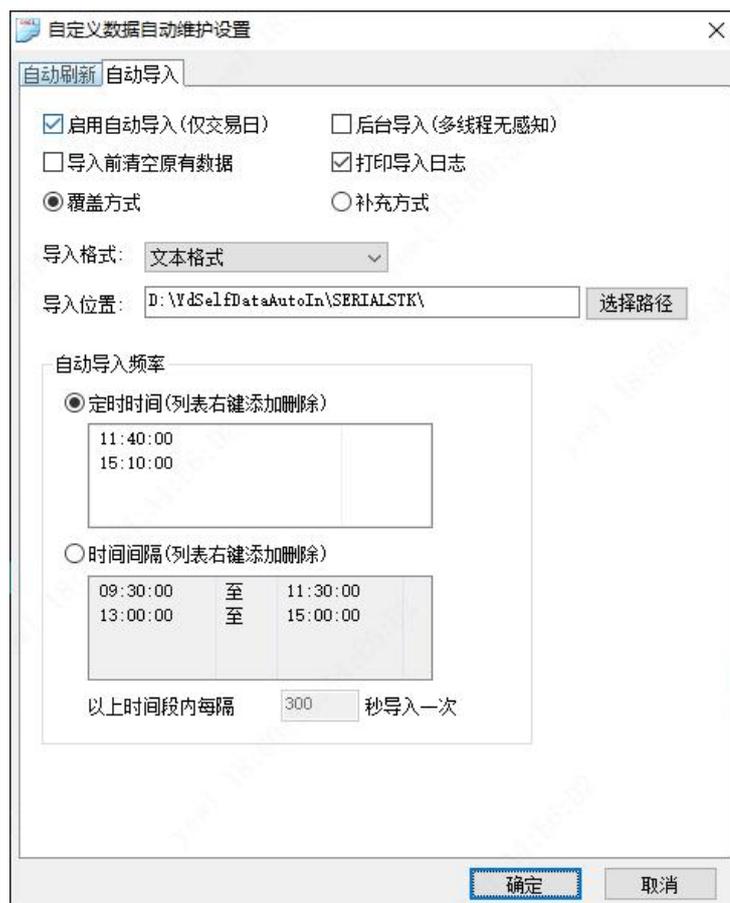
支持文本格式和系统格式两种，格式跟导出导入功能的文件格式相同。

自动导入仅在指定范围的品种的**交易日才导入**。

自动导入频率有两种选择：

定时时间导入：可以指定多个时间，时间达到执行一次。

时间间隔：可以指定多个时间段，在这些时间段内每隔多少个秒导入一次。



七、自定义数据排序

在动态显示牌，点击栏目抬头的右键菜单，可以选择自定义数据进行排序。

名称	最近	SINGLEDATA	涨跌	量比	涨速	
浦发银行	3R	8.56	8.69		0.00%	
白云机场	R	13.52	12.796		-0.15%	
东风汽车	R	6.18	6.704		-0.16%	
中国国贸	R	13.76	14.388		0.00%	
首创环保	R	3.16	3.406		0.00%	
上海机场	3R	52.00	49.206		0.00%	
包钢股份	3R	2.41	2.696		0.00%	
华能国际	3R	7.85	8.568		-0.13%	
皖通高速	R	7.04	7.168		0.00%	
华夏银行	3R	5.63	5.656		-0.18%	
民生银行	3R	3.90	3.922		0.00%	
日照港	R	2.71	2.834		-0.37%	
上港集团	3R	5.69	5.796		0.00%	
宝钢股份	3R	7.23	7.28		0.00%	
中原高速	R	3.13	3.26		0.00%	
上海电力	R	12.12	12.416		-0.08%	
山东钢铁	R	1.70	1.792		0.00%	
浙能电力	R	3.60	3.898		0.28%	
华能水电	3R	6.03	6.314	-0.16	0.89	0.00%

序	名称	指标	范围	计算方法
1	SINGLEDATA	MA.MA1(5,10,20...	沪深A股	指标数值
2	名称1			

八、自定义数据函数

自定义数据可以在公式系统中使用函数进行引用。

自定义数据目录
SelfDataPath
取得自定义数据目录。

<p>用法:</p> <p>SelfDataPath;</p> <p>返回自定义数据目录的字符串.</p>	
SelfDataFileName	自定义数据存储文件路径名
<p>取得自定义数据存储文件路径名。</p> <p>用法:</p> <p>SelfDataFileName(S, [T]),取得名为 S 的自定义数据的存储文件路径名, T 是自定义数据的类型, T 的取值按类型:</p> <p>单值数值为 0、单值字符串为 1、品种相关序列值为 2、品种无关序列值为 3; T 在 S 于四种类型没有重名的情况下可省略。</p> <p>对于品种相关序列值类型返回的是目录, 每个品种的数据是该目录下以 MarketLabel+StkLabel+'.sfd'命名的单个文件;其他数据类型返回的是文件路径名。</p>	
SelfData	自定义数据
<p>取得自定义数据。</p> <p>用法:</p> <p>SelfData(S, [T]),取得名为 S 的自定义数据, T 是自定义数据的类型, T 的取值按类型:</p> <p>单值数值为 0、单值字符串为 1、品种相关序列值为 2、品种无关序列值为 3; T 在 S 于四种类型没有重名的情况下可省略。</p> <p>比如: SelfData('AAA' , 2);</p> <p>取名称为 AAA,类型为品种相关序列值的自定义数据。</p> <p>该函数适用于任何数据周期。</p>	
SelfString	自定义字符串数据
<p>取得自定义字符串数据。</p> <p>用法:</p> <p>SelfString(S),取得名为 S 的自定义字符串数据</p> <p>例如:</p> <p>DrawTextAbs(2, 2, SelfString('所属板块'));</p> <p>输出命名为[所属板块]的自定义字符串数据。</p>	

九、文件接口函数

易得公式系统提供了一组直接访问自定义数据文件的函数。您可以使用这类函数直接读写文件，绕过创建自定义数据，方便用户从外部引入导出数据。

常用的应用场景是，您可以通过 Python 等脚本语言生成自定义数据导入导出格式的文件，在公式公式系统中直接读取此文件，将数据引入到量化交易中。

反之亦然，您也可以使用易得公式系统将公式计算结果不断写入指定文件，在外部由 Python 等脚本语言去读取该文件，从而实现数据的交换。

FileReadSingleDataTxt	读取文件 - 单值数值文本格式
<p>读取文件 - 单值数值文本格式</p> <p>用法: <code>FileReadSingleDataTxt(path, stock, real);</code></p> <p>参数 <code>path</code> 指定文件路径; <code>stock</code> 指定品种代码如 'SH600000'(MarketLabel+StkLabel), "" 空字符串表示当前品种; <code>real</code> 表示文件中的数据是否实时变动刷新, 如果实时刷新在盘中行情跳动都会重新读取一次文件, <code>real</code> 为 1 表示实时刷新, 为 0 表示不实时刷新, 不实时刷新在盘中不会重复读取文件, 性能更高。</p> <p>单值数值与自定义数据的单值数值类型含义相同, 每只股票一个数据。文本格式与自定义数据导出的文本格式相同。文本格式优点是可使用文本编辑器进行编辑和阅读, 缺点是存储较大读写较慢。</p> <p>举例:</p> <p><code>A:FileReadSingleDataTxt('D:\AAA.txt', "", 0);</code></p> <p>从 D:\AAA.txt 文件中按文本格式读取当前品种的单值数据, 盘中不重复读取; 读取结果放入变量 A。</p>	
FileReadSingleDataBin	读取文件 - 单值数值二进制格式
<p>读取文件 - 单值数值二进制格式</p> <p>用法: <code>FileReadSingleDataBin(path, stock, real);</code></p> <p>参数 <code>path</code> 指定文件路径; <code>stock</code> 指定品种代码如 'SH600000'(MarketLabel+StkLabel), "" 空字符串表示当前品种; <code>real</code> 表示文件中的数据是否实时变动刷新, 如果实时刷新在盘中行情跳动都会重新读取一次文件, <code>real</code> 为 1 表示实时刷新, 为 0 标识不实时刷新, 不实时刷新在盘中不会重复读取文件, 性能更高。</p> <p>单值数值与自定义数据的单值数值类型含义相同, 每只股票一个数据。二进制格式与自定义数据导出的系统格式相同。二进制的优点是存储较小读写较快, 缺点</p>	

是不可使用文本编辑器进行编辑和阅读。

举例：

```
A:FileReadSingleDataBin('D:\AAA.sfd', "", 0);
```

从 D:\AAA.sfd 文件中按二进制格式读取当前品种的单值数据，盘中不重复读取；读取结果放入变量 A。

FileWriteSingleDataTxt

写入文件 - 单值数值文本格式

写入文件 - 单值数值文本格式

用法：FileWriteSingleDataTxt(path, stock, val);

参数 path 指定文件路径；stock 指定品种代码如 'SH600000'(MarketLabel+StkLabel)，"" 空字符串表示当前品种；val 表示写入变量的最新数值。该函数无返回值。

单值数值与自定义数据的单值数值类型含义相同，每只股票一个数据。文本格式与自定义数据导出的文本格式相同。文本格式优点是可使用文本编辑器进行编辑和阅读，缺点是存储较大读写较慢。

举例：

```
A:MA(C, 5);
```

```
FileWriteSingleDataTxt('D:\AAA.txt', "", A);
```

将当前品种 5 日均线的最新值按文本格式写入 D:\AAA.txt 文件。

FileWriteSingleDataBin

写入文件 - 单值数值二进制格式

写入文件 - 单值数值二进制格式

用法：FileWriteSingleDataBin(path, stock, val);

参数 path 指定文件路径；stock 指定品种代码如 'SH600000'(MarketLabel+StkLabel)，"" 空字符串表示当前品种；val 表示写入变量的最新数值。该函数无返回值。

单值数值与自定义数据的单值数值类型含义相同，每只股票一个数据。二进制格式与自定义数据导出的系统格式相同。二进制的优点是存储较小读写较快，缺点是不可使用文本编辑器进行编辑和阅读。

举例：

```
A:MA(C, 5);
```

```
FileWriteSingleDataBin('D:\AAA.sfd', "", A);
```

将当前品种 5 日均线的最新值按二进制格式写入 D:\AAA.sfd 文件。

FileReadSingleStringTxt	读取文件 - 单值字符串文本格式
读取文件 - 单值字符串文本格式 用法: FileReadSingleStringTxt(path, stock); 参数 path 指定文件路径; stock 指定品种代码如 'SH600000'(MarketLabel+StkLabel), "" 空字符串表示当前品种。 单值字符串与自定义数据的单值字符串类型含义相同, 每只股票一个字符串。文本格式与自定义数据导出的文本格式相同。文本格式优点是可使用文本编辑器进行编辑和阅读, 缺点是存储较大读写较慢。 举例: A:FileReadSingleStringTxt('D:\AAA.txt', ''); 从 D:\AAA.txt 文件中按文本格式读取当前品种的单值字符串; 读取结果放入变量 A。	
FileReadSingleStringBin	读取文件 - 单值字符串二进制格式
读取文件 - 单值字符串二进制格式 用法: FileReadSingleStringBin(path, stock); 参数 path 指定文件路径; stock 指定品种代码如 'SH600000'(MarketLabel+StkLabel), "" 空字符串表示当前品种。 单值字符串与自定义数据的单值字符串类型含义相同, 每只股票一个字符串。二进制格式与自定义数据导出的系统格式相同。二进制的优点是存储较小读写较快, 缺点是不可使用文本编辑器进行编辑和阅读。 举例: A:FileReadSingleStringBin('D:\AAA.sfd', ''); 从 D:\AAA.sfd 文件中按二进制格式读取当前品种的单值字符串; 读取结果放入变量 A。	
FileWriteSingleStringTxt	写入文件 - 单值字符串文本格式
写入文件 - 单值字符串文本格式 用法: FileWriteSingleStringTxt(path, stock, val); 参数 path 指定文件路径; stock 指定品种代码如 'SH600000'(MarketLabel+StkLabel), "" 空字符串表示当前品种; val 表示写入的字符串。该函数无返回值。	

单值字符串与自定义数据的单值字符串类型含义相同，每只股票一个字符串。文本格式与自定义数据导出的文本格式相同。文本格式优点是可使用文本编辑器进行编辑和阅读，缺点是存储较大读写较慢。

举例：

```
A:BLKNAME;
```

```
FileWriteSingleStringTxt('D:\AAA.txt', '', A);
```

将当前品种的所属板块字符串按文本格式写入 D:\AAA.txt 文件。

FileWriteSingleStringBin 写入文件 - 单值字符串二进制格式

写入文件 - 单值字符串二进制格式

用法：FileWriteSingleStringBin(path, stock, val);

参数 path 指定文件路径；stock 指定品种代码如 'SH600000'(MarketLabel+StkLabel)，"" 空字符串表示当前品种；val 表示写入的字符串。该函数无返回值。

单值字符串与自定义数据的单值字符串类型含义相同，每只股票一个字符串。二进制格式与自定义数据导出的系统格式相同。二进制的优点是存储较小读写较快，缺点是不可使用文本编辑器进行编辑和阅读。

举例：

```
A:BLKNAME;
```

```
FileWriteSingleStringBin('D:\AAA.sfd', '', A);
```

将当前品种的所属板块字符串按二进制格式写入 D:\AAA.sfd 文件。

FileReadSerialDataTxt 读取文件 - 时间序列数值文本格式

读取文件 - 时间序列数值文本格式

用法：FileReadSerialDataTxt(path, IntraDay, real);

参数 path 指定文件路径；IntraDay 表示该数据的时间是否是带有时分秒的日内数据，1 表示日内数据，0 表示日频级别以上数据；real 表示文件中的数据是否实时变动刷新，如果实时刷新在盘中行情跳动都会重新读取一次文件，real 为 1 表示实时刷新，为 0 标识不实时刷新，不实时刷新在盘中不会重复读取文件，性能更高。

时间序列数值与自定义数据的品种相关序列值类型和品种无关序列值类型含义相同，是一组时间序列数值的数据。文本格式与自定义数据导出的文本格式相同。文本格式优点是可使用文本编辑器进行编辑和阅读，缺点是存储较大读写较慢。

举例：

```
path:='D:\'+MarketLabel+StkLabel+'.txt';
```

```
A:FileReadSerialDataTxt(path, 0, 0);
```

从 path 文件中按文本格式读取时间序列数据, 该数据不是日内数据; 盘中不重复读取; 读取结果放入变量 A。

注意: 文件中的时间序列必须是从小到大有序的。

FileReadSerialDataBin

读取文件 - 时间序列数值二进制格式

读取文件 - 时间序列数值二进制格式

用法: FileReadSerialDataBin(path, IntraDay, real);

参数 path 指定文件路径; IntraDay 表示该数据的时间是否是带有时分秒的日内数据, 1 表示日内数据, 0 表示日频级别以上数据; real 表示文件中的数据是否实时变动刷新, 如果实时刷新在盘中行情跳动都会重新读取一次文件, real 为 1 表示实时刷新, 为 0 标识不实时刷新, 不实时刷新在盘中不会重复读取文件, 性能更高。

时间序列数值与自定义数据的品种相关序列值类型和品种无关序列值类型含义相同, 是一组时间序列数值的数据。二进制格式与自定义数据导出的系统格式相同。二进制的优点是存储较小读写较快, 缺点是不可使用文本编辑器进行编辑和阅读。

举例:

```
path:='D:\'+MarketLabel+StkLabel+'.sfd';
```

```
A:FileReadSerialDataBin(path, 0, 0);
```

从 path 文件中按二进制格式读取时间序列数据, 该数据不是日内数据; 盘中不重复读取; 读取结果放入变量 A。

注意: 文件中的时间序列必须是从小到大有序的。

FileWriteSerialDataTxt

写入文件 - 时间序列数值文本格式

写入文件 - 时间序列数值文本格式

用法: FileWriteSerialDataTxt(path, val);

参数 path 指定文件路径; val 表示写入的序列数值。该函数无返回值。

时间序列数值与自定义数据的品种相关序列值类型和品种无关序列值类型含义相同, 是一组时间序列数值的数据。文本格式与自定义数据导出的文本格式相同。文本格式优点是可使用文本编辑器进行编辑和阅读, 缺点是存储较大读写较慢。

举例:

```
A:MA(C, 5);
```

```
FileWriteSerialDataTxt('D:\AAA.txt', A);
```

将 5 日均线的序列值按文本格式写入 D:\AAA.txt 文件。

FileWriteSerialDataBin

写入文件 - 时间序列数值二进制格式

写入文件 - 时间序列数值二进制格式

用法: FileWriteSerialDataBin(path, val);

参数 path 指定文件路径; val 表示写入的序列数值。该函数无返回值。

时间序列数值与自定义数据的品种相关序列值类型和品种无关序列值类型含义相同, 是一组时间序列数值的数据。二进制格式与自定义数据导出的系统格式相同。二进制的优点是存储较小读写较快, 缺点是不得使用文本编辑器进行编辑和阅读。

举例:

```
A:MA(C, 5);
```

```
FileWriteSerialDataBin('D:\AAA.sfd', A);
```

将 5 日均线的序列值按二进制格式写入 D:\AAA.sfd 文件。

十、导出导入文件结构

下面从易得自定义数据的四种类型分别展开介绍导出导入文件的格式。其中每种文件又有文本格式和二进制格式两种, 系统存储使用的是二进制格式。

文本格式优点是可使用文本编辑器进行编辑和阅读, 缺点是存储较大读写较慢。二进制格式的优点是存储较小读写较快, 缺点是不得使用文本编辑器进行编辑和阅读。

如果需要从外部导入数据到易得自定义数据系统, 了解其文件结构是非常必要的。二进制结构比较复杂, 一般用户了解文本格式即可。

1、单值数值文本格式

单值数据的文本格式非常简单, 市场代码+品种代码=数据, 每条数据一行。

市场代码定义如下:

SH: 上海证券交易所

SZ: 深圳证券交易所

ZJ: 中金所

SQ: 上期所

IN: 上期能源

DQ: 大商所

ZQ: 郑商所

B\$: 板块指数市场

文件格式如下：

```
SH600000=649.777  
SH600004=446.205  
SH600006=779.911
```

2、单值字符串文本格式

单值字符串的文本格式也非常相似，市场代码+品种代码=字符串，每条数据一行。格式如下：

```
SH600000=银行  
SH600004=航空机场  
SH600006=汽车整车
```

3、时间序列值文本格式

时间序列值的文本格式适用于品种相关序列值和品种无关序列值，都是一个时间一个数值。注意时间序列文件的时间必须是从小到大有序的。

其中日期的格式为: YYYY-MM-DD

日内时间格式为： YYYY-MM-DD HH:MM:SS 年月日跟时分秒用空格间隔开。

日频级别时间格式：

```
2021-09-08=9.34  
2021-09-09=9.3  
2021-09-10=9.41
```

日内时间格式：

```
2021-01-21 09:35:00=31047.768  
2021-01-21 09:40:00=31090.265  
2021-01-21 09:45:00=31073.61
```

4、单值数值二进制格式

单值数值二进制格式由任意个固定大小的结构体组成，每个结构体存储一个数据项。文件长度除以结构体大小就是数据项的个数。

每个结构体由 12 个字节长度的品种代码和 8 个字节的双精度浮点数组成。在 C 语言中结构定义为：

```
#pragma pack(1)
struct _SINGLE_DATA
{
    char  stkunilabel[12];
    double value;
};
#pragma pack()
```

5、单值字符串二进制格式

单值字符串二进制格式稍微复杂一点。文件头由一个 4 字节的标识号和 4 个字节的数据项个数组成。每个数据项有 2 个字符串组成，分别表示品种代码和字符串数据。

在 C 语言中结构定义为：

```
#pragma pack(1)
struct _SINGLE_STR_HEAD    // 文件头
{
    unsigned int tag; // 文件标识号，必须等于 0x234598eb
    unsigned int num; // 数据项个数
};

struct _SINGLE_STR_ITEM // 单个数据项
{
    string stkunilabel;
    string value;
};
#pragma pack()
```

数据项的存储按文件流的方式存储，字符串的存储是先写入 4 个字节的整型说明字

字符串的长度，后边跟着字符串的内容。以此类推。

6、时间序列值二进制格式

时间序列值二进制格式适用于品种相关序列值和品种无关序列值，都是一个时间一个数值。注意时间序列文件的时间必须是从小到大有序的。

时间序列值二进制格式由任意个固定大小的结构体组成，每个结构体存储一个数据项。文件长度除以结构体大小就是数据项的个数。

每个结构体由 4 个字节长度的时间和 8 个字节的双精度浮点数组组合。其中时间是按东八区从 1970 年 1 月 1 日 0 点开始的秒数，在 C 语言中结构定义为：

```
#pragma pack(1)
struct _SERIAL_DATA
{
    unsigned int time; // 1970.1.1 开始的秒数(UTC+8)
    double      value;
};
#pragma pack()
```

十一、自定义数据目录结构

自定义数据的目录位于安装路径下的“.\UserData\SelfData\”目录，里面有命名为 0、1、2、3 的文件夹分别代表了单值数值、单值字符串、品种相关序列值、品种无关序列值四种数据类型的存储所在。

文件名就是自定义数据名；如果是品种相关序列值，按自定义数据名生成一个目录，目录下在保存了每个品种的序列数据。其中.sfi 格式文件自定义数据属性信息，数据文件的存储后缀名为.sfd。

注意：因为在存储上文件以自定义数据名来命名，所以自定义数据的名称不能包含有\:*?/" <>|这些字符，因为操作系统的文件名不能包含这些字符。

如果您想要对自定义数据进行迁移，直接对拷相应目录下的所有文件即可。